
Schlussbericht

für das Vorhaben

First Look – Fast Initial In-orbit Identification of Scientific Satellites

(Förderkennzeichen 50 OY 0502)



ZARM, Universität Bremen
Hochschulring / Am Fallturm
28359 Bremen


Ralf Keil Tel.: 0421-218-9422
Matthias Maat Tel.: 0421-218-4187
Fax: 0421-218-2521

Dezember 2009



Inhaltsverzeichnis

1	Einleitung	3
2	Aufgabenstellung	4
2.1	Motivation	4
2.2	Stand der Forschung	5
2.3	Zusammenarbeit mit anderen Stellen	5
2.4	Zielsetzung	5
3	Planung und Ablauf des Vorhabens	7
3.1	AP 1000: Durchsicht der „First Look“Aktivitäten	7
3.2	AP 2000: Interdisziplinäre Durchmusterung	8
3.3	AP 3000: Verbesserung des globalen „First Look“	8
3.4	AP 4000: Anwendung auf Referenz-Missionen	8
4	Ergebnisse der Arbeitspakete	9
4.1	AP 1000	9
4.2	AP 2000	10
4.3	AP 3000	11
4.4	AP 4000	13
5	Zusammenfassung	15
6	Anhang	16

	Schlussbericht First Look - Fast Initial In-Orbit Identification of Scientific Satellites Fkz: 50 OY 0502	
---	--	--

1 Einleitung

Das vorliegende Dokument bildet den Schlussbericht für das Forschungsvorhaben 50 OY 0502

First Look - Fast Initial In-Orbit Identification of Scientific Satellites

Der Bericht ist in fünf Kapitel gegliedert plus einem Anhang. Die hier vorliegende Einleitung bildet das erste Kapitel.

Im zweiten Kapitel werden die Aufgabenstellung und die damit verbundene Zielsetzung des Forschungsvorhabens motiviert.

Gegenstand des dritten Kapitels ist die Struktur der Arbeitspakete und deren zeitliche Planung. Die in den jeweiligen Arbeitspaketen erzielten Ergebnisse werden dann im vierten Kapitel vorgestellt. Zu einem gewissen Teil machten die Resultate Anpassungen des weiteren Ablaufs erforderlich, die hier an entsprechender Stelle erläutert werden.

Kapitel fünf beschließt diesen Bericht mit einer Zusammenfassung der erbrachten Leistungen und einer Bewertung ihres Nutzens für weitere gegenwärtige und zukünftige Forschungsaktivitäten.

Während des Forschungsvorhabens wurden die erarbeiteten Ergebnisse in Form von *Technical Notes (TN)* dokumentiert, die dem Bericht vollständig im Anhang beigelegt wurden. Diese enthalten detaillierte Beschreibungen der durchgeführten Arbeiten.



2 Aufgabenstellung

2.1 Motivation

Ziel des Projekts war die Entwicklung von Algorithmen und Modellen zur In-Orbit-Identifikation von Satelliten und Experimentparametern bei wissenschaftlichen Missionen, um damit die Qualität wissenschaftlicher Ergebnisse sicherzustellen.

Mit der voranschreitenden Entwicklung von Technologien verwenden Wissenschaftsmissionen immer sensiblere Sensoren, die eine extrem störungsfreie Umgebung benötigen. Beispiele hierfür sind Gaia, STEP oder die Gravity Probe B Mission (GP-B), dessen Flug- und Messphase kurz nach Beginn des Projektes geendet hatte.

Infolge dieser Entwicklung sind zum einen die Messgeräte sehr stark mit dem Satellitenbus verbunden. Zum anderen benötigt die Prozessierung der Messdaten einen langen Zeitraum, ehe die wissenschaftlichen Ergebnisse vorliegen. Bei GP-B hat es etwa ein Jahr gedauert, bis die Daten verfügbar waren. Das Endergebnis von Gaia wird erst einige Jahre nach Ende der Mission verfügbar sein.

Es stellt sich die Frage, wie man zu Beginn einer Mission gewährleisten kann, dass die heute aufgenommenen Messwerte nicht mit unbekanntem nicht-detektierbaren Fehlern behaftet sind, die das Endergebnis in mehreren Jahren verfälschen. Eine Möglichkeit, dies sicher zu stellen, ist die schnelle Prozessierung der ersten Messdaten, um das Systemverhalten von Satellit und Instrument(en) sowie deren Parameter zu identifizieren. Diese Identifikation ermöglicht es, ein umfassendes Wissen über Satellit, Instrumente und deren Verhalten aufzubauen. Damit wird es möglich, die Qualität der wissenschaftlichen Daten kontinuierlich zu überwachen und auf Fehler bzw. Parameterveränderungen zu reagieren.

Für die o.g. Missionen besteht die Notwendigkeit, Methoden zur schnellen In-Orbit-Identifikation zu entwickeln, die sehr schnell erkennen können, ob und wie das Gesamtsystem funktioniert. Dies hat sich gerade bei der GP-B Mission gezeigt. Hier musste ein Mehraufwand während der Kalibrationsphase betrieben werden, der nicht eingeplant war und Zusatzkosten in Millionenhöhe verursacht hat.

Durch die enge Verbindung zwischen Satellit und Instrument liegt das Arbeitsgebiet der Entwicklung solcher Identifikationsmethoden in der Nische zwischen zwei Disziplinen: der wissenschaftlichen Datenauswertung und der Satellitentechnologie. Aus diesem Grund muss das Wissen von beiden Seiten (Wissenschaft und Technologie) in die Entwicklung einfließen.



2.2 Stand der Forschung

Zu Beginn des Forschungsvorhabens gab es nur wenige Missionen, die umfassend, d.h. einschließlich der Satellitendynamik, identifiziert worden waren. Bekanntestes Beispiel hierfür ist die HIPPARCOS-Mission. Die Vermessung von Sternpositionen aus der Lage des Satelliten hat eine genaue Identifikation des Satelliten erfordert, ohne die eine wissenschaftliche Datenauswertung nur teilweise bzw. mit geringerer Genauigkeit möglich gewesen wäre.

Ein weiteres Beispiel ist die GP-B Mission, die die Messung und Untersuchung zweier allgemein-relativistischen Effekte zum Ziel hatte: die des "Frame-Dragging-" und des "Geodetic"-Effektes. Die Flug- und Messphase hatte erst kurz nach Beginn des Projektes geendet und befand sich währenddessen in der Phase der Datenauswertung. Bei dieser Mission wurden bereits speziell für diese entwickelte Verfahren angewendet. In Bezug auf die besonderen Eigenschaften der Mission war dies einmalig, da GP-B bis dahin die einzige wissenschaftliche Satellitenmission mit Drag-Free-Regelungssystem war.


2.3 Zusammenarbeit mit anderen Stellen

Die Arbeiten an HIPPARCOS und Gaia wurden in enger Abstimmung mit dem Institute of Astronomy (IoA) der University of Cambridge (UK) durchgeführt, die bei der Planung und Umsetzung beider Missionen erheblich beteiligt war bzw. ist. Bezüglich GP-B und STEP trifft dies in gleicher Weise auf das Hansen Experimental Physics Laboratory (HEPL) der Stanford University (USA) zu, die ebenfalls für eine Zusammenarbeit gewonnen werden konnte. Somit konnten die Erfahrungen aus den Missionen aus erster Hand und von vornherein in das Projekt einfließen.

2.4 Zielsetzung

Ziel des Vorhabens war die Ableitung und Entwicklung von allgemeinen Verfahren für die schnelle In-Orbit-Identifikation von wissenschaftlichen Satelliten. Im Rahmen des Vorhabens sollte auch die Anwendbarkeit und Anpassung gezeigt werden. Dazu wurden aus den allgemeinen Verfahren auf einzelne Missionen zugeschnittene spezielle Methoden abgeleitet. Im Einzelnen wurden die Missionen GP-B, Gaia und STEP betrachtet.

Mit dem o.g. Hauptziel des Vorhabens waren weitere technische und wissenschaftliche Ziele verbunden. Diese waren unter anderem damit verknüpft, dass für die Identifikationsverfahren Modelle entwickelt und in einer Simulation umgesetzt werden mussten. Hier ist die Kooperation mit den Partnern in Cambridge und Stanford von immenser

	<p>Schlussbericht First Look - Fast Initial In-Orbit Identification of Scientific Satellites Fkz: 50 OY 0502</p>	
---	--	--

Bedeutung, da beiderorts Daten vorhanden waren, mit denen die Simulation, die Identifikationsmodelle als auch die Identifikationsverfahren in ihrer Gesamtheit validiert werden sollten. Daraus abgeleitet ergaben sich im wesentlichen die folgenden Ziele:

1. die Schaffung einer hochgenauen Simulationsumgebung, in die die Erfahrungen der Projektpartner aus früheren Missionen einfließen,
2. die Validierung dieses Simulators und der enthaltenen Modelle mit Hilfe der bei den Partnern vorhandenen Missionsdaten, und
3. die Erweiterung und Anwendung dieser Methoden auf zukünftige Missionen.

3 Planung und Ablauf des Vorhabens

Das folgende Kapitel enthält eine kurze Zusammenfassung der Aufgaben, die den einzelnen Arbeitspaketen zugeordnet sind. Insgesamt war das Fördervorhaben in vier Hauptabschnitte mit den Bezeichnungen AP 1000 bis AP 4000 eingeteilt. Die genaue Aufteilung der thematischen Ablaufs zeigt Abbildung 1.

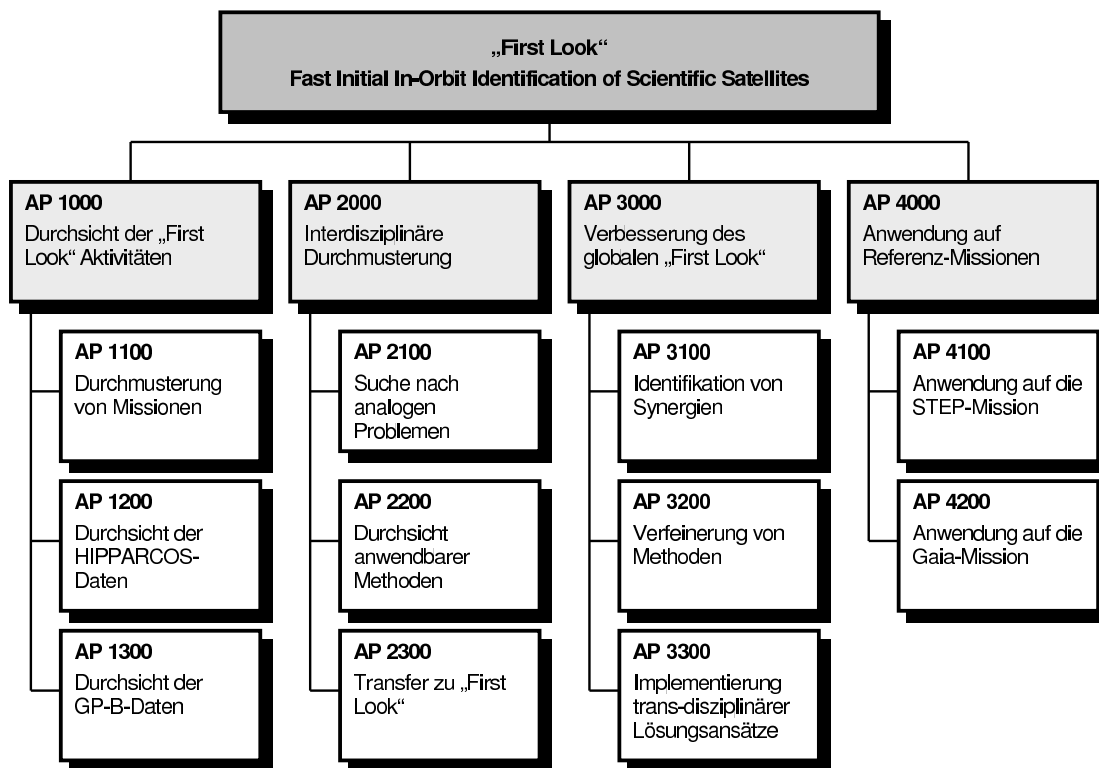


Abbildung 1: Arbeitsaufteilung des Vorhabens

Das Vorhaben wurde auf eine Laufzeit von 48 Monaten, mit Start im Juli 2005 und Ende im Mai 2009, ausgelegt. Die zeitliche Einordnung und Verteilung der Arbeitspakete ist aus Abbildung 2 ersichtlich.

3.1 AP 1000: Durchsicht der „First Look“ Aktivitäten

Dieses Paket beinhaltet die Aufarbeitung der Daten und Erfahrungen vergangener Missionen, speziell der HIPPARCOS Mission, die Vorgängermission zu Gaia, und der GP-B Mission, in der Techniken eingesetzt wurden, die auch für den STEP-Satelliten in ähnlicher und verbesserter Form zur Anwendung kommen sollen.




4 Ergebnisse der Arbeitspakete

4.1 AP 1000

Eine Bestandsaufnahme verschiedener Satellitenmissionen hat gezeigt, dass neben den bereits genannten Missionen HIPPARCOS, Gaia, GP-B und STEP Planungen einer Reihe weiterer Missionen existieren, bei denen eine In-orbit-Identifikation der Daten erforderlich ist oder werden könnte. Dazu gehören beispielsweise die MICROSCOPE-Mission, die wie STEP das Testen des Äquivalenzprinzips zum Ziel hat, die LISA-Mission zur Messung von Gravitationswellen oder LISA-Pathfinder, die eine technologische Demonstration von LISA darstellt. Die Durchmusterung ergab weiterhin, dass es sehr viele Gemeinsamkeiten zwischen all diesen Missionen gibt, die sich durch gleiche Modelle und Methoden beschreiben lassen.

Die Veröffentlichung der Ergebnisse von der GP-B Mission, welche für eine Weiterverarbeitung innerhalb des „First Look“-Projektes hätten benutzt werden sollen, wurde auf Grund zweier in ihrer Charakteristik unvorhergesehener Effekte in den Daten immer wieder verschoben. Zum einen sah man ein leichtes, kreisförmiges Schwanken des Experimentes, verursacht durch einander überlagernde Ungenauigkeiten in den Gyroskopen, mit einer zusätzlichen allmählichen Verlangsamung ihrer Drehbewegungen. Zum anderen entdeckte man ein Drehmoment an den Gyroskopen, welches erst nach Ende der Datennahme entdeckt wurde. Dieses entstand, als das Teleskop des Satelliten zeitweilig den Referenz-Stern nicht mehr im Gesichtsfeld hatte. Es ist möglicherweise mit kleinen elektrische Feldern assoziiert, die zwischen einigen Satellitenkomponenten auftreten. Eine schnelle Fehleranalyse ergab lediglich die Erkenntnis, dass das Problem ziemlich tief liegend wäre. Die Erfahrungen aus dieser Situation können zumindest helfen, zukünftige Missionen mit einer „Drag-Free“-Technologie besser vorzubereiten, insbes. durch die Entwicklung und Verwendung einer Simulationssoftware, welche die gekoppelte Satelliten-Nutzlast-Dynamik widerspiegeln kann, um das in-situ Betriebsverhalten des integrierten Satelliten abzubilden. Des Weiteren ist GP-B ein gutes Beispiel für die enge Kooperation zwischen Ingenieuren und Wissenschaftlern, einen optimalen Betrieb der wissenschaftlichen Instrumente zu realisieren.

Für Gaia konnte früher als erwartet mit dem Aufbau eines Simulators begonnen werden. Das hatte zwei wesentliche Gründe: Erstens hatte der Projekt-Partner vom Institute of Astronomy (IoA) zu Beginn des Jahres 2007 eine sehr umfangreiche Analyse des Datenmaterials der HIPPARCOS-Mission veröffentlicht, die wegen der 10-jährigen Arbeit daran wesentlich detaillierter durchgeführt worden war als noch im Arbeitspaket AP 1200 vorgesehen. Eine erneute Prozessierung der Daten innerhalb dieses Projektes sah man daher übereinstimmend als nicht mehr erforderlich an. Zweitens gewannen die Pläne für die Umsetzung der Gaia-Mission sehr schnell an Substanz, so dass die notwendigen

	Schlussbericht First Look - Fast Initial In-Orbit Identification of Scientific Satellites Fkz: 50 OY 0502	
---	--	--

Aufgaben, die im Rahmen von „First Look“ geleistet werden sollten, von der Gemeinschaft der Gaia-Projektmitglieder zügig konkretisiert wurden.

4.2 AP 2000

Für den Beginn der interdisziplinären Suche mussten zunächst die Kriterien definiert werden, die die im Projekt zu untersuchenden Problemstellungen kennzeichnen. In Kapitel 2 wurden die Probleme, die sich bei wissenschaftlichen Missionen ergeben, dargelegt. Daraus lassen sich die folgenden Kriterien ableiten:

1. Die Satelliten gehören zu einer neuen Generation wissenschaftlicher Satelliten, die Messdaten mit sehr viel höherer Genauigkeit als früher erbringen,
2. die Messung nimmt Einfluß auf die Dynamik des Satelliten, und
3. die Prozessierung der Messdaten benötigt im Vergleich zur Dauer der Mission eine sehr lange Zeit.

Diese Kriterien lassen sich allgemeiner fassen, indem man sich nicht speziell auf Satellitensysteme beschränkt. Allgemein sucht man nach Problemen,

1. die durch den Gebrauch neuer Technologien in komplexen Systemen verursacht werden,
2. bei denen die Dynamik des Messprozesses mit den Eigenschaften und der Dynamik der Apparatur verbunden sind, und
3. bei denen die Messwerte erst lange Zeit nach der Messphase verfügbar sind.

Hinzu kommt noch ein weiterer Punkt. Bei Satelliten-Missionen können bestimmte Kalibrierungen erst nach dem Start durchgeführt werden. Veränderungen an Bord sind jedoch dann nicht mehr möglich. Daher gilt als viertes Kriterium

4. Die Umstände des Problems gestatten keine angemessene Kalibration, weder bevor noch während der Zeit, in der das System in Betrieb ist, da dann keine Zugriff mehr für Modifikationen vorhanden ist.

In die Suche nach analogen Problemen wurden jene Forschungsfelder einbezogen, in denen häufig das Zusammenspiel komplexer Systeme einen Teil des Problems ausmacht und analysiert werden muss. Die Liste umfasst die folgenden größeren Fachgebiete:



- Luft- und Raumfahrttechnik
- Marinetechnik
- Fahrzeugbau
- Physik
- Chemie
- Naturwissenschaften

Allgemein waren die gefundenen Ergebnisse rar. Es konnte kein analoges Problem gefunden werden, dass *alle* vier definierten Kriterien erfüllte. Insofern verlief die Durchmusterung negativ. Jedoch hat die Suche ergeben, dass das Problem der Identifizierung dynamischer Probleme ein weit Verbreitetes ist. Es tritt bei allen Experimenten auf, bei denen die experimentellen Parameter vor der Messphase nicht zur Gänze bekannt sind. Die angewandten Lösungsstrategien lieferten für „First Look“ keine neuen Methoden. Am Anfang steht ein Prototyp, dessen Dynamik durch Modelle und Annahmen beschrieben wird. Die damit erbrachten Messungen lassen Rückschlüsse auf das System zu, mit denen das Modell verbessert werden kann, die jedoch erst bei zukünftigen Entwicklungen berücksichtigt werden können. Dieser Ansatz wurde auch bei „First Look“ verfolgt.

4.3 AP 3000

Wie sich in AP 1000 gezeigt hat, konnte man für eine ganze Reihe unterschiedlicher Missionen Gemeinsamkeiten erkennen. Es bot sich daher an, die für die Simulation zu entwickelnden Modelle möglichst allgemein zu formulieren, so dass man sie auf eine Vielzahl von Missionen direkt anwenden kann. Sinnvoll ist die Aufteilung in folgende Modelltypen:

- Dynamische Modelle für Satelliten und Testmassen,
- Umweltmodelle quantifizieren den Zustand der Weltraumumgebung, in denen der Satellit fliegt, und
- Störungsmodelle, die jene Kräfte und Drehmomente auf den Satelliten beschreiben, die durch den Einfluss der Weltraumumgebung hervorgerufen werden.

In praktischer Hinsicht war die beste Möglichkeit, diese Synergien zu bündeln, die Entwicklung und der Aufbau einer Modellbibliothek als Bestandteil eines Simulatorprogramms. Als Resultat ist der generische *High Performance Simulator (HPS)* entstanden, der auf den Programmpaketen MATLAB/Simulink basiert. Der generische Ansatz



war dabei sehr wichtig, da nur so die Anwendbarkeit für möglichst viele verschiedene Missionen gewährleistet werden kann. Eine der Hauptentwicklungsarbeiten lag dabei in der Umsetzung mathematischer Zusammenhänge, die den Modellen zu Grunde liegen, in effizienten Programmcode. Dieser Schritt umfasste darüberhinaus auch erforderliche Validierungsarbeiten, d.h. die Notwendigkeit, einzelne Routinen auf ihre Richtigkeit hin zu testen. Um den Simulator im Ganzen zu testen boten sich einige Daten der GP-B-Mission an.

Insgesamt wurde der HPS mit folgenden Modellen ausgestattet:

Dynamische Modelle

Diese Modelle bilden den Kern des Simulators, der auch das Lösen der Bewegungsgleichungen durch numerische Integration durchführt. Der Kern wurde mit folgenden Eigenschaften ausgestattet:

- Simulation der Satelliten- und Testmassen-Dynamik in sechs Freiheitsgrade durch numerische Integration der Bewegungsgleichungen,
- Unterstützung von bis zu vier Schwingungssensoren mit je zwei Testmassen,
- Berücksichtigung linearer und nichtlinearer Kräfte und Drehmomente zwischen Satellit und Testmassen sowie zwischen den Testmassen,
- Wahl eines Modells für das Erdgravitationsfeld: EGM96 und EIGEN-GL04C bis zum Grad und der Ordnung 360, GGM02C bis zum Grad und der Ordnung 200,
- Berücksichtigung des Gravitationsgradienten,
- Integrationsmethode: Runge-Kutta fünfter Ordnung, Bulirsch-Stoer, Euler-Cauchy (kleine Schrittweiten),
- 128-Bit Fließkomma-Arithmetik (QUAD-Präzision) auf ALPHA-Prozessoren.

Auch wurden mehrere Fehlerquellen berücksichtigt

- schlechte Lageausrichtung und Justierung,
- systematische Messabweichungen,
- Versetzungsfehler.



Umweltmodelle

- Erdgravitation,
- Positionen von Planeten und Erdmond,
- Zustand der Erdatmosphäre,
- Sonneneinstrahlung,
- Albedostrahlung der Erde.


Störungsmodelle

- Atmosphärische Reibung,
- Oberflächenkräfte und Drehmomente infolge von elektromagnetischer Strahlung (Sonne, Erdalbedo).

4.4 AP 4000

Zum Simulieren der Satellitendynamik von Gaia entschied man sich für die Entwicklung einer Gaia-spezifische Software, da für eine Anwendung des HPS auf die Gaia-Mission umfangreiche Modifikationen und Erweiterungen hätten integriert werden müssen (s. a. Abschnitt 'Beschreibung der Arbeitspakete'). Man sah hier, nicht zuletzt auch unter zeitlichen Gesichtspunkten, in einer eigenständigen Ausarbeitung den geringeren Aufwand bei höherem Nutzen für die Integration der Software in Programm-Projektentwicklungen anderer Kollegen innerhalb des Gaia-Projekts. Das fortan entwickelte Simulationsmodell namens GAM ('Gaia Attitude Model' ist ein Software-Paket, dass es sich zum Ziel gesetzt hat, die Dynamik und Kinematik des Satelliten als rotierenden starren Körper hochpräzise zu bestimmen. Hierbei berücksichtigt es einerseits externe physikalische Effekte, wie den Strahlungsdruck der Sonne, (Mikro-)Meteoroid-Ereignisse und stochastische Sprünge in der Lage des Satelliten sowie andererseits (modellierte) interne Hardware-Komponenten, die für die Kontrolle eines Satelliten im Orbit essentiell sind, wie z.B. eine Modellierung des Mikro-Triebwerkssystems. Das GAM vereint diese Komponenten in einer modular aufgebauten Programmierstruktur. Dieser Ansatz erlaubt es zudem, verschiedenartige Rauscherzeuger mit einzubinden. Eine entsprechende Untersuchung wurde vor Beginn der Entwicklung des GAM qualitativ durchgeführt und erlaubt, die bisher identifizierten Rauschquellen zu modellieren und in einem nächsten Schritt in die bestehende Struktur des GAM zu implementieren.

Bei der Bearbeitung der OGA-Aufgabe ('On-Ground Attitude'-Bestimmung) im Rahmen des zweiten Gaia-Beitrag des „First Look“-Projektes wurden unterschiedliche

	Schlussbericht First Look - Fast Initial In-Orbit Identification of Scientific Satellites Fkz: 50 OY 0502	
---	--	--

Algorithmen entwickelt, mit deren Hilfe die Rekonstruktion der Satellitenlage möglichst genau, robust und dennoch schnell gemacht werden kann. Durch eine Analyse dieser Algorithmen wurde derjenige ausgewählt, der die genauesten Schätzungen der rekonstruierten Satellitenlage unter einer akzeptablen Verarbeitungszeit liefert. Dabei musste auch darauf geachtet werden, dass er die Anforderungen an die OGA-Realisierung jederzeit erfüllen kann. Wie schon beim GAM wurde auch bei dieser Entwicklungsarbeit auf strenge Kompatibilität zu Entwicklungen anderer Gaia-Projektmitglieder geachtet, da sich diese Arbeit tief in die gesamte Prozesskette der ersten Datenverarbeitungsschritte, die am Boden erfolgen, einbetten lassen muss.

Für STEP konnte keine umfangreiche Simulation erstellt werden, wie dies ursprünglich vorgesehen war. Der Grund dafür war, dass auf Seiten der amerikanischen Institute der Zeitplan durch Budget-Kürzungen und Umbesetzungen in der Personalstruktur immer weiter verzögert wurde. In der Folge wurde amerikanisches Personal auf anderen Projekten eingesetzt, so dass Arbeiten, die für die Simulation von Wichtigkeit gewesen wären, nicht abgeschlossen werden konnten. Unter anderem fehlte es an einem Modell für ein Drag-Free-Regelungssystem, mit dessen Hilfe man in dem „First Look“-Rahmen hätte arbeiten können.



5 Zusammenfassung


Das „First Look“-Projekt wurde mit dem Ziel initiiert, Verfahren für die schnelle In-Orbit-Identifikation von wissenschaftlichen Satelliten zu entwickeln. Dazu wurden exemplarisch vier Missionen ausgewählt. Jedoch hat sich schon in einem sehr frühen Stadium gezeigt, dass die Entwicklungen des Projekts auch für weitere geplante Missionen von Interesse sind, da hier mit gleichen Problemen zu rechnen ist. Dies zeigt sich besonders daran, dass sich der HPS auch bei anderen Projekten außerhalb von „First Look“ als unschätzbare Werkzeug etabliert hat. Beispielsweise wird er intensiv für die Simulation der MICROSCOPE-Mission genutzt. Auch das DLR/Bremen ist mittlerweile gleichermaßen an der Verbesserung und Weiterentwicklung des Simulators beteiligt. Hier besteht der Wunsch, den Simulator für den Einsatz in Hardware-in-the-Loop-Verfahren zu erweitern.

Der wichtigste Beitrag zum Gaia-Projekt konnte mit den beiden Arbeiten über den Simulator GAM und den OGA-Algorithmus geliefert werden. Die in diesem Projekt begonnenen Arbeiten haben sich bereits in der Frühphase der Planungen zu dieser Satelliten-Mission als sehr wertvoll und hilfreich erwiesen, da vergleichbare Ansätze bei früheren Missionen nur vereinzelt aufzufinden und anwendbar waren, und die zu erwartende Genauigkeit in den Messdaten ohnehin eine Neuausrichtung der Herangehensweise erforderlich machte. „First Look“ zeigte, dass wichtige Aufgaben im Zusammenspiel zwischen wissenschaftlicher und ingenieurstechnischer Perspektive bewältigt werden können, zum Gewinn für beide Seiten.

Mit dem „First Look“-Projekt ist eine Grundlage geschaffen worden, aus deren Erkenntnissen und Erfahrungen zukünftige Missionen profitieren können. Die Empfindlichkeit von Messinstrumente wird sich steigern (müssen), um noch kleinere Effekte noch genauer detektieren und analysieren zu können. Daher wird der Erfolg einer Satellitenmission auch davon abhängen, wie gut die Interaktion zwischen einem Satellit und seiner Nutzlast verstanden ist.

	<p>Schlussbericht First Look - Fast Initial In-Orbit Identification of Scientific Satellites Fkz: 50 OY 0502</p>	
---	--	--

6 Anhang

	<p>Equations of Motion for Satellite and Test Mass Dynamics</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 2 of 44</p>
---	---	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	11/18/05	all	Initial Version by Ivanka Pelivan	√

Contents

1. Introduction	4
2. List of Symbols	5
3. Reference Coordinate Frames	7
3.1. Overview	7
3.2. Earth-Centered Inertial Frame	8
3.3. Satellite Body-Fixed Frame	8
3.4. Mechanical Body-Fixed Frame	9
3.5. Accelerometer Frame	9
3.6. Sensor Frame	10
3.7. Test Mass Body-Fixed Frame	11
4. Satellite and Test Mass Dynamics	12
4.1. Satellite Equations of Motion	12
4.2. Test Mass Equations of Motion	13
5. Forces and Torques	17
5.1. Overview	17
5.2. External Forces and Torques	17
5.3. Coupling Forces and Torques	17
6. Validation of Numerical Simulator	19
6.1. Definition of Test Cases	19
6.2. Test 1: Validation of Translational Motion	19
6.3. Test 2: Validation of Rotational Motion	23
6.4. Test 3: Validation of Coupling	24
7. Verification of New Features	28
7.1. Test of Satellite Torque through Coupling Force	28
7.2. Constant Variation of Satellite Mass	33
7.3. Constant Variation of Satellite Center of Mass	37
A. Euler Symmetric Parameters - Quaternions	42
A.1. Definition of Euler Symmetric Parameters	42
A.2. Quaternion Algebra	42
B. Time Derivative of a Vector in a Rotating Frame	44

1. Introduction

This document reviews the satellite and test mass dynamics with updated reference frame definitions for the generic drag-free simulator. The difference to the previous specifications is that the movement of the satellite center of mass due to fuel consumption is taken into account. The satellite body-fixed frame defined with its origin in the satellite's center of mass is therefore moving with the center of mass. To have a reference frame in a point on the satellite that can be located unambiguously a mechanical reference frame is introduced.

In section 3 all reference coordinate frames are described. Thereafter the satellite and test mass dynamics are derived including small changes due to the new reference frame definition. Section 5 comprises implications on force and torque models included in the equations of motion. In section 6 the modified simulator is validated for known test cases. Finally in section 7 the effects of a torque on the satellite through coupling forces and variation of satellite mass and satellite center of mass are investigated.

2. List of Symbols

The table below lists the symbols which are not explicitly explained within the text of this document in their order of appearance.

Symbol	Description
m_{sat}	Mass of satellite.
$\ddot{\underline{r}}_{i,b}^i$	Acceleration of the satellite relative to the inertial frame expressed in the inertial frame.
$\underline{g}_{i,b}^i$	Gravitational acceleration as a function of the satellite's position.
$\underline{F}_{control}^i$	Control force.
\underline{F}_{dist}^i	Sum of all disturbance forces acting on the satellite.
$\underline{F}_{coupl,sat}^i$	Force on the satellite due to the coupling between satellite and test mass(es).
$\underline{\omega}_{i,b}^b$	Angular velocity of the satellite w.r.t. inertial frame expressed in body-fixed coordinate frame.
\underline{I}_b	Moments of inertia matrix of the satellite.
$\underline{T}_{control}^b$	Control torques applied for attitude control expressed in the body-fixed frame.
\underline{T}_{dist}^b	Disturbance torques acting on the satellite expressed in the body-fixed frame.
$\underline{T}_{coupl,sat}^b$	Torques generated from satellite-test mass coupling expressed in the body-fixed frame.
\mathbf{q}_i^b	Attitude quaternion describing the orientation of the satellite body-fixed frame w.r.t. the inertial frame.
m_{tm}	Mass of test mass.
$\ddot{\underline{r}}_{i,tm}^i$	Acceleration of the test mass relative to the inertial frame expressed in the inertial frame.
$\underline{g}_{i,tm}^i$	Gravitational acceleration as a function of the test mass position.
$\underline{F}_{coupl,tm}^i$	Force acting on the test mass due to satellite-test mass coupling.
$\underline{F}_{dist,tm}^i$	Disturbance force directly acting on the test mass (no satellite-test mass interaction).
$\underline{\omega}_{i,tm}^{tm}$	Angular velocity of the test mass relative to the inertial frame expressed in the test mass body-fixed frame.
\underline{I}^{tm}	Moments of inertia matrix of the test mass.
$\underline{T}_{gg,tm}^{tm}$	Gravity gradient torque expressed in the test mass body-fixed frame.
$\underline{T}_{coupl,tm}^{tm}$	Torques generated from satellite-test mass coupling expressed in the test mass body-fixed frame.
$\underline{T}_{dist,tm}^{tm}$	Disturbance torque directly acting on the test mass (no satellite-test mass interaction).



Equations of Motion for Satellite and
Test Mass Dynamics

Doc.No.: FLK-SIM-TN-ZAR-001
Issue: 1.0
Page: 6 of 44

Symbol	Description
\mathbf{q}_i^{tm}	Attitude quaternion describing the orientation of the test mass body-fixed frame w.r.t. the inertial frame.
\underline{T}^{tm}	Sum of all torques acting on the test mass represented in the test mass frame.
$\underline{h}_{i,tm}^{tm}$	Angular momentum w.r.t. the inertial frame represented in the test mass frame.
$\underline{\omega}_{i,tm}^{tm}$	Angular velocity of the test mass w.r.t. the inertial frame.
$\underline{F}_{DC,ij}$	Constant part of the link force between masses i and j
$\underline{K}_{ij}, \underline{K}_{trans}$	Spring stiffness of link
$\underline{D}_{ij}, \underline{D}_{trans}$	Damping of link
$\underline{x}_{offset,ij}$	Offset defining the point where the spring is relaxed
$\underline{T}_{DC,ij}$	Constant part of the link torque
$\underline{K}_{T,ij}, \underline{K}_{rot}$	Rotational stiffness matrix
$\underline{D}_{T,ij}, \underline{D}_{rot}$	Rotational damping matrix
\mathbf{q}_i^j	Attitude quaternion of body j w.r.t. coordinate frame of body i
$\underline{\omega}_i$	Angular velocity of body i
$\underline{T}_{ext,m}$	Torque on satellite from external forces not acting on satellite center of mass.
a	Semimajor axis
e	Orbit eccentricity
C_μ	Gravitational Earth constant
ω_{orbit}	Orbit angular velocity
$\underline{r}_{m,b}^m$	Position of the satellite relative to the mechanical frame expressed in the mechanical frame.
$\underline{r}_{m,a}^m$	Accelerometer offset expressed in mechanical frame.
$\underline{r}_{a,sens}^a$	Sensitive axes offset expressed in accelerometer frame.
$\underline{A}_b^a, \underline{A}_a^{sens}$	Transformation matrices
$\underline{r}_{sens,tm}^{sens}$	Test mass position relative to the sensor frame expressed in the sensor frame.
$\underline{\omega}_{sens,tm}^b$	Angular velocity of the test mass w.r.t. sensor frame expressed in satellite body-fixed coordinate frame.
$\underline{r}_{offset}^{sens}$	Spring offset for linear translational coupling.
ω_n	Natural frequency
δ	Damping constant
\mathbf{q}_{sens}^{tm}	Attitude quaternion describing the orientation of the test mass frame w.r.t. the sensor frame.
$\underline{T}_{tr,sat}$	Torque on satellite from link forces not acting on satellite center of mass.

3. Reference Coordinate Frames

3.1. Overview

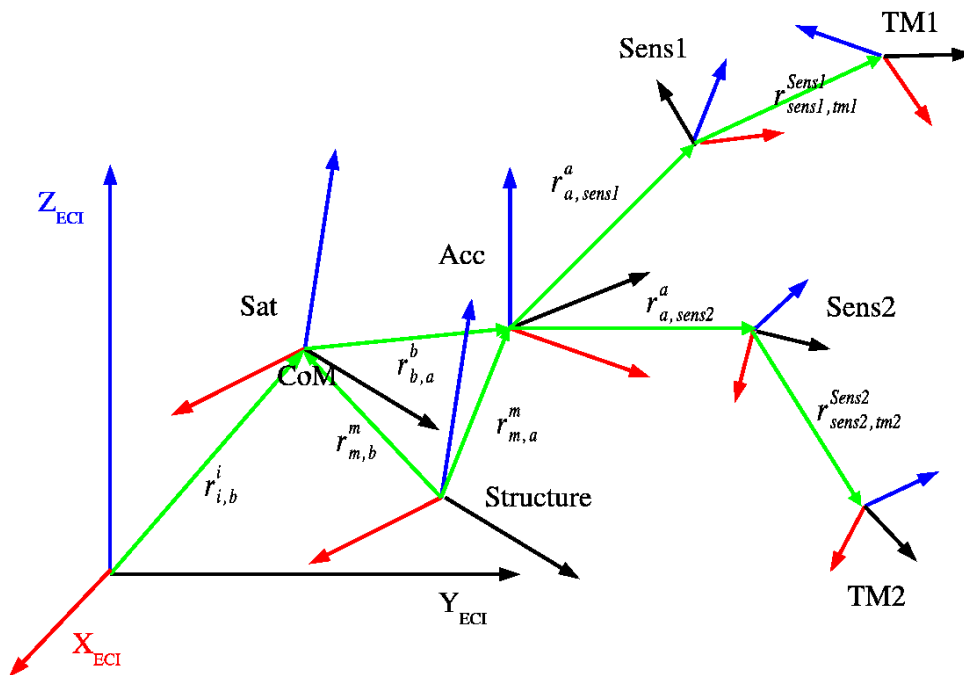


Figure 1: Overview of Coordinate Frames Used

To derive the equations of motion for the satellite and the test masses different frames have to be defined. Figure 1 shows all frames that are used for describing the motion of two test masses with respect to a satellite. The following types of coordinate frames are considered:

Name	Short Name	Index Symbol
Earth-centered inertial frame	ECI, inertial	i
Earth-centered Earth-fixed frame	ECEF, Earth-fixed	$e, ECEF$
Satellite body-fixed frame (in satellite CoM)	body, satellite	b
Mechanical body-fixed frame (on satellite structure)	mechanical, structure	m
Accelerometer frame	accelerometer	a
Sensor frame for test mass *	sensor	$sens^*$
Body-fixed frame for test mass *	test mass	tm^*

3.2. Earth-Centered Inertial Frame

The Earth-centered inertial frame (ECI) is the *Earth Mean Equator and Equinox of J2000 (EME2000)* (see [2] for reference) celestial equator system depicted in figure 2. The origin of the inertial frame is the center of the Earth. The x -axis points towards the vernal equinox for J2000 and lies inside the celestial equatorial plane. The z -axis is parallel to the Earth's angular momentum vector which is perpendicular to the equatorial plane. The right hand orthogonal system is completed by the y -axis which is also located inside the equatorial plane.

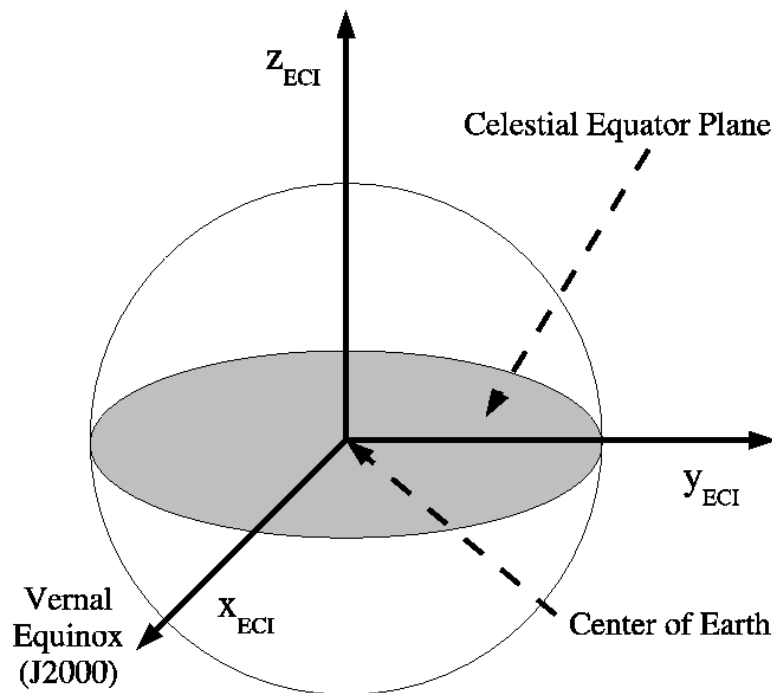


Figure 2: Earth Centered Inertial Frame

3.3. Satellite Body-Fixed Frame

The satellite body-fixed frame has its origin in the center of mass of the satellite. The axes can be defined arbitrarily, e.g. the z -axis can be defined to be the axially symmetrical axis of the satellite. The other axes are defined to be perpendicular to z . In figure 3 the satellite as well as the mechanical body-fixed frame described below are shown.

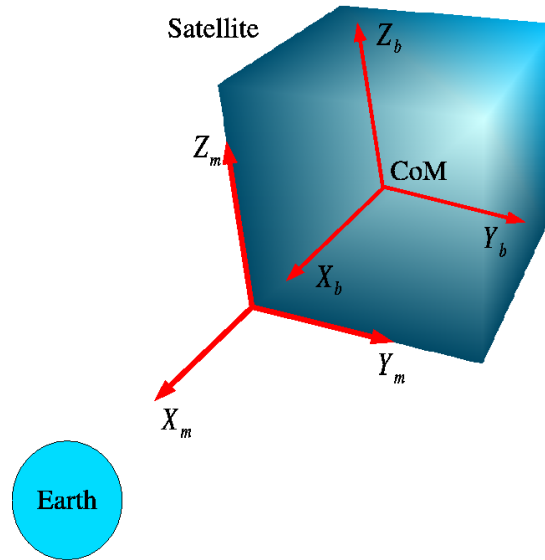


Figure 3: Satellite Body-Fixed Frame (b) and Mechanical Body-Fixed Frame (m)

3.4. Mechanical Body-Fixed Frame

The origin of the mechanical body-fixed frame is a fixed point on the satellite structure. The axes of the mechanical body-fixed coordinate system are parallel to those of the satellite body-fixed frame. Unlike the latter the origin of the mechanical frame is not intended to move with changed conditions. It is fixed to a reference point on the satellite structure where it can be located easily, like a corner of the satellite body. In figure 3 both the satellite and the mechanical body-fixed frame are shown.

3.5. Accelerometer Frame

The accelerometer frame is an arbitrarily defined reference frame for describing technical entities like offset and orientation. Its origin is usually placed in a mechanical reference point of the accelerometer assembly. Figure 4 shows four differential accelerometers as arranged for STEP (Satellite Test of the Equivalence Principle). Here the accelerometers are aligned with the satellite-body fixed frame. The z -axis should be aligned with the z -axis of the satellite body-fixed frame. The real position of the accelerometer frame may differ from this desired orientation due to alignment errors.

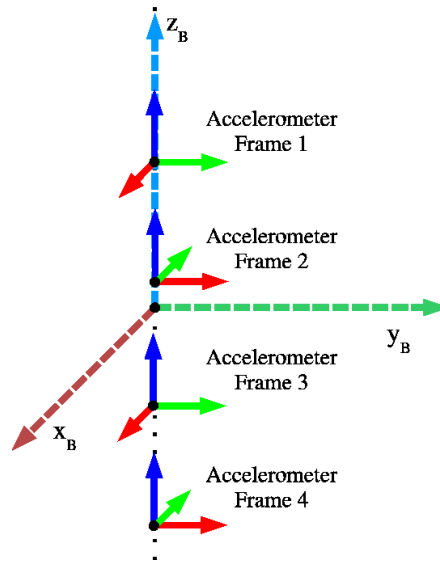


Figure 4: Accelerometer Frames w.r.t. Satellite Body-Fixed Frame

3.6. Sensor Frame

The sensor frame is needed to introduce misalignment and additional offsets inside the differential accelerometer. It defines the origin and direction for measuring the test mass motion with respect to the satellite. The origin is the point where the sensors which are measuring the test mass displacement outputs zero for all three coordinates.

For STEP, the x -axis is defined as the sensitive axis of the SQUID sensors. The y and z -axes are perpendicular to x . They coincide with the axes of the electrostatic measurement system for the displacement in the lateral axes.

Figure 5 shows the sensor frames for both test masses. The sensor frame of the outer test mass has a misalignment about the y -axis with respect to the accelerometer frame.

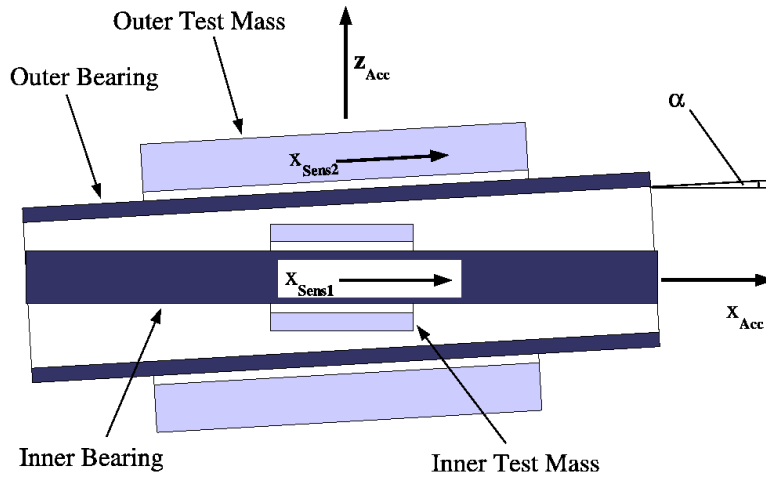


Figure 5: Sensor Frames for Inner and Outer Test Mass

Each sensor frame of a differential accelerometer can have a misalignment w.r.t. the accelerometer frame.

3.7. Test Mass Body-Fixed Frame

The test mass body-fixed frame describes the attitude of the test mass with respect to the sensor frame (see figure 6). The origin is the center of mass of the test mass. For STEP, the x -axis is defined as the axis of rotational symmetry for the non-spherical test masses. The y and z -axes are perpendicular to x .

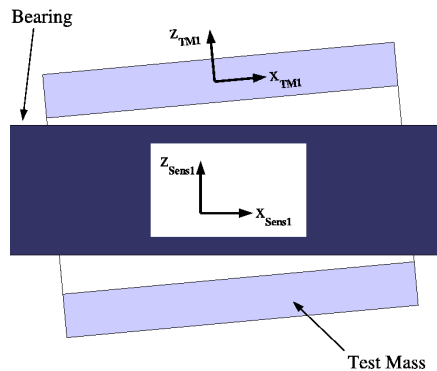



Figure 6: Test Mass Body-Fixed Frame w.r.t. Sensor Frame

	Equations of Motion for Satellite and Test Mass Dynamics	Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 12 of 44
---	---	---

4. Satellite and Test Mass Dynamics

The satellite and test mass equations of motion are re-written here as derived in [3]. Changes due to the displacement vector between body-fixed and mechanical frame are included.

4.1. Satellite Equations of Motion

The equations of motion for the satellite are derived in the inertial frame.

Satellite Translation

The differential equation for the position of the satellite mass center $\underline{r}_{i,b}^i$ is written below (see also [4]):

$$m_{sat}\ddot{\underline{r}}_{i,b}^i = m_{sat}\underline{g}_{i,b}^i(\underline{r}_{i,b}^i) + \underline{F}_{control}^i + \underline{F}_{dist}^i + \underline{F}_{coupl,sat}^i \quad (1)$$

If all terms are divided by the mass of the satellite the equation of motion is expressed in accelerations and specific forces:

$$\ddot{\underline{r}}_{i,b}^i = \underline{g}_{i,b}^i(\underline{r}_{i,b}^i) + \underline{f}_{control}^i + \underline{f}_{dist}^i + \underline{f}_{coupl,sat}^i \quad (2)$$

where \underline{f}_* denotes the corresponding specific forces.

Satellite Attitude

The attitude motion of the satellite is expressed by the angular velocity of the satellite body w.r.t. the inertial frame, $\omega_{i,b}^b$. The attitude itself is described by the Euler symmetric parameters \mathbf{q}_i^b which represent a transformation from inertial frame to satellite body fixed frame (see appendix A).

For the satellite a rigid body is assumed. Then the differential equations for the satellite attitude motion are as follows:

$$\dot{\underline{\omega}}_{i,b}^b = \left(\underline{I}_{=b}^b\right)^{-1} \left[\underline{T}_{control}^b + \underline{T}_{dist}^b + \underline{T}_{coupl,sat}^b - \underline{\omega}_{i,b}^b \times \left(\underline{I}_{=b}^b \underline{\omega}_{i,b}^b\right) \right] \quad (3)$$

$$\dot{\mathbf{q}}_i^b = \frac{1}{2} \hat{\underline{\omega}}_{i,b}^b \odot \mathbf{q}_i^b \quad (4)$$

The term $\hat{\underline{\omega}}_{i,b}^b$ is the quaternion representation of the angular velocity (see appendix A section A.2).

4.2. Test Mass Equations of Motion

The equations of motion for the test mass dynamics are derived in the inertial and in the sensor frame. When implementing the equations of motion for the test mass in the inertial frame numerical difficulties may arise in achieving high resolution for the small displacement between satellite and test mass. This is due to the fact that the displacement will be the result of a difference between two comparably large inertial position vectors. Therefore it is necessary to describe the equations of motion for the test mass in the sensor frame.

Test Mass Translation w.r.t. Inertial Frame

The translational motion for the test mass is described by the differential equation for $\underline{r}_{i,tm}^i$, the position of the test mass center w.r.t. the inertial frame (see [1]):

$$m_{tm} \ddot{\underline{r}}_{i,tm}^i = m_{tm} \underline{g}_{i,tm}^i (\underline{r}_{i,tm}^i) + \underline{F}_{\text{coupl},tm}^i + \underline{F}_{\text{dist},tm}^i \quad (5)$$

Only gravitational forces and coupling forces from the satellite are acting on the test mass. Therefore the test mass follows a pure gravitational orbit which is disturbed by the coupling between test mass and satellite.

If all terms are divided by the mass of the test mass the equation of motion is expressed in accelerations and specific forces:

$$\ddot{\underline{r}}_{i,tm}^i = \underline{g}_{i,tm}^i (\underline{r}_{i,tm}^i) + \underline{f}_{\text{coupl},tm}^i + \underline{f}_{\text{dist},tm}^i \quad (6)$$

where $\underline{f}_{\text{coupl},tm}^i$ denotes the specific force due to coupling.

Test Mass Rotation w.r.t. Inertial Frame

The attitude of the test mass relative to the inertial frame can be described by the Euler symmetric parameters (quaternions) \mathbf{q}_i^{tm} . The differential equation for the rotational motion of the test mass w.r.t. the inertial frame can be described in terms of the angular acceleration of the test mass:

$$\dot{\underline{\omega}}_{i,tm}^{tm} = \left(\underline{I}_{tm}^{tm} \right)^{-1} \left[\underline{T}_{\text{gg},tm}^{tm} + \underline{T}_{\text{coupl},tm}^{tm} + \underline{T}_{\text{dist},tm}^{tm} - \underline{\omega}_{i,tm}^{tm} \times \left(\underline{I}_{tm}^{tm} \underline{\omega}_{i,tm}^{tm} \right) \right] \quad (7)$$

$$\dot{\mathbf{q}}_i^{tm} = \frac{1}{2} \hat{\underline{\omega}}_{i,tm}^{tm} \odot \mathbf{q}_i^{tm} \quad (8)$$

The term $\hat{\underline{\omega}}_{i,tm}^{tm}$ is the quaternion representation of the angular velocity (see appendix A section A.2).

Test Mass Translation w.r.t. Satellite Body-Fixed Frame

The acceleration of the test mass w.r.t. the satellite expressed in the inertial frame can be obtained by subtracting the satellite equation of motion (2) from the test mass equation (6). The equations of motion of a test mass relative to the satellite thus becomes:

$$\begin{aligned}\ddot{\underline{r}}_{b,tm}^i &= \ddot{\underline{r}}_{i,tm}^i - \ddot{\underline{r}}_{i,b}^i \\ &= \underline{g}_{i,tm}^i(\underline{r}_{i,tm}^i) + \underline{f}_{\text{coupl},tm}^i + \underline{f}_{\text{dist},tm}^i - \underline{g}_{i,b}^i(\underline{r}_{i,b}^i) - \underline{f}_{\text{control}}^i - \underline{f}_{\text{dist}}^i - \underline{f}_{\text{coupl},sat}^i\end{aligned}\quad (9)$$

The difference of the gravitational acceleration can be expressed as:

$$\Delta \underline{g}_{b,tm}^i = \underline{g}_{i,tm}^i(\underline{r}_{i,tm}^i) - \underline{g}_{i,b}^i(\underline{r}_{i,b}^i) \quad (10)$$

where $\underline{r}_{i,tm}^i = \underline{r}_{i,b}^i + \underline{r}_{b,m}^i + \underline{r}_{m,a}^i + \underline{r}_{a,sens}^i + \underline{r}_{sens,tm}^i$.

The relative acceleration of the test mass is expressed in the rotating sensor frame. For that purpose the whole equation is multiplied by the transformation matrix describing the rotation from inertial frame to sensor frame. The second derivative of the position however is still formed w.r.t. the inertial frame:

$${}^{(i)}\ddot{\underline{r}}_{b,tm}^{sens} = \Delta \underline{g}_{b,tm}^{sens} + \underline{f}_{\text{coupl},tm}^{sens} + \underline{f}_{\text{dist},tm}^{sens} - \underline{f}_{\text{control}}^{sens} - \underline{f}_{\text{dist}}^{sens} - \underline{f}_{\text{coupl},sat}^{sens} \quad (11)$$

The second derivative in the sensor frame can be found by applying equation (124) of appendix B to the derivative of the test mass position in the inertial frame:

$$\begin{aligned}{}^{(i)}\ddot{\underline{r}}_{b,tm}^{sens} &= {}^{(sens)}\ddot{\underline{r}}_{b,tm}^{sens} + 2 \underline{\omega}_{i,sens}^{sens} \times {}^{(sens)}\dot{\underline{r}}_{b,tm}^{sens} + \\ &{}^{(sens)}\dot{\underline{\omega}}_{i,sens}^{sens} \times \underline{r}_{b,tm}^{sens} + \underline{\omega}_{i,sens}^{sens} \times (\underline{\omega}_{i,sens}^{sens} \times \underline{r}_{b,tm}^{sens})\end{aligned}\quad (12)$$

The angular velocity of the sensor frame w.r.t. the inertial frame can be expressed as the sum of angular velocities of each frame w.r.t. the neighboring frame:


$$\underline{\omega}_{i,sens} = \underline{\omega}_{i,b} + \underline{\omega}_{b,m} + \underline{\omega}_{m,a} + \underline{\omega}_{a,sens} \quad (13)$$

$$\dot{\underline{\omega}}_{i,sens} = \dot{\underline{\omega}}_{i,b} + \dot{\underline{\omega}}_{b,m} + \dot{\underline{\omega}}_{m,a} + \dot{\underline{\omega}}_{a,sens} \quad (14)$$

The same holds for the derivative of the angular velocity, equation (14). If we assume a rigid spacecraft the relative angular velocity between satellite and sensor frame becomes zero, which simplifies to:

$$\underline{\omega}_{i,sens} = \underline{\omega}_{i,b} \quad (15)$$

$$\dot{\underline{\omega}}_{i,sens} = \dot{\underline{\omega}}_{i,b} \quad (16)$$

	Equations of Motion for Satellite and Test Mass Dynamics	Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 15 of 44
---	---	---

Substituting equations (12), (15) and (16) into equation (11) the acceleration of the test mass w.r.t. the satellite expressed in the sensor frame becomes:

$$\ddot{\underline{r}}_{b,tm}^{sens} = \Delta \underline{g}_{b,tm}^{sens} - \underline{f}_{control}^{sens} - \underline{f}_{dist}^{sens} - \underline{f}_{coupl,sat}^{sens} + \underline{f}_{coupl,tm}^{sens} + \underline{f}_{dist,tm}^{sens} - 2 \underline{\omega}_{i,b}^{sens} \times \dot{\underline{r}}_{b,tm}^{sens} - \dot{\underline{\omega}}_{i,b}^{sens} \times \underline{r}_{b,tm}^{sens} - \underline{\omega}_{i,b}^{sens} \times (\underline{\omega}_{i,b}^{sens} \times \underline{r}_{b,tm}^{sens}) \quad (17)$$

For a rigid spacecraft the position and orientation of the sensor frame w.r.t. the satellite body-fixed frame is fixed. That means $\underline{r}_{b,sens} = \underline{r}_{b,m} + \underline{r}_{m,a} + \underline{r}_{a,sens}$ is constant. So the following equations for the first and second derivative of $\underline{r}_{b,sens}$,

$$\dot{\underline{r}}_{b,tm}^{sens} = \dot{\underline{r}}_{b,m}^{sens} + \dot{\underline{r}}_{m,a}^{sens} + \dot{\underline{r}}_{a,sens}^{sens} + \dot{\underline{r}}_{sens,tm}^{sens} \quad (18)$$

$$\ddot{\underline{r}}_{b,tm}^{sens} = \ddot{\underline{r}}_{b,m}^{sens} + \ddot{\underline{r}}_{m,a}^{sens} + \ddot{\underline{r}}_{a,sens}^{sens} + \ddot{\underline{r}}_{sens,tm}^{sens} \quad (19)$$

simplify to

$$\dot{\underline{r}}_{b,tm}^{sens} = \dot{\underline{r}}_{sens,tm}^{sens} \quad (20)$$

$$\ddot{\underline{r}}_{b,tm}^{sens} = \ddot{\underline{r}}_{sens,tm}^{sens} \quad (21)$$

The equation of motion for the translation of the test mass (17) thus becomes:

$$\ddot{\underline{r}}_{sens,tm}^{sens} = \Delta \underline{g}_{b,tm}^{sens} - \underline{f}_{control}^{sens} - \underline{f}_{dist}^{sens} - \underline{f}_{coupl,sat}^{sens} + \underline{f}_{coupl,tm}^{sens} + \underline{f}_{dist,tm}^{sens} - 2 \underline{\omega}_{i,b}^{sens} \times \dot{\underline{r}}_{sens,tm}^{sens} - \dot{\underline{\omega}}_{i,b}^{sens} \times (\underline{r}_{b,sens}^{sens} + \underline{r}_{sens,tm}^{sens}) - \underline{\omega}_{i,b}^{sens} \times (\underline{\omega}_{i,b}^{sens} \times (\underline{r}_{b,sens}^{sens} + \underline{r}_{sens,tm}^{sens})) \quad (22)$$

Test Mass Rotation w.r.t. Satellite Body-Fixed Frame

The equation of conservation of angular momentum states that

$$\underline{T}^i = \left(\frac{d\underline{h}}{dt} \right)^i \quad (23)$$

which in the present context means that the sum of all torques \underline{T}^i acting on the test mass is equal to the time derivative of the test mass' angular momentum \underline{h} in the inertial frame. Considering the total torque in the test mass frame (see appendix B equation 123) this can be written as:

$$\underline{T}^{tm} = {}^{(tm)} \left(\frac{d\underline{h}_{i,tm}^{tm}}{dt} \right) + \underline{\omega}_{i,tm}^{tm} \times \underline{h}_{i,tm}^{tm} \quad (24)$$

If we assume that the test mass is a rigid body the following holds for the angular momentum in the test mass frame:

$$\underline{h}_{i,tm}^{tm} = \underline{I}_{tm}^{tm} \cdot \underline{\omega}_{i,tm}^{tm} \quad (25)$$

$${}^{(tm)} \left(\frac{d \underline{h}_{i,tm}^{tm}}{dt} \right) = \underline{I}_{tm}^{tm} \cdot \underline{\dot{\omega}}_{i,tm}^{tm} \quad (26)$$

Substituting this into equation (24) the torque represented in the test mass frame becomes:

$$\underline{T}^{tm} = \underline{I}_{tm}^{tm} \cdot \underline{\dot{\omega}}_{i,tm}^{tm} + \underline{\omega}_{i,tm}^{tm} \times \left(\underline{I}_{tm}^{tm} \cdot \underline{\omega}_{i,tm}^{tm} \right) \quad (27)$$

This can be solved for the derivative of the angular acceleration:

$$\underline{\dot{\omega}}_{i,tm}^{tm} = \left(\underline{I}_{tm}^{tm} \right)^{-1} \left[\underline{T}^{tm} - \underline{\omega}_{i,tm}^{tm} \times \left(\underline{I}_{tm}^{tm} \cdot \underline{\omega}_{i,tm}^{tm} \right) \right] \quad (28)$$

The angular velocity of the test mass relative to the inertial frame can be expressed as the sum of angular velocities of each frame w.r.t. the neighboring frame:

$$\underline{\omega}_{i,tm}^{tm} = \underline{\omega}_{i,b}^{tm} + \underline{\omega}_{b,m}^{tm} + \underline{\omega}_{m,a}^{tm} + \underline{\omega}_{a,sens}^{tm} + \underline{\omega}_{sens,tm}^{tm} \quad (29)$$

Taking into account that the satellite is modeled as a rigid body (see equations (15) and (16)) the angular velocity and acceleration of the test mass w.r.t. the sensor frame simplify to:

$$\begin{aligned} \underline{\omega}_{sens,tm}^{tm} &= \underline{\omega}_{i,tm}^{tm} - \underline{\omega}_{i,b}^{tm} \\ \underline{\dot{\omega}}_{sens,tm}^{tm} &= \underline{\dot{\omega}}_{i,tm}^{tm} - \underline{\dot{\omega}}_{i,b}^{tm} \end{aligned} \quad (30)$$

The substitution of equation (30) into equation (28) yields for the angular acceleration of the test mass w.r.t. the sensor frame:

$$\underline{\dot{\omega}}_{sens,tm}^{tm} = \left(\underline{I}_{tm}^{tm} \right)^{-1} \left[\underline{T}^{tm} - \left(\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm} \right) \times \left(\underline{I}_{tm}^{tm} \left(\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm} \right) \right) \right] - \underline{\dot{\omega}}_{i,b}^{tm} \quad (31)$$

If the test mass inside the satellite is shielded from all external non-gravitational forces and torques, the torque in equation (31) is equal to the sum of the rotational coupling torques, the torques due to the gravity gradient acting on the test masses and internal disturbance torques:

$$\underline{T}^{tm} = \underline{T}_{coup,tm}^{tm} + \underline{T}_{gg,tm}^{tm} + \underline{T}_{dist,tm}^{tm} \quad (32)$$

The attitude of the test mass w.r.t. the sensor frame can be expressed by quaternions:

$$\dot{\mathbf{q}}_{sens}^{tm} = \frac{1}{2} \hat{\underline{\omega}}_{sens,tm}^{tm} \odot \mathbf{q}_{sens}^{tm} \quad (33)$$

5. Forces and Torques

5.1. Overview

The forces and torques appearing in the equations of motion for the satellite and test mass are due to various sources. For the satellite there are forces and torques due to the control actuation for satellite attitude and translation control, the atmosphere, electromagnetic radiation, external magnetic field impulses from space debris and meteoroid hits. These forces and torques can be summarized as control and disturbance forces and torques. Additionally forces and torques due to gravitation between satellite and test mass and celestial bodies as well as coupling forces and torques between the satellite and test mass occur. For the test mass, since it is shielded from external disturbances by the satellite, only gravitational effects and the coupling between satellite and test mass have to be taken into account. A detailed description of models for forces and torques can be found in [3].

5.2. External Forces and Torques

For the generic drag-free simulator there is the option to define external forces and torques:

$$\underline{F}_{external} = [\underline{F}^i, \underline{F}^o, \underline{F}^m] \quad (34)$$

$$\underline{T}_{external} = [\underline{T}^i, \underline{T}^o, \underline{T}^m] \quad (35)$$

Each input has three vector components, the first containing all elements acting in the inertial frame, the second vector is comprised of orbital elements and the third is defined w.r.t. the mechanical body-fixed frame. While the first two vectors are acting on the satellite center of mass, the third component is acting on the point where the mechanical reference frame has its origin. Therefore an additional torque w.r.t. the satellite center of mass arises:


$$\underline{T}_{ext,m}^m = -\underline{r}_{CoM}^m \times \underline{F}^m \quad (36)$$

with $\underline{r}_{CoM}^m = \underline{r}_{m,b}^m$ (see figure 1).

This torque has to be added to the total torque acting on the satellite center of mass.

5.3. Coupling Forces and Torques

Satellite - test mass coupling occurs because the observation of the test mass, i.e. the determination of the test mass position, cannot be accomplished force free. Coupling forces and torques between the satellite and test mass or between two test masses are modelled as a linear system link with a constant part, spring properties and damping. Details about this model can be found in Appendix E of [3]. In the following the link

	<p>Equations of Motion for Satellite and Test Mass Dynamics</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 18 of 44</p>
---	---	--

force and torque between two bodies 1 and 2 (satellite - test mass or test mass 1 - test mass 2) are outlined:

$$\underline{F}_{ij}^{(j)} = \underline{F}_{DC,ij} - \underline{K}_{ij}(\underline{x}_j - \underline{x}_i - \underline{x}_{offset,ij}) - \underline{D}_{ij}(\dot{\underline{x}}_j - \dot{\underline{x}}_i) \quad (37)$$

$$\underline{T}_{ij}^{(j)} = \underline{T}_{DC,ij} - 2 \cdot \underline{K}_{T,ij} \mathbf{q}_i^j - \underline{D}_{T,ij}(\underline{\omega}_j - \underline{\omega}_i) \quad (38)$$

For the test mass it is assumed that the link forces are acting on its center of mass. If the coupling force is not acting on the center of mass of the satellite it can produce a torque. This torque from translational coupling can be expressed as:

$$\underline{T}_{tr,0i}^{(0)} = \underline{r}_{0i}^{(0)} \times \underline{F}_{0i}^{(0)} \quad (39)$$

Here 0 is used for the satellite and $i = 1, 2$ is assigned to the test masses of one accelerometer. The vector $\underline{r}_{0i}^{(0)}$ is the lever arm from the center of mass of test mass i where the force $\underline{F}_{0i}^{(0)}$ is acting, to the center of mass of the satellite:

$$\underline{r}_{0i}^{(0)} = \underline{r}_{b,tm(i)} = \underline{r}_{m,a} - \underline{r}_{m,b} + \underline{r}_{a,sens} + \underline{r}_{sens,tm(i)} \quad (40)$$

6. Validation of Numerical Simulator

6.1. Definition of Test Cases

A number of test cases has been carried out to verify that the definition of the mechanical reference frame has been implemented correctly. Three test cases are defined according to [3] to validate

- translational motion of the test mass
- rotational motion of the test mass
- coupling.

6.2. Test 1: Validation of Translational Motion

For this test case the following conditions are applied:

- no coupling between satellite and test mass
- no disturbance torques and forces
- fixed satellite center of mass.

The initial conditions for the satellite are set to

$$\underline{r}_{i,b}^i = \begin{bmatrix} x_{i,b}^i \\ y_{i,b}^i \\ z_{i,b}^i \end{bmatrix} = \begin{bmatrix} a \cdot (1.0 - e) \\ 0 \\ 0 \end{bmatrix} m \quad (41)$$

$$\underline{\dot{r}}_{i,b}^i = \begin{bmatrix} 0 \\ \sqrt{C_\mu \cdot \left(\frac{2}{x_{i,b}^i} - \frac{1}{a} \right)} \\ 0 \end{bmatrix} \frac{m}{s} \quad (42)$$

where $a = (C_\mu / \omega_{orbit}^2)^{1/3}$, e is set to zero in this case, $C_\mu = 3986004.415 \cdot 10^8$ and $\omega_{orbit} = 1.108507726 \cdot 10^{-3}$.

The position vectors are defined as:

$$\underline{r}_{m,b}^m = \begin{bmatrix} 0.3 m \\ 0.2 m \\ 0.1 m \end{bmatrix} \quad (43)$$

$$\underline{r}_{m,a}^m = \begin{bmatrix} 0.6 m \\ 0.4 m \\ 0.2 m \end{bmatrix} \quad (44)$$

which is equal to

$$\underline{r}_{b,a}^b = -\underline{r}_{m,b}^m + \underline{r}_{m,a}^m = \begin{bmatrix} 0.3 \text{ m} \\ 0.2 \text{ m} \\ 0.1 \text{ m} \end{bmatrix} \quad (45)$$

The origin of the sensor coordinate system is at the center of mass of the satellite:

$$\underline{r}_{a,sens}^a = \begin{bmatrix} 0.1 \text{ m} \\ -0.3 \text{ m} \\ -0.2 \text{ m} \end{bmatrix} \quad (46)$$

$$(47)$$

The transformations are defined as:

$$\underline{\underline{A}}_b^a = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (48)$$

$$\underline{\underline{A}}_a^{sens} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \quad (49)$$

Test Case 1A

For an initial position of zero,

$$\underline{r}_{sens,tm}^{sens} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (50)$$

the test mass must follow the same purely gravitational orbit as the satellite since the starting position of satellite and test mass are the same.

Figure 7 shows that the position and the velocity remain at zero for zero displacement of the test mass.

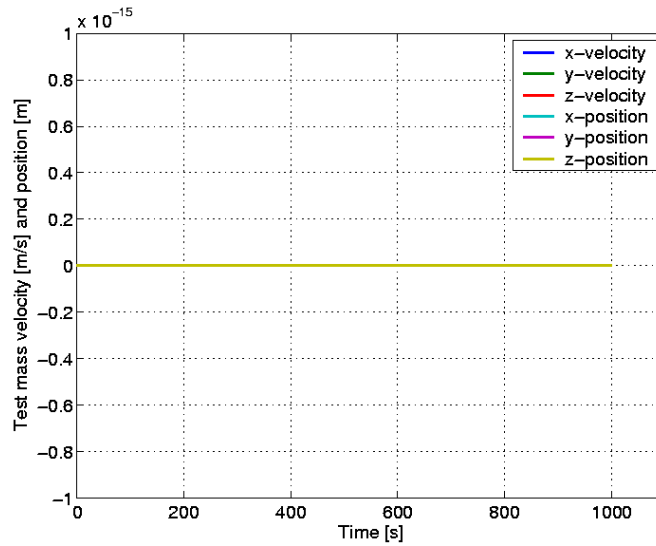


Figure 7: Position and velocity of the test mass w.r.t. the sensor frame; No initial displacement of the test mass.

Deviations from zero can occur due to numerical errors. For position vectors of

$$\underline{r}_{m,b}^m = \begin{bmatrix} 0.2 \text{ m} \\ 0.4 \text{ m} \\ 0.2 \text{ m} \end{bmatrix} \quad (51)$$

$$\underline{r}_{m,a}^m = \begin{bmatrix} 0.5 \text{ m} \\ 0.6 \text{ m} \\ 0.3 \text{ m} \end{bmatrix} \quad (52)$$

the numerical error after 1000s is of order 10^{-17} or a relative error of 10^{-16} . This development is shown in figure 8. It has to be kept in mind that these errors accumulate for longer simulation times. For the current simulator the same uncertainties remain when the simulation is carried out on an alpha machine since the vectors are defined using double precision. The error is inherent when changing them into quad precision after values have been assigned.

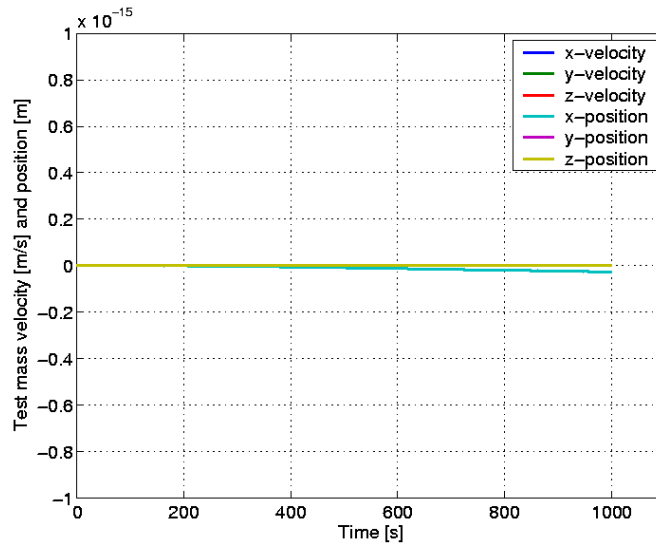


Figure 8: Position and velocity of the test mass w.r.t. the sensor frame; No initial displacement of the test mass.

For the following test the first set of position vectors is used, i.e.:

$$\underline{r}_{m,b}^m = \begin{bmatrix} 0.3 \text{ m} \\ 0.2 \text{ m} \\ 0.1 \text{ m} \end{bmatrix} \quad (53)$$

$$\underline{r}_{m,a}^m = \begin{bmatrix} 0.6 \text{ m} \\ 0.4 \text{ m} \\ 0.2 \text{ m} \end{bmatrix} \quad (54)$$

Test Case 1B

For an initial position of

$$\underline{r}_{sens,tm}^{sens} = \begin{bmatrix} 100 \text{ m} \\ 0 \text{ m} \\ 0 \text{ m} \end{bmatrix} \quad (55)$$

it is expected that, due to the fact that satellite and test mass are not coupled in their motion, they will follow their own gravitational orbits.

In figure 9 the inertial position of the test mass and the position w.r.t. the satellite are shown. The curves for the x-, y- and z-position differ from those shown in [3] because

different initial conditions for the satellite were used. When repeating the test case with the original simulator used in [3] and initial conditions (41) and (42) the results obtained are identical to those shown in figure 9.

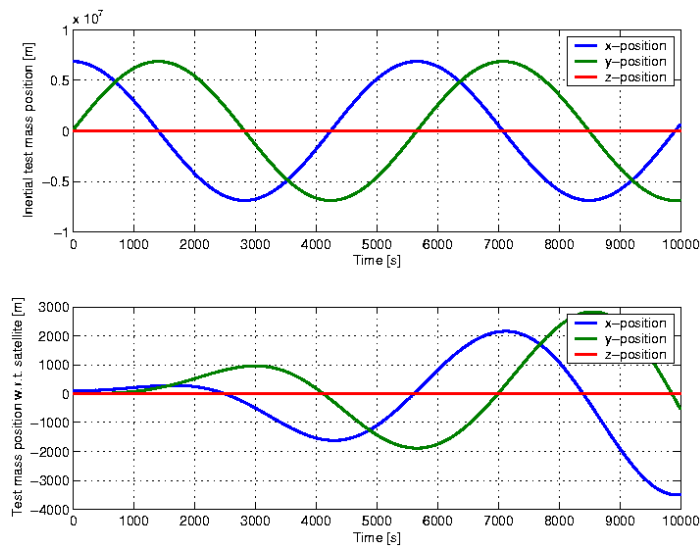


Figure 9: Position and velocity of the test mass w.r.t. inertial and satellite body-fixed frame; initial displacement of the test mass = 100m.

6.3. Test 2: Validation of Rotational Motion

For the validation of rotational motion the same conditions as for the first test case are applied. The transformations and position vectors also are defined as for the first test case.

The moments of inertia of the satellite are defined as:

$$\underline{\underline{I}}_b = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (56)$$

For this test case the satellite has no initial angular momentum and initial attitude rate.

In this test external torques are applied around the satellite body-fixed axes separately. Since the coupling between satellite and test mass is deactivated and the test mass has no initial rate it is expected that the test mass remains inertially fixed in its attitude and the satellite rotates around the test mass due to the external torque. If this is the case then the attitude of the test mass w.r.t. the sensor frame will be the inverse of the attitude of the satellite w.r.t. the inertial frame:

$$\begin{bmatrix} q_{i1}^b \\ q_{i2}^b \\ q_{i3}^b \\ q_{i4}^b \end{bmatrix} = \begin{bmatrix} -q_{sens1}^{tm} \\ -q_{sens2}^{tm} \\ -q_{sens3}^{tm} \\ q_{sens4}^{tm} \end{bmatrix} \quad (57)$$

The same holds for the angular velocity:

$$\underline{\omega}_{i,b}^b = -\underline{\omega}_{sens,tm}^b \quad (58)$$

In figure 10 results are shown for a torque of 10^{-14} Nm applied around the satellite body-fixed x-axis for 1000s. The sub-figure to the upper left shows the x-component of the attitude rates of satellite and test mass. The next sub-figure to the right shows the first attitude quaternion. The differences in angular velocity and attitude quaternion are shown in the last two sub-figures. The very small values are of order of the numerical precision limit and proof that in all cases (also for the other components not shown here) equations (58) and (57) are fulfilled.

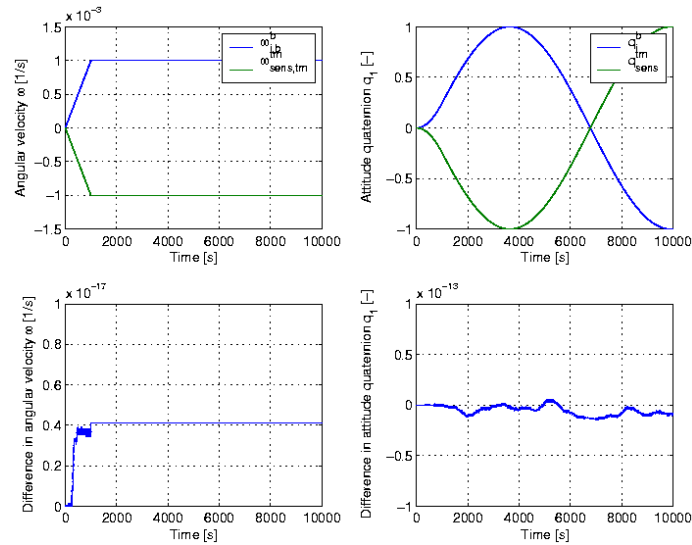



Figure 10: Comparison of attitude rates and quaternion.

6.4. Test 3: Validation of Coupling

For verification of coupling the following conditions are applied:

- Satellite position and attitude are fixed.
- The gravitational acceleration of the test mass is deactivated.

	Equations of Motion for Satellite and Test Mass Dynamics	Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 25 of 44
---	---	---

- Only one test mass is present which is coupled to the satellite via one single link.
- The link is modelled as a spring with stiffness, damping and constant part.

The mass of the test mass is set to 1kg.

Test Case 3A: Translational Coupling

For this test case the following parameters are chosen:

$$\underline{\underline{K}}_{trans} = \begin{bmatrix} \omega_n^2 & 0 & 0 \\ 0 & 2\omega_n^2 & 0 \\ 0 & 0 & 9\omega_n^2 \end{bmatrix} \frac{N}{m} \quad (59)$$

$$\underline{\underline{D}}_{trans} = \begin{bmatrix} \delta & 0 & 0 \\ 0 & 5.0\delta & 0 \\ 0 & 0 & 7.0\delta \end{bmatrix} \frac{N \cdot s}{m} \quad (60)$$

$$\underline{r}_{offset}^{sens} = \begin{bmatrix} 1 \cdot 10^{-9} \\ 2 \cdot 10^{-9} \\ 3 \cdot 10^{-9} \end{bmatrix} m \quad (61)$$

$$\underline{F}_{DC,trans} = \begin{bmatrix} 4 \cdot 10^{-14} \\ 5 \cdot 10^{-14} \\ 3 \cdot 10^{-14} \end{bmatrix} N \quad (62)$$

Since the off-diagonal elements are zero the oscillations in all directions are independent and no cross-coupling occurs. The different stiffness and damping constants make it possible to distinguish between the three directions. The natural frequency ω_n is chosen as

$$\omega_n = 2\pi \cdot 0.001 \cdot s^{-1} \quad (63)$$

such that the oscillation with the natural frequency will have a period of 1000s. The damping constant δ is set to

$$\delta = -\ln 0.5 \cdot 0.001. \quad (64)$$

which will result in an attenuation of the amplitude to half of its initial value after 2000s.

In figure 11 the simulation results for an oscillation of the test mass along the x-axis are shown. Results are similar for the y- and z-direction. The expected period of 1000s is clearly visible. In the first sub-figure the damping is deactivated and $\underline{r}_{offset}^{sens} = 0$. In the next three sub-figures damping is enabled. The damping causes the amplitude to half after 2000s. The second sub-figure in the upper right shows the elongation of the test mass for zero offset. The lower left figure shows the damped oscillation with an offset for

the spring origin. The last figure shows the damped oscillation with offset and DC force. The DC force corresponds to an elongation of 10^{-9} m s.t. the oscillation will approach the mean of twice the spring offset value.

The results are identical to those obtained with the original simulator without mechanical body-fixed frame (see [3]).

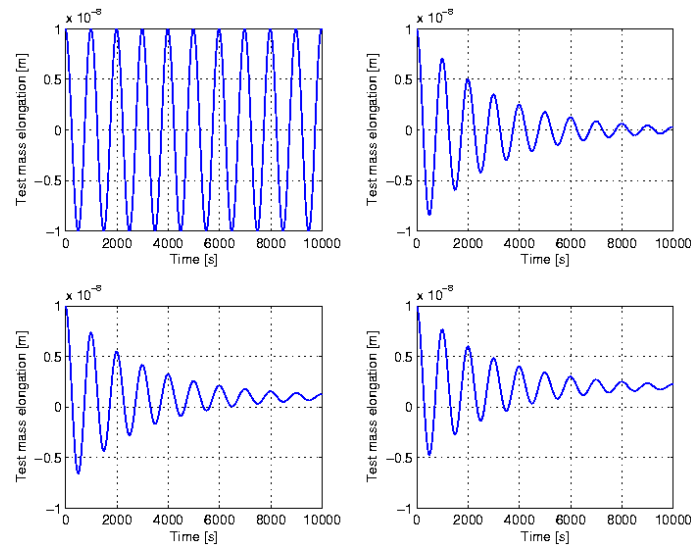


Figure 11: Oscillation of the test mass. From upper left to lower right: no damping, damping, damped with offset, damped with offset and DC force.

Test Case 3B: Rotational Coupling

The following parameters are chosen for the test of rotational coupling:

$$\underline{K}_{rot} = \begin{bmatrix} \omega_n^2 & 0 & 0 \\ 0 & 2\omega_n^2 & 0 \\ 0 & 0 & 9\omega_n^2 \end{bmatrix} N \cdot m \quad (65)$$

$$\underline{D}_{rot} = \begin{bmatrix} \delta & 0 & 0 \\ 0 & 5.0\delta & 0 \\ 0 & 0 & 7.0\delta \end{bmatrix} N \cdot m \cdot s \quad (66)$$

$$\underline{T}_{DC,rot} = \begin{bmatrix} 1 \cdot 10^{-10} \\ 5 \cdot 10^{-10} \\ 3 \cdot 10^{-10} \end{bmatrix} N \cdot m \quad (67)$$

The natural frequency ω_n and damping constant δ are defined as in (63) and (64).

Figure 12 shows simulation results for the x-component of the test mass angular velocity and first quaternion. In the upper two figures the angular velocity and quaternion are plotted for rotational oscillation without damping. The lower left figure shows the quaternion oscillation with damping. In the lower right figure the damped oscillation with DC torque is shown. As expected the period is 1000s and with damping the initial amplitude decreases to half of its value after 2000s. The DC torque shifts the mean of the oscillation about $1.5 \cdot 10^{-6}m$.

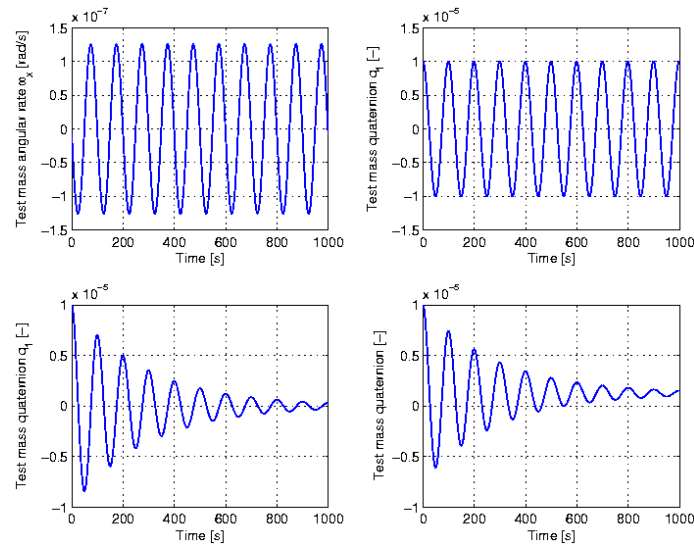


Figure 12: Oscillation of the test mass. From upper left to lower right: undamped angular velocity, undamped quaternion, damped, damped with DC torque.

7. Verification of New Features

In this section the effects of attenuation of satellite mass and a resulting variation of displacement between satellite center of mass and origin of mechanical reference frame are investigated. In the first test the existence of a fixed displacement resulting in a torque through coupling forces is examined.

The conditions applied for the following investigation are:

- no disturbance forces and torques
- gravitation is turned off
- only one test mass present
- constant variation of satellite mass
- constant variation of displacement between satellite center of mass and origin of mechanical reference frame

7.1. Test of Satellite Torque through Coupling Force

For the test mass it is assumed that the link forces are acting on its center of mass. If the coupling force is not acting on the center of mass of the satellite it can produce a torque. This torque from translational coupling from one test mass can be expressed as:

$$\underline{T}_{tr,sat} = \underline{r}_{b,tm} \times \underline{F}_{coupl,sat} \quad (68)$$

The vector $\underline{r}_{b,tm}$ is the lever arm from the center of mass of the test mass where the force $\underline{F}_{coupl,sat}$ is acting, to the center of mass of the satellite (see equation (40)).

For this test the satellite degrees of freedom are restricted to rotational motion only and for the test mass only translational movement is allowed. A system corresponding to this test case is shown in figure 13.

To emphasize the effects of a torque through coupling forces, mass reduction and movement of satellite center of gravity the gravitation is turned off. The position vectors are defined as:

$$\underline{r}_{m,b}^m = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (69)$$

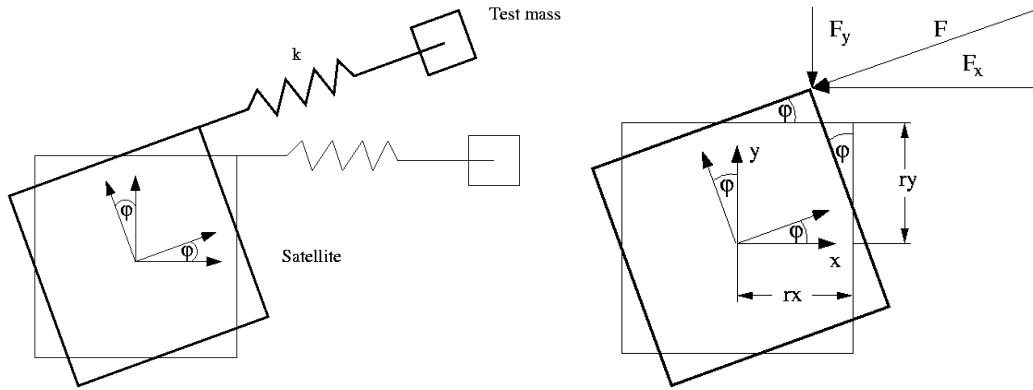


Figure 13: Model for test of satellite torque through coupling force.

$$\underline{r}_{m,a}^m = \underline{r}_{m,b}^m + \begin{bmatrix} 4.0 \\ 1.0 \\ 0.0 \end{bmatrix} m \quad (70)$$

The initial conditions for the satellite are set to zero. The initial test mass position is set to


$$\underline{r}_{sens,tm}^{sens} = \begin{bmatrix} 3.0 \\ 2.0 \\ 0.0 \end{bmatrix} \cdot 10^{-3} m \quad (71)$$

A link force with spring parameters according to the previous test of translational coupling is chosen:

$$\underline{K}_{trans} = \begin{bmatrix} \omega_n^2 & 0 & 0 \\ 0 & \omega_n^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{N}{m} \quad (72)$$

where ω_n is set to $2\pi \cdot 0.001 \cdot s^{-1}$. No damping or DC forces are present. The link force applied thus has two components $F_x = -k \cdot \Delta x$ and $F_y = -k \cdot \Delta y$ where $k = \omega_n^2$ (the mass of the test mass is set to 1kg) and Δx and Δy are the differences between satellite and test mass position in x- and y-direction. Due to the lever arm $\underline{r}_{b,tm}$ a torque $T_{tr,z}$ around the z-axis arises. Without disturbance and control torques the differential equation for the satellite attitude motion reduces to

$$\dot{\underline{\omega}}_{i,b}^b = \left(\underline{I}_b^b \right)^{-1} \left[\underline{T}_{tr}^b - \underline{\omega}_{i,b}^b \times \left(\underline{I}_b^b \underline{\omega}_{i,b}^b \right) \right] \quad (73)$$

	<p>Equations of Motion for Satellite and Test Mass Dynamics</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 30 of 44</p>
---	---	--

with one non-zero component, $\underline{\dot{\omega}}_{i,b,z}^b = (I_{zz}^b)^{-1} \underline{T}_{tr,z}^b$ (the satellite moments of inertia are defined in (56)).

In the following the analytical solution for the system shown in figure 13 is derived. The equations of motion for the test mass movement in x- and y-direction and satellite rotation φ are obtained utilizing the Lagrange method:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}} \right) - \frac{\partial E_k}{\partial q} + \frac{\partial E_p}{\partial q} = 0 \quad (q = x, y, \varphi) \quad (74)$$

With

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 1 & -\varphi \\ \varphi & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix} \right), \quad \begin{bmatrix} r_x \\ r_y \\ 0 \end{bmatrix} = \underline{r}_{m,a} - \underline{r}_{m,b} \quad (75)$$

the kinetic energy E_k and the potential energy E_p are found as:

$$E_k = \frac{1}{2} m \dot{x}^2 + \frac{1}{2} m \dot{y}^2 + \frac{1}{2} I_{zz} \dot{\varphi}^2 \quad (76)$$

$$E_p = \frac{1}{2} k (x + r_y \varphi - r_x)^2 + \frac{1}{2} k (y - r_x \varphi - r_y)^2 \quad (77)$$

In equation (75) simplifications are made which are appropriate for small angles: $\sin(\varphi) \approx \varphi$, $\cos(\varphi) \approx 1$.

Applying eqn. (74) the equations of motion are obtained:

$$\ddot{x} + \omega_n^2 x + \omega_n^2 r_y \varphi = \omega_n^2 r_x \quad (78)$$

$$\ddot{y} + \omega_n^2 y - \omega_n^2 r_x \varphi = \omega_n^2 r_y \quad (79)$$

$$\ddot{\varphi} + \omega_I^2 (r_x^2 + r_y^2) \varphi + \omega_I^2 r_y x - \omega_I^2 r_x y = 0 \quad (80)$$

Here $\omega_n^2 = k/m$, $\omega_I^2 = k/I_{zz}$.


The path to a general solution involves finding a solution to the homogeneous system of equations (i.e. with zero right-hand side), and afterwards finding a particular solution to the non-homogeneous system. Trying a solution of the form $q = \hat{q} e^{\lambda t}$ for the homogeneous system of equations yields a linear system of equations for the amplitude factors \hat{x} , \hat{y} and $\hat{\varphi}$:

$$(\lambda^2 + \omega_n^2) \hat{x} + \omega_n^2 r_y \hat{\varphi} = 0 \quad (81)$$

$$(\lambda^2 + \omega_n^2) \hat{y} - \omega_n^2 r_x \hat{\varphi} = 0 \quad (82)$$

$$(\lambda^2 + \omega_I^2 (r_x^2 + r_y^2)) \hat{\varphi} + \omega_I^2 r_y \hat{x} - \omega_I^2 r_x \hat{y} = 0 \quad (83)$$

A non-trivial solution exists if the determinant of the system vanishes. This leads to the characteristic equation

	Equations of Motion for Satellite and Test Mass Dynamics	Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 31 of 44
---	---	---

$$\begin{vmatrix} (\lambda^2 + \omega_n^2) & 0 & \omega_n^2 r_y \\ 0 & (\lambda^2 + \omega_n^2) & -\omega_n^2 r_x \\ \omega_I^2 r_y & -\omega_I^2 r_x & (\lambda^2 + b) \end{vmatrix} = \lambda^6 + \lambda^4(2\omega_n^2 + b) + \lambda^2\omega_n^2(\omega_n^2 + b) = 0 \quad (84)$$

$$b = \omega_I^2(r_x^2 + r_y^2)$$

As can be seen immediately the first eigenvalue $\omega_1^2 = -\lambda_1^2 = 0$. The second and third eigenvalues are calculated as $\omega_2^2 = -\lambda_2^2 = \omega_n^2$ and $\omega_3^2 = -\lambda_3^2 = \omega_n^2 + \omega_I^2(r_x^2 + r_y^2)$. The solution for x , y and φ can be expressed as

$$x = \hat{x}_1 \cos(\omega_1 t - \Psi_{x1}) + \hat{x}_2 \cos(\omega_2 t - \Psi_{x2}) + \hat{x}_3 \cos(\omega_3 t - \Psi_{x3}) \quad (85)$$

$$y = \hat{y}_1 \cos(\omega_1 t - \Psi_{y1}) + \hat{y}_2 \cos(\omega_2 t - \Psi_{y2}) + \hat{y}_3 \cos(\omega_3 t - \Psi_{y3}) \quad (86)$$

$$\varphi = \hat{\varphi}_1 \cos(\omega_1 t - \Psi_{\varphi1}) + \hat{\varphi}_2 \cos(\omega_2 t - \Psi_{\varphi2}) + \hat{\varphi}_3 \cos(\omega_3 t - \Psi_{\varphi3}) \quad (87)$$

$$(88)$$

The solution for the non-homogeneous system of equations is $x_p = r_x$, $y_p = r_y$. Using equations (81) to (83) for the relationship between the unknowns $\hat{x}_{(1,2,3)}$, $\hat{y}_{(1,2,3)}$, $\hat{\varphi}_{(1,2,3)}$, $\Psi_{(x,y,\varphi)(1,2,3)}$ and adding the particular solution we arrive at:

$$x = \hat{x}_1 + \hat{x}_2 \cos(\omega_2 t) + \hat{x}_3 \cos(\omega_3 t) + r_x \quad (89)$$

$$y = -\frac{r_x}{r_y}(\hat{x}_1 + \hat{x}_3 \cos(\omega_3 t)) + \frac{r_y}{r_x} \hat{x}_2 \cos(\omega_2 t) + r_y \quad (90)$$

$$\varphi = -\frac{1}{r_y} \hat{x}_1 + \frac{\omega_I^2 (r_x^2 + r_y^2)}{\omega_n^2 r_y} \hat{x}_3 \cos(\omega_3 t) \quad (91)$$

$$(92)$$

Now, only three unknowns \hat{x}_1 , \hat{x}_2 and \hat{x}_3 need to be determined from initial conditions:

$$\varphi(0) = 0 : \quad \hat{x}_1 = \frac{\omega_I^2}{\omega_n^2} (r_x^2 + r_y^2) \hat{x}_3 \quad (93)$$

$$x(0) = x_0 + r_x : \quad \hat{x}_2 = x_0 - \left(\frac{\omega_I^2}{\omega_n^2} (r_x^2 + r_y^2) + 1 \right) \hat{x}_3 \quad (94)$$

$$y(0) = y_0 + r_y : \quad \hat{x}_3 = \left(\frac{r_y}{r_x} x_0 - y_0 \right) \frac{r_x r_y}{\left(1 + \frac{\omega_I^2}{\omega_n^2} (r_x^2 + r_y^2) \right) (r_x^2 + r_y^2)} \quad (95)$$

with $x_0 = r_{sens,tm}(1)$, $y_0 = r_{sens,tm}(2)$.

In figures 15 and 16 the x- and y-position of the test mass are shown. The results from the modified simulator are compared against the original one and the analytical solution (equations (89) and (90)).

In figure 14 the satellite angular velocity computed with the modified simulator is compared against the original one and the first derivative of the analytical solution, equation (91).

There are small deviations between the analytical solution and the simulation results because of the simplifications made in the analytical approach.

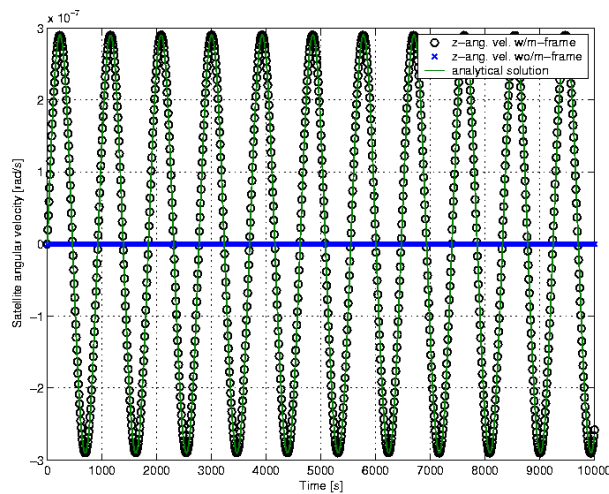


Figure 14: Satellite z-component of angular velocity calculated with modified drag-free simulator compared to original simulator and analytical solution.

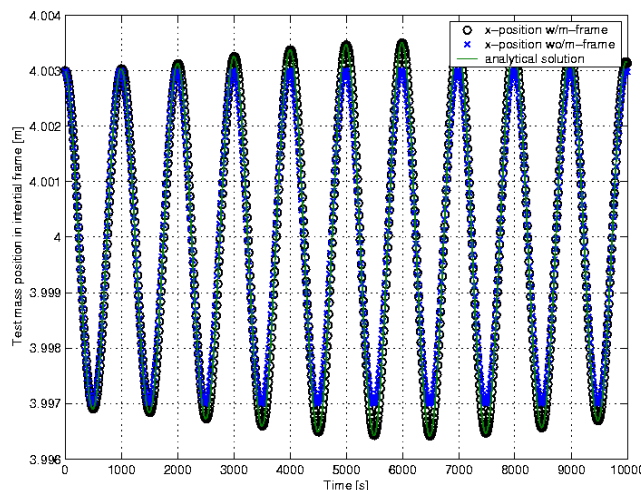


Figure 15: Test mass x-position calculated with modified drag-free simulator compared to original simulator and analytical solution.

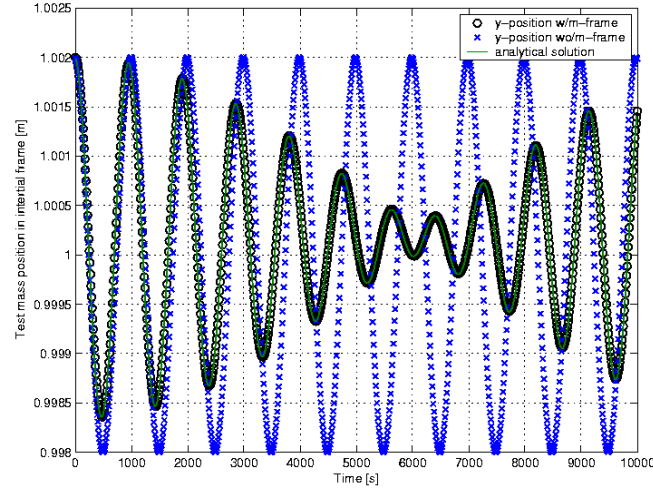


Figure 16: Test mass y-position calculated with modified drag-free simulator compared to original simulator and analytical solution.

7.2. Constant Variation of Satellite Mass

For this investigation a constant value is subtracted from the satellite mass at each time step. Although the satellite mass is changing with time the satellite mass is considered constant at each time step for the calculation of satellite and test mass equations of motion. This assumption is reasonable since the reduction of satellite mass through fuel consumption is rather small, i.e. of order $O(10^{-7}\text{kg})$ for cold gas to $O(10^{-11}\text{kg})$ for FEED.

For this test the satellite degrees of freedom are restricted to motion in the x-y-plane and rotation around the z-axis. For the test mass only translational motion is allowed. Note that for zero satellite position and velocity the transformation matrix from orbital to inertial frame will have only entries equal to zero.

To focus on the effect of changing satellite mass only the torque from coupling forces is eliminated by setting the lever arm $\underline{r}_{b,tm}$ to $\underline{r}_{sens,tm}$ (as defined in (98)), i.e. $\underline{r}_{m,a} - \underline{r}_{m,b} + \underline{r}_{a,sens} = 0$ and

$$\underline{\underline{K}}_{trans} = \begin{bmatrix} \omega_n^2 & 0 & 0 \\ 0 & \omega_n^2 & 0 \\ 0 & 0 & \omega_n^2 \end{bmatrix} \quad (96)$$

The torque from coupling forces becomes zero:

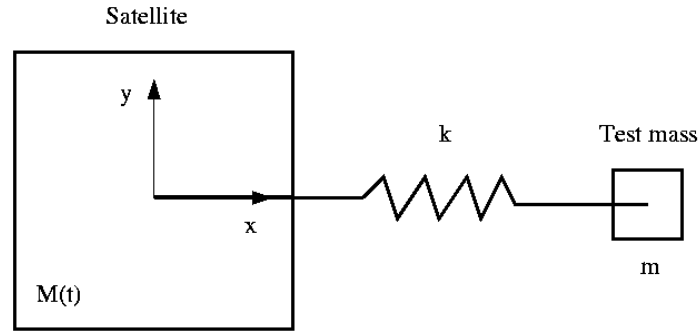


Figure 17: Set-up for test of variation of satellite mass.

$$\underline{T}_{tr} = \begin{bmatrix} r_{sens,tm,x} \\ r_{sens,tm,y} \\ r_{sens,tm,z} \end{bmatrix} \times \begin{bmatrix} \omega_n^2 & 0 & 0 \\ 0 & \omega_n^2 & 0 \\ 0 & 0 & \omega_n^2 \end{bmatrix} \cdot \begin{bmatrix} r_{sens,tm,x} \\ r_{sens,tm,y} \\ r_{sens,tm,z} \end{bmatrix} = 0 \quad (97)$$

(Note: The subtraction of equal numbers causes the modified simulator to become unstable due to numerical noise.)

The test mass position is set to

$$\underline{r}_{sens,tm}^{sens} = \begin{bmatrix} 3.0 \cdot 10^{-3} \\ 0 \\ 0 \end{bmatrix} m \quad (98)$$

All other initial conditions are set to zero.

The following figures show results for a mass reduction rate of 10^{-2} kg/s for the satellite and test mass. The results are compared against those obtained from a simple simulink model (see figure 18) for the system shown in figure 17.

The satellite acceleration is equal to $\ddot{r}_s = \underline{F}/M$ where M is the satellite mass. Since M is decreasing an increase in the amplitude of satellite position and velocity is expected. Figures 19 and 20 show simulation results for the x-position and velocity of the satellite. The expected increase in amplitude over time can be observed.

For the test mass the acceleration is calculated in dependency of the satellite acceleration, $\ddot{r}_{tm} = -\underline{F}/m - \ddot{r}_s = -(\underline{F}/m + \underline{F}/M)$ where m is the mass of the test mass. Figures 21 and 22 show simulation results for the non-zero x-components of the test mass state vector. The test mass velocity and elongation are slightly decreasing with time.

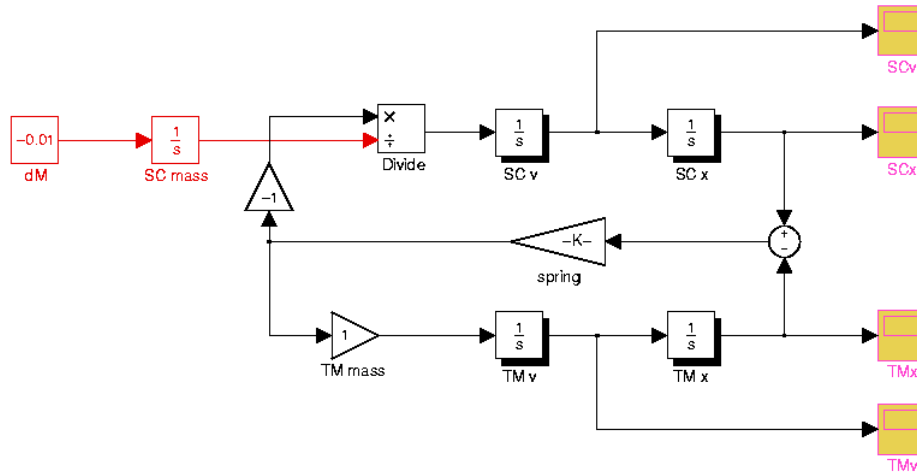


Figure 18: Simulink model for test of variation of satellite mass.

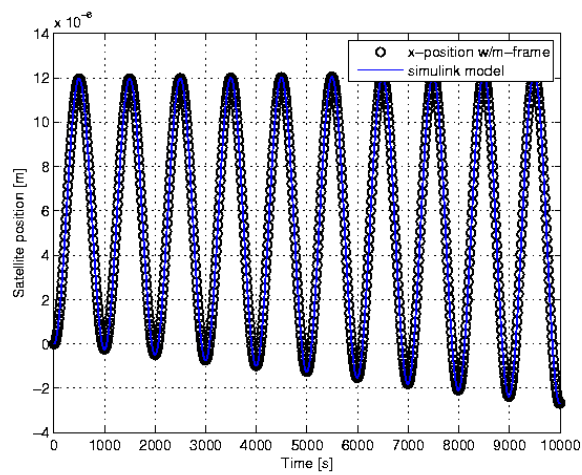


Figure 19: Constant Variation of Satellite Mass: Satellite x-position calculated with modified drag-free simulator and simulink model.

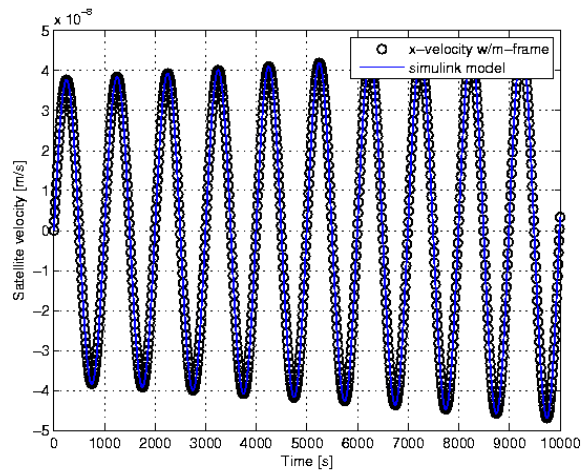


Figure 20: Constant Variation of Satellite Mass: Satellite x-velocity calculated with modified drag-free simulator and simulink model.

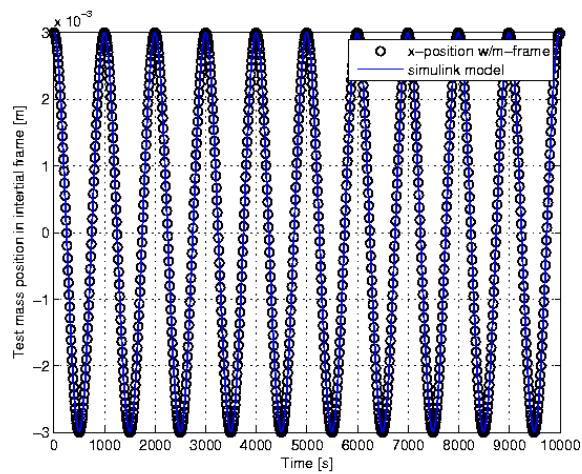


Figure 21: Constant Variation of Satellite Mass: Test mass x-position calculated with modified drag-free simulator and simulink model.

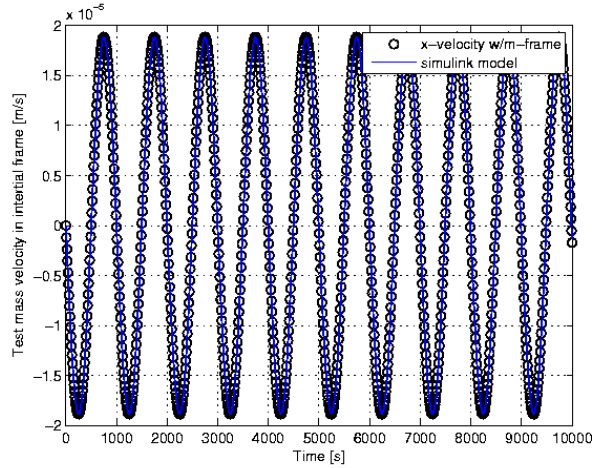


Figure 22: Constant Variation of Satellite Mass: Test mass x-velocity calculated with original and modified drag-free simulator.

7.3. Constant Variation of Satellite Center of Mass

For this investigation a constant value is added to the displacement vector $\underline{r}_{m,b}^m$ between satellite center of mass and origin of mechanical reference frame for each time step. The value chosen is defined as an offset of $\underline{r}_{m,b}^m$:

$$\underline{r}_{m,b,offset}^m = \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix}, \quad d = 1.5 \cdot 10^{-3}m \quad (99)$$

At each time step $\underline{r}_{m,b}^m$ is updated:


$$\left(\underline{r}_{m,b}^m\right)^n = \left(\underline{r}_{m,b}^m\right)^{n-1} + \underline{r}_{m,b,offset}^m, \quad n = 1, \dots, N_{end} \quad (100)$$

As a counter-check the same procedure is repeated with constant $\underline{r}_{m,b}^m$ and varying $\underline{r}_{m,a}^m$:

$$\left(\underline{r}_{m,a}^m\right)^n = \left(\underline{r}_{m,a}^m\right)^{n-1} - \underline{r}_{m,b,offset}^m, \quad n = 1, \dots, N_{end} \quad (101)$$

The initial position vectors for this test case are:

$$\underline{r}_{sens,tm}^{sens} = \begin{bmatrix} 3.0 \cdot 10^{-3} \\ 0 \\ 0 \end{bmatrix} m \quad (102)$$

	<p>Equations of Motion for Satellite and Test Mass Dynamics</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 38 of 44</p>
---	---	--

$$\underline{r}_{m,a}^m = (\underline{r}_{m,b}^m)^0 + \begin{bmatrix} 0.0 \\ 1.0 \\ 0.0 \end{bmatrix} m, \quad (\underline{r}_{m,b}^m)^0 = 0 \quad (103)$$

For the link force a matrix with only one non-zero component is applied:

$$\underline{\underline{K}}_{trans} = \begin{bmatrix} k & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{N}{m}, \quad k = \omega_n^2 \quad (104)$$

with $\omega_n = 2\pi \cdot 0.001 \cdot s^{-1}$. For this test the satellite degrees of freedom are again restricted to rotational motion only. In case that only translational movement is allowed for the test mass, the equations of motion for this test case can be written as

$$m\ddot{x} + kx + kr_y\varphi = 0 \quad (105)$$

$$I_{zz}\ddot{\varphi} + kr_y^2\varphi + kr_yx = 0 \quad (106)$$

Here, again $m = 1\text{kg}$, $I_{zz} = 100\text{kg}\cdot\text{m}^2$. For constant $r_y = r_{m,a}(2) - r_{m,b}(2)$, the analytical solution to equations (105) and (106) is:

$$x = \hat{x}_1 + \hat{x}_2\cos(\omega_2t) \quad (107)$$

$$\varphi = -\frac{1}{r_y}\hat{x}_1 + \frac{m}{I_{zz}}r_y\hat{x}_2\cos(\omega_2t) \quad (108)$$

with

$$\omega_2^2 = \frac{k}{m} + \frac{k}{I_{zz}}r_y^2, \quad \hat{x}_1 = \frac{mx_0}{mr_y^2 + I_{zz}}, \quad \hat{x}_2 = \frac{I_{zz}x_0}{mr_y^2 + I_{zz}}, \quad x_0 = r_{sens,tm}(1)$$

In figure 23 the solution for the satellite angular velocity, i.e.

$$\dot{\varphi} = -\omega_2 \frac{m}{I_{zz}} r_y \hat{x}_2 \sin(\omega_2 t) \quad (109)$$

is plotted as blue curve for comparison with the z-component of satellite angular velocity calculated with constantly varying lever arm from the satellite center of mass to the test mass as specified in equations (100) or (101).

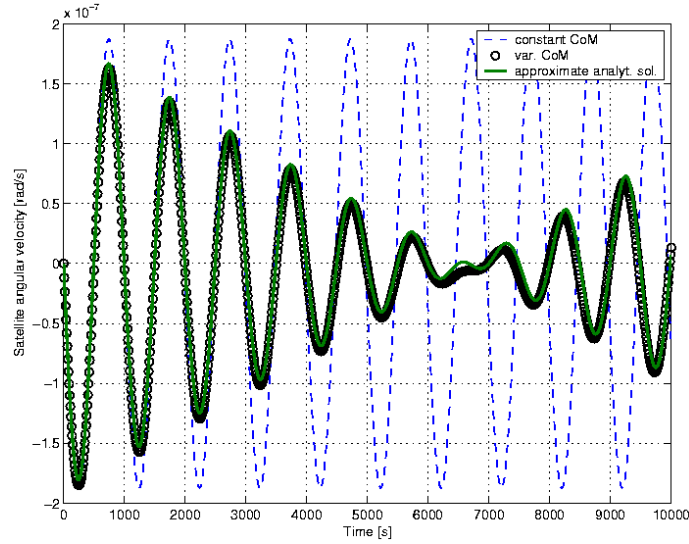


Figure 23: Constant Variation of Satellite Center of Mass: Satellite z-angular velocity calculated with modified drag-free simulator with and without CoM variation, comparison with approximate analytical solution

The equations of motion are rewritten in the following for constantly varying lever arm:

$$m\ddot{x}(t) + kx(t) + kr_y(t)\varphi(t) = 0 \quad (110)$$

$$I_{zz}\ddot{\varphi}(t) + kr_y^2(t)\varphi(t) + kr_y(t)x(t) = 0 \quad (111)$$

It is not attempted here to solve this nonlinear coupled system of equations. For small movement of satellite center of mass the solution (109) can serve as approximation to predict the trend of the angular velocity. If r_y is updated at each time step, the approximate analytical solution also shown in figure 23 predicts a zero-crossing between time steps $n = 6666$ and $n = 6667$ due to the lever arm $r_{m,a}(2) - (r_{m,b}(2))^n = 0$.

Below, an exact analytical solution is found for a simplified test case: In the following, the test mass degrees of freedom are all fixed. Then the equation of motion for the satellite angular velocity becomes:

$$I_{zz}\ddot{\varphi}(t) = -kr_y(t)\Delta x, \quad \Delta x = x_0 + r_y(t)\varphi_0 = x_0 \quad (\text{since } \varphi_0 = 0) \quad (112)$$

The lever arm is decreasing constantly with time, $r_y(t) = r_{y,c} - d \cdot t$, $r_{y,c} = r_{m,a}(2) - (r_{m,b}(2))^0$. The solution for the satellite angular velocity thus is:

$$\dot{\varphi}(t) = -\frac{kx_0}{I_{zz}} \left(r_{y,c}t - \frac{d}{2}t^2 \right) \quad (113)$$

In figure 24 the results from the drag-free simulator with fixed test mass degrees of freedom are compared against the analytical solution (113). As a reference also the corresponding simulator results for constant lever arm (fixed satellite center of mass) are plotted. This curve is a straight line.

For constant $r_{m,b}^m$ and varying $r_{m,a}^m$ according to (101) the same results are obtained as shown in figures 23 and 24.

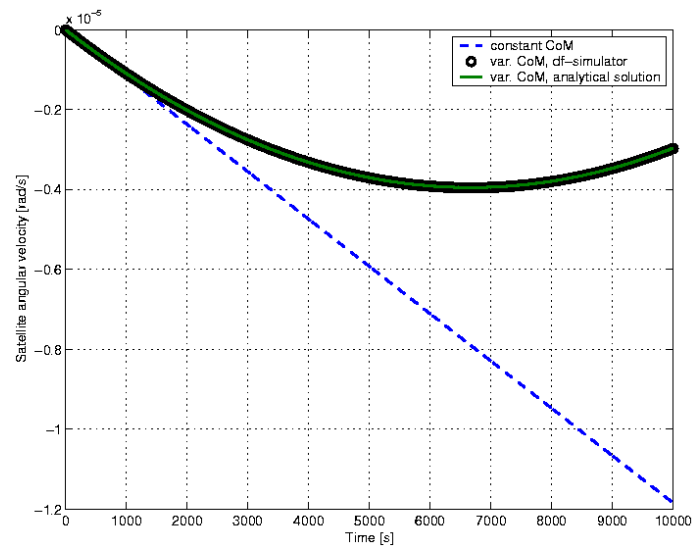



Figure 24: Constant Variation of Satellite Center of Mass: Satellite z-angular velocity calculated with modified drag-free simulator for fixed test mass degrees of freedom for constant and varying satellite center of mass. Green line: Analytical solution for fixed test mass degrees of freedom.

	Equations of Motion for Satellite and Test Mass Dynamics	Doc.No.: FLK-SIM-TN-ZAR-001 Issue: 1.0 Page: 41 of 44
---	---	---

References

- [1] Benjamin Lange. *The Control and Use of Drag-Free Satellites*. PhD thesis, Department of Aeronautics and Astronautics of Stanford University, June 1964.
- [2] O. Montenbruck and E. Gill. *Satellite Orbits*. Springer, 2000.
- [3] Stephan Theil. *Satellite and Test Mass Dynamics Modeling and Observation for Drag-free Satellite Control of the STEP Mission*. PhD thesis, Department of Production Engineering, University of Bremen, December 2002.
- [4] J.R. Wertz, editor. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978.

A. Euler Symmetric Parameters - Quaternions

A.1. Definition of Euler Symmetric Parameters

The direction cosine matrix can be parameterized in terms of Euler symmetric parameters q_1, q_2, q_3 and q_4 . They are defined as (see [4]):

$$\begin{aligned}
 q_1 &= e_x \sin \frac{\Phi}{2} \\
 q_2 &= e_y \sin \frac{\Phi}{2} \\
 q_3 &= e_z \sin \frac{\Phi}{2} \\
 q_4 &= \cos \frac{\Phi}{2}
 \end{aligned} \tag{114}$$

They can be regarded as the components of a quaternion.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{115}$$

Since a quaternion expresses a transformation the notation of super and subscripts corresponds to transformation matrices. So q_a^b denotes a quaternion describing the transformation from frame a to frame b .

A.2. Quaternion Algebra

Quaternion Multiplication

For combination of transformations quaternions can be multiplied. The multiplication is defined as:

$$\underline{q}'' = \underline{q}' \odot \underline{q} \tag{116}$$

where the operator \odot denotes the following matrix-vector operation:

$$\underline{q}'' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{117}$$

Direction Cosine Matrix from Quaternion

The direction cosine matrix can be derive from a quaternion by:

$$\underline{\underline{A}}(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2 q_1 q_2 + 2 q_3 q_4 & 2 q_1 q_3 - 2 q_2 q_4 \\ 2 q_1 q_2 - 2 q_3 q_4 & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2 q_2 q_3 + 2 q_1 q_4 \\ 2 q_1 q_3 + 2 q_2 q_4 & 2 q_2 q_3 - 2 q_1 q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (118)$$

Time-Derivative of Quaternion

If a quaternion represents the attitude of a rotating rigid body its derivative w.r.t. time is needed for propagation. For the attitude of body b w.r.t. the reference frame a the transformation can be expressed by \mathbf{q}_a^b . The angular velocity of b w.r.t. the reference a measured in the frame b is expressed as $\omega_{a,b}^b$.

The derivative w.r.t. time is:

$$\dot{\underline{\underline{q}}} = \frac{1}{2} \underline{\underline{\Omega}} \mathbf{q}_a^b \quad (119)$$

where $\underline{\underline{\Omega}}$ is defined as:

$$\underline{\underline{\Omega}} = \begin{bmatrix} 0 & \omega_{a,b_z}^b & -\omega_{a,b_y}^b & \omega_{a,b_x}^b \\ -\omega_{a,b_z}^b & 0 & \omega_{a,b_x}^b & \omega_{a,b_y}^b \\ \omega_{a,b_y}^b & -\omega_{a,b_x}^b & 0 & \omega_{a,b_z}^b \\ -\omega_{a,b_x}^b & -\omega_{a,b_y}^b & -\omega_{a,b_z}^b & 0 \end{bmatrix} \quad (120)$$

Using the quaternion multiplication rule the time derivative can be expressed as:

$$\dot{\underline{\underline{q}}} = \frac{1}{2} \hat{\underline{\underline{\omega}}}_{a,b}^b \odot \mathbf{q}_a^b \quad (121)$$

where:

$$\hat{\underline{\underline{\omega}}}_{a,b}^b = \begin{bmatrix} \omega_{a,b_x}^b \\ \omega_{a,b_y}^b \\ \omega_{a,b_z}^b \\ 0 \end{bmatrix} \quad (122)$$

B. Time Derivative of a Vector in a Rotating Frame

The time derivative of a vector \underline{r} in a frame A which is rotating w.r.t. the inertial frame B with $\underline{\omega}_{A,B}^B$ is:

$${}^{(A)}\left(\frac{d\underline{r}}{dt}\right) = {}^{(B)}\left(\frac{d\underline{r}}{dt}\right) + \underline{\omega}_{A,B}^B \times \underline{r} \quad (123)$$

Then the second derivative of vector \underline{r} w.r.t. time in the rotating frame A can be derived as:

$${}^{(A)}\left(\frac{d}{dt} {}^{(A)}\left(\frac{d\underline{r}}{dt}\right)\right) = {}^{(A)}\left(\frac{d}{dt} {}^{(B)}\left(\frac{d\underline{r}}{dt}\right)\right) + {}^{(A)}\left(\frac{d}{dt} (\underline{\omega}_{A,B}^B \times \underline{r})\right) \quad (124)$$

$${}^{(A)}\frac{d^2\underline{r}}{dt^2} = \frac{{}^{(B)}d^2\underline{r}}{dt^2} + \underline{\omega}_{A,B}^B \times \frac{{}^{(B)}d\underline{r}}{dt} + \frac{{}^{(A)}d\underline{\omega}_{A,B}^B}{dt} \times \underline{r} + \underline{\omega}_{A,B}^B \times \frac{{}^{(A)}d\underline{r}}{dt} \quad (125)$$

$$= \frac{{}^{(B)}d^2\underline{r}}{dt^2} + \underline{\omega}_{A,B}^B \times \frac{{}^{(B)}d\underline{r}}{dt} + \frac{{}^{(B)}d\underline{\omega}_{A,B}^B}{dt} \times \underline{r} + (\underline{\omega}_{A,B}^B \times \underline{\omega}_{A,B}^B) \times \underline{r} + \underline{\omega}_{A,B}^B \times \frac{{}^{(B)}d\underline{r}}{dt} + \underline{\omega}_{A,B}^B \times (\underline{\omega}_{A,B}^B \times \underline{r}) \quad (126)$$

$$= \frac{{}^{(B)}d^2\underline{r}}{dt^2} + 2\underline{\omega}_{A,B}^B \times \frac{{}^{(B)}d\underline{r}}{dt} + \frac{{}^{(B)}d\underline{\omega}_{A,B}^B}{dt} \times \underline{r} + \underline{\omega}_{A,B}^B \times (\underline{\omega}_{A,B}^B \times \underline{r}) \quad (127)$$



The STEP Test Case Repeated
Considering the Quadrupole Effect as
Common Mode Signal

Doc.No.: FLK-SIM-TN-ZAR-003

Issue: 1.0

Page: 1 of 19

The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal


Project: FIRST LOOK	Document No.: FLK-SIM-TN-ZAR-003
ZARM - Center of Applied Space Technology and Microgravity Am Fallturm Phone: +49-421-218-9422 D 28359 Bremen Fax: +49-421-218-4356 Germany E-Mail: pelivan@zarm.uni-bremen.de	

Doc. No.: FLK-SIM-TN-ZAR-003

Issue: 1.0

Written: Ivanka Pelivan Date: 21st March 2006

Approved: - Date: -


	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 2 of 19</p>
---	--	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	03/21/06	all	Initial Version by Ivanka Pelivan	√

Contents

1. Introduction	4
2. Satellite and Test Mass Dynamics	4
2.1. Satellite Equations of Motion	4
2.2. Test Mass Equations of Motion	6
3. STEP Test Case Setup	7
3.1. Reference Test Case	7
3.2. Modifications to the Reference Test Case	10
4. STEP Test Case Results	11
5. Parametric Study	13
6. Conclusion	16
A. Euler Symmetric Parameters - Quaternions	18
A.1. Definition of Euler Symmetric Parameters	18
A.2. Quaternion Algebra	18

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 4 of 19</p>
---	--	---

1. Introduction

A test case has been defined in [4] for simulating the satellite and test mass dynamics of the STEP (Satellite Test of the Equivalence Principle) mission. The test case consists of two simulations, one carried out with an assumed Equivalence Principle (EP) violation, the other one without. Comparison of the results for the differential accelerations between a pair of test masses is used to detect the simulated EP signal.

The following report summarizes the investigation of the previously defined test case for the STEP reference simulation [4] taking into account the presence of the satellite's quadrupolar potential. A parametric study is performed to evaluate certain effects on the test masses' differential accelerations observed due to interaction of quadrupolar acceleration and spring coupling.

In section 2 the satellite and test mass equations of motion are presented as used in the simulation. Section 3 describes the reference test case [4] and the modifications carried out in the present study. The following section 4 presents the obtained results. In the parametric study in section 5 the test case is evaluated with changes in the satellite's rotational rate, natural frequency of the test masses etc.

2. Satellite and Test Mass Dynamics

In the following the satellite and test mass equations of motion are introduced as implemented in the drag-free simulator.

2.1. Satellite Equations of Motion

The equations of motion for the satellite are derived in the inertial frame.

Satellite Translation

The differential equation for the position of the satellite center of mass $\underline{r}_{i,b}^i$ is written below (see also [5, 6]):

$$m_{sat}\ddot{\underline{r}}_{i,b}^i = m_{sat}g_{i,b}^i(\underline{r}_{i,b}^i) + \underline{F}_{control}^i + \underline{F}_{dist,sat}^i + \underline{F}_{coupl,sat}^i \quad (1)$$


where:

m_{sat}

Mass of satellite.

$\ddot{\underline{r}}_{i,b}^i$

Acceleration of the satellite relative to the inertial frame expressed in the inertial frame.

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 5 of 19</p>
---	--	---

$\underline{g}_{i,b}^i$	Gravitational acceleration as a function of the satellite's position.
$\underline{F}_{\text{control}}^i$	Control force.
$\underline{F}_{\text{dist,sat}}^i$	Sum of all disturbance forces acting on the satellite.
$\underline{F}_{\text{coupl,sat}}^i$	Force on the satellite due to the coupling between satellite and all test masses.

Dividing by the mass of the satellite the equation of motion is expressed in accelerations and specific forces:

$$\ddot{\underline{r}}_{i,b}^i = \underline{g}_{i,b}^i (\underline{r}_{i,b}^i) + \underline{f}_{\text{control}}^i + \underline{f}_{\text{dist,sat}}^i + \underline{f}_{\text{coupl,sat}}^i \quad (2)$$

where \underline{f}_* denotes the corresponding specific forces.

Satellite Attitude

The attitude motion of the satellite is expressed by the angular velocity of the satellite body w.r.t. the inertial frame, $\underline{\omega}_{i,b}^b$. The attitude itself is described by the Euler symmetric parameters \mathbf{q}_i^b which represent a transformation from inertial frame to satellite body fixed frame (see appendix A).


For the satellite a rigid body is assumed. Then the differential equations for the satellite attitude motion are as follows:

$$\dot{\underline{\omega}}_{i,b}^b = \left(\underline{I}_{\underline{b}}^b \right)^{-1} \left[\underline{T}_{\text{control}}^b + \underline{T}_{\text{dist,sat}}^b + \underline{T}_{\text{gg}}^b + \underline{T}_{\text{coupl,sat}}^b - \underline{\omega}_{i,b}^b \times \left(\underline{I}_{\underline{b}}^b \underline{\omega}_{i,b}^b \right) \right] \quad (3)$$

$$\dot{\mathbf{q}}_i^b = \frac{1}{2} \hat{\underline{\omega}}_{i,b}^b \odot \mathbf{q}_i^b \quad (4)$$

where:

$\underline{\omega}_{i,b}^b$	Angular velocity of the satellite w.r.t. inertial frame expressed in body-fixed coordinate frame.
$\underline{I}_{\underline{b}}^b$	Moments of inertia matrix of the satellite.
$\underline{T}_{\text{control}}^b$	Control torques applied for attitude control expressed in the body-fixed frame.
$\underline{T}_{\text{dist,sat}}^b$	Disturbance torques acting on the satellite expressed in the body-fixed frame.
$\underline{T}_{\text{gg}}^b$	Gravity gradient torque acting on the satellite expressed in the body-fixed frame.
$\underline{T}_{\text{coupl,sat}}^b$	Torques generated from satellite-test mass coupling expressed in the body-fixed frame.
\mathbf{q}_i^b	Attitude quaternion describing the orientation of the satellite body-fixed frame w.r.t. the inertial frame.
\odot	denotes a quaternion multiplication

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 6 of 19</p>
---	--	---

The term $\hat{\underline{\omega}}_{i,b}^b$ is the quaternion representation of the angular velocity (see appendix A section A.2).

2.2. Test Mass Equations of Motion

The equations of motion for the test mass dynamics are solved in the sensor frame.

Test Mass Translation

The equation of motion for the translation of the test mass has been derived as (see [2, 5]):

$$\begin{aligned} \ddot{\underline{r}}_{sens,tm}^{sens} = & \Delta \underline{g}_{b,tm}^{sens} - \underline{f}_{control}^{sens} - \underline{f}_{dist,sat}^{sens} - \underline{f}_{coupl,sat}^{sens} + \underline{f}_{coupl,tm}^{sens} + \underline{f}_{dist,tm}^{sens} - \\ & 2 \underline{\omega}_{i,b}^{sens} \times \dot{\underline{r}}_{sens,tm}^{sens} - \dot{\underline{\omega}}_{i,b}^{sens} \times (\underline{r}_{b,sens}^{sens} + \underline{r}_{sens,tm}^{sens}) - \\ & \underline{\omega}_{i,b}^{sens} \times (\underline{\omega}_{i,b}^{sens} \times (\underline{r}_{b,sens}^{sens} + \underline{r}_{sens,tm}^{sens})) \end{aligned} \quad (5)$$

where:

$\ddot{\underline{r}}_{sens,tm}^{sens}$	Acceleration of the test mass relative to the sensor frame expressed in the sensor frame.
$\Delta \underline{g}_{b,tm}^{sens}$	Difference of gravitational acceleration $\underline{g}_{i,tm}^{sens}(\underline{r}_{i,tm}^{sens}) - \underline{g}_{i,b}^{sens}(\underline{r}_{i,b}^{sens})$
$\underline{f}_{control}^{sens}$	Specific control force from satellite.
$\underline{f}_{dist,sat}^{sens}$	Specific disturbance forces acting on the satellite.
$\underline{f}_{coupl,(sat,tm)}^{sens}$	Specific coupling force on the test mass due to the interaction between satellite and other test masses.
$\underline{\omega}_{i,b}^{sens}$	Angular velocity of the satellite body w.r.t. the inertial frame, expressed in the sensor frame.


Test Mass Rotation

The attitude motion of a test mass is expressed by the angular velocity of the test mass w.r.t. the sensor frame, $\underline{\omega}_{sens,tm}^{tm}$.

$$\dot{\underline{\omega}}_{sens,tm}^{tm} = \left(\underline{I}_{tm}^{tm} \right)^{-1} \left[\underline{T}^{tm} - (\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm}) \times \left(\underline{I}_{tm}^{tm} (\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm}) \right) \right] - \dot{\underline{\omega}}_{i,b}^{tm} \quad (6)$$

where:

\underline{T}^{tm}	Sum of all torques acting on the test mass represented in test mass frame.
\underline{I}_{tm}^{tm}	Moments of inertia matrix of the test mass.
$\underline{\omega}_{i,b}^{tm}$	Angular velocity of the satellite w.r.t. inertial frame expressed in test mass frame.

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 7 of 19</p>
---	--	---

If the test mass inside the satellite is shielded from all external non-gravitational forces and torques, the torque in equation (6) is equal to the sum of the rotational coupling torques, the torques due to the gravity gradient acting on the test masses and internal disturbance torques:

$$\underline{T}^{tm} = \underline{T}_{coup,tm}^{tm} + \underline{T}_{gg,tm}^{tm} + \underline{T}_{dist,tm}^{tm} \quad (7)$$

where:

$\underline{T}_{gg,tm}^{tm}$	Gravity gradient torque for the test mass from Earth gravity field as well as from gravity gradient inside the satellite.
$\underline{T}_{coup,tm}^{tm}$	Torque on the test mass due to satellite test mass coupling.

The attitude of the test mass w.r.t. the sensor frame can be expressed by quaternions:

$$\dot{\mathbf{q}}_{sens}^{tm} = \frac{1}{2} \hat{\omega}_{sens,tm}^{tm} \odot \mathbf{q}_{sens}^{tm} \quad (8)$$

3. STEP Test Case Setup

3.1. Reference Test Case

The STEP satellite moves on an equatorial orbit with an eccentricity of $\epsilon = 0.01$ and an orbit frequency of $\omega_o = 0.00110850809175 \frac{rad}{s}$ corresponding to a period of $T_o = 5668.14s$. The satellites' z-axis is aligned with the orbit normal and the satellite rotates about z with a rate of $\omega_r = -2.0\omega_o$ w.r.t. the inertial frame.

The simulation is carried out using one differential accelerometer with two test masses. The satellite is free to move in six degrees of freedom (DOF) whereas the test mass motion is restricted to the translational DOF.

The simulation parameters are:

step size = 100	for a relatively fast simulation
simulation time = 1.0E+06	End time of simulation (yields a $10^{-6}Hz$ resolution in the frequency spectrum)
gravity model:	simplified model of the gravitational field up to degree $n = 6$ (see equation (9))

$$\Phi = \frac{\mu_E}{r} \left[1 + \sum_{n=2}^{\infty} \left(\frac{R_E}{r} \right)^n J_n P_{n0}(\sin\Theta) \right] \quad (9)$$

with:

R_E	Earth radius.
r	Position vector.
J_n	Earth zonal harmonic coefficients of degree n .
P_{n0}	Legendre polynomials of degree n and order 0.
Θ	Geocentric latitude.

degree $n = 1$	gravitational field up to degree $n = 1$ used (spherical Earth)
gravity-gradient (gg) matrix:	based on spherical field
gg-torques:	no gg-torques on satellite and test masses
gg-acceleration:	gg-accelerations between satellite and test masses computed from spherical gravitational potential

The satellite parameters are:

mass:	500 kg
moments of inertia:	$\underline{\underline{I}} = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 100 \end{bmatrix} \text{ kg m}^2$
initial state:	computed from orbital elements attitude $a = 6871000 \text{ m}$ eccentricity $\epsilon = 0.01$ all other orbital elements zero \rightarrow equatorial orbit
satellite rate:	$\omega_r = -2.0\omega_o$ around z-axis, $\omega_o = 0.00110850809175 \frac{\text{rad}}{\text{s}}$

The accelerometer parameters are:

number of test masses:	2
position w.r.t satellite-fixed frame:	[1.0E-08, 0, 0]

Test Mass 1

mass:	1 kg
moments of inertia:	$\underline{\underline{I}} = \begin{bmatrix} 5.5125 \cdot 10^{-5} & 0 & 0 \\ 0 & 5.5125 \cdot 10^{-5} & 0 \\ 0 & 0 & 5.5125 \cdot 10^{-5} \end{bmatrix} \text{ kg m}^2$

initial state: zero angular velocity
no initial velocity
initial position: [0.9E-09, 0, 0]

test mass alignment with sensor frame

Eötvös factor (EP violation factor): $\eta = 0$: no violation, $\eta = 1.0 \cdot 10^{-18}$ EP violation

Test Mass 2

mass: 1 kg

moments of inertia: $\underline{\underline{I}} = \begin{bmatrix} 3.6125 \cdot 10^{-5} & 0 & 0 \\ 0 & 3.6125 \cdot 10^{-5} & 0 \\ 0 & 0 & 3.6125 \cdot 10^{-5} \end{bmatrix} \text{ kg m}^2$

initial state: zero angular velocity
no initial velocity
initial position: [1.0E-09, 0, 0]

test mass alignment with sensor frame

Eötvös factor: $\eta = 0$: no violation

The coupling between satellite and test masses that is due to the position sensing is modelled as linear system link consisting of a spring stiffness only. Link 1/2 defines the coupling between test mass 1/2 and the satellite, link 3 accounts for the coupling between the two test masses.

Link 1 properties:


stiffness matrix: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \cdot 30.25 \cdot \omega_o^2$

Spring coupling offset vector: [1.0E-09, 0, 0]

Link 2 properties:

stiffness matrix: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \cdot 30.25 \cdot \omega_o^2 \cdot s$

common mode rejection factor: $s = 1.001$

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 10 of 19</p>
---	--	--

spring coupling offset vector: [0.9E-09, 0, 0]

The common mode rejection factor is based on the common mode rejection ratio which is defined as $CMRR = 20 \cdot \log_{10} \left(\frac{\text{Differential Mode}}{\text{Common Mode}} \right)$. If the ratio of spring constant to mass (frequency²) is the same for two masses, they will displace identically under a common acceleration, therefore producing no output in the differential mode. If their frequencies are different, the masses will displace different amounts. In this case a common mode acceleration will cause a disturbance in the differential mode.

Link 3 properties:

stiffness matrix:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot 30.25 \cdot \omega_0^2 / 3.0$$

spring coupling offset vector: [0, 0, 0]

3.2. Modifications to the Reference Test Case

The STEP reference simulation case defined in [4] is repeated here considering the presence of the satellite's quadrupolar force term. A spherical Earth is assumed. The gravitational acceleration on the satellite's and test mass' center of mass (COM) takes the form of equation (15) in [3],

$$\underline{F} = -\frac{\mu_E m}{R^3} \underline{R} + \underline{F}_{g,quad} \quad (10)$$

where $\mu_E = G \cdot M$, G is the gravitational constant of the Earth and M is the mass of the Earth, m is the mass of the satellite or test mass, \underline{R} (R) is the vector (distance) from the Earth's COM to the satellite's or test mass' COM, and $\underline{F}_{g,quad}$ is the gravity-gradient or quadrupolar force term,

$$\underline{F}_{g,quad} = \frac{3\mu_E}{2R^5} \left[5 \frac{\underline{R}^T \underline{I} \underline{R}}{R^2} \underline{R} - tr(\underline{I}) \underline{R} - 2 \underline{I} \underline{R} \right] \quad (11)$$

where \underline{I} is the inertia matrix and $tr(\underline{I})$ is the trace of the inertia matrix.

4. STEP Test Case Results

In a drag-free mission like STEP the on-board experiment, i.e. the test mass set-up, is shielded from all external forces and disturbances. The only forces acting on the test masses result from gravity and other internal disturbances. While the latter are not investigated here, the satellite's quadrupole potential acting as gravitational disturbance on the test masses is given some attention in the following. As pointed out in [1], monopolar test masses, i.e. test masses that do not possess a quadrupolar potential themselves, respond identically to gravitational disturbances due to the spacecraft provided that the test masses are located at the same position. In other words, the satellite's quadrupolar potential results in a common mode signal to both test masses. In the reference test case defined in [4] and detailed in the previous section, the test mass common mode rejection factor is set to 1.001. Thus, the ratio of spring constant to mass is different for the two test masses, which means that any internal disturbance producing a common acceleration will cause a disturbance in the test mass differential mode which might mimic or mask violations of the EP. In order to avoid this scenario the common mode rejection ratio (CMRR) has to be calibrated.

In the following the simulation results are compared via linear spectral density plots of the differential mode of the test masses. In figures 1 and 2 the differential mode of the test masses is shown under the influence of the quadrupolar force term (11) (blue curve) and with the quadrupolar force term omitted (red curve, original calculation). In figure 1 there is no violation of the EP and in figure 2 a violation of the EP is assumed. Therefore, the Eötvös factor for one test mass is set to $\eta = 10^{-18}$.

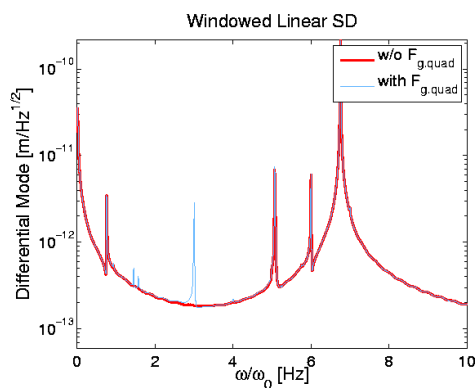


Figure 1: Differential Mode without Equivalence-Principle Violation.

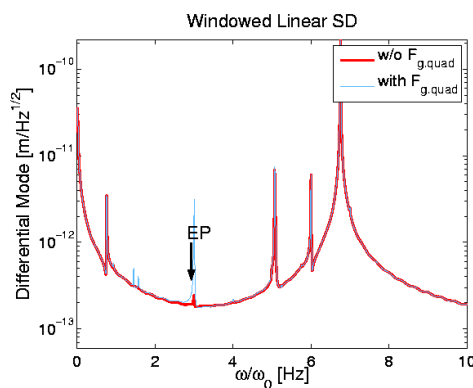


Figure 2: Differential Mode with Equivalence-Principle Violation.

In figure 3 the results with and without EP violation including the influence of the quadrupolar force term are compared against each other. It turns out that a peak due to quadrupole effects appears at the EP frequency and masks the EP signal. It is expected that the same will hold for other internal disturbances that act as common mode signal.

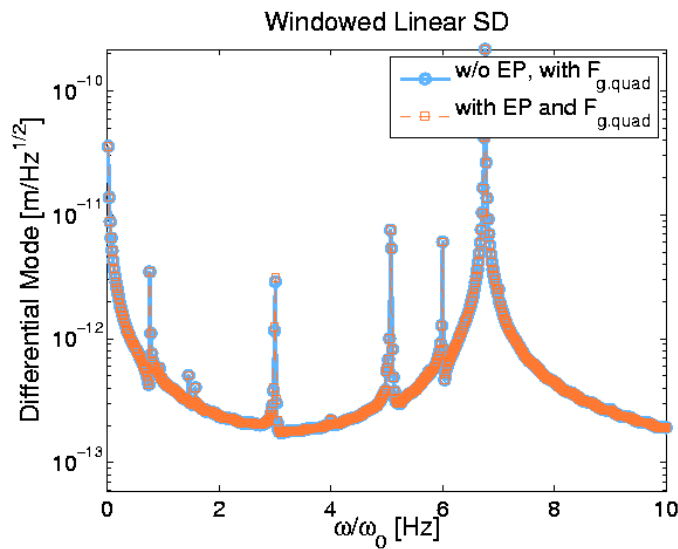


Figure 3: Differential Mode with and without Equivalence-Principle Violation with Quadrupole Effects.

In figures 4 and 5 the test case with EP violation is evaluated with different stiffness coupling and quadrupolar interaction. The difference in stiffness is due to the multiplication of the stiffness matrix of test mass 2 with the common mode rejection factor (CMRF). In figure 4 the quadrupolar force term is neglected. The red line shows the reference test case (CMRF = 1.001) and the dashed blue line depicts a case in which the coupling between test mass 1 and the satellite is equal to the coupling between test mass 2 and the satellite (CMRF = 1).

For equal coupling parameters the results for the differential acceleration are identical regardless of quadrupole effects being taken into account or not. This affirms the theoretical finding that monopolar test masses respond identically to gravitational disturbances due to the spacecraft provided that the test masses share one center of mass as it is realized for the STEP satellite mission where one test mass is located within the other.

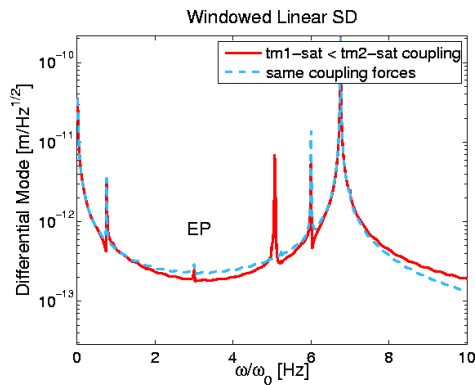


Figure 4: Differential Mode with EP Violation. Equal and unequal coupling interaction between test masses and satellite. Quadrupolar acceleration not considered.

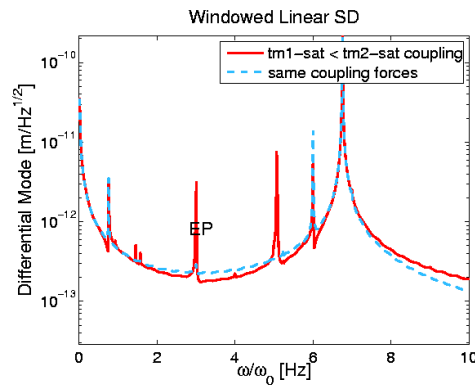


Figure 5: Differential Mode with EP Violation and quadrupolar acceleration. Equal and unequal coupling interaction between test masses and satellite.

5. Parametric Study

In figures 6, 7, and 8 the test case without EP violation has been repeated with the CMRF adjusted to 1.0001, 1.00005, and 1.00001 resp. What can be observed is that with decreasing CMRF the EP mimicking peak due to common mode signals becomes less dominant. For a CMRF factor of 1.00001 a signal is only visible after magnification. For comparison, the reference test case with EP violation and a CMRF of 1.00005 is plotted in figure 9.

Figures 10 and 11 show the test case with EP violation and variation in rotation rate. In figure 10 the rotation rate $\omega_r = -2.5\omega_o$ instead of $\omega_r = -2.0\omega_o$ (reference case) and in figure 11 the rotation rate $\omega_r = -3.0\omega_o$. The EP signal shifts according to $\omega_r - \omega_o$ and the peak associated with the quadrupolar force term does the same. Here, the stiffness multiplication factor $\text{CMRF} = 1.001$ according to the reference test case.

The quadrupolar acceleration shifts with the EP signal since the quadrupolar disturbance is constant. The sketch in figure 12 illustrates this: According to the satellite inertia tensor the STEP satellite possesses a pencil-like shape oriented in z-direction of the orbit reference frame. The satellite moves in the x-y plane with velocity v and rotates around z with rotation rate ω_r . The quadrupolar acceleration thus is independent of the

rotation rate.

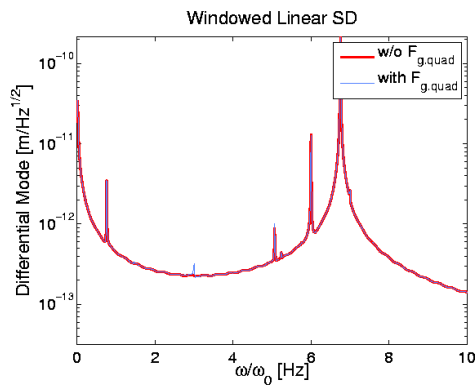


Figure 6: Differential Mode without EP Violation, CMRF = 1.0001

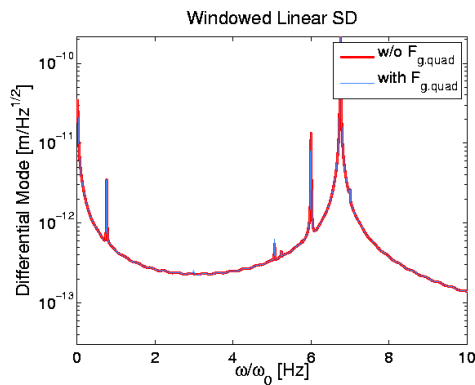


Figure 7: Differential Mode without EP Violation, CMRF = 1.00005

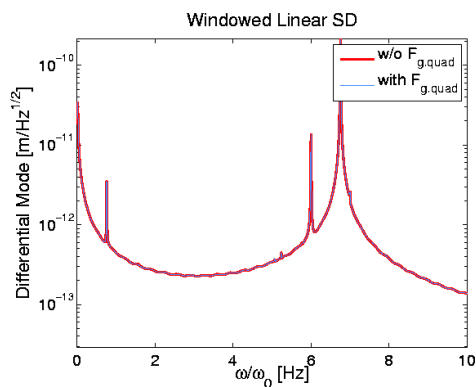


Figure 8: Differential Mode without EP Violation, CMRF = 1.00001

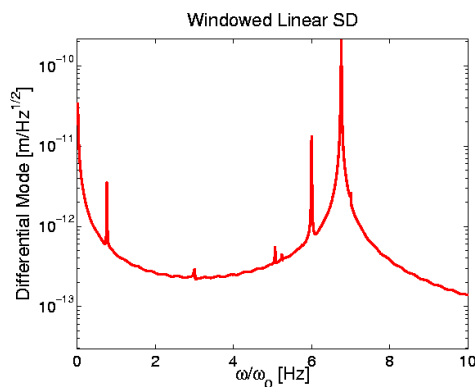


Figure 9: Differential Mode with EP Violation, CMRF = 1.00005

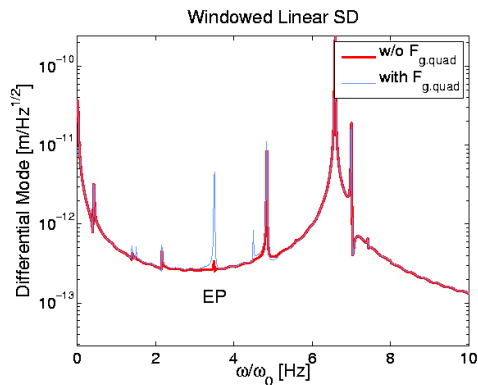


Figure 10: Differential Mode with EP Violation, Rotation Rate $\omega_r = -2.5\omega_0$

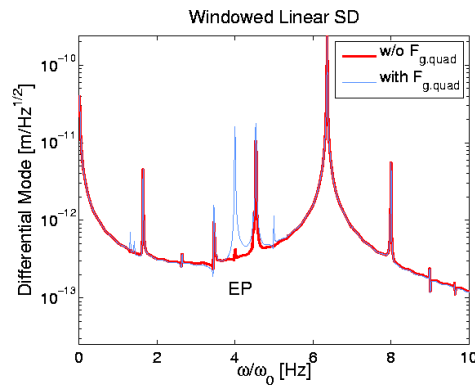


Figure 11: Differential Mode with EP Violation, Rotation Rate $\omega_r = -3\omega_0$

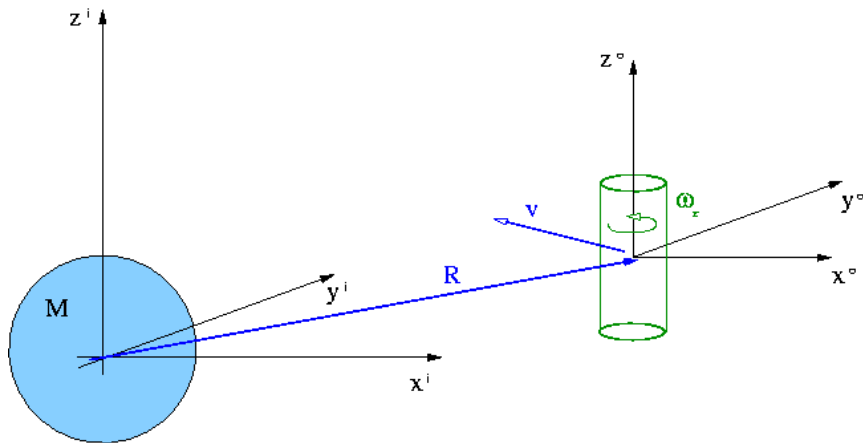


Figure 12: STEP test case reference frames.

In figure 13 the test mass accelerations are displayed for a simulation time of 10000 seconds. The dashed blue curve shows the test mass acceleration where the gravity-gradient induced acceleration is neglected (reference test case). The dash-dotted green curve shows the results after inclusion of the quadrupolar acceleration. For comparison the satellite quadrupolar acceleration is plotted as well (red line). The quadrupolar acceleration is of same order of magnitude as the test mass acceleration of the reference test case.

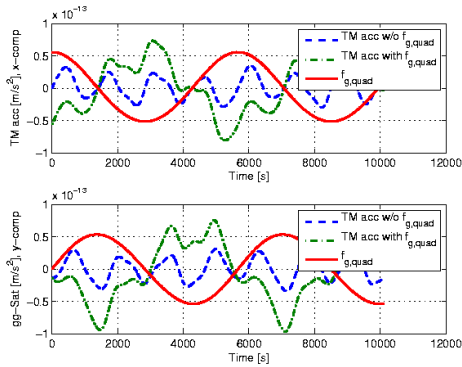


Figure 13: Test mass acceleration and satellite acceleration due to gravity-gradient

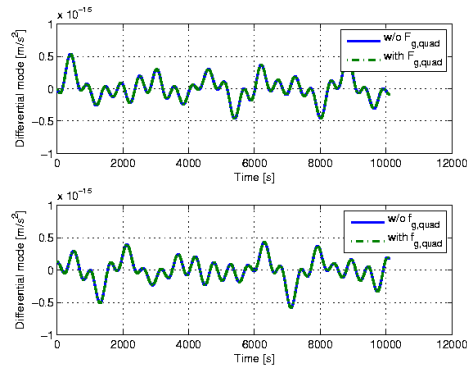


Figure 14: Difference in test mass acceleration


In figure 14 the difference in test mass acceleration for test mass 1 and 2 is plotted for completeness for the same simulation time.

6. Conclusion

Investigation of the STEP test case including coupling between test masses and the satellite from various sources has been carried out. In particular, stiffness effects between test masses and satellite have been modelled through a linear system link. For a satellite of arbitrary shape and homogeneity forces arise due to the gravity-gradient within the satellite. These quadrupolar forces act on the test masses within the satellite. For monopolar test masses located at the same position there is no differential signal due to quadrupolar interaction if the CMRR is sufficiently small. If the quadrupolar acceleration due to the spacecraft is considered together with a difference in the test masses' stiffness matrices, an additional signal emerges at the EP frequency. In order to avoid masking or imitation of the EP signal, the CMRR has to be calibrated to rather small values. Alternatively, the satellite's quadrupolar potential and other internal disturbances that introduce a common mode signal have to be minimized.

References

- [1] N.A. Lockerbie. Gravitational quadrupolar coupling to equivalence principle test masses: the general case. *Classical and Quantum Gravity*, (19):2063–2077, 2002.

	<p>The STEP Test Case Repeated Considering the Quadrupole Effect as Common Mode Signal</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-003 Issue: 1.0 Page: 17 of 19</p>
---	--	--

- [2] I. Pelivan. Equations of Motion for Satellite and Test Mass Dynamics. Technical Report FLK-SIM-TN-ZAR-001, ZARM, University of Bremen, 2005.
- [3] I. Pelivan. Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator. Technical Report FLK-SIM-TN-ZAR-002, ZARM, University of Bremen, 2006.
- [4] S. Theil. STEP Reference Simulation Case for the Generic Drag-Free Simulator. Technical Report STP-SIM-TN-ZAR-003, ZARM, University of Bremen, 2004.
- [5] Stephan Theil. *Satellite and Test Mass Dynamics Modeling and Observation for Drag-free Satellite Control of the STEP Mission*. PhD thesis, Department of Production Engineering, University of Bremen, December 2002.
- [6] J.R. Wertz, editor. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978.

A. Euler Symmetric Parameters - Quaternions

A.1. Definition of Euler Symmetric Parameters

The direction cosine matrix can be parameterized in terms of Euler symmetric parameters q_1, q_2, q_3 and q_4 . They are defined as (see [6]):

$$\begin{aligned}
 q_1 &= e_x \sin \frac{\Phi}{2} \\
 q_2 &= e_y \sin \frac{\Phi}{2} \\
 q_3 &= e_z \sin \frac{\Phi}{2} \\
 q_4 &= \cos \frac{\Phi}{2}
 \end{aligned} \tag{12}$$

They can be regarded as the components of a quaternion.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{13}$$

Since a quaternion expresses a transformation the notation of super and subscripts corresponds to transformation matrices. So q_a^b denotes a quaternion describing the transformation from frame a to frame b .

A.2. Quaternion Algebra

Quaternion Multiplication

For combination of transformations quaternions can be multiplied. The multiplication is defined as:

$$\underline{q}'' = \underline{q}' \odot \underline{q} \tag{14}$$

where the operator \odot denotes the following matrix-vector operation:

$$\underline{q}'' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{15}$$

Direction Cosine Matrix from Quaternion

The direction cosine matrix can be derive from a quaternion by:

$$\underline{\underline{A}}(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2 q_1 q_2 + 2 q_3 q_4 & 2 q_1 q_3 - 2 q_2 q_4 \\ 2 q_1 q_2 - 2 q_3 q_4 & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2 q_2 q_3 + 2 q_1 q_4 \\ 2 q_1 q_3 + 2 q_2 q_4 & 2 q_2 q_3 - 2 q_1 q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (16)$$

Time-Derivative of Quaternion

If a quaternion represents the attitude of a rotating rigid body its derivative w.r.t. time is needed for propagation. For the attitude of body b w.r.t. the reference frame a the transformation can be expressed by \mathbf{q}_a^b . The angular velocity of b w.r.t. the reference a measured in the frame b is expressed as $\omega_{a,b}^b$.

The derivative w.r.t. time is:

$$\dot{\underline{\underline{q}}} = \frac{1}{2} \underline{\underline{\Omega}} \mathbf{q}_a^b \quad (17)$$

where $\underline{\underline{\Omega}}$ is defined as:

$$\underline{\underline{\Omega}} = \begin{bmatrix} 0 & \omega_{a,b_z}^b & -\omega_{a,b_y}^b & \omega_{a,b_x}^b \\ -\omega_{a,b_z}^b & 0 & \omega_{a,b_x}^b & \omega_{a,b_y}^b \\ \omega_{a,b_y}^b & -\omega_{a,b_x}^b & 0 & \omega_{a,b_z}^b \\ -\omega_{a,b_x}^b & -\omega_{a,b_y}^b & -\omega_{a,b_z}^b & 0 \end{bmatrix} \quad (18)$$

Using the quaternion multiplication rule the time derivative can be expressed as:

$$\dot{\underline{\underline{q}}} = \frac{1}{2} \hat{\omega}_{a,b}^b \odot \mathbf{q}_a^b \quad (19)$$

where:

$$\hat{\omega}_{a,b}^b = \begin{bmatrix} \omega_{a,b_x}^b \\ \omega_{a,b_y}^b \\ \omega_{a,b_z}^b \\ 0 \end{bmatrix} \quad (20)$$



Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator


Project:	Document No.:
FIRST LOOK	FLK-SIM-TN-ZAR-002
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm	Phone: +49-421-218-9422
D 28359 Bremen	Fax: +49-421-218-2521
Germany	E-Mail: pelivan@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-002

Issue: 1.1

Written: Ivanka Pelivan Date: 12th April 2006

Approved: - Date: -


	<p>Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 2 of 27</p>
---	---	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	01/06/06	all	Initial Version by Ivanka Pelivan	√
1.1	04/12/06		minor corrections	√

Contents

1. Introduction	4
2. Definition of Coordinate Frames	5
3. Satellite and Test Mass Dynamics	5
3.1. Satellite Equations of Motion	5
3.2. Test Mass Equations of Motion	7
4. Gravity Force and Gravity-Gradient Torque	9
4.1. Gravity Force	9
4.2. Gravity-Gradient Torque	15
5. Validation of Numerical Simulator	17
5.1. Definition of Test Cases	17
5.2. Equal Satellite Inertia	17
5.3. Dumbbell Satellite	18
A. Euler Symmetric Parameters - Quaternions	26
A.1. Definition of Euler Symmetric Parameters	26
A.2. Quaternion Algebra	26

	Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 4 of 27
---	--	--

1. Introduction

This document describes the extension of the gravity force acting on a point mass to the force present for arbitrarily shaped bodies. The gravity force acting on an arbitrary body can be expressed as sum of monopolar and quadrupolar force terms. The monopolar force term is the term acting on the satellite as if it were a point mass or a body with equal inertia in its principal axes. The quadrupolar force term can be considered a correction for deviations from this ideal model. The implementation of the quadrupolar force term in the drag-free simulator is validated with test cases for a satellite with equal inertia in its principal axes and a dumbbell satellite.

2. Definition of Coordinate Frames

The coordinate frames defined for the drag-free simulator are shown in figure 1. The position vectors used in the following are displayed as well. For a more detailed description see [6].

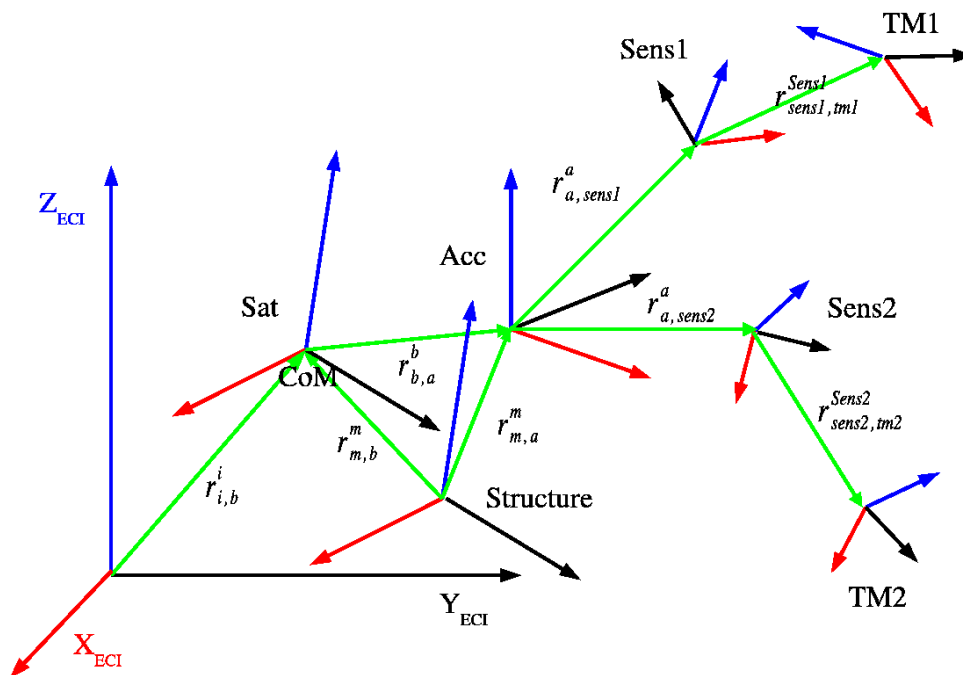


Figure 1: Overview of Coordinate Frames and Position Vectors

3. Satellite and Test Mass Dynamics

In the following the satellite and test mass equations of motion are re-written here as implemented in the drag-free simulator.

3.1. Satellite Equations of Motion

The equations of motion for the satellite are derived in the inertial frame.

Satellite Translation

The differential equation for the position of the satellite center of mass $\underline{r}_{i,b}^i$ is written below (see also [8, 10]):

$$m_{sat}\ddot{\underline{r}}_{i,b}^i = m_{sat}\underline{g}_{i,b}^i(\underline{r}_{i,b}^i) + \underline{F}_{control}^i + \underline{F}_{dist,sat}^i + \underline{F}_{coupl,sat}^i \quad (1)$$

where:

m_{sat}	Mass of satellite.
$\ddot{\underline{r}}_{i,b}^i$	Acceleration of the satellite relative to the inertial frame expressed in the inertial frame.
$\underline{g}_{i,b}^i$	Gravitational acceleration as a function of the satellite's position.
$\underline{F}_{control}^i$	Control force.
$\underline{F}_{dist,sat}^i$	Sum of all disturbance forces acting on the satellite.
$\underline{F}_{coupl,sat}^i$	Force on the satellite due to the coupling between satellite and all test masses.

Dividing by the mass of the satellite the equation of motion is expressed in accelerations and specific forces:

$$\ddot{\underline{r}}_{i,b}^i = \underline{g}_{i,b}^i(\underline{r}_{i,b}^i) + \underline{f}_{control}^i + \underline{f}_{dist,sat}^i + \underline{f}_{coupl,sat}^i \quad (2)$$

where \underline{f}_* denotes the corresponding specific forces.

Satellite Attitude

The attitude motion of the satellite is expressed by the angular velocity of the satellite body w.r.t. the inertial frame, $\underline{\omega}_{i,b}^b$. The attitude itself is described by the Euler symmetric parameters \mathbf{q}_i^b which represent a transformation from inertial frame to satellite body fixed frame (see appendix A).

For the satellite a rigid body is assumed. Then the differential equations for the satellite attitude motion are as follows:

$$\dot{\underline{\omega}}_{i,b}^b = \left(\underline{I}_b^b\right)^{-1} \left[\underline{T}_{control}^b + \underline{T}_{dist,sat}^b + \underline{T}_{gg}^b + \underline{T}_{coupl,sat}^b - \underline{\omega}_{i,b}^b \times \left(\underline{I}_b^b \underline{\omega}_{i,b}^b\right) \right] \quad (3)$$

$$\dot{\mathbf{q}}_i^b = \frac{1}{2} \underline{\hat{\omega}}_{i,b}^b \odot \mathbf{q}_i^b \quad (4)$$

where:

$\underline{\omega}_{i,b}^b$	Angular velocity of the satellite w.r.t. inertial frame expressed in body-fixed coordinate frame.
\underline{I}_b^b	Moments of inertia matrix of the satellite.
$\underline{T}_{control}^b$	Control torques applied for attitude control expressed in the body-fixed frame.
$\underline{T}_{dist,sat}^b$	Disturbance torques acting on the satellite expressed in the body-fixed frame.

\underline{T}_{gg}^b	Gravity gradient torque acting on the satellite expressed in the body-fixed frame.
$\underline{T}_{\text{coupl,sat}}^b$	Torques generated from satellite-test mass coupling expressed in the body-fixed frame.
\mathbf{q}_i^b	Attitude quaternion describing the orientation of the satellite body-fixed frame w.r.t. the inertial frame.
\odot	denotes a quaternion multiplication

The term $\hat{\underline{\omega}}_{i,b}^b$ is the quaternion representation of the angular velocity (see appendix A section A.2).

3.2. Test Mass Equations of Motion

The equations of motion for the test mass dynamics are solved in the sensor frame.

Test Mass Translation

The equation of motion for the translation of the test mass has been derived as (see [6, 8]):

$$\begin{aligned} \ddot{\underline{r}}_{\text{sens},tm}^{\text{sens}} = & \Delta \underline{g}_{b,tm}^{\text{sens}} - \underline{f}_{\text{control}}^{\text{sens}} - \underline{f}_{\text{dist,sat}}^{\text{sens}} - \underline{f}_{\text{coupl,sat}}^{\text{sens}} + \underline{f}_{\text{coupl,tm}}^{\text{sens}} + \underline{f}_{\text{dist,tm}}^{\text{sens}} - \\ & 2 \underline{\omega}_{i,b}^{\text{sens}} \times \dot{\underline{r}}_{\text{sens},tm}^{\text{sens}} - \dot{\underline{\omega}}_{i,b}^{\text{sens}} \times (\underline{r}_{b,\text{sens}}^{\text{sens}} + \underline{r}_{\text{sens},tm}^{\text{sens}}) - \\ & \underline{\omega}_{i,b}^{\text{sens}} \times (\underline{\omega}_{i,b}^{\text{sens}} \times (\underline{r}_{b,\text{sens}}^{\text{sens}} + \underline{r}_{\text{sens},tm}^{\text{sens}})) \end{aligned} \quad (5)$$

where:

$\ddot{\underline{r}}_{\text{sens},tm}^{\text{sens}}$	Acceleration of the test mass relative to the sensor frame expressed in the sensor frame.
$\Delta \underline{g}_{b,tm}^{\text{sens}}$	Difference of gravitational acceleration $\underline{g}_{i,tm}^{\text{sens}}(\underline{r}_{i,tm}^{\text{sens}}) - \underline{g}_{i,b}^{\text{sens}}(\underline{r}_{i,b}^{\text{sens}})$
$\underline{f}_{\text{control}}^{\text{sens}}$	Specific control force from satellite.
$\underline{f}_{\text{dist,sat}}^{\text{sens}}$	Specific disturbance forces acting on the satellite.
$\underline{f}_{\text{coupl,(sat,tm)}}^{\text{sens}}$	Specific coupling force on the test mass due to the interaction between satellite and other test masses.
$\underline{\omega}_{i,b}^{\text{sens}}$	Angular velocity of the satellite body w.r.t. the inertial frame, expressed in the sensor frame.

For a definition of the position vectors $\underline{r}_{\text{sens},tm}$, $\underline{r}_{b,\text{sens}} = \underline{r}_{m,a} - \underline{r}_{m,b} + \underline{r}_{a,\text{sens}}$ see figure 1.

Test Mass Rotation

The attitude motion of a test mass is expressed by the angular velocity of the test mass w.r.t. the sensor frame, $\underline{\omega}_{\text{sens},tm}^{\text{tm}}$.

$$\dot{\underline{\omega}}_{sens,tm}^{tm} = \left(\underline{I}_{tm}^{tm} \right)^{-1} \left[\underline{T}^{tm} - \left(\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm} \right) \times \left(\underline{I}_{tm}^{tm} \left(\underline{\omega}_{i,b}^{tm} + \underline{\omega}_{sens,tm}^{tm} \right) \right) \right] - \dot{\underline{\omega}}_{i,b}^{tm} \quad (6)$$

where:

\underline{T}^{tm}	Sum of all torques acting on the test mass represented in test mass frame.
\underline{I}_{tm}^{tm}	Moments of inertia matrix of the test mass.
$\underline{\omega}_{i,b}^{tm}$	Angular velocity of the satellite w.r.t. inertial frame expressed in test mass frame.

If the test mass inside the satellite is shielded from all external non-gravitational forces and torques, the torque in equation (6) is equal to the sum of the rotational coupling torques, the torques due to the gravity gradient acting on the test masses and internal disturbance torques:

$$\underline{T}^{tm} = \underline{T}_{coup,tm}^{tm} + \underline{T}_{gg,tm}^{tm} + \underline{T}_{dist,tm}^{tm} \quad (7)$$

where:

$\underline{T}_{gg,tm}^{tm}$	Gravity gradient torque for the test mass from Earth gravity field as well as from gravity gradient inside the satellite.
$\underline{T}_{coup,tm}^{tm}$	Torque on the test mass due to satellite test mass coupling.

The attitude of the test mass w.r.t. the sensor frame can be expressed by quaternions:

$$\dot{\mathbf{q}}_{sens}^{tm} = \frac{1}{2} \hat{\underline{\omega}}_{sens,tm}^{tm} \odot \mathbf{q}_{sens}^{tm} \quad (8)$$

4. Gravity Force and Gravity-Gradient Torque

4.1. Gravity Force

The gravity force needed in (2) and (5) to calculate the satellite and test mass motion is implemented as sum of monopolar and quadrupolar force terms. The monopolar force is the force acting on a point mass or a body with equal inertia terms in its principal axes. The quadrupolar force term corrects for deviations from this ideal model.

Considering the Earth as a spherical gravitational source, the force on a mass element of a satellite orbiting the Earth is:

$$d\underline{F} = dm \nabla \Phi = -\frac{\mu_E dm}{r^3} \underline{r}, \quad \Phi = \frac{\mu_E}{r} \quad (9)$$

where Φ is the Earth gravitational potential and $r = |\underline{r}|$, $\underline{r} = \underline{R} + \underline{\rho}$, $\underline{R} = \underline{r}_{i,b}^i$, $\underline{\rho}$ is the vector from the satellite center of mass to its mass element dm , $\mu_E = G \cdot M$, G is the gravitational constant of the Earth and M is the mass of the Earth (see also figure 2).

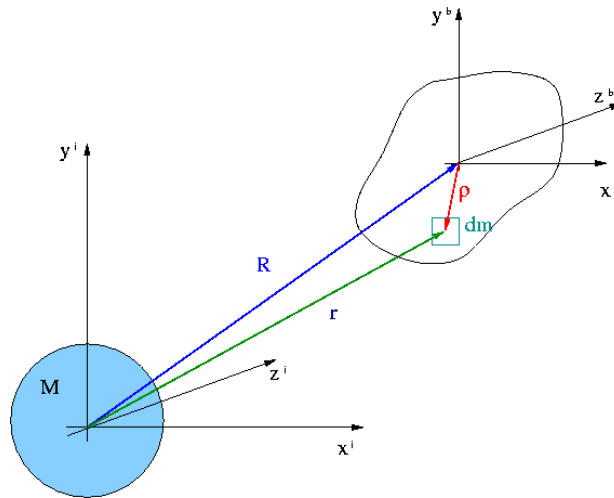



Figure 2: Reference Frames and Position Vectors for Gravity Force Determination

With $r = |\underline{R} + \underline{\rho}|$, r^{-3} can be rewritten and expanded in the following way:

$$\begin{aligned} r^{-3} &= |\underline{R} + \underline{\rho}|^{-3} = R^{-3} \left[1 + \frac{2(\underline{R} \cdot \underline{\rho})}{R^2} + \left(\frac{\underline{\rho}}{R} \right)^2 \right]^{-\frac{3}{2}} \\ &= R^{-3} \left[1 - 3 \frac{\underline{R} \cdot \underline{\rho}}{R^2} - \frac{3}{2} \left(\frac{\underline{\rho}}{R} \right)^2 + \frac{15}{2} \frac{(\underline{R} \cdot \underline{\rho})^2}{R^4} + O \left(\frac{\underline{\rho}}{R} \right)^3 \right] \end{aligned} \quad (10)$$

	<p style="text-align: center;">Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 10 of 27</p>
---	---	--

Next, $|\underline{\rho}|^2$ is expressed as

$$|\underline{\rho}|^2 = \rho^2 = x^2 + y^2 + z^2. \quad (11)$$

The inertia matrix is introduced as

$$\underline{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (12)$$

with

$$I_{xx} = \int_m (y^2 + z^2) dm, \quad I_{yy} = \int_m (x^2 + z^2) dm, \quad I_{zz} = \int_m (x^2 + y^2) dm, \quad (13)$$

$$I_{xy} = \int_m xy \, dm, \quad I_{xz} = \int_m xz \, dm, \quad I_{yz} = \int_m yz \, dm. \quad (14)$$

Substituting equation (10) into (9) and integrating equation (9) with the moments and products of inertia defined in (13) and (14), the resulting force becomes

$$\underline{F} = -\frac{\mu_E m}{R^3} \underline{R} + \underline{F}_{gg} \quad (15)$$

where \underline{F}_{gg} is the gravity-gradient or quadrupolar force term,

$$\underline{F}_{gg} = \frac{3\mu_E}{2R^5} \left[5 \frac{\underline{R}^T \underline{I} \underline{R}}{R^2} \underline{R} - tr(\underline{I}) \underline{R} - 2 \underline{I} \underline{R} \right] \quad (16)$$

where $tr(\underline{I})$ is the trace of the inertia matrix. Component wise, with $R^2 = (R_x)^2 + (R_y)^2 + (R_z)^2$, $R_x = x_{i,b}^i$, $R_y = y_{i,b}^i$, $R_z = z_{i,b}^i$, this can be written as:

$$\underline{F}_{gg} = \frac{3\mu_E}{2R^5} \left[\begin{array}{l} \left\{ \left(5 \frac{(R_x)^2}{R^2} - 3 \right) I_{xx} + \left(5 \frac{(R_y)^2}{R^2} - 1 \right) I_{yy} + \left(5 \frac{(R_z)^2}{R^2} - 1 \right) I_{zz} \right\} R_x \\ \left\{ \left(5 \frac{(R_x)^2}{R^2} - 1 \right) I_{xx} + \left(5 \frac{(R_y)^2}{R^2} - 3 \right) I_{yy} + \left(5 \frac{(R_z)^2}{R^2} - 1 \right) I_{zz} \right\} R_y \\ \left\{ \left(5 \frac{(R_x)^2}{R^2} - 1 \right) I_{xx} + \left(5 \frac{(R_y)^2}{R^2} - 1 \right) I_{yy} + \left(5 \frac{(R_z)^2}{R^2} - 3 \right) I_{zz} \right\} R_z \end{array} \right] \\ + 3 \frac{\mu_E}{R^5} \left[\begin{array}{l} R_y I_{xy} + R_z I_{xz} - 5 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_x \\ R_x I_{xy} + R_z I_{yz} - 5 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_y \\ R_x I_{xz} + R_y I_{yz} - 5 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_z \end{array} \right] \quad (17)$$

In principal axes form, the second term in (17) vanishes. It can also be seen that for equal inertia in principal axes form ($I_{xx} = I_{yy} = I_{zz} = I$, $I_{xy} = I_{xz} = I_{yz} = 0$) $\underline{F}_{gg} = 0$.

The derivation of the quadrupolar force term has been carried out in detail in [3] and [9]. For further references see also [4, 5, 10].

For most applications, the quadrupolar force term derived assuming a spherical Earth is adequate. Higher accuracy is achieved with the gravitational potential considering the Earth's nonsymmetric mass distribution written below (see also [7]):

$$\begin{aligned} \Phi = & \frac{\mu_E}{r} \left[1 + \sum_{n=2}^{\infty} \left(\frac{R_E}{r} \right)^n J_n P_{n0}(\sin\Theta) \right] \\ & + \frac{\mu_E}{r} \sum_{n=2}^{\infty} \sum_{m=1}^n \left(\frac{R_E}{r} \right)^n (C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)) P_{nm}(\sin\Theta) \end{aligned} \quad (18)$$

where:

R_E	Earth radius.
J_n	Earth zonal harmonic coefficients of degree n .
P_{nm}	Legendre polynomials of degree n and order m .
Θ	Geocentric latitude.
C_{nm}, S_{nm}	Tesseral harmonic coefficients for $n \neq m$, sectoral harmonic coefficients for $n = m$.
λ	Geocentric longitude.

In (18) the zonal harmonics J_n represent the Earth's oblateness. The tesseral and sectoral harmonics C_{nm} and S_{nm} account for longitudinal variations in the Earth's shape and become important in the case of geosynchronous spacecraft (see [10]).

The terms in (18) are at least two orders of magnitude smaller than the spherical potential defined in (9). According to [10], for attitude dynamics, the gravitational potential function including only the J_2 -term should be sufficient for computation of the gravity-gradient force terms and torques, subject to the accuracy needed. Also, inclusion of higher order terms is not reasonable due to the uncertainties inherent in other environmental disturbance forces and torques.

With

$$P_{20}(\sin\Theta) = \frac{1}{2} (1 - 3\sin^2\Theta) \quad (19)$$

and

$$\sin\Theta = \frac{r_z}{r}, \quad r_z = R_z + z, \quad (r_x = R_x + x, \quad r_y = R_y + y) \quad (20)$$

the gravitational potential function including the J_2 -term becomes:

$$\Phi = \frac{\mu_E}{r} \left[1 + \left(\frac{R_E}{r} \right)^2 J_2 \frac{1}{2} \left(1 - 3 \left(\frac{r_z}{r} \right)^2 \right) \right] \quad (21)$$

Using the gravitational potential defined in (21) the force on a mass element of a satellite orbiting the Earth is:

$$d\underline{F} = dm \nabla \Phi = -\frac{\mu_E dm}{r^3} \underline{r} - \frac{3 \mu_E R_E^2 dm}{2 r^5} J_2 \begin{bmatrix} \left(1 - 5 \left(\frac{r_z}{r}\right)^2\right) r_x \\ \left(1 - 5 \left(\frac{r_z}{r}\right)^2\right) r_y \\ \left(3 - 5 \left(\frac{r_z}{r}\right)^2\right) r_z \end{bmatrix} \quad (22)$$

This can be integrated term by term to obtain the gravity force on the satellite considering the Earth's oblateness:

$$\begin{aligned} \underline{F} &= \underline{F}_{g1} + \underline{F}_{g2} + \underline{F}_{g3} \\ &= - \int_m \frac{\mu_E}{r^3} \underline{r} dm - \int_m \frac{3 \mu_E R_E^2}{2 r^5} J_2 \begin{bmatrix} r_x \\ r_y \\ 3 \cdot r_z \end{bmatrix} dm + \int_m \frac{15 \mu_E R_E^2}{2 r^5} J_2 \left(\frac{r_z}{r}\right)^2 \underline{r} dm \end{aligned} \quad (23)$$

The integral solution for the first term \underline{F}_{g1} in (23) is presented in (15) and (16) or (17). The integrals for the second and third term are evaluated in the following.

The series expansions for r^{-5} and r^{-7} are:

$$\begin{aligned} r^{-5} &= |\underline{R} + \underline{\rho}|^{-5} = R^{-5} \left[1 + \frac{2(\underline{R} \cdot \underline{\rho})}{R^2} + \left(\frac{\rho}{R}\right)^2 \right]^{-\frac{5}{2}} \\ &= R^{-5} \left[1 - 5 \frac{\underline{R} \cdot \underline{\rho}}{R^2} - \frac{5}{2} \left(\frac{\rho}{R}\right)^2 + \frac{35}{2} \frac{(\underline{R} \cdot \underline{\rho})^2}{R^4} + O\left(\frac{\rho}{R}\right)^3 \right] \end{aligned} \quad (24)$$

$$\begin{aligned} r^{-7} &= |\underline{R} + \underline{\rho}|^{-7} = R^{-7} \left[1 + \frac{2(\underline{R} \cdot \underline{\rho})}{R^2} + \left(\frac{\rho}{R}\right)^2 \right]^{-\frac{7}{2}} \\ &= R^{-7} \left[1 - 7 \frac{\underline{R} \cdot \underline{\rho}}{R^2} - \frac{7}{2} \left(\frac{\rho}{R}\right)^2 + \frac{63}{2} \frac{(\underline{R} \cdot \underline{\rho})^2}{R^4} + O\left(\frac{\rho}{R}\right)^3 \right] \end{aligned} \quad (25)$$

Carrying out the second integral term in (23) using (24),

$$\begin{aligned} \underline{F}_{g2} &= - \int_m \frac{3 \mu_E R_E^2}{2 r^5} J_2 \begin{bmatrix} r_x \\ r_y \\ 3 \cdot r_z \end{bmatrix} dm \\ &= - \frac{3 \mu_E R_E^2}{2 R^5} J_2 \int_m \left[1 - 5 \frac{\underline{R} \cdot \underline{\rho}}{R^2} - \frac{5}{2} \left(\frac{\rho}{R}\right)^2 + \frac{35}{2} \frac{(\underline{R} \cdot \underline{\rho})^2}{R^4} \right] \begin{bmatrix} R_x + x \\ R_y + y \\ 3(R_z + z) \end{bmatrix} dm, \end{aligned} \quad (26)$$

neglecting terms of order three and higher in $\underline{\rho}$ and using

$$\int_m \underline{\rho} \, dm = 0, \quad (27)$$


the second term \underline{F}_{g2} becomes:

$$\begin{aligned} \underline{F}_{g2} = & -\frac{3 \mu_E R_E^2 J_2 m}{2 R^5} \begin{bmatrix} R_x \\ R_y \\ 3 \cdot R_z \end{bmatrix} \\ & + \frac{15 \mu_E R_E^2 J_2}{4 R^7} \begin{bmatrix} \left\{ \left(7 \frac{(R_x)^2}{R^2} - 4 \right) I_{xx} + \left(7 \frac{(R_y)^2}{R^2} - 2 \right) I_{yy} + \left(7 \frac{(R_z)^2}{R^2} - 2 \right) I_{zz} \right\} R_x \\ \left\{ \left(7 \frac{(R_x)^2}{R^2} - 2 \right) I_{xx} + \left(7 \frac{(R_y)^2}{R^2} - 4 \right) I_{yy} + \left(7 \frac{(R_z)^2}{R^2} - 2 \right) I_{zz} \right\} R_y \\ 3 \left\{ \left(7 \frac{(R_x)^2}{R^2} - 2 \right) I_{xx} + \left(7 \frac{(R_y)^2}{R^2} - 2 \right) I_{yy} + \left(7 \frac{(R_z)^2}{R^2} - 4 \right) I_{zz} \right\} R_z \end{bmatrix} \\ & + \frac{15 \mu_E R_E^2 J_2}{2 R^7} \begin{bmatrix} R_y I_{xy} + R_z I_{xz} - 7 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_x \\ R_x I_{xy} + R_z I_{yz} - 7 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_y \\ 3 \cdot (R_x I_{xz} + R_y I_{yz} - 7 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_z) \end{bmatrix} \end{aligned} \quad (28)$$

The third term, neglecting higher order terms in the integration, becomes:

$$\begin{aligned} \underline{F}_{g3} = & \frac{15 \mu_E R_E^2 J_2}{2 R^7} \left(R_z^2 m + \frac{I_{xx} + I_{yy} - I_{zz}}{2} \right) \underline{R} \\ & + \frac{15 \mu_E R_E^2 J_2}{2 R^7} R_z \begin{bmatrix} 2I_{xz} \\ 2I_{yz} \\ I_{xx} + I_{yy} - I_{zz} \end{bmatrix} \\ & - \frac{105 \mu_E R_E^2 J_2}{2 R^9} R_z \begin{bmatrix} 2R_x^2 I_{xz} + 2R_x R_y I_{yz} + R_x R_z (I_{xx} + I_{yy} - I_{zz}) \\ 2R_x R_y I_{xz} + 2R_y^2 I_{yz} + R_y R_z (I_{xx} + I_{yy} - I_{zz}) \\ 2R_x R_z I_{xz} + 2R_y R_z I_{yz} + R_z^2 (I_{xx} + I_{yy} - I_{zz}) \end{bmatrix} \\ & - \frac{105 \mu_E R_E^2 J_2}{4 R^9} R_z^2 \begin{bmatrix} \left\{ \left(9 \frac{(R_x)^2}{R^2} - 5 \right) I_{xx} + \left(9 \frac{(R_y)^2}{R^2} - 3 \right) I_{yy} + \left(9 \frac{(R_z)^2}{R^2} - 3 \right) I_{zz} \right\} R_x \\ \left\{ \left(9 \frac{(R_x)^2}{R^2} - 3 \right) I_{xx} + \left(9 \frac{(R_y)^2}{R^2} - 5 \right) I_{yy} + \left(9 \frac{(R_z)^2}{R^2} - 3 \right) I_{zz} \right\} R_y \\ \left\{ \left(9 \frac{(R_x)^2}{R^2} - 3 \right) I_{xx} + \left(9 \frac{(R_y)^2}{R^2} - 3 \right) I_{yy} + \left(9 \frac{(R_z)^2}{R^2} - 5 \right) I_{zz} \right\} R_z \end{bmatrix} \\ & - \frac{105 \mu_E R_E^2 J_2}{2 R^9} R_z^2 \begin{bmatrix} R_y I_{xy} + R_z I_{xz} - 9 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_x \\ R_x I_{xy} + R_z I_{yz} - 9 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_y \\ R_x I_{xz} + R_y I_{yz} - 9 \left\{ \frac{R_x R_y}{R^2} I_{xy} + \frac{R_x R_z}{R^2} I_{xz} + \frac{R_y R_z}{R^2} I_{yz} \right\} R_z \end{bmatrix} \end{aligned} \quad (29)$$

In the derivation of (28) and (29) the moments and products of inertia defined in (13) and (14) have been applied. Comparing equation (29) with equations (15) (\underline{F}_{gg} defined

	<p>Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 14 of 27</p>
---	---	--

in (17)) and (28), the three additional terms in (29) come from the multiplication of (25) with $r_z^2 = R_z^2 + 2R_z z + z^2$, neglecting higher order terms.

In a more compact form, the gravity force can be written as:

$$\underline{F} = \underline{F}_{mono} + \underline{F}_{gg} \quad (30)$$

with

$$\underline{F}_{mono} = -\frac{\mu_E m}{R^3} \underline{R} - \frac{3\mu_E R_E^2 m}{2R^5} J_2 \tilde{\underline{R}} + \frac{15\mu_E R_E^2 m}{2R^7} J_2 R_z^2 \underline{R} \quad (31)$$

and

$$\begin{aligned} \underline{F}_{gg} = & \frac{15\mu_E R_E^2 J_2}{2R^7} \left(\frac{I_{xx} + I_{yy} - I_{zz}}{2} \underline{R} + R_z \tilde{\underline{I}}_3 \right) \\ & + 105 \frac{\mu_E R_E^2 J_2}{R^9} R_z \left[\underline{R}^T \underline{I}_3 \underline{R} - R_z \frac{1}{2} tr(\underline{I}) \underline{R} \right] \\ & + \frac{3\mu_E}{2R^7} \underline{R}^T \underline{I} \underline{R} \left[5\underline{R} + \frac{35R_E^2 J_2}{2R^2} \tilde{\underline{R}} - \frac{315R_E^2 J_2}{2R^4} R_z^2 \underline{R} \right] \\ & - \frac{3\mu_E}{2R^5} tr(\underline{I}) \left[\underline{R} + 5 \frac{R_E^2 J_2}{R^2} \tilde{\underline{R}} - \frac{105R_E^2 J_2}{2R^4} R_z^2 \underline{R} \right] \\ & - \frac{3\mu_E}{2R^5} \underline{I} \left[2\underline{R} + 5 \frac{R_E^2 J_2}{R^2} \tilde{\underline{R}} - 35 \frac{R_E^2 J_2}{R^4} R_z^2 \underline{R} \right] \end{aligned} \quad (32)$$

where \underline{I}_3 is the third column of the inertia matrix defined in (12),

$$\underline{\underline{I}} = [\underline{I}_1 \quad \underline{I}_2 \quad \underline{I}_3], \quad \underline{I}_3 = \begin{bmatrix} -I_{xz} \\ -I_{yz} \\ I_{zz} \end{bmatrix}, \quad (33)$$

$\tilde{\underline{I}}_3$ is defined as:

$$\tilde{\underline{I}}_3 = \begin{bmatrix} 2I_{xz} \\ 2I_{yz} \\ I_{xx} + I_{yy} - I_{zz} \end{bmatrix} \quad (34)$$

and

$$\tilde{\underline{R}} = \begin{bmatrix} R_x \\ R_y \\ 3 \cdot R_z \end{bmatrix}. \quad (35)$$

As for the gravity-gradient force derived in (16), expression (32) also becomes zero for equal inertia in principal axes form.

4.2. Gravity-Gradient Torque

The gravity-gradient torque is calculated from

$$\underline{T}_{gg} = \int \underline{\rho} \times d\underline{F} \quad (36)$$

with $d\underline{F}$ defined in equation (9). Neglecting higher-order terms in $\underline{\rho}$ (see e.g. [4, 10]), the gravity-gradient torque becomes:

$$\underline{T}_{gg} = \frac{3\mu_E}{R^5} [\underline{R} \times (\underline{I} \cdot \underline{R})] \quad (37)$$

With $d\underline{F}$ defined in equation (22), the gravity-gradient torque can be written as:

$$\begin{aligned} \underline{T}_{gg} &= \underline{T}_{gg1} + \underline{T}_{gg2} + \underline{T}_{gg3} \\ &= \frac{3\mu_E}{R^5} [\underline{R} \times (\underline{I} \cdot \underline{R})] \\ &\quad - \frac{3\mu_E R_E^2}{R^5} J_2 \tilde{I}_2 + \frac{15 \mu_E R_E^2}{2 R^7} J_2 [\tilde{\underline{R}} \times (\tilde{\underline{I}} \cdot \underline{R})] \\ &\quad - \frac{15 \mu_E R_E^2}{2 R^7} J_2 R_z [\underline{R} \times \tilde{I}_3] - \frac{105 \mu_E R_E^2}{2 R^9} J_2 R_z^2 [\underline{R} \times (\underline{I} \cdot \underline{R})] \end{aligned} \quad (38)$$

where \tilde{I}_3 is defined in (34) and $\tilde{\underline{R}}$ in (35), \tilde{I}_2 is defined as

$$\tilde{I}_2 = \begin{bmatrix} I_{yz} \\ -I_{xz} \\ 0 \end{bmatrix} \quad (39)$$

and

$$\tilde{\underline{I}} = \begin{bmatrix} \frac{1}{2}(I_{xx} - I_{yy} - I_{zz}) & -I_{xy} & -I_{xz} \\ -I_{xy} & \frac{1}{2}(I_{yy} - I_{xx} - I_{zz}) & -I_{yz} \\ -I_{xz} & -I_{yz} & \frac{1}{2}(I_{zz} - I_{xx} - I_{yy}) \end{bmatrix}. \quad (40)$$

The torque expressions (37) and (38) become zero for equal inertia in principal axes form.

Alternatively, the gravity-gradient torque can be calculated using the gravity-gradient matrix, i.e. the second partial derivative of the gravitational potential (see [1, 2, 7],

$$\underline{T}_{gg} = \int \underline{\rho} \times \underline{G} \underline{\rho} dm \quad (41)$$

Hereby, higher order terms in the gravitational potential can be considered. However, the gravitational field is linearized at the center of mass of the satellite. Expanding the term $\underline{\rho} \times \underline{G} \underline{\rho}$, the gravity gradient torque can be integrated term by term which yields (see [2, 8]):

$$\underline{T}_{gg}^b = \begin{bmatrix} G_{23}(I_{zz} - I_{yy}) + G_{13}I_{xy} - G_{12}I_{xz} + I_{yz}(G_{33} - G_{22}) \\ G_{13}(I_{xx} - I_{zz}) - G_{23}I_{xy} + G_{12}I_{yz} + I_{xz}(G_{11} - G_{33}) \\ G_{12}(I_{yy} - I_{xx}) + G_{23}I_{xz} - G_{13}I_{yz} + I_{xy}(G_{22} - G_{11}) \end{bmatrix} \quad (42)$$


where

G_{ij} Element i, j of gravity-gradient matrix \underline{G} .

I_{ij} Element i, j of moments of inertia matrix \underline{I} .

The elements of the gravity-gradient matrix, $G_{ij} = \frac{\partial^2 \Phi}{\partial x_i \partial x_j}$, $i, j = 1 \dots 3$ with Φ defined in (21), are:

$$\begin{aligned} G_{11} &= -\frac{\mu_E}{R^3} \left[1 - 3 \left(\frac{R_x}{R} \right)^2 \right] \\ &\quad - \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[1 - 5 \left\{ \left(\frac{R_x}{R} \right)^2 + \left(\frac{R_z}{R} \right)^2 \right\} + 35 \left(\frac{R_x}{R} \right)^2 \left(\frac{R_z}{R} \right)^2 \right] \\ G_{22} &= -\frac{\mu_E}{R^3} \left[1 - 3 \left(\frac{R_y}{R} \right)^2 \right] \\ &\quad - \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[1 - 5 \left\{ \left(\frac{R_y}{R} \right)^2 + \left(\frac{R_z}{R} \right)^2 \right\} + 35 \left(\frac{R_y}{R} \right)^2 \left(\frac{R_z}{R} \right)^2 \right] \\ G_{33} &= -\frac{\mu_E}{R^3} \left[1 - 3 \left(\frac{R_z}{R} \right)^2 \right] - \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[3 - 30 \left(\frac{R_z}{R} \right)^2 + 35 \left(\frac{R_z}{R} \right)^4 \right] \\ G_{12} &= G_{21} = 3 \frac{\mu_E}{R^3} \left(\frac{R_x}{R} \right) \left(\frac{R_y}{R} \right) \\ &\quad + \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[5 \left(\frac{R_x}{R} \right) \left(\frac{R_y}{R} \right) - 35 \left(\frac{R_x}{R} \right) \left(\frac{R_y}{R} \right) \left(\frac{R_z}{R} \right)^2 \right] \\ G_{13} &= G_{31} = 3 \frac{\mu_E}{R^3} \left(\frac{R_x}{R} \right) \left(\frac{R_z}{R} \right) \\ &\quad + \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[15 \left(\frac{R_x}{R} \right) \left(\frac{R_z}{R} \right) - 35 \left(\frac{R_x}{R} \right) \left(\frac{R_z}{R} \right)^3 \right] \\ G_{23} &= G_{32} = 3 \frac{\mu_E}{R^3} \left(\frac{R_y}{R} \right) \left(\frac{R_z}{R} \right) \\ &\quad + \frac{3}{2} J_2 \frac{\mu_E}{R^3} \left(\frac{R_E}{R} \right)^2 \left[15 \left(\frac{R_y}{R} \right) \left(\frac{R_z}{R} \right) - 35 \left(\frac{R_y}{R} \right) \left(\frac{R_z}{R} \right)^3 \right] \end{aligned} \quad (43)$$

	<p>Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 17 of 27</p>
---	---	--

5. Validation of Numerical Simulator

5.1. Definition of Test Cases

In order to validate the correct implementation of the quadrupolar force term, two test cases are defined:

- Equal satellite inertia $I_{xx} = I_{yy} = I_{zz}$ on principal axes
- Dumbbell satellite

5.2. Equal Satellite Inertia

For identical entries in the inertia matrix in principal axes form (zero off-diagonal elements) the quadrupolar term (17) has to vanish as pointed out in section 4.1.

The moments of inertia of the satellite are set to:

$$\underline{I}_b^i = \begin{bmatrix} 20000 & 0 & 0 \\ 0 & 20000 & 0 \\ 0 & 0 & 20000 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (44)$$

The initial conditions for the satellite are set to

$$\underline{r}_{i,b}^i = \begin{bmatrix} x_{i,b}^i \\ y_{i,b}^i \\ z_{i,b}^i \end{bmatrix} = \begin{bmatrix} a \cdot (1.0 - e) \\ 0 \\ 0 \end{bmatrix} \text{ m} \quad (45)$$

$$\underline{\dot{r}}_{i,b}^i = \begin{bmatrix} 0 \\ 0 \\ \sqrt{\mu_E \cdot \left(\frac{2}{x_{i,b}^i} - \frac{1}{a} \right)} \end{bmatrix} \frac{\text{m}}{\text{s}} \quad (46)$$

where $a = (\mu_E / \omega_{orbit}^2)^{1/3}$, e is set to zero in this case, $\mu_E = 3.986004415 \cdot 10^{14}$ and $\omega_{orbit} = 1.108507726 \cdot 10^{-3}$.

This gives an orbit period of 5771s.

In figure 3 the resultant quadrupolar acceleration is shown for the satellite inertia matrix defined in (44). The values shown are below the numerical precision limit and can be regarded equal to zero as expected. Using equation (32) to obtain the gravity-gradient acceleration on the satellite, the results are zero as well. The small values are due to subtraction of equal numbers. In the case where those numbers are zero themselves, i.e. where the gravitay force component is zero, the result for the quadrupolar acceleration is also exactly zero.

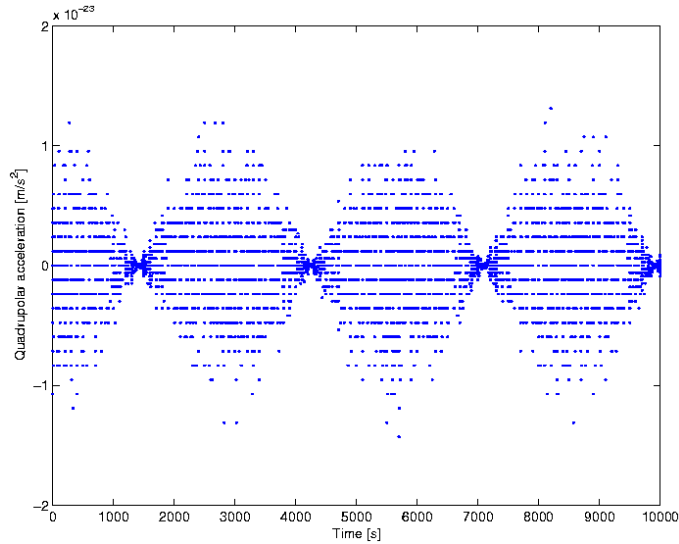


Figure 3: Quadrupolar acceleration for equal satellite inertia

5.3. Dumbbell Satellite

The simplest example for non-zero quadrupolar force terms is represented by a symmetrical dumbbell satellite, see figure 4. For this test case both masses of the dumbbell satellite have a weight of 1kg and the same diagonal inertia as tested in 5.2. The connecting rod between the two bodies is assumed to be massless.

For starting values of the simulation the initial conditions defined in 5.2, equations (45) and (46) are used. As usual they refer to the satellite center of mass, i.e. half-way between the dumbbell bodies. The initial placements of the dumbbell bodies are:

$$\underline{r}_{i,db1}^i = \begin{bmatrix} a \cdot (1.0 - e) + \rho \\ 0 \\ 0 \end{bmatrix} \quad (47)$$

$$\underline{r}_{i,db2}^i = \begin{bmatrix} a \cdot (1.0 - e) - \rho \\ 0 \\ 0 \end{bmatrix}, \quad \rho = 100\text{m} \quad (48)$$

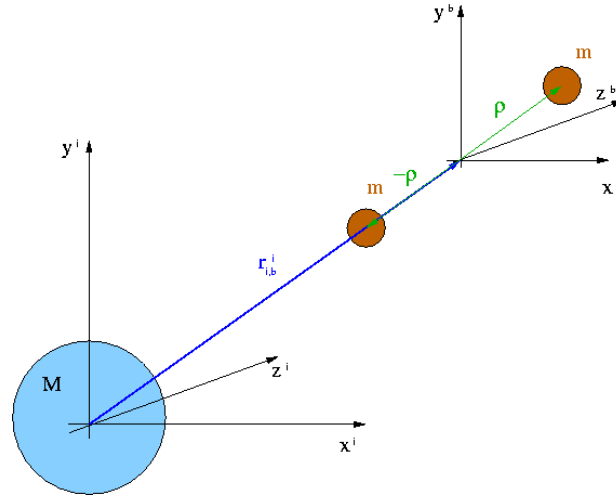


Figure 4: Reference Frames and Position Vectors for the Dumbbell Satellite

The satellite moments of inertia based on the dumbbell-body positions are:

$$\underline{I}_b^b = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 20000 & 0 \\ 0 & 0 & 20000 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (49)$$

The forces \underline{F}_1 and \underline{F}_2 acting on the dumbbell bodies are simply calculated by

$$\underline{F}_{1/2} = -\frac{\mu_E m}{|r_{i,db1/2}^i|^3} r_{i,db1/2}^i \quad (50)$$


The force \underline{F}_{mono} on the satellite center of mass (CoM) is accordingly:

$$\underline{F}_{mono} = -\frac{\mu_E \cdot 2m}{|r_{i,b}^i|^3} r_{i,b}^i \quad (51)$$

with $r_{i,b}^i$ defined in (45).

The quadrupolar force term for the satellite CoM is calculated utilizing equation (17) and $I_{xx} = I_{xy} = I_{xz} = I_{yz} = 0$, $I_{yy} = I_{zz} = I$,

$$\underline{F}_{quad} = \frac{3}{2} \frac{\mu_E}{|r_{i,b}^i|^5} \begin{bmatrix} \left[5 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 2 \right] I \cdot x_{i,b}^i \\ 0 \\ \left[5 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 4 \right] I \cdot z_{i,b}^i \end{bmatrix} \quad (52)$$

	Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 20 of 27
---	--	---

The difference between the forces on the dumbbell bodies and the satellite CoM can be calculated exactly. Since the quadrupolar force term is a measure for the deviation of the satellite from a point mass object or the ideal model presented in 5.2, the quadrupolar force term should equal the difference between the forces on the dumbbell bodies and the satellite CoM:

$$\underline{F}_{quad} \simeq \underline{F}_1 + \underline{F}_2 - \underline{F}_{mono} \quad (53)$$

Alternatively, an expression for the quadrupolar force term in this test case is computed by the gravity-gradient acceleration, i.e. the difference in acceleration between the satellite CoM and the dumbbell bodies. In the drag-free simulator the gravity-gradient acceleration is calculated directly or by utilizing the expansion of the spherical potential field described in [7]. The specific quadrupolar force term can be obtained through:

$$\underline{f}_{quad} \simeq \underline{a}_{gg,m1} + \underline{a}_{gg,m2} \quad (54)$$

where:

- \underline{f}_{quad} Specific quadrupolar force term.
- $\underline{a}_{gg,m1/2}$ Gravity gradient acceleration on dumbbell bodies 1 and 2.

If calculated directly,

$$\underline{a}_{gg,m1/2} = -\frac{\mu_E}{|\underline{r}_{i,db1/2}^i|^3} \underline{r}_{i,db1/2}^i + \frac{\mu_E}{|\underline{r}_{i,b}^i|^3} \underline{r}_{i,b}^i \quad (55)$$

the computation of \underline{f}_{quad} in (54) is identical to the approach in (53). Using the expansion described in [7] has the advantage, that a small difference obtained from the subtraction of large numbers will not be falsified through numerical precision errors.

In figure 5 the specific quadrupolar force term calculated with (52) is plotted with the cross symbol, the result obtained with (53) divided by the mass is plotted with a solid line and the sum in (54) is plotted with circles. Only the non-zero components are shown. As can be seen the results obtained with the three different methods match.

As for the previous test case, the orbit period is 5771s. Since no other forces are acting on the satellite and since it is assumed that the gravity-gradient torque is deactivated, the satellite attitude is inertially fixed. For an interpretation of the results it is easier to look at the equivalent of a satellite at fixed position which is rotating in the orbital plane. In one orbit period the quadrupolar acceleration exhibits twice the orbital frequency. The transformation from the orbital frame into the body-fixed frame yields three times the orbital rate. In figure 5 the results are plotted in the inertial frame which in this case has the same orientation as the body-fixed frame.

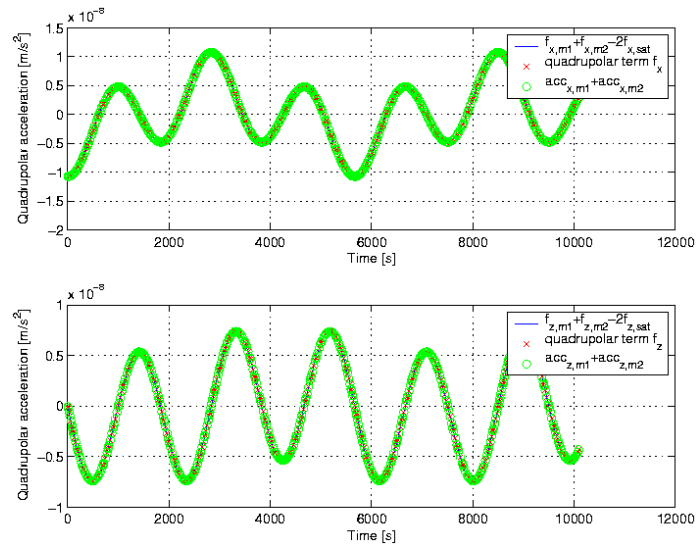


Figure 5: Quadrupolar acceleration. Vector addition of specific forces from dumbbell test masses and satellite, quadrupolar force term and sum of dumbbell accelerations.

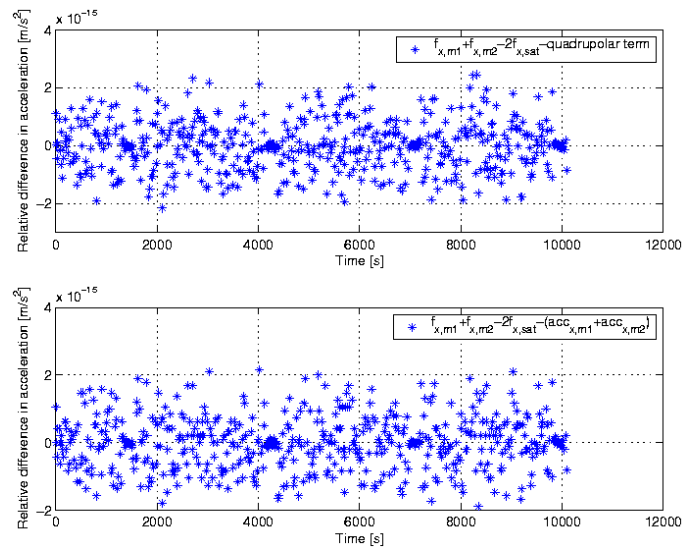


Figure 6: Top: Quadrupolar approximation error. Bottom: Error between alternate computation methods.

In figure 6 the difference between (53) and (52) (top) as well as the difference between

(53) and (54) (bottom) is shown for the x-component of the specific quadrupolar force. In both plots the difference is of order of the numerical precision limit which justifies the simplifications (neglect of higher order terms) in the derivation of the expression for the quadrupolar force term (16).

For the dumbbell satellite the gravity-gradient acceleration using equation (32) reduces to:

$$\begin{aligned}
 \mathbf{F}_{quad} = & \frac{3}{2} \frac{\mu_E}{|r_{i,b}^i|^5} \begin{bmatrix} \left[5 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 2 \right] I \cdot x_{i,b}^i \\ 0 \\ \left[5 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 4 \right] I \cdot z_{i,b}^i \end{bmatrix} + \frac{15}{4} \frac{\mu_E R_E^2 J_2}{|r_{i,b}^i|^7} \begin{bmatrix} \left[7 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 4 \right] I \cdot x_{i,b}^i \\ 0 \\ \left[21 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 18 \right] I \cdot z_{i,b}^i \end{bmatrix} \\
 & - \frac{105}{4} \frac{\mu_E R_E^2 J_2}{|r_{i,b}^i|^9} (z_{i,b}^i)^2 \begin{bmatrix} \left[9 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 6 \right] I \cdot x_{i,b}^i \\ 0 \\ \left[9 \left(\frac{z_{i,b}^i}{|r_{i,b}^i|} \right)^2 - 8 \right] I \cdot z_{i,b}^i \end{bmatrix} \quad (56)
 \end{aligned}$$

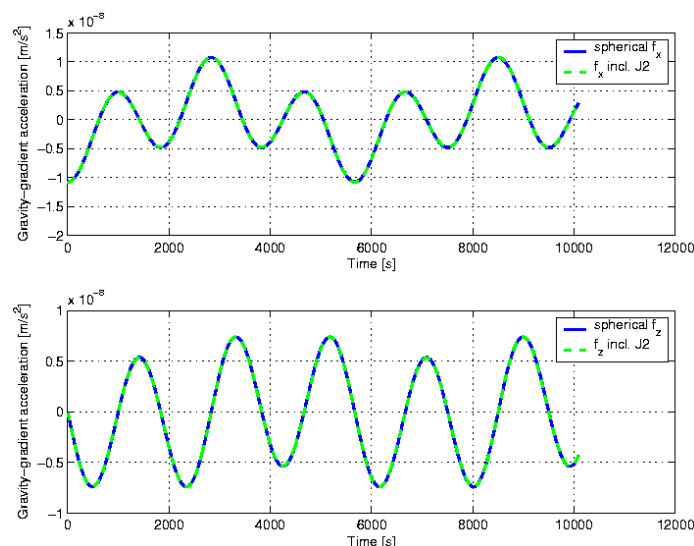


Figure 7: Gravity-gradient acceleration assuming a spherical Earth: solid line; gravity-gradient acceleration considering Earth oblateness (expressed through J_2 -term): dashed line.

In figure 7 the non-zero components of the gravity-gradient acceleration obtained from (32) are compared against the results using equation (16). The differences between the two results, i.e. the part of the acceleration that is due to the J_2 -term of the gravitational potential (21), is shown in figure 8.

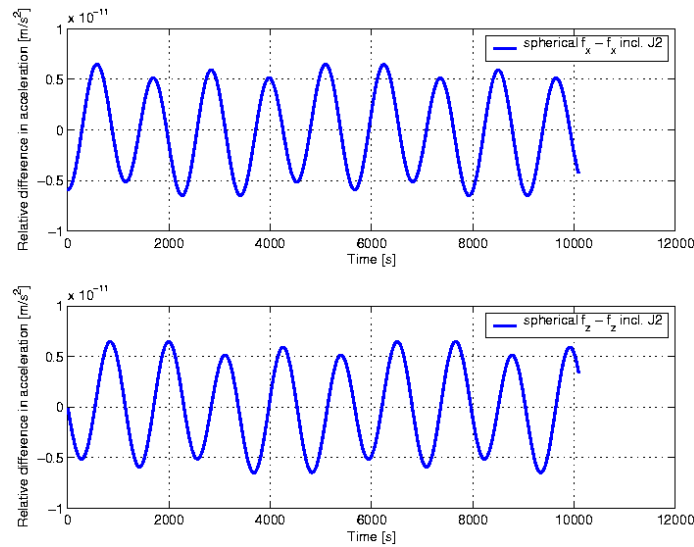


Figure 8: Gravity-gradient acceleration due to Earth oblateness.

As can be seen in figure 8, the contribution to the gravity-gradient acceleration due to the J_2 -term is three orders of magnitude smaller than the one resulting from the spherical Earth potential. This can be already inferred from the potential function in (21). The three terms in equation (21) would be of same order of magnitude if it was not for the constant $J_2 = 1.083 \cdot 10^{-3}$. For comparison, the neglected terms of order $O(\frac{\rho}{R})^3$ in the series expansions (10), (24) and (25) are six orders of magnitude smaller than the included terms of order $O(\frac{\rho}{R})^2$ and three orders of magnitude smaller than $O(\frac{\rho}{R})^2 \cdot J_2$. This estimation refers to scenarios of small spacecrafts with lengths less or equal to 10m in an altitude of 300km and more. The case where $O(\frac{\rho}{R})^3$ becomes equally important to terms of order $O(\frac{\rho}{R})^2 \cdot J_2$ starts with spacecraft dimensions 1000 times larger than the ones considered here. For the ISS with a length of 107m, the neglected terms of order $O(\frac{\rho}{R})^3$ are five orders of magnitude smaller than terms of order $O(\frac{\rho}{R})^2$ and two orders of magnitude smaller than $O(\frac{\rho}{R})^2 \cdot J_2$.

In figure 9 the gravity-gradient torque is calculated in four different ways: 1) assuming a spherical Earth according to (37), 2) including Earth's oblateness using expression (38), 3) calculation of the gravity-gradient torque with (42) assuming a spherical Earth, and 4) utilizing the gravity-gradient matrix defined in (43) to calculate the gravity-gradient

torque with (42). The magnified cut-out in the lower left of figure 9 shows the differences between the four results. The two methods involving a spherical Earth are identical and the two methods accounting for Earth's oblateness also match. In the lower right of figure 9 the J_2 -correction is plotted separately.

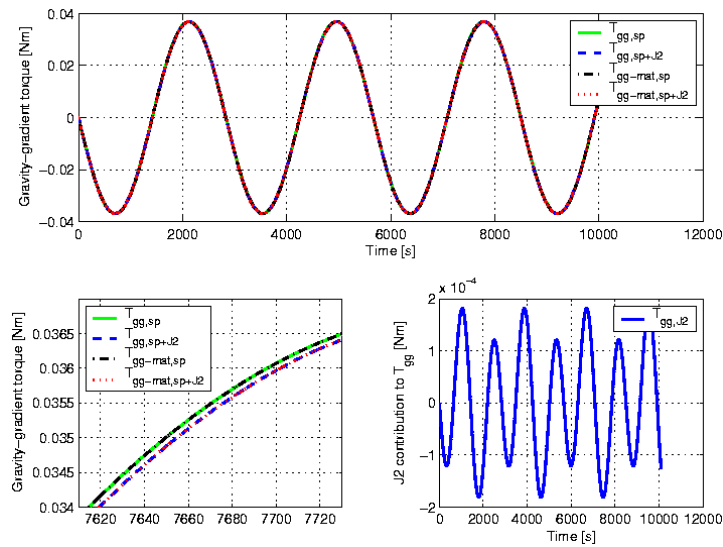



Figure 9: Top figure: Gravity-gradient torque assuming a spherical Earth, with J_2 -correction, calculated with gravity-gradient matrix (gg-mat) from spherical Earth potential and with J_2 -term; bottom left: magnified cut-out, right: J_2 -correction.

	<p>Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-002 Issue: 1.0 Page: 25 of 27</p>
---	---	--

References

- [1] David R. Glandorf. Gravity Gradient Torque for an Arbitrary Potential Function. *AIAA Journal of Guidance*, 9(1):122–124, February 1986.
- [2] Robert G. Gottlieb. Fast Gravity, Gravity Partial, Normalized Gravity, Gravity Gradient Torque and Magnetic Field: Derivation, Code and Data. NASA Contractor Report 188243, McDonnell Douglas Space Systems - Houston Division, February 1993.
- [3] M. S. Guilherme. Gravity Force in Cartesian Coordinate System. Technical Report INT-DYN-TN-ZAR-001, ZARM, University of Bremen, 2005.
- [4] Peter C. Hughes. *Spacecraft Attitude Dynamics*. John Wiley and Sons, New York, 1986.
- [5] Leonard Meirovitch. *Methods of Analytical Dynamics*. McGraw-Hill, 1970.
- [6] I. Pelivan. Equations of Motion for Satellite and Test Mass Dynamics. Technical Report FLK-SIM-TN-ZAR-001, ZARM, University of Bremen, 2005.
- [7] S. Scheithauer. Earth Gravity Model 96 – Implementation for SATSIM. Technical report, ZARM, University of Bremen, 2005.
- [8] Stephan Theil. *Satellite and Test Mass Dynamics Modeling and Observation for Drag-free Satellite Control of the STEP Mission*. PhD thesis, Department of Production Engineering, University of Bremen, December 2002.
- [9] Jozef C. van der Ha. Model of Gravity Force and Gravity Gradient Torque. Technical Report - Review of INT-DYN-TN-ZAR-001, ZARM, University of Bremen, 2005.
- [10] J.R. Wertz, editor. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978.

A. Euler Symmetric Parameters - Quaternions

A.1. Definition of Euler Symmetric Parameters

The direction cosine matrix can be parameterized in terms of Euler symmetric parameters q_1, q_2, q_3 and q_4 . They are defined as (see [10]):

$$\begin{aligned}
 q_1 &= e_x \sin \frac{\Phi}{2} \\
 q_2 &= e_y \sin \frac{\Phi}{2} \\
 q_3 &= e_z \sin \frac{\Phi}{2} \\
 q_4 &= \cos \frac{\Phi}{2}
 \end{aligned} \tag{57}$$

They can be regarded as the components of a quaternion.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{58}$$

Since a quaternion expresses a transformation the notation of super and subscripts corresponds to transformation matrices. So q_a^b denotes a quaternion describing the transformation from frame a to frame b .

A.2. Quaternion Algebra

Quaternion Multiplication

For combination of transformations quaternions can be multiplied. The multiplication is defined as:

$$\underline{q}'' = \underline{q}' \odot \underline{q} \tag{59}$$

where the operator \odot denotes the following matrix-vector operation:

$$\underline{q}'' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{60}$$

Direction Cosine Matrix from Quaternion

The direction cosine matrix can be derive from a quaternion by:

$$\underline{\underline{A}}(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2 q_1 q_2 + 2 q_3 q_4 & 2 q_1 q_3 - 2 q_2 q_4 \\ 2 q_1 q_2 - 2 q_3 q_4 & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2 q_2 q_3 + 2 q_1 q_4 \\ 2 q_1 q_3 + 2 q_2 q_4 & 2 q_2 q_3 - 2 q_1 q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (61)$$

Time-Derivative of Quaternion

If a quaternion represents the attitude of a rotating rigid body its derivative w.r.t. time is needed for propagation. For the attitude of body b w.r.t. the reference frame a the transformation can be expressed by \mathbf{q}_a^b . The angular velocity of b w.r.t. the reference a measured in the frame b is expressed as $\omega_{a,b}^b$.

The derivative w.r.t. time is:

$$\dot{\underline{\underline{q}}} = \frac{1}{2} \underline{\underline{\Omega}} \mathbf{q}_a^b \quad (62)$$

where $\underline{\underline{\Omega}}$ is defined as:

$$\underline{\underline{\Omega}} = \begin{bmatrix} 0 & \omega_{a,b_z}^b & -\omega_{a,b_y}^b & \omega_{a,b_x}^b \\ -\omega_{a,b_z}^b & 0 & \omega_{a,b_x}^b & \omega_{a,b_y}^b \\ \omega_{a,b_y}^b & -\omega_{a,b_x}^b & 0 & \omega_{a,b_z}^b \\ -\omega_{a,b_x}^b & -\omega_{a,b_y}^b & -\omega_{a,b_z}^b & 0 \end{bmatrix} \quad (63)$$

Using the quaternion multiplication rule the time derivative can be expressed as:

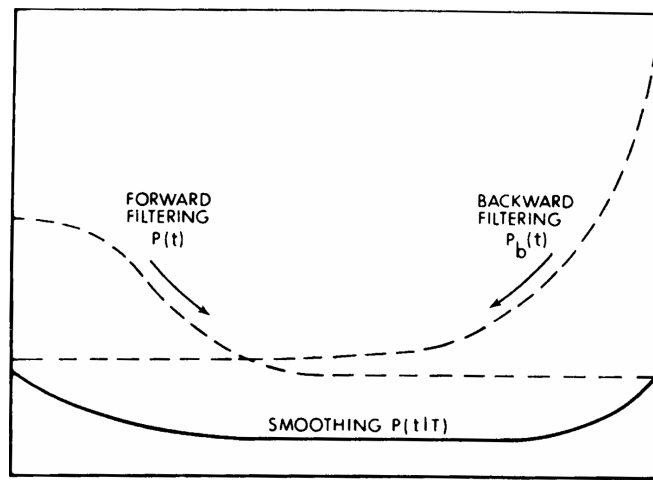
$$\dot{\underline{\underline{q}}} = \frac{1}{2} \hat{\omega}_{a,b}^b \odot \mathbf{q}_a^b \quad (64)$$

where:

$$\hat{\omega}_{a,b}^b = \begin{bmatrix} \omega_{a,b_x}^b \\ \omega_{a,b_y}^b \\ \omega_{a,b_z}^b \\ 0 \end{bmatrix} \quad (65)$$

GAIA

On-Ground Attitude Determination



Project: <div style="text-align: center; font-weight: bold; font-size: 1.2em;">FIRST LOOK</div>	Document No.: <div style="text-align: center; font-weight: bold; font-size: 1.2em;">FLK-SIM-TN-ZAR-005</div>									
<p style="text-align: center;">ZARM –Center of Applied Space Technology and Microgravity</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 33%;">Am Fallturn</td> <td style="width: 33%;">Phone:</td> <td style="width: 34%;">+49-421-218-9422</td> </tr> <tr> <td>D28359 Bremen</td> <td>Fax:</td> <td>+49-421-218-4356</td> </tr> <tr> <td>Germany</td> <td>E-Mail:</td> <td>jvdha@zarm.uni-bremen.de</td> </tr> </table>		Am Fallturn	Phone:	+49-421-218-9422	D28359 Bremen	Fax:	+49-421-218-4356	Germany	E-Mail:	jvdha@zarm.uni-bremen.de
Am Fallturn	Phone:	+49-421-218-9422								
D28359 Bremen	Fax:	+49-421-218-4356								
Germany	E-Mail:	jvdha@zarm.uni-bremen.de								

Doc. No.: FLK-SIM-ZAR-005


Issue: 1.0

Written: Jozef van der Ha Date: April 13, 2006

Approved: – Date: –

TABLE of CONTENTS

1.	INTRODUCTION.....	3
2.	GAIA FIRST LOOK OVERVIEW.....	4
2.1.	SUMMARY.....	4
2.2.	FIRST-LOOK ACTIVITIES.....	4
2.3.	BASIC GAIA DATA REDUCTION.....	5
2.4.	FIRST LOOK PREPROCESSING.....	5
2.5.	RELEVANT PAYLOAD CHARACTERISTICS.....	6
3.	GAIA ON-BOARD ATTITUDE DETERMINATION AND CONTROL.....	8
3.1.	MEASUREMENT & POINTING REQUIREMENTS.....	8
3.2.	JITTER REQUIREMENTS.....	10
3.3.	SUMMARY OF PERFORMANCES.....	10
4.	ATTITUDE DATA SIMULATION.....	12
4.1.	SIMULATOR CHARACTERISTICS.....	12
4.2.	SCANNING LAW.....	13
4.3.	ATTITUDE DETERMINATION.....	15
4.3.1.	UNDERSTANDING OF ATTITUDE NOISE.....	15
4.3.2.	ATTITUDE NOISE MODEL.....	17
4.3.3.	ON-BOARD ATTITUDE DETERMINATION AND CONTROL.....	18
4.3.4.	DISTURBANCE TORQUES.....	20
4.3.5.	BASIC ANGLE VARIATIONS.....	23
5.	SMOOTHING ESTIMATION PROCEDURE.....	24
5.1.	INTRODUCTION.....	24
5.2.	OPTIMAL SMOOTHER ALGORITHM.....	24
	REFERENCES.....	27

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 3 of 27
---	------------------------------------	---

1. Introduction

This paper provides a preliminary outline of the concept for the First-Look On-Ground Attitude Determination (OGAI) of the GAIA satellite.

The OGAI covers the very first step in the non-real-time on-ground First-Look scientific data processing of the GAIA satellite. OGAI will be a so-called smoothing type of estimation algorithm and will employ a relatively large set of data (e.g., one or more great circles) in its entirety.


This processing will likely result in much smoother and more consistent estimation errors than the on-board 'Real-Time Attitude Determination' algorithm which is based on only the past a posteriori states and has a somewhat different objective, i.e. stability and robustness instead of attitude accuracy and consistency.

The improvement factor will be somewhere between perhaps 2 and at most 5 to 10 (after some enhanced attitude modeling). However, without actual simulations and without detailed understanding of the on-board estimation algorithm and its performance, this 'factor' will remain essentially a 'guess'.

In any case, even when the average improvement factor may be relatively limited, the attitude errors will have fewer 'ups and downs' and the errors will be more representative as well as smaller.

Furthermore, specific effects that turn out to be of some significance when the data are analysed, could be modeled with reasonable accuracy on-ground, thereby further enhancing the attitude accuracy as well as its consistency.

Another important argument is that the on-ground attitude reconstitution (because of its more consistent error characteristics) would result in more meaningful residuals, i.e. the differences between the actual and the predicted measurements (based on the established best attitude estimates). These residuals are useful because they provide good insight into the characteristics of the attitude errors and will indicate when specific problems occur. Also they give insights into the actual performance of the on-board attitude sensors and any degradations (even if only very gradual) in these performances.

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 4 of 27
---	------------------------------------	---

2. GAIA FIRST LOOK OVERVIEW


The paper by Jordan et al in Ref. [4] provides a great deal of useful insights into the GAIA First Look concept.

2.1. Summary

1. Complicated and ambitious space mission like GAIA needs careful monitoring of all components of the satellite at different time scales, by different methods, and on different levels of precision
2. A first analysis of the science data quality and consistency is done by the “Science Quick Look”
3. Due to nominal scanning law, a full self-consistent calibration of the satellite and a determination of the astrometric and global parameters is not possible before about half a year into the mission
4. The latter implies a serious danger of loss of valuable observing time in case something is wrong
5. Thus, it is necessary to perform a “Detailed First Look” on a daily basis at micro-arcsec level
6. Two methods are proposed: a Block Iterative Solution (BIS?) procedure and a direct solution for monitoring all satellite parameters that in principle can be evaluated in a short amount of time.

2.2. First-Look activities

1. The First-Look aims at a rapid health monitoring at the target level of astrometric precision by means of the Global Iterative Solution (GIS) after several months of data have been gathered
2. The Quick-Look (QL) task comprises all operations activities at satellite level
3. The Science Quick-Look (ScQL) concerns scientific data health and will be some less precise and simplified version of the Detailed First Look task and works mainly at the data level (e.g. missing blocks); this will most efficiently be done at the same location as QL
4. The Detailed First Look is the in-depth scientific assessment of the quality of GAIA data within about 24 hours after reception at the Data and Processing centre (DPC)
5. It involves a restricted astrometric calibration to judge the measurement quality but will not necessarily provide absolute astrometric and calibration parameters. This task

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 5 of 27
---	------------------------------------	---

is called First-Look Processing (FLP) and is the most complicated process in the chain. FLP may be compared with the strictly one-dimensional Great-Circle Reduction of the HIPPARCOS mission

6. After FLP has been performed, the Detailed First-Look (DFL) task will produce diagnostics of the status of satellite and instrument in a more sophisticated manner than done by ScQL.

2.3. Basic GAIA Data Reduction

1. With known (even approximately) geometric calibration of the telescopes and focal plane, the pixel coordinates of the centroid can be transformed into spherical longitude and latitude coordinates, i.e. ‘the observed field angles’: η (along scan) and ζ (across scan); for a given instant of time, they depend on the astrometric parameters of the source and on GAIA’s attitude with respect to inertial space
2. In the Gaia Global Solution (GIS) the differences between observed field angles and the ones computed from approximate parameters are used to determine corrections in a linearized least-squares adjustment process; this is basic principle of the self-calibrating astrometric reduction of GAIA’s measurements.

2.4. First Look Preprocessing

1. With known (even approximately) geometric calibration of the telescopes and focal plane, the pixel coordinates of the centroid can be transformed into spherical longitude and latitude coordinates, i.e. ‘the observed field angles’: η (along scan) and ζ (across scan); for a given instant of time, they depend on the astrometric parameters of the source and on GAIA’s attitude with respect to inertial space
2. The main instrument measures only along scan (i.e., η only), whereas the Astrometric Sky Mapper (ASM) measures in two dimensions, but with a much lower precision across scan (1 to 3 mas, but only after calibration)
3. A HIPPARCOS-style great-circle reduction is not able to perform the First Look job for GAIA because the across-scan positions of the stars are not known with sufficient precision until after the first GIS has been completed. The great circle reduction may work if the ASM were calibrated.

More background and detailed information on the First-Look processing can be found in Jordan [4] and in other GAIA papers.

2.5. Relevant Payload Characteristics

The SRD, Ref. [2] pp. 20-23, provides details on the GAIA Payload module design. The payload contains two physically distinct instruments, i.e. Astro-1 and -2, with a common focal plane but with different viewing directions or Line-of-Sight (LoS), see Figure 2.1.

The angle between the two LoS's is known as the 'basic angle' with a value of 99.4 +/- 0.5 degrees (still to be selected). The stability of the basic angle is essential for the GAIA mission objectives. Therefore, the payload is equipped with a dedicated Basic Angle Monitoring (BAM) device for assessing the imposed stability conformance of the basic angle.

Both of the focal plane instruments are equipped with the following devices using CCD's for measuring the photon counts of the objects transiting over the focal plane:

- dedicated Astrometric Sky Mappers (ASM's), i.e. ASM-1 and -2, for object detection
- Astrometric Field (AF) for performing GAIA's principal astrometry mission
- Broad Band Photometer (BBP) for collecting color information of all observed objects

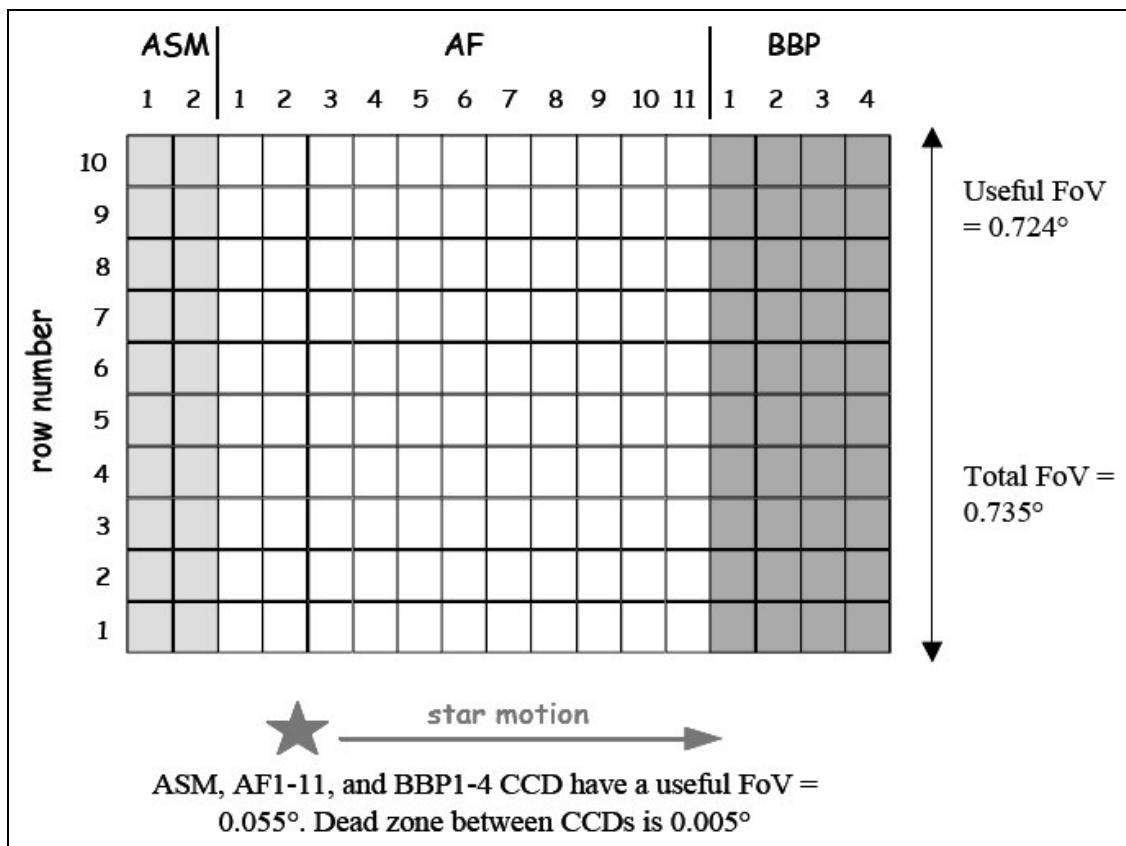




Figure 2.1 – Common Focal Plane of the Astro-1 and -2 Instruments

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 7 of 27
---	------------------------------------	---

It should be noted that ASM-1 is seen by telescope # 1 and ASM-2 by telescope # 2. The Sky Mapper measurements consist of the determination of the centroid crossing times of the stars over each of the ASM CCD arrays. These crossing times are combined with the subsequent crossing times of the same stars over the AF-1 CCD array, which is about 10 and 5 seconds for stars in FOV # 1 and FOV # 2, respectively. Finally, these measurements can be processed to estimate the speeds of the stars, both in the along and the across directions, relative the instrument focal plane.

	<p style="text-align: center;">GAIA Ground Attitude Determination</p>	<p>Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 8 of 27</p>
---	---	--

3. GAIA On-Board Attitude Determination and Control

3.1. Measurement & Pointing Requirements

The normal-mode on-board attitude determination and control of the GAIA satellite is described in some detail in Refs. [1] and [2] and in even more detail in Ref. [3].

The most recent determination and control performance requirements are summarized in ESA's GAIA Mission Requirements Document Ref. [1], section 4.7 on p. 31.

Table 1 summarizes the principal mission- and system-level requirements imposed on the attitude pointing and rate performances as provided in Ref. [1], p. 31. These requirements are applicable to both ASTRO telescope Line-of-Sight (LoS). The GAIA astrometric measurement principle and CCD operation makes that a distinction must be made in the accuracy requirements for quantities measured in the along-scan and those along the across-scan directions.


The following definitions may be recalled:

- **Attitude Measurement Error (AME)** refers to the instantaneous 3-axis angular separation between the estimated satellite attitude (as well as Astro instrument LoS) and the actual one
- **Rate Measurement Error (RME)** refers to the mean difference between the estimated spacecraft scan rate and the actual one
- **Absolute Pointing Error (APE)** refers to the instantaneous 3-axis angular separation between the desired satellite attitude (as well as Astro instrument LoS) and the actual one
- **Relative Pointing Error (RPE)** at a given time t is defined as the standard deviation of the absolute pointing error over the AF CCD integration time τ around t
- **Mean Error Rate (MRE)** is defined as the mean difference between the desired satellite scan rate attitude and the actual one; the averaging time is the time spent by an object from being detected to being confirmed in the astrometric field

Parameter	Requirement (99.73 % probability level, science operations)
AME - Attitude Measurement Error	< 20 arcsec
RME1 - Rate Measurement Error (along scan)	< 0.9 mas/s
RME2 - Rate Measurement Error (across scan)	< 2.7 mas/s
APE - Absolute Pointing Error	< 60 arcsec
RPE1 - Relative Pointing Error (along scan)*	< 5 mas (on along-scan axis, all time)
RPE2 - Relative Pointing Error (across scan)*	< 10 mas (on any across-scan axis, all time)
MRE1 - Mean Rate Error (along scan)	< 2 mas/s (on along-scan axis, all time)
MRE2 - Mean Rate Error (across scan)	< 10 mas/s (on any across-scan axis, all time)

* standard deviation of APE during single detector CCD integration time of 3.3 sec

Table 1 – Summary of AOCS Performance Requirements

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 10 of 27
---	------------------------------------	--

3.2. Jitter Requirements

In addition, there are limiting requirements on the levels of permissible jitter in the along-scan and across-scan directions of the ASTRO telescope line of sight, namely the requirements numbered SCI-740 and 750 given on p. 33 of Ref. [1]:

- $\Delta\vartheta < 3.4$ micro-arcsec rms (along-scan)
- $\Delta\vartheta < 100$ micro-arcsec rms (across-scan)


In both cases, the requirements refer to the angle $\Delta\vartheta$ that is defined in terms of the jitter's Power Spectral Density (PSD) integrated over the frequency interval above $1/(10T)$ with T the time for an object to cross the ASTRO focal plane, i.e. T is about 1 minute.

It is of interest to point out that more accurate attitude knowledge than specified in Table 1 will be achieved during nominal payload operational mode. This is because the more accurate attitude knowledge (in particular, along-scan and across scan rates) that can be derived from the CCD outputs is provided to the AOCS subsystem and is progressively 'merged' with available the star tracker attitude.

3.3. Summary of Performances

Ref. [3], section 3.5.5 contains the results of the AOCS performances produced during the detailed simulations. The following conditions and assumptions were taken during the simulations:

- attitude estimation is based on three spacecraft rotation rates (delivered by the payload measurements) and three attitude angles derived from the star tracker angular outputs and the integrated payload rates
- the star tracker random white noise was taken as 5 and 25 arcsec (rms, 5 Hz) for the cross and line-of-sight axes, respectively, and similar values have been taken for the random bias error
- the along-scan and across-scan rates are determined from the observed star scans over the ASM1 and AF1 CCD's for telescope 1 and the scans over the ASM2 and AF2 CCD's in the ASTRO focal plane for telescope 2; random white noise was taken as 4 mas/sec and 20 mas/sec for the cross and line-of-sight axes, respectively
- the attitude controller uses a straightforward Proportional-Derivative (PD) control law with a control bandwidth of 0.005 Hz and damping of 0.7. The attitude angle and rate errors are filtered using a low-pass filter with a cut-off frequency at 5 times control bandwidth
- the Field Emission Electric Propulsion (FEEP) actuation characteristics are taken as 1.2 mN max thrust level, 1 μ N quantification, and 0.1 % random white noise

	<p>GAIA Ground Attitude Determination</p>	<p>Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 11 of 27</p>
---	---	---

- the satellite dynamics includes a second-order rigid-body part and a first mode (Sun shield and booms) representation of 0.1 Hz cantilever frequency on cross axes; inertias are taken as 2160 kgm² for the cross axes and 2725 kgm² for the spin axis
- the solar radiation disturbance torques are modeled by a combination of a constant mean solar flux value plus integrated white noise with standard deviation of 4.5×10^{-6} times the mean value and 5 minute oscillation periods.

The resulting performances shown on pp. 45 - 47 of Ref. [3] indicate that the requirements of Table 1 are (essentially) fulfilled.

4. Attitude Data Simulation

4.1. Simulator Characteristics

Figure 4.1 illustrates the data interfaces and data processing required for the objective of assessing the performance of the potential on-ground attitude determination algorithms.

The left-hand-side of Figure 4.1 summarizes the various functions contained in the simulator. These are the same elements that are on-board the spacecraft with the exception that most of the elements do not need to be ‘high-fidelity’ in terms of their similarity with the on-board elements. In other words, most of these elements can be designed in a somewhat representative manner rather than in a fully identical to the way they are used on-board. The various elements of the simulator and their proposed simplifications are addressed in the following sections.

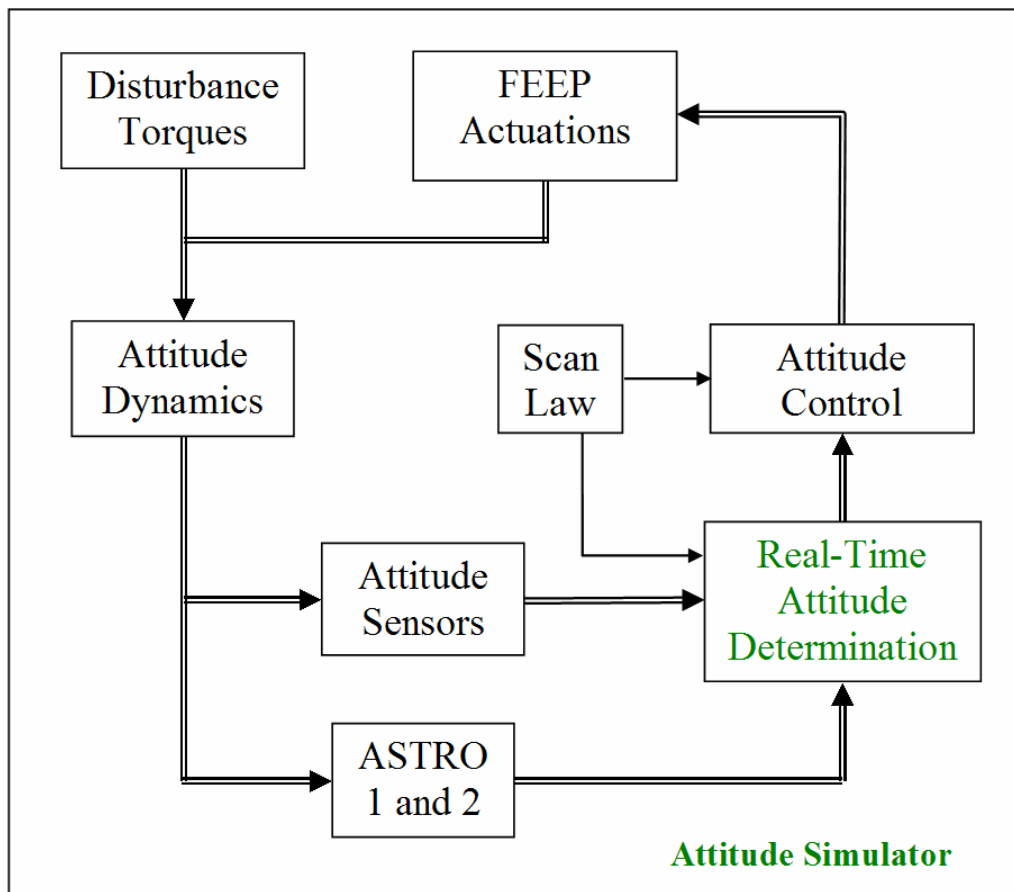


Figure 4.1 - Summary of Elements in Attitude Simulator

4.2. Scanning Law

The ‘Nominal Scanning Law’ or NSL used on-board the GAIA satellite defines the reference pointing attitude directions for the two astrometric fields of view on the sky. In reality, the attitude motion will not be exactly along the directions prescribed by the NSL but will have error angles of up to 20 arc-sec in all three directions (TBD).

The NSL prescribes a constant scan rate of 1 arcmin per second which implies that a full great circle is described in 6 hours. The scan axis is subjected to a slow precession at a rate (mean value over the year) of 0.173 arcsec/sec about the Sun to Earth direction. Furthermore, the angle between the scan axis and the Sun must be maintained at 45 degrees. More specific requirements on the NSL motion can be found in Ref. [1], pp. 14-15 and in more detail in the SRD Ref. [2], pp. 29-30

For the objectives of the present simulator it is perhaps not necessary (at least, in the first stage) to model the NSL in full detail but, on the other hand, a correct NSL modeling would lead to a realistic attitude framework for the simulations. Furthermore, the NSL model is (in principle, at least) relatively easy to implement in software. This is certainly the case when using the model given by Lindegren [5] who uses a formulation in terms of the 4-dimensional quaternion \mathbf{q} :

$$\mathbf{q} = [e_1 \sin(\Phi/2), e_2 \sin(\Phi/2), e_3 \sin(\Phi/2), \cos(\Phi/2)]^T \quad (4.1)$$

with $\mathbf{e} = (e_1, e_2, e_3)^T$ the unit vector along the instantaneous rotation vector and Φ the relevant rotation angle. Multiplication of two quaternions $\mathbf{a} = (a_1, a_2, a_3)^T$ and $\mathbf{b} = (b_1, b_2, b_3)^T$ is straightforward (see [5], eq. 4):

$$\mathbf{c} = \mathbf{ab} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} b_4 & b_3 & -b_2 & b_1 \\ -b_3 & b_4 & b_1 & b_2 \\ b_2 & -b_1 & b_4 & b_3 \\ -b_3 & -b_2 & -b_3 & b_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (4.2)$$

The nominal scanning law expresses the NSL reference axes, which describe the nominal evolution of the satellite attitude pointing over the sky, relative to the inertial J2000 frame. The NSL may be defined by 5 successive quaternion rotations, as shown by Lindegren in Ref. [5], eq. (6).

Unfortunately, the NSL definition in [5] is not complete because the functions $\lambda_s(t)$, $\nu(t)$, and $\Omega(t)$ have not been spelled out in any specific detail. In the ESA requirements, only the scan and precession rates are prescribed so that it is not possible (or at least not easy)

to reconcile and compare the Lindegren model with the ESA requirements. It is recommended to use a model similar to what Lindegren describes and to select the open functions in such a way that the resulting scan motion satisfies the ESA requirements.

Two later papers by Mignard [6] and [7] provide the missing details to the Lindegren model in terms of the definition of the revolving phase angle η on the Sun-centered cone.

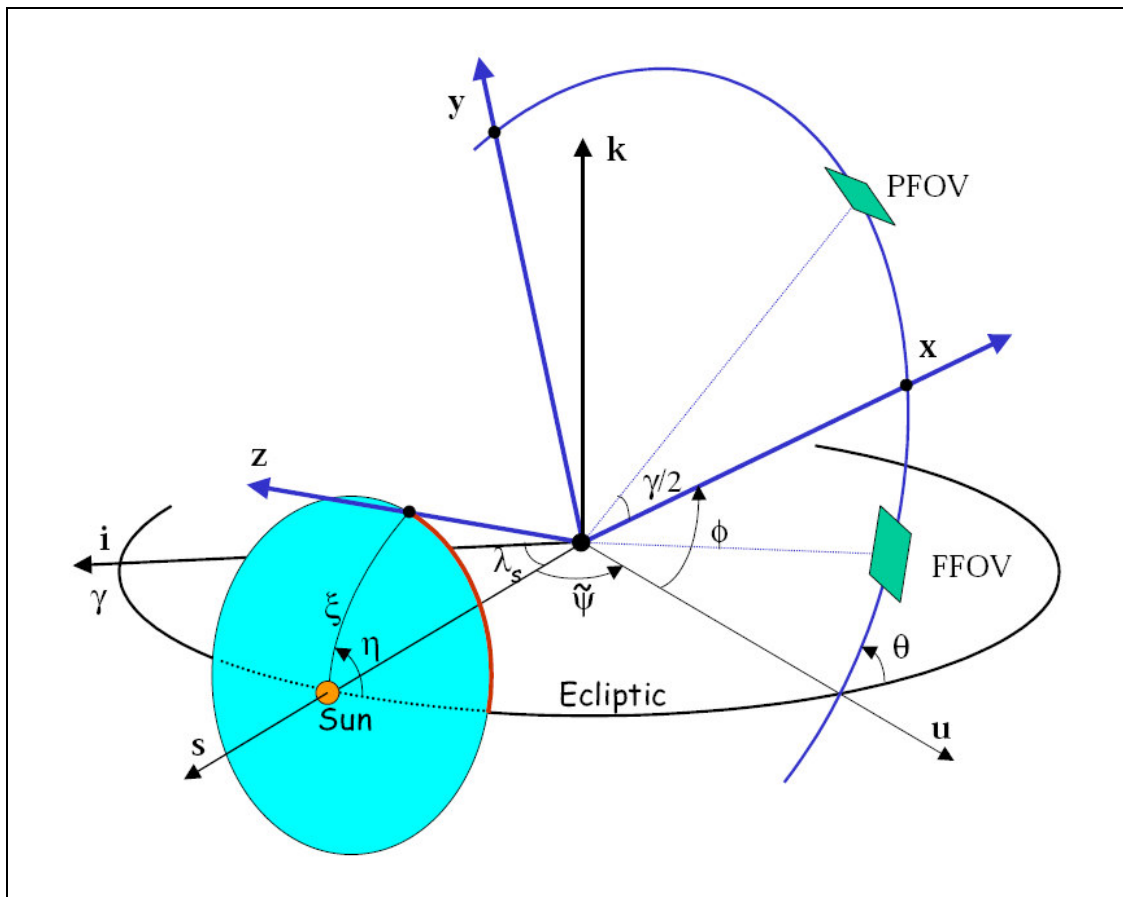



Figure 4.2 – Visualization of Nominal Scanning Law angles ξ and η [6]

Another important issue that must be kept in mind within the context of the scanning law definition is that astronomers and engineers use different reference frames as pointed out by Bastian [8]. The solution he proposes and which has been accepted is to maintain each of the two conventions for the different communities and to be aware of the rotational transformation between the two frames, as shown in [6], Figure 3 on p. 5, which is identical to Figure 4.3.1-1 on p. 29 of ESA's SRD in Ref. [2].

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 15 of 27
---	------------------------------------	--

Finally, it should be mentioned that no evidence has been found that the above reports on the NSL definition are in complete agreement with the ESA NSL requirements given in the MRD [1], pp. 14-15 and in the SRD [2], pp. 29-30, but this should in fact be the case when assuming that the scanning law design has been adequately coordinated.

4.3. Attitude Determination

4.3.1. Understanding of Attitude Noise

The On-Ground Attitude Determination (OGAD) consists of a robust, weighted least-squares fitting method of cubic spline functions to positional observations (i.e., centroid location in the CCD data stream) collected in ASM and AF1-11, as described by Lindegren [10]. The spline functions describe the four components of the attitude quaternions as smooth functions of time.

There are three fundamental time intervals that are relevant for defining and assessing the attitude determination error:

1. The CCD integration time $\tau \approx 3.3$ sec, with corresponding frequency $f_\tau \approx 0.3$ Hz
2. The FoV transit time $T \approx 40$ sec with corresponding frequency $f_T \approx 0.025$ Hz
3. The spline knot interval $\Delta t \approx 100$ sec

The ‘spline separation’ frequency f_s is defined as $f_s \approx 1/(2\Delta t) \approx 0.005$ Hz. It is evident that attitude noise above the frequency f_s can *not* be modeled by the spline function over the interval in question. This type of errors is therefore transmitted without attenuation into the OGAD results. Lower frequency noise, on the other hand, is effectively absorbed by the spline and is thus strongly attenuated.

It is important to note that the separation frequency is higher than the attitude control bandwidth, which is of the order of 0.001 Hz). This means that attitude errors induced by the star tracker and the real-time rate measurements are perfectly absorbed by the spline and do not contribute to the OGAD error (this is almost certainly true for $\Delta t < 100$ sec).

Attitude errors may be correlated over time scales comparable with a FOV transit (i.e., about 40 seconds), it is reasonable to assume that the OGAD errors on successive FOV transits of a given source are uncorrelated. Therefore, the final achievable mean attitude error σ_{att} can be attenuated by the mean number N of FOV transits: $\sigma_{att} = \sigma/\sqrt{N}$ where σ stands for the mean attitude error of an individual measurement and $N \approx 83$.

The individual attitude error σ is not derived from an instantaneous measurement but should be understood as a mean error over the ‘sampling interval’ τ which equals the CCD integration time of $\tau = 3.3$ seconds for the ASM and AF1 readings. When denoting the along-scan attitude, i.e. the attitude angular depointing with respect to the nominal

scanning law, as a function of time by $a(t)$, we can express the actual attitude measurement as a moving average over the interval τ :

$$\bar{a}(t) = (1/\tau) \int_{t-\tau/2}^{t+\tau/2} a(t) dt \quad (4.3)$$

The OGAD error $e(t)$ is now the difference between the actual attitude $a(t)$ and the spline function $s(t)$ over the time interval $T = 40$ sec corresponding to the FOV crossing. The variance of the attitude error over this interval is now:

$$\sigma_{\text{FoV}} = \langle \bar{e}_T^2(t) \rangle^{1/2} \quad \text{with:} \quad \bar{e}_T(t) = (1/T) \int_{t-T/2}^{t+T/2} \{a(t) - s(t)\} dt \quad (4.4)$$

The brackets $\langle x^2 \rangle$ stands for the statistical average, i.e. $\langle x^2 \rangle = E\{x^2\}$.

On the basis of the noise characteristics presented above we may split the OGAD error $e(t)$ in two components, namely:

$$e(t) = o(t) + m(t) \quad (4.5)$$


Here, $o(t)$ the low-frequency part dominated by the measurement noise in the AF centroiding data, and $m(t)$ the high-frequency noise consisting of the modeling errors.

The Power Spectral Density (PSD) of $e(t)$ may now be expressed as:

$$P_e(f) = P_o(f) + P_m(f) \quad (4.6)$$

It is shown by Lindegren [10], Figures 1 and 2 that the cubic spline effectively filters all low-frequency noise with $f < 0.2/\Delta t$, whereas the noise above $f > 0.7/\Delta t$ is not damped. Therefore, there is a relatively sharp boundary between these two contributions. As shown in [10], section 3.2, the noise covariances may be added as well:

$$\sigma_{\text{FoV}}^2 = \sigma_o^2 + \sigma_m^2 \quad (4.7)$$

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 17 of 27
---	------------------------------------	--

4.3.2. Attitude Noise Model

The error in the results of the GAIA attitude determination (for both the on-board RTAD and the on-ground OGAD) depends mainly on the noise in the readings of the sensors, i.e. the star tracker, the gyroscopes, as well as that of the CCD output samples that are employed for the determination of the rate measurements.

Lindegren [9] and [10] provides a rough but illustrative model for establishing the attitude determination error over different frequency intervals on the basis of the photon noise errors and the number of stars as a function of magnitude. Also the contribution of each noise frequency interval to the final error budget is determined.

His approach goes as follows:

1. An elementary measurement consists of the mean star location in a single CCD crossing lasting 0.86 seconds. Thus, all attitude variations with frequencies above about 1.2 Hz will be smoothed out. Their contributions to the error budget are accounted for by the Point Spread Function (PSF) smearing.
2. It can be shown that the contributions of the frequency intervals to the final attitude error budget will be negligible below a certain frequency f_0 . This defines the highest frequency at which the attitude determination will be useful and leads to the requirement that the actual attitude noise *above* that frequency is negligible.
3. If the determination of attitude errors is considered within a small frequency bandwidth Δf , an average of about 10 observations per cycle are required (because of the uneven star distributions) to fit harmonic functions of the scan angle in the course of a great circle scan. Lindegren [9] shows that the PSD of the final attitude determination noise is only weakly dependent on frequency for $f < 1$ Hz.
4. Let $P(f) = P$ (i.e., a constant) denote the PSD of the attitude estimation error in the data processing and let $P_0(f) = A f^{-\alpha}$ designate the PSD function of the actual attitude variations induced by the FEOP control actuators. The two PSD's cross over at $f_0 = (A/P)^{1/\alpha}$. Thus, the attitude must be estimated only up to frequencies f_0 since the actual variations are smaller than the estimation errors for higher frequencies.

The total rms attitude error follows as:

$$\sigma_{att}^2 = \int_0^{f_0} P(f)df + \int_{f_0}^{\infty} P_0(f)df = \alpha P f_0 / (\alpha - 1) \quad (4.3)$$

In the case when $\alpha = 2$, we find $\sigma_{att}^2 = 2P f_0$. If we now require that the attitude error should be smaller than 10 μ -arcsec and use a PSD value of $P = 1000$ (μ -arcsec)²/Hz, we find from the cross-over condition that $f_0 \approx 0.05$ Hz, see Figure 4.3. From the cross-over condition we can see that the value of $A = P(f_0)^\alpha$ corresponds to about 2.5 (μ -arcsec)²Hz.

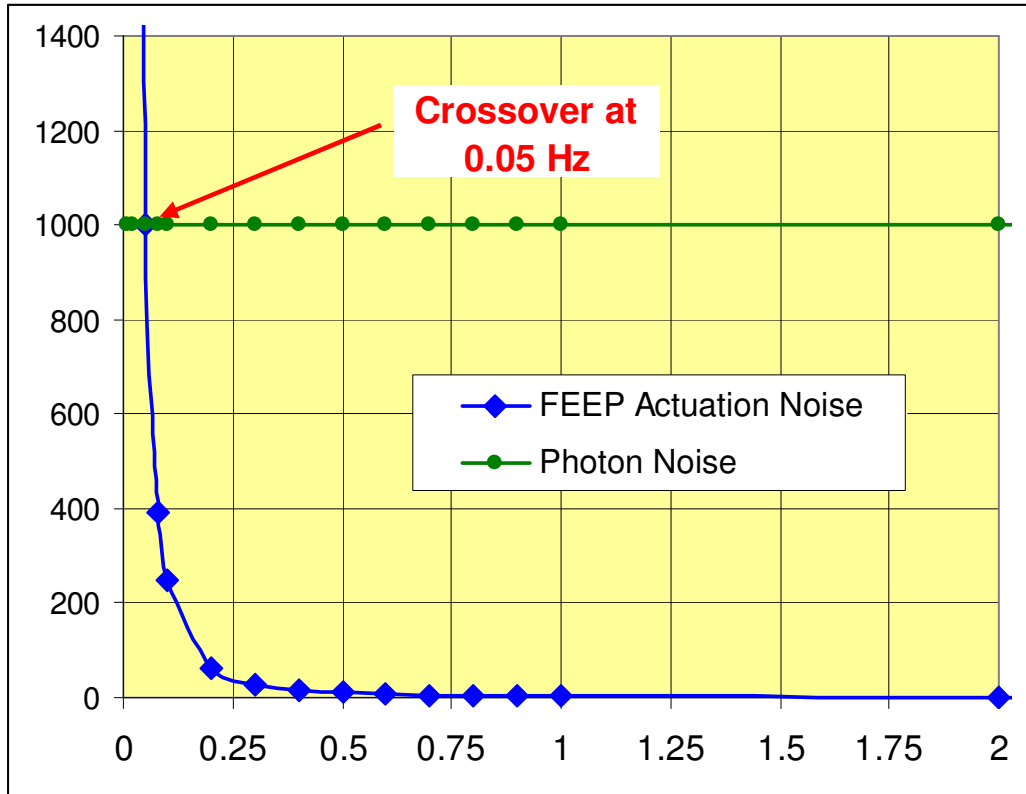


Figure 4.3 – Visualization of Attitude Noise Errors

4.3.3. On-board Attitude Determination and Control

The on-board Real-Time Attitude Determination (i.e., RTAD in HIPPARCOS jargon) and control concept is described in some (preliminary) detail in ESA-SCI(2000)4, Ref. [4], Chapter 4, and in particular pp. 201-208. More recent information can be found in Ref. [3], section 3.5, pp. 34-47.

Of relevance to the present study is only the ‘operational mode’, where the attitude sensing is performed by the star sensor and sky mappers, and attitude control by the FEEP thrusters. Table 4.1 provides a summary of their characteristics which has been collected from data presented in [3], p. 43.

<u>S</u> <u>A</u> Sensor / Actuator	Device	Objective	Performance Characteristics
<u>S</u> : Star Sensor	Galileo Avionica's (TBC) Autonomous Star Tracker (A-STR)	3-Axis Attitude Measurements	<ul style="list-style-type: none"> ▪ white noise, cross axes: 5" (rms)* ▪ white noise, LoS axes: 25" (rms)* ▪ random bias, cross axes: 5" (rms) ▪ random bias, LoS axes: 25" (rms)
<u>S</u> : Sky Mapper	Integrated in Payload: CCD Columns ASM and AF1	3-Axis Rate Measurements	<ul style="list-style-type: none"> ▪ white noise (1-σ), along-scan: 4 mas/sec per star ▪ white noise (1-σ), across-scan: 20 mas/sec per star
<u>A</u> : FEEP Thrusters	LISA 150 μ N Thrusters ?	3-Axis Absolute Pointing Maintenance	<ul style="list-style-type: none"> ▪ Thrust Level: 0.15 mN ▪ Thrust white noise: 0.1% ▪ Quantification: 1 μN

* noise taken at 5 Hz frequency; LoS: Line of Sight


Table 4.1 – Summary of GAIA AOCS Sensors and Actuators

It is important to note that, during the operational mode, a so-called ‘hybridization’ will be performed where the Kalman Filter estimator will employ the information provided by the star tracker (i.e., attitude angles) and by the instrument (i.e., attitude rates).

Finally, it should be mentioned that the GAIA attitude determination and control requirements during the operational mode can be met by the sensor and actuators with the performances as summarized in Table 4.1, as has been shown by the simulations in [3], pp. 45-47.

The selected Field Emission Electric Propulsion (FEEP) thrusters have a very high specific impulse ($I_{sp} = 6000$ sec) and require only 2.6 kg Caesium propellant for a 6-year mission, which is extremely mass-efficient. Another strong advantage (relative to the cold gas nitrogen thrusters selected for HIPPARCOS) is that their thrust levels can be proportionally controlled or modulated. This capability allows an accurate compensation of the disturbances over a wide frequency range. This concept also results in much lower high-frequency dynamic disturbances and the induced jitters can be made negligible. Therefore, the FEEP baseline represents the most suitable actuator option in view of GAIA’s demanding pointing and rate stability requirements.

The control of the attitude relative to the nominal scan law is performed quasi-continuously by commanding a continuous torque profile with values between 0 and 0.25

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 20 of 27
---	------------------------------------	--

mN using a simple proportional-derivative controller and an efficient low-pass filter (e.g., Cauer filter) to limit the measurement noise transmission.

4.3.4. Disturbance Torques

The main disturbance torque acting on the spacecraft is induced by the solar radiation pressure due to the fact that the center of pressure is at a different location from the center of mass. The torque contains a constant part and a harmonic part at the 6-hour scan period. The solar pressure itself is subject to low-frequency variations which may be modeled by a white noise process superimposed on the (seismological) 5-minute oscillations.

Because the control system design is driven by the solar radiation pressure torques, the control bandwidth must be selected to efficiently control these disturbances in order to meet the pointing stability, see Ref. [4], Figure 4.10 on p. 207, which is reproduced here as Figure 4.2. The numerical results of the detailed and realistic simulations are summarized in Table 4.2.

Smaller disturbances that may be affecting the instrument line-of-sight direction may be induced by a number of sources, for instance:

1. noise in the thrusters of the FEEP propulsion system
2. noise in the sensor's attitude and rate measurements
3. calibration error between star sensor frame and instrument line-of-sight
4. thermo-elastic fluctuations
5. center of mass variation due to propellant motion
6. thruster plume impingement effects

The first two effects are taken into account by the Attitude Control and Measurement System (ACMS). The disturbances produced by the FEEP thrusters include a white dispersion noise below the actuation frequency, a quantification noise, and a sampling noise corresponding to the thrust amplitude refreshing frequency.

The main sensor measurement errors are:

- the star sensor's noise-equivalent angle and random bias,
- the instrument's sky mapper rate measurement noise

Both of these are transmitted to the spacecraft attitude through the ACMS control bandwidth (of 0.005 Hz), the actuation system and the spacecraft dynamics. A first filtering of the measurement noise is performed by the Kalman Filter attitude estimator and a second one is done by the Cauer filter, i.e. high-frequency rejection. Furthermore, a third filtering is done by the controller, depending on its bandwidth, and a fourth by the spacecraft dynamics.

The calibration error between the star sensor and instrument (# 3 in the list above) has a *direct* impact on the pointing error. The initial misalignment can be determined through a dedicated calibration campaign using star position measurements by the star mapper and instrument. It is expected that most of the star sensor noise and random bias will be filtered and will have a negligible impact on the misalignment calibration.

The thermo-elastic fluctuations (# 4 in the list above) are induced by the satellite 6-hour scan period and generate a periodic misalignment between the star mapper and the instrument (similarly as was the case in the HIPPARCOS mission). However, it is expected that the effect will be much smaller than the pointing specification because the star tracker is mounted on the instrument optical bench which is designed and controlled to ensure 10 μ as basic angle stability.

The effect of the center of mass variation (# 5 in the list above) will be minimized by the use of dedicated trapping devices in the propellant tanks. Similarly, the effect of plume impingement (# 6 in the list above) will be minimized by using a specific configuration of the FEEP thrusters that ensures a 45° separation between the thrust direction from both the dust shield and the payload module (note: the FEEP ejection plume has a 40° half-cone angle).

Contributors	Absolute pointing error (mas)	Relative pointing error over 1 s (mas)	Pointing restitution error (μ as)
Measurement system noise	59	0.2	1.4
Star tracker random bias	25000	2.2	0
FEEP noise	0.4	0.015	4.9
Solar pressure noise	0.45	0.012	0.4
Harmonic torques	322	0.2	0
Constant torques	1931	0	0
Total (rms sum)	25.1 arcsec	2.2	5.1
Specification	5 arcmin	2	1000

Table 4.2 – Preliminary Pointing Error, Stability, and Restitution Budgets

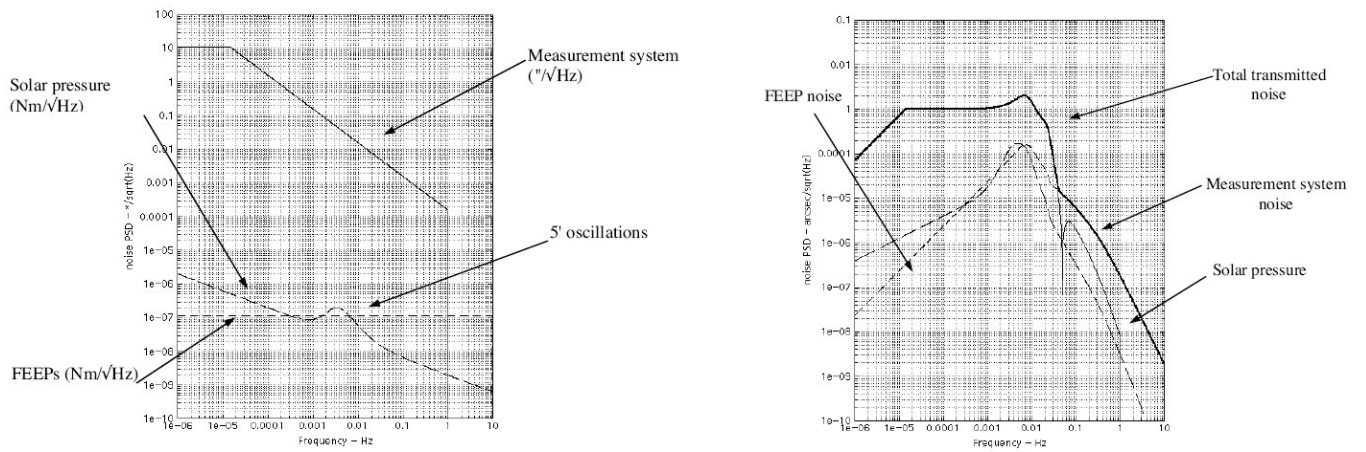



Figure 4.2 – Results of Preliminary Pointing Analysis
a) Noise PSD of Main Disturbances; b) Noise Contribution to Instrument Line-of-Sight Stability

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 23 of 27
---	------------------------------------	--

4.3.5. Basic Angle Variations

The basic angle monitoring concept is described in some (preliminary) detail in ESA-SCI(2000)4, Ref. [4], Chapter 4, and in particular pp. 193 and 200. More recent information can be found in Ref. [3], section 3.5, pp. 34-47.

The requirements on the basic angle fluctuations (i.e., amplitudes) and their monitoring have been formulated in the MRD, Ref. [1], SCI-280, p. 19:

- random contribution: $< 7 \mu\text{as}$
- systematic (over periods larger than about half a scan period) contribution: $< 4 \mu\text{as}$
- monitoring shall be accurate to better than $0.5 \mu\text{as}$ over 5 minutes science operations

Because of the stable thermal environment at the L2 point and the presence of a flat Sun shield shadowing the whole payload module, the satellite's mechanical/thermal design is capable to fulfill the $10 \mu\text{as}$ basic angle stability requirements passively. Nevertheless, it is necessary to implement a device for monitoring the *relative* Line-of-Sight (LoS) variations of the two viewing directions. The required short-term LoS stability (or at least knowledge) requirement of $1 \mu\text{as}$ (rms) over a spacecraft revolution is unusual and can not properly be calibrated by ground processing and is also extremely demanding in terms of the thermal requirements of the payload module structure, see [3], p. 194.

Thermal effects naturally lead to basic angle fluctuations and these have been assessed by thermal analysis (see [3], pp. 200-201). Fluctuations over time periods well above the scan period of 6 hours do not harm the mission performance, whereas relatively fast fluctuations are naturally filtered by the payload module's thermal inertia. The final results show that a passive basic angle stability of $6.4 \mu\text{as}$ is achievable.

Also the effects of satellite rotation and of the gravitational accelerations and gradients at L2 on the distortion of the payload structure have been assessed. Only spin-rate fluctuations need to be considered because a constant spin has no effect on the structural distortion. It has been shown that the fluctuations in the basic angle under the resulting structural flexure and torsion remain well below $1 \mu\text{as}$, see [3], p. 198.

Furthermore, external disturbance forces induced by the gravity gradient due to Sun and Earth can be shown to be of the order of 10^{-10} N and lead to negligible fluctuations in the telescopes line-of-sight of $10^{-5} \mu\text{as}$, see [3], p. 198.

5. SMOOTHING ESTIMATION PROCEDURE

5.1. Introduction

The so-called ‘smoothing’ estimation procedure refers to a non-real-time or ‘off-line’ algorithm that makes optimal use of all measurements collected during a given interval of time. The smoothing estimator provides thus the optimal state estimate based on all information delivered by the measurements sampled during the interval $0 \leq t \leq T$. The smoothed state estimate at time t is denoted by $\hat{\mathbf{x}}(t|T)$.

Typical smoothing algorithms are based on the combination of two individual optimal filters, namely the ‘forward’ and the ‘backward’ filters. The first filter uses all data *before* the time t which leads to the familiar classical Kalman filter estimate $\hat{\mathbf{x}}(t)$, whereas the backward filter operates on all data produced *after* the time t and is expressed as $\hat{\mathbf{x}}_b(t)$.

5.2. Optimal Smoother Algorithm

We seek the optimal smoother algorithm as the sum of the forward and backward filters as discussed above:

$$\hat{\mathbf{x}}(t|T) = [A] \hat{\mathbf{x}}(t) + [B] \hat{\mathbf{x}}_b(t) \quad (5.1)$$

with weighting matrices $[A]$ and $[B]$ that still need to be determined. It is important to note that it can be proven that the forward and backward filter estimates $\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{x}}_b(t)$ have errors that are *uncorrelated*. The proof of this statement is based on the white noise properties of the system and measurement noise processes, see Gelb [12], p. 156-157.

Under the assumption that both estimation errors $\tilde{\mathbf{x}}(t)$ and $\tilde{\mathbf{x}}_b(t)$ are unbiased (which is in fact the case for well-behaving Kalman Filter estimates) we may require that also the smoother estimate $\hat{\mathbf{x}}(t|T)$ should be *unbiased*. The estimation error of the smoother estimate follows now immediately from eq. (5.1):

$$\begin{aligned} \tilde{\mathbf{x}}(t|T) &= \hat{\mathbf{x}}(t|T) - \mathbf{x}(t) = [A] \{\mathbf{x}(t) + \tilde{\mathbf{x}}(t)\} + [B] \{\mathbf{x}(t) + \tilde{\mathbf{x}}_b(t)\} - \mathbf{x}(t) = \\ &= [A + B - I] \mathbf{x}(t) + [A] \tilde{\mathbf{x}}(t) + [B] \tilde{\mathbf{x}}_b(t) \end{aligned} \quad (5.2)$$

This estimate will be unbiased provided that the matrix $[A + B - I]$ vanishes, which implies that $[B] = [I - A]$ and eq. (5.1) can be simplified as follows:

$$\hat{\mathbf{x}}(t|T) = [A]\hat{\mathbf{x}}(t) + [I - A]\hat{\mathbf{x}}_b(t) \quad (5.3)$$

The error covariance matrix $[P(t|T)]$ of the smoother estimate $\hat{\mathbf{x}}(t|T)$ can now be established from the result in eq (5.3) as follows:

$$[P(t|T)] = E\{\tilde{\mathbf{x}}(t|T)\tilde{\mathbf{x}}^T(t|T)\} = [A][P(t)][A]^T + [I - A][P_b(t)][I - A]^T \quad (5.4)$$

The matrices $[P(t)]$ and $[P_b(t)]$ appearing here denote the covariances of the forward and backward optimal state estimates $\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{x}}_b(t)$, respectively. It should also be noted that the product terms between the estimation errors $\tilde{\mathbf{x}}(t)$ and $\tilde{\mathbf{x}}_b(t)$ vanish because of the fact that the corresponding forward and backward state estimates are completely uncorrelated.

The value of $[A(t)]$ at any time t is now selected to be the one that minimizes the trace of the covariance matrix $[P(t|T)]$, see Gelb [12], p. 158, eq. (5.1-7):

$$[A(t)] = [P_b(t)][P(t) + P_b(t)]^{-1} \quad (5.5)$$

It can readily be shown that:

$$[I - A(t)] = [P(t)][P(t) + P_b(t)]^{-1} \quad (5.6)$$

These results confirm the expected symmetry between the weighting matrices. The resulting covariance matrix $[P(t|T)]$ after the smoothing follows now as:

$$[P(t|T)] = [A][P(t)][A]^T + [I - A][P_b(t)][I - A]^T \quad (5.7)$$

After substituting the expressions from (5.5) and (5.6) the result of eq. (5.7) can be simplified considerably as shown by Gelb [12], p. 158, eqs. (5.1-10 and 11):

$$[P(t|T)] = \{[P(t)]^{-1} + [P_b(t)]^{-1}\}^{-1} \quad (5.8)$$

The form of this result is typical for the variance resulting from the combination of two uncorrelated estimates. It is obvious from the above result that the smoothed estimate is always better (or at worst equal) than the filtered estimate as illustrated schematically by the behavior of its state covariance in Figure 5.1 (from Gelb [12], p. 159, Figure 5.1-1).

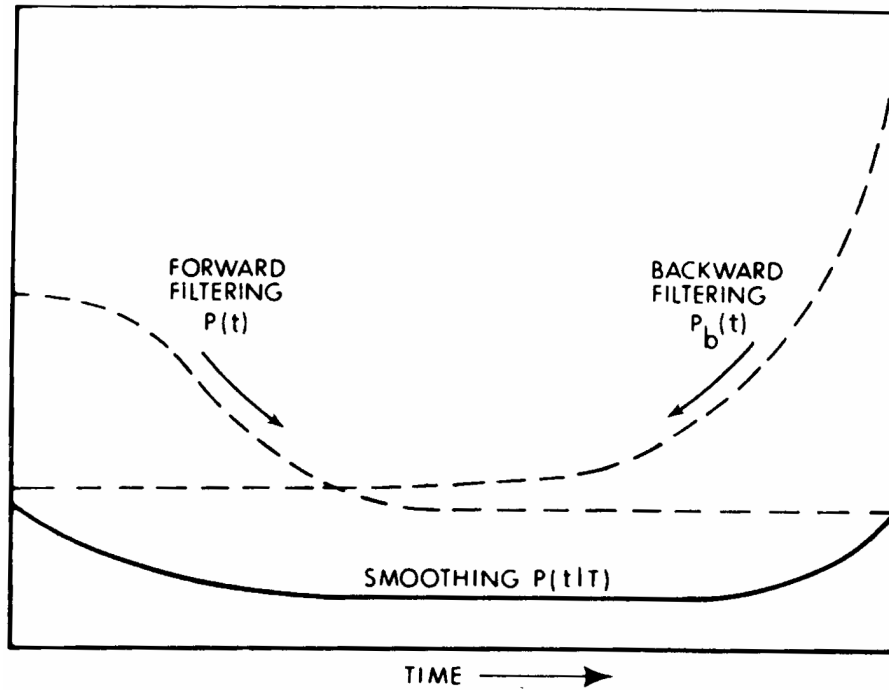



Figure 5.1 – Combined State Covariance of Optimal Smoothing Estimate

Finally, it is of interest to express the smoothed state estimate of eq. (5.3) in terms of the covariance matrices with the help of eqs. (5.5) and (5.6), Gelb [12], p. 158, eq. (5.1-12):

$$\hat{\mathbf{x}}(t|T) = [P(t|T)] \{ [P(t)]^{-1} \hat{\mathbf{x}}(t) + [P_b(t)]^{-1} \hat{\mathbf{x}}_b(t) \} \quad (5.9)$$

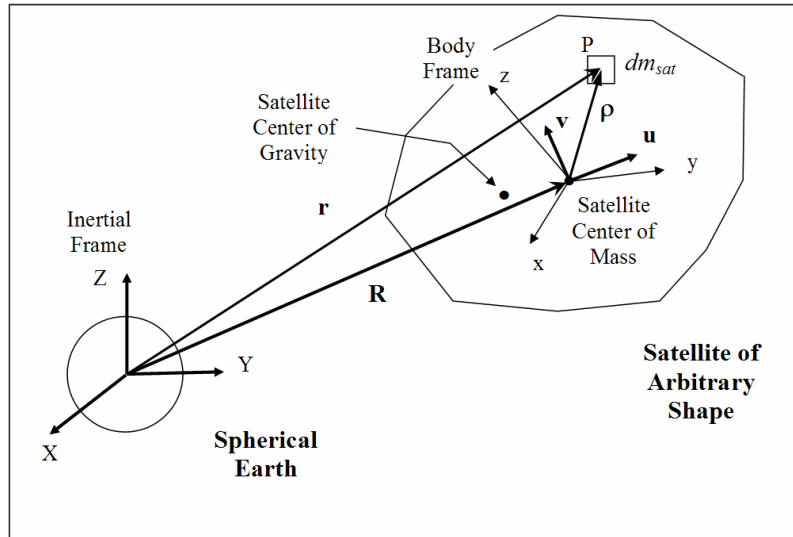
Equations (5.8) and (5.9) are the fundamental results presenting the smoothed estimate and its covariance matrix.

	GAIA Ground Attitude Determination	Doc.No.: FLK-SIM-ZAR-005 Issue: 1.0 Page: 27 of 27
---	------------------------------------	--

References

- [1] ESA, GAIA Mission Requirements Document, GAIA-EST-RD-00553, 1 July 2005.
- [2] ESA, GAIA System Requirements Document, GAIA-SRD-001, 30 March 2004.
- [3] Astrium, GAIA System Level Technical Reassessment Study, Final Report, EF5/FR/PC/038.02, 17/06/2002, Chapter 3.5.
- [4] ESA, GAIA: Composition, Formation and Evolution of the Galaxy, Concept & Technology Study Report, ESA-SCI(2000)4, July 2000.
- [5] L. Lindegren, Attitude Parametrization for GAIA, SAG-LL-30, 1 July 2000.
- [6] F. Mignard, A Practical Scanning Law for GAIA Simulations, GAIA-FM-010, 10 Dec. 2001.
- [7] F. Mignard, The Practical Scanning Law: an Update, GAIA-FM-017, 30 Jan. 2004.
- [8] U. Bastian, Problems with “Satellite Reference Systems”, GAIA-ARI-BAS-004, 13 Oct. 2003.
- [9] Lindegren, L., Requirements on the Attitude Noise, SAG-LL-020, 26 June 1998.
- [10] Lindegren, L., Requirements on the Attitude Noise (2), SAG-LL-048, 20 December 2003.
- [11] S. Jordan, U. Bastian, H. Lenhardt, H.-H. Bernstein, S. Hirte, M. Bierman, GAIA First Look, Symposium “The Three Dimensional Universe with GAIA”, Observatoire de Meudon, Paris, 4-7 Oct. 2004, pp. 405-411.
- [12] A. Gelb (Editor), Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

Model of Gravity Force and Gravity Gradient Torque



Project:	Document No.:
FIRST LOOK	FLK-SIM-TN-ZAR-004
ZARM –Center of Applied Space Technology and Microgravity	
Am Fallturm D28359 Bremen Germany	Phone: +49-421-218-9422 Fax: +49-421-218-4356 E-Mail: jvdha@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-004

Issue: 1.0

Written: Jozef van der Ha


Date: April 13, 2006

Approved: –

Date: –

CONTENTS

1	INTRODUCTION.....	3
2	DEFINITIONS.....	4
2.1	GEOMETRY.....	4
2.2	ASSUMED EARTH MODEL	5
3.	ANALYSIS OF GRAVITY FORCE.....	6
3.1	EXPANSION OF FORCE	6
3.2	ANALYSIS OF INERTIA TERMS.....	7
3.3	REDUCTION OF INTEGRAL TERMS.....	9
3.4	FINAL FORM OF GRAVITY FORCE	10
3.5	INTERPRETATION OF FORCE RESULT	11
4.	RESULT FOR GRAVITY POTENTIAL	13
5.	GRAVITY GRADIENT TORQUE.....	14
5.1	EXPANSION OF TORQUE EXPRESSION	14
5.2	INTERPRETATION OF TORQUE RESULT	16
5.3	ALTERNATIVE TORQUE DERIVATION.....	17
6.	POSITION OF CENTER OF GRAVITY	20
6.1	ASSESSMENT OF [1]	20
6.2	ALTERNATIVE DEFINITION OF CENTER OF GRAVITY	21
	REFERENCES	24
	APPENDIX A. DUMBBELL SATELLITE ILLUSTRATION.....	25
A.1	GEOMETRY	25
A.2	EXPANSION OF GRAVITY FORCE	26
A.3	RESULT OF GRAVITY FORCE MODEL	28
A.4	CENTER OF GRAVITY	29
A.5	GRAVITY-GRADIENT TORQUE	30
A.6	ILLUSTRATION OF RESULTS	31
	APPENDIX B. CYLINDRICAL SATELLITE ILLUSTRATION	35
B.1	GEOMETRY	35
B.2	RESULT OF GRAVITY FORCE MODEL	36
B.3	GRAVITY-GRADIENT TORQUE.....	37
B.4	CYLINDER NORMAL TO ORBIT PLANE.....	37
	APPENDIX C. COMPARISON OF FORCE RESULTS	39
C.1	FORCE EXPRESSED IN MOMENTS OF INERTIA.....	39
C.2	FORCE EXPRESSED IN INERTIA INTEGRALS	40
	APPENDIX D. GRAVITY FORCE IN ORBIT FRAME.....	43

	Model of Gravity Force and Gravity Gradient Torque	Doc.No.: FLK-SIM-TN-ZAR-004 Issue: 1.0 Page: 3 of 45
---	---	--

1 Introduction

This paper summarizes the detailed evaluation of the analysis leading to the gravity force in a cartesian system presented in Ref. [1]. It provides an independent derivation of the force equation and leads to a more compact representation of the force model.

Furthermore, compact expressions are derived and established for the gravitational potential function, for the gravity gradient torque, and also for the center of gravity position.


In the analysis, we consider a general satellite configuration with arbitrary moments of inertia and an arbitrary satellite orientation relative to the local orbital reference frame. We find that the resulting gravity force vector does not pass through the center of attraction, even when assuming an ideal spherical gravity field and the adoption of a body reference frame based on the principal inertia axes.

The position of the center of gravity does in general not lie on the orbital radius vector and also not on a satellite principal axis.

The results obtained in the analyses are interpreted as far as feasible in terms of their physical meanings and implications.

Finally, the results are illustrated using actual satellite examples, namely an idealized dumbbell satellite system and a more realistic satellite configuration with cylindrical moments of inertias.

The red text shown in this Report identifies comments and issues that are related to the contents of Ref. [1].

	Model of Gravity Force and Gravity Gradient Torque	Doc.No.: FLK-SIM-TN-ZAR-004 Issue: 1.0 Page: 4 of 45
---	---	--

2 DEFINITIONS

2.1 Geometry

The following reference frames will be used in this study (see Figure 1):

1. X, Y, Z: Earth-Centered *Inertial* Reference Frame (ECIRF), i.e. J2000 equatorial
2. u, v, w: *local* Orbital Reference Frame (ORF), with axes u along local vertical, v along local horizontal, and w along orbit-normal (normal to paper in Figure 1)
3. x, y, z: *Body-Centered* Reference Frame (BCRF) consisting of a well-defined spacecraft geometric frame with origin at the satellite center of mass (Figure 1)
4. x_p, y_p, z_p : *Body Principal* Reference Frame (BPRF), which points along the axes of minimum and maximum moments of inertia with origin at the center of mass.

Sometimes, the body and principal reference frame are taken to be identical, mainly for convenience reasons. It should be noted that, for real-world satellite applications, this assumption is invalid (or at least misleading). The reason for this is that the *actual* principal axes, in contrast to the body reference axes, are not known to good accuracy because of pre-launch balancing imperfections and in-orbit variations in the moments of inertia (for instance, due to propellant usage).

The BCRF and ORF frames are assumed to be equal in [1], Figure 1 and B1.2. This assumption is too restrictive because these frames are different in general. In particular, the ORF moves along with the orbital radius and describes a full revolution per orbit within the inertial frame. On the other hand, the inertial motion of the satellite body frame is determined by the mode of attitude stabilization:

- In the case of a three-axis stabilized satellite, the body frames would be (almost) fixed in inertial space, apart from the attitude motion within the three-axis control dead-band.
- For a scanning satellite (like HIPPARCOS and GAIA), the body frames move slowly, i.e. at a rate of one revolution every few hours, within inertial space
- For a spin-stabilized satellite, the spin axis is essentially fixed in inertial space (apart from very slow motion of up to a few degrees over months induced by perturbing torques) but the other two body axes move quickly under the spin motion.

Another issue is that Figures 1 and 2 on p. 7 and 8 in Ref. [1] are *not* consistent. The orbital radius vector \mathbf{R} points from the center of attraction (Earth) to the center of mass of the satellite as shown (correctly) in Figure 1 of Ref. [1]. In Figure 2 of Ref. [1], however, \mathbf{R} points in the opposite direction, i.e. from the satellite to the center of attraction. The plausible reason for this inconsistency comes from the fact that Figure 2 is taken literally from the book by Meirovitch [2], p. 431, who uses this uncommon (at least, within the satellite community) definition. We recommend that the geometry of Figure 1 below should be employed with the vector \mathbf{R} pointing in the *positive* radial direction.

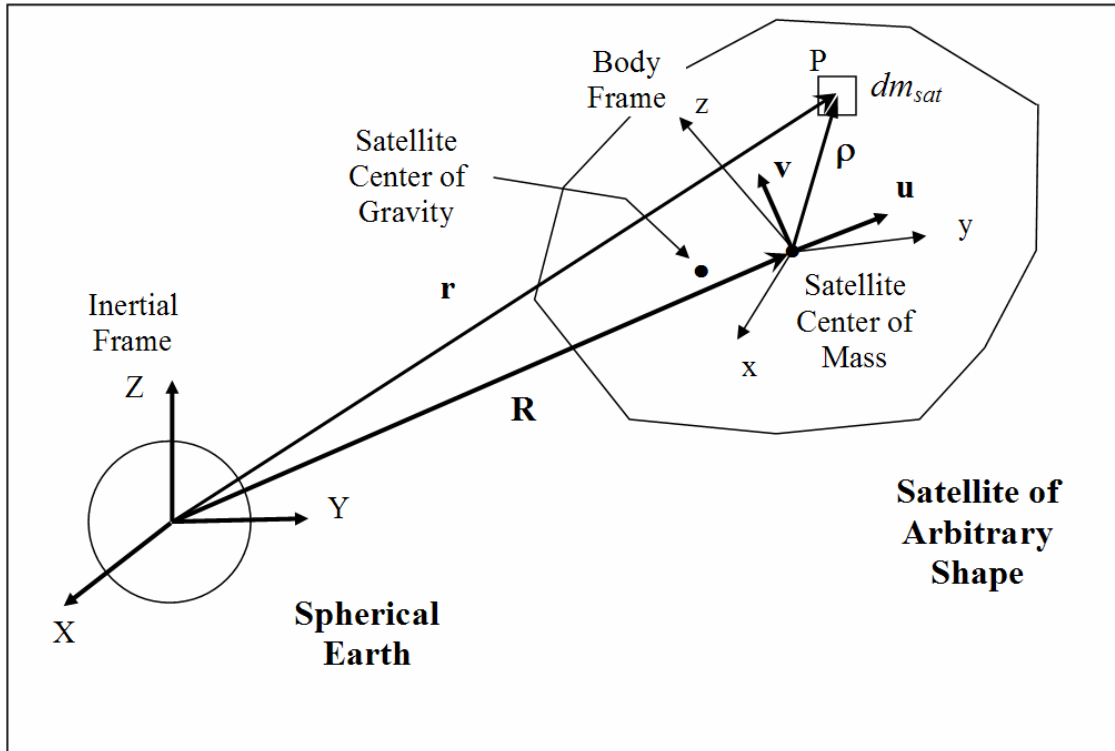


Figure 1 – Illustration of Geometry of Earth and Satellite

2.2 Assumed Earth Model

The last sentence on p. 7 of [1] should be formulated in a different manner in order to clarify and justify the reason behind the assumption that an ideal spherical Earth with uniform mass distribution may be assumed in the model (mathematical convenience is not the main justification). A spherical Earth is equivalent to a point mass in terms of its gravity force. The effects of the Earth's non-spherical shape (i.e., the zonal and tesseral harmonics of the Earth's gravity field) primarily affect the satellite's orbit.

The spherical Earth model appears to be acceptable for the purpose of calculating torques acting on the satellite body. The errors induced by the uncertainty and unpredictability of other external forces (e.g., solar radiation pressure and geomagnetic field) are expected to be larger than those due to the non-spherical shape of the Earth. Also there are unknown errors in the knowledge of the satellite attitude pointing and in the location of the inertia axes within the satellite (typically of a magnitude of up to 0.1 deg.). Both of these effects may also be more important than the Earth's oblateness effect on the attitude motion.

On the other hand, when it comes to the modeling of the motion of the test mass relative to the satellite, oblateness effects on the gravity force (and perhaps also torque?) may well need to be incorporated. In any case, models for evaluating the effects of the J_2 Earth oblateness on the force and torque models, are available in the satellite literature, see for instance Beletskii, Ref. [3], p. 11-13, and Hughes [4], pp. 240-243.

3. ANALYSIS OF GRAVITY FORCE

3.1 Expansion of Force

Eq. (3) of Ref. [1] gives the expansion of r^{-3} (with r representing the magnitude of the radius vector \mathbf{r} of the satellite mass element dm with respect to the center of attraction). The expansion is performed in terms of powers of R , i.e. the orbital distance of the satellite center of mass from the center of the attraction. Due to the sign inconsistency mentioned in Section 1, we must replace the vectors in eq. (2) of Ref. [1] as follows:

$$\mathbf{r} \rightarrow -\mathbf{r}; \quad \mathbf{R} \rightarrow -\mathbf{R} \quad (1a)$$

$$\boldsymbol{\rho} = \mathbf{R} - \mathbf{r} \rightarrow \boldsymbol{\rho} = \mathbf{r} - \mathbf{R} \quad (1b)$$

$$r = |\mathbf{R} - \boldsymbol{\rho}| \rightarrow r = |\mathbf{R} + \boldsymbol{\rho}| \quad (1c)$$

When writing $R = |\mathbf{R}|$ and $\rho = |\boldsymbol{\rho}|$, we can express $r = |\mathbf{r}|$ as follows:

$$r = |\mathbf{R} + \boldsymbol{\rho}| = R \left\{ 1 + \frac{2(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} + \left(\frac{\rho}{R} \right)^2 \right\}^{1/2} \quad (2)$$

Because the ratio (ρ/R) is small, it is useful to expand r^{-3} using the generic formula:

$$(1 + \varepsilon)^{-3/2} = 1 - (3/2)\varepsilon + (15/8)\varepsilon^2 + O(\varepsilon^3) \quad \text{with } \varepsilon = O(\rho/R) \quad (3)$$

We can now calculate the expression of eq. (3) in Ref. [1] as follows:

$$\begin{aligned} r^{-3} &= |\mathbf{R} + \boldsymbol{\rho}|^{-3} = R^{-3} \left\{ 1 + \frac{2(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} + \left(\frac{\rho}{R} \right)^2 \right\}^{-3/2} = \\ &= R^{-3} \left\{ 1 - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2}{R^4} + O\left(\frac{\rho}{R} \right)^3 \right\} \end{aligned} \quad (4)$$

This result is consistent with Eq. (3) in Ref. [1] when the sign of the vector \mathbf{R} is reversed. For a satellite with dimension of at most 3 meter in length and an orbit of at least 600 km altitude we find that the error terms in eqs. (3) and (4) have a magnitude of at most:

$$O(\varepsilon^3) = O\left(\frac{\rho}{R} \right)^3 \approx 8 \times 10^{-20} \quad (5)$$

The next step is the calculation of the gravity force acting on the satellite. First, the force acting on the satellite's mass element dm_{sat} is given by eq. (1) in [1] with an extra minus sign due to the sign reversal of \mathbf{r} :

$$d\mathbf{F} = - \frac{G m_{Earth} dm_{sat}}{r^3} \mathbf{r} \quad (6)$$

When integrating the gravity force over all mass elements dm_{sat} of the satellite body and substituting the expansion of eq. (3), we obtain the resulting gravity force acting on the satellite as in eq. (4) of [1] (note the minus sign in front and $\mathbf{R} - \boldsymbol{\rho} \rightarrow \mathbf{R} + \boldsymbol{\rho}$):

$$\mathbf{F} = - \frac{\mu_E}{R^3} \int_{m_{sat}} \left\{ 1 - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2}{R^4} + O \left(\frac{\rho}{R} \right)^3 \right\} (\mathbf{R} + \boldsymbol{\rho}) dm_{sat} \quad (7)$$

with $\mu_E = Gm_{Earth}$ as in eq. (5) of [1]. We can rewrite eq. (4) in the form:

$$\begin{aligned} \mathbf{F} = - \frac{\mu_E}{R^3} \int_{m_{sat}} \left\{ \mathbf{R} + \boldsymbol{\rho} - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})\mathbf{R}}{R^2} - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})\boldsymbol{\rho}}{R^2} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 \mathbf{R} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 \boldsymbol{\rho} + \right. \\ \left. + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2 \mathbf{R}}{R^4} + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2 \boldsymbol{\rho}}{R^4} + O \left(\frac{\rho}{R} \right)^3 \mathbf{R} \right\} dm_{sat} \quad (8) \end{aligned}$$

This result is consistent with eq. (7) of [1] after we correct for the sign reversal of \mathbf{R} . The leading term within the integrand $\{ \dots \}$ of eq. (8) is \mathbf{R} so that the magnitude of the term $(\rho/R)^2 \boldsymbol{\rho}$ is of the order $(\rho/R)^3$ times smaller than R . It is obvious that the same argument applies also to the term $(\mathbf{R} \cdot \boldsymbol{\rho})^2 \boldsymbol{\rho}/R^4$ so both of these terms are negligible. The remaining terms can be reduced by employing the moments of inertia (shown next).

3.2 Analysis of Inertia Terms

We can express the position vectors $\boldsymbol{\rho}$ and \mathbf{R} in terms of coordinates in the 'Body Centered Reference Frame' (BCRF), see Figure 1, as follows:

$$\boldsymbol{\rho} = (x, y, z)^T; \quad \mathbf{R} = R (l, m, n)^T \quad \rightarrow \quad (\mathbf{R} \cdot \boldsymbol{\rho}) = R (l x + m y + n z) \quad (9a-c)$$

It should be noted that the body frame BCRF and thus also the vector $\boldsymbol{\rho}$ have their origins at the satellite's center of mass [1]. Because the center of mass is taken at the origin of the adopted reference frame, we have the following results:

$$\int_{m_{sat}} \boldsymbol{\rho} dm_{sat} = \mathbf{0} \quad \rightarrow \quad \int_{m_{sat}} x dm_{sat} = \int_{m_{sat}} y dm_{sat} = \int_{m_{sat}} z dm_{sat} = 0 \quad (10)$$

These expressions imply also the following results:

$$\int_{m_{sat}} (\mathbf{R} \cdot \boldsymbol{\rho}) dm_{sat} = 0 ; \quad \int_{m_{sat}} (\mathbf{R} \cdot \boldsymbol{\rho}) \mathbf{R} dm_{sat} = \left\{ \int_{m_{sat}} (\mathbf{R} \cdot \boldsymbol{\rho}) dm_{sat} \right\} \mathbf{R} = 0 \quad (11)$$

From elementary geometry we have:

$$|\boldsymbol{\rho}|^2 = \rho^2 = (x^2 + y^2 + z^2) ; \quad |\mathbf{R}|^2 = R^2 \quad (12)$$

(Note that [1], eq. (9) contains the superfluous terms $l^2 + m^2 + n^2$, which is equal to 1.)

$$(\mathbf{R} \cdot \boldsymbol{\rho})^2 = \{(lx)^2 + (my)^2 + (nz)^2 + 2(lmxy + lnxz + mnyz)\} \quad (13)$$

For a body of arbitrary shape and mass distribution, we introduce the second-order ‘inertia integrals’ defined by:

$$J_x = \int_{m_{sat}} x^2 dm_{sat} = \frac{1}{2} (I_y + I_z - I_x) \quad (14a)$$

$$J_y = \int_{m_{sat}} y^2 dm_{sat} = \frac{1}{2} (I_x + I_z - I_y) \quad (14b)$$

$$J_z = \int_{m_{sat}} z^2 dm_{sat} = \frac{1}{2} (I_x + I_y - I_z) \quad (14c)$$

$$J_{xy} = I_{xy} = \int_{m_{sat}} xy dm_{sat} \quad (14d)$$

$$J_{xz} = I_{xz} = \int_{m_{sat}} xz dm_{sat} \quad (14e)$$

$$J_{yz} = I_{yz} = \int_{m_{sat}} yz dm_{sat} \quad (14f)$$

The I_x, I_y, I_z appearing here are the well known conventional ‘moments of inertia’, which are defined as:

$$I_x = \int_{m_{sat}} y^2 + z^2 dm_{sat} ; \quad I_y = \int_{m_{sat}} x^2 + z^2 dm_{sat} ; \quad I_z = \int_{m_{sat}} x^2 + y^2 dm_{sat} \quad (15a-c)$$

We introduce also the ‘inertia matrix’ [I] as:

$$[\mathbf{I}] = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \quad (16)$$

When using the definitions given in eqs. (14 - 16), it is easy to establish the identities:

$$\int_{m_{sat}} \rho^2 dm_{sat} = J_x + J_y + J_z = \frac{1}{2} (I_x + I_y + I_z) = \frac{1}{2} tr[\mathbf{I}] \quad (17)$$

Here, $tr[\mathbf{I}] = I_x + I_y + I_z$ denotes the trace of the inertia matrix $[\mathbf{I}]$. With the definitions introduced above, we can now evaluate the two integrals:

$$\begin{aligned} \int_{m_{sat}} \boldsymbol{\rho} (\mathbf{R} \bullet \boldsymbol{\rho}) dm_{sat} &= R \int_{m_{sat}} \begin{pmatrix} lx^2 + mxy + nxz \\ lxy + my^2 + nyz \\ lxz + myz + nz^2 \end{pmatrix} dm_{sat} = R \begin{pmatrix} lJ_x + mJ_{xy} + nJ_{xz} \\ lJ_{xy} + mJ_y + nJ_{yz} \\ lJ_{xz} + mJ_{yz} + nJ_z \end{pmatrix} = \\ &= \{ \frac{1}{2} tr[\mathbf{I}] - [\mathbf{I}] \} \mathbf{R} \end{aligned} \quad (18a)$$

$$\begin{aligned} \int_{m_{sat}} (\mathbf{R} \bullet \boldsymbol{\rho})^2 dm_{sat} &= R^2 \int_{m_{sat}} \{ (lx)^2 + (my)^2 + (nz)^2 + 2(lmxy + lnxz + mnyz) \} dm_{sat} = \\ &= R^2 \{ l^2 J_x + m^2 J_y + n^2 J_z + 2(lm J_{xy} + ln J_{xz} + mn J_{yz}) \} = \frac{1}{2} R^2 tr[\mathbf{I}] - \mathbf{R}^T [\mathbf{I}] \mathbf{R} \end{aligned} \quad (18b)$$

It can be seen that the quadratic form $\mathbf{R}^T [\mathbf{I}] \mathbf{R}$ stands for the scalar quadratic expression:

$$\mathbf{R}^T [\mathbf{I}] \mathbf{R} = R^2 \{ l^2 I_x + m^2 I_y + n^2 I_z - 2(lm I_{xy} + ln I_{xz} + mn I_{yz}) \} \quad (19)$$

3.3 Reduction of Integral Terms

With the help of the results presented above, we can reduce the individual terms that appear within the brackets $\{ \dots \}$ of the gravity force expression of eq. (8) as follows:

$$\text{Term 1:} \quad \int_{m_{sat}} \{ \mathbf{R} + \boldsymbol{\rho} \} dm_{sat} = m_{sat} \mathbf{R} \quad (20a)$$

$$\text{Term 2:} \quad \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \bullet \boldsymbol{\rho}) \mathbf{R}}{R^2} \right\} dm_{sat} = \left(\mathbf{R} \bullet \left\{ \int_{m_{sat}} \boldsymbol{\rho} dm_{sat} \right\} \right) \mathbf{R} / R^2 = \mathbf{0} \quad (20b)$$

$$\text{Term 3:} \quad \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \bullet \boldsymbol{\rho}) \boldsymbol{\rho}}{R^2} \right\} dm_{sat} = \{1/2 \text{tr}[\mathbf{I}] - [\mathbf{I}]\} \mathbf{R} / R^2 \quad (20c)$$

$$\text{Term 4:} \quad \int_{m_{sat}} \left\{ \left(\frac{\rho}{R} \right)^2 \mathbf{R} \right\} dm_{sat} = 1/2 \text{tr}[\mathbf{I}] \mathbf{R} / R^2 \quad (20d)$$

$$\text{Term 5:} \quad \int_{m_{sat}} \left\{ \left(\frac{\rho}{R} \right)^2 \boldsymbol{\rho} \right\} dm_{sat} = O\left(\frac{k}{R}\right) [\mathbf{I}] / R \quad (\text{i.e., negligible}) \quad (20e)$$

$$\text{Term 6:} \quad \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \bullet \boldsymbol{\rho})^2 \mathbf{R}}{R^4} \right\} dm_{sat} = 1/2 \text{tr}[\mathbf{I}] \mathbf{R} / R^2 - (\mathbf{R}^T [\mathbf{I}] \mathbf{R}) \mathbf{R} / R^4 \quad (20f)$$

$$\text{Term 7:} \quad \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \bullet \boldsymbol{\rho})^2 \boldsymbol{\rho}}{R^4} \right\} dm_{sat} = O\left(\frac{k}{R}\right) [\mathbf{I}] / R \quad (\text{i.e., negligible}) \quad (20g)$$

The parameter k is the ‘radius of gyration’ and is representative of the satellite’s effective size. The neglected terms in eqs. (20e) and (20g) are smaller than the retained terms by a factor of the order of $k/R \approx 3 \times 10^{-7}$ for a typical-size satellite above 600 km altitude.

3.4 Final Form of Gravity Force

After collecting and rearrangement of terms, we find the final result for the gravity force up to second-order terms $(\rho/R)^2$ from eq. (8). A further simplification is achieved by writing $\mathbf{R} = R \mathbf{u}$ where \mathbf{u} represents the unit-vector along the orbital radius (Figure 1):

$$\mathbf{F} \approx -m_{sat} (\mu_E / R^2) \mathbf{u} + (3/2) (\mu_E / R^4) \{5 (\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} - \text{tr}[\mathbf{I}] \mathbf{u} - 2[\mathbf{I}] \mathbf{u}\} \quad (21)$$

After projecting the vectors $\mathbf{u} = (l, m, n)^T$, from eq. (9b), and $[\mathbf{I}] \mathbf{u}$ upon the axes of the spacecraft-fixed (x, y, z) body reference frame, we find the following result for the force components (see also Appendix C for more details):

$$\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} \approx -m_{sat} (\mu_E / R^2) \begin{pmatrix} l \\ m \\ n \end{pmatrix} +$$



$$+ (3/2) (\mu_E/R^4) \begin{pmatrix} l\{(5l^2-3)I_x+(5m^2-1)I_y+(5n^2-1)I_z-10(lmI_{xy}+lnI_{xz}+mnI_{yz})+2(mI_{xy}+nI_{xz})\} \\ m\{(5l^2-1)I_x+(5m^2-3)I_y+(5n^2-1)I_z-10(lmI_{xy}+lnI_{xz}+mnI_{yz})+2(lI_{xy}+nI_{yz})\} \\ n\{(5l^2-1)I_x+(5m^2-1)I_y+(5n^2-3)I_z-10(lmI_{xy}+lnI_{xz}+mnI_{yz})+2(lI_{xz}+mI_{yz})\} \end{pmatrix} \quad (22)$$

This result corresponds to the expression in eq. (24) of [1]. The agreement between the results is perfect, **except for the factor 4 in [1] instead of 2 in front of the $(mI_{xy} + nI_{xz})$ term at the end of the first row (and similarly for the other rows as well). Furthermore, in eq. (24) of [1], this term is wrongly multiplied by the components l , m , and n .**

In the case when the spacecraft body axes are selected to be the principal axes, the cross-product terms I_{xy} , I_{xz} , I_{yz} of the inertia matrix vanish and the second result of eq. (22) can be simplified further:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} \approx -m_{sat} (\mu_E/R^2) \begin{pmatrix} l \\ m \\ n \end{pmatrix} + (3/2) (\mu_E/R^4) \begin{pmatrix} (5l^2-3)I_x+(5m^2-1)I_{yy}+(5n^2-1)I_{zz} \\ (5l^2-1)I_{xx}+(5m^2-3)I_y+(5n^2-1)I_{zz} \\ (5l^2-1)I_{xx}+(5m^2-1)I_{yy}+(5n^2-3)I_z \end{pmatrix} \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (23)$$

This result is identical to the one given in eq. (25) of [1], **apart from the sign reversal.**

Appendix D provides further results for the gravity gradient force expressed within the orbital reference frame.

3.5 Interpretation of Force Result

The compact result in eq. (21) allows the following dynamical interpretations:

1. The leading (monopole) term of the gravity force has the order of magnitude $m_{sat} (\mu_E/R^2)$ and represents the gravity force acting on an idealized ‘point-mass’ satellite without physical dimensions; this force is responsible for the satellite motion along its orbit.
2. In the special case of an ideal spherical satellite with uniform mass distribution, all spacecraft axes are principal axes and the inertias I_x , I_y , I_z are identical. Thus, the second-order (quadrupole) term of order μ_E/R^4 in (23) can be shown to vanish and the monopole term represents again the total force acting on the body. Thus, the sphere is attracted as if all its mass were concentrated in its center of mass and its orbital and attitude motion are uncoupled, at least up to second-order. The orbit is the same as that of a point-mass and the attitude motion remains unaffected by the gravity force.



3. The observations for a spherical satellite in point 2 above hold also true for a cubic satellite with its body axes along the principal axes and with equal moments of inertia along these axes (i.e., $I_x = I_y = I_z$). The inertia product terms vanish in this case (i.e., $I_{xy} = I_{yz} = I_{zx} = 0$).
4. The quadrupole term of the force is caused by the small variations in the gravity force acting at different points in the extended satellite body; its magnitude is of the order of $(\mu_E [I] / R^4)$ and is smaller than the leading term (μ_E / R^2) by a factor $(k/R)^2$, which is roughly 10^{-13} for a typical satellite in an orbit above 600 km.
5. For a point-mass satellite, the gravity force obviously acts exactly along the direction of the orbital radius; for an extended satellite, however, this is not true anymore because the vector term $([I]\mathbf{u})$ is in general *not* directed along the instantaneous orbital radius (note that the other terms of eq. (21) *are* oriented along the direction \mathbf{u}). This leads to the final terms of eq. (22) which constitute the non-radial contributions originating from $[I]\mathbf{u}$. It can easily be seen that, for instance, the term $2(mI_{xy} + nI_{xz})$ in the component F_x is not proportional to l and the same behavior is true for the other two components. It will be shown in Section 5 below that the term $[I]\mathbf{u}$ is responsible for the gravity torque.
6. The quadrupole term has in general an effect on the resulting orbital motion through an attitude / orbit coupling effect. For instance, when considering an infinitesimal rod or ‘pencil-satellite’ with inertias $(0, I, I)$ pointing along the local vertical, we find $l = 1$, $m = n = 0$, so the force in eq. (21) will be along the spacecraft x-axis which coincides with the orbit radial direction:

$$F_{\text{radial}} \approx -m_{\text{rod}} (\mu_E / R^2) + (3/2) (\mu_E / R^4) (-2I_{\text{rod}}) = -m_{\text{rod}} (\mu_E / R^2) \{1 + (\rho/R)^2\} \quad (24)$$

Here, $I_{\text{rod}} = m_{\text{rod}} \rho^2 / 3$ with ρ the half-length of the rod. For a circular orbit, the gravity force is in equilibrium with the centrifugal force which implies that the orbital rate of a point-mass satellite is given by the Keplerian rate $\omega_{\text{orbital}} = \sqrt{(\mu_E / R^3)}$. Eq. (24) indicates that the orbital rate for the pencil-satellite will be a factor $\sqrt{\{1 + (\rho/R)^2\}}$ higher than ω_{orbital} , while assuming that the rod remains oriented along the local vertical. The point where this higher rate happens to be equal to the Kepler rate at its distance from the center of the Earth may be called the ‘center of motion’. It can be shown by means of a first-order Taylor expansion that this center of motion is located a small distance $-(1/3)(\rho/R)\rho$ below the satellite’s center of mass.

7. In the case when a gravity model with still higher accuracy is desired, the full gravitational potential, including zonal and tesseral harmonics of the Earth’s potential field may be considered. In particular, the leading J_2 contribution may be of most interest. Ref. [7] gives the explicit expressions for the resulting force and torque expressions, including the J_2 terms.

4. RESULT FOR GRAVITY POTENTIAL

It is of interest to establish the corresponding expression for the ‘gravitational potential’ V of an arbitrary rigid body in a spherical gravitational field:

$$\begin{aligned}
 V &= -\mu_E \int_{m_{sat}} \frac{1}{r} dm_{sat} = -\frac{\mu_E}{R} \int_{m_{sat}} \left\{ 1 + \frac{2(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} + \left(\frac{\rho}{R} \right)^2 \right\}^{-1/2} dm_{sat} = \\
 &= -\frac{\mu_E}{R} \int_{m_{sat}} \left\{ 1 - \frac{(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} - \frac{1}{2} \left(\frac{\rho}{R} \right)^2 + \frac{3}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2}{R^4} + O \left(\frac{\rho}{R} \right)^3 \right\} dm_{sat} \quad (25)
 \end{aligned}$$

By applying the identities established in eqs. (20b, d, f), we can reduce the potential function to its final form:

$$V \approx -m_{sat} (\mu_E / R) + 1/2 (\mu_E / R^3) \{ 3(\mathbf{u}^T [\mathbf{I}] \mathbf{u}) - tr[\mathbf{I}] \} \quad (26)$$

It can be shown that the force expression of eq. (21) can be obtained from the potential function for a homogeneous spherical Earth by calculating $\mathbf{F} = -\partial V / \partial \mathbf{R}$.

When writing eq. (26) in explicit terms we find the expression:

$$V \approx -m_{sat} (\mu_E / R) + 1/2 (\mu_E / R^3) \{ (3I^2 - 1)I_x + (3m^2 - 1)I_y + (3n^2 - 1)I_z - 6(lmI_{xy} + lnI_{xz} + mnI_{yz}) \} \quad (27)$$

In case when the spacecraft axes are selected to be the principal axes, the last term containing the products of inertia will vanish, and the result of eq. (27) can be simplified accordingly.

The results in eq. (26) and (27) can be shown to be identical with the expressions given in Hughes [4], eqs. (24) and (25), Ch. 8, p. 238 (apart from the signs of the cross-products of inertia which are defined with opposite signs by Hughes).

5. GRAVITY GRADIENT TORQUE

5.1 Expansion of Torque Expression

To calculate the torque exerted by the gravity force upon an extended satellite body, we start from the infinitesimal force acting upon the satellite mass element dm_{sat} (eq. 6), perform the cross-product with the lever arm $\boldsymbol{\rho}$ (i.e., the vector from the satellite center of mass to the mass element dm_{sat}), and perform the integration over the satellite body:

$$d\mathbf{T} = \boldsymbol{\rho} \times d\mathbf{F} \quad \rightarrow \quad \mathbf{T} = -\mu_E \int_{m_{sat}} \left\{ \frac{\boldsymbol{\rho} \times \mathbf{r}}{r^3} \right\} dm_{sat} \quad (28)$$

The integrand will be similar to the one in eq. (7) but with the final term $(\mathbf{R} + \boldsymbol{\rho})$ replaced by $\boldsymbol{\rho} \times \mathbf{r} = \boldsymbol{\rho} \times (\mathbf{R} + \boldsymbol{\rho}) = \boldsymbol{\rho} \times \mathbf{R}$. We write out all terms as in eqs. (7) and (8):

$$\begin{aligned} \mathbf{T} &= -\frac{\mu_E}{R^3} \int_{m_{sat}} \left\{ 1 - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})}{R^2} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2}{R^4} + O\left(\frac{\rho}{R} \right)^3 \right\} (\boldsymbol{\rho} \times \mathbf{R}) dm_{sat} = \\ &= -\frac{\mu_E}{R^3} \int_{m_{sat}} \left\{ (\boldsymbol{\rho} \times \mathbf{R}) - 3 \frac{(\mathbf{R} \cdot \boldsymbol{\rho})(\boldsymbol{\rho} \times \mathbf{R})}{R^2} - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 (\boldsymbol{\rho} \times \mathbf{R}) + \right. \\ &\quad \left. + \frac{15}{2} \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2 (\boldsymbol{\rho} \times \mathbf{R})}{R^4} + O\left(\frac{\rho}{R} \right)^3 (\boldsymbol{\rho} \times \mathbf{R}) \right\} dm_{sat} \end{aligned} \quad (29)$$

Because of the definition of the satellite center of mass (i.e., eqs. 10), we find that the first term in the integrand of eq. (29) vanishes after integration:

$$\int_{m_{sat}} \{(\boldsymbol{\rho} \times \mathbf{R})\} dm_{sat} = \left\{ \int_{m_{sat}} \boldsymbol{\rho} dm_{sat} \right\} \times \mathbf{R} = \mathbf{0} \quad (30)$$

The second term in the integrand can be reduced with the help of eq. (20c) as follows:

$$\begin{aligned} \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \cdot \boldsymbol{\rho})(\boldsymbol{\rho} \times \mathbf{R})}{R^2} \right\} dm_{sat} &= -\mathbf{R} \times \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \cdot \boldsymbol{\rho})\boldsymbol{\rho}}{R^2} \right\} dm_{sat} = \\ &= -\mathbf{R} \times \{ \frac{1}{2} \text{tr}[\mathbf{I}] \mathbf{R} - [\mathbf{I}]\mathbf{R} \} / R^2 = \mathbf{R} \times \{ [\mathbf{I}]\mathbf{R} \} / R^2 \end{aligned} \quad (31)$$

This result is a consequence of the fact that the vector $[\mathbf{I}]\mathbf{R}$ is in general not aligned with \mathbf{R} as can be seen from the last term in eq. (22) and mentioned in point 5 of section 3.5.

After expressing $[\mathbf{I}]\mathbf{R}$ in its vector components, we find:

$$\mathbf{R} \times \{[\mathbf{I}]\mathbf{R}\}/R^2 = \begin{pmatrix} mn(I_z - I_y) + ln I_{xy} - lm I_{xz} + (n^2 - m^2)I_{yz} \\ ln(I_x - I_z) + lm I_{yz} - mn I_{xy} + (l^2 - n^2)I_{xz} \\ lm(I_y - I_x) + mn I_{xz} - ln I_{yz} + (m^2 - l^2)I_{xy} \end{pmatrix} \quad (32)$$

The third term in the integrand of eq. (29) can be reduced similarly as was done in eq. (20e). It can be shown that it is smaller, by a factor of the order of k/R (with k the satellite radius of gyration), than the leading term given in eq. (31) and can thus be neglected:

$$\int_{m_{sat}} \left\{ \left(\frac{\rho}{R} \right)^2 (\boldsymbol{\rho} \times \mathbf{R}) \right\} dm_{sat} = O\left(\frac{k}{R}[\mathbf{I}]\right) \quad (\text{i.e., negligible}) \quad (33)$$

The fourth term in the integrand of eq. (29) can be shown to be of the same order of magnitude as the term in eq. (33) above and is thus also negligible:

$$\int_{m_{sat}} \left\{ \frac{(\mathbf{R} \cdot \boldsymbol{\rho})^2 (\boldsymbol{\rho} \times \mathbf{R})}{R^4} \right\} dm_{sat} = O\left(\frac{k}{R}[\mathbf{I}]\right) \quad (\text{i.e., negligible}) \quad (34)$$

The final expression for the gravity-gradient torque follows from eqs. (29), (31) and (32):

$$\begin{aligned} \mathbf{T} &\approx \frac{3\mu_E}{R^3} \int_{m_{sat}} \left\{ \frac{(\mathbf{R} \cdot \boldsymbol{\rho})(\boldsymbol{\rho} \times \mathbf{R})}{R^2} \right\} dm_{sat} = \frac{3\mu_E}{R^5} \{ \mathbf{R} \times [\mathbf{I}]\mathbf{R} \} = \frac{3\mu_E}{R^3} \{ \mathbf{u} \times [\mathbf{I}]\mathbf{u} \} = \\ &= \frac{3\mu_E}{R^3} \begin{pmatrix} mn(I_z - I_y) + ln I_{xy} - lm I_{xz} + (n^2 - m^2)I_{yz} \\ ln(I_x - I_z) + lm I_{yz} - mn I_{xy} + (l^2 - n^2)I_{xz} \\ lm(I_y - I_x) + mn I_{xz} - ln I_{yz} + (m^2 - l^2)I_{xy} \end{pmatrix} \end{aligned} \quad (35)$$

This result is in agreement with eq. (49) of [1] because the sign reversal of \mathbf{R} has no effect on the torque (because it cancels due to double appearances).

When comparing this result with eqs. (8) on p. 18 of NASA SP-8024, Ref. [8], full compatibility is achieved when recognizing that the signs of the inertia cross-products definitions in Ref. [8] are opposite to those adopted here in eqs. (14). The definition used in Ref. [8] is $I_{xy} = -\int \{xy\} dm$ as shown in Ref. [8], eq. (4) on p. 8. Hughes [4] neglects to specify the definition of his off-diagonal inertia products but there are strong indications that he also uses the alternative definitions (i.e., $I_{xy} = -\int \{xy\} dm$). This may be concluded since he mentions that his result in eq. (22) on p. 238 agrees with Ref. [8] eq. (4) on p. 8.



5.2 Interpretation of Torque Result

As mentioned already, almost all of the terms (except for the very last one) in the force expression of eq. (21) are directed along the direction from the center of attraction to the satellite's center of mass. These terms can *not* produce a torque on the satellite (because the lever arm of the force vanishes). The one exception is the last term in eq. (21), i.e. $-3[\mathbf{l}]\mathbf{u}$. It is this force term that is responsible for the so-called 'gravity-gradient' torque about the satellite's center of mass as shown in eqs. (31) and (35) above.

By calculating the dot-product of the vector $\mathbf{R} = R(l, m, n)$ with the torque expression in eq. (35), it can easily be found that $(\mathbf{T} \bullet \mathbf{R})$ vanishes. Thus, the gravity gradient torque points always in a direction normal to the instantaneous orbital radius. It is of interest to note that this property originates from the nature of the gravity field which is symmetrical about the orbital radius, at least under the present assumption that the Earth is assumed to be a homogeneous sphere. It is an interesting exercise to check that this property does indeed not hold anymore when zonal harmonic terms are included as was done in [7].

If the 'Body Principal Reference Frame' (BPRF) frame is selected as the reference frame, the products of inertia terms vanish and eq. (35) gives the result that was most likely first published by Beletskii [3], p. 9:

$$\mathbf{T} \approx 3 \frac{\mu_E}{R^3} \begin{pmatrix} mn(I_z - I_y) \\ ln(I_x - I_z) \\ lm(I_y - I_x) \end{pmatrix} \quad (36)$$

It can be shown that, if two principal moments of inertia are identical, the gravity-gradient torque about the third axis vanishes (e.g., $T_z = 0$ when $I_x = I_y$). Furthermore, the gravity gradient torque vanishes completely in the following situations:

1. if *all* principal moments of inertia are identical (i.e., $I_x = I_y = I_z$), for instance in the case of a spherical satellite.
2. if two principal moments of inertia are identical and the third direction cosine vanishes, e.g. a symmetrical satellite (i.e., $I_x = I_y$) with its z -axis pointing along the orbit normal (i.e., $n = 0$) and the orientations of the x , y axes can be arbitrary. Note that this property holds even when Earth oblateness is taken into account, Hughes [4], p. 243.
3. for a tri-axial satellite ($I_z \neq I_y \neq I_x$), the torque vanishes if and only if 2 of the 3 direction cosines (of the \mathbf{R} vector) vanish; this implies that one of the principal axes must be pointing along the orbital radius (for instance, $l = 1, m = n = 0$). These observations are compatible with those in Hughes, Ref. [4], p. 238.

It may also be mentioned that a sufficient condition for ensuring the stability of the attitude pointing along the local vertical is given by Beletskii [3], pp. 30-31: $I_z > I_y > I_x$,

i.e. the principal axis of inertia along the local vertical should be smallest and the one normal to the orbit should be largest.

For a satellite in a three-axis stabilization mode, with its principal x_p and y_p axes within the orbital plane pointing in a fixed inertial direction, we have $l = \cos(\omega t)$, $m = \sin(\omega t)$; $n = 0$ which means that only the torque component normal to the orbit plane remains:

$$\mathbf{T} \approx (3/2) \frac{\mu_E}{R^3} \begin{pmatrix} 0 \\ 0 \\ \sin(2\omega t)(I_y - I_x) \end{pmatrix} \quad (37)$$

Thus, there is only a torque component along the z -axis and it varies periodically at a frequency of twice the orbital rate ω . In the case of a symmetrical cylindrical satellite with $I_x = I_y$, also the third torque component vanishes.

5.3 Alternative Torque Derivation

It is pointed out by Pelivan [7] that the gravity-gradient torque may also be calculated by means of the gravity-gradient matrix, which is formed by the second-order partial derivatives of the gravity potential.

This approach considers an arbitrary point P within the rigid-body satellite at a distance $\mathbf{r} = \mathbf{R} + \mathbf{p}$ from the center of attraction (Figure 1). The second-order expansion of the gravity potential function is performed around the reference point, which normally corresponds to the satellite's center of mass. This leads to the following approximate gravity potential acting at the location P, expressed in quantities belonging to the satellite center of mass at the distance \mathbf{R} from the center of the Earth (see also [6], p. 128):

$$V \approx V_c + (\mathbf{r} - \mathbf{R})^T \cdot \nabla V_c - 1/2 (\mathbf{r} - \mathbf{R})^T [G_c] (\mathbf{r} - \mathbf{R}) \quad (38)$$

The nabla operator ∇ stands for the vector of partial derivatives $\partial/\partial\mathbf{r} = (\partial/\partial r_1, \partial/\partial r_2, \partial/\partial r_3)^T$ with r_j ($j = 1, 2, 3$) denoting the components of the gradient of \mathbf{r} in the inertial (X, Y, Z) reference frame, Figure 1. The vector ∇V_c represents $\partial/\partial\mathbf{r}$ evaluated at the location of the center of mass. The so-called gravity-gradient matrix $[G_c]$ appearing in eq. (38) is defined by the partial derivatives of ∇V_c at the center of mass position \mathbf{R} :

$$[G_c] = -[\nabla(\nabla V_c)] \quad \text{or:} \quad [G_c]_{jk} = -\partial^2 V_c / (\partial R_j \partial R_k) \quad (39)$$

The minus sign has been introduced in order that the final resulting expressions are compatible with those presented by Gottlieb, Ref. [9].



The matrix elements $[G_c]_{jk}$ have been calculated by Wertz [6], pp. 128-129, for a gravity potential that includes the oblateness (J_2) terms, see also [7], eqs. (41), but his sign convention is opposite to what is taken in Ref. [9], see also remarks below eq. (40).

The gravitational acceleration \mathbf{f} (i.e., gravity force per unit of mass) acting at the point P can now be expressed in terms of the acceleration acting at the satellite center of mass by means of the result in eq. (38):

$$\mathbf{f} = -\partial V/\partial \mathbf{r} = -\nabla V \approx -\nabla V_c + [G_c] (\mathbf{r} - \mathbf{R}) = \mathbf{f}_c + [G_c] (\mathbf{r} - \mathbf{R}) \quad (40)$$

It should be noted that the minus-sign introduced in the definition in eq. (39) has resulted in the expected plus sign in the Taylor expansion of the gravity field in eq. (40). This makes eq. (40) fully consistent with eq. (8-18) on p. 21 of Gottlieb [9].

On the other hand, it should be noted that the first-order expansion of the gravity force in Wertz [6], eq. (5-24) on p. 128, employs the opposite definition of $[G_c]$, namely (in our notation): $\mathbf{f} = \mathbf{f}_c + [G_c] (\mathbf{R} - \mathbf{r})$. This implies that Wertz defines his gravity gradient matrix $[G_c]$ in eq. (39) as $+\nabla(\nabla V_c)$ as shown in his eq. (5-26). This selection is the opposite of the definitions used in our eq. (40) and in the expansion of eq. (8-18) in Gottlieb [9].

To proceed further, it must be recognized that, in the present formulation, the vectors \mathbf{f} , \mathbf{f}_c , \mathbf{R} , and \mathbf{r} refer to an inertial reference frame fixed to the Earth. This is convenient if also the zonal and tesseral harmonics of the gravity field needs to be incorporated in the potential function. Thus, also the vector $\mathbf{r} - \mathbf{R}$ should be considered within the inertial frame. The coordinate transformation from the body frame vector $\boldsymbol{\rho} = (x, y, z)^T$ to the inertial reference frame vector $\mathbf{r} - \mathbf{R}$ depends on the spacecraft attitude orientation and may be written as:

$$\mathbf{r} - \mathbf{R} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = [B] \boldsymbol{\rho} \quad (41)$$

with $b_{jk} = (\mathbf{x}_j \bullet \mathbf{X}_k)$ and \mathbf{x}_j and \mathbf{X}_k denoting the unit-vectors along the reference axes of the inertial (X, Y, Z) and body (x, y, z) reference frames.

Because the inertias can be most conveniently calculated within the body-centered reference frame BCRF, we transform eq. (40) to the body reference frame and perform the integration of the force over the satellite body:

$$\mathbf{F}_b = [B]^T \int_{m_{sat}} \{\mathbf{f}_c + [G_c](\mathbf{r} - \mathbf{R})\} dm_{sat} = \int_{m_{sat}} \{[B]^T \mathbf{f}_c + [B]^T [G_c] [B] \boldsymbol{\rho}\} dm_{sat} \quad (42)$$

with the transformed gravity matrix $[G_b] = [B]^T [G_c][B]$ within the body reference frame where the gravity gradient matrix $[G_c]$ must be evaluated within the Earth-equatorial centered inertial reference frame as was done in Wertz [6], pp. 127-128.

The gravity torque within the body frame follows now as:

$$\mathbf{T} = \int_{m_{sat}} \boldsymbol{\rho} \times \{ [B]^T \mathbf{f}_c + [G_b] \boldsymbol{\rho} \} dm_{sat} = \int_{m_{sat}} \{ \boldsymbol{\rho} \times [G_b] \boldsymbol{\rho} \} dm_{sat} \quad (43)$$

Here, the first term has vanished because of the center of mass definition in eq. (10).

We can readily evaluate the cross-product within the integrand of eq. (43):

$$\text{'integrand'} = \begin{pmatrix} g_{23}(y^2 - z^2) + (g_{33} - g_{22})yz + g_{13}xy - g_{12}xz \\ g_{13}(z^2 - x^2) + (g_{11} - g_{33})xz + g_{12}yz - g_{23}xy \\ g_{12}(x^2 - y^2) + (g_{22} - g_{11})xy + g_{23}xz - g_{13}yz \end{pmatrix} \quad (44)$$

The coefficients g_{jk} appearing here are the elements of the matrix $[G_b]$. When integrating eq. (43) using the definitions of the moments of inertia in eqs. (14) we find:

$$\mathbf{T} = \begin{pmatrix} g_{23}(I_z - I_y) + (g_{33} - g_{22})I_{yz} + g_{13}I_{xy} - g_{12}I_{xz} \\ g_{13}(I_x - I_z) + (g_{11} - g_{33})I_{xz} + g_{12}I_{yz} - g_{23}I_{xy} \\ g_{12}(I_y - I_x) + (g_{22} - g_{11})I_{xy} + g_{23}I_{xz} - g_{13}I_{yz} \end{pmatrix} \quad (45)$$

This result is fully identical to eq. (8-36) on p. 24 of Ref. [9]. Note also that the definitions of the cross-products of inertia in eqs (8-17) of Ref. [9] are consistent with the ones that have been adopted here in eqs. (14).

6. POSITION of CENTER of GRAVITY

6.1 Assessment of [1]

We consider a set of n mass particles m_j for $j = 1, \dots, n$ at individual distances \mathbf{r}_j from the center of attraction (i.e., Earth in the present case). The location of the center of gravity \mathbf{c} (relative to the center of mass) represents the ‘weighted average position’ of the gravity force and is defined in Ref. [1], eq. (32) in a similar manner as the center of mass, namely as the point \mathbf{c} at which:

$$(\mathbf{c} \bullet \mathbf{F}) = \sum_{j=1}^n (\mathbf{r}_j \bullet \mathbf{F}_j) \quad \text{with: } \mathbf{F} = \sum_{j=1}^n (\mathbf{F}_j) \quad (46)$$

In the case of a rigid body, the number n of infinitesimal particles is taken in the limit for $n \rightarrow \infty$. After substituting the force expression of eq. (6), and integrating over the mass within the rigid body, we find with $\mathbf{r} = \mathbf{R} + \boldsymbol{\rho}$:

$$(\mathbf{c} \bullet \mathbf{F}) = \int_{m_{sat}} (\mathbf{r} \bullet d\mathbf{F}) = -\mu_E \left\{ \int_{m_{sat}} \frac{(\mathbf{r} \bullet \mathbf{r})}{r^3} dm_{sat} \right\} = -\mu_E \left\{ \int_{m_{sat}} \frac{1}{|\mathbf{R} + \boldsymbol{\rho}|} dm_{sat} \right\} \quad (47)$$

The subsequent analysis in [1] leads to a reduction of this result to [1], eq. (35), which states (**apart from the familiar sign reversal**):

$$\mathbf{c} = -\mu_E \left\{ \int_{m_{sat}} \frac{1}{|\mathbf{R} + \boldsymbol{\rho}|} dm_{sat} \right\} \mathbf{F} / |\mathbf{F}|^2 \quad (48)$$

By virtue of the definition of $(\mathbf{c} \bullet \mathbf{F})$ given in eq. (47), the result of eq. (48) simply states that $\mathbf{c} = (\mathbf{c} \bullet \mathbf{F}) \mathbf{F} / |\mathbf{F}|^2$.

At first sight, this result may appear to be equivalent to eqs. (46) and (47). It should be recognized, however, that eq. (48) actually constrains the (assumed arbitrary) vector \mathbf{c} to be pointing along the direction of the force vector \mathbf{F} . In other words, the vector \mathbf{c} resulting from the definition in eq. (48) has been assumed to be pointing along the direction of the resulting force vector \mathbf{F} .

The more general (and physically more meaningful) definition of \mathbf{c} should have three (in the general three-dimensional case) or at least two (for a planar configuration) components without any a priori constraints on its direction. One component may be along \mathbf{F} and the other in a direction \mathbf{N} normal to \mathbf{F} according to the general vector projection relationship:

$$\mathbf{c} = (\mathbf{c} \cdot \mathbf{F})\mathbf{F} / |\mathbf{F}|^2 + (\mathbf{c} \cdot \mathbf{N})\mathbf{N} / |\mathbf{N}|^2 \quad (49)$$

When employing eq. (48) as was done in [1], the second term in eq. (49) has been ignored. This means that one makes the *implicit assumption* that the vector \mathbf{c} points along the force vector \mathbf{F} and that the component along \mathbf{N} is absent. This assumption is certainly true in many special cases when indeed the vector \mathbf{c} is along the resulting force vector, e.g. for a dumbbell satellite oriented along the local vertical or local horizontal.

In general, however, this result is not correct because the position of the center of gravity can have a component away from the force vector as will be demonstrated in Appendix A for the dumbbell satellite when it is oriented in a direction away from the local vertical or local horizontal direction.

Therefore, it is recommended to *not* force the position of the center of gravity to lie on the total force vector a priori but to adopt a more general starting point and to let the analysis determine where the point is actually located.

6.2 Alternative Definition of Center of Gravity

It is unfortunate that a discussion of the center of gravity definition could not be found in the literature. Because it is felt that the center of gravity definition mentioned above is too constraining, we adopt a different approach here. We postulate the following plausible working definition for the ‘center of gravity’ as the location at which:

“the gravity force acting upon a hypothetical point-mass (with mass equal to the total body mass) is identical to the total gravity force acting upon the extended rigid body”.

The position of the center of gravity relative to the center of mass within the BCRF frame is denoted by \mathbf{p}_{cg} so we have $\mathbf{r}_{cg} = \mathbf{R} + \mathbf{p}_{cg}$. The force \mathbf{F}_{cg} acting upon a point-mass at the (so far unknown) location of the centre of gravity is described by the leading monopole term of eq. (21) and can be expanded using eq. (4):

$$\begin{aligned} \mathbf{F}_{cg} &= -m_{sat}\mu_E \frac{\mathbf{R} + \mathbf{p}_{cg}}{|\mathbf{R} + \mathbf{p}_{cg}|^3} \approx -m_{sat} \frac{\mu_E}{R^3} \left\{ 1 - 3 \frac{(\mathbf{R} \cdot \mathbf{p}_{cg})}{R^2} - \frac{3}{2} \left(\frac{\rho_{cg}}{R} \right)^2 + \frac{15}{2} \frac{(\mathbf{R} \cdot \mathbf{p}_{cg})^2}{R^4} \right\} (\mathbf{R} + \mathbf{p}_{cg}) = \\ &= -m_{sat} \frac{\mu_E}{R^3} \left\{ \mathbf{R} + \mathbf{p}_{cg} - 3 \frac{(\mathbf{R} \cdot \mathbf{p}_{cg})(\mathbf{R} + \mathbf{p}_{cg})}{R^2} - \frac{3}{2} \left(\frac{\rho_{cg}}{R} \right)^2 \mathbf{R} + \frac{15}{2} \frac{(\mathbf{R} \cdot \mathbf{p}_{cg})^2}{R^4} \mathbf{R} + O\left(\frac{\rho_{cg}}{R}\right)^3 \mathbf{R} \right\} \quad (50) \end{aligned}$$

When equating this expression with the actual force acting on the rigid body given in eq. (21) we see that the leading monopole terms $-m_{sat} (\mu_E / R^3) \mathbf{R}$ are identical (as expected).

The next largest contributions within the brackets { ... } of eq. (50) are the terms of order ρ_{cg} , i.e.:

$$-m_{sat} \frac{\mu_E}{R^3} \left\{ \rho_{cg} - 3 \frac{(\mathbf{R} \bullet \rho_{cg}) \mathbf{R}}{R^2} \right\} \quad (51)$$

It should be noted that the remaining terms within the brackets { ... } of eq. (50) are of second, i.e. $(\rho_{cg}/R)^2 \mathbf{R}$, or higher orders and are considered negligible at the level of accuracy considered here.

The expression in eq. (51) should be equated to the $(3/2) (\mu_E / R^4)$ { ... } term in eq. (21). When writing $\mathbf{R} = R \mathbf{u}$ we find the following equation for ρ_{cg} :

$$\rho_{cg} - 3(\mathbf{u} \bullet \rho_{cg}) \mathbf{u} = - \frac{3}{2Rm_{sat}} \left\{ 5(\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} - tr[\mathbf{I}] \mathbf{u} - 2[\mathbf{I}] \mathbf{u} \right\} = - \frac{3}{2Rm_{sat}} \mathbf{U} \quad (52)$$

The vector \mathbf{U} is defined by the expression { ... }, see also Appendix D, eq. (D.1) and (D.11) where \mathbf{U} is expanded in terms of its components along the body and orbit reference axes, respectively.

For the present objective, it is most convenient to adopt the orbital reference frame. We expand the unknown ρ_{cg} in components along the (u, v, w) axes of the orbital reference frame (ORF), with u along the orbital radius, v along the local horizontal and w along the orbit-normal directions:

$$\rho_{cg} = c_u \mathbf{u} + c_v \mathbf{v} + c_w \mathbf{w} = \begin{pmatrix} c_u \\ c_v \\ c_w \end{pmatrix} \quad (53)$$

On the basis of the result established in eq. (D.10) of Appendix D, we find the following expressions for the three components of ρ_{cg} from the equality in eq. (52):

$$\rho_{cg} - 3(\mathbf{u} \bullet \rho_{cg}) \mathbf{u} = \begin{pmatrix} -2c_u \\ c_v \\ c_w \end{pmatrix} = - \frac{3}{2Rm_{sat}} \mathbf{U}_{ORF} = - \frac{3}{2Rm_{sat}} \begin{pmatrix} 2\hat{I}_x - \hat{I}_y - \hat{I}_z \\ 2\hat{I}_{xy} \\ 2\hat{I}_{xz} \end{pmatrix} \quad (54)$$

When assuming that the body frame is aligned with the principal axes, we find for the components of the center of gravity from eq. (D.11):



$$c_u = \frac{3}{4Rm_{sat}} (2\hat{I}_x - \hat{I}_y - \hat{I}_z) = \frac{3}{4Rm_{sat}} \{ 3(a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z) - (I_x + I_y + I_z) \} \quad (55a)$$

$$c_v = -\frac{3}{Rm_{sat}} \hat{I}_{xy} = \frac{3}{Rm_{sat}} (a_{11}a_{12} I_x + a_{21}a_{22} I_y + a_{31}a_{32} I_z) \quad (55b)$$

$$c_w = -\frac{3}{Rm_{sat}} \hat{I}_{xz} = \frac{3}{Rm_{sat}} (a_{11}a_{13} I_x + a_{21}a_{23} I_y + a_{31}a_{33} I_z) \quad (55c)$$

In the *special* case when the satellite's principal axes are aligned with the orbital frame, the coefficients a_{jk} are identical to 1 for $j = k$ and vanish for $j \neq k$, so we find:

$$c_u = \frac{3}{4Rm_{sat}} (2I_x - I_y - I_z); \quad c_v = c_w = 0 \quad (56)$$


This result implies that the center of gravity lies on the orbital radius vector in this special case. It may be recalled from eq. (22) and point 5 of section 3.5 that, in this special case, the products of inertia are absent and the gravity force is aligned with the orbital radius vector so that the position of the center of gravity lies on the orbital radius vector as well as on the force vector.

The distance of the center of gravity from the center of mass (which itself is at the distance R from the center of the Earth) is of the order of:

$$c_u \approx (3/2) \Delta I (k/R)^2 R \approx 3 \times 10^{-14} R (\Delta I) \quad (57)$$

The inertia parameter ΔI is defined as $(2I_x - I_y - I_z)/(2I_x)$, which vanishes for a 'spherical' satellite, and k is the radius of gyration of I_x . For a typical satellite with a radius of gyration < 1 m and $\Delta I < 0.1$, we find that c_u (i.e., the distance of the center of gravity from the center of mass) is at most of the order of 20 nano-meter.

In general, when the satellite's principal axes are not along the orbital axes, the center of gravity will not lie on the orbital radius vector as can be seen from eqs. (55).

	<p>Model of Gravity Force and Gravity Gradient Torque</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-004 Issue: 1.0 Page: 24 of 45</p>
---	---	--

References

- [1] Michel Silas Guilherme, **Gravity Force in Cartesian Coordinate System**, INT-DYN-TN-ZAR-001, Issue 2, 5 Oct. 2005.
- [2] Leonard Meirovitch, **Methods of Analytical Dynamics**, McGraw-Hill Book Company, New York, USA, 1970.
- [3] V. V. Beletskii, **Motion of an Artificial Satellite about its Center of Mass**, Published in Moscow, 1965 (Israel Program for Scientific Translation, Jerusalem, 1966).
- [4] Peter C. Hughes, **Spacecraft Attitude Dynamics**, John Wiley & Sons, New York, USA, 1986.
- [5] Marshall H. Kaplan, **Modern Spacecraft Dynamics & Control**, John Wiley & Sons, New York, USA, 1976.
- [6] James R. Wertz (Editor), **Spacecraft Attitude Determination and Control**, Kluwer, Dordrecht, the Netherlands, 1978.
- [7] Ivanka Pelivan, **Implementation of Quadrupole Effects on Satellite and Test Masses in the Drag-Free Simulator**, FLK-SIM-TN-ZAR-002, Issue 1.1, April 12, 2005.
- [8] NASA, **Spacecraft Gravitational Torques**, NASA-SP-8024, May 1969.
- [9] Robert G. Gottlieb, **Fast Gravity, Gravity Partial, Normalized Gravity, Gradient Gravity Torque and Magnetic Field: Derivation and Codes**, NASA-CR-188243, McDonnell Douglas Space Systems – Houston Division, February 1993.

APPENDIX A. DUMBBELL SATELLITE ILLUSTRATION

We will now illustrate and interpret the results established in the preceding Chapters using the relatively straightforward example of the so-called ‘dumbbell’ satellite.

A.1 Geometry

Figure 2 illustrates the geometry of a dumbbell satellite orbiting an ideal spherical Earth.

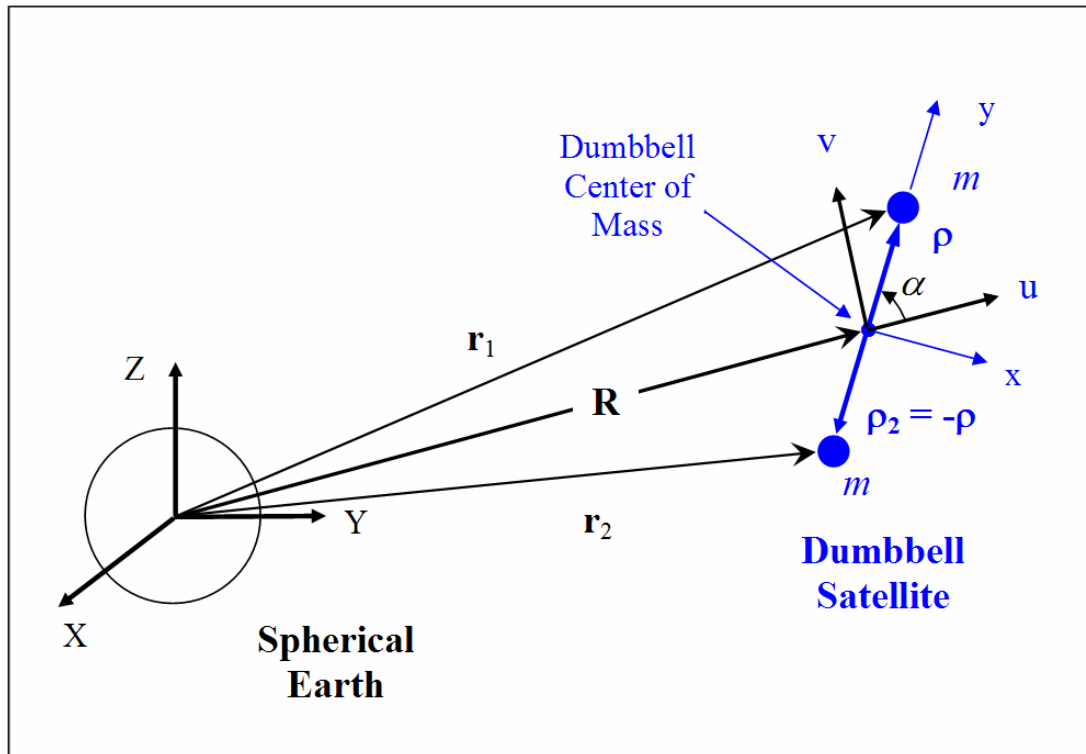


Figure 2 – Illustration of (Planar) Geometry of Earth and Dumbbell Satellite

For simplicity, the dumbbell satellite is assumed to move only within its orbital plane at all times. Therefore, the attitude is constrained to arbitrary in-plane librations defined by the angle α relative to the orbital radius (Figure 2). We introduce the dumbbell’s body-centered reference frame BCRF (x, y, z) . The x -axis points normal to the (mass-less) rod connecting the two point masses (with equal masses m) and lies within the orbit plane. The y -axis is directed along the rod and lies also in the orbit plane. The z -axis (not shown) is normal to the orbit plane. Also we introduce the local orbit frame ORF (u, v, w) with the u -axis along the instantaneous orbital radius vector, the v -axis along the local horizontal direction, and the w -axis is identical to the z -axis.

The following coordinate transformation holds between the body-centered and the orbital reference frames (i.e., BCRF and ORF):

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} \sin \alpha & \cos \alpha & 0 \\ -\cos \alpha & \sin \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (\text{A.1})$$

When comparing this transformation with eq. (D.2) in Appendix D, we can readily calculate the corresponding a_{jk} coefficients (e.g., $a_{12} = -\cos \alpha$).

Figure 2 shows that the orbital radius is oriented along the u axis, i.e. $\mathbf{R} = R \mathbf{u}$. (Note that \mathbf{u} , \mathbf{v} , \mathbf{w} and \mathbf{x} , \mathbf{y} , \mathbf{z} denote the unit-vectors along the associated reference axes.) The position vectors of the two point-masses of the dumbbell are given by $\boldsymbol{\rho}_1 = \boldsymbol{\rho} = \rho \mathbf{y}$ and $\boldsymbol{\rho}_2 = -\boldsymbol{\rho} = -\rho \mathbf{y}$. The inertial position vectors \mathbf{r}_1 and \mathbf{r}_2 shown in Figure 2 can now be expressed in the (x, y, z) body coordinates:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{R} + \boldsymbol{\rho}_1 = R\{(\sin \alpha) \mathbf{x} + (\cos \alpha + \rho/R) \mathbf{y}\} = R\{(1 + \rho/R \cos \alpha) \mathbf{u} + \rho/R (\sin \alpha) \mathbf{v}\} \\ \mathbf{r}_2 &= \mathbf{R} + \boldsymbol{\rho}_2 = R\{(\sin \alpha) \mathbf{x} + (\cos \alpha - \rho/R) \mathbf{y}\} = R\{(1 - \rho/R \cos \alpha) \mathbf{u} - \rho/R (\sin \alpha) \mathbf{v}\} \end{aligned} \quad (\text{A.2})$$

A.2 Expansion of Gravity Force

The combined force on the two masses of the dumbbell satellite is given by (see eq. 6):

$$\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2 = -\mu_E \sum_{j=1}^2 \left\{ m_j \frac{\mathbf{r}_j}{r_j^3} \right\} = -\mu_E \sum_{j=1}^2 \left\{ m_j \frac{(\mathbf{R} + \boldsymbol{\rho}_j)}{r_j^3} \right\} \quad (\text{A.3})$$

On the basis of the geometry presented above, and using $(\mathbf{R} \cdot \boldsymbol{\rho}) = \rho R \cos \alpha$, we can expand the r_j^{-3} , ($j = 1, 2$) terms in a similar way as was done in eq. (4):

$$\begin{aligned} r_1^{-3} &= |\mathbf{R} + \boldsymbol{\rho}_1|^{-3} = R^{-3} \left\{ 1 + \frac{2\rho}{R} \cos \alpha + \left(\frac{\rho}{R} \right)^2 \right\}^{-3/2} \approx R^{-3} \left\{ 1 - \frac{3\rho}{R} \cos \alpha - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \left(\frac{\rho}{R} \right)^2 \cos^2 \alpha \right\} \\ r_2^{-3} &= |\mathbf{R} + \boldsymbol{\rho}_2|^{-3} = |\mathbf{R} - \boldsymbol{\rho}|^{-3} \approx R^{-3} \left\{ 1 + \frac{3\rho}{R} \cos \alpha - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \left(\frac{\rho}{R} \right)^2 \cos^2 \alpha \right\} \end{aligned} \quad (\text{A.4})$$

We find the following result for the combined force on the two masses of equal mass m up to terms of second order in ρ/R :

$$\begin{aligned}
 \mathbf{F} &\approx -\frac{\mu_E}{R^3} m \left\{ 1 - 3 \left(\frac{\rho \cos \alpha}{R} \right) - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \left(\frac{\rho \cos \alpha}{R} \right)^2 \right\} (\mathbf{R} + \boldsymbol{\rho}) + \\
 &\quad -\frac{\mu_E}{R^3} m \left\{ 1 + 3 \left(\frac{\rho \cos \alpha}{R} \right) - \frac{3}{2} \left(\frac{\rho}{R} \right)^2 + \frac{15}{2} \left(\frac{\rho \cos \alpha}{R} \right)^2 \right\} (\mathbf{R} - \boldsymbol{\rho}) = \\
 &= -2m \frac{\mu_E}{R^3} \mathbf{R} + 6m \frac{\mu_E}{R^3} \left(\frac{\rho \cos \alpha}{R} \right) \boldsymbol{\rho} + 3m \frac{\mu_E}{R^3} \left(\frac{\rho}{R} \right)^2 (1 - 5 \cos^2 \alpha) \mathbf{R} \quad (\text{A.5})
 \end{aligned}$$

This force expression can be transformed to the (u, v, w) frame by expanding $\boldsymbol{\rho}$ in its components along the u - and v -axes, i.e. $\boldsymbol{\rho} = \rho \mathbf{y} = \rho (\cos \alpha \mathbf{u} + \sin \alpha \mathbf{v})$ and $\mathbf{R} = R\mathbf{u}$:

$$\mathbf{F} \approx -2m \frac{\mu_E}{R^2} \mathbf{u} + 3m \frac{\mu_E \rho^2}{R^4} \{ (1 - 3 \cos^2 \alpha) \mathbf{u} + \sin(2\alpha) \mathbf{v} \} \quad (\text{A.6})$$

The leading term $-2m(\mu_E/R^2)$ is produced by the ‘ideal’ inverse-square central force, whereas the second term, which is proportional to $(\mu_E \rho^2/R^4)$, can be considered a perturbing force caused by the separation between the two dumbbell masses within the gravity field. It follows that the magnitude of the perturbing part is of the order of $3/2(\rho/R)^2$ times the leading central force (this amounts to a factor of about 2×10^{-7}).

Finally, we may also write the force result in its components along the body-centered (x, y, z) coordinates using the transformation of \mathbf{R} in eq. (A.1) as shown in Figure 2:

$$\mathbf{F} \approx -2m \frac{\mu_E}{R^2} \mathbf{u} + 3m \frac{\mu_E \rho^2}{R^4} \{ \sin \alpha (1 - 5 \cos^2 \alpha) \mathbf{x} + \cos \alpha (3 - 5 \cos^2 \alpha) \mathbf{y} \} \quad (\text{A.7})$$

When the dumbbell is aligned with the orbital radius or local vertical (i.e., when $\alpha = 0^\circ$), the x -component (normal to the dumbbell axis) vanishes so the force acts in the direction of the orbital radius, i.e. along the unit-vector $\mathbf{y} = \mathbf{u}$. In the case when the dumbbell is aligned normal to the orbital radius ($\alpha = 90^\circ$), the y -component vanishes so the force is again aligned with the orbital radius which is now along the unit-vector $\mathbf{x} = \mathbf{u}$. In both situations, the force will have a small perturbing component (on top of the ideal inverse-square central force) directed along the orbital radius with unit-vector \mathbf{u} as can be seen from eq. (A.6):

$$\text{for } \alpha = 0^\circ : -6m \frac{\mu_E \rho^2}{R^4} \mathbf{u}; \quad \text{for } \alpha = \pm 90^\circ : 3m \frac{\mu_E \rho^2}{R^4} \mathbf{u} \quad (\text{A.8})$$



In the former case ($\alpha = 0^\circ$), the perturbing force augments the negative central force because the mass-point that is closest to the Earth is subjected to a progressively stronger force than the mass that is furthest away. On the other hand, in the latter case $\alpha = 90^\circ$, the positions of the two masses are at the same distance but further away from the Earth than the center of mass so the perturbing force reduces the negative ideal central force.

Although the position of the effective center of gravity is away from the center of mass in the special cases considered, the lever arm of the gravity gradient torque points along the force direction so the gravity gradient torque will vanish in both cases.

An example of a case where the center of gravity does not lie on the orbital radius is given by $\alpha = 45^\circ$. The components of the perturbing force follow from eq. (A.6):

$$\text{for } \alpha = 45^\circ : \quad 3m \frac{\mu_E \rho^2}{R^4} \{-1/2 \mathbf{u} + \mathbf{v}\} \quad (\text{A.9})$$

In this case, the sum of the individual forces acting on the two point-masses has a non-radial component because of the asymmetrical geometry of the dumbbell relative to the Earth. Thus, the resulting gravity force does not pass through the Earth's center.

A.3 Result of Gravity Force Model

The force results in eqs. (A.6-7) have been obtained directly from first principles. They can of course also be established by means of the formal force model presented in eq. (21). As a first step, we use eq. (15c) to calculate:

$$I_x = 2m\rho^2 ; \quad I_y = 0 ; \quad I_z = 2m\rho^2 \quad (\text{A.10})$$

Thus, we have:

$$[\mathbf{I}] = 2m\rho^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; \quad \text{tr}[\mathbf{I}] = 4m\rho^2 \quad (\text{A.11})$$

When using the results above and the explicit BCRF components for $\mathbf{R} = R \mathbf{u}$ given in eq. (A.1), the terms 1 to 3 of eqs. (24) can be simplified considerably:

$$\text{Term 1:} \quad (\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} = 2m\rho^2 \sin^2 \alpha (\sin \alpha \mathbf{x} + \cos \alpha \mathbf{y}) \quad (\text{A.12a})$$

$$\text{Term 2:} \quad \text{tr}[\mathbf{I}] \mathbf{u} = 4m\rho^2 (\sin \alpha \mathbf{x} + \cos \alpha \mathbf{y}) \quad (\text{A.12b})$$

$$\text{Term 3:} \quad [\mathbf{I}] \mathbf{u} = 2m\rho^2 \sin \alpha \mathbf{x} \quad (\text{A.12c})$$

With the help of these expressions the force expression in eq. (21) can be shown to be identical to the result given in eq. (A.7):

$$\mathbf{F} \approx -2m \frac{\mu_E}{R^2} \mathbf{u} + 3m \frac{\mu_E \rho^2}{R^4} \left\{ \sin \alpha (1 - 5 \cos^2 \alpha) \mathbf{x} + \cos \alpha (3 - 5 \cos^2 \alpha) \mathbf{y} \right\} \quad (\text{A.13})$$

This illustrates the agreement between the results derived by the two approaches and confirms the consistency of the proposed model (at least for this particular example).

It can also readily be confirmed that the expression in eq. (A.6) is fully consistent with the force components obtained when using eq. (D.11) in Appendix D.

A.4 Center of Gravity

The results for the center of gravity position presented in eqs. (45) can easily be applied to the dumbbell satellite using the transformation matrix of eq. (A1) and we find:

$$c_u = \frac{3}{4} \frac{\rho^2}{R} (1 - 3 \cos^2 \alpha) ; \quad c_v = -\frac{3}{2} \frac{\rho^2}{R} \sin(2\alpha) ; \quad c_w = 0 \quad (\text{A.14})$$

It is of interest to calculate the distance σ and the phase angle χ of the center of gravity location relative to the center of mass, i.e.

$$\chi = \arctan \left\{ \frac{c_v}{c_u} \right\} = \arctan \left\{ \frac{-2 \sin(2\alpha)}{1 - 3 \cos^2 \alpha} \right\} ; \quad \sigma = \sqrt{c_u^2 + c_v^2} \quad (\text{A.15})$$

The components c_u and c_v of the center of gravity position as well as the phase angle χ are summarized for a number of special cases in Table 1.

Table 1 shows that the center of gravity is in general (except for $\alpha = 0$ or 90°) located away from the orbital radius (i.e., when $c_v \neq 0$). The distance of the CoG position relative to the center of mass varies between -1.5 and 0.75 times $(\rho/R)\rho$. The largest value occurs at $\alpha = 0$ and 180° and the minimum at 90° . The reversal of sign of c_u at $\alpha = 54.7^\circ$ is due to the fact that the center of gravity passes under the center of mass at this point.

Finally, we present the components of the CoG position within the body x, y reference frame with the help of the transformation matrix in eq. (A.1):

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = \begin{bmatrix} \sin \alpha & -\cos \alpha \\ \cos \alpha & \sin \alpha \end{bmatrix} \begin{pmatrix} c_u \\ c_v \end{pmatrix} = \frac{3}{4} \frac{\rho^2}{R} \begin{pmatrix} \sin \alpha (\cos^2 \alpha + 1) \\ \cos \alpha (\cos^2 \alpha - 3) \end{pmatrix} \quad (\text{A.16})$$

Dumbbell Angle α	Components of CoG Location			
	$(R/\rho^2) c_u$	$(R/\rho^2) c_v$	σ	χ
0	-3/2	0	1.5	0
30°	-15/16	$-\frac{3}{4}\sqrt{3}$	1.602	-126°
45°	-3/8	-1.5	1.546	-104°
54.7°	0	$-\sqrt{2}$	1.414	-90°
60°	3/16	$-\frac{3}{4}\sqrt{3}$	1.312	-82°
90°	$\frac{3}{4}$	0	0.75	0

Table 1 – Summary of Center of Gravity Components

A.5 Gravity-Gradient Torque

The gravity-gradient torque induced by the gravity force can now be calculated in explicit form for the dumbbell satellite using the general torque expression in eq. (36):

$$\mathbf{T} \approx \frac{3\mu_E}{R^3} \{ \mathbf{u} \times [\mathbf{I}] \mathbf{u} \} = -3m \frac{\mu_E}{R^3} \rho^2 \sin(2\alpha) \mathbf{z} \quad (\text{A.17})$$

This result shows that the dumbbell will move towards the local vertical direction under the gravity gradient torque (this is the stable position). Eq. (A.17) also shows that the torque vanishes when $\alpha = 0^\circ$ and $\pm 90^\circ$ as already mentioned in the previous section. The maximum possible torque occurs when $\alpha = 45^\circ$ and 135° :

$$|\mathbf{T}_{\max}| = 3m \frac{\mu_E}{R^3} \rho^2 \quad (\text{A.18})$$

The general result for the gravity gradient torque acting on the dumbbell given in eq. (A.17) may also be calculated directly from the gravity force acting at the center of gravity and the torque's lever arm (i.e., from the center of mass to the center of gravity).

This can best be done in the orbital reference frame using eqs. (A.6) and (A.14), which both have non-zero components within the orbital plane:

$$\mathbf{T} = \boldsymbol{\rho}_{cg} \times \mathbf{F} = c_u F_v - c_v F_u \approx -c_v F_u = -3m \frac{\mu_E}{R^3} \rho^2 \sin(2\alpha) \mathbf{z} \quad (\text{A.19})$$

The leading part of F_u (i.e., $-2m \mu_E / R^2$) in combination with the lever arm component c_v dominates the result. The remaining terms are smaller, by a factor $(\rho/R)^2$, and may be neglected in comparison (they can easily be calculated from the model above, if needed).

A.6 Illustration of Results

In order to visualize the results established above we provide here the plots of the relevant functions.

Figure 3 shows the behavior of the ‘perturbing functions’ $f_u(\alpha)$ and $f_v(\alpha)$ which represent the u and v components of the normalized (with respect to ρ^2/R^2) perturbing parts of the total force function presented in eq. (A.6):

$$\begin{aligned} \mathbf{F} &\approx -2m \frac{\mu_E}{R^2} \mathbf{u} + 3m \frac{\mu_E \rho^2}{R^4} \{ (1 - 3\cos^2 \alpha) \mathbf{u} + \sin(2\alpha) \mathbf{v} \} = \\ &= -2m \frac{\mu_E}{R^2} \left\{ \mathbf{u} + \left(\frac{\rho}{R} \right)^2 [f_u(\alpha) \mathbf{u} + f_v(\alpha) \mathbf{v}] \right\} \end{aligned} \quad (\text{A.20})$$

with:

$$f_u(\alpha) = -(3/2)(1 - 3\cos^2 \alpha); \quad f_v(\alpha) = -(3/2)\sin(2\alpha) \quad (\text{A.21})$$

The Figure confirms the well known fact that for a dumbbell along the orbital radius (i.e., $\alpha = 0$), the perturbing force f_u augments to the central force whereas f_v is absent. For small values of α , the f_u component starts to decrease slowly (and passes through 0 at $\alpha = 54.7^\circ$, see Table 1). The f_v component builds up a negative contribution in line with expectations. When $\alpha = 90^\circ$ the dumbbell is normal to the orbital radius and f_u reaches its (negative) minimum because both of the masses are now at the same distance behind the dumbbell center of mass (as seen from the Earth).

Similarly, Figure 4 illustrates the force along the x , y axes. From eq. (A.7) we can establish the perturbing components $f_x(\alpha)$ and $f_y(\alpha)$ in a similar way as was done above:

$$f_x(\alpha) = -(3/2)\sin \alpha(1 - 5\cos^2 \alpha); \quad f_y(\alpha) = -(3/2)\cos \alpha(3 - 5\cos^2 \alpha) \quad (\text{A.22})$$

The behavior of $f_y(\alpha)$ provides insights on the force felt by the mass-less rod of the dumbbell. It is seen that, for α near 0 and 180 degrees, the force reaches its maximum under the gravity gradient effect over the rod extended along the orbit radial direction. For $\alpha = 90^\circ$, the tension along the y -axis vanishes as the dumbbell rod lies essentially along the gravity field lines. Also at $\alpha = 39.2^\circ$ and 140.8° the torque vanishes because the tension components along the rod that are produced by the two masses are identical to those acting at the center of mass.

The x component is somewhat smaller and reaches its maxima values for $\alpha = 30^\circ$ and 150° on the positive side and for $\alpha = 90^\circ$ on the negative side. The torque vanishes at $\alpha = 63.4^\circ$ because the tension components are identical.

Figures 5 and 6 provide the position of the center of gravity relative to the (u, v) and (x, y) reference axes. The behavior shown in Figure 5 may be interpreted on the basis of the geometry shown in Figure 2. For $\alpha = 0, 90^\circ$, and 180° , we find that $c_v = 0$ which implies that the center of gravity lies on the orbital radius vector which is evident by inspection and interpretation of the geometry in Figure 2. It is clear that the center of gravity is in general not on the orbital radius vector.

For $\alpha = 0$ and 180° , the center of gravity is closer to the Earth than the center of mass by a distance of $-1.5(\rho/R)R$. When $\alpha = 90^\circ$, however, the center of gravity is further away from the Earth than the center of mass by a distance of $0.5(\rho/R)R$ as shown by the behavior of c_u . This can be understood by the fact that both masses are further from the center of Earth than the center of mass itself. In any case, it is evident that the center of gravity lies in general away from the dumbbell axis.

Finally, it may be mentioned that the dumbbell's 'center of motion' (see also the discussion in point 6 of section 3.5) is at a distance of $-(\rho/R)\rho$ below the center of mass position. Here, it has been assumed that the dumbbell maintains its orientation along the local vertical. This result can be proven by substituting $\alpha = 0$ in eq. (A.6) to obtain the radial force, by calculating the radial distance at which the centrifugal force equals the resulting radial force, and by expanding for small values of (ρ/R) . Furthermore, the well known relationship that the center of motion equals $(R R_{cg})^{1/3}$ can easily be confirmed by substituting the expression for c_u given in eq. (A.14).

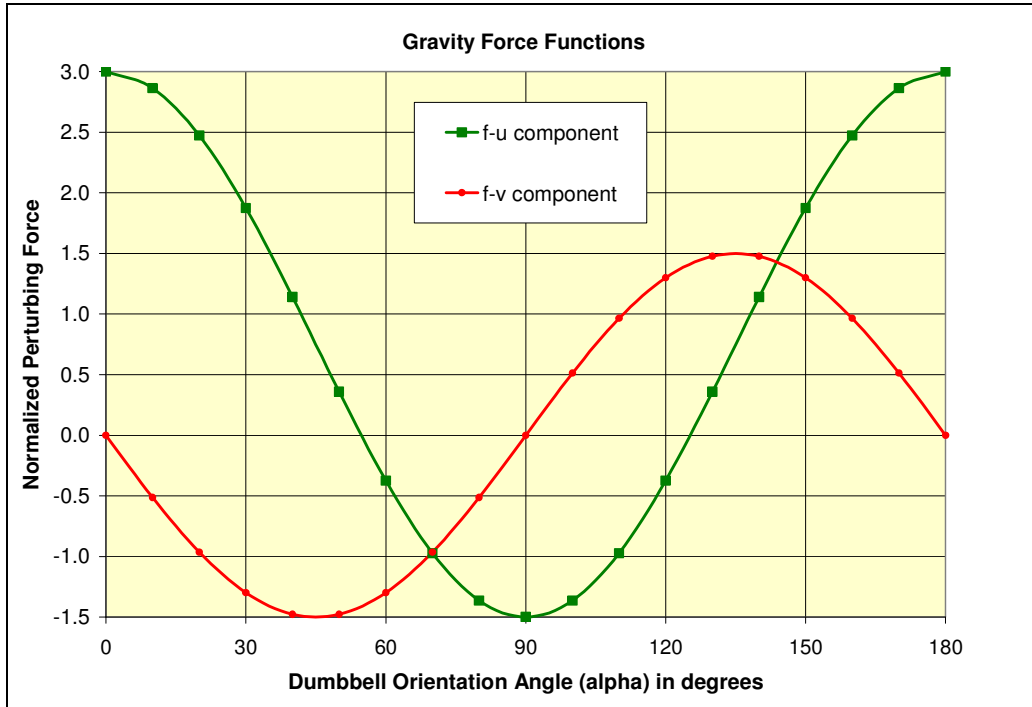


Figure 3 – Normalized Perturbing Force for Dumbbell Satellite in (u, v) Frame

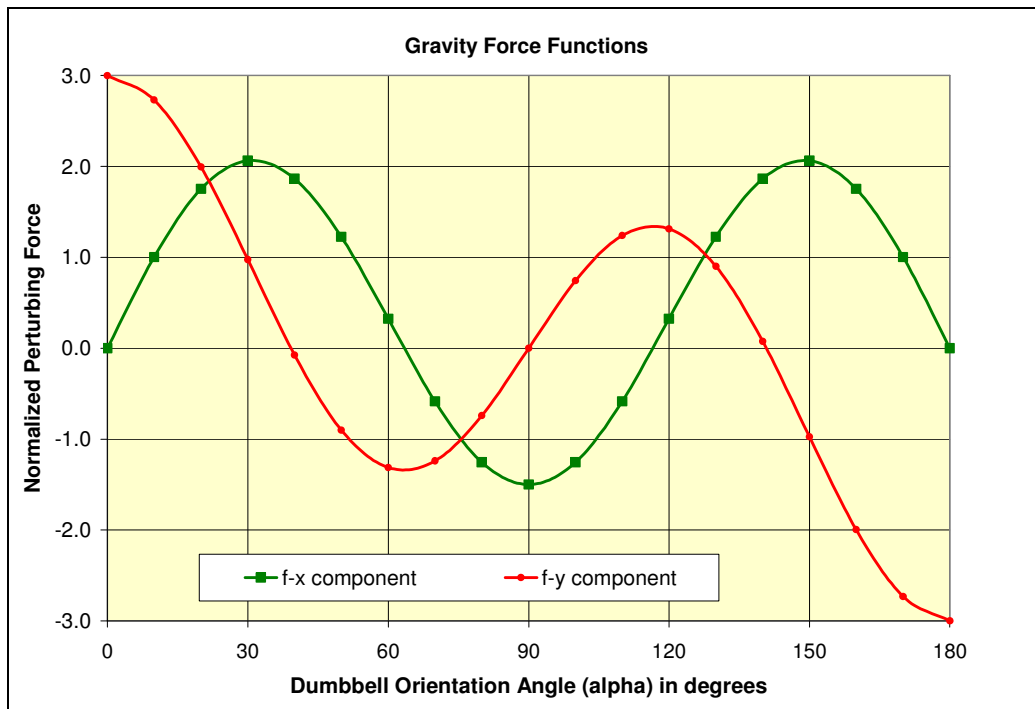


Figure 4 – Normalized Perturbing Force for Dumbbell Satellite in (x, y) Frame

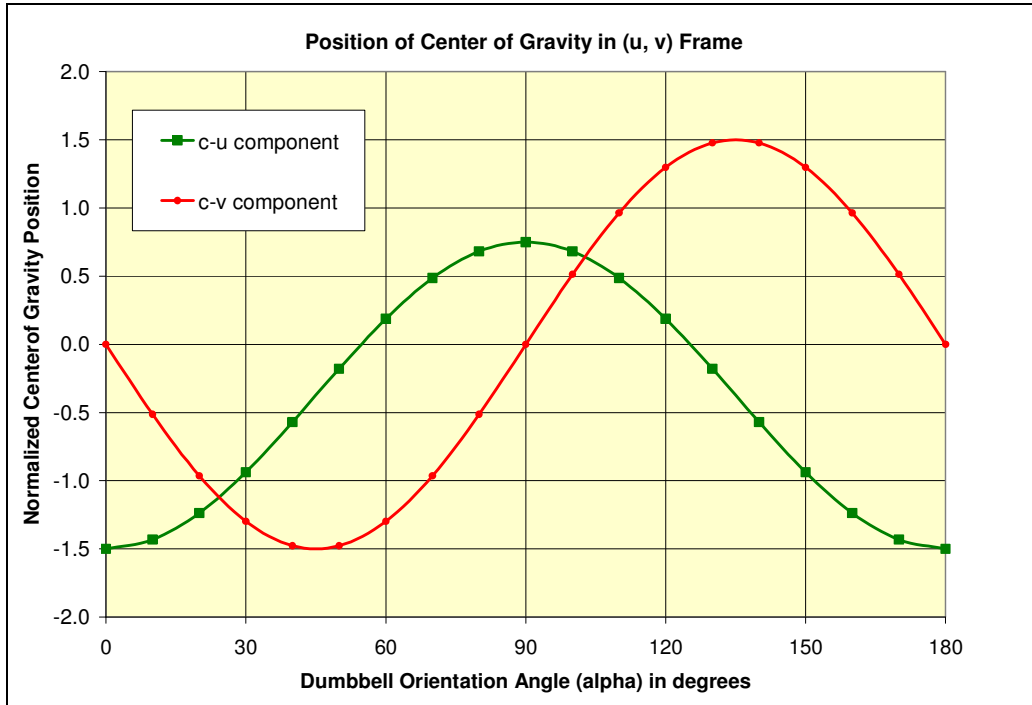


Figure 5 – Center of Gravity Position for Dumbbell Satellite in (u, v) Frame

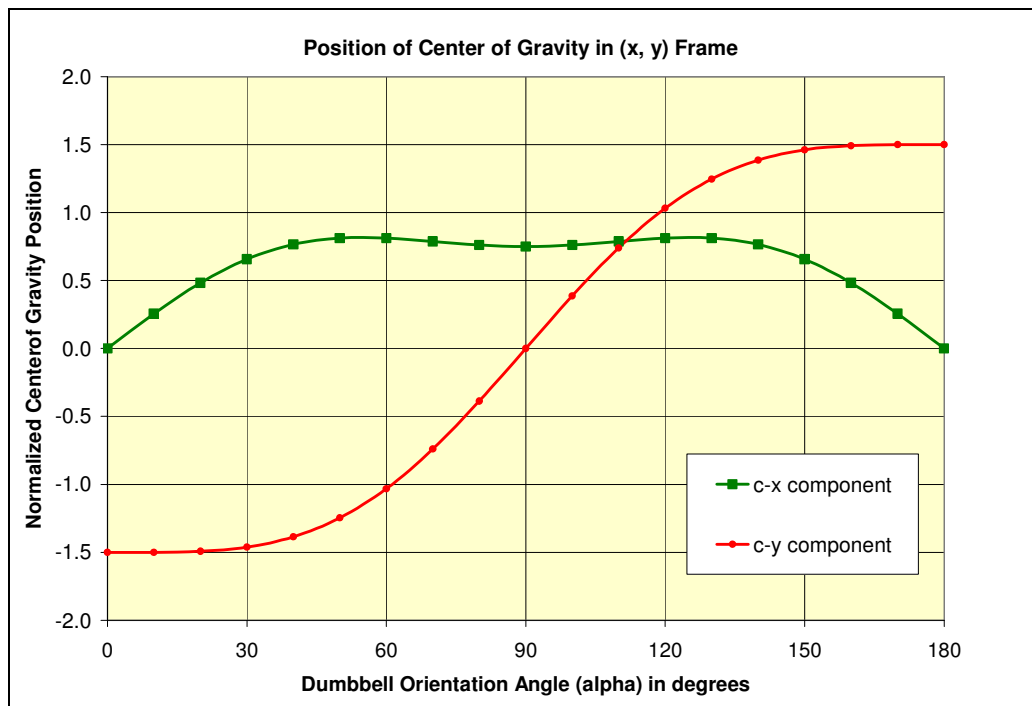


Figure 6 – Center of Gravity Position for Dumbbell Satellite in (x, y) Frame

APPENDIX B. CYLINDRICAL SATELLITE ILLUSTRATION

We will now study another straightforward example, namely a cylindrical satellite.

B.1 Geometry

Figure 7 illustrates the geometry of a (assumed homogeneous) cylindrical satellite in an orbit around an idealized spherical Earth.

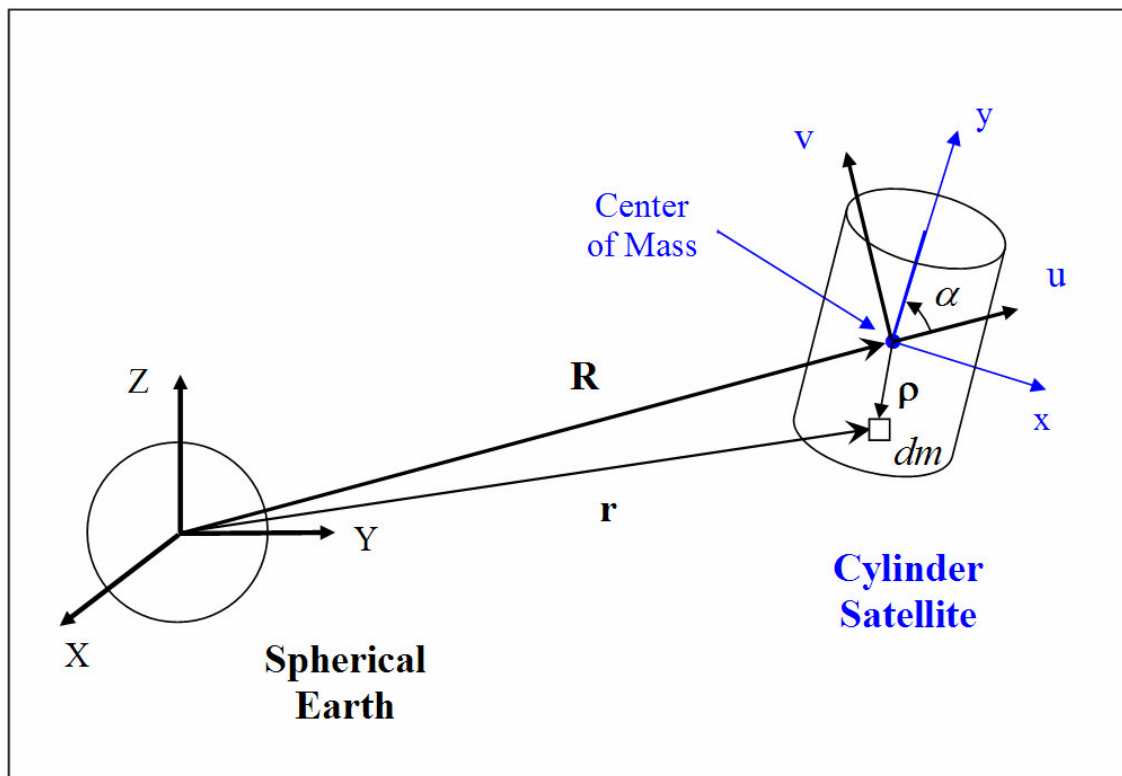


Figure 7 – Illustration of (Planar) Geometry of Cylindrical Satellite

For simplicity, the satellite attitude is assumed to be constrained to in-plane librations so that the BCRF x, y axes lie within the orbital plane at all times. The x -axis points normal to the cylinder mantle and the y -axis is directed along the symmetry axis of the cylinder. The z -axis (not shown) is normal to the orbit plane. All of these three axes are assumed to be principal axes. Also we introduce the local ORF orbit frame (u, v, w) with the u -axis along the instantaneous orbital radius vector, the v -axis along the local horizontal direction, and the w -axis is along the orbit-normal and identical to the z -axis. The geometrical transformation between the ORF and BCRF reference frames in eq. (A.1) holds also here.

Since a direct calculation of the force is rather involved we proceed to the application of the general force model presented eq. (21) to the cylindrical satellite.

B.2 Result of Gravity Force Model

The moments of inertias of a cylinder can be calculated by eq. (15c)

$$I_x = \frac{1}{4} m \rho^2 + mL^2/12 ; \quad I_y = \frac{1}{2} m \rho^2 ; \quad I_z = \frac{1}{4} m \rho^2 + mL^2/12 \quad (\text{B.1})$$

with ρ denoting the cylinder's radius and L its length. The inertia matrix follows now as:

$$[\mathbf{I}] = \frac{1}{2} m \rho^2 \begin{bmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} ; \quad \text{tr}[\mathbf{I}] = (\frac{1}{2} + d) m \rho^2 \quad (\text{B.2})$$

with the constant $d = \frac{1}{2} + (L/\rho)^2/6$. This indicates that there is 'cubical' symmetry in the special case when $d = 1$, i.e. when L equals $\sqrt{3}\rho$. In this case, the quadrupole contribution disappears completely.

By using the results above and the explicit expression for $\mathbf{u} = (\sin\alpha, \sin\alpha, 0)^T$ along the satellite's body axes (as in eq. A.1), we simplify the terms of eqs. (24) as follows:

$$\text{Term 1:} \quad (\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} = \frac{1}{2} m \rho^2 \{1 + (d-1) \sin^2\alpha\} (\sin\alpha \mathbf{x} + \cos\alpha \mathbf{y}) \quad (\text{B.3a})$$

$$\text{Term 2:} \quad \text{tr}[\mathbf{I}]\mathbf{u} = (\frac{1}{2} + d) m \rho^2 (\sin\alpha \mathbf{x} + \cos\alpha \mathbf{y}) \quad (\text{B.3b})$$

$$\text{Term 3:} \quad [\mathbf{I}] \mathbf{u} = \frac{1}{2} m \rho^2 (d \sin\alpha \mathbf{x} + \cos\alpha \mathbf{y}) \quad (\text{B.3c})$$

With the help of these expressions the force expression in eq. (21) takes the form:

$$\mathbf{F} \approx -m \frac{\mu_E}{R^2} \mathbf{u} + \frac{3}{4} m \frac{\mu_E \rho^2}{R^4} (1-d) \{ \sin\alpha (5\cos^2\alpha - 1) \mathbf{x} + \cos\alpha (5\cos^2\alpha - 3) \mathbf{y} \} \quad (\text{B.4})$$

Again it can be seen that the quadrupole term disappears when $d = 1$. The result in Eq. (B.4) is strikingly similar in structure to the expression of the dumbbell given in eq. (A.7). This can be understood by considering a cylinder with zero radius ($\rho \rightarrow 0$). In that case the inertia matrices of the dumbbell and the cylinder in eqs. (A.11) and (B.2) can be made identical by equating:

$$2m_{db} \rho_{db}^2 \equiv m_{cyl} L_{cyl}^2 \quad (\text{B.5})$$



with db referring to parameters of the dumbbell and cyl to quantities belonging to the cylinder.

The result of eq. (B.4) may be reduced further in terms of components along the u, v axes as was done in eq. (A.4). Naturally, the cylindrical satellite can further be analyzed in a similar way as was done in the previous section for the dumbbell but the results will be (at least qualitatively) identical to those presented for the dumbbell.

B.3 Gravity-Gradient Torque

The gravity-gradient torque induced by the gravity force can now be calculated in explicit form for the dumbbell satellite from the expression in eq. (40):

$$\mathbf{T} \approx \frac{3\mu_E}{R^3} \{ \mathbf{u} \times [\mathbf{I}] \mathbf{u} \} = -\frac{3}{4} m \frac{\mu_E}{R^3} \rho^2 (1-d) \sin(2\alpha) \mathbf{z} \quad (\text{B.6})$$

This result confirms that the torque vanishes when $\alpha = 0^\circ$ and $\pm 90^\circ$ similarly as was the case for the dumbbell. The maximum possible torque occurs when $\alpha = 45^\circ$ and 135°

$$|\mathbf{T}_{\max}| = \frac{3}{4} m \frac{\mu_E}{R^3} \rho^2 (1-d) \quad (\text{B.7})$$

Alternatively, we may calculate the gravity gradient torque by using the torque's lever arm which is represented by the vector from the center of gravity to the center of mass (still to be done).

B.4 Cylinder Normal to Orbit Plane

Finally, it is of interest to consider a satellite with its symmetry axis normal to the orbit plane. In this case, the inertias about the x and y axes will be equal and the inertia matrix follows from eq. (B.2) as follows:

$$[\mathbf{I}] = \frac{1}{2} m \rho^2 \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.8})$$

and the force component $[\mathbf{I}]\mathbf{u}$ can be calculated as:

$$[\mathbf{I}]\mathbf{u} = \frac{1}{2} m \rho^2 \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \sin \alpha \\ \cos \alpha \\ 0 \end{pmatrix} = \frac{1}{2} m \rho^2 d \begin{pmatrix} \sin \alpha \\ \cos \alpha \\ 0 \end{pmatrix} \quad (\text{B.9})$$

which is in this case perfectly aligned with the orbital radius \mathbf{u} . Therefore, the results of the term in eqs. (B.3a) can be simplified further in this case:

$$\text{Term 1:} \quad (\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} = \frac{1}{2} m \rho^2 d \mathbf{u} \quad (\text{B.10a})$$

$$\text{Term 2:} \quad \text{tr}[\mathbf{I}] \mathbf{u} = (\frac{1}{2} + d) m \rho^2 \mathbf{u} \quad (\text{B.10b})$$

$$\text{Term 3:} \quad [\mathbf{I}] \mathbf{u} = \frac{1}{2} m \rho^2 d \mathbf{u} \quad (\text{B.10c})$$

The force expression of eq. (21) takes now the very straightforward form:

$$\begin{aligned} \mathbf{F} &\approx -m \frac{\mu_E}{R^2} \mathbf{u} + \frac{3}{4} m \frac{\mu_E \rho^2}{R^4} \{5d - (1+2d) - 2d\} = \\ &= -m \frac{\mu_E}{R^2} \mathbf{u} + (9/4) (d - 1) m \frac{\mu_E \rho^2}{R^4} \end{aligned} \quad (\text{B.11})$$

APPENDIX C. COMPARISON OF FORCE RESULTS

C.1 Force Expressed in Moments of Inertia

The force result presented in eq. (21) should be identical to the one given in eq. (24) of [1]. This can immediately be confirmed for the dominant inverse-square monopole term which is proportional to μ_E/R^2 .

In order to check the agreement for the more complicated second-order contribution (i.e., the term multiplied by μ_E/R^4), we consider the relevant second-order quadrupole terms of eq. (21):

$$\text{Term 1: } (\mathbf{u}^T[\mathbf{I}]\mathbf{u})\mathbf{u} = \{l^2 I_x + m^2 I_y + n^2 I_z - 2(lm I_{xy} + ln I_{xz} + mn I_{yz})\} \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (\text{C.1a})$$

$$\text{Term 2: } \text{tr}[\mathbf{I}]\mathbf{u} = (I_x + I_y + I_z) \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (\text{C.1b})$$

$$\text{Term 3: } [\mathbf{I}]\mathbf{u} = \begin{pmatrix} lI_x - mI_{xy} - nI_{xz} \\ -lI_{xy} + mI_y - nI_{yz} \\ -lI_{xz} - mI_{yz} + nI_z \end{pmatrix} \quad (\text{C.1c})$$

The expression between brackets {...} in the gravity force of eq. (C.1a) can now be written as follows:

$$\begin{aligned} \{ \dots \} &= 5 (\text{Term 1}) - (\text{Term 2}) - 2 (\text{Term 3}) = \\ &= \{ (5l^2 - 1)I_x + (5m^2 - 1)I_y + (5n^2 - 1)I_z - 10(lm I_{xy} + ln I_{xz} + mn I_{yz}) \} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + \\ &\quad - 2 \begin{pmatrix} lI_x - mI_{xy} - nI_{xz} \\ -lI_{xy} + mI_y - nI_{yz} \\ -lI_{xz} - mI_{yz} + nI_z \end{pmatrix} \quad (\text{C.2}) \end{aligned}$$

The force expression in eq. (21) can now be written in the final form:

$$\mathbf{F} \approx -m_{sat} \frac{\mu_E}{R^2} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + (3/2) \frac{\mu_E}{R^4} \left\{ \begin{pmatrix} (5l^2 - 3)I_x + (5m^2 - 1)I_y + (5n^2 - 1)I_z \\ (5l^2 - 1)I_x + (5m^2 - 3)I_y + (5n^2 - 1)I_z \\ (5l^2 - 1)I_x + (5m^2 - 1)I_y + (5n^2 - 3)I_z \end{pmatrix} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + \right. \\ \left. - 10(lmI_{xy} + lnI_{xz} + mnI_{yz}) \begin{pmatrix} l \\ m \\ n \end{pmatrix} + 2 \begin{pmatrix} mI_{xy} + nI_{xz} \\ lI_{xy} + nI_{yz} \\ lI_{xz} + mI_{yz} \end{pmatrix} \right\} \quad (C.3)$$

When comparing this result with eq. (24) of [1] we find agreement (**apart from the expected sign reversal**) in all of the terms **except for the final term**:

1. the factor 2 multiplier for the last entry of eq. (26) corresponds to a factor 4 in eq. (24) of [1]
2. furthermore, in eq. (24) of [1], the last vector term shown in eq. (C.3) is wrongly multiplied by the components l , m , and n .

In the case when the satellite reference frame is taken along its principal axes, we have $I_{xy} = I_{xz} = I_{yz} = 0$ so that the result in eq. (26) can be simplified significantly as follows:

$$\mathbf{F} \approx -m_{sat} \frac{\mu_E}{R^2} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + (3/2) \frac{\mu_E}{R^4} \begin{pmatrix} (5l^2 - 3)I_x + (5m^2 - 1)I_y + (5n^2 - 1)I_z \\ (5l^2 - 1)I_x + (5m^2 - 3)I_y + (5n^2 - 1)I_z \\ (5l^2 - 1)I_x + (5m^2 - 1)I_y + (5n^2 - 3)I_z \end{pmatrix} \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (C.4)$$

This result is in *complete* agreement (**apart from the sign reversal**) with eq. (25) of [1].

C.2 Force Expressed in Inertia Integrals

We may also write the expression in {...} of eq. (21) in terms of the inertial integrals and compare the final result with eq. (18) of [1]. First, we note the following identity relationships which follow from eqs. (14a-c):

$$I_x = J_y + J_z; \quad I_y = J_x + J_z; \quad I_z = J_x + J_y \quad (C.5)$$

When substituting these expressions, along with the inertia products in eqs. (14d-f), into the terms given in eqs. (C.1a-c), we find:

Term 1: $(\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u}/R^2 =$

$$= \{(1 - l^2) J_x + (1 - m^2) J_y + (1 - n^2) J_z - 2(lm J_{xy} + ln J_{xz} + mn J_{yz})\} \begin{pmatrix} l \\ m \\ n \end{pmatrix} =$$

$$= - \{l^2 J_x + m^2 J_y + n^2 J_z + 2(lm J_{xy} + ln J_{xz} + mn J_{yz})\} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + (J_x + J_y + J_z) \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (\text{C.6a})$$

Term 2: $tr[\mathbf{I}]\mathbf{u} = (I_x + I_y + I_z) \mathbf{u} = 2 (J_x + J_y + J_z) \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (\text{C.6b})$

Term 3: $[\mathbf{I}] \mathbf{u} = \begin{pmatrix} lI_x - mI_{xy} - nI_{xz} \\ -lI_{xy} + mI_y - nI_{yz} \\ -lI_{xz} - mI_{yz} + nI_z \end{pmatrix} = \begin{pmatrix} l(J_y + J_z) - mJ_{xy} - nJ_{xz} \\ -lJ_{xy} + m(J_x + J_z) - nJ_{yz} \\ -lJ_{xz} - mJ_{yz} + n(J_x + J_y) \end{pmatrix} \quad (\text{C.6c})$

Now we add 5 times the last term in eq. (C.6a), subtract the term (C.6b), and subtract twice the term (C.6c). After writing out the 3 components we find:

$$\mathbf{PART1} = \begin{pmatrix} 3lJ_x + lJ_y + lJ_z + 2mJ_{xy} + 2nJ_{xz} \\ mJ_x + 3mJ_y + mJ_z + 2lJ_{xy} + 2nJ_{yz} \\ nJ_x + nJ_y + 3nJ_z + 2lJ_{xz} + 2mJ_{yz} \end{pmatrix} \quad (\text{C.7})$$

The remaining contribution comes from the first term in (C.6a), which is written in the form:

$$\mathbf{PART2} = \{l^2 J_x + m^2 J_y + n^2 J_z + 2(lm J_{xy} + ln J_{xz} + mn J_{yz})\} \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (\text{C.8})$$

With the help of these expressions, it is possible to rewrite the force expression in eq. (26) in the same form as eq. (18) in [1]:

$$\begin{aligned}
 \mathbf{F} \approx & -m_{sat} \frac{\mu_E}{R^2} \begin{pmatrix} l \\ m \\ n \end{pmatrix} + (3/2) \frac{\mu_E}{R^4} \left\{ \begin{pmatrix} 3lJ_x + lJ_y + lJ_z + 2mJ_{xy} + 2nJ_{xz} \\ mJ_x + 3mJ_y + mJ_z + 2lJ_{xy} + 2nJ_{yz} \\ nJ_x + nJ_y + 3nJ_z + 2lJ_{xz} + 2mJ_{yz} \end{pmatrix} + \right. \\
 & \left. - 5 \{ l^2 J_x + m^2 J_y + n^2 J_z + 2(lm J_{xy} + ln J_{xz} + mn J_{yz}) \} \begin{pmatrix} l \\ m \\ n \end{pmatrix} \right\} \quad (C.9)
 \end{aligned}$$

This result is completely in agreement with eq. (18) in [1] as long as we remember that the vector $\mathbf{u} = (l, m, n)^T$ as employed here has the opposite sign of the orbital radius direction used in [1].

APPENDIX D. GRAVITY FORCE in ORBIT FRAME

The gravity force given in eq. (21) has been established on the basis of the inertia matrix $[\mathbf{I}]$ as well as the orbital radius vector \mathbf{R} expressed within the body reference frame (BCRF). Sometimes, it is advantageous to know the gravity gradient force components expressed in the orbital reference frame (u, v, w) . In particular, this implies that the \mathbf{U}_{body} vector, defined by:

$$\mathbf{U}_{\text{body}} = \{5 (\mathbf{u}^T [\mathbf{I}] \mathbf{u}) \mathbf{u} - \text{tr}[\mathbf{I}] \mathbf{u} - 2[\mathbf{I}] \mathbf{u}\} \quad (\text{D.1})$$

must be transformed to the ORF frame. Thereto, we introduce the transformation matrix $[\mathbf{A}]$ of the body reference frame within the orbital reference frame:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = [\mathbf{A}] \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (\text{D.2})$$

The inertia matrix $[\mathbf{I}]$, which has originally been defined relative to the body frame, can be transformed to the matrix $[\hat{\mathbf{I}}]$ in the orbital reference frame as follows:

$$[\mathbf{I}] = \rho^2 [\mathbf{E}] - \int_{m_{\text{sat}}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} (x, y, z) dm_{\text{sat}} = [\mathbf{A}] \left\{ \rho^2 [\mathbf{E}] - \int_{m_{\text{sat}}} \begin{pmatrix} u \\ v \\ w \end{pmatrix} (u, v, w) dm_{\text{sat}} \right\} [\mathbf{A}]^T = [\mathbf{A}] [\hat{\mathbf{I}}] [\mathbf{A}]^T$$

$$\rightarrow [\hat{\mathbf{I}}] = [\mathbf{A}]^T [\mathbf{I}] [\mathbf{A}] \quad (\text{D.3})$$

with $\rho^2 = x^2 + y^2 + z^2$ and $[\mathbf{E}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, i.e. the unity matrix.

Thus, eq. (D.1) can now be expressed in the orbital reference frame as:

$$\mathbf{U}_{\text{ORF}} = \{5 (\mathbf{u}^T [\hat{\mathbf{I}}] \mathbf{u}) \mathbf{u} - \text{tr}[\hat{\mathbf{I}}] \mathbf{u} - 2[\hat{\mathbf{I}}] \mathbf{u}\} \quad (\text{D.4})$$

It is important to note that the orbital radius vector \mathbf{R} is oriented along the local vertical, i.e. the u axis. Therefore, $\mathbf{u} = (1, 0, 0)^T$ within the orbital frame and we can calculate:

1. term 1:

$$(\mathbf{u}^T [\widehat{\mathbf{I}}] \mathbf{u}) = \widehat{I}_x = a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z - 2(a_{11}a_{21}I_{xy} + a_{11}a_{31}I_{xz} + a_{21}a_{31}I_{yz}) \quad (\text{D.5a})$$

$$(\mathbf{u}^T [\widehat{\mathbf{I}}] \mathbf{u}) \mathbf{u} = \widehat{I}_x \mathbf{u} = \begin{pmatrix} a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z - 2(a_{11}a_{21}I_{xy} + a_{11}a_{31}I_{xz} + a_{21}a_{31}I_{yz}) \\ 0 \\ 0 \end{pmatrix} \quad (\text{D.5b})$$

If the spacecraft body frame were the principal reference frame, the cross-product terms cancel and the results of (D.5) could be simplified somewhat.

When comparing the result in eq. (D.5a) with that in eq. (19) we can see full consistency between these two expressions because of the coordinate transformation matrix in eq. (D.2) which shows:

$$\mathbf{R}/R = l \mathbf{x} + m \mathbf{y} + n \mathbf{z} = \begin{pmatrix} l \\ m \\ n \end{pmatrix} = [\mathbf{A}] \begin{pmatrix} u \\ v \\ w \end{pmatrix} = a_{11} \mathbf{u} + a_{21} \mathbf{v} + a_{31} \mathbf{w} \quad (\text{D.6})$$

2. term 2:

$$\text{tr}[\widehat{\mathbf{I}}] \mathbf{u} = \text{tr}[\mathbf{I}] \mathbf{u} = (I_x + I_y + I_z) \mathbf{u} = \begin{pmatrix} I_x + I_y + I_z \\ 0 \\ 0 \end{pmatrix} \quad (\text{D.7})$$

It can be shown that $\text{tr}[\mathbf{I}]$ is in fact invariant to coordinate transformations.

3. term 3:

$$\begin{aligned} [\widehat{\mathbf{I}}] \mathbf{u} &= \begin{bmatrix} \widehat{I}_x & -\widehat{I}_{xy} & -\widehat{I}_{xz} \\ -\widehat{I}_{xy} & \widehat{I}_y & -\widehat{I}_{yz} \\ -\widehat{I}_{xz} & -\widehat{I}_{yz} & \widehat{I}_z \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \widehat{I}_x \\ -\widehat{I}_{xy} \\ -\widehat{I}_{xz} \end{pmatrix} = \\ &= \begin{pmatrix} a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z - 2(a_{11}a_{21}I_{xy} + a_{11}a_{31}I_{xz} + a_{21}a_{31}I_{yz}) \\ a_{11}a_{12}I_x + a_{21}a_{22}I_y + a_{31}a_{32}I_z - (a_{11}a_{22} + a_{12}a_{21})I_{xy} - (a_{11}a_{32} + a_{12}a_{31})I_{xz} - (a_{21}a_{32} + a_{22}a_{31})I_{yz} \\ a_{11}a_{13}I_x + a_{21}a_{23}I_y + a_{31}a_{33}I_z - (a_{11}a_{23} + a_{13}a_{21})I_{xy} - (a_{11}a_{33} + a_{13}a_{31})I_{xz} - (a_{21}a_{33} + a_{23}a_{31})I_{yz} \end{pmatrix} \end{pmatrix} \quad (\text{D.8})$$

This result simplifies considerably when considering the principal body frame as the reference:

$$[\hat{\mathbf{I}}]\mathbf{u} = \begin{pmatrix} \hat{I}_x \\ -\hat{I}_{xy} \\ -\hat{I}_{xz} \end{pmatrix} = \begin{pmatrix} a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z \\ a_{11} a_{12} I_x + a_{21} a_{22} I_y + a_{31} a_{32} I_z \\ a_{11} a_{13} I_x + a_{21} a_{23} I_y + a_{31} a_{33} I_z \end{pmatrix} \quad (\text{D.9})$$

With these inputs, explicit expressions can be established for the vector \mathbf{U} in eq. (D.4):

$$\mathbf{U}_{\text{ORF}} = \begin{pmatrix} 2\hat{I}_x - \hat{I}_y - \hat{I}_z \\ 2\hat{I}_{xy} \\ 2\hat{I}_{xz} \end{pmatrix} = \begin{pmatrix} 3(a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z) - (I_x + I_y + I_z) - 6(a_{11} a_{21} I_{xy} + a_{11} a_{31} I_{xz} + a_{21} a_{31} I_{yz}) \\ -2(a_{11} a_{12} I_x + a_{21} a_{22} I_y + a_{31} a_{32} I_z) + 2(a_{11} a_{22} + a_{12} a_{21}) I_{xy} + 2(a_{11} a_{32} + a_{12} a_{31}) I_{xz} + 2(a_{21} a_{32} + a_{22} a_{31}) I_{yz} \\ -2(a_{11} a_{13} I_x + a_{21} a_{23} I_y + a_{31} a_{33} I_z) + 2(a_{11} a_{23} + a_{13} a_{21}) I_{xy} + 2(a_{11} a_{33} + a_{13} a_{31}) I_{xz} + 2(a_{21} a_{33} + a_{23} a_{31}) I_{yz} \end{pmatrix} \quad (\text{D.10})$$

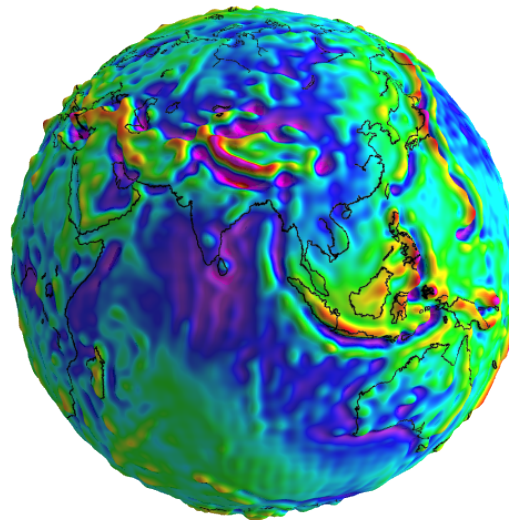
A more compact result is found for the case when the body frame happens to be the principal reference frame:

$$\mathbf{U}_{\text{ORF}} = \begin{pmatrix} 3(a_{11}^2 I_x + a_{21}^2 I_y + a_{31}^2 I_z) - (I_x + I_y + I_z) \\ -2(a_{11} a_{12} I_x + a_{21} a_{22} I_y + a_{31} a_{32} I_z) \\ -2(a_{11} a_{13} I_x + a_{21} a_{23} I_y + a_{31} a_{33} I_z) \end{pmatrix} \quad (\text{D.11})$$

These results are consistent with those in Hughes, [4], p. 287, eqs. (17), when accounting for the different definitions of the orbital reference axes and sign conventions.



Earth Gravity Model Update for the Generic Drag-Free Simulator



Project:	Document No.:
FIRST LOOK	FLK-SIM-TN-ZAR-007
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm	Phone: +49-421-218-9422
D 28359 Bremen	Fax: +49-421-218-4356
Germany	E-Mail: pelivan@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-007

Issue: 1.0

Written: Ivanka Pelivan Date: 15th February 2007

Approved: - Date: -

	<p style="text-align: center;">Earth Gravity Model Update for the Generic Drag-Free Simulator</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-007 Issue: 1.0 Page: 2 of 19</p>
---	---	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	02/15/07	all	Initial Version by Ivanka Pelivan	√

Contents

1. Introduction	5
2. Earth Gravity Field Models	5
2.1. Calculation Method	5
2.2. Model Overview	5
3. Implementation of New Models	12
3.1. Additions to the Drag-Free Simulator Project	12
3.2. Data Formats	12
3.3. Application and Modifications to the Drag-Free Simulator	14
3.4. Guidelines for Future Model Update	15
A. Tidal Effects on the Earth's Gravity Field	16
A.1. Tide Systems	16

Acronyms and Abbreviations

Symbol	Description
<i>CHAMP</i>	CHALLENGING Mini-satellite Payload
<i>CLS</i>	Collecte Localisation Satellites
<i>CSR</i>	Center for Space Research (University of Texas)
<i>ECCO</i>	Estimating the Circulation and Climate of the Ocean
<i>EGM96</i>	Earth Gravity Model 1996
<i>EIGEN</i>	European Improved Gravity model of the Earth by New techniques
<i>GFZ</i>	GeoForschungsZentrum (Potsdam)
<i>GGM</i>	GRACE Gravity Model
<i>GRACE</i>	Gravity Recovery and Climate Experiment
<i>IAG</i>	International Association of Geodesy
<i>ICGEM</i>	International Centre for Global Earth Models
<i>Lageos</i>	Laser Geodynamics Satellites
<i>NGA</i>	National Geospatial Intelligence Agency
<i>NRCan</i>	Natural Resources Canada

1. Introduction

The Earth Gravity Model EGM96 [9] provided by NASA has been the standard model for gravity and gravity-gradient calculation used in the generic drag-free simulator [10]. The spherical harmonic coefficients of the Earth potential field are based on pre-CHAMP, multi-satellite plus surface data. Newer models from CHAMP, GRACE, and Lageos tracking data combined with surface data or as satellite-only model are available. The combination models (satellite plus surface data) are intended to replace the EGM96 since they have shown an accuracy improvement of several orders of magnitude for modelling the Earth gravity field.

2. Earth Gravity Field Models

The ICGEM homepage [6] provides an overview of available models for gravity field modelling with data for download.

2.1. Calculation Method

As described in [15] the calculation of Earth gravity and gravity-gradient is based on the Earth potential field which is expressed by the spherical harmonic model:

$$\Phi = \frac{GM}{R} \left[1 + \sum_{n=1}^{\infty} \sum_{m=0}^n \left(\frac{R_{\text{Earth}}}{R} \right)^n (C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)) P_{nm}(\sin(\phi_c)) \right] \quad (1)$$

M is the mass of the Earth, G is the gravitational constant, R_{Earth} is the Earth radius, R the position vector, $P_{nm}(\sin(\phi_c))$ are the Legendre polynomials of the geocentric latitude ϕ_c , λ is the geocentric longitude and C_{nm}, S_{nm} are the spherical harmonic coefficients of the Earth potential field which are available up to degree and order $n = m = 360$ depending on the model used. With equation (1) all effects of the Earth's non-symmetric mass distribution can be described. Earth oblateness and the main variations in the Earth's shape are covered by the first six degrees of the harmonic series in (1).

2.2. Model Overview

Utilization of equation (1) to calculate the Earth gravity field requires supply of the coefficients C_{nm}, S_{nm} . A number of gravity field models is available, among them NASA's EGM96 and newer models based on data from more recent satellite missions like CHAMP and GRACE. An overview of available models for gravity field modelling can be found on the ICGEM homepage [6].

EGM96 Model

The standard model used so far for calculation of the Earth gravity and gravity gradient has been the EGM96 (Earth Gravity Model, see [15, 9]). The coefficients C_{nm} , S_{nm} for utilization of equation (1) are provided by NASA for download [9].

EGM96 was developed by combining data from multiple near-Earth satellites and an extensive set of land-based measurements spanning a period of more than thirty years.

New Models

New models are based on data from the GRACE mission. GRACE-only as well as combination models including other satellite missions and surface data are available. Science data processing, distribution, archiving and product verification regarding the GRACE mission are managed under a cooperative arrangement between the University of Texas Center for Space Research (CSR), the Jet Propulsion Laboratory (JPL) and the Geoforschungszentrum Potsdam (GFZ).

Mean monthly estimates of spherical harmonic coefficients for the Earth gravitational potential field are distributed through the PO.DAAC (Physical Oceanography Distributed Active Archive Center) at JPL or ISDC (Information System and Data Center) at GFZ Potsdam websites [11, 5]. Several months of data are combined to produce an estimate of the mean or static gravity field. Mean or static gravity field products are provided by GFZ and CSR as satellite-only or combination models. In generating a combination model, the long- to medium wavelengths features of the Earth's gravity field resolved from space are preserved while the shorter wavelengths are derived from surface data.

Gravity Field EIGEN by GFZ

Model data from the satellite missions CHAMP, GRACE and Lageos and from surface data is provided by the GRACE Scientific Results website of GFZ Potsdam [4]. One to two model updates are provided every year. The download area up-to-date includes the following coefficient data:

- Satellite-only gravity model EIGEN-GL04S1 complete to degree and order 150 from GRACE and Lageos data, released May 24, 2006
- Combined gravity field model EIGEN-GL04C complete to degree and order 360 from GRACE, Lageos and surface gravity data, released on March 31, 2006
- Combined gravity field model EIGEN-CG03C complete to degree and order 360 from CHAMP, GRACE and surface gravity data, released on May 12, 2005
- Combined gravity field model EIGEN-CG01C complete to degree and order 360 from CHAMP, GRACE and surface gravity data, released on October 29, 2004

- GRACE satellite-only Earth gravity field model EIGEN-GRACE02S complete to degree and order 150 released on February 13, 2004 to the GRACE Science Team and August 9, 2004 to the public
- First GFZ GRACE gravity field model EIGEN-GRACE01S released on July 25, 2003

Hereby, the naming convention of the GFZ Potsdam-derived global gravity field solution is the following:

EIGEN	European Improved Gravity model of the Earth by New techniques
-GL	from GRACE and Lageos data
-CG	from CHAMP and GRACE data
*	01,02,03,04: version
S	solely from satellite data
C	combination model from satellite and surface gravity data

EIGEN-GRACE01S is the first GRACE gravity model and is based on 39 days of preliminary GRACE flight instrument data. According to [4] this model has proven about five times more accurate than the latest CHAMP field model and about 50 times more accurate than pre-CHAMP satellite only gravity models at 1000 km half wavelength. The solution strategy for combination of satellite and surface data resulting in the global gravity field model EIGEN-CG01C includes downweighting of surface data relative to the satellite-only information in separating the satellite normal equation system up to degree 70. Contributions for the coefficients with degree 71 through 109 were allowed to overlap for the satellite and surface data. The derivation of the combination model EIGEN-CG01C is explained in [14].

Combination models like EGM96 are compiled from satellite altimetry, ship-borne gravimetry over the oceans, air-borne and terrestrial gravimetry over land and the north polar region. Compared to EGM96, the long- to medium wavelength part benefits from the unmatched performance of the CHAMP and GRACE missions and partly also from improvements in the higher frequency part due to a more complete surface data record (apart from Antarctica an almost complete global coverage is reached). According to [14] there are seven surface data sets used for the combination model EIGEN-CG01C:

1. Gravity anomalies from the Arctic Gravity Project (ArcGP, see [3])
2. NRCan gravity anomalies for North America
3. NGA altimetric gravity anomalies over the ocean
4. Geoid undulations over the oceans using CLS01 mean sea surface and the ECCO sea surface topography

5. NGA terrestrial gravity anomalies not covered by sets 1-3
6. NGA ship-borne gravity over water depths less than 2000m.

The data sets already incorporated in the EGM96 solution are the NGA sets 4, 6 and 7. Improvements to the global gravity field model from CHAMP and GRACE data is most obvious for the polar caps that have not been resolved by former satellite-only models, and also for regions in Africa, Asia and South America which were not homogeneously covered by gravimetric data previously. The newly available or improved surface data sets also mainly contribute to higher resolution for the polar regions.

In [13] the GRACE satellite-only Earth gravity field model EIGEN-GRACE02S is described. For this model, 110 days of GRACE tracking data have been used. Documentation of the Earth gravity field combination model EIGEN-CG03C is not available yet (as of February 2007) but an application is presented in [7]. An overview of the latest models listed above however, including models EIGEN-CG03C and EIGEN-GL04C, is given in [2].

EIGEN-CG03C is an upgrade of model EIGEN-CG01C using the same CHAMP mission and surface data but taking into account nearly twice as much GRACE mission data: For EIGEN-CG01C 200 days of mission data were processed while for EIGEN-CG03C data from 376 days were available. Both models use 860 days of CHAMP satellite gravity data combined with 0.5 x 0.5 deg surface data (gravimetry and altimetry) for generation of a high resolution global gravity field model. The satellite-only part of EIGEN-GL04C is provided as EIGEN-GL04S1 for applications that require a pure satellite-only gravity model like geodetic and altimeter missions, e.g. for oceanographic applications targeting the precise recovery of sea surface topography features from altimetry. The gravity field combination model EIGEN-GL04C is derived from GRACE and Lageos mission plus 0.5 x 0.5 degrees gravimetry and altimetry surface data and is announced as an upgrade of EIGEN-CG03C. Both combination models use the same surface data but for EIGEN-GL04C the geoid undulations over the oceans have been derived from a new GFZ mean sea surface height (MSSH) model minus the ECCO sea surface topography [4]. The combination with Lageos observations is carried out for stabilization of the long-wavelength part of the model, see [2]. Data with obvious problems have been substituted by EIGEN values, see e.g. figure 2. The improvement of the EIGEN-GL04C compared to the precursor EIGEN-CG03C model is demonstrated in different comparisons and tests, some of which are shown on the GRACE Scientific Results website of the GFZ [4]. The plots in figures 1 and 2 are taken from [4] to exemplify the improvements.

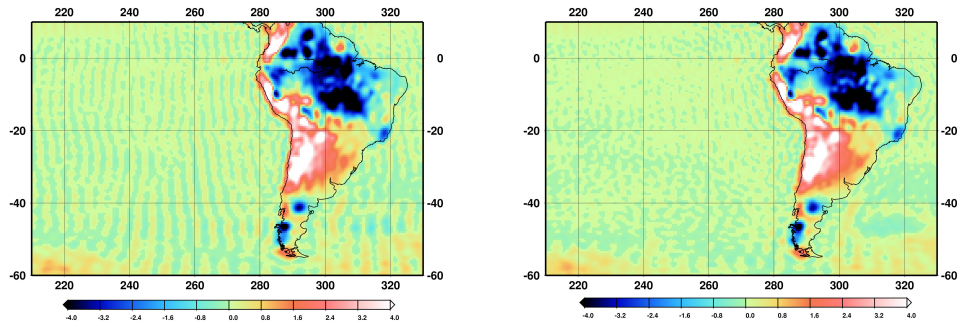


Figure 1: South Pacific geoid heights [m] of EIGEN-CG03C (left) and EIGEN-GL04C (right) after subtraction of a terrestrial gravity data based geoid.

Figure 1 left shows an unrealistic meridional striping pattern in the EIGEN-CG03C model which could be observed in all precursor EIGEN models. The right plot shows that this striping could be much reduced in the EIGEN-GL04C model which is attributed to the new GFZ MSSH data and to improvements in the GRACE satellite-only models.

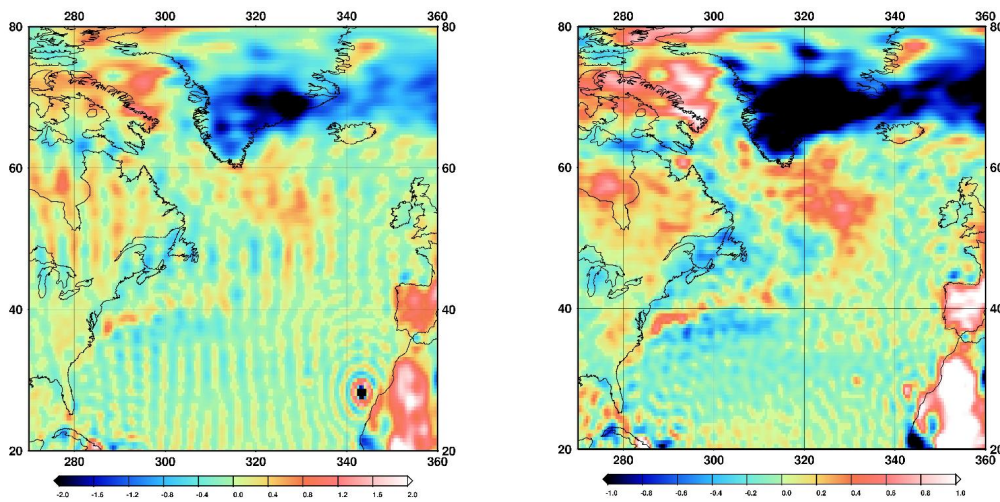


Figure 2: North Atlantic geoid heights [m] of EIGEN-CG03C (left) and EIGEN-GL04C (right) after subtraction of a terrestrial gravity data based geoid (Note the differences in color scale).

Figure 2 shows that the unrealistic ringing pattern in the precursor EIGEN models has been eliminated by substitution of Canary Island NGA gravity anomaly data by EIGEN data.

Gravity Field GGM by CSR

In addition to the gravity field data provided by the GFZ Potsdam, models based on GRACE data are provided by the University of Texas Center for Space Research [18]. The first GRACE gravity model GGM01S released on July 21, 2003 is based upon a preliminary analysis of 111 days of in-flight data gathered during the commissioning phase of the GRACE mission [16]. The model appears to be 10 to 50 times more accurate than all previous Earth gravity models at the long and medium wavelengths. The GGM01S field is estimated up to degree/order 120 but it has been noted that the performance beyond degree/order 90 or so is not so good recommending to use the model up to about degree 95 only. The GGM01S model is comparable to the EIGEN-GRACE01S model from the GFZ but including nearly three times as much data (111 days vs. 39 days of selected GRACE data for EIGEN-GRACE01S). By combination of the GRACE data with marine and land gravity data similar to that used in EGM96 the preliminary field GGM01C was produced extending the degree and order to 200. The successor GGM02 [17] is based on the analysis of 363 days of GRACE in-flight data. This model is available as satellite-only field GGM02S complete to harmonic degree 160 and as combination model GGM02C. It is recommended to use GGM02S not beyond degree 110 as is since rapidly increasing errors make the coefficients unreliable at higher degrees.

Improvement greater than a factor of two over the previous GGM01 is reached with GGM02. The plots in figure 3 taken from the GRACE Gravity Model website of CSR [18] show the improvement from pre-GRACE gravity modelling efforts to GGM01 and GGM02 considering satellite only data. Prior to GRACE, the Earth's gravity field was determined from various tracking measurements of varying quality and incomplete geographical coverage limiting the gravity field resolution to wavelengths of 700km or longer. At shorter wavelengths, the errors were too large to be useful such that only broad geophysical features of the Earth's structure could be detected (figure 3 top). With GRACE data, the satellite-only models could be improved in the long-wavelength as well as the medium wavelength parts. Improvements to the Earth gravity models at medium and short wavelengths have to come from the use of measurements of terrestrial and marine gravity.

GGM02C is extended to degree 200 by constraining it with terrestrial gravity information using a weighted combination of GGM02S and EGM96 spherical harmonic coefficients. Because the data span used for GGM02 is relatively short, the J_2 harmonic was constrained to its long-term mean value from EGM96 incorporating multi-decade satellite and surface data. Otherwise, at the low degrees the surface gravity information was downweighted since the GRACE data is orders of magnitude more accurate.

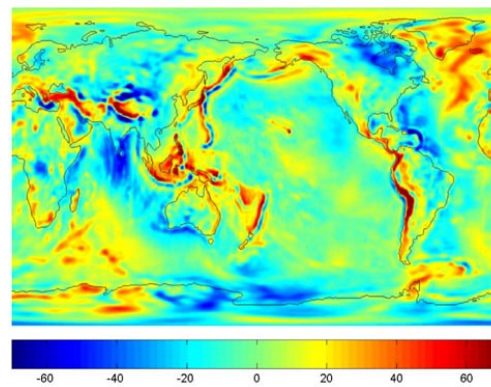
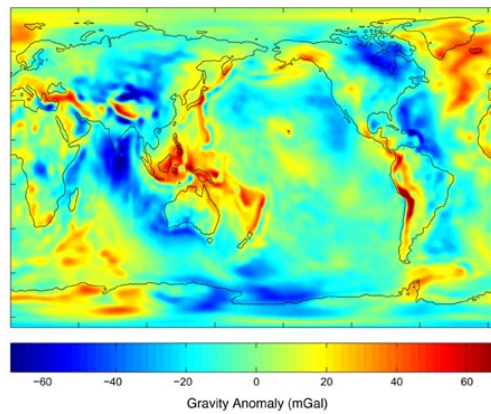
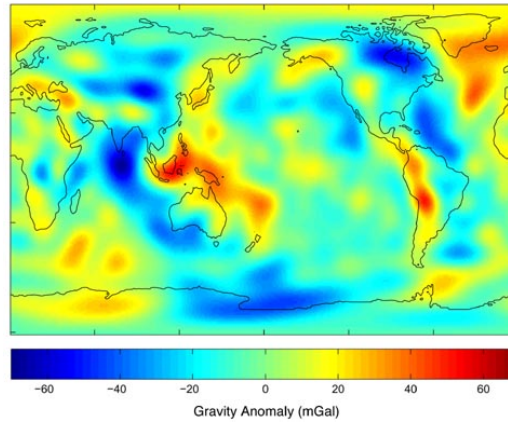


Figure 3: Gravity anomalies from top: decades of tracking Earth-orbiting satellites pre GRACE, middle: GGM01S, bottom: GGM02S

GGM02S has shown solution departure above degree 120, see [17]. Starting at approximately degree 130 the high-degree GGM02S information therefore was downweighted such that the higher degree estimates significantly derive from the EGM96 coefficients. In other words, there is a transition from GRACE-based information at lower degrees to terrestrial gravity information at higher degrees taking place around degree 110 to 120. Since the higher degrees are constrained to EGM96, the model can be smoothly extended to degree and order 360 by filling in the coefficients above 200 with the EGM96 coefficients if a higher degree model is required. The model improvement between GGM01C and GGM02C shows in the reduction of the North-South striations that are also apparent in the predecessors of EIGEN-GL04C. Residual striations indicate that the downweighting of the GRACE information at the higher degrees may still be insufficient; future models will address this issue.

3. Implementation of New Models

From the models described above the newest two combination models EIGEN-GL04C from GFZ and GGM02C from CSR have been included for coefficient upload in the generic drag-free simulator project.

3.1. Additions to the Drag-Free Simulator Project

Write routines for coefficient data sorting to provide the format required by the simulator have been used to generate new include files `coeff_C{name}.inc` and `coeff_S{name}.inc` where `name` is a placeholder for `gl04c` or `ggm02c`.

The corresponding write routines are called `write_coeffGL04C` and `write_coeffGGM02C`. The include files are generated by executing `make write_coeff{name}` (see [10] to learn more about the simulator `make` commands). This command generates the data files `coeff_C{name}.inc` and `coeff_S{name}.inc`. The coefficient data can be found in the svn path `df_simulator/trunk/environment/gravity/f90/include`. Subfolders `egm96`, `ICGEM` and `GGM` hold additional information on the model coefficients, e.g. the data format. The write routines are stored in svn path `df_simulator/trunk/environment/gravity/f90/source`.

3.2. Data Formats

The original coefficient files `g005_eigen_gl04c_coef` for EIGEN-GL04C and `GGM02C.GEO` for GGM02C are also stored in svn path `df_simulator/trunk/environment/gravity/f90/include`.

EIGEN Format

From EIGEN-CG03C to EIGEN-GL04C there has been a format change in data storage. For the latest model EIGEN-GL04C the ICGEM format (a description of which can be found in subfolder ICGEM) supplied with the model data in `g005_eigen_gl04c_coef` has been used. If the format is not changing for future models, the routine `write_coeffGL04C` can be used for future updates; the only modification necessary is the file name containing the spherical harmonic coefficients of the Earth potential field.

The coefficient files contain a header and a data section. The header contains information on authors, release details (year), and model specifications:

- Model name (EIGEN-GL04C)
- Earth's gravitational constant C_MU and Earth's equatorial radius C_R_EARTH with respect to which the spherical harmonic coefficients were computed (for EIGEN-GL04C they are: $C_MU = 0.3986004415 \cdot 10^{15}$ and $C_R_EARTH = 0.6378136460 \cdot 10^7$)
- Maximum degree (360)
- Errors (calibrated)
- Norm (fully normalized)
- Tide system (tide free)

The header end is marked by the keyword `end_of_head`.

In the data section the spherical harmonic coefficients are listed by degree L and order M . Each line is preceded with a keyword like `gfc`, `gfct` or `dot`. Lines beginning without one of these keywords are comments. The number of parameters following these keywords depend on the error specification, i.e. "calibrated", "formal" or "no" (meaning errors are not included). In any case only the first four parameters are of interest, i.e. the degree followed by the order and the spherical harmonic coefficients C and S for keywords `gfc` and `gfct`. For keyword `dot` the parameters following L and M are the temporal derivatives of C and S . They are typically provided for coefficients of degree/order 2/0, 3/0 and 4/0. The write routine therefore needs a keyword filter to only use the lines starting with `gfc` (the difference between `gfc` and `gfct` is that for the latter also a time is given).

GGM Format

The GGM models can be downloaded from the ICGEM homepage [6] in the format described above and also directly from the provider's website [18]. For the drag-free simulator project the latter source is used since the model data format is explicitly given

in the coefficient file which facilitates the coefficient import and reduces the possibilities to introduce errors which can occur if a false data format is assumed.

The header part of `GGM02C.GEO` has three lines, the first line providing the format for the next line and the rate of J_2 employed (from which the rate of C_{20} can be inferred). The second line starts with the model name, the values for C_{MU} and $C_{R.EARTH}$ and the Epoch which are for GGM02C $C_{MU} = 0.3986004415 \cdot 10^{15}$ and $C_{R.EARTH} = 0.6378136300 \cdot 10^7$ (= values used for EGM96), and the Epoch is 2000. The last header line specifies the format for the coefficient data section starting in the next line. From line four on the data is listed starting with keyword `RECOEF` followed by degree, order, the coefficients **C** and **S**, the standard deviations of **C** and **S** and a normalization flag (here: -1 = normalized). The coefficients provided in `GGM02C.GEO` are fully normalized.

3.3. Application and Modifications to the Drag-Free Simulator

In order to use the new model data, the parameter `i_option_g` has to be set to `I_G_OPT_GFZ` (which corresponds to an integer value of 3) for model data `EIGEN-GL04C` or `I_G_OPT_CSR` (4) for GGM02C in the simulator initialization script (in Matlab script `load_sim_data.m`). For integer values equal 2 or higher than 4 for `i_option_g` the current default EGM96 model data is loaded. For use of the new model data for computation of gravity-gradient acceleration, option `i_option_gg_acc` has to be set to `I_GG_ACC_OPT_EGM` as used before for inclusion of the set of spherical harmonic coefficients up to the degree and order specified through options `i_option_g_degree` and `i_option_g_order` where `EGM` = `Earth Gravity Model` is not referring to EGM96 solely but depending on the model coefficients loaded based on option `i_option_g`.

Note:

1. Since the C_{20} coefficient of the GGM02C model is provided as zero-tide value, it is converted to the tide-free system by adding $4.173 \cdot 10^{-9}$ (as specified by the GGM02 notes `GGM02_Notes.pdf` provided with the data). This is done since the previous default EGM96 is disseminated as conventional tide-free model. *However, the use of conventional tide free models like EGM96 does NOT conform to Resolution 16 of the 18th General Assembly of the IAG (1983) recommending that “the indirect effect due to the permanent yielding of the Earth be not removed”, i.e. the use of zero-tide values, cf. [8].* Since the use of zero-tide values provides a more realistic model of the Earth’s gravity field, the new model option `i_option_g_tide` has been introduced to enable that option, i.e. if `i_option_g_tide = I_TIDE_FREE` (this corresponds to an integer value of 0), the gravity field is given with tide-free coefficients and for `i_option_g_tide = I_TIDE_ZERO` (which corresponds to an integer value of 1) the zero-tide system is used. This model option can be enhanced to include other permanent and periodic tidal effects in the future. A definition of the tidal system can be found in appendix A.

2. The value of the reference Earth radius R with respect to which the spherical harmonic coefficients were computed is slightly different for GGM02 and the EIGEN models, i.e. GGM02 uses $R = 6378136.30$ which corresponds to the EGM96 reference radius while the EIGEN data is based on 6378136.46 . When using the EIGEN coefficients the parameter C_R_EARTH which is the simulation reference Earth radius has to be reset to the reference radius used for the derivation of the spherical harmonic coefficients. In the drag-free simulator and the gravity library this is done in the initialization routines.
3. Since the gravity library requires spherical harmonic coefficients up to degree and order 360 and GGM02C is only complete to degree 200, the include files `coeff_Cggm02c.inc` and `coeff_Sggm02c.inc` are patched with EGM96 coefficients for the higher degrees. This is possible since the GGM02C model is by design seamlessly extendable beyond degree/order 200 to 360 using the EGM96 coefficients, see [17].

3.4. Guidelines for Future Model Update

New model coefficients have to be provided in the format required by the gravity library module used in the drag-free simulator. The following steps are necessary to include a new Earth gravity model:

- Create new coefficient files `coeff_C.inc` and `coeff_S.inc` in the format n, m, C_{nm} and n, m, S_{nm} in directory
`df_simulator/trunk/environment/gravity/f90/include`
- Check if the Earth's gravitational constant C_MU and Earth's equatorial radius C_R_EARTH are identical to those with respect to which the spherical harmonic coefficients were computed. If this is not the case, they have to be adapted to the new model by changing the corresponding values in modules `dynamics_params` and `mod1` and Matlab file `dynamics_params.m` in svn paths
`df_simulator/trunk/dynamics/f90/modules`,
`df_simulator/trunk/common/f90/modules` and
`df_simulator/trunk/run/m-files/structures` resp.

A. Tidal Effects on the Earth's Gravity Field

The gravitational potential in the vicinity of the Earth is a combination of the tidal gravitational potential of external bodies like the Sun, Moon and planets, and the Earth's own potential which is perturbed by the external tidal potential. Tidal attraction thus acts in a direct and indirect way: The direct attraction deforms the elastic Earth and thereby causes an indirect change in the gravitational potential. The external tidal and the tide-induced Earth potentials contain both time-independent (permanent) and time-dependent (periodic) parts. The permanent Earth tide is low in the polar regions and high in the equatorial area due to the Moon and the Sun moving fairly close to the equator. In deriving Earth gravity models, the periodic parts are commonly removed, however, various gravity and geoid systems originate from different ways of handling the permanent tide.

A.1. Tide Systems

The different treatment of the permanent tide has led to mainly three tide systems:

- **mean tide:** All permanent tidal distortions are included.
- **zero tide:** The permanent direct effects are removed but the indirect effects related to the elastic deformation of the Earth due to external bodies are kept, i.e. the permanent tidal attraction is eliminated whereas the permanent tidal deformation is retained.
- **non-tidal or tide free:** All tidal effects are removed.

Following these definitions mean gravity includes the permanent tidal attraction caused by masses which are outside the geoid. The application of the different tide systems affect altimeter reduction and geoid definition and parameters like the second degree zonal harmonic of the geopotential and the equatorial radius. If the permanent tidal deformation is eliminated (as in the tide-free system), other quantities to change accordingly are the Earth's moments of inertia, its rotational velocity and its centrifugal force. According to [1], the change in rotational velocity would correspond to shortening the day by 5ms.

According to [12] it is reasonable to remove the tidal attractions due to the Sun and the Moon since these can be directly computed from astronomical tidal theory. It is recommended to precisely consider the permanent tides created by the influences of the Sun and the Moon in satellite and terrestrial analysis. The tide-free system is a theoretical construct in which the gravitational potential is calculated by removing the tidal attraction caused by external (perturbing) bodies AND by allowing the equatorial bulge to relax due to the absence of masses outside the geoid. This results in a redistribution

	Earth Gravity Model Update for the Generic Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-007 Issue: 1.0 Page: 17 of 19
---	---	---

of Earth mass. A truly tide-free quantity is unobservable because the perturbing bodies are always present and it is not known how much the Earth would relax in response to the absence of the Sun, Moon and planets. The tide-free system is characterized by so-called Love numbers which give a measure of how much a planet's surface and interior move in response to the gravitational pull of nearby bodies, i.e. how elastic the planet is. In order to convert from tide free to zero tide it has to be known which Love number was assumed in the system.

References

- [1] M. Ekman. Impacts of geodynamic phenomena on systems for height and gravity. *Bulletin Géodésique*, 63:281–196, 1989.
- [2] F. Flechtner, R. Schmidt, U. Meyer, T. Schöne, S. Esselborn, Ch. Förste, R. Stubenvoll, S. Rudenko, R. König, K.H. Neumayer, and M. Rothacher. The benefit of EIGEN gravity field models for altimetry and vice versa. In *Proceedings of the Symposium on 15 years of Progress in Radar Altimetry*. ESA, CNES, March 2006.
- [3] R. Forsberg and S. Kenyon. Gravity and geoid in the Arctic region - The northern gap now filled. In *Proceedings of the 2nd GOCE User Workshop*. ESA Publication Division, Noordwijk, The Netherlands, 2004. ESA SP-569.
- [4] GFZ Potsdam. *GRACE Scientific Results*. http://www.gfz-potsdam.de/pb1/op/grace/results/index_RESULTS.html.
- [5] Information System and Data Center at GFZ Potsdam. <http://isdc.gfz-potsdam.de/grace>.
- [6] International Centre for Global Earth Models. *The ICGEM Homepage*. <http://icgem.gfz-potsdam.de/ICGEM/ICGEM.html>.
- [7] L. Iorio. The impact of the new Earth gravity model EIGEN-CG03C on the measurement of the Lense-Thirring effect with some existing Earth satellites. *General Relativity and Gravitation*, 38:523–527, March 2006.
- [8] D. D. McCarthy and G. Petit. *IERS Conventions (2003)*. Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie, 2004.
- [9] National Aeronautics and Space Administration. *NASA's Earth Gravity Model 1996 (EGM96)*. <http://cddisa.gsfc.nasa.gov/926/egm96/egm96.html>.
- [10] I. Pelivan. The Modular Drag-Free Simulator. Technical Report FLK-SIM-TN-ZAR-006, Center of Applied Space Technology and Microgravity (ZARM), February 2007.
- [11] Physical Oceanography Distributed Active Archive Center at JPL. *PO.DAAC GRACE Home*. <http://podaac.jpl.nasa.gov/grace>.
- [12] R.H. Rapp, R.S. Nerem, C.K. Shum, S.M. Klosko, and R.G. Williamson. Consideration of Permanent Tidal Deformation in the Orbit Determination and Data Analysis for the TOPEX/Poseidon Mission. Technical Report NASA Tech. Memo. 100775, National Aeronautics and Space Administration - NASA, January 1991.

- [13] C. Reigber, R. Schmidt, F. Flechtner, R. König, U. Meyer, K. H. Neumayer, P. Schwintzer, and S. Y. Zhu. An Earth gravity field model complete to degree and order 150 from GRACE: EIGEN-GRACE02S. *Journal of Geodynamics*, 39:1–10, January 2005.
- [14] Ch. Reigber, P. Schwintzer, R. Stubenvoll, R. Schmidt, F. Flechtner, U. Meyer, R. König, H.-K. Neumayer, Ch. Förste, F. Barthelmes, S. Y. Zhu, G. Balmino, R. Biancale, J.-M. Lemoine, H. Meixner, and J. C. Raimondo. *A High Resolution Global Gravity Field Model Combining CHAMP and GRACE Satellite Mission and Surface Data: EIGEN-CG01C*. Number STR06/07 in Scientific Technical Report. Geoforschungszentrum Potsdam, Potsdam, August 2006.
- [15] S. Scheithauer. Earth Gravity Model 96 – Implementation. Technical Report SIM-ENV-TN-ZAR-001, ZARM, University of Bremen, 2006.
- [16] B.D. Tapley, S. Bettadpur, M.M. Watkins, and Ch. Reigber. The Gravity Recovery and Climate Experiment: Mission Overview and Early Results. *Geophys. Res. Lett.*, 31(9), 2004.
- [17] B.D. Tapley, J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, B. Gunter, P. Nagel Z. Kang, R. Pastor, T. Pekker, S. Poole, and F. Wang. GGM02 - An improved Earth gravity field model from GRACE. *Journal of Geodesy*, 79(8):467–478, 2005.
- [18] The University of Texas at Austin, Center for Space Research. *GRACE Gravity Model*. <http://www.csr.utexas.edu/grace/gravity/>.



Adaptation of the Generic Drag-free
Simulator to the Gravity Probe B
Mission

Doc.No.: FLK-GPB-TN-ZAR-001

Issue: 1.1

Page: 1 of 33

Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission



Project:	Document No.:
FIRST LOOK	FLK-GPB-TN-ZAR-001
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm D 28359 Bremen Germany	Phone: +49-421-218-9422 Fax: +49-421-218-4356 E-Mail: pelivan@zarm.uni-bremen.de

Doc. No.: FLK-GPB-TN-ZAR-001


Issue: 1.1

Written: Ivanka Pelivan

Date: 13th April 2007

Approved: -

Date: -


	<p style="text-align: center;">Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission</p>	<p>Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 2 of 33</p>
---	---	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	12/07/06	all	Initial Version by Ivanka Pelivan	√
1.1	04/13/07	all	Update	√


Contents

1	Introduction	4
2	Conventions	5
2.1	Acronyms and Abbreviations	5
3	Reference Frames and Coordinate Transformations	6
3.1	Elementary Transformations	6
3.2	Reference Frames	6
3.3	Reference Frame Transformations	11
4	Equations of Motion	14
4.1	Satellite Dynamics	14
4.2	Test Mass Dynamics	15
5	Simulator Overview	16
5.1	GP-B Engineering Simulator	16
5.2	Generic Drag-Free Simulator	18
5.3	Adaptation of the Generic Drag-Free Simulator to GP-B	19
5.4	Modifications to the GP-B Engineering Simulator Control Modules	20
6	Simulator Initialization	24
6.1	Set-up Scripts	24
6.2	Initial Conditions	25
6.3	Inputs and Parameters to the Dynamics Core	26
7	Preliminary Results	31
7.1	Comparison between Simulators	31

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 4 of 33
---	--	--

1 Introduction

In combining the drag-free simulator dynamics core developed at ZARM with the control simulator established at HEPL, Stanford University, a number of difficulties arose from uncertainties regarding frame and parameter definitions. This document gives an overview of the coordinate frames used in the drag-free simulator and in the engineering simulator developed at Stanford University along with the coordinate transformations between frames. The step by step simulator adaptation to Gravity Probe B is outlined starting with a listing of the parameters used and the derivation of initial conditions. Finally, the cross-check using simplified simulator models is provided to verify the simulator agreement between the Stanford engineering simulator and the current modelling effort.

	<p style="text-align: center;">Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission</p>	<p>Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 5 of 33</p>
---	---	---

2 Conventions

2.1 Acronyms and Abbreviations

Symbol	Description
<i>COM</i>	Center of Mass
<i>DOF</i>	Degrees of Freedom
<i>GP – B</i>	Gravity Probe B
<i>GSS</i>	Gyro(scope) Suspension System
<i>SC</i>	Spacecraft
<i>TM</i>	Test Mass, gyroscope

3 Reference Frames and Coordinate Transformations

3.1 Elementary Transformations

Coordinate frame transformations often involve elementary transformations about one axis. The elementary transformations are defined positive from the original axes to the new axes. The following transformation matrices are used in the document:

- Elementary transformation by θ about x:

$$\mathbf{T}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (1)$$

- Elementary transformation by ϕ about y:

$$\mathbf{T}_2(\phi) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \quad (2)$$

- Elementary transformation by ψ about z:

$$\mathbf{T}_3(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

3.2 Reference Frames

The types of coordinate frames considered for the dynamics core are listed in table 1.

Table 1: Coordinate Frames for the Dynamics Core

Name	Short Name	Index Symbol
1) Earth-centered inertial frame	ECI, inertial	i , <i>ECI</i>
2) Earth-centered Earth-fixed frame	ECEF, Earth-fixed	<i>ECEF</i>
3) Satellite body-fixed frame (in satellite COM)	body, satellite	b
4) Mechanical body-fixed frame (on satellite structure)	mechanical, structure	m
5) Accelerometer frame	accelerometer	a , <i>acc</i>
6) Sensor frame for test mass *	sensor	<i>sens*</i>
7) Body-fixed frame for test mass *	test mass	<i>tm*</i>

Figure 1 shows all frames that are used for describing the motion of one test mass with respect to a satellite.

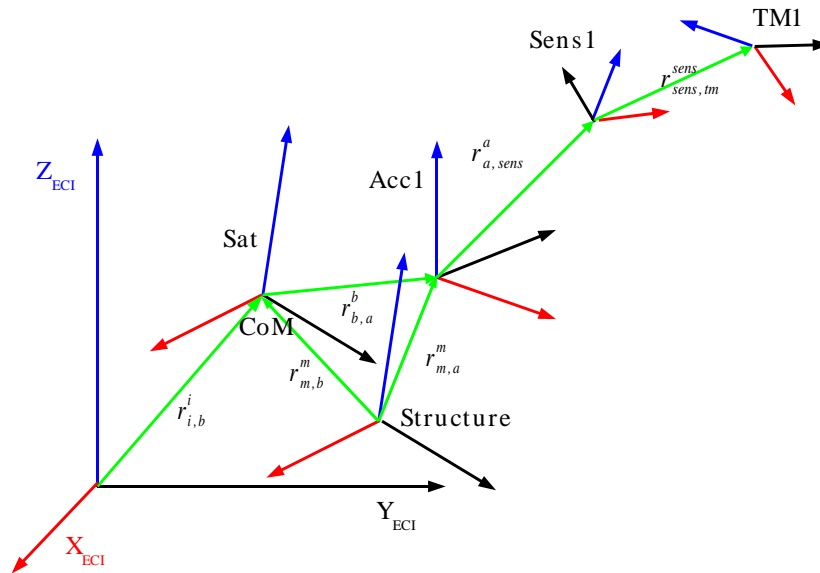


Figure 1: Coordinate Frame Overview for one Accelerometer

In figure 2 the frames listed in table 1 are adapted to the GP-B configuration.

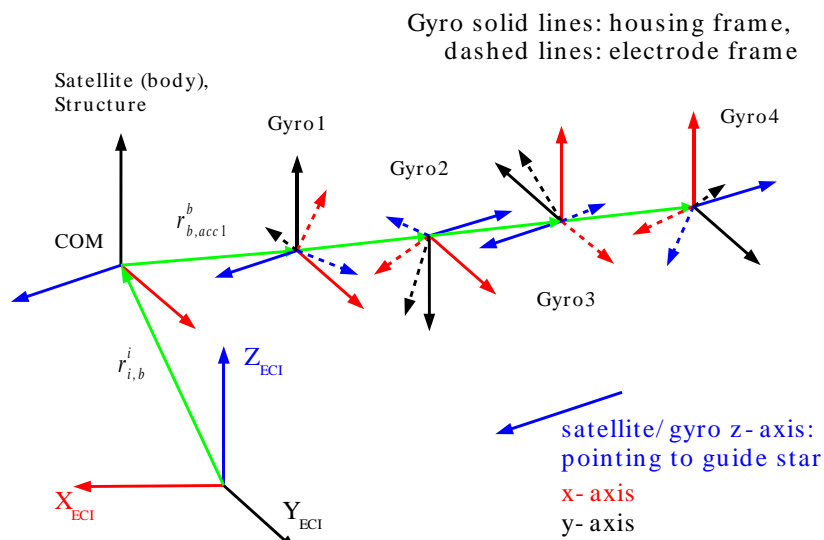


Figure 2: Coordinate Frame Overview for the GP-B Simulator

1) Earth-Centered Inertial Frame

The Earth-centered inertial (ECI) frame is the *Earth Mean Equator and Equinox of J2000 (EME2000)* (see [7] for reference) celestial equator system depicted in figure 3(a). Its definition is based on the IERS (International Earth Rotation Service) standards for the ICRF (International Celestial Reference Frame). The origin of the non-rotating ECI frame is the center of the Earth (the ICRF's origin is located at the barycenter of the solar system). The x -axis points towards the vernal equinox for J2000 and lies inside the celestial equatorial plane. The z -axis is parallel to the Earth's angular momentum vector which is perpendicular to the equatorial plane. The right hand orthogonal system is completed by the y -axis which is also located inside the equatorial plane.

2) Earth-Centered Earth-Fixed Frame

The Earth-Centered Earth-Fixed (ECEF) frame is equal to the International Terrestrial Reference Frame (ITRF) and originates at the Earth's center of mass. This system is also maintained by the IERS. The ECEF frame is a rotating frame but fixed to the Earth's surface, such that it exhibits no net rotation with respect to the Earth's crust. The x -axis lies in the equatorial plane and points towards the Greenwich meridian which is the IERS Reference Meridian (IRM). The z -axis points towards the IERS Reference Pole (IRP) (see figure 3(b)). The y -axis completes the right hand orthogonal system, which lies in the equatorial plane.

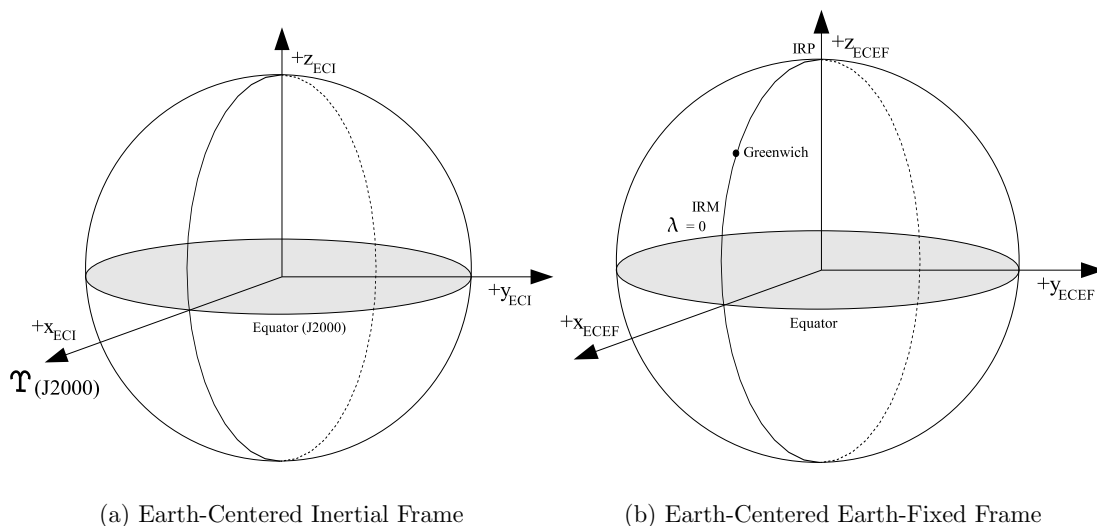



Figure 3: Earth-Centered Frames

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 9 of 33
---	--	--

3) Satellite Body-Fixed Frame

The satellite body-fixed frame has its origin in the center of mass of GP-B's quartz block assembly. It is identical to the housing frame 10) described below (see also table 2).

4) Mechanical Body-Fixed Frame

The origin of the mechanical body-fixed frame is a fixed point on the satellite structure. The axes of the mechanical body-fixed coordinate system are parallel to those of the satellite body-fixed frame (assuming rigid body dynamics). With the mechanical body-fixed frame a moving satellite center of mass (and thus a varying origin for the satellite body-fixed frame 3)) can be accommodated. If the satellite center of mass stays fixed (or is assumed to be fixed), the mechanical body-fixed frame can be chosen to coincide with the satellite body-fixed frame 3) for convenience.

5) Accelerometer Frame

The accelerometer frame is an arbitrarily defined reference frame for describing technical entities like offset and orientation. For GP-B, the accelerometer frame for the first gyroscope (the gyroscope closest to the quartz block COM) is nominally aligned with the satellite body-fixed frame. The other accelerometer frames are rotated w.r.t. the satellite body-fixed frame. The rotations are described in section 3.3. The accelerometer frame is the same as the gyroscope housing frame 11) described below.

6) Sensor Frames

The sensor frame defines the origin and direction for measuring the test mass motion with respect to the satellite. Misalignment and additional offsets inside the differential accelerometer can be introduced. The origin is the point where the sensors which are measuring the test mass displacement output zero for all three coordinates.

The sensor frame is the same as the electrode frame described below using the right-hand convention. The engineering simulator developed at Stanford University is using a left-rotating system.

7) Test Mass Frames

The test mass body-fixed frame describes the attitude of the test mass with respect to the sensor frame. The origin is the center of mass of the test mass. For the GP-B simulator under development the test mass frame is initially aligned with the sensor frame.

The control modules work with the coordinate frames listed in table 2.

Table 2: Coordinate Frames for the Control Modules

Name	Short Name	Index Symbol
8) Nadir frame	nadir	n
9) Inertial guide-star pointing frame	inertial, guide-star fixed	iG
10) Satellite housing frame (body-fixed, in satellite COM)	quartz (block)	q
11) Housing frame for test mass *	gyro housing	h^*
12) Electrode frame for test mass *	electrode	e^*
13) Gyroscope (body-fixed) frame for test mass *	gyro	g^*

8) Nadir Frame

The nadir frame is an Earth center pointing coordinate frame with origin at the spacecraft center of mass. The z-axis always points to the center of Earth, the y-axis points in the negative direction of the nominal orbital angular momentum vector, and the x-axis completes the right-handed set.

9) Guide-Star Frame


The guide star frame is defined w.r.t. the orbit and the direction of the guide star, i.e. the z-axis is pointing to the guide star, the y-axis is in the plane of the orbit pointing north of the equatorial plane and the x-axis is completing the right-hand set.

10) Satellite Housing Frame

The satellite housing frame is a body-fixed frame that has its origin in the center of mass of the quartz block assembly. The GP-B gyroscope drifts were measured w.r.t. this frame.

11) Gyroscope Housing Frames

The gyroscope housings consist of a read-out half (containing the read-out loop) and a spin-up half (containing the spin-up channel). The gyroscope housing frame has its origin in the geometrical center of the gyroscope cavity with its z-axis parallel to the nominal rotor-spin direction determined by the orientation of the spin-up channel and the direction of gas flow in the channel (see [4]). The x-axis is chosen to point away from the read-out half with the y-axis completing a right-handed set s.t. the read-out loop lies in the y/z-plane. In figure 2 the orientation of the gyro housing frames compared to the satellite quartz-block frame are shown.

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 11 of 33
---	--	---

12) Electrode Frames

The location and polarities of the electrodes determine the electrode frame. The electrode axes pass through the centers of the electrodes pointing away from the spin-up half. The origin of the electrode frame is the capacitive center of its gyroscope cavity. The electrodes are arranged symmetrically and such that each unit-vector component of an electrode axis is $\frac{1}{\sqrt{3}}$ along the housing frame x-axis x_H . Therefore, the angle between each electrode axis and the x_H -axis is $90^\circ + \arcsin(\frac{1}{\sqrt{3}}) = 125.264^\circ$. Also, the electrode y-axis lies in the x_H/y_H -plane. The transformation from the gyroscope housing to the electrode frame is given by

$$\mathbf{T}_{E,H} = \mathbf{T}_2(-135^\circ) \cdot \mathbf{T}_3\left(\arcsin\left(\frac{1}{\sqrt{3}}\right)\right) \quad (4)$$

with \mathbf{T}_2 and \mathbf{T}_3 defined in (2) and (3).

Note: The Stanford engineering simulator has not implemented (4) to express a transformation from gyroscope housing to electrode frame but is using the left-rotating frame expressed by the transformation (10) instead.

13) Gyroscope Frames

The gyroscope frames are body-fixed frames that rotate w.r.t. the electrode and gyroscope housing frames.

3.3 Reference Frame Transformations

The satellite body-fixed frames 3) and 10) are identical. The electrode frames 6) and 12) differ in their rotation: 12) is a left-rotating while 6) is a conventional right-rotating coordinate frame. Without satellite rotation frames 9) and 10) would be identical and defined with the third (z-) axis pointing to the guide star, the second (y-) axis pointing away from Earth and the first (x-) axis completing the right-hand set. The controls simulation is set-up such that when starting on top of the Earth's North pole all coordinate frames (8) nadir, 9) inertial, 10) body) coincide ([5]). A transformation from ECI 1) as used in the dynamics core to the initial satellite housing frame 10) (= guide star frame iG) is achieved by applying the transformation

$$\mathbf{T}_{iG,ECI} = \mathbf{T}_{iG,INT} \cdot \mathbf{T}_{INT,ECI} = \mathbf{T}_1(\phi) \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (5)$$

with \mathbf{T}_1 defined in (1) and $\phi = 16.841231 \cdot (\pi/180)$ is the guide star declination in radians. The transformation matrix $\mathbf{T}_{iG,ECI}$ is used to calculate the initial satellite attitude

quaternion for the dynamics core.

The transformation from ECI to guide star frame is visualized in figure 4.

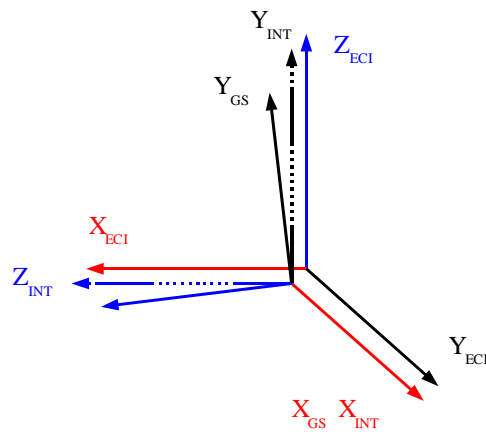


Figure 4: Transformation from ECI to guide-star frame GS through intermediate frame INT

The transformation from satellite body-fixed to sensor frame is realized in the dynamics core by

$$\mathbf{T}_{sens,b} = \mathbf{T}_{sens,acc} \cdot \mathbf{T}_{acc,b} \quad (6)$$

where $\mathbf{T}_{sens,acc}$ is the transformation matrix from accelerometer to sensor frame and $\mathbf{T}_{acc,b}$ is the transformation matrix from satellite body to accelerometer frame. For the first gyroscope, $\mathbf{T}_{acc1,b1}$ is a unit matrix, i.e. the housing frame of the first gyroscope is aligned with the satellite body-fixed frame originating in the quartz block assembly (see figure 2). The other gyroscope housing frames are rotated w.r.t. this frame using the elementary transformations from section 3.1, i.e.

$$\mathbf{T}_{acc2,b2} = \mathbf{T}_1(180^\circ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (7)$$

$$\mathbf{T}_{acc3,b3} = \mathbf{T}_3(90^\circ) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$



$$\mathbf{T}_{acc4,b4} = \mathbf{T}_1(180^\circ) \cdot \mathbf{T}_3(90^\circ) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (9)$$

The transformation matrix from accelerometer to sensor frame (i.e. from gyroscope housing to electrode frame) is a left-rotating frame for the GP-B engineering simulator developed at Stanford:

$$\mathbf{T}_{sens,acc}^L = \begin{bmatrix} -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} & 0 \end{bmatrix} \quad (10)$$

The dynamics core expects right-rotating frames, therefore the following formulation is used:

$$\mathbf{T}_{sens,acc} = \mathbf{T}_{E,H} = \begin{bmatrix} -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} & 0 \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (11)$$

$\mathbf{T}_{E,H}$ is also defined in (4).

The transformation from satellite body to sensor frame for gyroscope * is achieved by multiplying (11) with $\mathbf{T}_{acc*,b*}$ from (7) to (9) for gyroscopes 2, 3 and 4, i.e.


$$\mathbf{T}_{sens2,b2} = \mathbf{T}_{sens,acc} \cdot \mathbf{T}_1(180^\circ) \quad (12)$$

$$\mathbf{T}_{sens3,b3} = \mathbf{T}_{sens,acc} \cdot \mathbf{T}_3(90^\circ) \quad (13)$$

$$\mathbf{T}_{sens4,b4} = \mathbf{T}_{sens,acc} \cdot \mathbf{T}_1(180^\circ) \cdot \mathbf{T}_3(90^\circ) \quad (14)$$

For the first gyroscope,

$$\mathbf{T}_{sens1,b1} = \mathbf{T}_{sens,acc} \cdot \quad (15)$$

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 14 of 33
---	--	---

4 Equations of Motion

The satellite states are computed in the inertial and satellite body-fixed reference frames 1) and 3)/10) while the test mass states are calculated w.r.t. the sensor and test mass frames 6) and 7). The equations of motion solved in the dynamics core are based on [6] and [11].

4.1 Satellite Dynamics

The satellite translation basically is the motion of a rigid body in a gravity field with additional coupling forces between the satellite and test masses,

$$\ddot{\mathbf{r}}_{i,b}^i = \mathbf{g}_{i,b}^i(\mathbf{r}_{i,b}^i) + \mathbf{a}_{\text{control}}^i + \mathbf{a}_{\text{dist}}^i + \mathbf{a}_{\text{coupl,sat}}^i \quad (16)$$

where

$\mathbf{r}_{i,b}^i$	satellite position
$\mathbf{g}_{i,b}^i$	gravity acceleration
$\mathbf{a}_{\text{control}}^i$	control accelerations
$\mathbf{a}_{\text{dist}}^i$	external disturbances
$\mathbf{a}_{\text{coupl,sat}}^i$	acceleration due to the satellite - test mass coupling.

The attitude motion of the satellite is expressed by the angular velocity of the satellite body w.r.t. the inertial frame, $\omega_{i,b}^b$. The attitude itself is described by the Euler symmetric parameters \mathbf{q}_i^b which represent a transformation from inertial frame to satellite body fixed frame. The differential equations for the satellite attitude motion are as follows:

$$\dot{\omega}_{i,b}^b = (\mathbf{I}_b^b)^{-1} [\mathbf{T}_{\text{control}}^b + \mathbf{T}_{\text{dist}}^b - \mathbf{T}_{\text{coupl,sat}}^b - \omega_{i,b}^b \times (\mathbf{I}_b \omega_{i,b}^b)] \quad (17)$$


where

$\omega_{i,b}^b$	angular velocity of satellite
$\dot{\omega}_{i,b}^b$	angular acceleration of satellite
\mathbf{I}_b^b	inertia matrix of satellite
$\mathbf{T}_{\text{control}}^b$	control torques
$\mathbf{T}_{\text{dist}}^b$	disturbance torques
$\mathbf{T}_{\text{coupl,sat}}^b$	torques generated from satellite-test mass coupling,

$$\dot{\mathbf{q}}_i^b = \frac{1}{2} \dot{\omega}_{i,b}^b \otimes \mathbf{q}_i^b, \quad (18)$$

with

$\dot{\omega}_{i,b}^b$	quaternion representation of angular velocity
\otimes	quaternion multiplication.

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 15 of 33
---	--	---

4.2 Test Mass Dynamics

The equations of motion for the test mass dynamics are derived in the sensor frame which is satellite fixed.

The acceleration of the test mass w.r.t. the satellite expressed in the inertial frame can be obtained by subtracting the satellite equation of motion (16) from the test mass motion. After transformation into the sensor frame, the equations of motion of a test mass relative to the satellite become:

$$\ddot{\mathbf{r}}_{b,tm}^{\text{sens}} = \Delta \mathbf{g}_{b,tm}^{\text{sens}}(\mathbf{r}_{b,tm}^{\text{sens}}) + \mathbf{a}_{\text{coupl},tm}^{\text{sens}} - \mathbf{a}_{\text{coupl},\text{sat}}^{\text{sens}} + \mathbf{a}_{\text{rotation}}^{\text{sens}} - \mathbf{a}_{\text{control}}^{\text{sens}} - \mathbf{a}_{\text{dist}}^{\text{sens}} \quad (19)$$

where

$\Delta \mathbf{g}_{b,tm}^{\text{sens}}$	gravity gradient acting on the test masses
$\mathbf{a}_{\text{coupl},\text{sat}}^{\text{sens}}$	coupling acceleration due to satellite-test mass coupling
$\mathbf{a}_{\text{coupl},tm}^{\text{sens}}$	coupling acceleration due to test mass - test mass coupling for more than one test mass
$\mathbf{a}_{\text{rotation}}^{\text{sens}}$	rotation in sensitive axes frame
$\mathbf{a}_{\text{control}}^{\text{sens}}$	acceleration due to thruster firing
$\mathbf{a}_{\text{dist}}^{\text{sens}}$	acceleration due to external disturbances.

For more details on the derivation of Eq. (19) and the single acceleration terms see [11].

The test mass attitude is expressed in the test mass frame and derived based on the equation of conservation of angular momentum. The resulting equations for the test mass rotation write:

$$\dot{\hat{\omega}}_{\text{sens},tm}^{\text{tm}} = (\mathbf{I}_{tm}^{\text{tm}})^{-1} \mathbf{T}^{\text{tm}} - (\mathbf{I}_{tm}^{\text{tm}})^{-1} (\omega_{i,b}^{\text{tm}} + \omega_{\text{sens},tm}^{\text{tm}}) \times (\mathbf{I}_{tm}^{\text{tm}} (\omega_{i,b}^{\text{tm}} + \omega_{\text{sens},tm}^{\text{tm}})) - \dot{\omega}_{i,b}^{\text{tm}} \quad (20)$$

where

$\omega_{i,b}^{\text{tm}}$	angular velocity of satellite w.r.t. inertial frame
$\omega_{\text{sens},tm}^{\text{tm}}$	angular velocity of test mass w.r.t. sensitive axes frame
$\dot{\omega}_{i,b}^{\text{tm}}$	angular acceleration of satellite w.r.t. inertial frame
$\mathbf{I}_{tm}^{\text{tm}}$	inertia matrix of test mass
\mathbf{T}^{tm}	torques due to rotational coupling and gravity gradient,

$$\dot{\mathbf{q}}_{\text{sens}}^{\text{tm}} = \frac{1}{2} \hat{\omega}_{\text{sens},tm}^{\text{tm}} \otimes \mathbf{q}_{\text{sens}}^{\text{tm}} \quad (21)$$

5 Simulator Overview

The drag-free simulator developed at ZARM is assembled in a generic modular way to allow for adaptation to very different science missions that require a precise model of satellite and experiment. The generic modules include environment and disturbance models to account for external effects due to gravitation, magnetic field, atmosphere, and solar pressure as well as models for the satellite itself and its sub-systems. The control modules usually are mission specific. A control simulator has been established at HEPL, Stanford University. The control modules including a gyroscope suspension system, drag free controller, and attitude controller, are combined with the generic dynamics, environment and disturbance modules. The generic simulator thereby is adapted to Gravity Probe B for validation purpose, model improvement and knowledge enhancement regarding future science missions.

5.1 GP-B Engineering Simulator

The GP-B engineering simulator has advanced capabilities including hardware-in-the loop to test flight software and hardware interfaces. A software version of this simulator has been established for other purposes. In this project the software version shown in figure 5 is used for simulator verification.

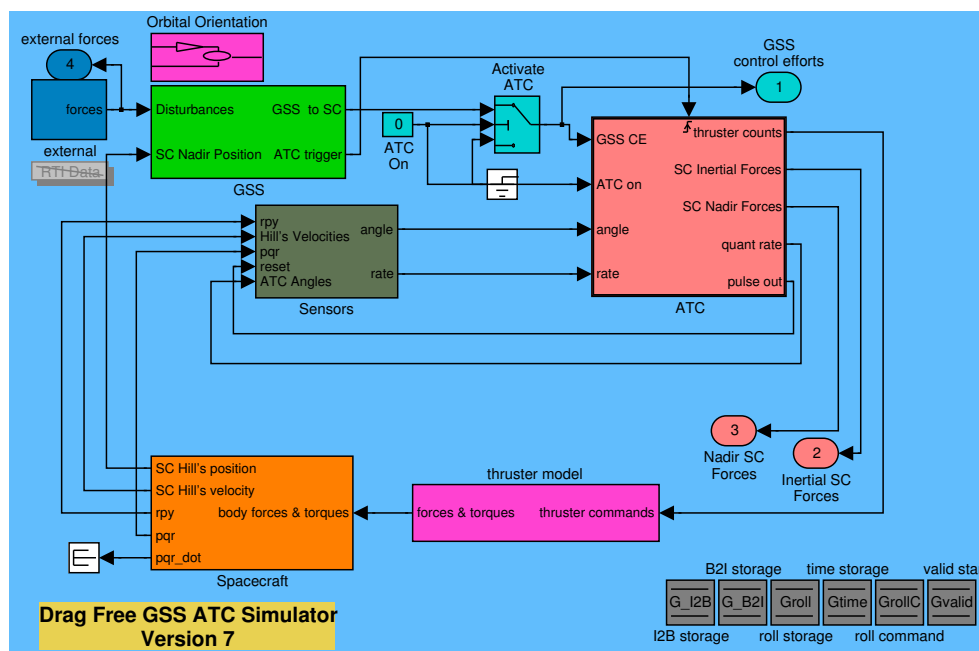



Figure 5: Gravity Probe B Engineering Simulator

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 17 of 33
---	--	---

The simulator depicted in figure 5 is fully functional in having its own dynamics and models for some (environmental) disturbances. To model the spacecraft and gyroscope dynamics Hill's equations are employed. The engineering simulator therefore can be used for cross-check of the generic simulator if simplified dynamics are used and environmental disturbances are neglected (this cross-check is carried out in section 7.1).

The control modules (Gyroscope Suspension System, Sensors, ATC, Actuators) are combined with the drag-free simulator modules. A brief description of the control modules is given below. A slightly more detailed description of these modules and the generic modules in the next section is given in [9] and [10].

The Gyroscope Suspension System

The GP-B gyroscopes are electrostatically suspended by applying AC voltages to three perpendicular sets of electrodes, thus enabling the gyros to spin freely within their quartz housings. Simulated suspension is based on the difference in housing cavity position and rotor position. The controller incorporated is a 3-axis-AOD (Authority on Demand) controller [4] with an adaptive LQG (Linear Quadratic Gaussian) type of control. The model further accounts for charge effects, i.e. charge built up on the gyro from radiation and other sources effects the forces applied by the suspension system. The rotor is discharged by a UV light. In simulation, the nature of the charge and the UV light response can be defined in a matlab script.


The ATC

The simulated attitude and position control system incorporates the same control logic as in the ATC code used for flight during the science mission including two separate PID controllers for the attitude and translation control. The control gains are the same gains as used in flight.

Control effort from the Gyroscope Suspension System (GSS) is passed to the ATC module for translation control. Attitude control uses sensor measurements from rate gyro, telescope and star tracker to keep the roll rate constant and other rates and position errors zero. The attitude control module includes an observer which is used instead of the telescope when the guide star is eclipsed by the Earth. When the guide star is in view the pointing errors are determined by the telescope. The commanded forces from translation control and the commanded torques from attitude control are passed on to the actuator as thruster counts.

Sensors

The sensor output for translation control is provided by the GSS. For attitude determination, three other sensor sets are utilized: 1) A telescope, pointing at a guide star (IM Pegasus for GP-B) gives accurate measurements for attitude in the perpendicular

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 18 of 33
---	--	---

plane of the sensor. The telescope model also accounts for orbital and annual aberrations. White noise is combined with the telescope signal at the same variance as found in the flight telescope. 2) Integrating rate gyros are used on GP-B to measure the three angular rates of the spacecraft. Simulated rate gyros integrate the simulated attitude at a rate of 10Hz. Drift rates estimated from flight data during GP-B's science mission are included. 3) A simulated star tracker updates the attitude measurements twice per roll.

Actuators

The 16 proportional helium thrusters on GP-B are modeled each as a point force equal to the specific impulse from on-orbit data. The thrust counts from the ATC module are converted to forces and torques based on the combination of thrusters used. Conversion from ATC commands to thruster counts and thrust distribution is realized utilizing the same logic as on the actual spacecraft.


5.2 Generic Drag-Free Simulator

The generic modules, including the dynamics core, environmental and disturbance models, are described below.

The Dynamics Core

The satellite and test mass dynamics are computed via an s-function calling the Fortran dynamics core. The satellite and test mass dynamics are solved in six degrees of freedom by numerical integration of the equations of motion. The default integration method is a 5th order Runge-Kutta scheme. The satellite states are computed in the inertial and satellite body-fixed reference frame while the test mass states are calculated w.r.t. the test mass and sensor frame. The equations of motion solved in the dynamics core are given in section 4. In solving the spacecraft and test mass equations of motion the dynamics core includes the following features:

- Consideration of linear and nonlinear coupling forces and torques between satellite and test masses as well as between multiple test masses
- Modeling of cross-coupling interaction
- Earth gravity model up to 360th degree and order
- Gravitational influence of Sun, Moon and planets
- Gravity-gradient torques and attitude-dependent gravity-gradient forces
- Choice of integration methods (Runge-Kutta, Bulirsch-Stoer, Euler-Cauchy)

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 19 of 33
---	--	---

For the coupling forces, an internal model providing spring and damping parameters can be used as long as the coupling can be assumed linear. Alternatively, a variable-size external link input is provided. For GP-B, the coupling forces from the GSS module are passed to the dynamics core through the link input. Other disturbance and control actions on the satellite enter through inputs for external forces and torques.

Environmental Models

The Environment block is divided in spacecraft environment and test mass environment. The spacecraft environment contains models for the atmospheric density, solar flux, magnetic field and albedo. The gravitational influence is contained as a library in the dynamics module as driving force for the satellite and gyroscope dynamics.

The atmospheric density model is based on the Mass Spectrometer Incoherent Scatter Radar Extended (NRLMSISE-00) model [8] provided by the Naval Research Laboratory. This model has been enhanced by a short-term density variation model (DVM) based on the data from the CHAMP satellite mission [12] to improve the high frequency characteristics.

For most environmental disturbances a structural model of the satellite needs to be built and divided into subelements or subvolumes. For the solar flux model the flux on each element is calculated and summed to obtain the total. A visibility check is carried out beforehand to determine back faces and shadowed areas to reduce the computation time.

To model the Earth magnetic field the IGRF (International Geomagnetic Reference Field) released by the International Association of Geomagnetism and Aeronomy (IAGA) is used [1].

The Earth Albedo model according to [3] is developed from reflectivity data taken from the NASA project TOMS (Total Ozone Mapping Spectrometer) which is given online [2]. A statistical model of the reflectivity is developed from data averages and standard deviations derived for each latitude.

The calculation of environmental disturbances is done in pre-processing and look-up tables or parametric models are utilized during the orbit simulation.

The Disturbance Module

The Disturbance module is structured like the Environment module, i.e. the Disturbance block is divided in spacecraft and test mass disturbances. The Disturbance module transforms the environmental outputs into forces and torques.

5.3 Adaptation of the Generic Drag-Free Simulator to GP-B

The generic drag-free modules are combined with the control modules from the GP-B Engineering Simulator described above in 5.1. In figure 6 the new GP-B simulator is

shown.

In addition to the modules described in the previous section, transformation libraries are added since the various modules require a representation of states, forces and torques in different coordinate systems. The forces and torques transformation block carries out the necessary transformations for quantities that are not provided in the required coordinates, i.e. the inputs for the spacecraft dynamics in inertial, orbital and mechanical reference frames and for the gyroscope dynamics in the sensitive axes frame. In the state selection module the inputs for all modules but the dynamics core are selected and necessary coordinate transformations are carried out.

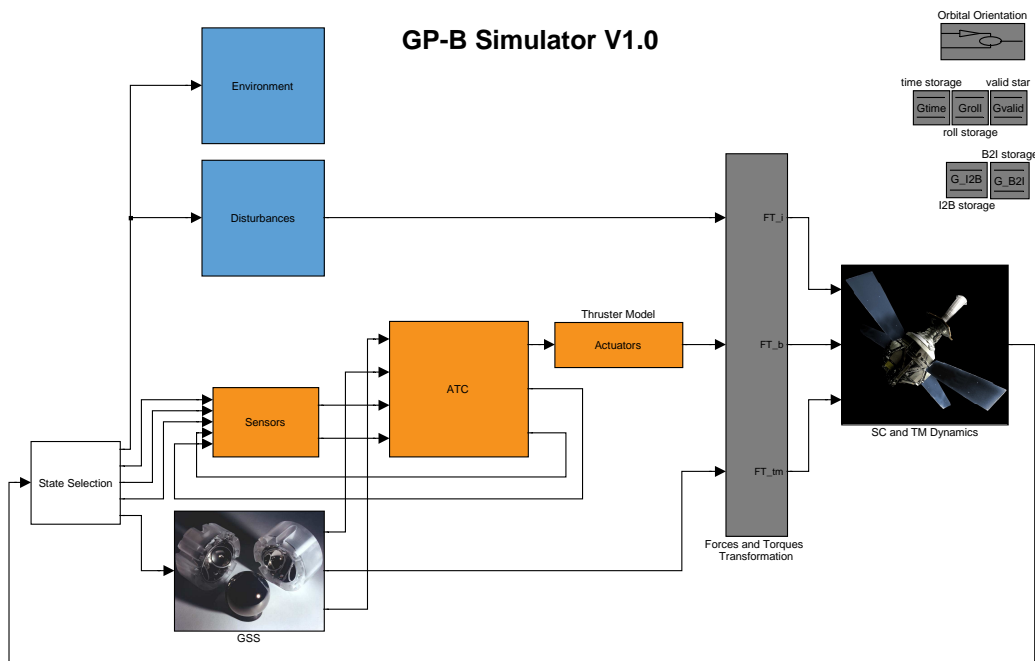


Figure 6: Gravity Probe B Simulator Integrating Generic and Mission-Specific (Control) Modules

5.4 Modifications to the GP-B Engineering Simulator Control Modules

In the following, the module contents comprising the control part are shown in more detail. In figure 7 the sensor and ATC modules are depicted. Both modules are identical

to those from the engineering simulator from figure 5. The ATC subsystem includes the ATC activation logic.

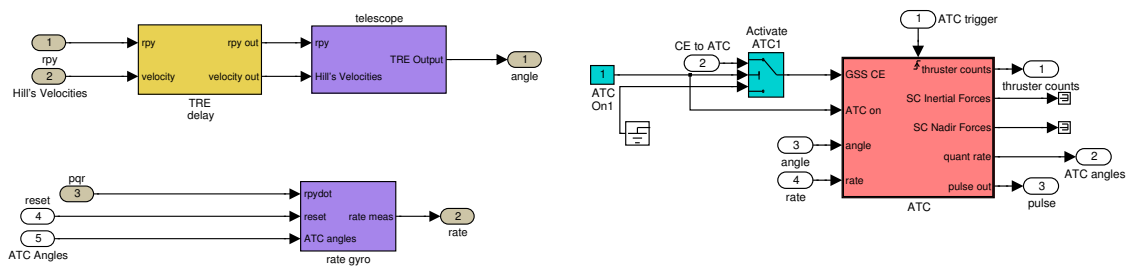


Figure 7: Modules Sensors (left) and ATC (right)

In figure 8 the thrust actuation is shown. This module is almost identical to the GP-B Engineering Simulator thruster model except that the spacecraft null bias is subtracted at this point before entering the dynamics module via the forces and torques transformation.

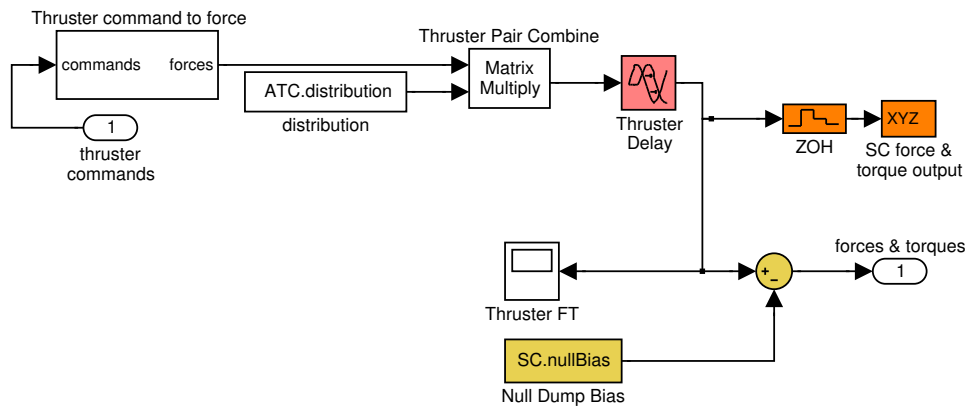


Figure 8: Module Actuators

The GSS module is shown in figure 9. It contains the software model from the GP-B Engineering Simulator, a rotor charge simulation and necessary transformations for the control efforts passed on to the ATC. Also at this point external gyroscope forces and polhode motion are added to the gyroscope suspension forces.

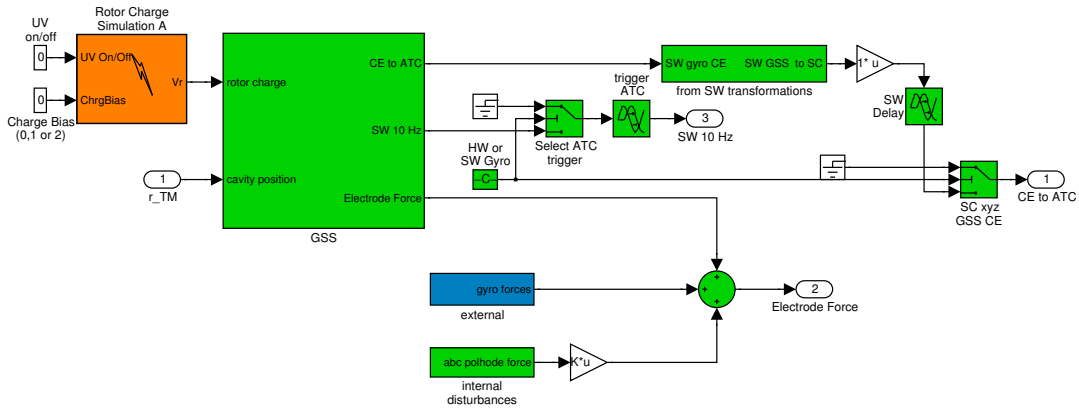


Figure 9: Gyroscope Suspension System

The GSS module from the GP-B Engineering Simulator includes its own dynamics based on Hill's equations, see figure 10. The GSS block in figure 9 is taken from the GSS subsystem used in the Engineering simulator shown in figure 10; the other blocks are omitted and the cavity position input (labelled r_{TM} in figure 9) now is provided by the dynamics core module.

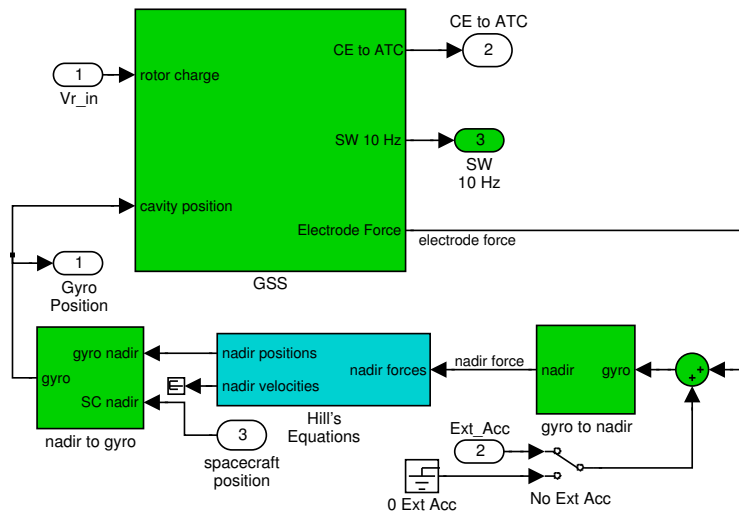


Figure 10: GP-B Engineering Simulator Gyroscope Suspension System Dynamics Model

In the state selection block (see figure 11) necessary transformations for the control inputs (and environmental inputs), mainly for the sensors module, are carried out.

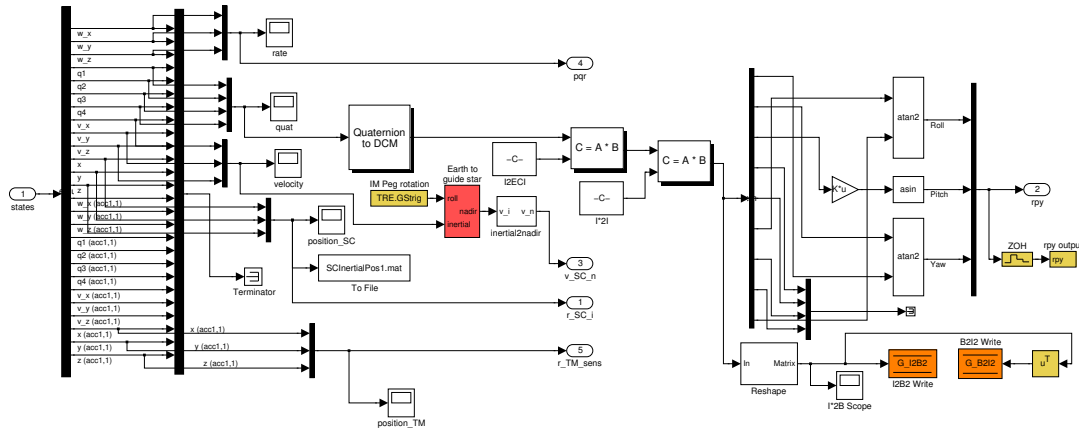


Figure 11: State Selection

The forces and torques transformation block connecting the control modules to the spacecraft and gyroscope dynamics is just a pass-through for the forces and torques coming from the actuators and the GSS since they are already in satellite body-fixed and sensor frame coordinates.

6 Simulator Initialization

The dynamics and control integrated simulator is set-up in accordance with the original GP-B engineering simulator developed at the W.W.Hansen Experimental Physics Lab, Stanford University for cross-check. The simulator initialization and set-up is carried out through data files. These scripts are listed and explained in the following.

6.1 Set-up Scripts

Each module in the simulator has its own set-up script(s). Since environmental and disturbance modules are not incorporated in this preliminary investigation, only the initialization files for the dynamics core and the control modules are described briefly in the following. The environment and disturbance modules will be added at a later point.

The data files for simulator core set-up are the Matlab scripts listed in table 3.


Table 3: Matlab Simulator Initialization Scripts

Name	Function
<code>load_sim_data.m</code>	<i>Simulator options:</i> model name, input/output options, time options, integration options, gravity options
<code>load_sat_data.m</code>	<i>Satellite options:</i> DOF, initial satellite states
<code>load_acc_data_[1..4].m</code>	<i>Accelerometer and test mass options:</i> initial test mass states, displacements and transformations to satellite body, test mass properties (mass, inertia) and geometrical entries, DOF, constant link parameters (stiffness, damping, DC inputs)

These data files use parameters provided by the setup files for the original engineering simulator (see table 4).

Table 4: Matlab Parameter Scripts

Name	Function
<code>setupDFsim.m</code>	<i>GP-B parameters;</i> initializes the drag free control loop models. Sets simulation time, sets drag-free gyro, spacecraft orbit period, rate, mass properties for spacecraft and gyroscope, spacecraft roll, delays, gyroscope offsets from quartz block COM, transformation matrices quartz block to electrode frames based on the left-rotating transformation in (10), Helium properties amongst others. This file calls <code>asim.m</code> , <code>ATCattSys.m</code> , <code>thrustModel.m</code> , <code>sensors.m</code>


	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 25 of 33
---	--	---

asim.m	<i>Parameters for structures GSS, SC, SUBU, dSpace and AOD; calls constants.m, control gains and biases, transfer function parameters, AOD gain change, excitation parameters, noise amplitudes amongst others</i>
constants.m	<i>Constants in structures GSS, SC, SUBU, dSpace and AOD: gyro mass, radius, electrode angle, voltage gains and biases, Earth constant for spacecraft, initial spacecraft position and velocity, spacecraft inertia amongst others</i>
ATCattSys.m	<i>ATC parameters; sets the ATC control loop parameters for attitude control: ATC roll rate, period, inertia, limits, biases, filter frequencies, damping ratios</i>
thrustModel.m	<i>Thruster parameters (SC and ATC); mex functions wrappsinv.c and GGtorq.c biases, thruster modes, thruster distribution, gains amongst others</i>
sensors.m	<i>Structures CGYRO and TRE; initializes sensor model parameters by calling telescope_setup.m and control_gyro_setup.m</i>
telescope_setup.m	<i>Telescope (related) parameters: guide star properties, Earth orbit properties (mean anomaly, longitude, eccentric obliquity), transformation matrix to convert values in J2000 coordinate system to GP-B guide star coordinate system</i>
control_gyro_setup.m	<i>Parameters for the control gyroscope sensor model: start and end of guide star valid angle, pitch and yaw rate offsets, roll, pitch and yaw rate noise</i>

At simulation start, these parameters have to be initialized as well. The satellite and test mass initial conditions and parameters are detailed below.

6.2 Initial Conditions

The initial conditions for satellite and gyroscopes are set according to those which are used in the Stanford engineering simulator for cross-check. The engineering simulator utilizes Hill's equations for calculation of the dynamics. Initial conditions for the engineering simulator are provided in nadir coordinates. Therefore, transformations from nadir to ECI and satellite-body fixed frames are necessary for the satellite states and a transformation from nadir to gyroscope sensor frame has to be carried out for the test mass states.

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 26 of 33
---	--	---

Satellite Initial Conditions

Initial conditions for the satellite can be calculated from orbital elements or entered directly. For simulator cross-check, the initial conditions used in an engineering simulator reference case are adopted.

Test Mass Initial Conditions

In the preliminary investigation the gyroscopes are assumed to be non-rotating. Therefore, the rotation rate is set to zero.

6.3 Inputs and Parameters to the Dynamics Core

The parameters for the simulation are set in three different files for the simplified case simulating one gyroscope only, see also table 3,

- `load_sim_data.m`
- `load_sat_data.m`
- `load_acc_data_1.m`

Simulation Inputs

The non-zero simulation inputs, provided in Simulink constant blocks, are the mass and moments of inertia of the satellite:

- Mass of the satellite in $[kg]$: `SC.mass = 2822`
- The moments of inertia are set to $SC.inertia = I_b^b = \begin{bmatrix} 5061 & 0 & 0 \\ 0 & 4988 & 0 \\ 0 & 0 & 3514 \end{bmatrix} kg\ m^2$

Also, the gyroscope suspension forces enter the s-function block for the dynamics core. At a later stage, also forces and torques from the thruster action is supplied.


Simulation Parameters

The simulation parameters can be found in Matlab file `load_sim_data.m`:

- `df_sim.const_links = 0`
Enables external link input for coupling between satellite and gyroscopes.
- `acc_opt.acc_id = 1`
Additional outputs for accelerations on the gyroscope of accelerometer 1.



- `df_core_param.d_MJD = 53249`
The chosen number can be arbitrary since it has no influence for this case.
- `df_core_param.d_t_step = GSS.period/10 = 4.5455 · 10-4`
`GSS.period = 1/GSS.cpuRate, GSS.cpuRate = 220`
- `df_sim_time.d_t_start = 0`
- `df_sim_time.d_t_final = dSpace.endSim = 100`
End time of simulation.
- `df_core_param.d_eps_dp = 1.0E-10`
Default.
- `df_core_param.d_h_min_dp = 1.0E-40`
Default.
- `df_core_param.d_h_1_dp = GSS.period/10`
- `df_core_param.i_option_g = constant_params.I_G_OPT_SIMPLE`
Simple gravity model utilizing up to six zonal coefficients.
- `df_core_param.i_option_g_order = 0`
Using the simplest field: spherical Earth.
- `df_core_param.i_option_g_degree = 1`
Using the simplest field: spherical Earth.
- `df_core_param.i_option_g_cog = 0`
Correction for gravity force acting in center of gravity instead of center of mass neglected.
- `df_core_param.i_option_g_planets(i) = 0`
No influence of other celestial bodies.
- `df_core_param.i_option_gg_model = constant_params.I_GG_OPT_J2`
Gravity-gradients based on oblate Earth.
- `df_core_param.i_option_ggt = constant_params.I_GGT_OPT_MATRIX`
Gravity gradient torques on satellite calculated from gravity-gradient matrix
- `df_core_param.i_option_gg_acc = constant_params.I_GG_ACC_OPT_SPHERE`
Using the simplest field for gravity-gradient accelerations on test mass.
- `df_core_param.i_option_ggt_tm = constant_params.I_GGT_OPT_OFF`
No gravity gradient torques on test masses.

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 28 of 33
---	--	---

Satellite Parameters


The satellite parameters can be found in Matlab file `load_sat_data.m`:

- `df_core_param.d_num_acc = 1`
One single accelerometer is chosen.
- `df_core_param.i_option_sat_dof = constant_params.I_ALL_FREE`
The satellite moves in all degrees of freedom.
- `df_core_state.da_y(1:2) = 0`
- `df_core_state.da_y(3) = SC.rollW = 0.0811`
The initial attitude rate `df_core_state.da_y(1:3)` has to be provided in the satellite body-fixed frame.
- `df_core_state.da_y(4:7)`
The initial attitude quaternions of the satellite are calculated by the Matlab function `att2quat` which transforms a direction cosine matrix into a quaternion. The input to this function is the transformation matrix (5) from ECI to guide star frame on top of the orbit. Since the starting point of the simulation is chosen such that initially nadir, guide star and satellite body fixed frames collapse, the transformation matrix $\mathbf{T}_{b,i} = \mathbf{T}_{iG,ECI}$ from (5) provides the relationship between ECI and satellite body fixed frame at start of the simulation.
- `df_core_state.da_y(8:10) = $\mathbf{T}_{i,b} \cdot (0, 0, SC.orbitV)^T$`
The initial satellite velocity has to be provided in the inertial (ECI) frame. $\mathbf{T}_{i,b}$ is the transpose of (5) and $(0, 0, SC.orbitV)^T$ is the initial condition which is given for the GP-B engineering simulator in satellite body-fixed coordinates.
- `df_core_state.da_y(11:13) = $\mathbf{T}_{i,b} \cdot (0, SC.orbitR, 0)^T$`
The initial satellite position has to be provided in the inertial (ECI) frame. $\mathbf{T}_{i,b}$ is the transpose of (5) and $(0, SC.orbitR, 0)^T$ is the initial condition given for the engineering simulator in satellite body-fixed coordinates.

Accelerometer Parameters

For one single accelerometer the parameter file `load_acc_data_1.m` is used. The gyroscope simulated actually is the proof mass which is the third gyroscope in line after the satellite COM position. For the complete simulation including all four gyroscopes the proof mass data file will be `load_acc_data_3.m`.

The accelerometer file holds the properties for two test masses. Since GP-B has one gyroscope per accelerometer, the (zeros) listing of parameters for the second test mass will be omitted. The parameters for constant link inputs are all zero since the external

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 29 of 33
---	--	---

link input from the gyroscope suspension system is used. Therefore the link parameters will also not be listed in the following.

- `i_acc = 1`
Local variable specifying the number of the accelerometer the parameter file is used for.
- `i_tm = 1`
Local variable specifying the test mass for which the states are currently initialized. First it is set to 1 and the initial states for the first test mass are defined, then it is set to 2 and the states for the second test mass are initialized. For GP-B, only one test mass is present for the single accelerometer.
- `i_base`
Index variable, not to be changed!
- Initial angular velocity of first test mass:
`df_core_state.da_y(i_base + 1) = 0`
`df_core_state.da_y(i_base + 2) = 0`
`df_core_state.da_y(i_base + 3) = 0`
 No angular velocity for the test mass.
- Initial attitude of first test mass:
`df_core_state.da_y(i_base + 4) = 0`
`df_core_state.da_y(i_base + 5) = 0`
`df_core_state.da_y(i_base + 6) = 0`
`df_core_state.da_y(i_base + 7) = 1`
 Test mass is aligned with sensor frame.
- Initial velocity of first test mass:
`df_core_state.da_y(i_base + 8:i_base + 10) =`
 $\mathbf{T}_{sens3,b3} \cdot [(\text{SC.orbitW}, 0, 0)^T \times \text{GSS.cm}\{3\}]$
 In the GP-B engineering simulator the test mass dynamics are calculated using Hill's equations. The initial test mass velocity is zero in the nadir frame. For the dynamics core the initial test mass velocity has to be provided in sensor frame coordinates. The offset of the gyroscope from the satellite COM is provided by the gyroscope position $\text{GSS.cm}\{3\} = (0, 0, -0.405)^T$ in satellite body-fixed coordinates. The initial test mass velocity in satellite body-frame coordinates is determined by taking the cross-product of the orbit rate $\text{SC.orbitW} = 0.0011$ (along the x-axis, note that the x-axes for nadir, orbital, inertial and also initially for satellite body-fixed frames collapse, see section 3.3) and the gyroscope position. The velocity is transformed to the sensor frame by utilizing the transformation matrix $\mathbf{T}_{sens3,b3}$ from (13).



- Initial position of first test mass:
 $df_core_state.da_y(i_base + 11) = 0$
 $df_core_state.da_y(i_base + 12) = 0$
 $df_core_state.da_y(i_base + 13) = 0$
No offset of test mass.
- $df_core_param.s_accelero1.d_num_tm = 1$
Number of test masses.
- Position of accelerometer frame w.r.t. satellite-fixed frame:
 $df_core_param.s_accelero1.da_r_m_ma(1:3) = GSS.cm\{3\}$
- $df_core_param.s_accelero1.da_T_a_b$
For simplicity the 3×3 transformation matrix is set to $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- $df_core_param.s_accelero1.sa_test_mass1.d_mass = main.gyroMass = 0.0633$
Mass of test mass 1 (= gyroscope 3) in $[kg]$
- $df_core_param.s_accelero1.sa_test_mass1.da_moi$
The matrix is set to $\begin{pmatrix} 9.1880 \cdot 10^{-8} & 0 & 0 \\ 0 & 9.1880 \cdot 10^{-8} & 0 \\ 0 & 0 & 9.1880 \cdot 10^{-8} \end{pmatrix} kg\ m^2$
- $df_core_param.s_accelero1.sa_test_mass1.da_moi_inv$
Inverse of the test mass inertia matrix.
- $df_core_param.s_accelero1.sa_test_mass1.da_r_a_asens = [0,0,0]$
- $df_core_param.s_accelero1.sa_test_mass1.da_T_sens_a = T_{sens3,b3}$
Since $df_core_param.s_accelero1.da_T_a_b$ is set to the identity matrix, the transformation from satellite body to test mass sensor frame is carried out here by utilizing the matrix defined in (13).
- $df_core_param.s_accelero1.sa_test_mass1.da_switch_trans_DOF(1) = 1$
 $df_core_param.s_accelero1.sa_test_mass1.da_switch_trans_DOF(2) = 1$
 $df_core_param.s_accelero1.sa_test_mass1.da_switch_trans_DOF(3) = 1$
Allows translational motion.
- $df_core_param.s_accelero1.sa_test_mass1.da_switch_rot_DOF(1) = 1$
 $df_core_param.s_accelero1.sa_test_mass1.da_switch_rot_DOF(2) = 1$
 $df_core_param.s_accelero1.sa_test_mass1.da_switch_rot_DOF(3) = 1$
Allows rotations. However, for the first simulation approach there is no rotation since the initial angular velocity of the test mass $df_core_state.da_y(i_base + 1: i_base + 3)$ is set to zero.

7 Preliminary Results

7.1 Comparison between Simulators

A cross-check using simplified simulator models is provided to verify the simulator agreement between the GP-B engineering simulator and the current modelling effort. The software version of the GP-B engineering simulator was used to produce the data for comparison. As mentioned above, the engineering simulator uses simplified dynamics based on Hill's equations. The dynamics for the extended generic simulator were therefore also simplified in assuming a spherical Earth and modeling the gravity-gradient on the spacecraft based on a spherical Earth potential field including the first zonal harmonic only.

Figures 12 and 13 show the position coordinates for the drag-free gyroscope. The results from the GP-B engineering simulator are shown with a red dashed line and the blue curves labeled full dynamics refer to the generic simulator including the dynamics core described in subsection 5.2 but simplified as outlined in the previous paragraph.

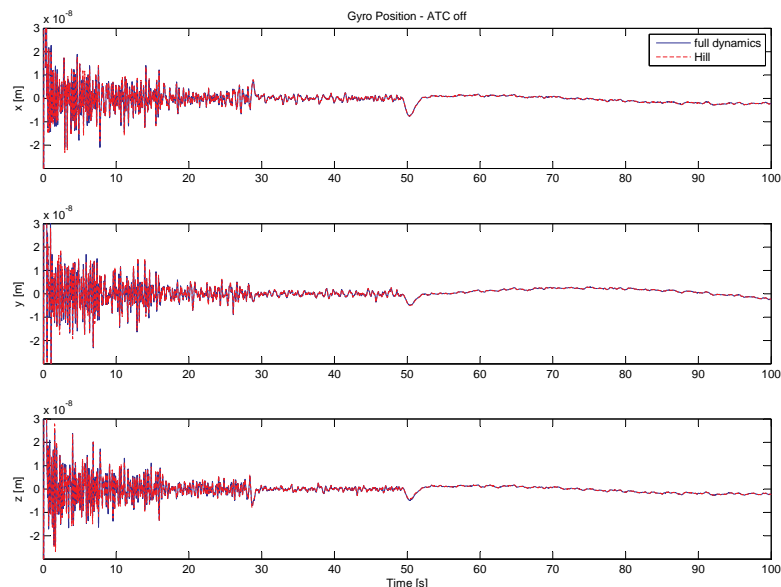


Figure 12: Gravity Probe B Simulator Comparison - Gyro Position, ATC off

In a first modelling effort (figure 12), only the GSS is combined with the generic dynamics

core. In Figure 13 the ATC, sensor and thruster models are included. In both cases the agreement is near perfect.

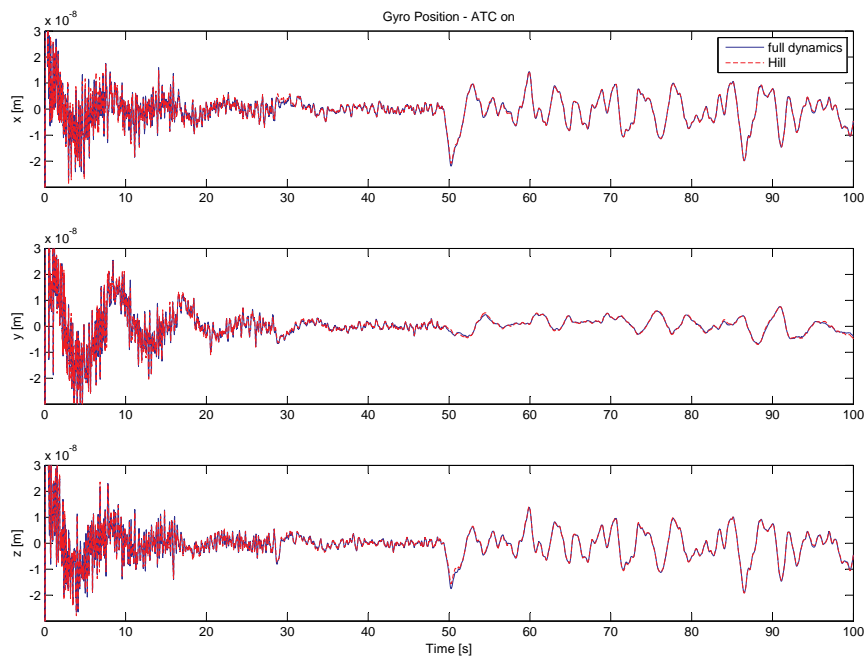




Figure 13: Gravity Probe B Simulator Comparison - Gyro Position, ATC on

Note: In generating the above results, the s-function GGtorq calling GGtorqF.c has been modified. Presumably a gravity-gradient disturbance has been imposed on the dynamics to test the control capabilities. For an accurate simulation the gravity-gradient matrix in GGtorqF.c has been sign-corrected in this study and the missing transformation from ECEF to guide star frame is included.

	Adaptation of the Generic Drag-free Simulator to the Gravity Probe B Mission	Doc.No.: FLK-GPB-TN-ZAR-001 Issue: 1.1 Page: 33 of 33
---	--	---

References

- [1] *International Geomagnetic Reference Field (IGRF)*. <http://nssdc.gsfc.nasa.gov/space/model/models/igrf.html>.
- [2] *Total Ozone Mapping Spectrometer (TOMS)*. <http://toms.gsfc.nasa.gov/>.
- [3] P. Appel, S. Theil, S. Winkler, and A. Schleicher. Attitude Estimation from Magnetometer and Earth-albedo-corrected Coarse Sun Sensor Measurements. In *5th International Conference on Spacecraft Guidance, Navigation and Control Systems; Frascati, Italy*, October 2002.
- [4] M.L. Eglinton. *Authority-On-Demand Adaptive Suspension Control for the Gravity Probe B Gyroscopes*. PhD thesis, Department of Aeronautical and Astronautical Engineering of Stanford University, August 2000.
- [5] T. Holmes. private conversation, 2006.
- [6] Benjamin Lange. *The Control and Use of Drag-Free Satellites*. PhD thesis, Department of Aeronautics and Astronautics of Stanford University, June 1964.
- [7] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits - Models, Methods, Applications*. Springer-Verlag, 2000.
- [8] National Aeronautics and Space Administration. *NRLMSISE-00 Model 2001*. <http://nssdc.gsfc.nasa.gov/space/model/atmos/nrlmsise00.html>.
- [9] I. Pelivan, S. Smoot, S. Theil, and Y. Ohshima. Modeling of Dynamics and Control for Gravity Probe B. In *AAS/AIAA Space Flight Mechanics Meeting, Sedona, Arizona*, February 2007.
- [10] S. Smoot, I. Pelivan, Y. Ohshima, and S. Theil. High Fidelity Controls Simulation for Gravity Probe B. In *30th Annual AAS Guidance and Control Conference, Breckenridge, Colorado*, February 2007.
- [11] Stephan Theil. *Satellite and Test Mass Dynamics Modeling and Observation for Drag-free Satellite Control of the STEP Mission*. PhD thesis, Department of Production Engineering, University of Bremen, December 2002.
- [12] M. Zijlstra, S. Theil, and S. Scheithauer. Model for Short-term Atmospheric Density Variations. In *2nd CHAMP Science Meeting, Potsdam, Germany*, September 2003.

	<p>GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation</p>	<p>Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 1 of 27</p>
---	---	---

GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation

<p>Project:</p> <p style="text-align: center;">First Look Project</p>	<p>Document No.:</p> <p style="text-align: center;">FLK-ALG-TN-ZAR-002</p>
<p>ZARM - Center of Applied Space Technology and Microgravity</p> <p>Am Fallturm Phone: +49-421-218-4796 D 28359 Bremen Fax: +49-421-218-2521 Germany E-Mail: samaan@zarm.uni-bremen.de</p>	

Doc. No.: FLK-ALG-TN-ZAR-002


Issue: 1.0

Written: Malak A. Samaan Date: 28th June 2007

Approved: - Date: -

Contents

1	Introduction	3
2	Attitude Measurement and Pointing Requirements	3
3	Attitude Data Simulation	4
3.1	The Star Tracker Model	6
3.2	The Angular Rates Model	8
4	The Extended Kalman Filter	9
4.1	The Kalman Filter Results	12
4.2	Smoothing Estimation Technique	14
5	The Batch Attitude Estimation	16
5.1	The Maximum Likelihood Parameters Estimation	17
5.2	The Batch Least Squares Results	18
6	Gaia Disturbance Torques	22
7	Conclusion and Future Work	25

	<p>GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation</p>	<p>Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 3 of 27</p>
---	---	---

1 Introduction

In this document we design and analyze different algorithms representing the operational mode of the On-Ground Attitude Determination System (ADS) for the GAIA mission using the On-Board measurements sensor data and the Kalman filtering. The principal feature of the GAIA astrometry mission is a highly stable payload consisting of two scientific instrument telescopes with one big focal plane containing an array of 106 CCDs. In order to achieve this highly stable requirements the GAIA spacecraft should have very accurate attitude determination sensors and also very accurate control actuators. The objectives of this document are:

1. Represent inputs from the different GAIA system design and requirement documents for generating a complete set of the mission attitude parameters for certain period of time.
2. Define and assume the on-board ADS requirements imposed by the scientific requirements and specify the Gaia ADS interfaces with the external elements.
3. Design and analyze different techniques representing the On-Ground ADS using the data given by the star sensor measurement and the scientific instrument rates.
4. Validate the results from the ADS algorithms with the GAIA mission attitude and rate errors requirements and choose the more accurate algorithm for the On-Ground ADS.

2 Attitude Measurement and Pointing Requirements

The measurement and pointing requirements for on-board Operational-Mode attitude determination and control of the GAIA satellite is described in some detail in (**GAIA-EST-RD-00553**)[1], and (**GAIA-SRD-001**)[2]. The most recent determination and control performance requirements are summarized in ESA GAIA Mission Requirements Document [1], section 4.7. Table 1 summarizes the principal mission- and system-level requirements imposed on the attitude pointing and rate performances as provided. These requirements are applicable to both ASTRO telescope Line-Of-Sight (LOS). The GAIA astrometric measurement principle and CCD operation makes that a distinction must be made in the accuracy requirements for quantities measured in the along-scan and those along the across-scan directions.

Where the definitions in table 1 are [3]

- **Attitude Measurement Error (AME)** refers to the instantaneous 3-axis angular separation between the estimated satellite attitude (as well as Astro instrument LOS) and the actual one.

Parameter	Requirement (99.73 % probability level)
AME - Attitude Measurement Error	< 20 arcsec
RME1 - Rate Measurement Error (along scan)	< 0.9 mas/s
RME2 - Rate Measurement Error (across scan)	< 2.7 mas/s
APE - Absolute Pointing Error	< 60 arcsec
RPE1 - Relative Pointing Error (along scan)	< 5 mas
RPE2 - Relative Pointing Error (across scan)	< 10 mas
MRE1 - Mean Rate Error (along scan)	< 2 mas/s
MRE2 - Mean Rate Error (across scan)	< 10 mas/s

Table 1: Summary of AOCS Performance Requirements

- **Rate Measurement Error (RME)** refers to the mean difference between the estimated spacecraft scan rate and the actual one.
- **Absolute Pointing Error (APE)** refers to the instantaneous 3-axis angular separation between the desired satellite attitude (as well as Astro instrument LOS) and the actual one.
- **Relative Pointing Error (RPE)** at a given time t is defined as the standard deviation of the absolute pointing error over the AF CCD integration time τ around t .
- **Mean Rate Error (MRE)** is defined as the mean difference between the desired satellite scan rate attitude and the actual one; the averaging time is the time spent by an object from being detected to being confirmed in the astrometric field.

3 Attitude Data Simulation

The Nominal Scanning Law (NSL) [4] used on-board GAIA satellite defines the reference pointing attitude for the measurements of the two astrometric telescopes fields of view on the sky. This measurement principle relies on the systematic and repeating observation of the star positions in two fields of view. For this purpose, the spacecraft slowly rotates at a constant angular rate of 1 deg/min around an axis perpendicular to those two fields of view, describing a great circle on the sky in 6 hours. However, the real attitude motion will not be exactly along the directions prescribed by the NSL but they will have error angles in all three directions.

The scan axis is subjected to a slow precession at a rate (mean value over the year) of 0.173 arcsec/sec about the Sun to Earth direction. Furthermore, the angle between the scan axis and the Sun must be maintained at 45 degrees.

Using the rigid body dynamics and the kinematic equations [7], the GAIA attitude data in the form of the spacecraft quaternions and angular velocities are simulated for one great circle revolution (6 hours).

Figure (18) shows the true spacecraft angular velocities (arcsec/sec) using the NSL parameters.

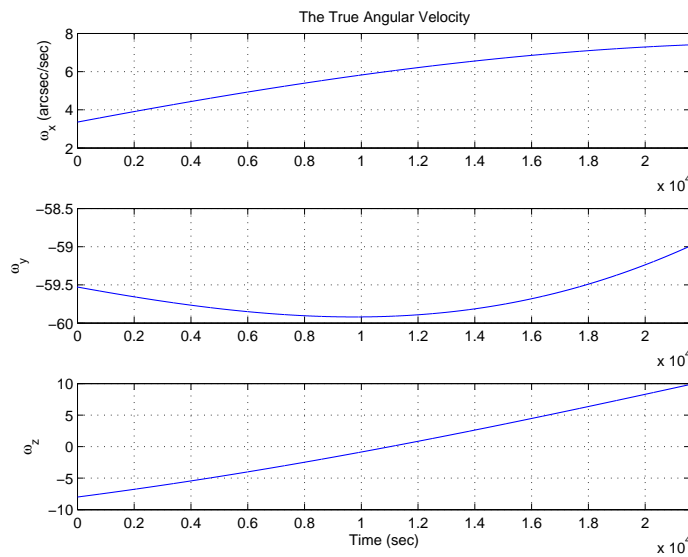


Figure 1: The True Spacecraft Angular Velocities

The spacecraft attitude is assumed to be represented by the quaternion vector \mathbf{q} , defined as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{13} \\ q_4 \end{bmatrix}$$

With

$$\mathbf{q}_{13} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\mathbf{n}} \sin \theta/2 \text{ and } q_4 = \cos \theta/2$$

Where $\hat{\mathbf{n}}$ is a unit vector corresponding to the axis of rotation, and θ is the angle of rotation.

The elements of the quaternion satisfy a unit norm constraint. $q^T q = q_{13}^T q_{13} + q_4^2 = 1$
 The quaternion kinematics equations of motion are derived by using the spacecraft's angular velocity (ω), given by

$$\frac{d}{dt} \mathbf{q} = \frac{1}{2} \Omega(\omega) \mathbf{q}$$

where $\Omega(\omega)$ and are defined by

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}$$

The initial spacecraft quaternions q_o is assumed to be compliant with the Nominal Scanning Law (**SAG-LL-0144**) [4]. Figure (20) shows the true (reference) spacecraft quaternions for for one great circle duration using the ODE45 to integrate the quaternion kinematics equations of motion.

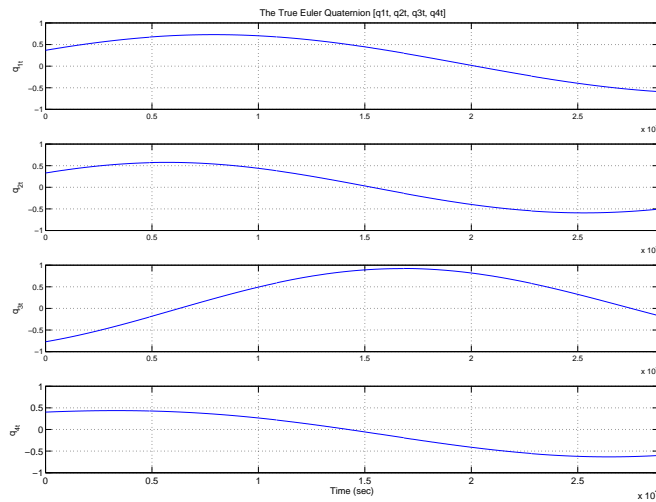


Figure 2: The true (reference) spacecraft quaternions

3.1 The Star Tracker Model

The star tracker model with some measurement errors and bias are used to simulate the measurement quaternions. The star tracker specifications are given in [5] (**GAIA.ASU.SP.ESM.00010**) some of these specifications are required in the simulation like:

- The measurement frequency: **2 HZ** (time intervals 0.5 sec)
- The Optical axis alignment w.r.t. GAIA body frame: [0.7071068241 ; -0.6123723733 ; 0.3535534125]

- The measurement errors should be better than: **13** arc-sec (3σ) around the star tracker X and Y axes and **15** arc-sec (3σ) around the star tracker Z axes
- The star tracker FOV is **assumed to be 8 x 8 deg.**
- The magnitude threshold for the star tracker is **assumed** to be 5.5

The quaternion multiplication can be used to translate from the true spacecraft body quaternion to the star tracker quaternions and also to add the attitude errors to the star tracker quaternions as shown by the following equation

$$\mathbf{q}'' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \mathbf{q}$$

Where \mathbf{q}' is a quaternion that translates from quaternion \mathbf{q} to quaternion \mathbf{q}''

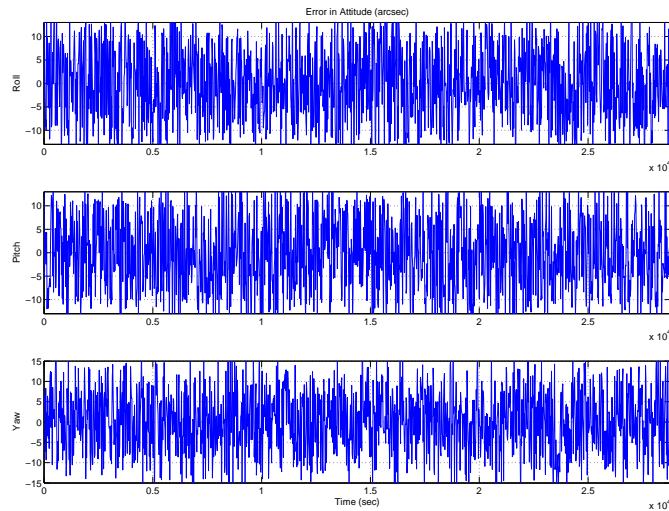



Figure 3: The Star Tracker Attitude Errors Measurements

Figure (3) shows the roll, pitch and yaw attitude errors between the true spacecraft attitude and the measured attitude from the star tracker model. We could notice from figure (3) that as the yaw-axis of image coordinate system coincides with the bore-sight axis of the star tracker camera, less information is available for the rotation about the yaw-axis, which corresponds to higher covariance in the yaw angle estimates.

	GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation	Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 8 of 27
---	--	--

3.2 The Angular Rates Model

The scientific payload is used as a high accuracy stellar gyro for achieving the attitude determination and enabling Time Delay Integration (TDI) mode. The star speed is calculated along and across scan from Astrometric Sky Mapper (ASM) and Astrometric Field (AF) measurements. When considering the two telescopes, one obtains four angular speed measurements, from which the satellite rigid body angular speed vector can be calculated.

Since the star flow is fixed and is determined by the instrument sensitivity and the star density, the attitude reconstruction will no longer be effective in the high frequency domain, typically above 0.1 Hz [6] (**GAIA-CH-TN-EADS-FS-001-1**). Therefore, sudden attitude change must be avoided and the high frequency attitude error must be maintained at a level of a few micro-arcseconds during science measurements.

It should be mentioned that ASM-1 is seen by telescope No. 1 and ASM-2 by telescope No. 2. The Sky Mapper measurements consist of the determination of the centroid crossing times of the stars over each of the ASM CCD arrays. These crossing times are combined with the subsequent crossing times of the same stars over the AF-1 CCD array, which is about 10.69 and 5.83 seconds for stars in FOV1 and FOV2, respectively. These measurements can be processed to estimate the speeds of the stars, both in the along and the across directions, relative the instrument focal plane. The rate measurements accuracy along scan is equal to 2.7 marcsec/sec (3σ) and across scan is equal to 14.3 marcsec/sec (3σ).

The attitude angles and rotation rate errors (difference between measurements and commands) are filtered using a *low-pass filter*, in order to reject high frequency content higher than the cut-off frequency. The cut-off frequency of this filter for the angular rate measurement for the GAIA scientific instrument is equal to

$$f = \frac{1}{2\Delta T}$$

where ΔT is the time between the subsequent crossing times of the same star.

Figure 4 shows a block diagram for a low pass filter used to filter the true simulated angular rates with the following transfer function

$$\frac{\omega_{LBF}}{\omega_T} = \frac{2\pi f}{s + 2\pi f}$$

For the purpose of simulations, we assumed that the angular rate measurements from the GAIA on-board scientific instruments have the following model

$$\begin{aligned}\omega_{meas} &= \dot{\theta}_{meas} = \omega_{LPF} + b + \eta_1 \\ \dot{b} &= \eta_2\end{aligned}$$

Where b is the rate bias vector $[b_x, b_y, b_z]^T$, $\dot{\theta}_{meas}$ and ω_{true} are the measured and true rate of change of the spacecraft orientation respectively. Both η_1 and η_2 are zero-mean, white noise processes and they are independent.

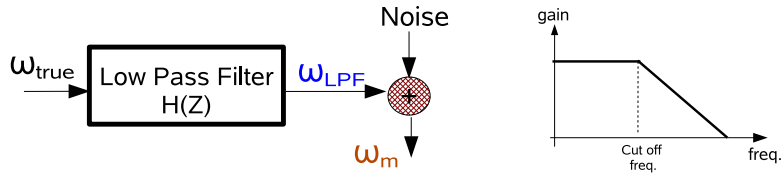


Figure 4: The Low-Pass Filter for the Simulated Angular Velocity

The rate bias is simulated using standard deviation for gyro noise $\eta_1 = 14.3e^{-03} \text{ arcsec/sec}$ and standard deviation for bias $\eta_2 = 3e^{-6} \text{ arcsec/sec/sec}$. The sample rate for the rate gyro is assumed to be 0.5 sec . The error between the true and the measured angular velocity is shown in figure (5).

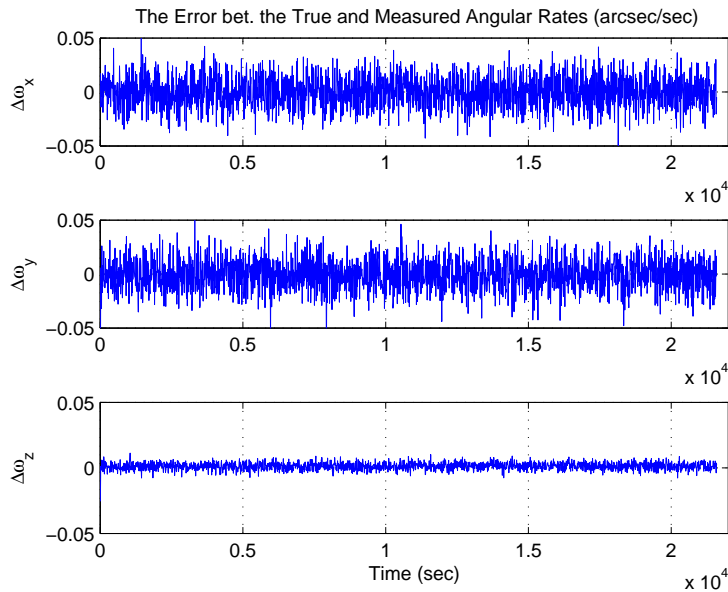


Figure 5: The Error bet. the True and Measured Angular Velocity

4 The Extended Kalman Filter

The Kalman filter is a means of obtaining an optimal estimate of satellite attitude given a dynamic model, sensor measurements, and noise characteristics of the sensors and actuators. Kalman filter have been widely applied to the spacecraft attitude determination problem. The Kalman filter minimizes the trace of the covariance of the estimated error between the model response and the actual measurements in the least squares sense, but

due to its recursive structure. A residual covariance analysis can be used to check the consistency of the resulting attitude errors. The main advantage of using the Kalman filter for attitude determination is that it optimally combines the rate data with the attitude sensor data to produce more accurate results [8].

The Kalman filter state vector for the GAIA ADS is comprised of ten components: the 4 components of the attitude quaternions, the 3 components of the Gyro bias and the 3 components of the angular rates.

$$x = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad b_x \quad b_y \quad b_z \quad \omega_x \quad \omega_y \quad \omega_z]$$

The system model equation has the following form

$$\dot{x}(t) = f(x(t), t) + G(x(t), t)w(t)$$

Also, the measurement model at time t_k is given by

$$Y_k = h(x(t_k), t) + \nu(t)$$

Where $w(t)$ is the process noise and $\nu(t)$ is the measurement noise, both are discrete Gaussian white-noise processes

$$\begin{aligned} w(t) &= \sim N(0, Q(t)) \\ \nu(t) &= \sim N(0, R(t)) \end{aligned}$$

The Kalman filter dynamic equations of state are given by

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \frac{1}{2}\Omega(\omega)\mathbf{q}(t) \\ \dot{\mathbf{b}}(t) &= \eta_2 \\ \dot{\omega}(t) &= I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \end{aligned}$$

Where I_{sc} is the moment of inertia for GAIA and T_e is the total disturbance and control torques acting on the spacecraft. the Kalman filter propagation equations are defined by

$$\begin{aligned} \Delta x(t) &= F(t)\Delta x(t) + G(t)w(t) \\ \dot{P} &= \hat{F}P + P\hat{F}^T + \hat{G}Q\hat{G}^T \end{aligned}$$

For the case of no external or control torques ($T_e = 0$), the F matrix is given by

$$F = \begin{bmatrix} 0.5\Omega(\hat{\omega}) & 0_{4 \times 3} & -0.5\Xi(\hat{\mathbf{q}}) \\ 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} & F_{\hat{\omega}\omega} \end{bmatrix}$$

Where

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

$$F_{\dot{\omega}\omega} = -I_{sc}^{-1} ([\omega \times] I_{sc} - [I_{sc} \omega \times])$$

The matrix $[\underline{a} \times]$ is the skew symmetric matrix of the vector \underline{a} .

Also, the Kalman filter update equations are defined by

$$\begin{aligned} \hat{x}_k(+) &= \hat{x}_k(-) + K_k [Y_k - h_k(\hat{x}_k(-))] \\ P_k(+) &= [I - K_k H_k(\hat{x}_k(-))] P_k(-) \\ K_k &= P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1} \end{aligned}$$

The measurement vector at each time step $Y_k(7 \times 1)$ consists of the simulated star tracker measurement quaternions and the simulated angular rate measurements from the GAIA on-board scientific instrument filter.

$$h(x_k) = \begin{bmatrix} \mathbf{q} \\ \omega \end{bmatrix} = \begin{bmatrix} q_{st} + \eta_{st} \\ \omega_{meas} + b + \eta_1 \end{bmatrix}$$

The angular velocities rates is assumed to have the same frequency as the star tracker measurement (2 Hz). The measurement sensitivity matrix is given by

$$H_k = \frac{\partial h(x(t_k), t)}{\partial x(t)} = \begin{bmatrix} I_{4 \times 4} & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 4} & I_{3 \times 3} & I_{3 \times 3} \end{bmatrix}$$

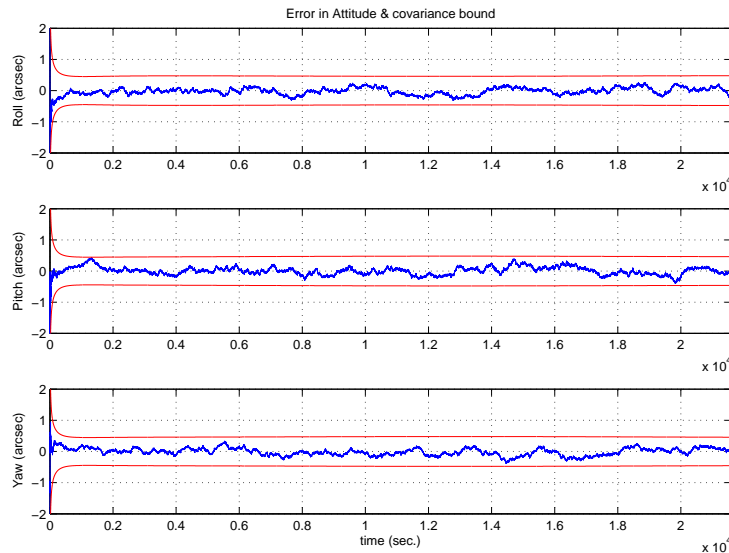


Figure 6: The Attitude Errors Using Kalman Filter

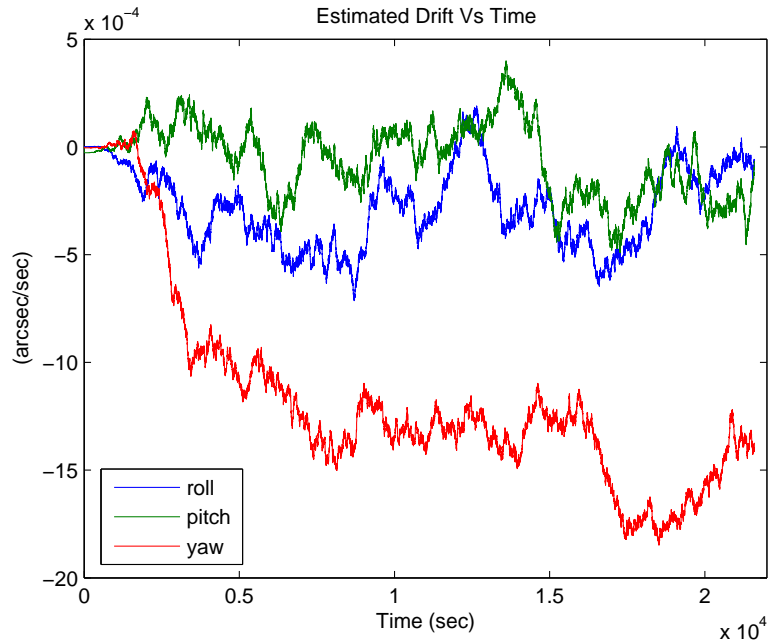


Figure 7: The Gyro Bias Errors

4.1 The Kalman Filter Results

The Kalman filter algorithm is designed and implemented using the above system models and equations. Figure (6) shows the improvement of using the angular rate measurements from the GAIA scientific instruments with the Kalman filter simulation for the attitude errors.

The resulting attitude determination errors (3σ) is less than 5 arcsec (all time). The estimated gyro bias errors is shown in figure (7) and the estimated angular rate errors is also shown in figure (8). The resulting angular rates errors is less than 1 *mas/sec* which is less the GAIA AOCS performance requirements (see table 1).

Finally, figure (9) shows the Relative Attitude Measurement Error (RAME) at any given time t which is defined as the standard deviation of the attitude measurement error over the AF CCD integration time $\tau = 3.3\text{sec}$ around t .

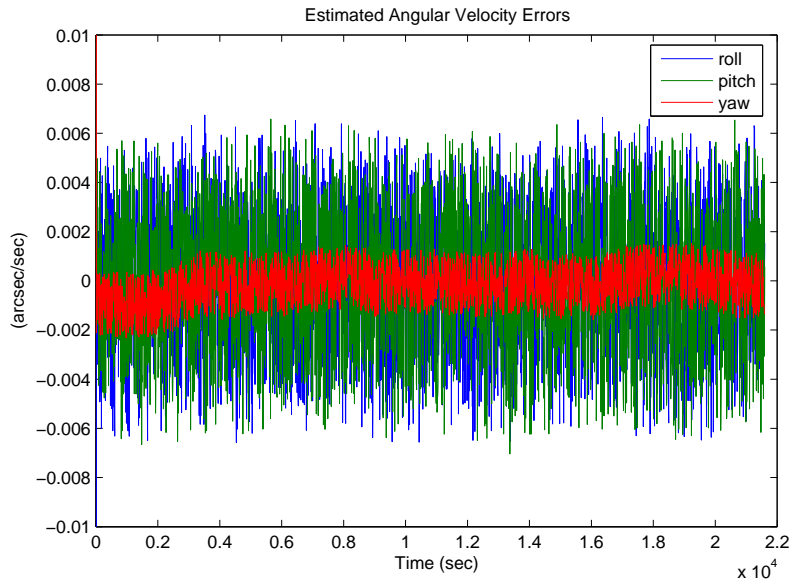


Figure 8: The Angular Velocity Errors (arcsec/sec)

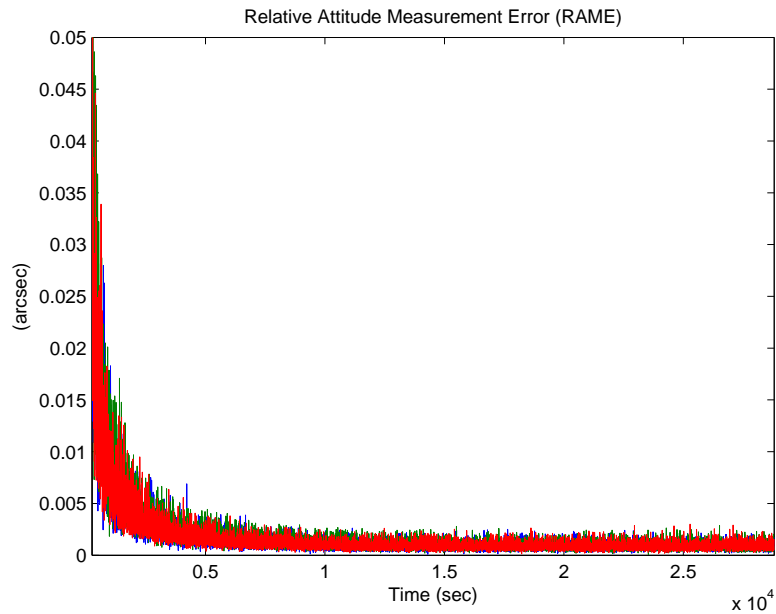


Figure 9: Relative Attitude Measurement Error (RAME)

4.2 Smoothing Estimation Technique

The *smoothing* estimation technique refers to a non-real-time or *off-line* algorithm that makes optimal use of all measurements collected during a given interval of time. The smoothing estimator provides thus the optimal state estimate based on all information delivered by the measurements sampled during the interval $0 < t < T$. The smoothed state estimate at time t is denoted by $\hat{x}(t|T)$. Typical smoothing algorithms are based on the combination of two individual optimal filters, namely the *forward* and the *backward* filters. The first filter uses all data before the time t which leads to the familiar classical Kalman filter estimate $\hat{x}(t)$, whereas the backward filter operates on all data produced after the time t and is expressed as $\hat{x}_b(t)$. Together these two filters utilize all the available information as shown in figure (10).

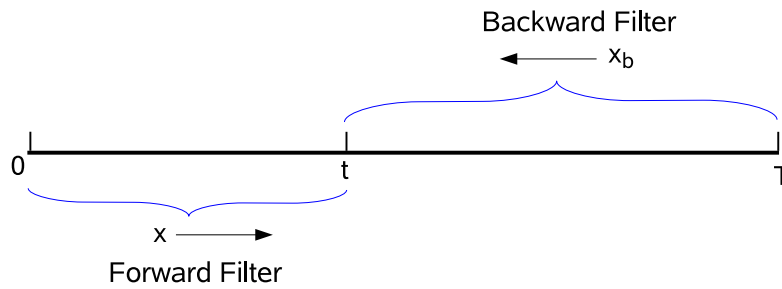


Figure 10: Relationship of Forward and Backward Filters

All smoothing algorithms depend, in some way, on the forward filtering solution. Therefore, accurate forward filtering is prerequisite to accurate smoothing. The smoothed states $\hat{x}(t|T)$ will be in the form

$$\hat{x}(t|T) = P(t|T) [P^{-1}(t)\hat{x}(t) + P_b^{-1}(t)\hat{x}_b(t)]$$

where \hat{x}_b is the backward estimated states and $P_b(t)$ is the backward error covariance matrix. The smoother error covariance matrix $P(t|T)$ is given by

$$P(t|T) = [P^{-1}(t) + P_b^{-1}(t)]^{-1}$$

Several forms for the smoothing equations may be used for smoothing the states. One of these forms is the one by 'Rauch-Tung-Striebel' [9] which does not involve backward filtering but directly deriving the smoothed state estimate by

$$\hat{x}(t_k|T) = \hat{x}(t_k) + K_k (\hat{x}(t_{k+1}|T) - \Phi_k \hat{x}(t_k) - b_k)$$

$$\hat{x}(T|T) = \hat{x}(T)$$

$$K_k = P_k \Phi_k^T (\Phi_k P_k \Phi_k^T + Q_k)^{-1}$$

$$P(t_k|T) = P(t_k) + K_k (P(t_{k+1}|T) - \Phi_k P_k \Phi_k^T - Q_k) K_k^T$$

Where $P(t_k)$ and $\hat{x}(t_k)$ are obtained from a forward filter and $P(t_k|T)$ is (approximately) the error covariance matrix associated with $\hat{x}(t_k|T)$. The matrix Φ_k is the state transition matrix at time t_k . The quantity b_k is the solution to the differential equation

$$\dot{b}(t) = F(\hat{x}(t), t)b(t) + f(\hat{x}(t), t) - F(\hat{x}(t), t)\hat{x}(t)$$

$$\dot{b}(T) = 0$$

The forward Kalman filter outputs for the covariance matrices and the expected states at each time step are fed to the smoothing algorithm to produce the backward and the smoothed states and also the backward and the smoothed covariance matrices. Figure (11) shows the output of the smoothing algorithm for the smoothed attitude errors with the covariance bounds. Furthermore, figure (12) demonstrate the difference between the smoothed roll attitude and the forward Kalman filter roll attitude.

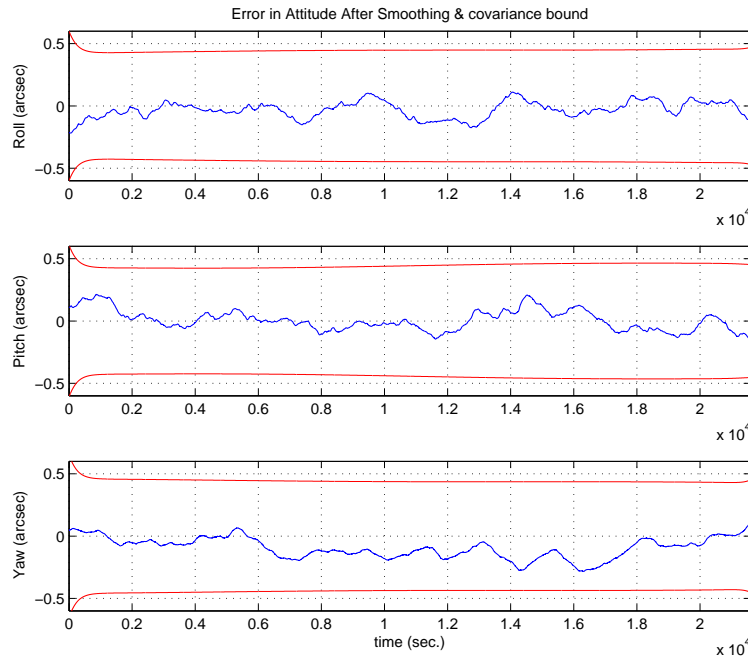


Figure 11: The Smoothed Attitude Errors

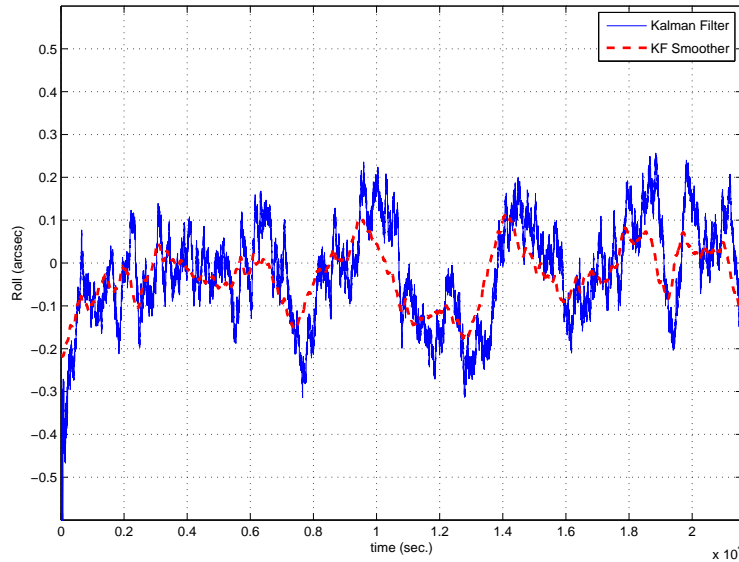


Figure 12: The Difference between the Smoothed and the Forward Filter Roll Attitude

5 The Batch Attitude Estimation


The Batch Least Squares (BLS) estimation is the process that can be used to estimate the state vector at an epoch time after accumulating batch measurement data for a given interval. The BLS solution selects a state estimate x that minimizes the sum of the squares of the calculated residuals. Although the BLS is computationally intensive for large number of measurements but leads to a good estimation of the state vector [10]. The big advantage of the BLS is that it addresses the problem by using a big batch of data. By finding the best fit through the data, the process is much more likely to converge to a solution even if some of the data does not conform to the dynamic equations which we are trying to fit to.

The normal equation for batch estimation are

$$\delta \hat{x}_k = (H_k^T R_k^{-1} H_k + \bar{P}_k^{-1})^{-1} (H_k^T R_k^{-1} z_k + \bar{P}_k^{-1} \delta \bar{x}_k)$$

$$P_k = (H_k^T R_k^{-1} H_k + \bar{P}_k^{-1})^{-1}$$

Where $\delta \hat{x}_k$ and P_k are the estimated state and covariance matrix at epoch time t_k while $\delta \bar{x}_k$ and \bar{P}_k are the a priori state and covariance matrix at the same epoch time. Without any previous estimates, \bar{P}_k is specified with arbitrarily large values. The terms

	<p style="text-align: center;">GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation</p>	<p>Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 17 of 27</p>
---	---	--

containing the measurement covariance matrix, R_k in the above equations are actually the accumulation of the star tracker measurement and the measured angular velocities, from the payload scientific instrument, over *pre-selected* duration for batch. These terms are computed as follows

$$H_k^T R_k^{-1} H_k = \sum_{i=1}^n (H_i \Phi(t_i, t_k))^T R_i^{-1} H_i \Phi(t_i, t_k)$$

$$H_k^T R_k^{-1} \vec{z}_k = \sum_{i=1}^n (H_i \Phi(t_i, t_k))^T R_i^{-1} \vec{z}_i$$

Where H_i and R_i are the sensitivity and measurement noise covariance matrices at time t_i respectively. \vec{z}_i is the measurement residual vector vector at time t_i . Also, $\Phi(t_i, t_k)$ is the state transition matrix from time t_i to time t_k .

5.1 The Maximum Likelihood Parameters Estimation

We can use another technique to estimate the unknown parameters of the GAIA OGA for each batch of measurements, which include the initial conditions of the attitude quaternions and the initial condition of the angular rates as well as the rate bias for that batch of measurements. This technique is the maximum likelihood function for parameters estimation . The maximum likelihood estimates are obtained by minimization of the following cost function

$$J(\Theta) = \frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)]$$

where $z(t_k)$ is the measurement vector (quaternions and angular rates) at time t_k , and $y(t_k)$ is the estimated vector (quaternions and angular rates) at time t_k . The unknown parameters vector Θ includes the initial conditions for quaternions and angular rates for each batch interval and also the constant value for the rate bias for that batch. The diagonal elements of the measurement noise covariance matrix R representing the variances of the quaternions errors and the angular rate errors. Thus the cost function $J(\Theta)$ in the above equation is nothing but the weighted sum of squares of the response error and is quadratic in nature. Any optimization methods, direct search or gradient based, can be applied to obtain estimates of the best values for the vector of the unknown parameters Θ .

As we can see from the above cost function $J(\Theta)$ is that it is dependent on the measurement noise covariance matrix R . In general, the covariance matrix of the measurement errors is unknown and, hence, has to be estimated. If we treat the R as another unknown which makes the optimization of the cost function more involved. Then the brute-force method would be to include the elements of the covariance matrix in the unknown parameters vector Θ , and then apply one of the optimization methods. In order to estimate

the covariance matrix we can drive the cost function for the maximum likelihood estimate of R by partial differentiation of the cost function $J(\Theta)$ with respect to R and setting to zeros, this leads to the following equation

$$R = \frac{1}{N} \sum_{k=1}^N [z(t_k) - y(t_k)] [z(t_k) - y(t_k)]^T$$

Once we obtain a maximum likelihood estimate of R , then the cost function $J(\Theta)$ could be reduced to [11]

$$J(\Theta) = \det(R)$$

In other words, the determination of the parameters vector Θ will be the one which minimizes $\det(R)$.

5.2 The Batch Least Squares Results

By using the simulated GAIA gyro data, the batch least estimator with the fixed 20 minutes batch interval can determine the GAIA attitude as shown in figure (13). The duration of the time span or the batch interval is usually decided based on the required accuracy and on how much data is available. Figure (14) demonstrates how the batch interval can affect the determined attitude accuracy by showing the pitch errors for different batch intervals. With longer batch intervals, the accuracy of the final determined attitude is more accurate than for shorter batch estimates because the collected amount of data tend to reduce the estimated cost function of the errors between the measured and estimated states. Moreover, By using shorter interval, the batch estimation method will be faster than using the longer batch intervals but with higher attitude errors between the intervals as shown in figure (14) .

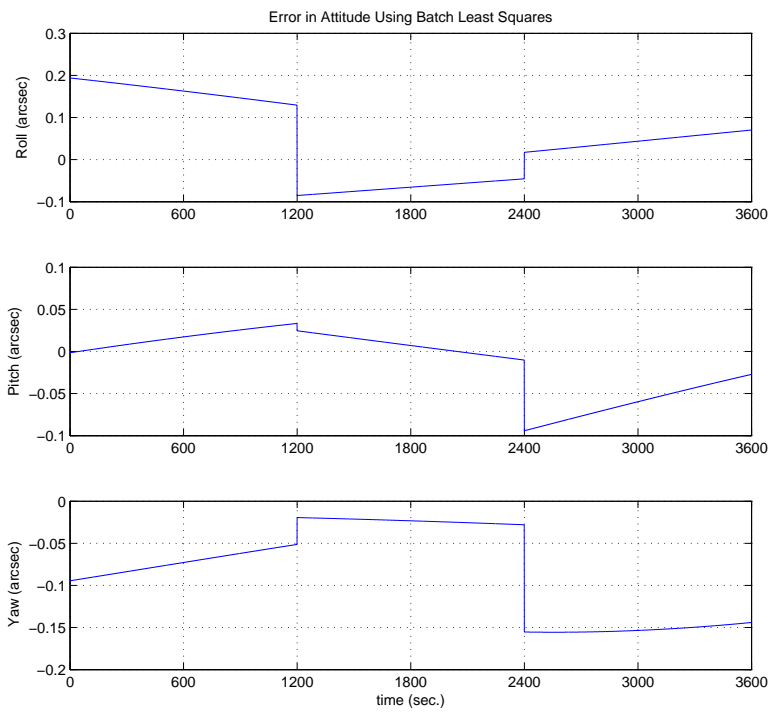


Figure 13: GAIA Attitude Errors Using the Batch Least Estimation

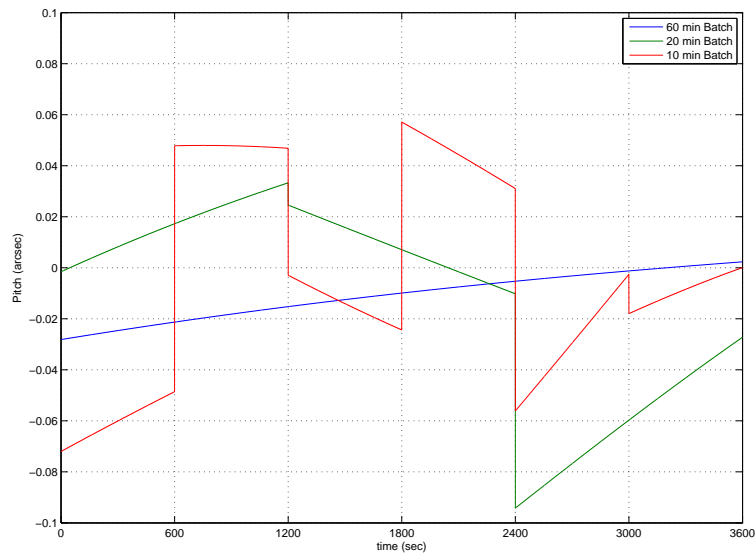


Figure 14: The Effect of Different Batch Interval on Attitude Errors

Finally, to compare and choose the best On-Ground attitude estimation algorithm, the results of the BLS, the Kalman filter and the KF smoother are demonstrated in figures (16),(15) and (17). As a conclusion from these results, the *Kalman Filter Smoother* produces the improved attitude determination over the Extended Kalman Filter. The Kalman filter has the advantage that it constantly update the present attitude using the latest measurements in conjunction with the collected estimation of the previous measurements. Moreover, the Smoother algorithm is giving more accurate attitude results because the final state estimate is used as an initialization for the backward filter which results a more accurate state estimates than the EKF. On the other hand, the Batch Least Squares estimates gives the most accurate OGA for long batch interval (60 min). But for short interval the BLS almost produce similar attitude accuracy to that produced by the smoother KF. In terms of processing time, for a 60 min simulated data, the required processing time for the EKF algorithm is 33.87 sec, for the KF smoother is about 5.27 sec more than the EKF time, and about 35 min for the BLS.

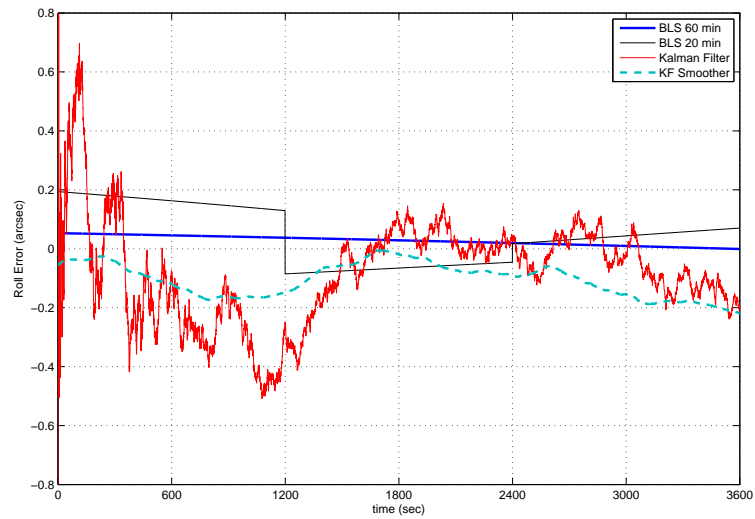


Figure 15: The Roll Attitude Errors Using Different Estimation Techniques

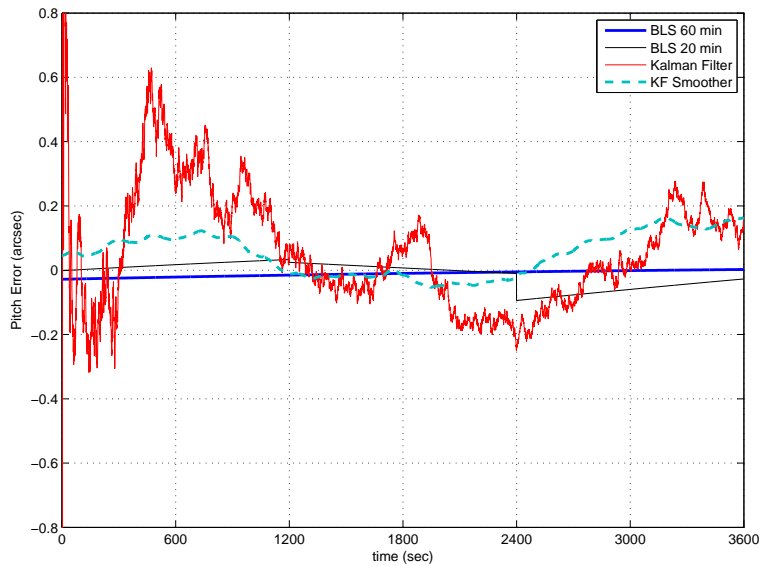


Figure 16: The Pitch Attitude Errors Using Different Estimation Techniques

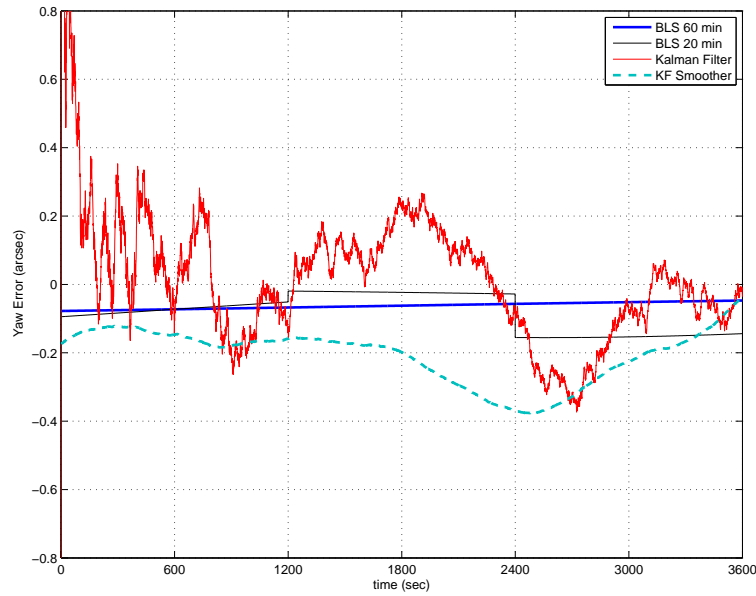


Figure 17: The Yaw Attitude Errors Using Different Estimation Techniques

6 Gaia Disturbance Torques

Through understanding of the dynamical behaviour of the spacecraft, the accuracy of the reconstructed attitude and torques can be improved, and is therefore of great relevance for obtaining the highest accuracies and best overall statistical properties for the main scientific products of the mission: the astrometric parameters of the target stars. The attitude dynamics of the Gaia describes the changes in the angular rates around the principle axes of the inertia tensor. These changes are the results of the internal and the external torques and are described for a rigid body by the Euler equations. Two types of torques can be distinguished: internal torques (due to thruster firing) and external torques (due to solar radiation or external particles hits). In this work analysis, the underlying external torques are assumed to be a continuous function up to its second derivative in time.

The quaternion kinematics equations of motion are derived by using the S/C angular velocity (ω), given by

$$\frac{d}{dt}\mathbf{q} = \frac{1}{2}\Omega(\omega)\mathbf{q}$$

where $\Omega(\omega)$ and are defined by

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}$$

The Euler dynamic equations are given by

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \frac{1}{2}\Omega(\omega)\mathbf{q}(t) \\ \dot{\omega}(t) &= I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \end{aligned}$$

Where I_{sc} is the moment of inertia for Gaia and T_e is the total disturbance and control torques acting on the spacecraft.

We will study the cases where T_e exists in the attitude simulation in two cases:

- Case 1; Body Fixed Disturbance Torques ($T_B = const.$).

$$T_e = T_B = I_{sc}\dot{\omega}[n_1 \ n_2 \ n_3]^T$$

where $\dot{\omega}$ is assumed to be equal to the total allowed angular rate errors for one great circle (6 hours) and n_i are random integers from 2 to 10. Figures 18(a) and 18(b) show the attitude residuals with and without predicting the body fixed disturbance torques. The attitude errors using Batch Least Squares (BLS) before and after external torque prediction is shown in figure 19.

- Case 2; Inertial Fixed Disturbance Torques ($T_I = const.$).

$$T_e = A(t) * T_I = A(t) * I_{sc}\dot{\omega}[n_1 \ n_2 \ n_3]^T$$

where $A(t)$ is the attitude matrix of GAIA at time t . Figures 20(a) and 20(b) show the attitude residuals with and without predicting the inertial fixed disturbance torques. The attitude errors using BLS before and after external torque prediction is shown in figure 21.

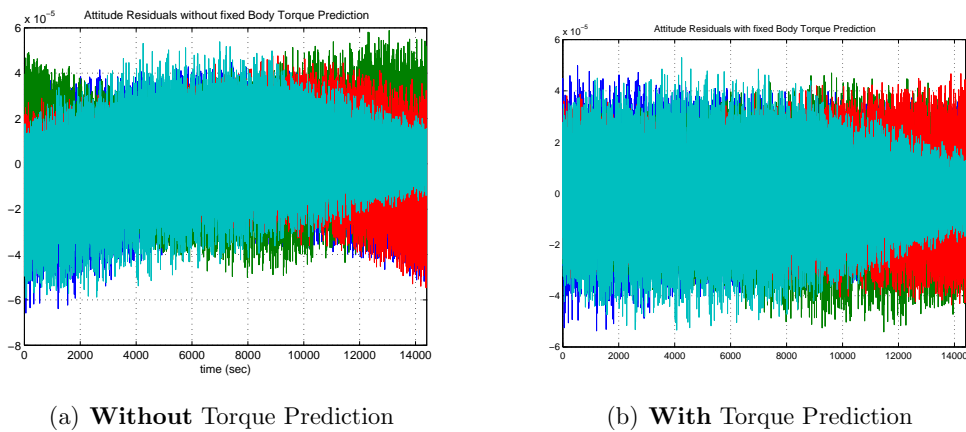


Figure 18: Attitude Residuals for Body Fixed Disturbance Torque.

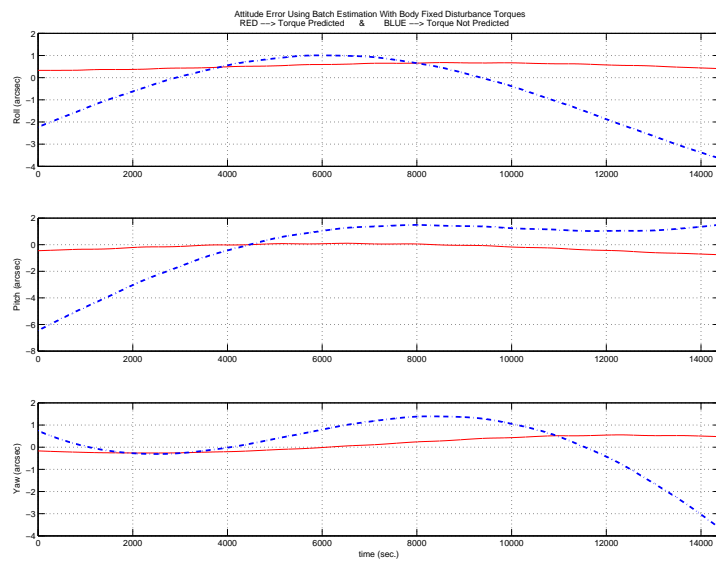


Figure 19: Attitude Errors **with** and **without** *Body Fixed* Torque Prediction

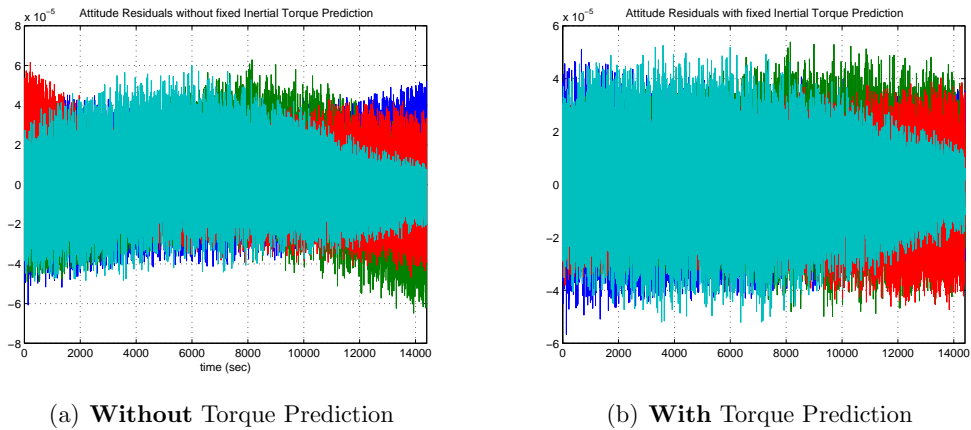


Figure 20: Attitude Residuals for Inertial Fixed Disturbance Torque.

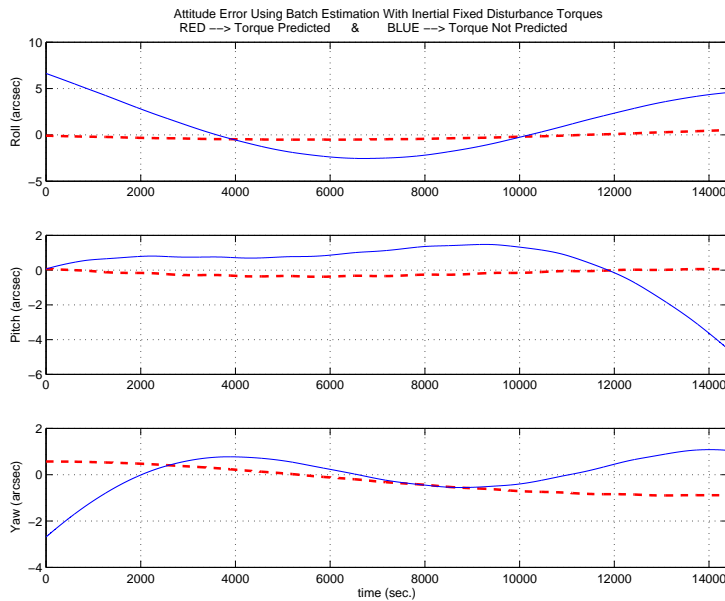




Figure 21: Attitude Errors **with and without** *Inertia Fixed* Torque Prediction

7 Conclusion and Future Work

The GAIA on-board attitude determination algorithm is designed and discussed in this document. The Kalman filter is used to produce an accurate on-board attitude by

	<p style="text-align: center;">GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation</p>	<p>Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 26 of 27</p>
---	---	--

utilizing the star tracker measurements for rough spacecraft quaternions and the payload scientific telescopes measurements for angular rates. The obtained results, using the prescribed algorithm in this document, meet the attitude determination requirements which given in the GAIA Mission Requirements Document. The attitude determination errors and the rate measurement errors along scan and across scan and also the relative pointing errors (given in table 1) have been validated using the proposed EKF algorithm. Also, in this document, we design and establish Smoothing type estimation algorithms for the Initial On Ground Attitude Determination (IOGAD) to make the attitude errors much less than the On-board attitude estimation. The Batch Least Square estimation is also designed and established in this document to demonstrate the differences between variety of estimations techniques. The BLS with long batch time has been showed to be the best On-Ground ADS suitable for the GAIA mission over the EKF and the KF smoother. In the future work, the star measurements of the scientific payload CCDs will be included to improve the resulting attitude accuracy. The expected improvement factor on the IOGAD will be somewhere between 2 and at most 5 (after some enhanced attitude modeling).

	<p>GAIA On-Ground Attitude and External Torque Determination Using Kalman Smoother and Batch Estimation</p>	<p>Doc.No.: FLK-ALG-TN-ZAR-002 Issue: 1.0 Page: 27 of 27</p>
---	---	--

References

- [1] ESA, “GAIA Mission Requirements Document”, GAIA-EST-RD-00553, 1 July 2005.
- [2] ESA, “GAIA System Requirements Document”, GAIA-SRD-001, 30 March 2004.
- [3] Astrium, “GAIA System Level Technical Reassessment Study, Final Report”, EF5/FR/PC/038.02, 17/06/2002, Chapter 3.5.
- [4] L. Lindegren, “Attitude Parametrization for GAIA”, SAG-LL-30, 1 July 2000.
- [5] Astrium, “Autonomous Star Tracker Requirements Specification”, Issue 1, 2006.
- [6] Frederic Safa, “A Summary of the Gaia Spacecraft Design”, 15 March 2006.
- [7] Wertz, J. R., “Spacecraft Attitude Determination and Control”, Kluwer Academic Publishers, the Netherlands, 1990.
- [8] Gelb, A. , “ Applied Optimal Estimation”, MIT Press, Cambridge, MA, 1974.
- [9] Rauch, H.E., Tung, F., and Striebel, C.T. , “ Maximum Likelihood Estimation of Dynamic Systems”, AIAA Journal, Vol.3, No. 8, August 1965.
- [10] Crassidis, J.L. , Junkins, J.L., “Optimal Estimation of Dynamic Systems”, Chapman and Hall/CRC, 2004.
- [11] Ravindra V. Jategaonkar, “Flight Vehicle System Identification”, American Institute fo Aeronatutics and Astronautics, Vol. 216, 2006.



Analysis and implementation for the IDT process “OGA1 determination”

prepared by: Malak A. Samaan
approved by: Stephan Theil
reference: GAIA-C3-TN-ZARM-MS-001-01
issue: 1
revision: 1
date: October 19, 2007
status: Issued

1 Introduction

The work presented in this document concerns the accurate On-Ground Attitude (OGA) reconstruction for the astrometry spacecraft Gaia in the presence of disturbance and control torques acting on the spacecraft. The reconstruction of the expected environmental torques which influence the spacecraft dynamics will be also investigated.

The telemetry data from the spacecraft will include the on-board real time attitude which is of order of several arcsec. This *raw* attitude is the starting point for the further attitude reconstruction. The OGA will use the inputs from the field coordinates of known stars (attitude stars) and also the field coordinate differences of objects on the Sky Mapper (SM) and Astrometric Field (AF) payload instruments to improve this raw attitude. The on-board attitude determination uses Kalman Filter (KF) to minimize the attitude errors and produce more accurate attitude estimation than the pure star tracker measurement. Therefore the first approach for the OGA will be an adapted KF. Furthermore, we will design a smoothing type estimation algorithm and batch least squares algorithms for the OGA to get more accurate OGA estimation. Finally, a comparison between these different attitude determination techniques in terms of accuracy, robustness, speed and memory required will be evaluated in order to choose the best attitude algorithm for the OGA. The expected resulting accuracy for the OGA determination will be on the order of milli-arcsec [1].

2 Background and Objectives

The principal feature of the Gaia astrometry mission is to accurately measure the positions, distances, space motions, and many physical characteristics of some one billion stars in our Galaxy and beyond on orbit at the vicinity of the second Lagrange point L2. In order to achieve the targeted measurement accuracies of the Gaia imaged stars, the real time on-board attitude should be improved using the on-ground attitude reconstruction [2].

The main objective of the work presented here is to reconstruct the *non-real-time* On-Ground Attitude (OGA) with very high accuracy for further processing. The accuracy requirements for the attitude reconstruction for the Gaia on-ground data analysis would enforce studying different attitude estimation techniques and choosing the best one in terms of accuracy, robustness and speed. The current goal for the OGA1 accuracy is about 50 milli-arcsec for the along and across scan attitude accuracy. These techniques shall be part of the Initial Data Treatment (IDT) chain.

The Gaia scientific payload is used as a high accuracy stellar gyro for achieving the attitude determination and enabling Time Delay Integration (TDI) mode. The star speed is calculated along and across scan from Sky Mapper (SM) and Astrometric Field (AF) measurements. When considering the two telescopes, one obtains four angular speed measurements, from which the

satellite rigid body angular speed vector can be calculated. It should be mentioned that SM-1 is seen by telescope No. 1 and SM-2 by telescope No. 2. The Sky Mapper measurements consist of the determination of the centroid crossing times of the stars over each of the SM CCD arrays [3].

The Gaia Initial Data Treatment (IDT) will process the newly arrived telemetry data and also the various pieces of auxiliary data from the Gaia spacecraft. The IDT will have many processes and subtasks, one of them is the attitude reconstruction or the OGA determination.

In order to match the new observations reliably with the existing source list, it is essential to have an accurate attitude. It is therefore suggested to carry out an early attitude updating during the IDT. This will require a cross matching with a dedicated Attitude Star Catalog, followed by an attitude updating.

At this stage we are only aiming for attitude accuracy of a tens of milliarcsec level, so some simplifications may be allowed.

The IDT will have to run in several concurrent processes, and the attitude improvement is done separately for each process. The same time interval will therefore be processed several times, and we need only to update a given attitude interval if it has not already been updated in a previous process.

Through understanding of the dynamical behavior of the spacecraft, the accuracy of the reconstructed attitude and torques can be improved, and is therefore of great relevance for obtaining the highest accuracies and best overall statistical properties for the main scientific products of the mission; the Astrometric parameters of the target stars.

Given all these mission data and parameters, the attitude data simulation, in the form of the satellite quaternion and angular velocities, could be simulated for any time span. In order to simulate the algorithms for OGA determination, we have to simulate attitude observations with the following requirements:

- The simulated data shall be provided for the length of at least one great circle (6 hours).
- The data shall contain the true (ideal) attitude quaternion as well as the true rate in the body fixed frame. Also, the raw (measured) attitude quaternion [ideal quaternion + noise].
- The simulated observations of attitude stars shall contain:
 - The time of observation.
 - The indicator for the FOV.

- The position on the sky of the observed star (the right ascension α , and declination δ).
- The field coordinates of the star (η and ζ) for SM and AF1-9 and G-magnitude.
- The Transit ID and the Attitude Star Catalog.
- Cross-Match Table.
- The attitude quaternion at each observation time will be calculated using the B-Spline as described in the next section.

3 Gaia Attitude Presentation

The Gaia spacecraft attitude is considered to be represented by the quaternion vector defined as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{13} \\ q_4 \end{bmatrix} \quad (1)$$

with

$$\mathbf{q}_{13} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\mathbf{n}} \sin\left(\frac{\theta}{2}\right) \text{ and } q_4 = \cos\left(\frac{\theta}{2}\right) \quad (2)$$

where $\hat{\mathbf{n}}$ is a unit vector corresponding to the axis of rotation, and θ is the rotation angle.

The elements of the quaternion satisfy a unit norm constraint

$$\mathbf{q}^T \mathbf{q} = \mathbf{q}_{13}^T \mathbf{q}_{13} + q_4^2 = 1 \quad (3)$$

The frequency of the on-board attitude (1 Hz) is different from the frequency of the star centroiding and crossing observation times in the SM and AF which measured in an uneven sequences and in order of nanosec. In order to start the OGA determination, the attitude quaternion at every time of observation should be evaluated by interpolating the on-board attitude.

Both the on-board real-time attitude and the on-ground reconstructed attitude, and perhaps also the nominal scanning law attitude [4], can be represented by means of cubic splines followed by normalization (to validate the unit length condition). So, according to [5] at any time interval $[t_{beg}, t_{end}]$ there is a vector $\mathbf{s}(t)$ consists of four cubic spline functions $s_m(t)$, $m = 1, \dots, 4$ from this vector the attitude quaternions at time t is calculated from

$$\mathbf{q}(t) = \mathbf{s}(t) / \text{norm}(\mathbf{s}(t)) \quad (4)$$

Also we can derive the instantaneous scan rate (ω) in the fields of view by computing the time derivative of the quaternion at time t from

$$\dot{\mathbf{q}}(t) = [\dot{\mathbf{s}} - \mathbf{q}\mathbf{q}^T\dot{\mathbf{s}}] / \text{norm}(\mathbf{s}(t)) \quad (5)$$

In this case the scan rate at any given time is derived from

$$\omega(t) = \frac{2}{\text{norm}(\mathbf{q})} \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \dot{\mathbf{q}}(t) \quad (6)$$

Each of the functions $s_m(t)$ is a cubic spline defined on $[t_{beg}, t_{end}]$, it is therefore continuous in this interval and its first two derivatives $\dot{s}_m(t)$ and $\ddot{s}_m(t)$ are also continuous. The third derivatives is discontinuous at discrete points, called the knots, and constant between the knots. The frequency of knots determines the flexibility of the spline, or in other words, more knots gives more flexible spline. The used knot sequence is used for all the four components of $\mathbf{s}(t)$, with a knot interval of 15 sec.

A cubic spline can be represented in a number of different ways, e.g. by means of its value and second derivative at each knot. However, the method proposed here is to write the spline as a linear combination of non-negative local basis functions, called B-splines (See [6]). Let $B_n(t)$, $n = 1, \dots, N$, be the N B-splines defined on the attitude interval $[t_{beg}, t_{end}]$. Because the B-splines are linearly independent basis functions, N is also the number of degrees of freedom of the spline. Then

$$s_m(t) = \sum_{n=1}^N S_{mn} B_n(t), \quad m = 1, \dots, 4 \quad (7)$$

and

$$\dot{s}_m(t) = \sum_{n=1}^N S_{mn} \dot{B}_n(t), \quad m = 1, \dots, 4 \quad (8)$$

The B-splines B_n (and their derivatives $\dot{B}_n(t)$) are uniquely defined by the knot sequence

$$\tau = \{\tau_{-1}, \tau_0, \tau_1, \tau_2 \equiv t_{beg}, \tau_3, \dots, \tau_{N-2}, \tau_{N-1} \equiv t_{end}, \tau_N, \tau_{N+1}, \tau_{N+2}\} \quad (9)$$

The knot sequence can be evaluated for any time t according to a simple and stable recurrence relation. Moreover, it should be non-decreasing ($\tau_i \leq \tau_{i+1}$, $i = -1, \dots, N+2$). At any given time $t \in [\tau_n, \tau_{n+1}]$ the only non-zero B-splines are B_{n-1} , B_n , B_{n+1} and B_{n+2} . Therefore, for the whole attitude interval $[t_{beg}, t_{end}]$ the non-zero B-splines are B_1, B_2, \dots, B_N . The additional knots on either side of the attitude interval are needed to define B_1, B_2, B_3 and B_{N-2}, B_{N-1}, B_N , and can be chosen arbitrary as long as τ is non-decreasing [6].

3.1 External Torque Estimation

The first task in the OGA determination is to reconstruct the external torque as a continuous function of time from the telemetry attitude dynamics. In order to achieve this task, we consider the spacecraft as a rigid body then we can express the angular velocity using the following equation;

$$\dot{\omega}(t) = I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \quad (10)$$

where I_{sc} is the moment of inertia of the Gaia S/C and T_e is the total disturbance and control torques acting on the spacecraft.

At each observation time the attitude quaternion is calculated from Eq. 4 and the time derivative of the quaternion from Eq. 5. By using Eq. 6 the angular rates at each observation time are determined.

Now to calculate the estimated external torque, we could rewrite Eq. 10 in the form

$$T_e(t_{obs}) = I_{sc}\dot{\omega} + \omega \times I_{sc}\omega \quad (11)$$

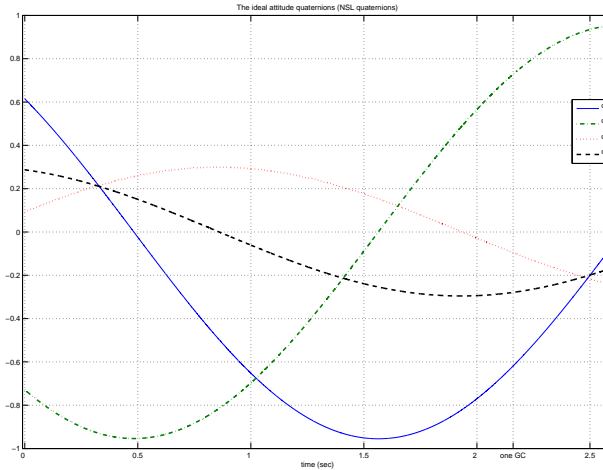


Figure 1: The simulated quaternion using NSL.

The Nominal Scanning Law (NSL) [4] is used to simulate the attitude quaternion for more than one great circle (6 hours) as illustrated in figure (1). Consequently, when Eq. 6 is used to calculate the angular rates at each time step the resulting rates are illustrated in figure (2). On the other hand, figure (3) demonstrates the resulting external torques which are supposed to be

the sum of all the environmental disturbance torques, internal torques and control torques acting on the Gaia spacecraft while following the NSL.

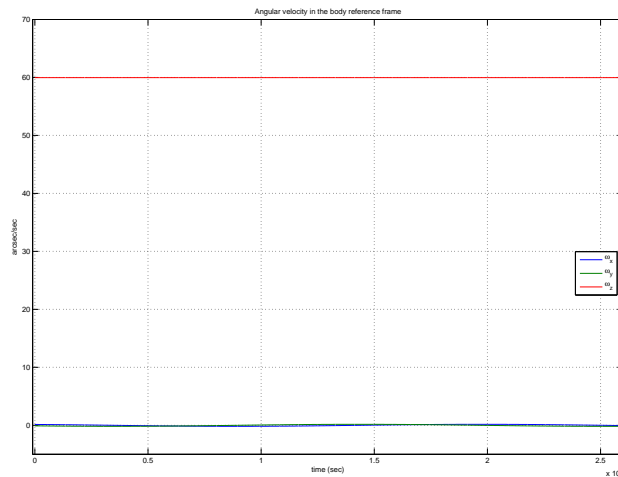


Figure 2: The simulated angular rates using the B-Splines.

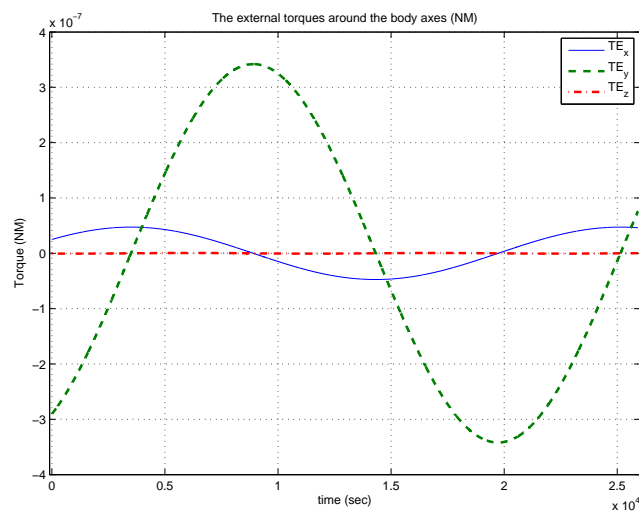


Figure 3: The external (disturbance and control) torques acting on the S/C.

4 The Kalman Filter Model

The Kalman filter state vector for the Gaia OGA1 is consisting of the 4 components of the attitude quaternion and the three components for the angular rates.

$$x = [q_1 \ q_2 \ q_3 \ q_4 \ \omega_x \ \omega_y \ \omega_z]^T \quad (12)$$

The system model equation has the following form

$$\dot{x}(t) = f(x(t), t) + G(x(t), t)w(t) \quad (13)$$

Also, the measurement model at time t_k is given by

$$Y_k = h(x(t_k), t) + \nu(t) \quad (14)$$

where $w(t)$ is the process noise and $\nu(t)$ is the measurement noise, both are discrete Gaussian white-noise processes

$$\begin{aligned} w(t) &\simeq N(0, Q(t)) \\ \nu(t) &\simeq N(0, R(t)) \end{aligned} \quad (15)$$

The Kalman filter dynamic equations of state are given by

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \frac{1}{2}\Omega(\omega)\mathbf{q}(t) \\ \dot{\omega}(t) &= I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \end{aligned} \quad (16)$$

Where

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \quad (17)$$

Also the external torque T_e is calculated at each observation time from Eq. 11.

The Kalman filter propagation equations are defined by

$$\begin{aligned} \dot{x}(t) &= F(t)x(t) + G(t)w(t) \\ \dot{P} &= FP + PF^T + GQG^T \end{aligned} \quad (18)$$

where P is the covariance matrix.

The F matrix is given by

$$F = \begin{bmatrix} 0.5\Omega(\omega) & -0.5\Xi(\mathbf{q}) \\ 0_{3 \times 4} & F_{\dot{\omega}\omega} \end{bmatrix} \quad (19)$$

where

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (20)$$

$$F_{\dot{\omega}\omega} = -I_{sc}^{-1} ([\omega \times] I_{sc} - [I_{sc}\omega \times]) \quad (21)$$

The matrix $[\underline{a} \times]$ is the skew symmetric matrix of the vector \underline{a} .

Also, the Kalman filter update equations are defined by

$$\begin{aligned} \hat{x}_k(+) &= \hat{x}_k(-) + K_k [Y_k - h_k(\hat{x}_k(-))] \\ P_k(+) &= [I - K_k H_k(\hat{x}_k(-))] P_k(-) \\ K_k &= P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1} \end{aligned} \quad (22)$$

Each measurement is essentially an association of a certain time instant t with certain field coordinate angles of the measured star (η_m and ζ_m) for a certain field of view in the instrument frame.

The calculated field angles at each observation time are given by

$$h(x_t) = [\eta_c^i \quad \zeta_c^i] \quad (23)$$

where $i = 1$ or 2 is field of view number. These field angles depend on the basic angle between the two fields of view, which is equal to 106.5° [1].

The measurement sensitivity matrix is given by

$$H_k = \frac{\partial h(x_t)}{\partial x(t)} = \begin{bmatrix} \frac{\partial \eta_k}{\partial \mathbf{q}} & 0_{1 \times 3} \\ \frac{\partial \zeta_k}{\partial \mathbf{q}} & 0_{1 \times 3} \end{bmatrix} \quad (24)$$

4.1 The Kalman Filter Results

The Kalman filter algorithm is designed and implemented using the above system model and equations. The measurement noise matrix R is chosen such that;

$$R = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix} \quad (25)$$

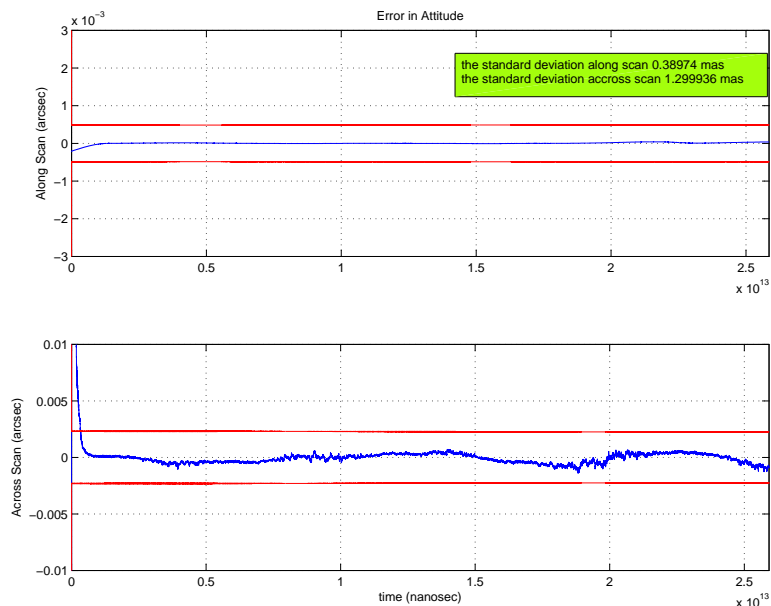


Figure 4: The attitude errors using Kalman Filter.

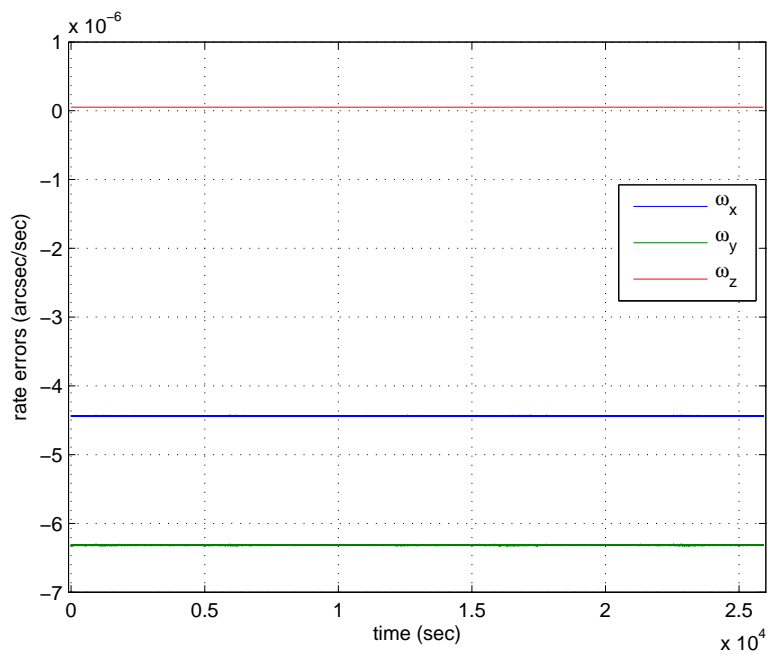


Figure 5: The angular rates errors using Kalman Filter.

The standard deviation for the field angle errors along scan (σ_η) is considered to be 0.1 marcsec and for across scan (σ_ζ) is considered to be 0.5 marcsec. A total of 10^5 of the field angles measurements were simulated for about 7 hours, corresponding to about 4 measurements per sec.

Figure 4 illustrates the attitude estimation errors for the along scan and across scan directions. The along scan direction is around the rotation axis or z-axis while the across scan direction is considered to be the average of the x and y axes or the axes perpendicular to the spin axis. The estimated external torques (which illustrated in figure (3)) are used to obtain the propagated states (\mathbf{q} and $\boldsymbol{\omega}$) at each observation time. In the states update step the Gaia scientific instrument measurements (the field angles) are combined with the Kalman filter simulation.

Using different initial values, the resulting attitude determination errors (3σ) always were less than 2 marcsec along scan and 20 marcsec across scan. The estimated angular rate errors are also shown in figure 5. The resulting angular rates errors are less than 0.007 marcsec/sec around the three axes. The process noise matrix Q used for the Kalman filter propagation equations (18) are considered to be

$$Q = \text{diag} \left([(10^{-12})^2, (10^{-12})^2, (10^{-12})^2, (10^{-12})^2, (5 \times 10^{-10})^2, (5 \times 10^{-10})^2, (10^{-11})^2] \right)$$

The Kalman Filter update equations are based on the updated value of the covariance matrix $P(t)$ which is calculated from Eq. 18. On the other hand, the $P(t)$ is a function of the process noise matrix Q , so the KF tuning for the values of the diagonal element of the Q matrix is very important in determining the resulting accuracy of the estimation. Figures 6 and 7 demonstrate the effect of changing the process noise matrix to be much bigger than the measurement noise matrix R or in other word, the Kalman Filter will rely more on measurements than on the dynamic model. The process noise matrix Q in this case is chosen to be

$$Q = \text{diag} \left([(10^{-6})^2, (10^{-6})^2, (10^{-6})^2, (10^{-6})^2, (5 \times 10^{-5})^2, (5 \times 10^{-5})^2, (2 \times 10^{-6})^2] \right)$$

In figure 6 the overall along scan and across quaternion errors is better than that illustrated in figure 4 but it is more noisy. This is the more realistic case for Gaia. Moreover, the angular rates estimation illustrated in figure 7 is better than the angular rates errors shown in figure 5. All the estimated attitude and rates shown in the previous figures (4, 5, 6, and 7) are based on ideal (error-less) star catalog.

At the beginning of the IDT process, the OGA1 will be using an initial coarse star catalog that will have errors of about 50 marcsec until the more accurate star catalog is created. At this stage we suggest to have the OGA1 dependent more on the system dynamic model or in other words to choose the process noise matrix such that it should be less than the measurement noise matrix. After the Astrometric Global Iterative Solution (AGIS) has become available (after 10 months or more) we can easily change the process noise matrix to let the filter rely more on the measurement which will be more accurate than the system model.

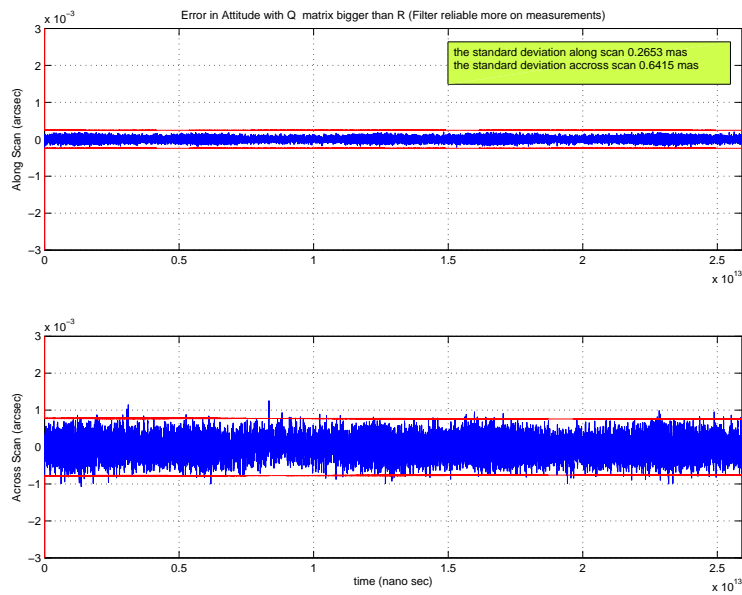


Figure 6: The attitude errors using different tuning.

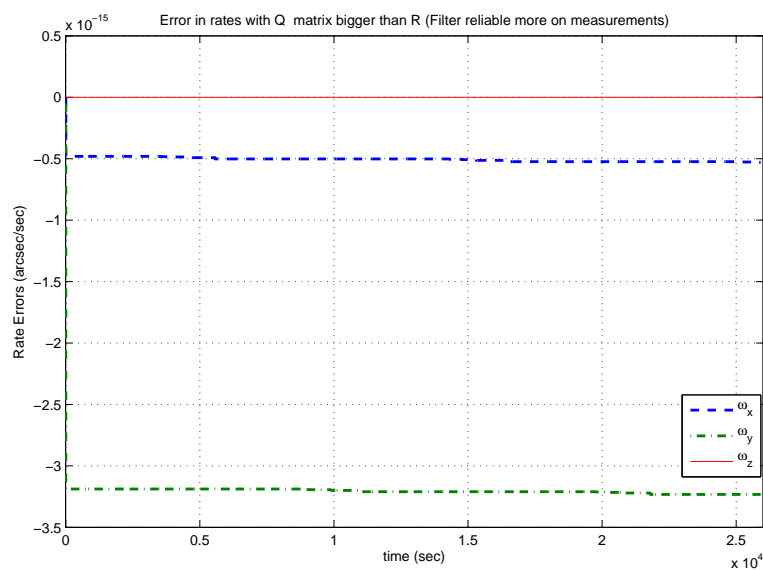


Figure 7: The angular rates errors using different tuning.

Figure 8 illustrates the effects of using a star catalog that has errors of about 50 mas in both the along and across scan directions on the estimation of the attitude errors when the system relies more on the system dynamic model. On the other hand, Figure 9 shows the resulting attitude errors when the system relies more on the measurements.

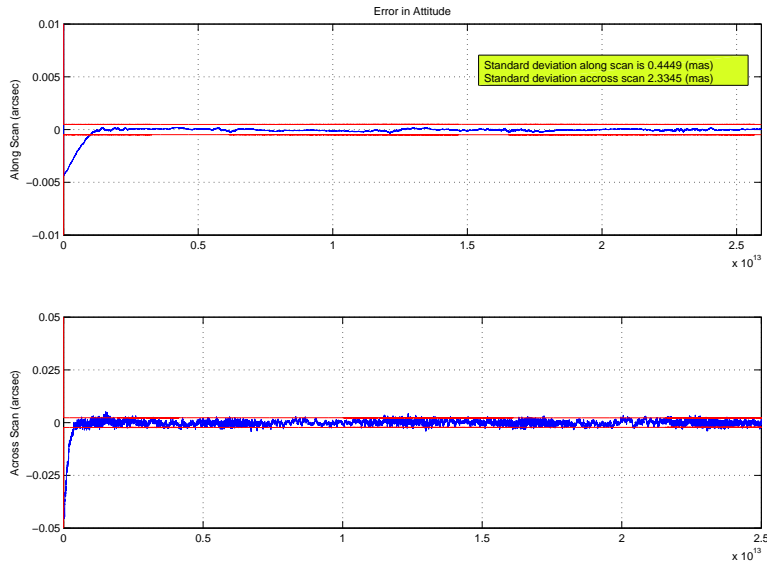


Figure 8: The attitude errors using catalog with 50 marcsec errors.

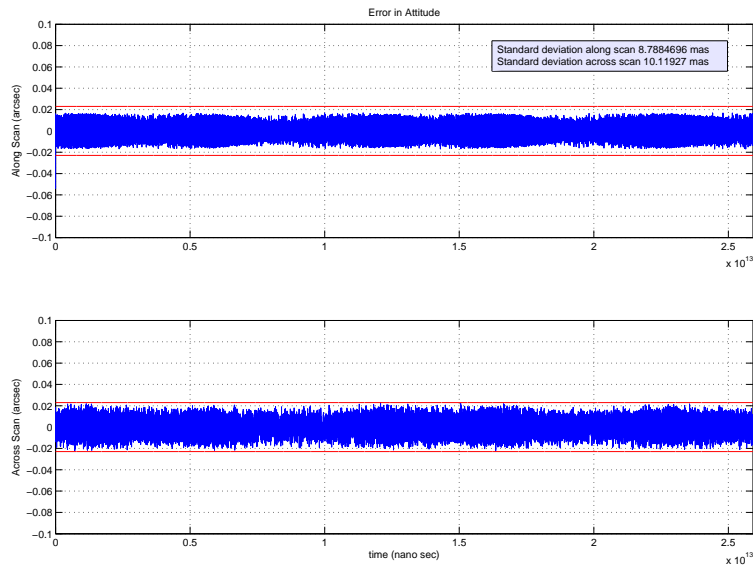


Figure 9: The attitude errors using catalog with 50 marcsec errors and different KF tuning.

5 The Batch Least Squares

The Batch Least Squares (BLS) estimation is the process that can be used to estimate the state vector at an epoch time after accumulating batch measurement data for a given interval. The BLS solution selects a state estimate x that minimizes the sum of the squares of the calculated residuals. Although the BLS is computationally intensive for a large number of measurements, it leads to a good estimation of the state vector [11]. The BLS estimation could be done using one of two approaches; the first approach is the Maximum Likelihood estimates and the other is the Least Squares Trajectory estimates.

5.1 Maximum Likelihood Estimates

The basic idea of the maximum likelihood function for parameters estimation is to obtain the best attitude quaternion and angular rates for a given interval of time. In order to have the most accurate estimate of the unknown parameters of the Gaia OGA for each batch of measurements, which include the initial conditions of the attitude quaternions and the initial condition of the angular rates, we have to minimize the following cost function;

$$J(\Theta) = \frac{1}{2} \sum_{k=1}^N (\eta_{1,k}^m - \eta_{1,k}^e)^2 + (\eta_{2,k}^m - \eta_{2,k}^e)^2 + (\zeta_{1,k}^m - \zeta_{1,k}^e)^2 + (\zeta_{2,k}^m - \zeta_{2,k}^e)^2 \quad (26)$$

Where Θ is the unknown parameters vector which includes the initial conditions for quaternions and angular rates for each batch interval and also the constant value for the rate bias for that batch. $\eta_{i,k}^m$ and $\zeta_{i,k}^m$; are the measured field coordinate angles for field of view number i ($i = 1$ or 2) at the observation time t_k , and

$\eta_{i,k}^e$ and $\zeta_{i,k}^e$; are the estimated field coordinate angles for field of view number i at the observation time t_k , which are calculated as follows;

1. For any set of Θ we can use the S/C dynamics equations (Eq. 16) to obtain the quaternion and angular rates for the given interval time (i.e. 1 hour). Then the associated B-Splines is derived for these estimated quaternion by using Eq. 7.
2. At any observation time (t_{obs}) the measured quaternion vector (\mathbf{q}_m) is calculated using Eq. 4.
3. The position on the sky of the observed star (α_i, δ_i) for the associated FOV is used to calculate the proper direction of the observed star (\mathbf{u}),

4. The estimated field coordinate angles are then computed from

$$\begin{bmatrix} \cos \zeta \cos \eta \\ \cos \zeta \sin \eta \\ \sin \zeta \end{bmatrix} = \mathbf{A}(\mathbf{q})\mathbf{u} \quad (27)$$

Where, $\mathbf{A}(\mathbf{q})$ is the attitude matrix associated with the measured quaternion at t_{obs} .

5. The cost function $J(\Theta)$ is then computed from Eq. 26, and then another set of Θ is chosen so that the cost function is minimized.

The objective 'cost' function in equation 26 is an *implicit function* of initial conditions Θ . Therefore, any optimization methods, differential evolution or gradient based, can be applied to obtain the best estimated values for the vector of the unknown parameters.

In other words, our optimization problem is then; Find values of the variables Θ that minimize the objective function $\{J(\Theta)\}$ while satisfying the following constraint

$$\Theta_{obt} \in [\Theta_o - \delta\Theta_o, \Theta_o + \delta\Theta_o] \quad (28)$$

Where δ is a very small scalar value that could be chosen based on the accuracy of the measurements, and Θ_o is the initial guess for the unknown parameters, in this case we could start with the on-board initial quaternion and rates for each batch.

5.2 Least Squares Trajectory Estimates

The least squares trajectory estimates is based on finding the trajectory and the model parameters for which the square of the difference between the modeled observations and the actual measurements becomes as small as possible [13]. In other words, find a trajectory which best fits the observations in a least-squares of the residuals sense, as illustrated in figure (10).

The dynamic equations for the quaternion and angular rates are given in Eq. 16 with an initial value $\mathbf{x}_o = \mathbf{x}(t_o)$ at epoch t_o , where $\mathbf{x}(t_i)$ is the state vector defined on Eq. 12.

Furthermore, if we rewrite the measurements ($\eta(t)$ and $\zeta(t)$) in a vector form as

$$\mathbf{z} = (z_1 \ . \ . \ . \ . \ z_{2n})^T = (\eta_1 \ \zeta_1 \ . \ . \ . \ . \ \eta_n \ \zeta_n)^T \quad (29)$$

\mathbf{z} is an $2n$ -dimensional vector of measurements taken at times t_1, \dots, t_n . The observations are described by

$$z_i(t_i) = g_i(t_i, \mathbf{x}(t_i)) + \epsilon_i = h_i(t_i, \mathbf{x}_o) + \epsilon_i \quad (30)$$

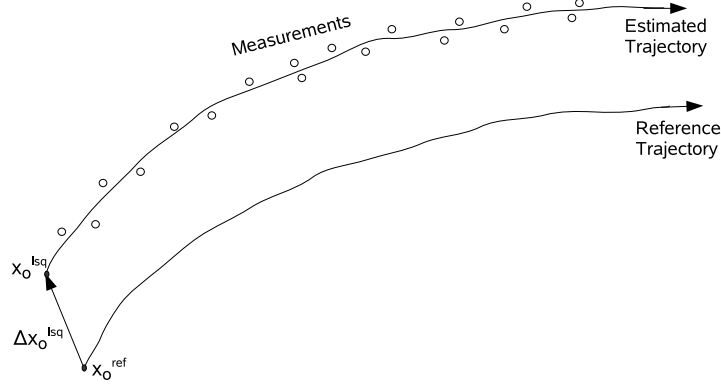


Figure 10: The parameters of a reference trajectory are corrected using the measurements and the least squares fit.

or briefly

$$\mathbf{z} = \mathbf{h}_i(\mathbf{x}_o) + \epsilon \quad (31)$$

Where, g_i denotes the model value of the i^{th} observation as a function of time t_i and the instantaneous state $\mathbf{x}(t_i)$, whereas h_i denotes the same value as a function of the state \mathbf{x}_o at the reference epoch t_o . The quantities ϵ_i account for the difference between the actual and the modeled observations due to measurement errors, which are usually assumed to be randomly distributed with zero mean value.

The least squares trajectory estimates problem may be defined as finding the state \mathbf{x}_o^{lsq} , that minimizes the cost function

$$J(\mathbf{x}_o) = \epsilon^T \epsilon = (\mathbf{z} - \mathbf{h}_i(\mathbf{x}_o))^T (\mathbf{z} - \mathbf{h}_i(\mathbf{x}_o)) \quad (32)$$

The linearized version of Eq. 16 around \mathbf{x}_o^{ref} , which is initially given from the on-board coarse attitude data, could be written in the form

$$\begin{aligned} \mathbf{q}(i+1) &= [I_{4 \times 4} + 0.5\Omega(\omega(\mathbf{t}))\delta t_{obs}] \mathbf{q}(i) = \chi(\omega(\mathbf{t})) \mathbf{q}(i) \\ \omega(i+1) &= I_{sc}^{-1}(T_e - \omega(\mathbf{i}) \times I_{sc}\omega(\mathbf{i}))\delta t_{obs} + \omega(i) \end{aligned} \quad (33)$$

where $\delta t_{obs} = t_{obs}(i+1) - t_{obs}(i)$

So, from the above equation, we can write the instantaneous attitude quaternion $\mathbf{q}(t_i)$ as a function of the initial attitude quaternion $\mathbf{q}(t_o)$

$$\mathbf{q}(t_i) = \chi(\omega(\mathbf{t}_{i-1}))\chi(\omega(\mathbf{t}_{i-2}))\dots\chi(\omega(\mathbf{t}_1))\chi(\omega(\mathbf{t}_o))\mathbf{q}(t_o) \quad (34)$$

The solution of the cost function $J(\mathbf{x}_o)$ is given by taking the derivatives w.r.t. \mathbf{x}_o which leads to

$$\Delta\mathbf{x}_o^{lsq} = (H^T H)^{-1}(H^T \Delta\mathbf{z}) \quad (35)$$

Where, H is the partial derivatives of the modeled observations ($\eta(t)$ and $\zeta(t)$) with respect to the state vector at the reference epoch t_o . However, because the observations are only dependent on the attitude quaternions and not on the angular rates then the H could be written as follows

$$H = \frac{\partial h(\mathbf{q}_o)}{\partial \mathbf{q}_o} = \begin{bmatrix} \frac{\partial \eta(t)}{\partial \mathbf{q}(t)} \frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} \\ \frac{\partial \zeta(t)}{\partial \mathbf{q}(t)} \frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} \end{bmatrix} \quad (36)$$

where

$$\frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} = \chi(\omega(\mathbf{t}_{i-1}))\chi(\omega(\mathbf{t}_{i-2}))\dots\chi(\omega(\mathbf{t}_1))\chi(\omega(\mathbf{t}_o)) \quad (37)$$

The new updated values of the initial quaternion for each batch are determined from

$$\mathbf{q}_o^{lsq} = \mathbf{q}_o^{ref} + \Delta\mathbf{q}_o^{lsq} \quad (38)$$

Also, the associated angular rates for this new attitude quaternion batch may be calculated from equation 6.

The results of the least squares trajectory estimates are illustrated in figure 11 for 2 hours batch length and the initial conditions are set by using the on-board attitude. The resulting attitude determination errors (3σ) is less than 4 marcsec along scan and less than 60 marcsec across scan. These results are not better than the estimated by Kalman Filter with good tuning but the Least Squares Estimates algorithm is easier to implement than the Kalman Filter algorithm and also it is not sensitive to initial conditions and the choice of the measurement and system noise covariance matrices as in KF case.

Furthermore, figure 12 demonstrates the effect of different batch interval on the attitude estimation. Three batch intervals are investigated; 0.5 hour, 1 hour and 3.5 hours, in which we can see that increasing the batch interval to 3.5 hours has more attitude errors than the other two. On the other hand, decreasing the interval less than one hour has no significant effect on the total attitude errors.

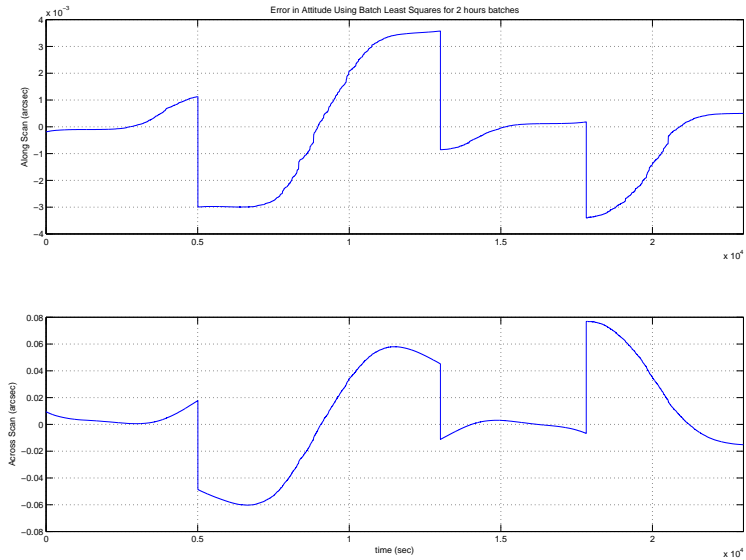


Figure 11: The attitude errors using least squares trajectory for 2 hours batch.

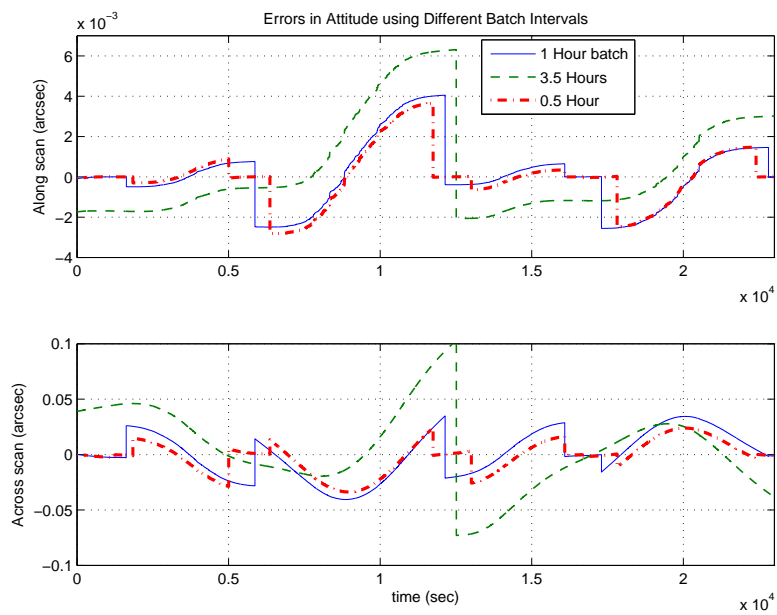


Figure 12: The attitude errors using batch intervals.

Conclusion and Decision

An algorithm for the Gaia On-Ground Attitude (OGA1) determination is designed and discussed in this report. The Extended Kalman Filter (EKF) is used to produce an accurate on-ground attitude estimation. It utilizes the on-board star tracker measurements for coarse spacecraft quaternion which have errors of about several arcsec and combines it with the payload scientific telescopes measurements which have accuracy of less than milli-arcsec.

The obtained results, using the prescribed algorithm in this study, meet the expected OGA1 determination requirements. The attitude determination errors and the rate measurement errors (along scan and across scan) have been validated using the proposed EKF algorithm. Moreover, in this study, we design and establish a Batch Least Square (BLS) estimation algorithm to demonstrate the differences between several estimation techniques in terms of accuracy, speed and robustness. The accuracy obtained using the BLS is less by a factor of two than the accuracy obtained using the EKF. Moreover, the required processing time using the EKF is less by factor of three than using the BLS. On the other hand, the robustness for both methods are almost the same.

As a conclusion of all the simulation results obtained in this study we consider the EKF to be the best OGA1 algorithm as it meets all the expected requirements with less processing time than the BLS.

References

- [1] Frederic Safa, “A Summary of the Gaia Spacecraft Design Document”, 15 March 2006.
- [2] Samaan M. A., Theil, S., “Accurate On-Ground Attitude Determination for the Gaia Mission Using Kalman Smoother”, AAS/AIAA Space Flight Mechanics Meeting, Sendona, AZ, Paper 07-100, Jan. 2007.
- [3] Astrium EADS, “GAIA AOCS Preliminary Design and Performance Analysis”, GAIA-ASU-TCN-ESM-00027, Feb. 2007.
- [4] Lindegren L., “Attitude Parameterization for GAIA”, SAG-LL-30, 1 July 2000.
- [5] L. Lindegren, “ Proposed Prototype Processes for the GAIA Global Iterative Solution”, GAIA-LL-34-V.2, 9 March 2001.
- [6] C. de Boor, “ A Practical Guide to Splines”, Springer 1978.
- [7] Astrium, “Autonomous Star Tracker Requirements Specification”, Issue 1, 2006.
- [8] Wertz, J. R., “Spacecraft Attitude Determination and Control”, Kluwer Academic Publishers, the Netherlands, 1990.
- [9] Gelb, A. , “ Applied Optimal Estimation”, MIT Press, Cambridge, MA, 1974.
- [10] Rauch, H.E., Tung, F., and Striebel, C.T. , “ Maximum Likelihood Estimation of Dynamic Systems”, AIAA Journal, Vol.3, No. 8, August 1965.
- [11] Crassidis, and J.L. , Junkins, J.L., “Optimal Estimation of Dynamic Systems”, Chapman and Hall/CRC, 2004.
- [12] Ravindra V. Jategaonkar, “Flight Vehicle System Identification”, American Institute for Aeronautics and Astronautics, Vol. 216, 2006.
- [13] Montenbruck O., and Gill E. “Satellite Orbits - Models, Methods, and Applications”, Springer-Berlin, 2000.

ON-GROUND ATTITUDE AND TORQUE RECONSTRUCTION FOR THE GAIA MISSION

Malak A. Samaan* and Stephan Theil†

The work presented in this paper concerns the accurate On-Ground Attitude (OGA) reconstruction for the astrometry spacecraft Gaia in the presence of disturbance and control torques acting on the spacecraft. The reconstruction of the expected environmental torques which influence the spacecraft dynamics will be also investigated.

The telemetry data from the spacecraft will include the on-board real time attitude which is of order of several arcsec. This *raw* attitude is the starting point for the further attitude reconstruction. The OGA will use the inputs from the field coordinates of known stars (attitude stars) and also the field coordinate differences of objects on the Sky Mapper (SM) and Astrometric Field (AF) payload instruments to improve this raw attitude. The on-board attitude determination uses Kalman Filter (KF) to minimize the attitude errors and produce more accurate attitude estimation than the pure star tracker measurement. Therefore the first approach for the OGA will be an adapted version of KF. Furthermore, we will design a batch least squares algorithm to investigate having more accurate OGA estimation. Finally, a comparison between these different attitude determination techniques in terms of accuracy, robustness, speed and memory required will be evaluated in order to choose the best attitude algorithm for the OGA. The expected resulting accuracy for the OGA determination will be on the order of milli-arcsec.

INTRODUCTION

The principal feature of the Gaia astrometry mission is to accurately measure the positions, distances, space motions, and many physical characteristics of some one billion stars in our Galaxy and beyond on orbit at the vicinity of the second Lagrange point L2. In order to achieve the targeted measurement accuracies of the Gaia imaged stars, the real time on-board attitude should be improved using the on-ground attitude reconstruction.²

The main objective of the work presented here is to reconstruct the *non-real-time* On-Ground Attitude (OGA) with very high accuracy for further processing. The accuracy requirements for the attitude reconstruction for the Gaia on-ground data analysis would enforce studying different attitude estimation techniques and choosing the best one in terms of accuracy, robustness and speed. The current goal for the OGA accuracy is about 50 milli-arcsec for the along and across scan attitude accuracy. These techniques shall be part of the Initial Data Treatment (IDT) chain.

The Gaia scientific payload is used as a high accuracy stellar gyro for achieving the attitude determination and enabling Time Delay Integration (TDI) mode. The star speed is calculated along and across scan from Sky Mapper (SM) and Astrometric Field (AF) measurements. When considering

*Assistant Research Professor, Center for Applied Space Technology and Microgravity, University of Bremen, Germany, samaan@zarm.uni-bremen.de.

†Head of the GNC and Avionics, German Aerospace Center (DLR), Am Fallturm 1 - Bremen, Germany, stephan.theil@dlr.de.

the two telescopes, one obtains four angular speed measurements, from which the spacecraft rigid body angular speed vector can be calculated. It should be mentioned that SM-1 is seen by telescope No. 1 and SM-2 by telescope No. 2. The Sky Mapper measurements consist of the determination of the centroid crossing times of the stars over each of the SM CCD arrays.³

The Gaia Initial Data Treatment (IDT) will process the newly arrived telemetry data and also the various pieces of auxiliary data from the Gaia spacecraft. The IDT will have many processes and subtasks, one of them is the attitude reconstruction or the OGA determination.

In order to match the new observations reliably with the existing source list, it is essential to have an accurate attitude. It is therefore suggested to carry out an early attitude updating during the IDT. This will require a cross matching with a dedicated Attitude Star Catalog, followed by an attitude updating.

At this stage we are only aiming for attitude accuracy of a tens of milliarcsec level, so some simplifications may be allowed.

The IDT will have to run in several concurrent processes, and the attitude improvement is done separately for each process. The same time interval will therefore be processed several times, and we need only to update a given attitude interval if it has not already been updated in a previous process.

Through understanding of the dynamical behavior of the spacecraft, the accuracy of the reconstructed attitude and torques can be improved, and is therefore of great relevance for obtaining the highest accuracies and best overall statistical properties for the main scientific products of the mission; the Astrometric parameters of the target stars.

Given all these mission data and parameters, the attitude data simulation, in the form of the spacecraft quaternion and angular velocities, could be simulated for any time span. In order to simulate the algorithms for OGA determination, we have to simulate attitude observations with the following requirements:

- The simulated data shall be provided for the length of at least one great circle (6 hours).
- The data shall contain the true (ideal) attitude quaternion as well as the true rate in the body fixed frame. Also, the raw (measured) attitude quaternion.
- The simulated observations of attitude stars shall contain:
 - The time of observation.
 - The indicator for the FOV.
 - The position on the sky of the observed star (the right ascension α , and declination δ).
 - The field coordinates of the star (η and ζ) for SM and AF1-9 and G-magnitude.
 - The transit ID and the Attitude Star Catalog (ASC).
 - Cross-match table.
 - The attitude quaternion at each observation time will be calculated using the B-Spline as described in the next section.

GAIA ATTITUDE PRESENTATION

The Gaia spacecraft attitude is considered to be represented by the quaternion vector defined as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{13} \\ q_4 \end{bmatrix} \quad (1)$$

with

$$\mathbf{q}_{13} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\mathbf{n}} \sin\left(\frac{\theta}{2}\right) \text{ and } q_4 = \cos\left(\frac{\theta}{2}\right) \quad (2)$$

where $\hat{\mathbf{n}}$ is a unit vector corresponding to the axis of rotation, and θ is the rotation angle.

The elements of the quaternion satisfy a unit norm constraint

$$\mathbf{q}^T \mathbf{q} = \mathbf{q}_{13}^T \mathbf{q}_{13} + q_4^2 = 1. \quad (3)$$

The frequency of the on-board attitude (1 Hz) is different from the frequency of the star centroiding and crossing observation times in the SM and AF which measured in an uneven sequences and in order of nanosec. In order to start the OGA determination, the attitude quaternion at every time of observation should be evaluated by interpolating the on-board attitude.

Both the on-board real-time attitude and the on-ground reconstructed attitude, and perhaps also the nominal scanning law attitude,⁴ can be represented by means of cubic splines followed by normalization (to validate the unit length condition). So, according to⁵ at any time interval $[t_{beg}, t_{end}]$ there is a vector $\mathbf{s}(t)$ consists of four cubic spline functions $s_m(t), m = 1, \dots, 4$ from this vector the attitude quaternion at time t is calculated from

$$\mathbf{q}(t) = \mathbf{s}(t) / \text{norm}(\mathbf{s}(t)). \quad (4)$$

Also we can derive the instantaneous scan rate (ω) in the fields of view by computing the time derivative of the quaternion at time t from

$$\dot{\mathbf{q}}(t) = [\dot{\mathbf{s}} - \mathbf{q}\mathbf{q}^T \dot{\mathbf{s}}] / \text{norm}(\mathbf{s}(t)). \quad (5)$$

In this case the scan rate at any given time is derived from

$$\omega(t) = \frac{2}{\text{norm}(\mathbf{q})} \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \dot{\mathbf{q}}(t). \quad (6)$$

Each of the functions $s_m(t)$ is a cubic spline defined on $[t_{beg}, t_{end}]$, it is therefore continuous in this interval and its first two derivatives $\dot{s}_m(t)$ and $\ddot{s}_m(t)$ are also continuous. The third derivatives is discontinuous at discrete points, called the knots, and constant between the knots. The frequency of knots determines the flexibility of the spline, or in other words, more knots gives more flexible spline. The used knot sequence is used for all the four components of $\mathbf{s}(t)$, with a knot interval of 15 sec.

A cubic spline can be represented in a number of different ways, e.g. by means of its value and second derivative at each knot. However, the method proposed here is to write the spline as a

linear combination of non-negative local basis functions, called B-splines (See⁶). Let $B_n(t), n = 1, \dots, N$, be the N B-splines defined on the attitude interval $[t_{beg}, t_{end}]$. Because the B-splines are linearly independent basis functions, N is also the number of degrees of freedom of the spline. Then

$$s_m(t) = \sum_{n=1}^N S_{mn} B_n(t), \quad m = 1, \dots, 4 \quad (7)$$

and

$$\dot{s}_m(t) = \sum_{n=1}^N S_{mn} \dot{B}_n(t), \quad m = 1, \dots, 4. \quad (8)$$

The B-splines B_n (and their derivatives $\dot{B}_n(t)$) are uniquely defined by the knot sequence

$$\tau = \{\tau_{-1}, \tau_0, \tau_1, \tau_2 \equiv t_{beg}, \tau_3, \dots, \tau_{N-2}, \tau_{N-1} \equiv t_{end}, \tau_N, \tau_{N+1}, \tau_{N+2}\} \quad (9)$$

The knot sequence can be evaluated for any time t according to a simple and stable recurrence relation. Moreover, it should be non-decreasing ($\tau_i \leq \tau_{i+1}, i = -1, \dots, N+2$). At any given time $t \in [\tau_n, \tau_{n+1}]$ the only non-zero B-splines are B_{n-1}, B_n, B_{n+1} and B_{n+2} . Therefore, for the whole attitude interval $[t_{beg}, t_{end}]$ the non-zero B-splines are B_1, B_2, \dots, B_N . The additional knots on either side of the attitude interval are needed to define B_1, B_2, B_3 and B_{N-2}, B_{N-1}, B_N , and can be chosen arbitrary as long as τ is non-decreasing.⁶

External Torque Estimation

The first task in the OGA determination is to reconstruct the external torque as a continuous function of time from the telemetry attitude dynamics. In order to achieve this task, we consider the spacecraft as a rigid body then we can express the angular velocity using the following equation;

$$\dot{\omega}(t) = I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \quad (10)$$

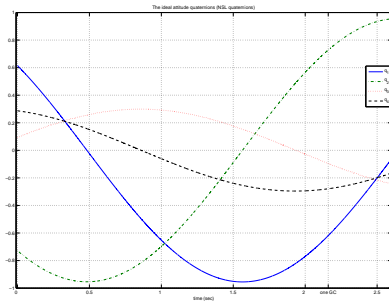
where I_{sc} is the moment of inertia of the Gaia S/C and T_e is the total disturbance and control torques acting on the spacecraft.

At each observation time the attitude quaternion is calculated from Eq. 4 and the time derivative of the quaternion from Eq. 5. By using Eq. 6 the angular rates at each observation time are determined.

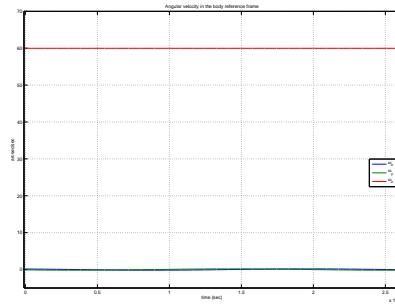
Now to calculate the estimated external torque, we could rewrite Eq. 10 in the form

$$T_e(t_{obs}) = I_{sc}\dot{\omega} + \omega \times I_{sc}\omega. \quad (11)$$

The Nominal Scanning Law (NSL)⁴ is used to simulate the attitude quaternion for more than one great circle (6 hours) as illustrated in figure (1(a)). Consequently, when Eq. 6 is used to calculate the angular rates at each time step the resulting rates are illustrated in figure (1(b)). On the other hand, figure (2) demonstrates the resulting external torques which are supposed to be the sum of all the environmental disturbance torques, internal torques and control torques acting on the Gaia spacecraft while following the NSL.



(a) The simulated quaternion using NSL.



(b) The simulated quaternion using NSL.

Figure 1 The simulated attitude and rates using the B-Splines.

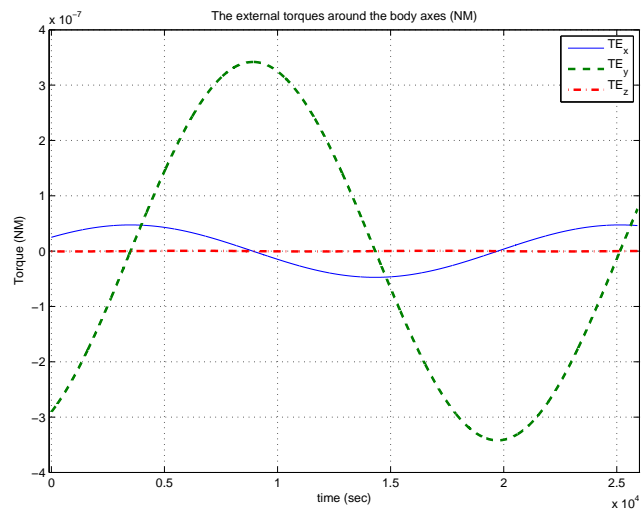


Figure 2 The external (disturbance and control) torques acting on the S/C.

THE KALMAN FILTER MODEL

The Kalman filter state vector for the Gaia OGA is consisting of the 4 components of the attitude quaternion and the three components for the angular rates.

$$x = [q_1 \ q_2 \ q_3 \ q_4 \ \omega_x \ \omega_y \ \omega_z]^T. \quad (12)$$

The system model equation has the following form

$$\dot{x}(t) = f(x(t), t) + G(x(t), t)w(t). \quad (13)$$

Also, the measurement model at time t_k is given by

$$Y_k = h(x(t_k), t) + \nu(t) \quad (14)$$

where $w(t)$ is the process noise and $\nu(t)$ is the measurement noise, both are discrete Gaussian white-noise processes

$$\begin{aligned} w(t) &\simeq N(0, Q(t)), \\ \nu(t) &\simeq N(0, R(t)). \end{aligned} \quad (15)$$

The Kalman filter dynamic equations of state are given by

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \frac{1}{2}\Omega(\omega)\mathbf{q}(t) \\ \dot{\omega}(t) &= I_{sc}^{-1}(T_e - \omega \times I_{sc}\omega) \end{aligned} \quad (16)$$

where

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}. \quad (17)$$

Also the external torque T_e is calculated at each observation time from Eq. 11.

The Kalman filter propagation equations are defined by

$$\begin{aligned} \dot{x}(t) &= F(t)x(t) + G(t)w(t) \\ \dot{P} &= FP + PF^T + GQG^T \end{aligned} \quad (18)$$

where P is the covariance matrix.

The F matrix is given by

$$F = \begin{bmatrix} 0.5\Omega(\omega) & -0.5\Xi(\mathbf{q}) \\ 0_{3 \times 4} & F_{\dot{\omega}\omega} \end{bmatrix} \quad (19)$$

where

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (20)$$

$$F_{\dot{\omega}\omega} = -I_{sc}^{-1}([\omega \times] I_{sc} - [I_{sc} \omega \times]). \quad (21)$$

The matrix $[\underline{a} \times]$ is the skew symmetric matrix of the vector \underline{a} .

Also, the Kalman filter update equations are defined by

$$\begin{aligned} \hat{x}_k(+) &= \hat{x}_k(-) + K_k [Y_k - h_k(\hat{x}_k(-))] \\ P_k(+) &= [I - K_k H_k(\hat{x}_k(-))] P_k(-) \\ K_k &= P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1}. \end{aligned} \quad (22)$$

The measurement (Y_k) is essentially an association of a certain time instant t with certain field coordinate angles of the measured star. Each measurement vector consists of the along scan angle (η_m) and the across scan angle (ζ_m) for a certain field of view in the instrument frame.

The calculated field angles at each observation time are given by

$$h(x_t) = [\eta_c^i \quad \zeta_c^i] \quad (23)$$

where $i = 1$ or 2 is field of view number. These field angles depend on the basic angle between the two fields of view, which is equal to 106.5° .¹

The measurement sensitivity matrix is given by

$$H_k = \frac{\partial h(x_t)}{\partial x(t)} = \begin{bmatrix} \frac{\partial \eta_k}{\partial \mathbf{q}} & 0_{1 \times 3} \\ \frac{\partial \zeta_k}{\partial \mathbf{q}} & 0_{1 \times 3} \end{bmatrix}. \quad (24)$$

The Kalman Filter Results

The Kalman filter algorithm is designed and implemented using the above system model and equations. The measurement noise matrix R is chosen such that;

$$R = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix} \quad (25)$$

The standard deviation for the field angle errors along scan (σ_η) is considered to be 0.1 marcsec and for across scan (σ_ζ) is considered to be 0.5 marcsec. A total of 10^5 of the field angles measurements were simulated for about 7 hours, corresponding to about 4 measurements per sec.

Figure 3 illustrates the attitude estimation errors for the along scan and across scan directions. The along scan direction is around the rotation axis or z-axis while the across scan direction is considered to be the average of the x and y axes or the axes perpendicular to the spin axis. The estimated external torques (which illustrated in figure (2)) are used to obtain the propagated states (\mathbf{q} and ω) at each observation time. In the states update step the Gaia scientific instrument measurements (the field angles) are combined with the Kalman filter simulation.

Using different initial values, the resulting attitude determination errors (3σ) were always less than 2 marcsec along scan and 20 marcsec across scan. The estimated angular rate errors are also

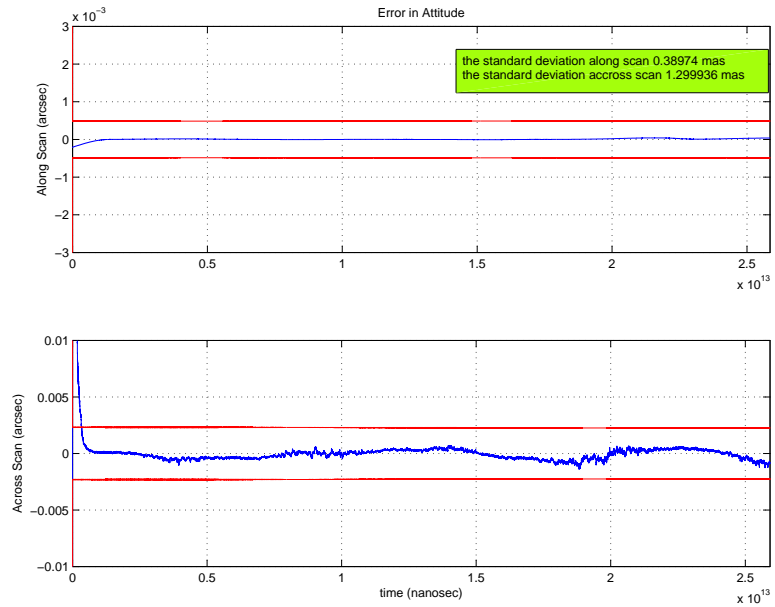


Figure 3 The attitude errors using Kalman Filter.

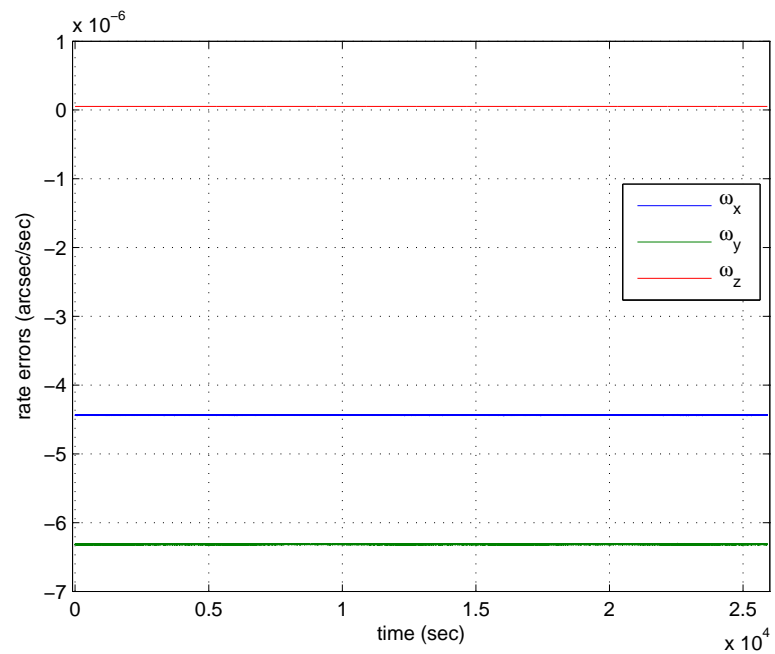


Figure 4 The angular rates errors using Kalman Filter.

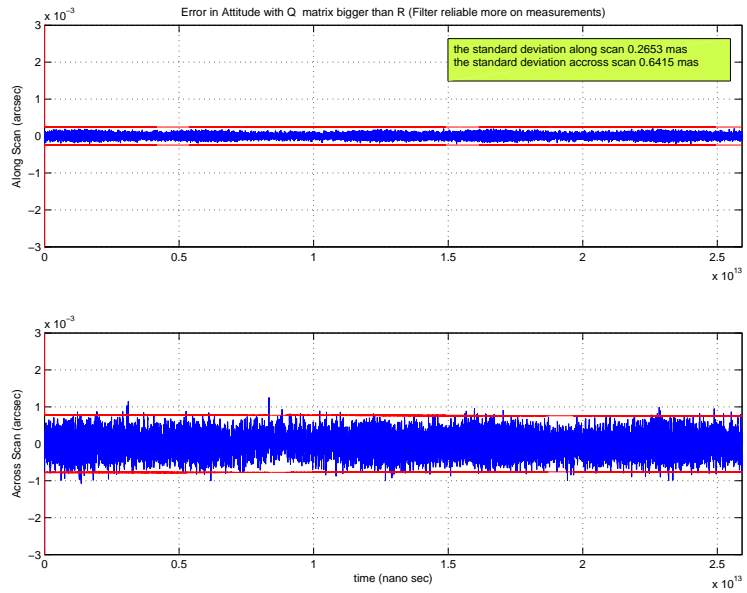


Figure 5 The attitude errors using different tuning.

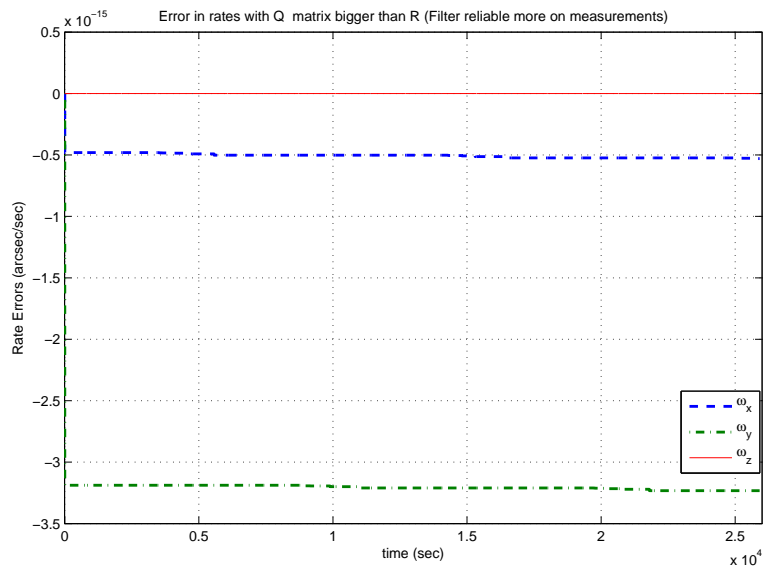


Figure 6 The angular rates errors using different tuning.

shown in figure 4. The resulting angular rates errors are less than 0.007 marcsec/sec around the three axes. The process noise matrix Q used for the Kalman filter propagation equations (18) are considered to be

$$Q = \text{diag}([(10^{-12})^2, (10^{-12})^2, (10^{-12})^2, (10^{-12})^2, (5 \times 10^{-10})^2, (5 \times 10^{-10})^2, (10^{-11})^2])$$

The Kalman filter update equations are based on the updated value of the covariance matrix $P(t)$ which is calculated from Eq. 18. On the other hand, the $P(t)$ is a function of the process noise matrix Q , so the KF tuning for the values of the diagonal element of the Q matrix is very important in determining the resulting accuracy of the estimation. Figures 5 and 6 demonstrate the effect of changing the process noise matrix to be much bigger than the measurement noise matrix R or in other word, the KF will rely more on measurements than on the dynamic model. The process noise matrix Q in this case is chosen to be

$$Q = \text{diag}([(10^{-6})^2, (10^{-6})^2, (10^{-6})^2, (10^{-6})^2, (5 \times 10^{-5})^2, (5 \times 10^{-5})^2, (2 \times 10^{-6})^2])$$

In figure 5 the overall along scan and across quaternion errors is better than that illustrated in figure 3 but it is more noisy. This is the more realistic case for Gaia. Moreover, the angular rates estimation illustrated in figure 6 is better than the angular rates errors shown in figure 4. All the estimated attitude and rates shown in the previous figures (3, 4, 5, and 6) are based on ideal (error-less) star catalog.

At the beginning of the IDT process, the OGA will be using an initial coarse star catalog that will have errors of about 50 marcsec until the more accurate star catalog is created. At this stage we suggest to have the OGA dependent more on the system dynamic model or in other words to choose the process noise matrix such that it should be less than the measurement noise matrix. After the Astrometric Global Iterative Solution (AGIS)⁵ has become available (after 10 months or more) we can easily change the process noise matrix to let the filter rely more on the measurement which will be more accurate than the system model.

Figures 7 illustrates the effects of using star catalog that has errors of about 50 marcsec in both the along and across scan directions on the estimation of the attitude errors when the system rely more on the system dynamic model. On the other hand, figure 8 shows the resulting attitude errors when the system rely more on the measurements.

THE BATCH LEAST SQUARES

The Batch Least Squares (BLS) estimation is the process that can be used to estimate the state vector at an epoch time after accumulating batch measurement data for a given interval. The BLS solution selects a state estimate x that minimizes the sum of the squares of the calculated residuals. Although the BLS is computationally intensive for a large number of measurements, it leads to a good estimation of the state vector.¹¹ The BLS estimation could be done using one of two approaches; the first approach is the Maximum Likelihood estimates and the other is the Least Squares Trajectory estimates.

Maximum Likelihood Estimates

The basic idea of the maximum likelihood function for parameters estimation is to obtain the best attitude quaternion and angular rates for a given interval of time. In order to have the most accurate

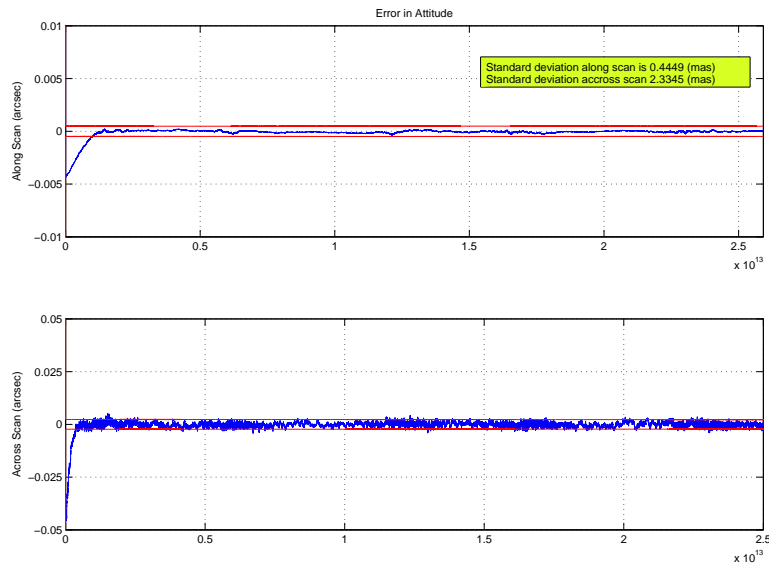


Figure 7 The attitude errors using catalog with 50 marcsec errors.

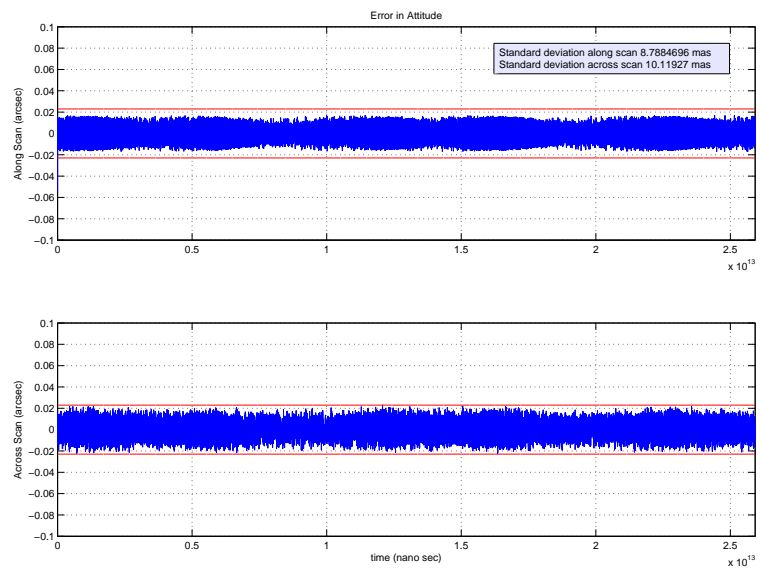


Figure 8 The attitude errors using catalog with 50 marcsec errors and different KF tuning.

estimate of the unknown parameters of the Gaia OGA for each batch of measurements, which include the initial conditions of the attitude quaternions and the initial condition of the angular rates, we have to minimize the following cost function;

$$J(\Theta) = \frac{1}{2} \sum_{k=1}^N (\eta_{1,k}^m - \eta_{1,k}^e)^2 + (\eta_{2,k}^m - \eta_{2,k}^e)^2 + (\zeta_{1,k}^m - \zeta_{1,k}^e)^2 + (\zeta_{2,k}^m - \zeta_{2,k}^e)^2 \quad (26)$$

where Θ is the unknown parameters vector which includes the initial conditions for quaternions and angular rates for each batch interval and also the constant value for the rate bias for that batch. $\eta_{i,k}^m$ and $\zeta_{i,k}^m$; are the measured field coordinate angles for field of view number i ($i = 1 \text{ or } 2$) at the observation time t_k , and

$\eta_{i,k}^e$ and $\zeta_{i,k}^e$; are the estimated field coordinate angles for field of view number i at the observation time t_k , which are calculated as follows;

1. For any set of Θ we can use the S/C dynamics equations (Eq. 16) to obtain the quaternion and angular rates for the given interval time (i.e. 1 hour). Then the associated B-Splines is derived for these estimated quaternion by using Eq. 7.
2. At any observation time (t_{obs}) the measured quaternion vector (\mathbf{q}_m) is calculated using Eq. 4.
3. The position on the sky of the observed star (α_i, δ_i) for the associated FOV is used to calculate the proper direction of the observed star (\mathbf{u}),
4. The estimated field coordinate angles are then computed from

$$\begin{bmatrix} \cos \zeta \cos \eta \\ \cos \zeta \sin \eta \\ \sin \zeta \end{bmatrix} = \mathbf{A}(\mathbf{q})\mathbf{u} \quad (27)$$

where, $\mathbf{A}(\mathbf{q})$ is the attitude matrix associated with the measured quaternion at t_{obs} .

5. The cost function $J(\Theta)$ is then computed from Eq. 26, and then another set of Θ is chosen so that the cost function is minimized.

The objective 'cost' function in Eq. 26 is an *implicit function* of the initial conditions vector Θ . Therefore, any optimization methods (e.g. differential evolution or gradient based) can be applied to obtain the best estimated values for the vector of the unknown parameters.

In other words, our optimization problem is then; Find the best values of the vector of variables Θ that minimize the objective function $\{J(\Theta)\}$ while satisfying the following constraint

$$\Theta_{obt} \in [\Theta_o - \delta\Theta_o, \Theta_o + \delta\Theta_o] \quad (28)$$

where δ is a very small scalar value that could be chosen based on the accuracy of the measurements, and Θ_o is the initial guess for the unknown parameters, in this case we could start with the on-board initial quaternion and rates for each batch.

Least Squares Trajectory Estimates

The least squares trajectory estimates is based on finding the trajectory and the model parameters for which the square of the difference between the modeled observations and the actual measurements becomes as small as possible.¹³ In other words, find a trajectory which best fits the observations in a least-squares of the residuals sense, as illustrated in figure (9).

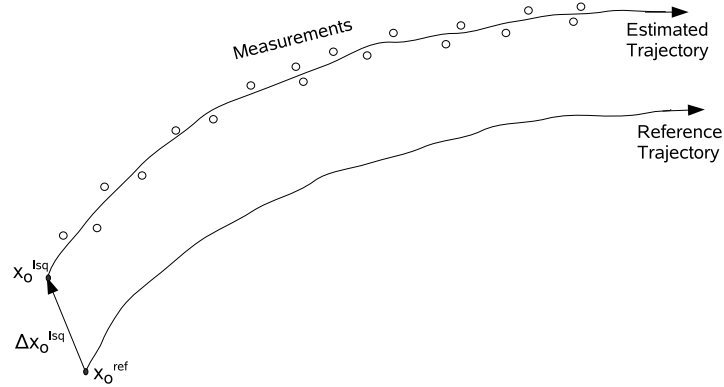


Figure 9 The parameters of a reference trajectory are corrected using the measurements and the least squares fit.

The dynamic equations for the quaternion and angular rates are given in Eq. 16 with an initial value $\mathbf{x}_o = \mathbf{x}(t_o)$ at epoch t_o , where $\mathbf{x}(t_i)$ is the state vector defined on Eq. 12.

Furthermore, we could rewrite the measurements ($\eta(t)$ and $\zeta(t)$) in a vector form as

$$\mathbf{z} = (z_1 \quad . \quad . \quad . \quad z_{2n})^T = (\eta_1 \quad \zeta_1 \quad . \quad . \quad . \quad \eta_n \quad \zeta_n)^T \quad (29)$$

\mathbf{z} is an $2n$ -dimensional vector of measurements taken at times t_1, \dots, t_n . The observations are described by

$$z_i(t_i) = g_i(t_i, \mathbf{x}(t_i)) + \epsilon_i = h_i(t_i, \mathbf{x}_o) + \epsilon_i \quad (30)$$

or briefly

$$\mathbf{z} = \mathbf{h}_i(\mathbf{x}_o) + \epsilon \quad (31)$$

where, g_i denotes the model value of the i^{th} observation as a function of time t_i and the instantaneous state $\mathbf{x}(t_i)$, whereas h_i denotes the same value as a function of the state \mathbf{x}_o at the reference epoch t_o . The quantities ϵ_i account for the difference between the actual and the modeled observations due to measurement errors, which are usually assumed to be randomly distributed with zero mean value.

The least squares trajectory estimates problem may be defined as finding the state \mathbf{x}_o^{lsq} , that minimizes the cost function

$$J(\mathbf{x}_o) = \epsilon^T \epsilon = (\mathbf{z} - \mathbf{h}_i(\mathbf{x}_o))^T (\mathbf{z} - \mathbf{h}_i(\mathbf{x}_o)). \quad (32)$$

The linearized version of Eq. 16 around \mathbf{x}_o^{ref} , which is initially given from the on-board coarse attitude data, could be written in the form

$$\begin{aligned}\mathbf{q}(i+1) &= [I_{4 \times 4} + 0.5\Omega(\omega(\mathbf{t}))\delta t_{obs}] \mathbf{q}(i) = \chi(\omega(\mathbf{t}))\mathbf{q}(i) \\ \omega(i+1) &= I_{sc}^{-1}(T_e - \omega(\mathbf{i}) \times I_{sc}\omega(\mathbf{i}))\delta t_{obs} + \omega(i)\end{aligned}\quad (33)$$

where $\delta t_{obs} = t_{obs}(i+1) - t_{obs}(i)$.

So, from the above equation, we can write the instantaneous attitude quaternion $\mathbf{q}(t_i)$ as a function of the initial attitude quaternion $\mathbf{q}(t_o)$

$$\mathbf{q}(t_i) = \chi(\omega(\mathbf{t}_{i-1}))\chi(\omega(\mathbf{t}_{i-2}))\dots\chi(\omega(\mathbf{t}_1))\chi(\omega(\mathbf{t}_o))\mathbf{q}(t_o) \quad (34)$$

The solution of the cost function $J(\mathbf{x}_o)$ is given by taking the derivatives w.r.t. \mathbf{x}_o which leads to

$$\Delta \mathbf{x}_o^{lsq} = (H^T H)^{-1}(H^T \Delta \mathbf{z}) \quad (35)$$

where, H is the partial derivatives of the modeled observations ($\eta(t)$ and $\zeta(t)$) with respect to the state vector at the reference epoch t_o . However, because the observations are only dependent on the attitude quaternions and not on the angular rates then the H could be written as follows

$$H = \frac{\partial h(\mathbf{q}_o)}{\partial \mathbf{q}_o} = \begin{bmatrix} \frac{\partial \eta(t)}{\partial \mathbf{q}(t)} & \frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} \\ \frac{\partial \zeta(t)}{\partial \mathbf{q}(t)} & \frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} \end{bmatrix} \quad (36)$$

where

$$\frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_o} = \chi(\omega(\mathbf{t}_{i-1}))\chi(\omega(\mathbf{t}_{i-2}))\dots\chi(\omega(\mathbf{t}_1))\chi(\omega(\mathbf{t}_o)). \quad (37)$$

The new updated values of the initial quaternion for each batch are determined from

$$\mathbf{q}_o^{lsq} = \mathbf{q}_o^{ref} + \Delta \mathbf{q}_o^{lsq}. \quad (38)$$

Also, the associated angular rates for this new attitude quaternion batch may be calculated from equation 6.

The results of the least squares trajectory estimates are illustrated in figure 10 for 2 hours batch length and the initial conditions are set by using the on-board attitude. The resulting attitude determination errors (3σ) is less than 4 marcsec along scan and less than 60 marcsec across scan. These results are not better than the estimated by Kalman filter with good tuning but the least squares estimates algorithm is easier to implement than the Kalman filter algorithm and also it is not sensitive to initial conditions and the choice of the measurement and system noise covariance matrices as in KF case.

Furthermore, figure 11 demonstrates the effect of using different batch interval on the attitude estimation errors. Three batch intervals are investigated; 0.5 hour, 1 hour and 3.5 hours, in which we can see that increasing the batch interval to 3.5 hours has more attitude errors than the other two. On the other hand, decreasing the interval less than one hour has no significant effect on the total attitude errors.

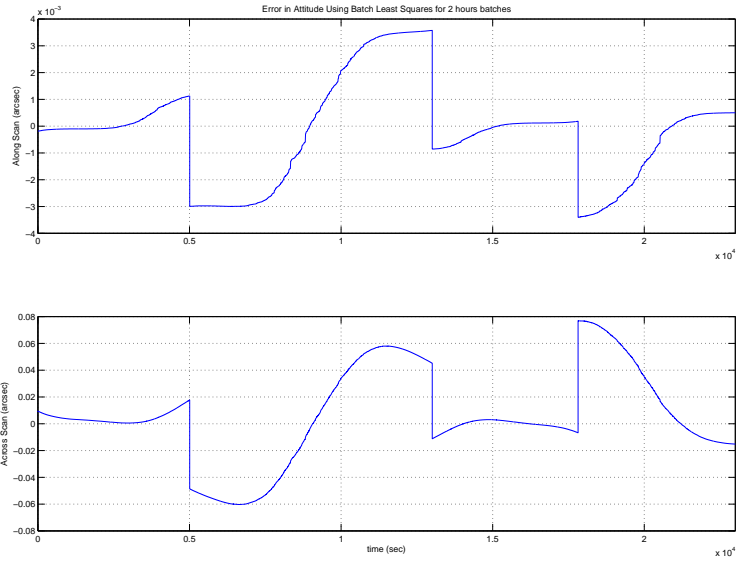


Figure 10 The attitude errors using least squares trajectory for 2 hours batch.

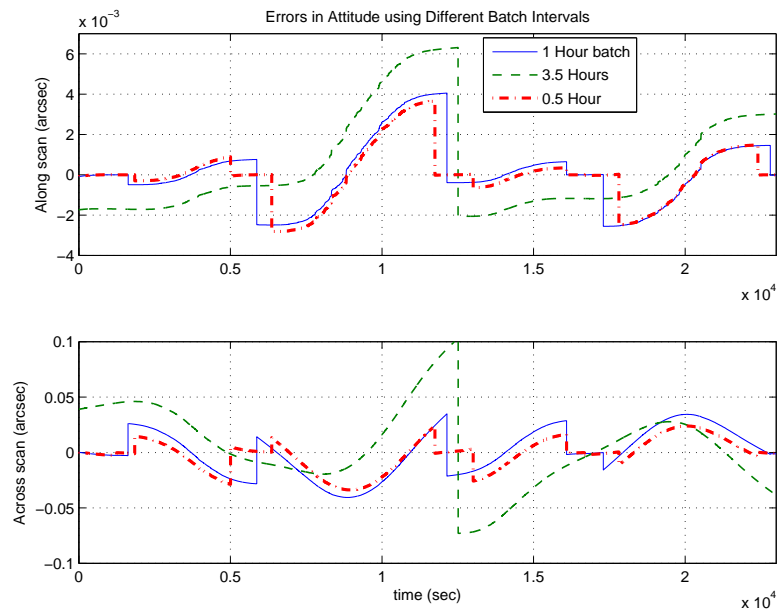


Figure 11 The attitude errors using batch intervals.

CONCLUSION AND DECISION

An algorithm for the Gaia On-Ground Attitude (OGA) determination is designed and discussed in this paper. The Extended Kalman Filter (EKF) is used to produce an accurate on-ground attitude estimation. It utilizes the on-board star tracker measurements for coarse spacecraft quaternion which have errors of about several arcsec and combines it with the payload scientific telescopes measurements which have accuracy of less than milli-arcsec.

The obtained results, using the prescribed algorithm in this work, meet the expected OGA determination requirements. The attitude determination errors and the rate measurement errors (along scan and across scan) have been validated using the proposed EKF algorithm. Moreover, in this study, we design and establish a Batch Least Square (BLS) estimation algorithm to investigate the differences between several estimation techniques in terms of accuracy, speed and robustness. The accuracy obtained using the BLS is less by a factor of two than the accuracy obtained using the EKF. Moreover, the required processing time using the EKF is less by factor of three than using the BLS. On the other hand, the robustness for both methods are almost the same.

As a conclusion of all the simulation results obtained in this study we consider the EKF to be the best OGA algorithm as it meets all the expected requirements with less processing time than the BLS.

ACKNOWLEDGMENT

The authors wish to thank Dr. Ulrich Bastian from University of Heidelberg for his technical advices and his precious support. We wish also to thank the IDT team at University of Barcelona for their collaboration and providing the simulation data to get the current results.

REFERENCES

- [1] Frederic Safa, "A Summary of the Gaia Spacecraft Design Document", 15 March 2006.
- [2] Samaan M. A., Theil, S., "Accurate On-Ground Attitude Determination for the Gaia Mission Using Kalman Smoother", AAS/AIAA Space Flight Mechanics Meeting, Sendona, AZ, Paper 07-100, Jan. 2007.
- [3] Astrium EADS, "GAIA AOCS Preliminary Design and Performance Analysis", GAIA-ASU-TCN-ESM-00027, Feb. 2007.
- [4] Lindegren L., "Attitude Parameterization for GAIA", SAG-LL-30, 1 July 2000.
- [5] L. Lindegren, "Proposed Prototype Processes for the GAIA Global Iterative Solution", GAIA-LL-34-V.2, 9 March 2001.
- [6] C. de Boor, "A Practical Guide to Splines", Springer 1978.
- [7] Astrium, "Autonomous Star Tracker Requirements Specification", Issue 1, 2006.
- [8] Wertz, J. R., "Spacecraft Attitude Determination and Control", Kluwer Academic Publishers, the Netherlands, 1990.
- [9] Gelb, A. , "Applied Optimal Estimation", MIT Press, Cambridge, MA, 1974.
- [10] Rauch, H.E., Tung, F., and Striebel, C.T. , "Maximum Likelihood Estimation of Dynamic Systems", AIAA Journal, Vol.3, No. 8, August 1965.
- [11] Crassidis, and J.L. , Junkins, J.L., "Optimal Estimation of Dynamic Systems", Chapman and Hall/CRC, 2004.
- [12] Ravindra V. Jategaonkar, "Flight Vehicle System Identification", American Institute for Aeronautics and Astronautics, Vol. 216, 2006.
- [13] Montenbruck O., and Gill E. "Satellite Orbits - Models, Methods, and Applications", Springer-Berlin, 2000.

GENERIC DRAG FREE CONTROL SIMULATION – LESSONS LEARNED FROM GRAVITY PROBE B

Ivanka Pelivan,^{*} Sara Smoot,[†] David Hipkins[†] and Stephan Theil[‡]

A generic drag-free simulator has been developed to aid in the design, on-orbit and post-mission data analysis phases of scientific satellite missions. Adaptable to missions as different in nature as Gaia (Global Astrometric Interferometer for Astrophysics) and STEP (Satellite Test of the Equivalence Principle), this simulator will provide necessary modeling capability to increasingly complex future missions. A complete mission software simulator including controls, full-body dynamics and comprehensive spacecraft environment disturbances has been established for Gravity Probe B (GP-B). Reproduction of the mission is being carried out to validate the simulator with actual flight data and refine the underlying models. The importance of this effort lies in the challenge to meet rising science requirements in the area of maximum disturbance rejection. Future missions such as Gaia, STEP, LISA (Laser Interferometer Space Antenna) and others require a minimum of 3 orders of magnitude improvement over the GP-B performance of 10^{-9} m/sec². While technology advancements will certainly be required to achieve these levels, it became increasingly clear to the scientists and engineers who delivered the GP-B results that the ability to monitor and adjust the coupling of spacecraft to subsystem controllers, at all stages of the mission is essential to optimizing mission results. We provide a look at the progress to date of this effort.

INTRODUCTION

GP-B was designed to test two predictions of Einstein's general theory of relativity by measuring the orientations of four high-precision gyroscopes relative to a distant guide star. While in theory a simple concept, the high accuracy required to achieve the mission goals imposed high performance demands on the measurement instrument. The resulting challenging realization led to technology spin-offs now benefitting future science missions that require a comparable or even lower bound on acceptable disturbances. For concept verification, a control software simulator has been established along with the mission development which could be combined with hardware in the loop to test attitude translation control (ATC) and the gyroscope suspension system for GP-B. With GP-B in orbit the simulator has also proven to be invaluable during the initial orbit check-out phase to aid in anomaly resolution, examples of which are given in [1].

The current simulation effort targets to provide a readily available tool for future science missions when addressing pre-flight scenario investigation, to ensure in-flight data quality and to aid in post-mission data reduction.

^{*} Center for Applied Space Technology and Microgravity, University of Bremen, Germany, pelivan@zarm.unibremen.de/W.W.Hansen Experimental Physics Lab, Stanford University, California, U.S.A.

[†] W.W.Hansen Experimental Physics Lab, Stanford University, California, U.S.A.

[‡] German Aerospace Center (DLR), Institute of Space Systems, Bremen, Germany.

THE GRAVITY PROBE B EXPERIENCE

The Gravity Probe B Relativity Mission was launched April 20, 2004 and completed September 28, 2005. It has been described as one of the most technically challenging science satellite missions ever flown by NASA. The experiment was proposed independently by Leonard Schiff and George Pugh shortly after the first successful satellites were delivered into space in the late 1950's and early 1960's. The experiment tests the geodetic and frame-dragging effects predicted by Einstein's General Relativity Theory, by measuring the precession of gyroscopes orbiting at 642 km around the Earth's poles. These two predictions, if valid, would result in a geodetic precession of 6614 milliarcseconds and a drift due to frame-dragging of 42 milliarcseconds after one year. To measure such small changes meant virtually complete elimination of classical torques that could contribute to the movement of the gyroscope spin axis, and an experimental instrument insulated sufficiently such that systematic variations in the data acquisition do not mask the relativistic effects. These extremely challenging requirements were impossible to achieve technologically at the time the experiment was proposed and the following six "near zeros" were identified as milestones toward achieving the necessary readiness.

Near Zero 1: Gyroscope Rotor Inhomogeneity. The difference in the geometric location of the rotor's mass center from its physical center can provide a lever arm that when acted on by the Earth's gravitational field will result in a torque. This placed a requirement on the manufacture of the GP-B rotors of having a "mass unbalance" of less than 300 nanometers. This requirement was operationally checked prior to gyroscope selection for flight. Once in orbit gyroscope performance has shown to be even better than estimated (see Table 1).

Table 1

Gyro #	1	2	3	4
Prelaunch estimate	18.8	14.5	16.8	13.5
On-orbit data	6.9	4.4	3.3	6.0

Near Zero 2: Drag-Free Control of the Spacecraft. Related to the first near zero requirement, the drag-free control is needed to reduce the force acting on the mass unbalance. The drag-free requirements for GP-B are broadly 1×10^{-9} m/sec² with a tighter requirement for 1×10^{-11} m/sec² in a narrow band centered at the roll frequency (13 mHz; 77.5 sec period) transverse to the direction of gyroscope spin.

Near Zero 3: Rotor Asphericity. Also a possible source of classical torques is the interaction between the gyroscope suspension system and the surface features of the gyroscope rotor. The gyroscope rotors are controlled to within one nanometer of the capacitive electrodes center using a capacitance bridge readout and electrostatic voltage, the gyroscope suspension system. The suspension voltages required are approximately 100 mV and can exert a torque on the rotor spin axis via imperfections in the rotor shape. This placed a manufacturing limit of 0.1 micrometers on the peak to valley difference which was successfully accomplished (see Figure 1).

Near Zero 4: Magnetic Field. In order to observe relativistic effects the gyroscope spin axis direction had to be monitored. It is not possible to mark the rotor for that purpose without violating the first and third near zero requirements. To measure the spin axis ori-

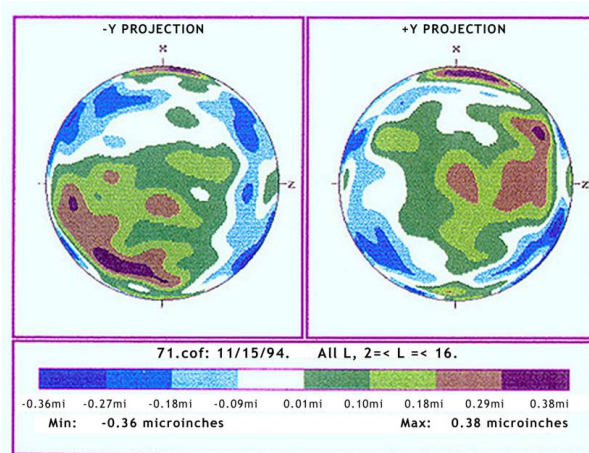


Figure 1 Gyroscope Surface Features

entation the London moment has been employed. The gyroscopes are coated with niobium, a superconducting metal. A spinning superconductor has a dipole magnetic moment (the London moment) perfectly aligned with its spin axis, exactly what was needed for the GP-B measurement. To implement this technique the magnetic field in the proximity of the rotor had to be smaller than the London moment. This was achieved using the Superconducting Lead Bag Technology developed for the program. The fields achieved were approximately 0.1 microgauss.

Near Zero 5: Ultra-Low Pressure. To ensure the proper use of the London moment as a reliable measurement the gyroscope spin speed had to be constant over the length of the mission. This was achieved by what is called the low temperature bake out and the associated use of a cryopump device. Once the gyroscopes were spun to their science spin speeds there was residual helium remaining in the vacuum can enclosing the science instrument. In the course of the mission this helium would eventually evaporate and act as a “spin-down” source by interacting with the gyroscope rotors. Table 2 shows that the bake out procedure dramatically improved the vacuum properties in the probe and exceeded the requirement with great margin.

Table 2 Gyroscope Spin-Down Rate On-Orbit (Years)

Gyroscope	Before bakeout	After bakeout
Gyro # 1	50	15,800
Gyro # 2	40	13,400
Gyro # 3	40	7,000
Gyro # 4	40	25,700

Near Zero 6: Rotor Charge. High energy particles that penetrate the spacecraft are a cause for rotor charging. It was necessary to maintain a charge level less than 15 mV throughout the mission. The technology developed to achieve this involves the use of UV light and gold plated bias electrodes placed in the gyroscope housing. Electrons were

liberated from the rotor/electrode system by emitting UV light and biasing the electrode according to the polarity the charge was adjusted to.

Each of the near zero requirements were achieved over several years of development and performed to specification, and in most cases beyond, during the mission.

Lessons Learned

In the final report² produced for the Gravity Probe B mission a set of lessons learned and best practices were recommended for future missions involving the use of drag-free technology. It specifically suggests the use of a mission simulation that can reflect coupled spacecraft and payload dynamics to mirror the in situ performance of the integrated satellite. This is particularly important for missions such as LISA, GAIA, STEP and others where new key technologies are involved to meet disturbance requirements that exceed those of what is, at this point in time, the state of the art in low disturbance performance, GP-B.

In the execution of the GP-B mission it was noted that there was an unprecedented level of cooperation between engineers and scientists in achieving the final performance level of the science instrument. In reflecting on the challenges met, particular focus of the discussions was on the IOC (initial on-orbit calibration) phase of the mission. The IOC schedule was planned for over 18 months prior to launch. More than six segments of the schedule covering several weeks of operations were practiced using an Integrated Test Facility (ITF), combining flight-like hardware with software spacecraft simulators, operated from the flight mission operations center located at Stanford University to provide actual telemetry to engineers to build familiarity and experience under “test it as you fly it” conditions. The IOC schedule was to last 45 days and had 15 days of contingency for a total of 60 days. When the spacecraft entered its science data taking phase, however, 128 days had past.

Investigating the reasons for the prolonged IOC phase the review committee concluded that in spite of the engineering expertise available, and the time devoted to anticipation of anomalous events and the near perfect performance of each of the spacecraft and payload subsystems, what was lacking was the ability to adequately reproduce the coupled payload and spacecraft dynamics in an integrated simulation to efficiently address the challenges. The mission had made extensive use of simulators, both software and hardware alike, however the shortcoming was in failing to integrate them dynamically. Evidence of the importance in having dynamically coupled simulation including flight-like hardware in the loop was demonstrated in the post-launch upgrade made to the gyroscope suspension system/gyroscope simulator.

There are two modes to establish a drag-free environment.³ In the unsuspended mode the proof mass is allowed to float freely and the satellite is commanded to follow the proof mass orbit. In the second (suspended) mode the proof mass is suspended and the drag-free system controls the spacecraft orbit such that suspension forces on the gyroscope are minimized. It was decided to operate the spacecraft vehicle in suspended drag-free mode after observing an unacceptable acceleration bias when using the baseline unsuspended mode that would have led to a slowly changing orbit of the spacecraft. This required modifications to the ATC parameters to meet the science requirements as those determined prior to launch for the suspended mode performed unsatisfactorily. It has proved to be most beneficial to

modify the GSS/gyroscope simulator to include an ATC simulation creating a drag-free simulator to optimize the drag-free control parameters. This decision sought to leverage the extensive knowledge of the GSS/gyroscope dynamics gained during ground testing and subsequently validated with on-orbit data. The ATC model was designed and validated also using orbit data and integrated in a matter of weeks. The resulting simulator was used to produce modifications that exceeded the science mission requirements. After two iterations the drag-free controller was optimized for the final science configuration. The process was a success but valuable time had been lost.

To summarize, it is clear that Gravity Probe B's traditional use of its simulators prior to launch to deliver individual systems that met their requirements, to verify and validate its software and to train its operations team was very successful. While this was believed to be a requirement for a successful mission what was unexpected was that this was not sufficient.

Quoting from the Gravity Probe B Post Flight Analysis Final Report:

“Invest in high fidelity simulations. Hardware-in-the-loop simulators are critical to validate the overall effectiveness of a complicated scientific instrument and satellite system. Where the instrument cannot be operated on the ground, suitable high fidelity simulations with flight compatible interfaces must be developed. These simulators, however, must be vetted against flight data and updated once actual performance data is known. This is required to be able to resolve operational anomalies.”

SIMULATOR DEVELOPMENT

With the GP-B mission completed in development and science phase and on-going data reduction activities a unique opportunity is given to comprehend the overall mission in a simulation environment. For the first time, flight data is available for an experimental setup where drag-free control has been applied in all degrees of freedom. Comparison with flight data is essential to establish a validated tool ready to aid future mission design where drag-free control is applied to provide an undisturbed environment for science measurements or else where close links between spacecraft and measurement instrument exist as is the case for GP-B.

The simulator development has undergone several stages. Based on [4] and [5] generic modules for spacecraft and experimental test-mass dynamics have been completed. Models for environmental disturbance calculation are developed and updated based on available data. Each of the generic modules is described in detail through a documentation package including technical notes on underlying physics and a separate or attached user manual on how to apply the module. Mission-specific control modules have been developed alongside the overall mission design to test and verify control algorithms. The generic modules are mainly coded in Fortran and C/C++ and can be integrated into a Matlab/Simulink environment via S-function blocks. The mission-specific control modules are primarily developed in Matlab/Simulink. In the following, module and simulator development is briefly outlined. A more detailed description can be found e.g. in [5, 1].

Generic Simulator Modules

Currently, the dynamics core can handle a nine-body system in anticipation of application to STEP which up to date includes the largest number of experimental masses in a scientific satellite mission. This number however can be increased if required. The satellite and test mass dynamics are calculated w.r.t. the most natural or common frames, i.e. the satellite states are either represented in the Earth centered inertial (ECI) or in the body-fixed frame with its origin in the satellite center of mass (COM). For the test masses, computations are carried out w.r.t. the corresponding test mass body-fixed frames centered in their respective COMs, or corresponding housing frames. As driving force for the dynamics, special emphasis is given to the derivation of a most accurate Earth gravity model. Based on GRACE (Gravity Recovery and Climate Experiment) data, spherical harmonics are provided to 360th degree and order. On the availability of new releases (see [6, 7]) the most up-to-date GRACE model is implemented. Gravitational influences due to the Sun, Moon and planets can be enabled. External disturbances due to atmospheric drag, solar radiation and magnetic fields are accounted for through parametric models and look-up tables generated in pre-processing utilizing finite element discretization of structural models. Element or volume forces are computed using standard model data^{8,9,10} and enhancements^{11,12} to account for characteristics not included elsewhere.

If the test masses are shielded from external disturbances by drag-free control the only other major disturbance source comes from satellite-gyroscope interaction through the measurement instrument. Simple spring-damper models including DC offsets are provided by the generic modules to approximate coupling interaction between the satellite and experiment. These can be replaced by external typically more advanced non-linear coupling algorithms or by calculated controller output. In case of GP-B the electrode force output from the gyroscope suspension system model is fed into the coupling link between spacecraft and gyroscope.

Mission-Specific Modules

The GSS model is part of the mission specific controls simulator developed concurrently with GP-B. It uses the difference in gyroscope and its housing reference position provided by the dynamics module to calculate suspension efforts. Control effort from the GSS is passed to the ATC to keep the spacecraft centered around the drag-free test mass. Attitude control uses sensor measurements from rate gyros, telescope and star tracker. The simulator model for ATC control uses the actual flight logic applied during the science phase of the mission. ATC force and torque commands are relayed to the actuator. The actuator model is comprised of 16 thrusters modeled as point forces to provide specific impulses. Converted into body forces and torques they are passed on to the dynamics model to update the satellite and gyroscope states.

For integration of the GP-B control modules with the generic simulator interfaces had to be established containing transformations from generic to mission-specific reference frames.

1-Gyro Simulator

The first mission simulator (shown in Figure 2(a)) has been assembled to match the original software version of the controls simulator for GP-B: a two-body system consisting of spacecraft and drag-free gyro. The controls simulator used Hill's equations to obtain satellite and gyroscope positions and velocities. For a cross-check the generic dynamics have been simplified to be comparable to the Hill's simulator and the idealized orbit conditions used to initialize the Hill's simulator have been adopted. The cross-check carried out in [5] shows the successful integration of the control modules into the overall simulator and especially confirms that the necessary transformations have been implemented correctly. At this point the simulator comprises a dynamically enhanced version of the engineering simulator developed along with GP-B. This engineering simulator can be executed with one or two gyros, one being a hardware gyro. The current modular structure of the generic simulator applied to GP-B also allows for module replacement with hardware in the loop.

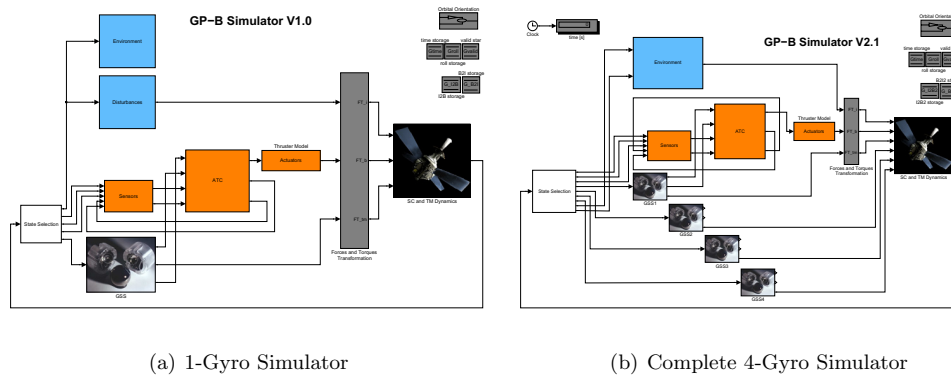


Figure 2 Simulator Development.

4-Gyro Simulator

In the next step the simulator has been enhanced to account for all four gyros as shown in Figure 2(b). The disturbance module has been combined with the environmental module since it uses the environmental outputs to calculate external forces and torques on the satellite. In Figure 2(b) the first gyro is set as drag-free gyro feeding an ATC trigger pulse and the GSS control effort to the ATC. Every GSS module is set up such that every gyro can function as drag-free reference mass. In fact, during the mission the drag-free gyro has switched between gyro 1 and 3 a couple of times for several reasons.

With the 4-Gyro Simulator for the first time a complete dynamics and control simulator has been established for GP-B. For the following comparison to flight data the major part of the environmental influences has been disabled to keep the comparison simple. Simulator version 2.0, a preliminary version without environmental disturbances except gravitational, is used to verify dynamics with science data. This approach is justified as long as a comparison to nominal flight data i.e. where no disruptions due to external influences have

been observed, is carried out. In this case, dynamics due to external disturbances other than gravitational are small enough to be neglected for the moment as we aim at dynamics module verification and not at mission reproduction.

SIMULATOR VERIFICATION

The major control modules have already been vetted with flight data available, see e.g. [1]. This part of the paper therefore focuses on validation of the dynamics core. In the following, two-orbit simulations are carried out with gyro 3 as drag-free proof mass and the results are compared to flight data spanning the same time period. Purely theoretical consideration leads to the conclusion that a gravity-gradient signal must appear eight times in the gyroscope data within a two-orbit time slot. It is also expected that gyro 1 shows the most pronounced gravity-gradient signal since it is farthest away from the drag-free gyro 3. This can be clearly seen in the topmost plots of Figures 3 and 4 where the body x-axis of the position vector for gyro 1 is displayed.

As a by-product and minor model improvement, verification with flight data revealed that adjustment was necessary for the modelled rate gyro noise. Although based on flight data, the estimation has been too high, masking the dynamical features. This can be seen in Figure 3 where especially for gyros 2 and 3 the imposed noise led to an increase in gyroscope position magnitude by a factor of four on average of the actual flight data. Decreasing the noise level by an order of magnitude leads to the more favorable comparison shown in Figure 4.

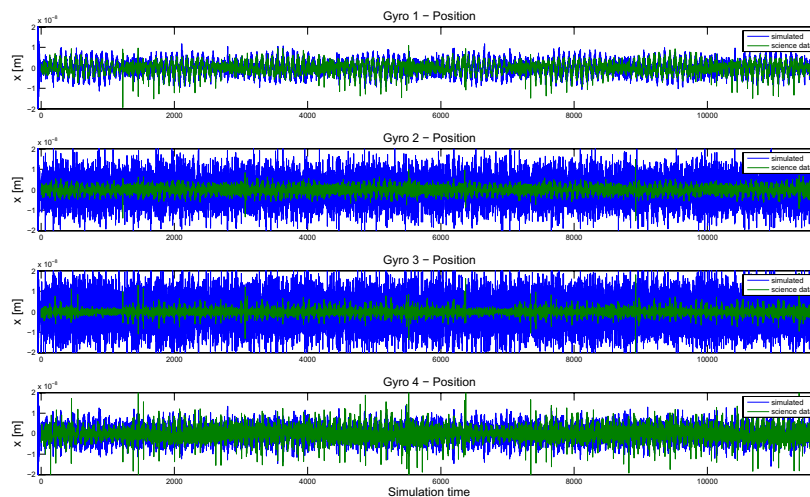


Figure 3 Gyroscope Position

The comparisons in Figures 3 and 4 have been carried out with the idealized input conditions the Hill's simulator has been initialized with, e.g. a perfectly polar orbit starting above the North pole. Two orbit periods of nominal flight data have been extracted from the

science data base starting at arbitrary vehicle time. Since the idealized simulation always starts at the top of the orbit the simulated and flight data do not match in phase. To line up with flight data, the simulated states have been shifted such that they can be compared directly to the flight data.

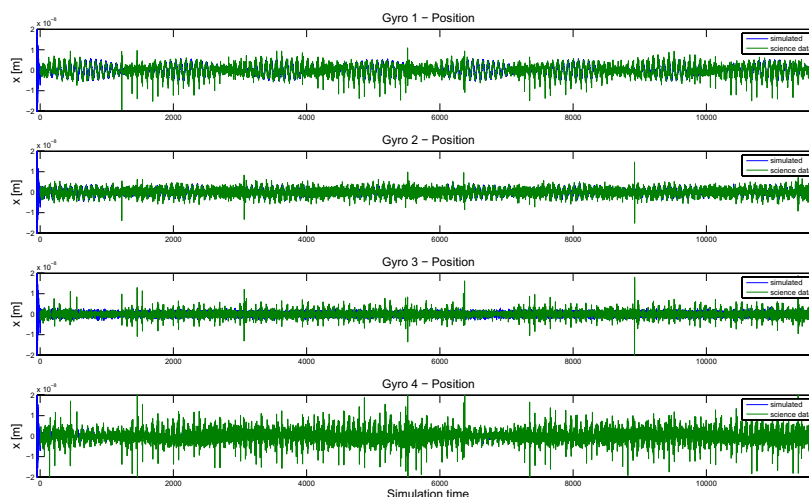


Figure 4 Gyroscope Position

Figure 4 shows, that even for the idealized simulation case good agreement exists between simulated and measured gyroscope data. This does not hold for the spacecraft dynamics since the idealized conditions result in a different orbit for the simulated versus measured spacecraft trajectory (see Figure 5(a)). For future investigations beyond simulator verification, the simulator has been adapted to run with arbitrary input conditions. Initializing the simulation with flight data leads to the results shown in Figures 5(b) and 7. Thereby, the most straight-forward comparison between simulated and flight data is possible. Furthermore, with the simulator adapted to use actual flight data as initial conditions the option to investigate anomalous flight conditions reflected in the data is now enabled.

What can be observed in Figure 5(b) is that towards the end of the two-orbit period there is a slight misalignment between the simulation results and the measured data. The same misalignment trend is visible in the blown-up section for the gyroscope position shown in Figure 7. The simulation has been run incorporating a simplified spherical Earth model and without external disturbances. This configuration has been carried over from simulator integration and cross-check between the generic and the GP-B controls simulator.

The discrepancies between the two data sets are mainly attributed to the neglect of higher-order terms in the gravitational field of the Earth. Figures 6(a) and 6(b) display a detail blown-up part of the spacecraft position comparison between simulation and flight results towards the end of a two-orbit period. In Figure 6(a) the detail simulation from Figure 5(b) is shown and in Figure 6(b) the simulation is repeated including higher order spherical terms in the Earth gravitational field. For the latter case a more favorable match

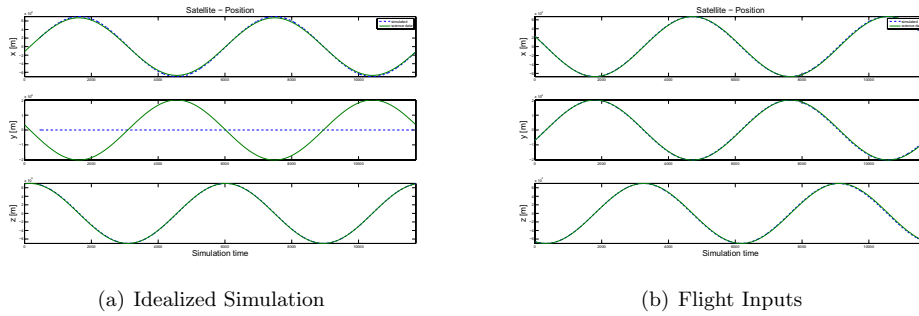


Figure 5 Spacecraft Position.

between simulated and flight data also towards the end of two orbits is achieved. For the resolution shown, already the first higher order harmonic modelling the oblate Earth accounts for the previous difference.

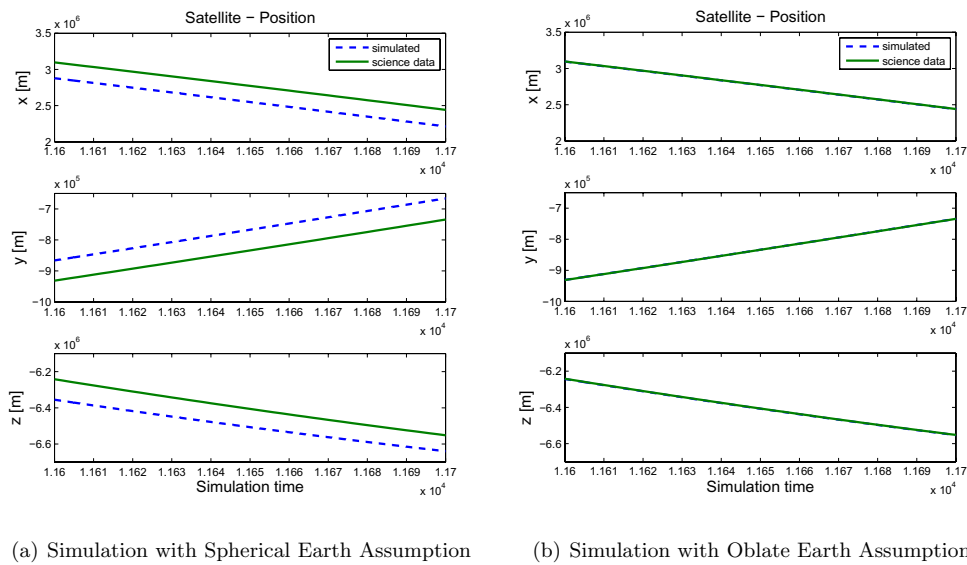


Figure 6 Comparison of Simulated Spacecraft Position and Flight Data.

For the gyroscope position, Figure 7 furthermore shows that different noise levels are apparent in the gyroscope data. The flight data is generally noisier than the simulated data, also there is a higher noise level in gyro 4 measurements compared to the other gyros. Both of these findings suggest further investigation on appropriate dynamic or noise modelling.

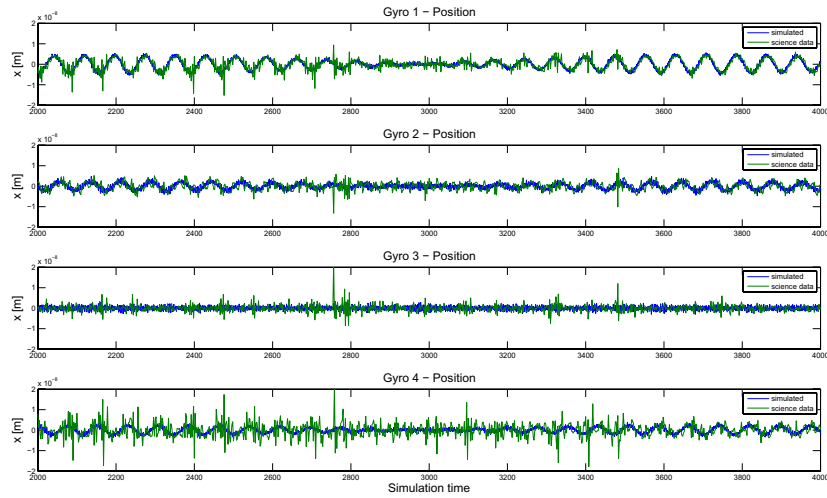


Figure 7 Gyroscope Position, Flight Inputs

SUMMARY AND OUTLOOK

A high-fidelity dynamics and control simulator has been adapted to the Gravity Probe B mission for simulator verification and model improvement. While validation for simplified dynamics, e.g. comparison to Hill's or Mathieu solutions have been carried out in the past, this is the first time that the full dynamics have been verified in their main features. The GP-B simulator also serves as reference for missions where high requirements on measurement accuracies and disturbance reduction exist.

In order to achieve mission goals commonly new key technologies have to be developed which have to be tested and verified in advance. The generic simulator combined with mission specific control tools can be applied for that purpose as well as prediction of mission scenarios. It furthermore aims to aid in on-orbit anomaly resolution and post-mission data analysis. Out of these four targets the first two have been accomplished for GP-B to a certain level, i.e. test and verification of key control technologies plus post-simulation of the nominal science phase which shows the simulator's predictive ability for undisturbed orbits. One outlook for the near future is anomaly reconstruction and investigation with the full simulator including environment and dynamics adapted to observed anomalous conditions, e.g. presence of higher environmental disturbances than usual or spacecraft module failure. Application to future science missions is anticipated in an on-going collaborative effort with the prime candidate STEP.

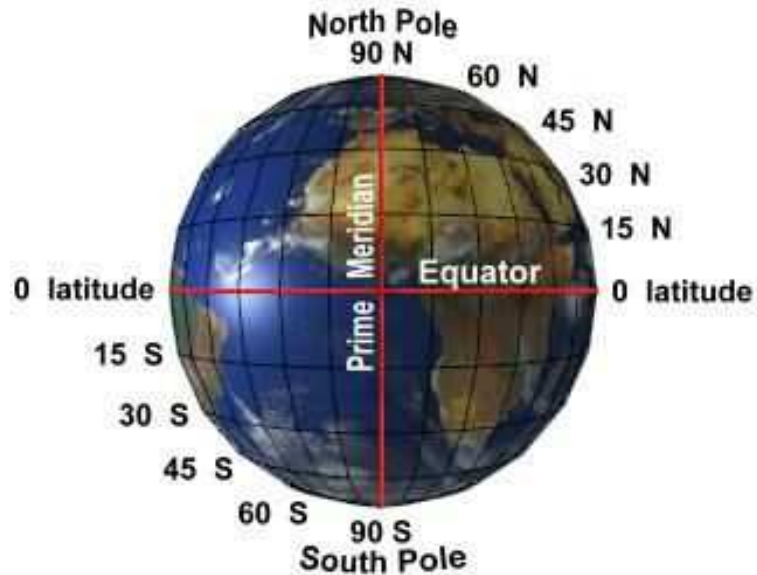
ACKNOWLEDGMENTS

This work is supported by the Marie-Curie program of the European Union under contract number MKTD-CT-2004-014188.

REFERENCES

- [1] S. Smoot, I. Pelivan, Y. Ohshima, and S. Theil, "High Fidelity Controls Simulation for Gravity Probe B," *30th Annual AAS Guidance and Control Conference, Breckenridge, Colorado*, February 2007.
- [2] "The Gravity Probe B Experiment. Post-Flight Analysis - Final Report," tech. rep., Stanford University, March 2007.
- [3] W. J. Bencze, D. B. DeBraP, L. Herman, T. Holmes, M. Adams, G. M. Keiser, and C. W. F. Everitt, "On-orbit Performance of the Gravity Probe B Drag-Free Translation Control System," *29th Annual AAS Guidance and Control Conference, Breckenridge, Colorado*, February 2006.
- [4] S. Scheithauer and S. Theil, "Generic Drag-Free Simulator," *AIAA Modeling and Simulation Conference, Monterey, California*, August 2002.
- [5] I. Pelivan, S. Smoot, S. Theil, and Y. Ohshima, "Modeling of Dynamics and Control for Gravity Probe B," *AAS/AIAA Space Flight Mechanics Meeting, Sedona, Arizona*, February 2007.
- [6] Physical Oceanography Distributed Active Archive Center at JPL, *PO.DAAC GRACE Home*. <http://podaac.jpl.nasa.gov/grace>.
- [7] Information System and Data Center at GFZ Potsdam. <http://isdc.gfz-potsdam.de/grace>.
- [8] National Aeronautics and Space Administration, *NRLMSISE-00 Model 2001*. <http://nssdc.gsfc.nasa.gov/space/model/atmos/nrlmsise00.html>.
- [9] *International Geomagnetic Reference Field (IGRF)*. <http://nssdc.gsfc.nasa.gov/space/model/models/igrf.html>.
- [10] *Total Ozone Mapping Spectrometer (TOMS)*. <http://toms.gsfc.nasa.gov/>.
- [11] M. Zijlstra, S. Theil, and S. Scheithauer, "Model for Short-term Atmospheric Density Variations," *2nd CHAMP Science Meeting, Potsdam, Germany*, September 2003.
- [12] P. Appel, S. Theil, S. Winkler, and A. Schleicher, "Attitude Estimation from Magnetometer and Earth-albedo-corrected Coarse Sun Sensor Measurements," *5th International Conference on Spacecraft Guidance, Navigation and Control Systems; Frascati, Italy*, October 2002.

Transformation Library



Project:	Document No.:
First Look	FLK-SIM-TN-ZAR-005
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm	Phone: +49-421-218-4803
D 28359 Bremen	Fax: +49-421-218-4356
Germany	E-Mail: grotjan@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-005

Issue: 1.3

Written: Stefanie Grotjan Date: 14th March 2008

Approved: - Date: -



Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	02/06/06	all	Initial Version by Stefanie Grotjan	✓
1.1	07/06/06	appendix added	Corrected version	✓
1.2	24/08/06	chapter 1 and 6 added, chapter 5 and 7 changed	Extended version	✓
1.3	14/03/08	update of chapter 2, 7 and appendix	Changed by Stefanie Bremer	✓



Contents

1	Introduction	4
2	Time transformations	4
2.1	Modified Julian Date from Calendar Date	4
2.2	Calendar Date and DOY from Modified Julian Date	5
2.3	Decimal Date from Calendar Date	6
3	Inertial and Earth Fixed Reference Frame	7
3.1	Earth Centered Inertial Frame ECI	7
3.2	Earth Centered Earth Fixed Frame ECEF	8
4	Transformation between ECI and ECEF	8
4.1	Precession	9
4.2	Nutation	11
4.3	Earth rotation	13
4.4	Polar motion	14
5	Spherical Coordinates	14
5.1	Geocentric coordinates	15
5.2	Geodetic coordinates	15
6	Transformation between ECEF and NED	16
7	The Simulink Library Transformation_lib.mdl	17
7.1	ECI to ECEF transformation	17
7.2	Transformation from Cartesian to geodetic coordinates	20
A	Appendix	21



1 Introduction

This document describes the transformation library consisting of several models to compute coordinate and time transformations. The source code for these models is mainly taken from the book *Satellite Orbits, Models, Methods, Applications*, written by Oliver Montenbruck and Eberhard Gill [4]. In addition to the C-code summarised in the *lib-transformation.a*, a Simulink Library *Transformation.Lib.mdl* containing all necessary blocks to build the transformation models can be used (cf. section 7). Besides, this Simulink Library provides some useful Matlab functions, that are not part of the *lib-transformation.a*.

2 Time transformations

Usually, time is measured in years, months, days, hours, minutes and seconds. For astronomy and many computer applications a continuous time scale that contains all informations about the actual date and time is more convenient. The *Julian Date* is such a time scale. It was introduced by Joseph Justus Scalinger in the 16th century. The Julian Date (*JD*) is measured in days including the actual time as fraction of day. The zero point of the Julian Date time scale was arbitrary defined at noon January 1, 4713 BC. As the number of days since 4713 BC is quite large and a start of counting the days at midnight is more practical, the *Modified Julian Date MJD* was introduced:

$$MJD = JD - 2400000.5 \quad (1)$$

The transformation between *MJD* and calendar date is presented below. A different method of continuous time measurement is the decimal date. The decimal date provides the actual date and time in terms of years and is described in section 2.3.

2.1 Modified Julian Date from Calendar Date

The algorithm presented in this section is taken from [4]. The Modified Julian Date is calculated from the date expressed in year (*Y*), month (*M*) and day (*D*). The day as well as the Modified Julian Date may include the fraction of day. The algorithm contains some expression in square brackets [*x*]. This is defined as integer which is smaller or equal to *x*.

For the computation of the *MJD* the year starts at March 1 and runs to the end of February. Thus the consideration of leap years is simplified. The values of *Y* and *M* are replaced as follows:

$$y = \begin{cases} Y - 1 & \text{if } M \leq 2 \\ Y & \text{otherwise} \end{cases} \quad (2)$$

and

$$m = \begin{cases} M + 12 & \text{if } M \leq 2 \\ M & \text{otherwise} \end{cases} \quad (3)$$

The number of leap days is calculated depending on the calendar system:

$$B = \begin{cases} -2 + [(Y + 4716)/4] - 1179 & \text{until 4 Oct. 1582, Julian calendar} \\ [Y/400] - [Y/100] + [Y/4] & \text{from 10 Oct. 1582, Gregorian calendar} \end{cases} \quad (4)$$

The Modified Julian Date can now be derived from the following equation:

$$MJD = 365y - 679004 + B + [30.6001(m + 1)] + D \quad (5)$$

2.2 Calendar Date and DOY from Modified Julian Date

To calculate the calendar date from Modified Julian Date, several steps are necessary. At first, the Julian Date at noon is computed:

$$a = [MJD] + 2400001 \quad (6)$$

The fraction of day can be derived from the Modified Julian Date and its integer part:

$$q = MJD - [MJD] \quad (7)$$

The following auxiliary quantities are calculated afterwards:

$$b = \begin{cases} 0 & \text{if } a < 2299161(\text{Julian calendar}) \\ [(a - 1867216.25)/36524.25] & \text{otherwise (Gregorian calendar)} \end{cases} \quad (8)$$

$$c = \begin{cases} a + 1524 & \text{if } a < 2299161(\text{Julian calendar}) \\ a + b - [b/4] + 1525 & \text{otherwise (Gregorian calendar)} \end{cases} \quad (9)$$

$$d = [(c - 121.1)/365.25] \quad (10)$$

$$e = [365.25d] \quad (11)$$

$$f = [(c - e)/30.6001] \quad (12)$$

The day D including the fraction of day is given by

$$D = c - e - [30.6001f] + q \quad (13)$$

The month M is obtained as follows:

$$M = f - 1 - 12[f/14] \quad (14)$$

Finally, the year Y is determined:

$$Y = d - 4715 - [(7 + M)/10] \quad (15)$$

Now the number that indicates the Day of Year DOY can be computed according to [3]:

$$DOY = \left[\frac{275 \cdot M}{9} \right] - k \cdot \left[\frac{M + 9}{12} \right] + D - 30, \quad k = \begin{cases} 1 & \text{if } Y \text{ is a leap year} \\ 2 & \text{otherwise} \end{cases} \quad (16)$$

2.3 Decimal Date from Calendar Date

The algorithm for the calculation of the decimal date in terms of years is based on the subroutine *julday* which is part of the program source code *geomag60.c* distributed by [1].

The computation of the decimal date from year Y , month M and day D , which is including the time as fraction of day, is done in several steps. First, it must be determined whether the year is a leap year or not. This yields the auxiliary quantity y_{leap} :

$$y_{leap} = \begin{cases} 0 & \begin{array}{l} \text{= NO leap year} \\ \text{if } Y \text{ is not divisible by 4 or} \\ \text{ } Y \text{ is a century year, that is NOT divisible by 400} \end{array} \\ 1 & \begin{array}{l} \text{= leap year} \\ \text{if } Y \text{ is divisible by 4 or} \\ \text{ } Y \text{ is a century year, that is divisible by 400} \end{array} \end{cases} \quad (17)$$

Table 1 contains the number of past days at the beginning of each month. The table is not valid for leap years, because the 29th February is not included. Therefore a second auxiliary quantity d_{leap} is needed:

$$d_{leap} = \begin{cases} 1 & \text{if } M > 2 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Now, the number of past days for the actual date can be calculated:

$$n_d = N_d(M) + D - 1 + d_{leap} \quad (19)$$

Depending on y_{leap} , the decimal date $date_{dec}$ in terms of years is given by:

Month M	Number of days N_d
January	1
February	32
March	60
April	91
May	121
June	152
July	182
August	213
September	244
October	274
November	305
December	335

Table 1: Number of past days at the beginning of each month

$$date_{dec} = \begin{cases} Y + n_d/365 & \text{if } y_{leap} = 0 \\ Y + n_d/366 & \text{if } y_{leap} = 1 \end{cases} \quad (20)$$

3 Inertial and Earth Fixed Reference Frame

In order to describe the motion of a satellite around the Earth, it is necessary to define reference frames. Usually, two different reference systems are used: an Earth Centered Inertial frame (ECI) and an Earth Centered Earth Fixed frame (ECEF). There are many different historically grown concepts for these two frame types. The transformation library uses the International Earth Rotation Service (IERS) definition. A more detailed description of the coordinates frames is given below.

3.1 Earth Centered Inertial Frame ECI

The definition of the Earth Centered Inertial Frame is based on the International Celestial Reference Frame (ICRF). The ICRF was introduced by the International Astronomical Union (IAU) in 1991 and is in use since 1998. It is maintained by the IERS. The system is free of rotation. The axes are fixed with respect to distant extragalactic radio objects. They closely agree with the Earth Mean Equator and Equinox of the year 2000 (EME2000). The ECI axes are aligned with the ICRF axes. In contrast to the ICRF, whose origin is located at the barycenter of the solar system, the ECI frame's origin is located at the Earth's center of mass. Hence, the x -axis of the ECI points towards the vernal equinox of J2000 and lies inside the equatorial plane. The z -axis is parallel to the Earth's angular momentum vector, which is perpendicular to the

equatorial plane. The y -axis lies in the equatorial plane and completes the right hand orthogonal system (cf. figure 1).

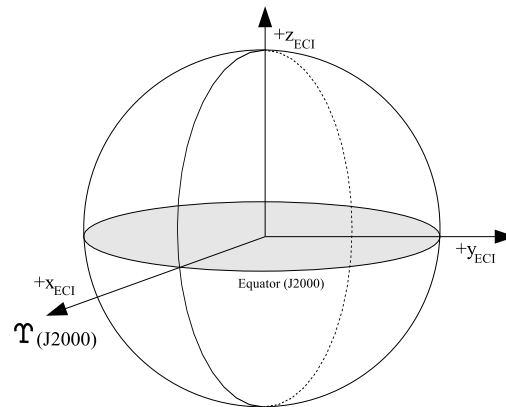


Figure 1: Earth Centered Inertial Frame

The latest information about the ICRF can be found on the IERS homepage:
<http://www.iers.org/>.

3.2 Earth Centered Earth Fixed Frame ECEF

The Earth Centered Earth Fixed frame, which is used in the transformation library, is equal to the International Terrestrial Reference Frame (ITRF). This system is also maintained by the IERS. Its origin is located at the Earth's center of mass. The frame is fixed to the Earth's surface, which means that it exhibits no net rotation with respect to the Earth's crust. The x -axis lies in the equatorial plane and points towards the IERS Reference Meridian (IRM), which is also called the Greenwich meridian. The z -axis points towards the IERS Reference Pole (IRP). The right hand orthogonal system is completed by the y -axis, which lies in the equatorial plane (cf. figure 2).

More informations about the ITRF can be found on the ITRF homepage:
<http://itrf.ensg.ign.fr/>.

4 Transformation between ECI and ECEF

The definition of the ECI and ECEF frames leads to a complex relationship between both systems, which contains the following aspects:

- Earth precession $\underline{\underline{P}}$
- Earth nutation $\underline{\underline{N}}$
- Earth rotation $\underline{\underline{\Theta}}$

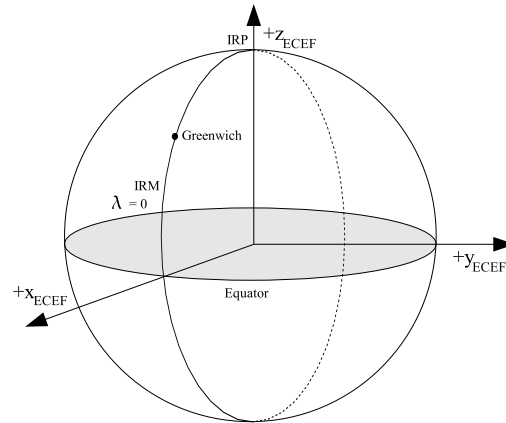


Figure 2: Earth Centered Earth Fixed Frame

- polar motion $\underline{\underline{\Pi}}$

The resulting transformation may be expressed as

$$\underline{r}_{ECEF} = \underline{\underline{\Pi}}(t)\underline{\underline{\Theta}}(t)\underline{\underline{N}}(t)\underline{\underline{P}}(t)\underline{r}_{ECI} \quad (21)$$

where \underline{r}_{ECI} is the position vector in inertial coordinates and \underline{r}_{ECEF} is the position vector in Earth-fixed coordinates. The underlying models of these transformation matrices are explained in the next sections.

4.1 Precession

The Earth may be considered as rotationally symmetric gyroscope. Due to external torques \underline{D} , which result from gravitational forces of the sun and the moon, the angular momentum vector \underline{l} moves around the North pole of the ecliptic (cf. figure 3). This secular movement of the Earth's axis is called precession. One complete revolution takes almost 26000 years.

Apart from the influence of the sun and the moon which causes the change of the orientation of the Earth's axis and the equatorial plane, the influence of the planets yields a precession of the ecliptic. This results in a decreasing obliquity of the ecliptic. Figure 4 illustrates the combined effects of lunisolar and planetary precession.

The model for precession in the transformation library includes the mentioned effects and is based on the IAU 1976 precession theory. According to this theory the orientation of the mean equator and equinox of epoch T ("mean-of-date", mod) with respect to the equator and equinox of J2000 is defined by the following three angles:

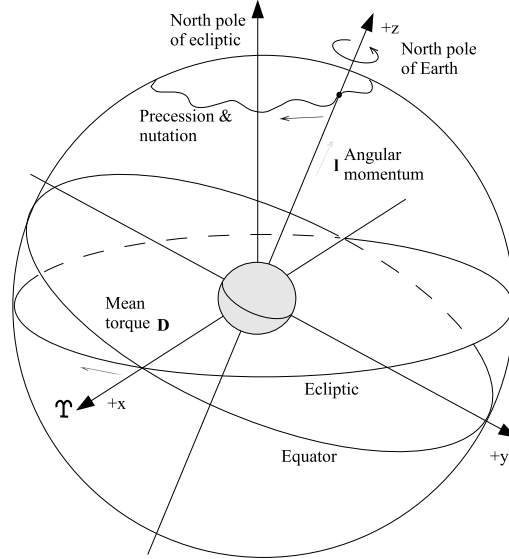


Figure 3: Motion of the Earth's axis under the influence of solar and lunar torques [4]

$$\zeta = 2306.''2181T + 0.''30188T^2 + 0.''017998T^3 \quad (22)$$

$$\vartheta = 2004.''3109T - 0.''42665T^2 - 0.''041833T^3 \quad (23)$$

$$z = \zeta + 0.''79280T^2 + 0.''000205T^3 \quad (24)$$

with

$$\begin{aligned} T &= (MJD - MJD_{J2000})/36525 \\ &= (MJD - 51544.5)/36525 \end{aligned} \quad (25)$$

The Modified Julian Date MJD is measured in Terrestrial Time TT . Terrestrial Time is a uniform time scale and would be measured by an ideal clock on the surface of the geoid (cf. [4]). Its unit is one SI second (1 day = 86400 SI seconds).

The transformation matrix from the ECI to the "mean-of-date" coordinates may be derived from the introduced angles, where $\underline{\underline{P}}$ is the product of three consecutive rotations:

$$\begin{aligned} \underline{\underline{P}} &= \underline{\underline{R}}_z(-z)\underline{\underline{R}}_y(\vartheta)\underline{\underline{R}}_z(-\zeta) \\ &= \begin{pmatrix} \cos(-z) & \sin(-z) & 0 \\ -\sin(-z) & \cos(-z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\vartheta) & 0 & -\sin(\vartheta) \\ 0 & 1 & 0 \\ \sin(\vartheta) & 0 & \cos(\vartheta) \end{pmatrix} \begin{pmatrix} \cos(-\zeta) & \sin(-\zeta) & 0 \\ -\sin(-\zeta) & \cos(-\zeta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (26)$$

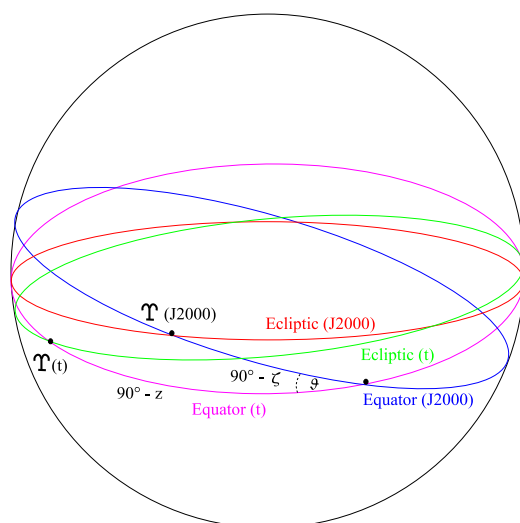


Figure 4: The effects of precession on the ecliptic, equator, and vernal equinox [4]

The transformation may now be written as:

$$\underline{r}_{mod} = \underline{P}(t)r_{ECI} \quad (27)$$

4.2 Nutation

The precession theory considers an averaged lunisolar torque. In fact, the torques of sun and moon vary monthly and annual, which causes a movement of the Earth's axis, that superposes the precessional motion. This small periodic perturbation is called nutation. Nutation is especially caused by the moon. During the 18.6-year nodal period of the moon the orientation of the lunar orbit with respect to the Earth's equator changes continuously. This causes a periodic shift of the vernal equinox $\Delta\Psi$ and a change of the obliquity $\Delta\varepsilon$ ($\varepsilon' = \varepsilon + \Delta\varepsilon$, cf. figure 5). In consequence of this motion the true celestial pole describes a small ellipse around the mean celestial pole. The superposition of both the precessional and the nutational motion results in a wavelike movement (cf. figure 3).

The nutation model in the transformation library is based on the IAU 1980 nutation theory. The nutation angles $\Delta\Psi$ and $\Delta\varepsilon$ arise from the sum of 106 terms,

$$\Delta\Psi = \sum_{i=1}^{106} (\Delta\Psi)_i \cdot \sin(\phi_i) \quad (28)$$

$$\Delta\varepsilon = \sum_{i=1}^{106} (\Delta\varepsilon)_i \cdot \cos(\phi_i) \quad (29)$$

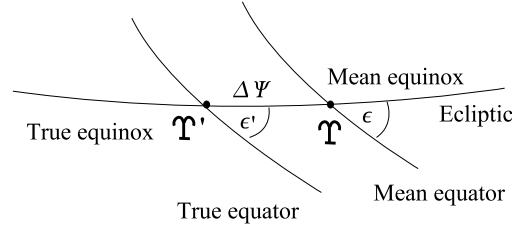


Figure 5: The shift in the position of the ecliptic, equator, and vernal equinox caused by nutation [4]

where ϕ_i is a periodic function of the moon's mean anomaly (l), the sun's mean anomaly (l'), the mean distance of the moon from the ascending node (F), the difference between the mean longitudes of the sun and the moon (D), and the mean longitude of the ascending node of the lunar orbit (Ω).

$$\phi_i = p_{l,i}l + p_{l',i}l' + p_{F,i}F + p_{D,i}D + p_{\Omega,i}\Omega \quad (30)$$

$$l = 134^\circ 57' 46.''733 + 477198^\circ 52' 02.''633T + 31.''310T^2 + 0.''064T^3 \quad (31)$$

$$l' = 357^\circ 31' 39.''804 + 35999^\circ 03' 01.''224T - 0.''577T^2 - 0.''012T^3 \quad (32)$$

$$F = 93^\circ 16' 18.''877 + 483202^\circ 01' 03.''137T - 13.''257T^2 + 0.''011T^3 \quad (33)$$

$$D = 297^\circ 51' 01.''307 + 445267^\circ 06' 41.''328T - 6.''891T^2 + 0.''019T^3 \quad (34)$$

$$\Omega = 125^\circ 02' 40.''280 - 1934^\circ 08' 10.''539T + 7.''455T^2 + 0.''008T^3 \quad (35)$$

The coefficients $p_{l,i}$, $p_{l',i}$, $p_{F,i}$, $p_{D,i}$, and $p_{\Omega,i}$ and the expressions for $\Delta\Psi_i$ and $\Delta\varepsilon_i$ are summarised in the IAU 1980 nutation theory table ([4], p.179). As in the precession theory, the Modified Julian Date is measured in Terrestrial Time and T is derived as introduced in equation 25.

The nutation angles and the mean obliquity of the ecliptic ε

$$\varepsilon = 23.^\circ 43929111 - 46.''8150T - 0.''00059T^2 + 0.''001813T^3 \quad (36)$$

lead to the transformation matrix $\underline{\underline{N}}$:

$$\begin{aligned} \underline{\underline{N}} &= \underline{\underline{R}}_x(-\varepsilon - \Delta\varepsilon)\underline{\underline{R}}_z(-\Delta\Psi)\underline{\underline{R}}_x(\varepsilon) \\ &= \underline{\underline{R}}_x(-\varepsilon')\underline{\underline{R}}_z(-\Delta\Psi)\underline{\underline{R}}_x(\varepsilon) \end{aligned} \quad (37)$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\varepsilon') & \sin(-\varepsilon') \\ 0 & -\sin(-\varepsilon') & \cos(-\varepsilon') \end{pmatrix} \begin{pmatrix} \cos(-\Delta\Psi) & \sin(-\Delta\Psi) & 0 \\ -\sin(-\Delta\Psi) & \cos(-\Delta\Psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varepsilon) & \sin(\varepsilon) \\ 0 & -\sin(\varepsilon) & \cos(\varepsilon) \end{pmatrix}$$

The "mean-of-date" coordinates may now be transformed to "true-of-date" (tod) coordinates:

$$\underline{r}_{tod} = \underline{N}(t)\underline{r}_{mod} \quad (38)$$

4.3 Earth rotation

The introduced precession and nutation theory yield the transformation from ICRF to "true-of-date" coordinates. The Earth's axis, the z -axis of the "true-of-date" coordinate system, points now towards a new reference point, the so-called Celestial Ephemeris Pole (CEP). The rotation about the Earth's axis is described by the Greenwich Hour Angle or the Greenwich Mean Sidereal Time ($GMST$). It is the angle between the Greenwich meridian and the mean vernal equinox of date. The $GMST$ may be derived from the following equation:

$$GMST = 24110.54841 + 8640184.812866T_0 + 1.002737909350795UT1 + 0.093104T^2 - 0.0000062T^3 \quad (39)$$

with

$$T_0 = (MJD(0^hUT1) - MJD_{J2000})/36525 \quad (40)$$

$$T = (MJD(UT1) - MJD_{J2000})/36525 \quad (41)$$

The time used for $GMST$ computation is Universal Time $UT1$. This time scale represents a mean solar day of 24 hours. Because the Earth's spin period is not constant, the length of one second of Universal Time varies depending on the actual mean length of the solar day. The Earth rotation cannot be predicted accurately, so the difference between Universal Time and Terrestrial Time is determined retrospectively (cf. [4]).

As the $GMST$ measures the angle between the Greenwich meridian and the mean vernal equinox, the Greenwich Apparent Sidereal Time ($GAST$) is referred to the true equinox of date (\rightarrow nutation). Both times are coupled by the equation of the equinoxes:

$$GAST - GMST = \Delta\Psi \cos(\varepsilon) \quad (42)$$

The transformation matrix $\underline{\Theta}$ from "true-of-date" coordinates to "equator-Greenwich-meridian" (egm) coordinates may be derived from $GAST$:

$$\begin{aligned} \underline{\Theta} &= \underline{R}_z(GAST) \\ &= \begin{pmatrix} \cos(GAST) & \sin(GAST) & 0 \\ -\sin(GAST) & \cos(GAST) & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (43)$$

The transformation may now be written as:

$$\underline{r}_{egm} = \underline{\underline{\Theta}}(t)\underline{r}_{tod} \quad (44)$$

4.4 Polar motion

In the previous section, the Celestial Ephemeris Pole (CEP) was introduced. It is the common z -axis of both the "true-of-date" and the "equator-Greenwich-meridian" coordinate system. The CEP is not fixed with respect to the Earth's surface and therefore differs from the IERS Reference Pole (IRP), the z -axis of the ITRF/ECEF. The Earth is considered as rotationally symmetric gyroscope again. In absence of external torques, the axis of rotation moves around the axis of figure. Regarding the Earth, this movement is called polar motion. It results from interaction of free precession with a period of 435 days and an annual movement due to changes in the Earth's mass distribution (water and air flows).

The difference between CEP and IRP is given by the coordinates x_p rotationally denoting the CEP offset with respect to the IRP. Polar motion cannot be predicted and therefore must be determined by observations. The values of x_p and y_p are published regularly in Bulletin B of the IERS (cf. figure 8). The pole coordinates yield the transformation matrix $\underline{\underline{P}}$:

$$\begin{aligned} \underline{\underline{P}} &= \underline{\underline{R}}_y(-x_p)\underline{\underline{R}}_x(-y_p) \\ &= \begin{pmatrix} \cos(-x_p) & 0 & -\sin(-x_p) \\ 0 & 1 & 0 \\ \sin(-x_p) & 0 & \cos(-x_p) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-y_p) & \sin(-y_p) \\ 0 & -\sin(-y_p) & \cos(-y_p) \end{pmatrix} \end{aligned} \quad (45)$$

The "equator-Greenwich-meridian" coordinates may now be transformed to ECEF coordinates:

$$\underline{r}_{ECEF} = \underline{\underline{P}}(t)\underline{r}_{egm} \quad (46)$$

5 Spherical Coordinates

For several application, spherical coordinates are necessary. The difference between geodetic and geocentric coordinates is illustrated in figure 6. The longitude λ , the angle between the Greenwich meridian and the meridian through the considered point, is identical in both systems, but the geodetic latitude φ and the geodetic altitude h differ from the geocentric coordinates due to the Earth's flattening. Similar to the ECI and ECEF systems, different reference ellipsoids for the Earth are in use. The computation of geodetic coordinates from Cartesian coordinates is done in the same way for each ellipsoid and is presented in section 5.2.

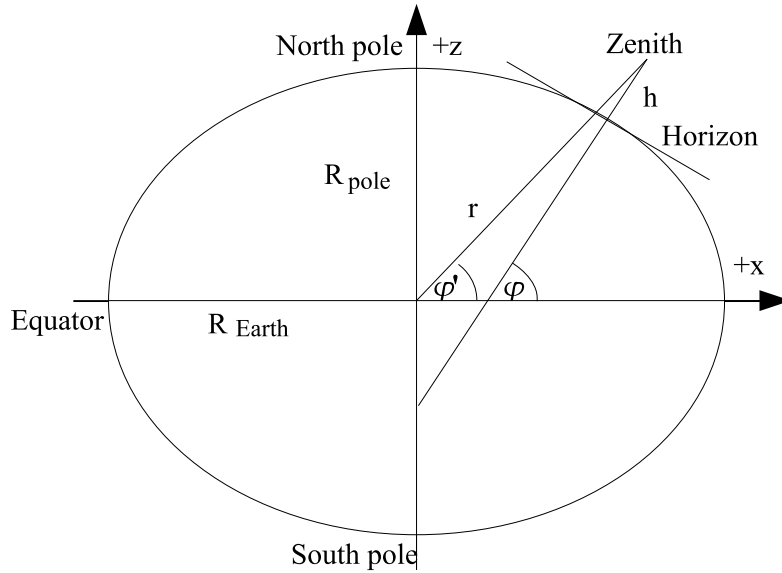


Figure 6: Geocentric and geodetic latitude [4]

5.1 Geocentric coordinates

For the computation of geocentric coordinates, the Earth is considered as sphere. The longitude λ and the geocentric latitude φ' are calculated from Cartesian coordinates as follows:

$$\lambda = \arctan\left(\frac{y}{x}\right) \quad (47)$$

$$\varphi' = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \quad (48)$$

In conjunction with geocentric coordinates, the radius r is often used instead of the altitude h :

$$r = \sqrt{x^2 + y^2 + z^2} \quad (49)$$

5.2 Geodetic coordinates

The transformation library uses an iterative method to compute the geodetic coordinates for given Cartesian coordinates. The values for the latitude and the altitude can be calculated with the following expression:

$$\sin \varphi = \frac{z + \Delta z}{\sqrt{x^2 + y^2 + (z + \Delta z)^2}} \quad (50)$$

$$N = \frac{R_{\oplus}}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (51)$$

$$\Delta z = Ne^2 \sin \varphi \quad (52)$$

where e is the eccentricity of the reference ellipsoid.

$$e = \sqrt{1 - 1(1 - f)^2} \quad (53)$$

The flattening f of the Earth is defined as:

$$f = \frac{R_{\oplus} - R_{pole}}{R_{\oplus}} \quad (54)$$

The initial value of Δz is set to $e^2 z$ as first approximation. When the iteration converges, the geodetic coordinates can be computed as follows:

$$\lambda = \arctan\left(\frac{y}{x}\right) \quad (55)$$

$$\varphi = \arctan\left(\frac{z + \Delta z}{\sqrt{x^2 + y^2}}\right) \quad (56)$$

$$h = \sqrt{x^2 + y^2 + (z + \Delta z)^2} - N \quad (57)$$

6 Transformation between ECEF and NED

In certain cases it is helpful to use so-called local tangent plane reference frames. These frames are right hand orthogonal systems, but not related to the Earth's center. One example is the North-East-Down coordinate system. Its origin is located at the origin of the regarded force. The *North*-axis (x_{NED}) points towards the North Pole of the Earth. The *Down*-axis (z_{NED}) points downwards and is aligned with the local plumb line on the Earth ellipsoid. The *East*-axis (y_{NED}) completes the right hand orthogonal system and points towards ascending East longitude.

The transformation from ECEF to NED coordinates can be obtained by a rotation about the z -axis with λ and afterwards a rotation about the new y -axis with $-(\varphi + \pi/2)$ [2]. This yields the following transformation matrix:

$$\underline{\underline{A}}_{ECEF}^{NED} = \begin{pmatrix} -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \varphi \cos \lambda & -\cos \varphi \sin \lambda & -\sin \varphi \end{pmatrix} \quad (58)$$

Since $\underline{\underline{A}}_{ECEF}^{NED}$ is a rotation matrix, the inverse transformation matrix $\underline{\underline{A}}_{ECEF}^{NED}^{-1}$ (to calculate ECEF coordinates from NED coordinates) is equal to the transpose:

$$\underline{\underline{A}}_{ECEF}^{NED}^{-1} = \underline{\underline{A}}_{ECEF}^{NED T} \quad (59)$$

$$\implies \underline{\underline{A}}_{NED}^{ECEF} = \underline{\underline{A}}_{ECEF}^{NED T} \quad (60)$$

7 The Simulink Library Transformation_lib.mdl

The Simulink library *Transformation_lib.mdl* contains all blocks that are necessary to make the time transformations mentioned in section 2 and to compute the transformation matrix $\underline{\underline{A}}_{ECI}^{ECEF} = \underline{\underline{\Pi}}(t)\underline{\underline{\Theta}}(t)\underline{\underline{N}}(t)\underline{\underline{P}}(t)$. Besides, blocks are provided to transform Earth-fixed Cartesian to geodetic or geocentric coordinates as well as Earth-fixed coordinates to NED coordinates and vice versa (cf. figure 7). The next sections give an overview about the *ECI2ECEF* transformation and the *ECEF2GEOD* blocks, because these two need some detailed description.

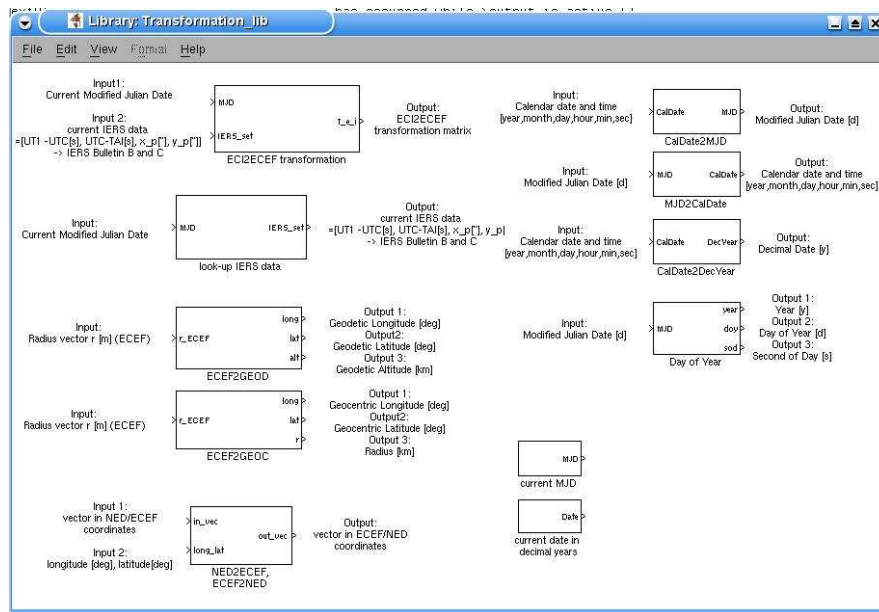


Figure 7: Simulink Library Transformation_lib.mdl

7.1 ECI to ECEF transformation

According to the introduced transformation theories, the computation of the transformation matrix $\underline{\underline{A}}_{ECI}^{ECEF}$ needs several time-dependent inputs:



- current MJD(*TT*)
- current MJD(*UT1*)
- y_p
- x_p

But the Modified Julian Date in the simulation is measured in Coordinated Universal Time *UTC*. This time scale is based on atomic time scales with the unit of one SI second and is in use as clock time for everyday purposes. As the precession and nutation theories need Terrestrial Time and the computation of *GAST* needs Universal Time *UT1*, the *UTC* must be converted. The time difference between *UT1* and *UTC* is published in the IERS Bulletin B, as well as the polar coordinates x_p and y_p for polar motion computation (cf. figure 8).

BULLETIN B 135
03 May 1999

Contents described in the Explanatory Supplement, mailed with Bulletin B133

1 - EARTH ORIENTATION PARAMETERS (IERS evaluation).
The values in this section are samplings of section 2 given at five-day intervals.

...
...

2 - SMOOTHED VALUES OF X, Y, UT1, D, DPSI, DEPSILON (IERS EVALUATION)
at one-day intervals. For smoothing characteristics, see Table2 in the explanatory supplement. The reference system is described in the 1997 IERS Annual Report.

1999	MJD	x	y	UT1-UTC	UT1-UT1R	D	dPsi	dEpsilon
(0 h UTC)		"	"	s	ms	ms	0.001"	0.001"
MAR 1	51238	.06996	.24187	.652011	-.154	.960	-45.4	-5.7
MAR 2	51239	.06888	.24160	.651078	-.187	.944	-45.4	-5.7
MAR 3	51240	.06806	.24156	.650161	-.277	.930	-45.7	-5.7
MAR 4	51241	.06740	.24173	.649232	-.391	.990	-46.0	-5.7
MAR 5	51242	.06657	.24214	.648193	-.490	1.053	-46.3	-5.6
MAR 6	51243	.06546	.24245	.647136	-.541	.975	-46.5	-5.6
MAR 7	51244	.06436	.24260	.646250	-.516	.856	-46.6	-5.6
MAR 8	51245	.06328	.24262	.645427	-.400	.800	-46.6	-5.7
MAR 9	51246	.06204	.24243	.644649	-.191	.737	-46.4	-5.7
MAR 10	51247	.06089	.24205	.643946	.098	.684	-46.2	-5.8
MAR 11	51248	.05984	.24162	.643269	.439	.693	-46.0	-5.8

...

Figure 8: Bulletin B abstract

The difference between *TT* and *UTC* must be derived indirectly from the *UTC* and International Atomic Time (*TAI*) difference. Terrestrial Time and International Atomic Time are two differently grown concepts of atomic time with a constant offset (cf. equation 61).



$$TT = TAI + 32.184s \quad (61)$$

The difference between those time scales and Universal Time increases 0.5 to 1.0 seconds per year. In contrast Coordinated Universal Time never deviates more than 0.9 seconds from Universal Time, which is achieved by the introduction of leap seconds. The resulting difference between *UTC* and *TAI* is published in the IERS Bulletin C (cf. figure 9). A detailed description of all mentioned time scales can be found in [4], chapter 5.1.

INTERNATIONAL EARTH ROTATION SERVICE (IERS)
SERVICE INTERNATIONAL DE LA ROTATION TERRESTRE

BUREAU CENTRAL DE L'IERS
OBSERVATOIRE DE PARIS
61, Av. de l'Observatoire 75014 PARIS (France)
Tel. : 33 (0) 1 40 51 22 26
FAX : 33 (0) 1 40 51 22 91
Internet : iers@obspm.fr

Paris, 17 July 1998

Bulletin C 16

To authorities responsible for
the measurement and distribution
of time

UTC TIME STEP
on the 1st of January 1999

A positive leap second will be introduced at the end of December 1998.
The sequence of dates of the UTC second markers will be:

1998 December 31,	23h 59m 59s
1998 December 31,	23h 59m 60s
1999 January 1,	0h 0m 0s

The difference between UTC and the International Atomic Time TAI is:

from 1997 July 1, 0h UTC, to 1999 January 1, 0h UTC	: UTC-TAI = - 31s
from 1999 January 1, 0h UTC, until further notice	: UTC-TAI = - 32s

Leap seconds can be introduced in UTC at the end of the months of December or June, depending on the evolution of UT1-TAI. Bulletin C mailed every six months, either to announce a time step in UTC, or to confirm that there will be no time step at the next possible date.

Daniel GAMBIS
Director
Central Bureau of IERS

Figure 9: Bulletin C

Hence, the input for the transformation matrix $\underline{\underline{A}}_{ECI}^{ECEf}$ consists of:

- current MJD(UTC)
- IERS set:

- $UT1 - UTC$
- $UTC - TAI$
- y_p
- x_p

The *Transformation_lib.mdl* contains a block to compute the current Modified Julian Date from the *MJD* at simulation start. The IERS data (IERS set) for the regarded simulation time interval must be taken from the Bulletins B and C (download via ftp: <ftp://hpiers.obspm.fr/iers/bul/>) and stored in a *.txt file (e.g. *IERS_data.txt*) as illustrated in figure 10. Then, the block "look-up IERS data" is able to provide time-dependent IERS data.

%MJD	x_p	y_p	UT1-UTC	UTC-TAI
51238	0.06996	0.24187	0.652011	-32
51239	0.06888	0.2416	0.651078	-32
51240	0.06806	0.24156	0.650161	-32
51241	0.0674	0.24173	0.649232	-32
51242	0.06657	0.24214	0.648193	-32
51243	0.06546	0.24245	0.647136	-32
51244	0.06436	0.2426	0.64625	-32
51245	0.06328	0.24262	0.645427	-32
51246	0.06204	0.24243	0.644649	-32
51247	0.06089	0.24205	0.643946	-32
51248	0.05984	0.24162	0.643269	-32

Figure 10: IERS data format for simulation

For test purposes three different transformation methods are implemented in the transformation library. The user can choose one of the following options:

- 0: only Earth rotation
- 1: Earth rotation, IAU 1976 precession and IAU 1980 nutation theory
- 2: Earth rotation, IAU 1976 precession and IAU 1980 nutation theory, polar motion

7.2 Transformation from Cartesian to geodetic coordinates

The model for the computation of geodetic coordinates needs only the Earth-fixed position vector in Cartesian coordinates as time-dependent input. Furthermore, a reference ellipsoid, i.e. the reference Earth radius R_{\oplus} and flattening f , must be chosen depending on the application. Two examples are listed in table 2.

Datum	R_{\oplus}	f
ITRF (GRS-80)	6378137 m	1/298.257222101
WGS84	6378137 m	1/298.257223563

Table 2: Earth reference ellipsoids [4]

References

- [1] *International Geomagnetic Reference Field (IGRF)*.
- [2] Gustavo Baldo Carvalho. Reference Frame Definitions. Technical Report INT-GEN-DF-ZAR-001, ZARM, University of Bremen, 2005.
- [3] J. Meeus. *Astronomical Formulae for Calculators*. Willmann-Bell, Inc, Richmond, Virginia, 4th edition, 1988.
- [4] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits - Models, Methods, Applications*. Springer-Verlag, 2000.

A Appendix

Table 5 contains all functions, that are implemented in the *libtransformation.a*. They are in parts taken from the source code provided by [4] and supplemented by functions taken from [2]. Some of these functions are available as *s-functions* in the *Transformation_lib.mdl* (cf. table 3). If required, the additional functions provided by the *libtransformation.a* may be used to build new *s-functions*. Finally, table 4 contains the Matlab *m-functions* that are also part of the *Transformation_lib.mdl*.

Function	Source file	Short Description
DayOfYear_sf	DayOfYear_sf.c	Day of Year and Second of Day from <i>MJD</i>
ECI2ECEF_matrix_sf	ECI2ECEF_matrix_sf.c	Computes $\underline{A}_{ECI}^{ECEF}$
NEDandECEF_sf	NEDandECEF_sf.c	Transformation between NED and ECEF
geocentric_sf	geocentric_sf.c	Computes geocentric from ECEF coordinates
geodetic_sf	geodetic_sf.c	Computes geodetic from ECEF coordinates

Table 3: *s-functions* for the *Transformation_lib.mdl*



Function	Source file	Short Description
Cal2MJD	Cal2MJD.m	Computes MJD from calendar date
IERS_lookup	IERS_lookup.m	Look-up function for IERS data
Date2DecYear	Date2DecYear.m	Computes decimal date from calendar date
MJD2Cal	MJD2Cal.m	Computes calendar date from MJD

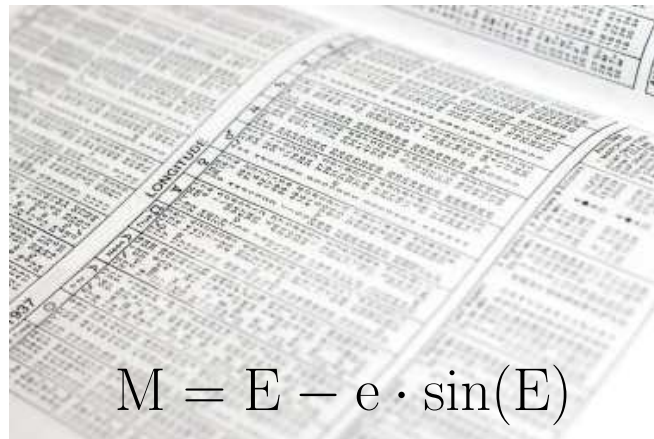
Table 4: Matlab *m-functions* for the *Transformation_lib.mdl*

Function	Source file	Short Description
AzEl	SAT_RefSys1.cpp	Azimuth and elevation from local tangent coordinates
cart2sph	cart2sph.c	Computes spherical from Cartesian coordinates
CalDat	SAT_Time1.cpp	Calendar date and time from Modified Julian Date
DayOfYear	DayOfYear.c	DOY and SOD of Day from Modified Julian Date
ecef2ned	NEDandECEF.c	Computes NED from ECEF coordinates
eci2ecef_matrix	ECI2ECEF_matrix.c ECI2ECEF_matrix_f90.c	Computes $\underline{\underline{A}}_{ECEF}^{ECEF}$ Computes $\underline{\underline{A}}_{ECI}^{ECEF}$, Fortran wrapper
EclMatrix	SAT_RefSys1.cpp	Transformation of equatorial to ecliptical coordinates
EqnEquinox	SAT_RefSys1.cpp	Computation of the equation of the equinoxes
GAST	SAT_RefSys1.cpp	Computes Greenwich Apparent Sidereal Time
Geodetic	SAT_RefSys1.cpp	Computes geodetic from Cartesian coordinates
GHAMatrix	SAT_RefSys1.cpp	Computation of Earth rotation matrix $\underline{\underline{\Theta}}$
GMST	SAT_RefSys1.cpp	Computes Greenwich Mean Sidereal Time
LTCMatrix	SAT_RefSys1.cpp	Greenw. meridian system to local tangent coordinates
MeanObliquity	SAT_RefSys1.cpp	Computes the mean obliquity of the ecliptic
Mjd	SAT_Time1.cpp	Modified Julian Date from calendar date and time
ned2ecef	NEDandECEF.c	Computes ECEF from NED coordinates
NutAngles	SAT_RefSys1.cpp	Computation of nutation angles
NutMatrix	SAT_RefSys1.cpp	Nutation matrix $\underline{\underline{N}}$
PrecMatrix	SAT_RefSys1.cpp	Precession matrix $\underline{\underline{P}}$
PoleMatrix	SAT_RefSys1.cpp	Computation of polar motion matrix $\underline{\underline{\Pi}}$

Table 5: Functions included in the *libtransformation.a*



Ephemeris Calculation in the Drag-Free Simulator



Project:	Document No.:
FIRST LOOK	FLK-SIM-TN-ZAR-008
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm D 28359 Bremen Germany	Phone: +49-421-218-8939 Fax: +49-421-218-4356 E-Mail: rkeil@zarm.uni-bremen.de pelivan@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-008


Issue: 1.0

Written: Ralf Keil, Ivanka Pelivan

Date: 18th March 2008

Approved: -

Date: -

	Ephemeris Calculation in the Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-008 Issue: 1.1 Page: 2 of 14
---	--	--

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	10/04/07	all	Initial Version by Ralf Keil, Ivanka Pelivan	√
1.1	18/03/08	Chapter 3	Updated Version by Stefanie Bremer	√

Contents

1	Introduction	5
2	Theoretical considerations	5
2.1	The algorithm of Montenbruck & Pfleger	5
2.2	The JPL ephemeris	8
3	Implementation of the Ephemeris Algorithms	9
3.1	The algorithm of Montenbruck & Pfleger	10
3.2	The JPL ephemeris	13



Acronyms and Abbreviations

Symbol	Description
<i>JPL</i>	Jet Propulsion Laboratory
<i>PPN</i>	Parameterized Post-Newtonian formalism
<i>DE/LE</i>	Development Ephemeris / Lunar Ephemeris
<i>IERS</i>	International Earth Rotation and Reference Systems Service
<i>ICRF</i>	International Celestial Reference Frame
<i>TDB</i>	Barycentric Dynamical Time
<i>SSB</i>	Solar System Barycenter
<i>M</i>	Mean anomaly
<i>M'</i>	Mean anomaly of the Sun
<i>l, b, r</i>	helioc. coordinates: longitude l, latitude b, radius r
<i>AU</i>	Astronomical Unit
...	...

1 Introduction

Modern ephemeris algorithms are able to calculate the position and velocity of objects in the solar system to an unprecedented degree of precision. Their results are indispensable for observations of celestial bodies like planets, moons, etc. as well as for the navigation of spacecrafts. The validity of this data is usually confined to several centuries – beginning in a reasonable point of time in the past to a later point of time often in the future.

2 Theoretical considerations

Long-term (secular) perturbations of orbits due to gravitational interactions with objects of unknown mass cannot be accounted for adequately in the ephemeris calculation and therefore prevent the algorithms to compute highly accurate results for a very long time span. Against this background, the algorithms can be divided into two classes according to the prescription of the way the ephemeris are being computed and the secular factors are being considered. One class uses analytical formulae and the programming code contains numerous perturbation terms (see the Montenbruck & Pfleger algorithm in subsection 2.1) while the other class models the orbits of the celestial bodies by applying numerical integration methods to the equations of motions (e.g. the JPL ephemeris, subsection 2.2).

2.1 The algorithm of Montenbruck & Pfleger

Since Kepler and Newton various analytical models have been conceived. Beginning with the unperturbed Keplerian motion of the planets, series of corrections have been added to these “planetary theories” in order to take into account the mutual gravitational interaction with other planets and further bodies in the solar system. Many authors pushed the elaboration and refinements of the theories to a high extent, publishing analytical formulae to understand the observations of the most prominent celestial bodies. Nowadays, these studies are treated as the first important step towards a complete computation of an orbit, since they guarantee merely the minimum of time and effort to spend, for instance in terms of computer performance. The assumptions made are usually well described and include often the orbit’s eccentricity and its inclination (see below). In comparison with pure numerical integration processes the analytical approach with series expansions is appealing to everyone interested in understanding the ephemeris calculations from a close astrodynamical point of view, moreover, a considerable reduction in computation time is achieved. However, their derivation is far from being simple but laborious and lengthy.

Montenbruck & Pfleger revealed in their book[1] a rather straightforward approach to the planetary orbit calculations based on a (semi-)analytical description of the Kepler

problem. The low eccentricities and low inclinations of the orbits (except for Mercury and the dwarf planet Pluto) have two essential consequences for further examinations. Firstly, the mathematical treatment of non-perturbed Kepler orbits can be simplified when using the mean anomaly instead of the true quantity, the semi-major axis of the Earth's orbit instead of the true time-dependent expression, and introducing a small approximation to the orbit's plane that takes into account the deviation of a planet's plain from the ecliptic. Secondly, the mutual gravitational forces between the major planets are orders of magnitude smaller than the gravitational force by the Sun. Therefore, changes in planetary orbits can be described by averaged Keplerian ellipses, superimposed by small perturbations of certain periodicities.

According to previous works on analytical descriptions of planetary motions, Montenbruck & Pfleger express the Keplerian motion and the perturbation terms as recurrent corrections to the heliocentric ecliptic coordinates longitude l , latitude b , and radius r of an orbit. The elaboration of a series expansion for Kepler's equation leads to mathematical expressions for l, b , and r as functions of the mean anomaly M . Any perturbation of a body's motion, like the mutual gravitational attraction of the planets, can be considered as additional terms in the series expansion, being reflected in additional M -terms (one term for the perturbed object and another one for the perturbing object). Furthermore, perturbations due to other planets and bodies will also affect the mean orbital elements on long time scales. They need to be included as explicitly time-dependent terms in the series expansions, e.g. from the motion of the vernal equinox. The number of perturbation corrections can exceed several hundred terms per planet, if the accuracies of the ephemeris should reach the arcsec-level. As a consequence numerical methods have been implemented to store preliminary results from previous calculations and to simplify the evaluation of trigonometrical functions. Basically, they allow to reorganise the programming code to a sufficient compactness and provide a comprehensible processing of the disturbance data.

For a given time (TDB or JD), the complete primary algorithm returns the heliocentric ecliptic longitude l , latitude b , and radius r of the planet requested (or the geocentric position of the Sun), with respect to the ecliptic and vernal equinox of date. The code provides different types of coordinates, apparent and astrometric. In contrast to the geometric coordinates, usually used for heliocentric planetary positions (emphasizing the location of a body in space at a certain time), the apparent and astrometric coordinates are important for getting the information in which direction the body can be observed. The latter generally provide the geocentric ephemeris of an object, e.g. the position of the Sun. Apparent coordinates (relative to the true equinox of date) take into account corrections for the precession, nutation, aberration, and light-time, whereas the astrometric coordinates (equinox B1950 or J2000) include the precession and light-time alone. In the case of Pluto, the evaluation of the coordinates includes the high inclination (leading to numerous secular terms) and, hence, is done relative to the equinox of B1950

plus a subsequent transformation to the equinox of date. The underlying procedures are limited to the time period of 1890AD and 2100AD, since they are based on a Fourier analysis of ephemeris numerically determined instead of using the perturbation theory.

Concerning the period 1750AD – 2250AD the algorithm gives the error estimations summarized in Table 1. For times in the 19th and 20th centuries the level of accuracy is in the order of arcsec.

Table 1: Mean errors of the planetary ephemeris (1750 – 2250; Pluto: 1890 – 2100)

PLANET	Δl ["]	Δb ["]	Δr [10^{-6} AU]
Mercury	1.0 – 1.5	0.7 – 1.0	1.0 – 1.5
Venus	0.5 – 1.0	0.2 – 2.5	0.5 – 1.5
Sun – Earth	0.5 – 2.0	0.0 – 0.1	0.4 – 1.5
Mars	0.5 – 2.5	0.1 – 1.0	3 – 10
Jupiter	2 – 8	0.7 – 1.0	20 – 30
Saturn	2 – 11	0.8 – 1.5	40 – 100
Uranus	3 – 8	0.7 – 1.0	50 – 200
Neptune	3 – 40	1.0 – 2.0	500 – 2000
Pluto	1 – 9	0.2 – 2.2	200 – 1000

(l,b,r: helioc. longitude l, latitude b, and distance r)

The motion of the Earth’s Moon is one of the best studied astrodynamical phenomena. From the beginning of the 18th century, many efforts have been required to develop a general prescription for computing not only the Keplerian orbit on a monthly scale, but also determine the inequalities produced by the gravitational attraction of the Sun and the perturbations caused by the figure of the Earth as well as the attraction of the planets. The major inequalities have been known since Ptolemy and Brahe, and were named evection, variation, and annual variation. A quantitative estimation of the relative size of these inequalities can be given by the following expression for the true longitude of the Moon, λ :

$$\begin{aligned}
\lambda &= \text{mean longitude} + \text{major inequality} + \\
&\quad + \text{evection} + \text{variation} + \text{annual inequality} + \text{parallactic inequality} \\
&= \text{mean longitude} + [377.33' \sin(M) + 12.86' \sin(2M) + \dots] + \\
&\quad + 76.43' \sin(2D - M) + 39.5' \sin(2D) - 11.13' \sin(M') - [2.08' \sin(D) + \dots] + \dots
\end{aligned}$$


where M and M' are the mean anomalies of the Moon and the Sun (M being measured from the mean position of the perigee), and D is the difference between the mean longi-

tudes of Moon and Sun. The term with the argument $(2D - M)$ is called “evection”, and has a period of 31.823 days, whereas the term with argument $2D$ has received the name “variation” (period: 14.767 days). The terms with arguments $M, 2M, 3M, \dots$ are called “elliptic” (their sum gives the “equation of the centre”). The term with argument M is called the “annual variation”. Its period is one year and depends on the ellipticity of the orbit. The terms with arguments $D, 3D, \dots$ are called the “parallactic inequality”, with periods of 29.5306 days, 9.8435 days, etc, respectively. Analogous terms can be deduced in the expansions of the radius vector and the latitude of the Moon. In addition to these periodic inequalities, secular terms occur in two of the elements of the orbit – the longitude of the perigee and the longitude of the node. The perigee accomplishes a complete revolution in $3232^d 13^h 48^m 29.6^s$ or 8.8505 years, and the line of nodes moves in the opposite direction, performing a complete revolution in $6793^d 9^h 23^m 9.3^s$ (18.5997 years). According to his lunar theory Brown developed one of the best-known analytical models of lunar motion containing more than one thousand perturbation terms. He computed 155 periodic terms with coefficients greater than 0.1 arcsec in the expression for the lunar longitude, and more than 500 terms with smaller coefficients. To calculate the longitude of the Moon with an accuracy within 0.1 arcsec, it is necessary to add together 655 terms; for the latitude, about 300 terms are sufficient. Montenbruck & Pfleger use a fraction of them (the “Improved Lunar Ephemeris” of 1954) to obtain an arcsec-level of accuracy for the equatorial position of the Moon in the geocentric reference frame, relative to the equinox of date.

The results of the Montenbruck & Pfleger algorithm have been modified according to the needs within the program structure of the Drag-Free Simulator (e.g. the output of geocentric rectangular coordinates, see section 3.1).

2.2 The JPL ephemeris

The most accurate ephemeris calculations in our time come from the JPL in Pasadena, CA, USA. Their numerical integration models – the Development/Lunar Ephemeris DE4xx/LE4xx – are based on n-body-calculations of approx. 300 solar system objects including the PPN of the general theory of relativity. The accuracy within a 50-year time span is typically 5 mas. The DE200 was the first model based on the J2000 equinox and became the basis of the Astronomical Almanac since 1984 (see [3] and [4]). It includes data from 1600AD to 2169AD with nutation corrections but no librations. The latter were started to be integrated within its successor, the DE403 model[6]. This description includes times between 1950AD and 2049AD and became a standard for the IERS. Finally, the third enhanced model, DE405[5], was developed in 1997 and covers the time period between 1600 AD and 2200 AD. It is based upon the ICRF and is widely used as the state-of-the-art ephemeris source. The model DE406 is an extended version of the DE405 in terms of the time span, covering the long range from 3000BC to 3000AD, but has no corrections for nutations nor librations, thus the accuracy of the interpolating

	Ephemeris Calculation in the Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-008 Issue: 1.1 Page: 9 of 14
---	---	--

polynomials and subsequently that of the positions and velocities has been lessened.

Concerning the Moon, the models LE200[4], LE403[6], and their follow-up models LE405/LE406[5] have been developed. As an essential part of the integration calculations boundary conditions are incorporated into the procedures which are deduced from diverse independent sources, like astrometric and radar observations of the planets and their moons, together with lunar laser ranging data, as well as tracking data of interplanetary space probes and artificial probes in orbits around other planets. For DE406/LE406, the interpolating accuracy is no worse than 25 meters for any planet and no worse than one meter for the Moon.

The biggest uncertainties on planetary positions arise from perturbations of minor solar system bodies emerging as secular deviations from the well-known celestial mechanics theory. They have to be considered in a non-straightforward way, that is the JPL has to publish revised data sets on ephemeris at intervals of 20 years.

The JPL planetary and lunar ephemeris packages (data + subroutines) are available via Internet from the ftp server “[ssd.jpl.nasa.gov](ftp://ssd.jpl.nasa.gov)” in the subdirectory “[pub/eph/planets/](ftp://ssd.jpl.nasa.gov/pub/eph/planets/)”. The data consist of a large binary file containing chebyshev coefficients, forming the basis for subroutines to calculate the rectangular coordinates (x,y,z) and the velocities of the Sun, Moon, and the major planets with respect to any other, including the Sun, the SSB, and the Earth-Moon Barycenter at the desired general relativistic point of time, the barycentric dynamic time (TDB). It also provides the equation of equinoxes (of use in calculating nutation) and the librations of the Moon. The data in the binary file is a result of a simultaneous solution to the general relativistic equations of motion for all the planets and most major moons. It spans a period of five centuries ([7]).

Moreover, several other ephemeris data have been published, mostly in conjunction with various spacecraft missions (e.g. DE410: used for Mars Exploration Rover navigation, 2003).

3 Implementation of the Ephemeris Algorithms

The previously described algorithm of Montenbruck & Pflieger , and the JPL ephemeris alternatively replace the routine Posofplanets for planetary position calculation used up to now (September 2007) in the Drag-Free Simulator [2]. In the outdated routine hard-coded orbital elements of a certain year are transformed into cartesian coordinates. For missing time spans an interpolation is carried out. Moreover, only planetary positions for Sun, Mercury, Venus, Mars, Jupiter, Saturn and Moon are calculated. The new routines also provide the ephemeris of Uranus, Neptune and Pluto.

3.1 The algorithm of Montenbruck & Pfleger

From simulator version 2.1 up to 2.2 the routine of Montenbruck & Pfleger is called inside the dynamics core to calculate the gravitational influences of the planets. Since version 3.0 the routine is removed from the kernel and instead executed by the Matlab/Simulink s-function *ephemeris_sf*. It provides a vector containing the positions of the regarded celestial bodies which is supplied to the dynamics core by an input port. The corresponding simulink library is described below.

Modifications to the Original Code

Originally the planets have been in sequence:

[Sun, Mercury, Venus, **Earth**, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto]

They have been re-ordered since the Drag-Free Simulator's inertial frame is Earth centered:

[Sun, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, **Earth**]

The Earth ephemeris is not calculated. The algorithm of Montenbruck & Pfleger provides a separate routine for the Moon. Up to version 2.2 inside the dynamics core and from version 3.0 further on, the following sequence is used:

[Sun, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, **Moon**]

Further code modifications include:

- The vector $[R_x, R_y, R_z]$ has been renamed to $[R_{xe}, R_{ye}, R_{ze}]$ (e for ephemeris) since $[R_x, R_y, R_z]$ is already used in the gravity/transformation library.
- Note: NutMatrix from the ephemeris code and NutMatrixSimple from the transformation library are identical except for the input. For now, NutMatrix from the ephemeris code is renamed into NutMatrixE to not interfere with the more complex NutMatrix from the transformation library.
- There is a loop over planets based on user input in the gravityfield routine (up to 2.2) and in the *ephemeris_sf* (since 3.0) respectively which calls the routines of Montenbruck & Pfleger. The internal loop over all planets in the routine of Montenbruck & Pfleger is therefore removed.

Core Update

Which planet ephemeris is actually calculated is defined by user input. The Matlab data file `load_sim_data.m` contains a parameter array (see below) enabling which planets are to be included in the calculation. The old planetary sequence

[Sun, Mercury, Venus, Mars, Jupiter, Saturn, Moon]

is replaced by

[Sun, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, Moon],

i.e. the option array switching on/off (1/0) the effects of Sun, Moon and planets on the gravitational acceleration has been extended,

```
df_core_param.i_option_g_planets(1) = 0.0;    % Sun
df_core_param.i_option_g_planets(2) = 0.0;    % Mercury
df_core_param.i_option_g_planets(3) = 0.0;    % Venus
df_core_param.i_option_g_planets(4) = 0.0;    % Mars
df_core_param.i_option_g_planets(5) = 0.0;    % Jupiter
df_core_param.i_option_g_planets(6) = 0.0;    % Saturn
df_core_param.i_option_g_planets(7) = 0.0;    % Uranus
df_core_param.i_option_g_planets(8) = 0.0;    % Neptune
df_core_param.i_option_g_planets(9) = 0.0;    % Pluto
df_core_param.i_option_g_planets(10) = 0.0;   % Moon
```

Comparison of Algorithms

The test program `ephemeris_test` compares the planetary positions of Sun, Mercury, Venus, Mars, Jupiter, Saturn and Moon calculated by the algorithm of Montenbruck & Pfleger and the routine `Posofplanets` used previously for a Modified Julian Date of 52840. The test program output is the planetary position in geocentric rectangular coordinates in km, for further reference the coordinates are referred to as [x,y,z]; the fourth number is the distance $\sqrt{x^2 + y^2 + z^2}$. From the output of the test program shown in table 2, it can be seen that most numbers at least resemble each other except for the z-positions of the Moon (highlighted in blue).

Table 2: Comparison of Test Results

MJD 52840.0000000000

Results of subroutine `ephemeris_f90.cpp`:

Sun	-68670364.8485006	124436292.627204	53948160.2529671	152021096.976403
Mercury	-125895084.260804	124350828.845557	59836751.0606223	186797049.369370
Venus	-81884386.6375069	221510414.541443	98457692.7552339	255862919.916127
Mars	64905280.6429815	-19615667.3955709	-15733590.5901755	69606147.2542268
Jupiter	-735860056.079905	527959598.156366	243156773.346283	937740143.026590
Saturn	-152386218.574566	1369444061.92863	571789458.131074	1491824983.96858
Uranus	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
Neptune	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
Pluto	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
Moon	388965.766064924	89824.1734905758	11515.1404531131	399368.686520855

Comparison to old subroutine `posofplanets`:

Sun	-68666490.1866247	124439085.025462	53949817.3156089	152022220.563021
Mercury	-125816051.128508	124610638.251602	59968067.6410615	186958976.846917
Venus	-81912854.5496980	221508621.293131	98459586.5974048	255871208.307217
Mars	65464129.7759885	-19151478.7092674	-15535581.1541698	69954883.3597540
Jupiter	-736392826.207185	527262284.982664	242868915.088462	937691431.959785
Saturn	-152453899.217207	1369442075.69681	571792262.181848	1491831150.35394
Moon	390773.468792822	85294.8641196819	-25077.0565707196	400759.250078923

The Ephemeris Library

The Simulink library `ephemeris_lib.mdl` (cf. figure 1) contains the s-function block to execute the routine of Montenbruck & Pflieger as described above. It requires the Modified Julian Date as input. Besides the s-function parameter `ephemeris.i_option_planets` to switch on=1/off=0 the calculation of the specific body's position is needed. It can be initialized by the m-file `init_ephemeris.m`:

```

ephemeris.i_option_planets(1) = 0.0;      % Sun
ephemeris.i_option_planets(2) = 0.0;      % Mercury
ephemeris.i_option_planets(3) = 0.0;      % Venus
ephemeris.i_option_planets(4) = 0.0;      % Mars
ephemeris.i_option_planets(5) = 0.0;      % Jupiter
ephemeris.i_option_planets(6) = 0.0;      % Saturn
ephemeris.i_option_planets(7) = 0.0;      % Uranus
ephemeris.i_option_planets(8) = 0.0;      % Neptune
ephemeris.i_option_planets(9) = 0.0;      % Pluto
ephemeris.i_option_planets(10) = 0.0;     % Moon

```

For `ephemeris.i_option_planets(i) = 0`, the corresponding elements in the output vector are filled with zeros.

NOTE: If `df_core_param.i_option_g_planets(i)` is set to 1 the corresponding `ephemeris.i_option_planets(i)` must also be adapted!
Currently `df_core_param.i_option_g_planets` is set to `ephemeris.i_option_planets` in `load_sim_data.m` to avoid problems.

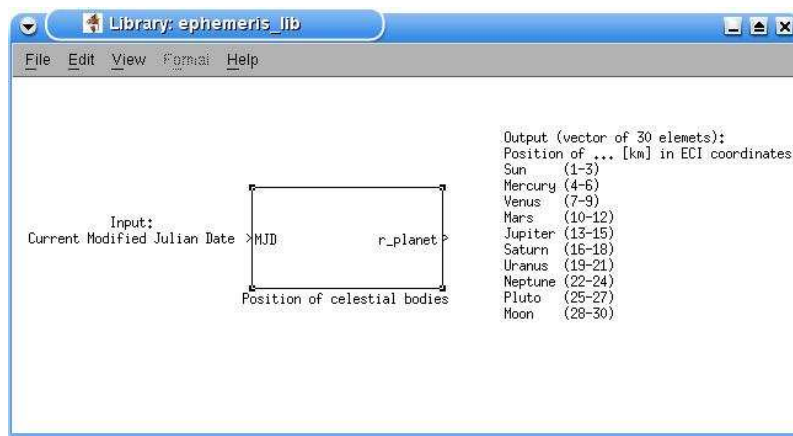



Figure 1: Simulink Library `ephemeris_lib.mdl`

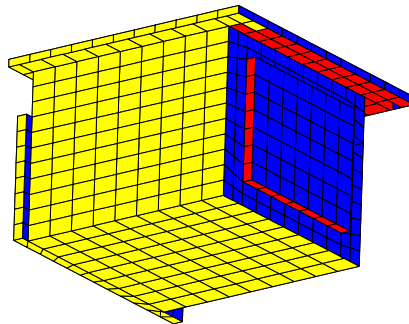
3.2 The JPL ephemeris

	Ephemeris Calculation in the Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-008 Issue: 1.1 Page: 14 of 14
---	---	---

References

- [1] O. Montenbruck, T. Pfleger, and S. Dunlop. *Astronomy on the personal computer – 4th completely revised ed.* Springer, (C++ programming language, corrected 2nd printing), 2002.
- [2] I. Pelivan. The Modular Drag-Free Simulator. Technical Report FLK-SIM-TN-ZAR-006, Center of Applied Space Technology and Microgravity (ZARM), February 2007.
- [3] E. M. Standish, Jr. Orientation of the JPL Ephemerides, DE 200/LE 200, to the dynamical equinox of J 2000. *A&A*, 114:297–302, October 1982.
- [4] E. M. Standish, Jr. The observational basis for JPL’s DE 200, the planetary ephemerides of the Astronomical Almanac. *A&A*, 233:252–271, July 1990.
- [5] E. M. Standish, Jr. JPL planetary and lunar ephemerides, DE405/LE405. *JPL Interoffice Memo.*, F-98-048, 312, 1998.
- [6] E. M. Standish, Jr., X. X. Newhall, J. G. Williams, and W. F. Folkner. JPL planetary and lunar ephemerides, DE403/LE403. *JPL Interoffice Memo.*, 314:10–127, 1995.
- [7] J. G. Williams and E. M. Standish. Dynamical reference frames in the planetary and Earth-Moon Systems. In J. Kovalevsky, I. I. Mueller, and B. Kolaczek, editors, *Reference Frames*, volume 154 of *Astrophysics and Space Science Library*, page 67, 1989.

Modeling of surface forces - Tutorial



Project:	Document No.:
First Look	FLK-SIM-TU-ZAR-001
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm	Phone: +49-421-218-4803
D 28359 Bremen	Fax: +49-421-218-4356
Germany	E-Mail: grotjan@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TU-ZAR-001


Issue: 1.1

Written: Stefanie Grotjan

Date: 4th April 2008

Approved:

Date:

	<p>Modeling of surface forces - Tutorial</p>	<p>Doc.No.: FLK-SIM-TU-ZAR-001 Issue: 1.1 Page: 2 of 18</p>
---	--	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	14/09/06	all	Initial version by Stefanie Grotjan, based on [2]	✓
1.1	04/04/08	chapter 3 and 4, Appendix	Updated version by Stefanie Bremer	✓



Contents

1	Introduction	4
2	Model preparation	4
2.1	Working with ANSYS	4
2.2	Building a model	5
2.2.1	Element choice	5
2.2.2	Material choice	6
2.2.3	Geometry definition	6
2.2.4	Meshing	8
2.2.5	Data export	9
3	Look-up table creation	10
3.1	Data files for the calculation of surface forces and torques	11
3.2	Calculation of the surface force and torque for a specific incident angle .	12
3.3	Creation of look-up tables	14
3.4	C Code	15
4	Simulation in Matlab/Simulink	15
A	Appendix	18

1 Introduction

This tutorial intends to describe the method for the determination of surface forces on a satellite. The whole process for the calculation of surface forces consists of three parts:

1. Model preparation
2. Look-up table creation
3. Simulation in Matlab/Simulink

Different programs are used to work on the specific parts. The tutorial is based on the master thesis *Entwicklung und Validierung einer Methode zur Berechnung der Oberflächenkräfte auf Satelliten* by Mehmet Cicek [3] and the corresponding tutorial [2].

2 Model preparation

At first the geometry of the satellite has to be converted into a finite element model. This is done with the finite element program ANSYS. The element data can be exported and becomes available for Matlab.

2.1 Working with ANSYS

ANSYS provides two ways of working:

- Graphical User Interface (GUI)
- ANSYS Parametric Design Language (APDL)

Figure 1 shows the GUI of ANSYS. On the left hand side one can see the ANSYS main menu. The roll-up menu *Preprocessor* provides all functions that are necessary to build a finite element model. For an unexperienced user, the work with the GUI is the best way to become familiar with ANSYS.

Beneath the menu bar, one can find the ANSYS command prompt. Here, the APDL commands are entered. APDL is a programming language that can be used for automation or for the development of parameterised models. APDL provides various possibilities like loops, vector and matrix operations. A big advantage of APDL is the possibility of creating macros. The APDL commands can be stored in a *.mac file (e.g. *mymacro.mac*). This script is executed by typing *mymacro* into the command prompt.

TIP: All actions are recorded in a log file (File → List → Log File). There you can find the corresponding APDL command to your GUI action. Thus you are able to create your own macro by simple copy and paste.

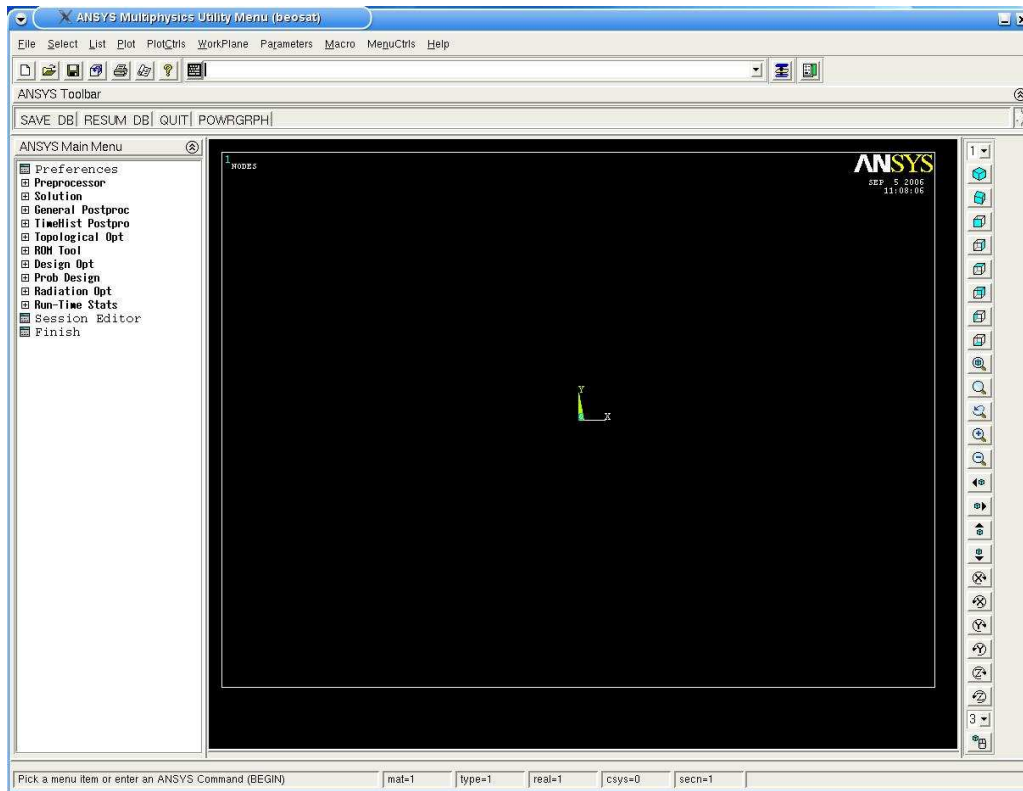


Figure 1: Graphical User Interface of ANSYS

One problem may occur when you are working with ANSYS: a *UNDO* button does not exist. You have to work with the *SAVE_DB* and *RESUM_DB* buttons in the ANSYS toolbar, i.e. you should save your model when you have finished something, so you can reload this state if anything goes wrong afterwards.

2.2 Building a model

This section intends to explain the specific steps that are necessary to build a finite element model and to export the element data for the next process in Matlab.

2.2.1 Element choice

At first, the element type has to be chosen. The ANSYS element library contains approx. 200 different elements for various applications. With regard to the calculation of surface forces, the element type *Shell 63* with four nodes is used, because this number of nodes is required for the following computation. Besides *Shell 63* is well suited for the meshing. You define this element type with the GUI as follows:

Preprocessor → Element Type → Add/Edit/Delete

A window will pop up where you have to click the button **Add**. Then a list with all element types will appear. Choose

Shell → Elastic 4node 63.

The corresponding command in APDL is:

```
ET,EN,SHELL63
```

2.2.2 Material choice

In order to distinguish the different surface properties of the satellite, several material types have to be defined. The definition includes the specification of the elastic modulus (EX) and the major Poisson's ration (PRXY) of each material. These two values may be chosen arbitrary, because they are not used in the following computation process. Only the material number is exported, which is the indicator for the according reflexion and diffusion coefficient of the specific surface. The material type definition is done in the following way:

Preprocessor → Material Props → Material Models

Again, a window will pop up and you may choose an isotropic, linear elastic structural element:

Structural → Linear → Elastic → Isotropic

As mentioned before, the material properties may be chosen arbitrary. The following APDL command defines the same material where the elastic modulus (EX) and the Poisson's ratio (PRXY) is set to zero:

```
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,0
MPDATA,PRXY,1,,0
```

Add as many material models as you need (**Material** → **New model...** in the menu bar of the window). For more details on the material properties definition see ANSYS help.

2.2.3 Geometry definition

The geometry model may be imported from a CAD program or it may be created in ANSYS itself. This section deals with the construction of a satellite model in ANSYS. The model can be build from different geometry elements like keypoints, lines, areas and volumes. The Preprocessor provides two ways of creating a model:

- Bottom-up method: The model is build directly from the metioned basic geometry elements.

- Top-down method: Primitives like rectangles, circles, blocks, cylinders and spheres are used and the model is build through different conjunctions.

The top-down method is especially for simple models the easiest approach and is described in more detail below.

TIP: All geometry elements are numbered. In the menu bar you can find

PlotCtrls → Numbering ...

where you can choose the element type, whose numbering should be visible.

The numbering is important, e.g. if you like to select a special area.

In the menu

Preprocessor → Modeling → Create

a multitude of options to build specific geometries is provided. For example, a block may be build in three different ways:

- Volumes → Block → By 2 Corners & Z
- Volumes → Block → By Centr, Cornr, Z
- Volumes → Block → By Dimensions

One must choose the method that is most suited to the intended satellite geometry. The corresponding APDL command for the creation of a block **By Dimensions** is really simple and looks like:

```
BLOCK,X1,X2,Y1,Y2,Z1,Z2
```

where X1,X2,Y1,Y2,Z1 and Z2 denotes the starting point and the endpoint respectively for each coordinate axis. A detailed description for each APDL command can be found in the ANSYS help.

The creation of a model is different for each satellite, so an explicit instruction can not be provided. At the beginning, try to build the whole satellite geometry with the provided volumes. In

Preprocessor → Modeling → Operate → Booleans

one can find some helpful tools to add, intersect, overlap or substract volumes. By subtraction a cylinder from a block e.g., one get a new volume that contains a hole.

ANSYS works with several coordinate systems. By default, all actions are executed in the so-called **WorkPlane** coordinate system. At the beginning it is aligned with the global coordinate system. But one can move and rotate the **WorkPlane** with respect to the global system arbitrary. It is often easier to move the **WorkPlane** to a specific point

and use a simple create-block-by-dimensions command as to build the same block in the global coordinate system. The rotation and movement of the **WorkPlane** is done in the **WP Offset** window, which can be found in the menu bar under:

WorkPlane → **Offset WP by Increments ...**

The offsets for movement along and the rotation about the coordinate axis is adjusted here:

WorkPlane → **WP Settings ...**

After finishing your volume satellite model, you have to delete all volumes, because the meshing process needs areas only. This is done as follows:

Preprocessor → **Modeling** → **Delete** → **Volumes Only**

If you use the **Pick All** button in the window, all volumes will be selected at once. The model vanishes, because there are no volumes left. With

Plot → **Areas**

in the menu bar, the model is replotted with area elements.

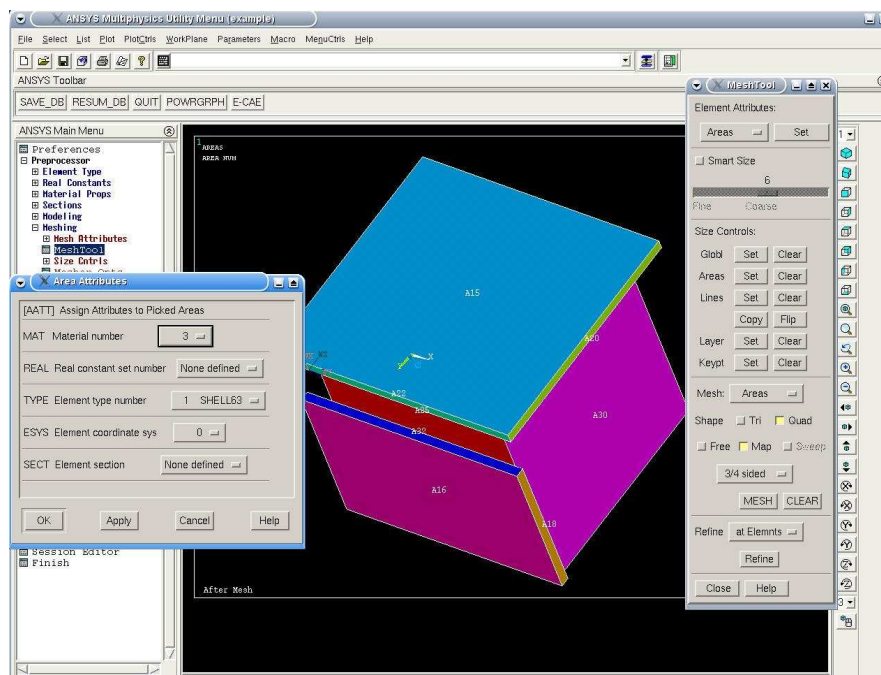


Figure 2: Specification of area attributes

2.2.4 Meshing

The meshing of the satellite model is done with the **Mesh Tool**, which is started with:

Preprocessor → Meshing → Mesh Tool

The Mesh Tool window contains all options, that can be adjusted for the meshing process. At first, the surface properties have to be defined. Choose Areas in the Element Attributes panel and click Set. A window will pop up, where you can enter a list of areas with the same surface properties. Afterwards specify the properties in the Area Attributes window (cf. figure 2).

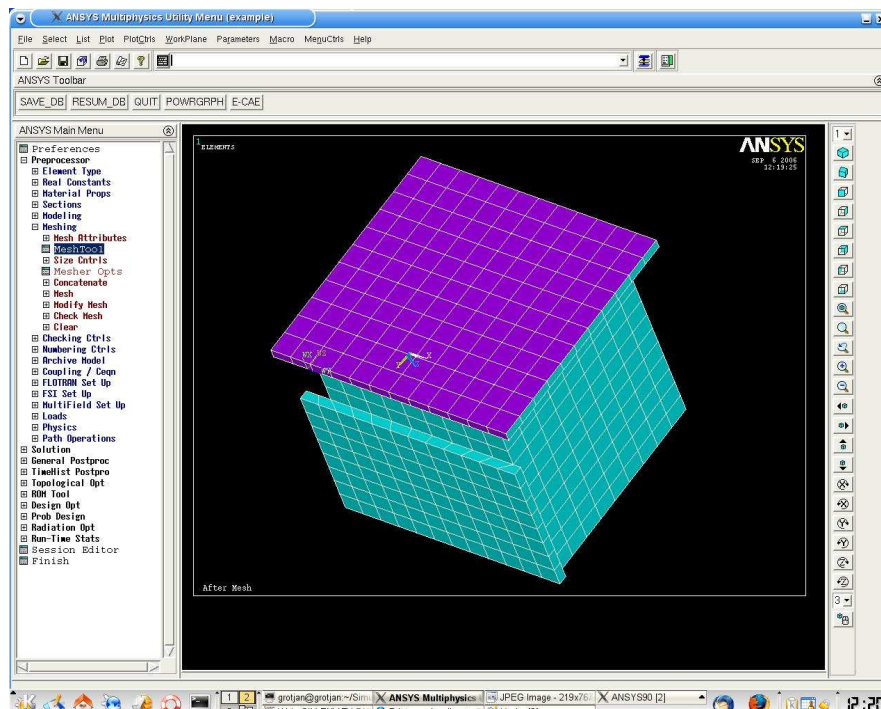


Figure 3: Satellite model after meshing

When all area properties are specified, the element size for the meshing have to be defined. Click the Set button next to Areas in the Size Controls panel. Enter the element edge length for all areas. The element type should be Map. This will produce rectangular elements. Click MESH and select all areas to mesh the whole satellite model. The result looks similar to figure 3.

2.2.5 Data export

The geometry data can be is exported with the macro routine *tables.mac*. This routine must be placed in the current ANSYS working directory. Change the filename *example* in *tables.mac* to *yourfilename*. Afterwards execute the macro by typing *tables* into the command prompt. The routine creates two tables, *et.txt* and *nt.txt*, whose content is listed in table 1 and 2.

Column	Content
1	Element number
2	Area of the element
3	x -coordinate of center point in global coordinate system
4	y -coordinate of center point in global coordinate system
5	z -coordinate of center point in global coordinate system
6	Node number 1
7	Node number 2
8	Node number 3
9	Node number 4
10	Material number
11	Element type number
12	Real constant set number

 Table 1: Element table *et.txt*

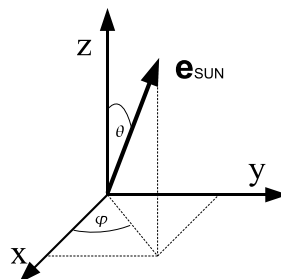
Column	Content
1	x -coordinate of node in global coordinate system
2	y -coordinate of node in global coordinate system
3	z -coordinate of node in global coordinate system

 Table 2: Node table *nt.txt*

3 Look-up table creation

Originally, the Matlab source code for the look-up table creation was written in order to calculate the disturbance forces and torques due to solar pressure, where the force depends on the incident angle of the sun. The intention of the look-up table creation is the calculation of surface forces and torques for arbitrary incident angles with the aid of the geometry that is defined in the element table *et.txt* and the node table *nt.txt*.

Although the code was written for solar pressure problems and therefore most of the


 Figure 4: Definition of polar angle θ and azimuth angle φ

comments in the m-files are referred to this aspect, the algorithms can be used to calculate the influence of different disturbance sources, e.g. the aerodynamic drag. Here the sun vector has to be replaced by the velocity vector und the reflexion coefficients have to be adjusted. Furthermore the Matlab functions, that are presented below, provide only the normalized force and torque, i.e. the results have to be multiplied with the solar pressure for the calculation of solar radiation pressure or with the dynamic pressure for the calculation of aerodynamic drag.

3.1 Data files for the calculation of surface forces and torques

Three data tables are needed as input files for the calculation of surface forces and torques in addition to the geometry tables *et.txt* and *nt.txt*. For the calculation of the relative position of the satellite and the sun, the incident angle is needed. The angle is given in spherical coordinates (cf. figure 4) and is imported from the angle table *at.txt*. The content of *at.txt* is illustrated in table 3. The table may be customized for each purpose.

Column	Content
1	Polar angle θ [$0^\circ \dots 180^\circ$]
2	Azimuth angle φ [$0^\circ \dots 360^\circ$]

Table 3: Angle table *at.txt*

The reflexion coefficients of the surfaces, that consider the different surface properties, are stored in the file *ct.txt*, where each row denotes a different material. The material number (*et.txt*, column 10) refers to the row number. Table 4 shows the content of *ct.txt*.

Column	Content
1	Coefficient of specular reflexion
2	Coefficient of diffuse reflexion

Table 4: Reflexion coefficient table *ct.txt*

For the calculation of the torque due to solar pressure, another information is required. The file *rt.txt* contains the vector to the barycenter of the satellite. The vector is stored as illustrated in table 5.

Column	Content
1	x -coordinate of reference point in global coordinate system
2	y -coordinate of reference point in global coordinate system
3	z -coordinate of reference point in global coordinate system

Table 5: Reference point table *rt.txt*

The file *init_data.m* loads *et.txt*, *nt.txt*, *rt.txt* and *ct.txt* and can be customized, that is one can enter the names of the files that should be used, e.g.:

```
addpath ../projects/Test_case
load nt_testcase.txt;
nt = nt_testcase;
load et_testcase.txt;
et = et_testcase;
load rt_testcase.txt;
rt = rt_testcase;
load ct_testcase.txt;
ct = ct_testcase;
rmpath ../projects/Test_case
```

3.2 Calculation of the surface force and torque for a specific incident angle

The calculation of the surface force and torque for a specific incident angle is done by the Matlab m-file *sat_lighting.m*. It is a helpful tool to check the model, that was built in ANSYS. The illumination conditions of the satellite are coupled to the normal vectors of the surface elements. In some cases, the normal vector of a certain area points inwards due to the node numbering of ANSYS. This results in wrong illumination conditions, e.g. an area in sunlight will be regarded as back side area. So, it is necessary to check the model previous to the creation of the look-up tables.

TIP: Check each side of the satellite model by choosing perpendicular solar radiation. If certain areas are wrong visualised, search the corresponding element numbers in ANSYS. Use the function *flip_normal_vector* for the according rows in *et.txt* (element number = row number). The algorithm changes the order of the nodes in the specific lines, thus the normal vector points in the different direction afterwards.

Customize the m-file *init_data.m* first. The calculation is started by entering *sat_lighting* into the Matlab command prompt. At first, the incident angle has to be chosen by either entering the according line number of *at.txt* or entering directly the values of θ and φ . Afterwards, one may decide whether the force and the torque should be calculated or not. **NOTE:** The calculation yields the normalized force and the torque, a multiplication with the solar pressure is necessary afterwards!

In the end, the satellite model is visualised whereas the colours yellow, red and blue denote the different illumination conditions (cf. figure 5).

The function *sat_lighting* uses other functions, that are provided in the m-files listed in table 6. These m-files as well as the txt-files must be placed in the same working directory

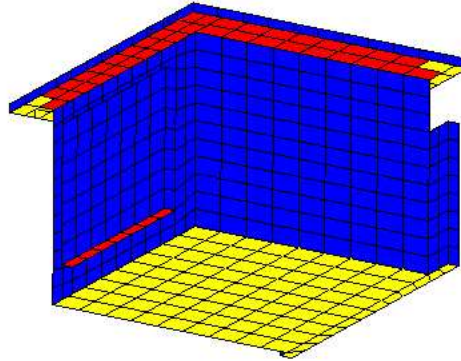


Figure 5: Illumination conditions of satellite model with $\theta = 45^\circ$ and $\varphi = 45^\circ$,
 Yellow: Illuminated Area, Red: Shadow, Blue: Back side

as *sat_lighting.m* or the corresponding Matlab path must be added (e.g. *addpath ../tests*).
 The whole algorithm is described in detail in [1].

m-file	Description
data_storage.m	Computes data matrix <i>dm</i> that contains satellite geometry data and back side flag
eNormal.m	Computes the normal vector of a surface element with 3 points
force_torque.m	Computes the force and torque due to solar radiation for each element
init_data.m	Loads data tables <i>et.txt</i> and <i>nt.txt</i>
inside.m	Function that determines whether a point is inside or outside of a given triangle
mask1.m	Visibility algorithm (1 = element is visible, 0 = element is on back side)
plot_satellite.m	Creates a plot of the satellite model. The illumination conditions are illustrated through different colours.
radforce.m	Computes the normalized force from radiation pressure for a surface element
sat_lighting_setup.m	Sets calculation parameters for <i>sat_lighting</i>
shadow_flag.m	Shadow algorithm returns "1" if a visible element is shaded, otherwise it returns "0"
sp_lookup_setup.m	Sets calculation parameters for <i>sp_lookup</i>
sun.m	Computes the unit vector of the sun in spherical polar coordinate system
torque.m	Computes the torque of each surface element
trans.m	Computes transformation matrix <i>TM</i> : body frame \rightarrow z-axis aligned with sun direction
triangle.m	Divides square element into two triangles

Table 6: Functions, that are used by *sat_lighting* and *sp_lookup*

3.3 Creation of look-up tables

In order to get the forces and torques for arbitrary incident angles, look-up tables are built with the m-file *sp_lookup*. Here, the satellite plots are optional. In contrast to *sat_lighting*, the function *sp_lookup* does not use *at.txt*. One has to define the minimum and maximum value for θ and φ with:

- $0^\circ \leq \theta_{min} < \theta_{max} \leq 180^\circ$ for the polar angle
- $0^\circ \leq \varphi_{min} < \varphi_{max} \leq 360^\circ$ for the azimuth angle

Thus it is possible to investigate a special area of interest. Besides the desired resolution *res* must be chosen. With this information a vector for each angle is built as illustrated in table 7.

Polar angle	
θ_1	0°
θ_2	5°
θ_3	10°
θ_4	15°
θ_5	20°

Azimuth angle	
φ_1	10°
φ_2	20°
φ_3	30°
φ_4	40°

Table 7: Example for $\theta_{min} = 0^\circ$, $\theta_{max} = 20^\circ$, $res_\theta = 5^\circ$
and $\varphi_{min} = 10^\circ$, $\varphi_{max} = 40^\circ$, $res_\varphi = 10^\circ$

Now for each angle combination, the force and the torque is calculated and the specific components are stored in the matrices *FX*, *FY*, *FZ*, *TX*, *TY* and *TZ* (cf. example *FX* in table 8).

	φ_1	φ_2	φ_3	φ_4
θ_1	<i>FX</i> ₁₁	<i>FX</i> ₁₂	<i>FX</i> ₁₃	<i>FX</i> ₁₄
θ_2	<i>FX</i> ₂₁	<i>FX</i> ₂₂	<i>FX</i> ₂₃	<i>FX</i> ₂₄
θ_3	<i>FX</i> ₃₁	<i>FX</i> ₃₂	<i>FX</i> ₃₃	<i>FX</i> ₃₄
θ_4	<i>FX</i> ₄₁	<i>FX</i> ₄₂	<i>FX</i> ₄₃	<i>FX</i> ₄₄
θ_5	<i>FX</i> ₅₁	<i>FX</i> ₅₂	<i>FX</i> ₅₃	<i>FX</i> ₅₄

Table 8: Content of *FX*

The polar angles and the azimuth angles are stored in the vectors *Tht* and *Phi* respectively. The mentioned variables are merged in the *.mat files *force.mat*, *torque.mat* and *angles.mat*. Thus they are available for further usage.

3.4 C Code

The execution of the *sp_lookup* Matlab code and even of *sat_lighting* for models with a fine mesh is very time-consuming. Hence the whole code has been translated to C. Some changes in the handling of the programmes are required which will be outlined in this section.

The corresponding file to *init_data.m* is *setup_data.c*. This file reads from *setup.txt*. Here one can enter the names of the data files that should be used, e.g.:

```
../projects/Test_case/nt_testcase.txt
../projects/Test_case/et_testcase.txt
../projects/Test_case/rt_testcase.txt
../projects/Test_case/ct_testcase.txt
```

As it is not possible to get a plot of the satellite directly in C, the corresponding programme creates the file *data_matrix.txt*, which contains all information that is required to make the satellite plot. This is again done in Matlab but with the file *plot_satellite_c.m*.

The routine *sp_lookup.c* works in the same way as the according Matlab code though without plot option. The results are stored in *force_FX.txt*, *force_FY.txt*, *force_FZ.txt*, *torque_TX.txt*, *torque_TY.txt*, *torque_TZ.txt*, *angle_tht.txt* and *angle_phi.txt* for later usage.

Since Matlab already provides some functions that are not available in C, they had to be added. A complete overview of all required c-files and header files is given in table 9 in the appendix.

4 Simulation in Matlab/Simulink

The model for the force/torque simulation is provided in the Simulink library *SurfaceForce_lib.mdl*, cf. figure 6.

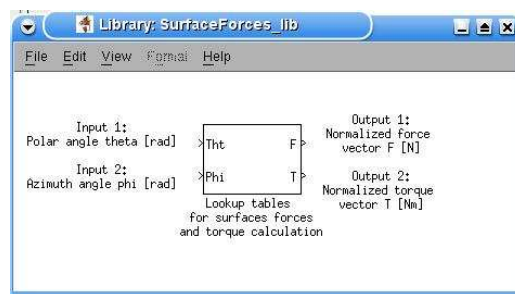


Figure 6: Simulink library SurfaceForce_lib.mdl

Figure 7 shows the content of the library module. It provides a look-up table for each component of the force and the torque vector respectively. The look-up tables work

with an interpolation/extrapolation method. Thus not every angle combination must be considered during the creation of the look-up tables. During the simulation the force and torque is calculated against the incoming polar angle θ and the azimuth angle φ . These angles result from the sun vector or the velocity vector, expressed in the satellite body frame.

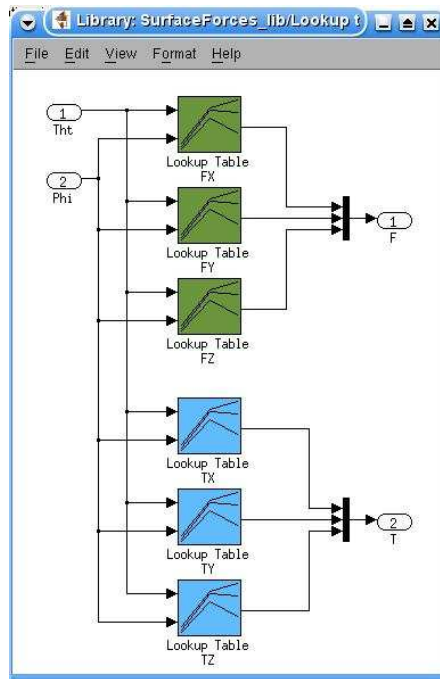


Figure 7: Look-up tables for forces and torques

The input data, i.e. the variables FX , FY , FZ , TX , TY , TZ , $Thet$ and Phi have to be loaded into the Matlab workspace before starting the simulation. That is done by entering

- load forces
- load torque
- load angles

into the Matlab command prompt. In case C code was used to produce the look-up tables, all *.txt files mentioned in section 3.4 must be loaded into the Matlab workspace (*load force_FX.txt*; etc.). Afterwards the variable name must be adjusted either in the Matlab workspace ($FX = force_FX$) or in the Simulink look-up table block.

The output vectors are normalized (cf. section 3), so they have to be multiplied with the solar pressure or the dynamic pressure to get the corresponding force and torque.



References

- [1] S. Bremer. Modeling of solar radiation force and torque. Technical Report FLK-SIM-TN-ZAR-009, ZARM, University of Bremen, 2008.
- [2] M. Cicek. Anleitung zur Modellierung von Oberflächenkräften. Technical Report SIM-ENV-TU-ZAR-001, ZARM, University of Bremen, 2005.
- [3] M. Cicek. Entwicklung und Validierung einer Methode zur Berechnung der Oberflächenkräfte auf Satelliten. Master's thesis, Hochschule Reutlingen, June 2005.

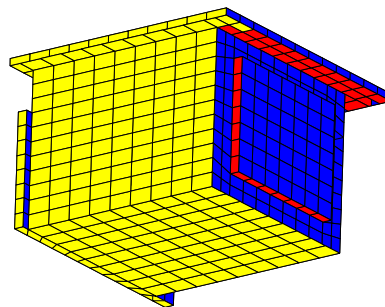
A Appendix

c-file	header	Description
condition.c	condition.h	Determines condition of a matrix
data_storage.c	data_storage.h	Computes data matrix that contains satellite geometry data and back side flag
eNormal.c	eNormal.h	Computes the normal vector of a surface element with 3 points
force_torque.c	force_torque.h	Computes the force and torque due to solar radiation for each element
functions.c	functions.h	Mathematical auxiliary functions
inside.c	inside.h	Function that determines whether a point is inside or outside of a given triangle
inverse.c	inverse.h	Calculates inverse of a given matrix
mask1.c	mask1.h	Visibility algorithm (1 = element is visible, 0 = element is on back side)
mat_vec_mult.c	mat_vec_mult.h	Matrix - Vector - Multiplication
quicksort.c	quicksort.h	Sorting algorithm
radforce.c	parameters.h radforce.h	Includes main parameters for the whole programme Computes the normalized force from radiation pressure for a surface element
sat_lighting.c	sat_lighting.h	Main programme
sat_lighting_cal.c	sat_lighting_cal.h	Additional subroutine for some calculations
sat_lighting_setup.c	sat_lighting_setup.h	Sets calculation parameters for <i>sat_lighting</i>
setup_data.c	setup_data.h	Loads data files of specific project
shadow_flag.c	shadow_flag.h	Shadow algorithm returns "1" if a visible element is shaded, otherwise it returns "0"
sp_lookup.c		Main programme
sp_lookup_setup.c	sp_lookup_setup.h	Sets calculation parameters for <i>sp_lookup</i>
sun.c	sun.h	Computes the unit vector of the sun in spherical polar coordinate system
torque.c	torque.h	Computes the torque of each surface element
trans.c	trans.h	Computes transformation matrix TM : body frame $\rightarrow z$ -axis aligned with sun direction

Table 9: List of required c-files and header files



Modeling of solar radiation force and torque



Project:	Document No.:
First Look	FLK-SIM-TN-ZAR-009
ZARM - Center of Applied Space Technology and Microgravity	
Am Fallturm	Phone: +49-421-218-4803
D 28359 Bremen	Fax: +49-421-218-2521
Germany	E-Mail: bremer@zarm.uni-bremen.de

Doc. No.: FLK-SIM-TN-ZAR-009


Issue: 1.1

Written: Stefanie Bremer

Date: 10th April 2008

Approved:

Date:

	<p style="text-align: center;">Modeling of solar radiation force and torque</p>	<p>Doc.No.: FLK-SIM-TN-ZAR-009 Issue: 1.1 Page: 2 of 12</p>
---	---	---

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	18/03/08	all	Initial Version by Stefanie Bremer	✓
1.1	10/04/08	all	Updated Version by Stefanie Bremer	✓

Contents

1	Introduction	4
2	General idea	4
3	Illumination conditions	4
3.1	Backside flag	7
3.2	Shadow algorithm	8
4	Force and torque calculation	11

1 Introduction

This document describes the modeling of forces and torques that affect a satellite due to solar radiation. As extension to the document *FLK-SIM-TU-ZAR-001: Modeling of surface forces - Tutorial*, this paper deals with the theoretical background of the presented Matlab/C code functions (cf. [2], chapter 3). The documentation is based on the master thesis *Entwicklung und Validierung einer Methode zur Berechnung der Oberflächenkräfte auf Satelliten* by Mehmet Cicek [1].

2 General idea

The magnitude of the solar radiation force depends on several factors. Besides the incident sun angle and the illuminated satellite area, the different surface properties of the satellite, i.e. the coefficient of specular and diffuse reflexion respectively, must be considered, too.

For coarse estimations of disturbance forces due to solar radiation pressure, it is often sufficient to assume a reference satellite area with average reflexion coefficients. For a detailed analysis, a more accurate approach is required, whereas the illuminated area is calculated against the incident sun angle and the satellite geometry, i.e. shadowing effects must also be taken into account. Furthermore, it is necessary to distinguish between different materials.

The general idea to cope with this problem is to use a model of the regarded satellite whose surface is discretized with small quadrangular elements. Thus it is possible to compute the resulting force for each element in consideration of their material properties. Besides the division in single elements provides the basis for the algorithm that calculates the illumination conditions. This algorithm is presented in section 3. The following derivation of the force and the torque is outlined in section 4. The required geometry information is given in the element table *et.txt* and the node table *nt.txt*, whose contents is listed in the tables 1 and 2. These tables can be created with ANSYS (details: [2], chapter 2).

There are two different Matlab/C programmes to calculate the solar radiation force and torque with the methods presented below. The programme *sat_lighting* computes the radiation force and torque for one selected illumination case, *sp_lookup* considers a specific range of both angles, i.e. a multitude of different cases is regarded. For details on the usage of both programmes refer to [2], chapter 3.

3 Illumination conditions

The computation of the illumination conditions is divided into two parts. First, all elements must be identified that are exposed to the sun's radiation. Afterwards the

Column	Content
1	Element number
2	Area of the element
3	x -coordinate of center point in global coordinate system
4	y -coordinate of center point in global coordinate system
5	z -coordinate of center point in global coordinate system
6	Node number 1
7	Node number 2
8	Node number 3
9	Node number 4
10	Material number
11	Element type number
12	Real constant set number

Table 1: Element table *et.txt*

Column	Content
1	x -coordinate of node in global coordinate system
2	y -coordinate of node in global coordinate system
3	z -coordinate of node in global coordinate system

Table 2: Node table *nt.txt*

visible elements are examined again to distinguish whether they are shaded by other elements or not.

Starting point of each calculation is the incident sunray. This vector is considered in the satellite body-fixed coordinate system in polar coordinates θ and φ (cf. figure 1). The polar angle θ reaches from 0 (= z -axis) to π ($-z$ -axis) and the azimuth angle φ reaches from 0 to 2π (x - y -plane). Thus any sun vector can be described.

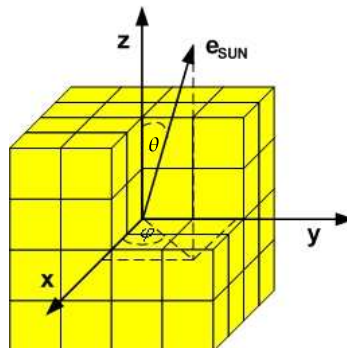


Figure 1: Sun vector in body-fixed polar coordinates

The normalised sun vector \underline{e}_{sun} is the baseline for all further computations (*sun.m/sun.c*):

$$\underline{e}_{sun} = \begin{pmatrix} \sin \theta \cos \varphi \\ \sin \theta \sin \varphi \\ \cos \theta \end{pmatrix} \quad (1)$$

In order to prepare the shadow algorithm, the following actions take place: The \underline{z} -axis of the satellite body-fixed frame is aligned with the sun direction \underline{e}_{sun} and becomes $\hat{\underline{z}}$ of the red system as illustrated in picture 2.

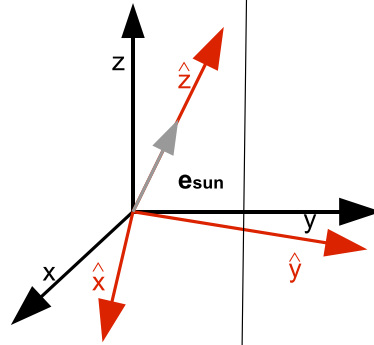


Figure 2: Coordinate system transformation

According to [3], p.761, the following transformation matrix can be used for this operation:

$$\underline{A} = \begin{pmatrix} \hat{x} \cdot \underline{x} & \hat{x} \cdot \underline{y} & \hat{x} \cdot \underline{z} \\ \hat{y} \cdot \underline{x} & \hat{y} \cdot \underline{y} & \hat{y} \cdot \underline{z} \\ - & - & \hat{z} \cdot \underline{z} \end{pmatrix} \quad (2)$$

Here the unit vectors in the satellite body-fixed frame (black system) are given with:

$$\underline{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \underline{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \underline{z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

An auxiliary x-axis may be chosen arbitrary. Here it is aligned with the old z-axis, unless the sun vector matches the old one:

$$\hat{\underline{x}}^* = \begin{cases} - & = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ \begin{pmatrix} -1 & 0 & 0 \end{pmatrix} & \text{if } \hat{\underline{z}} = \begin{pmatrix} 0 & 0 & -1 \end{pmatrix} \\ -\underline{z} & \text{otherwise} \end{cases} \quad (4)$$

The $\hat{\underline{x}}^*$

Afterwards the correct \hat{x} -axis is derived as follows:

$$\hat{x} = \hat{y} \times \hat{z} \quad (6)$$

Now all information is available that is required to compute the transformation matrix \underline{A} . This is done by the function *trans.m/trans.c*. Afterwards the center coordinates and the node coordinates are transformed into the new coordinate system (*data_sotrage.m/data_storage.c*):

$$\begin{aligned} \hat{C} &= \underline{A} \cdot C \\ \hat{N}_1 &= \underline{A} \cdot N_1 \\ \hat{N}_2 &= \underline{A} \cdot N_2 \\ \hat{N}_3 &= \underline{A} \cdot N_3 \\ \hat{N}_4 &= \underline{A} \cdot N_4 \end{aligned} \quad (7)$$

The center vector of each element \underline{C} and the numbers of the four nodes are given in *et.txt* (cf. table 1). The according node coordinates are stored in *nt.txt* (e.g. node number = 5 \implies row 5 of node table). The transformed elements are now arranged along the $\hat{z} = e_{sun}$ axis in descending order (*sortrows/quicksort.c*).

3.1 Backside flag

The so-called backside flag indicates whether an element is hit by a sunray (backside flag = 1) or not (backside flag = 0) without considering shadow effects. All backside elements are identified by comparing the angle between the sun vector and the normal vector \underline{e}_N of the regarded element. Its normal vector is derived from the node coordinates \underline{N}_1 to \underline{N}_3 (*eNormal.m/eNormal.c*):

$$\underline{e}_N = \frac{(\underline{N}_2 - \underline{N}_1) \times (\underline{N}_3 - \underline{N}_1)}{|(\underline{N}_2 - \underline{N}_1) \times (\underline{N}_3 - \underline{N}_1)|} \quad (8)$$

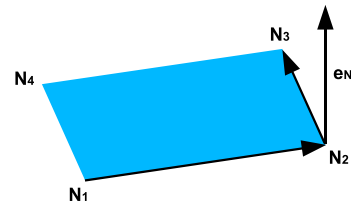


Figure 3: Element normal vector

NOTE: Here, the original node coordinates must be used, not the transformed ones!

If the angle α , calculated with the scalar product of e_{sun} and \underline{e}_N (equation 9), is larger than 90° , the element is not illuminated by the sun (cf. figure 4). This computation takes place in *mask1.m/mask1.c*.

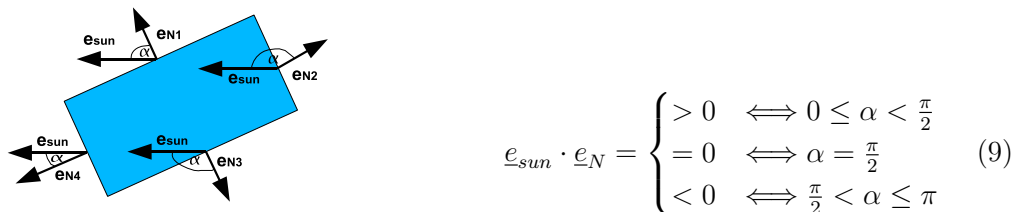


Figure 4: Backside determination

All elements are now classified as visible and invisible, respectively. Figure 5 shows an example satellite after performing the backside algorithm. Blue elements are backside elements, the yellow ones are in principle illuminated by the sun, since shadow effects are not taken into account so far.

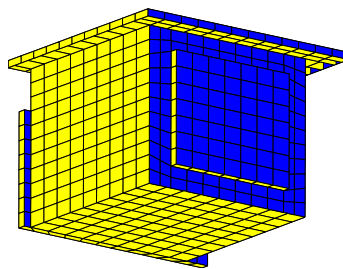


Figure 5: Example satellite after backside determination

3.2 Shadow algorithm

The shadow algorithm (*shadow_flag.m/shadow_flag.c*) is applied to all visible elements only. The idea of this algorithm is illustrated in figure 6.

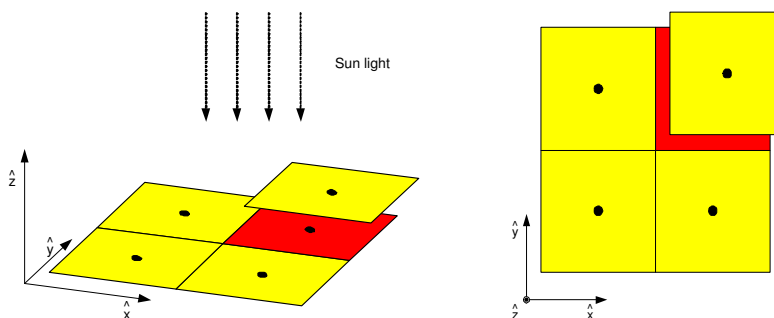


Figure 6: Illustration of shadow algorithm; left: dimetric projection, right: top view

As one can see, the algorithm does not produce a clear, sharp line of shadow, but it checks for each element whether it is shaded by another one or not. This decision depends on the position of the center point of an element with respect to the area of the elements above. If the center point is located inside an area, the element is considered as shaded (shadow flag = 1). In this way, errors occur as half of the element area might be considered wrong. Since this happens in both directions, the overall error is small. Apart from this, the error can be minimized by reducing the element size. The mathematical description of the algorithm is presented below.

All elements are regarded in the \hat{x} - \hat{y} -plane (new coordinate system, introduced above) where they are hit by perpendicular sunrays. Depending on the direction of the sunlight, the transformed elements are deformed or even become lines. This does not affect the function of this shadow algorithm, therefore it must not be taken into account.

A loop is applied to all visible elements starting from the element that is nearest to the sun (\implies sorting, see above), because it is in either case illuminated. As depicted in figure 6, only elements in a certain radius beneath the currently regarded element may be shaded. Hence, only these elements are considered in the current calculation step, basically in order to speed up the whole calculation process. The reference radius for this purpose is given by:

$$r_{ref} = 1.5 \cdot a_{max} \quad (10)$$

Here, a_{max} denotes the maximal edge length of all elements, which is derived from *et.txt*. A safety factor of 1.5 is introduced to assure, that highly deformed elements are taken into account as well.

Now the currently regarded element is divided into two triangles as illustrated in figure 7 (*triangle.m/triangle.c*).

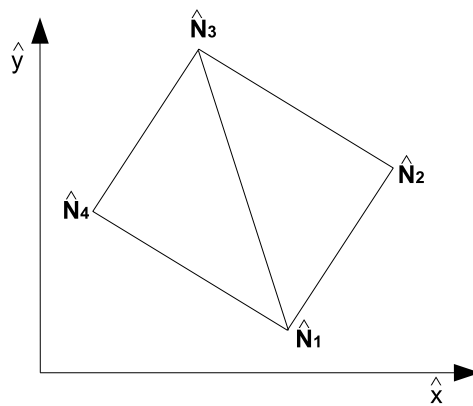


Figure 7: Rectangle \longrightarrow 2 triangles

Two vectors are obtained for each of the triangles, that contain the nodes' \hat{x} - and \hat{y} -

coordinates respectively:

$$\begin{aligned}
 \underline{a}_1 &= (x_{\hat{N}_1} \quad x_{\hat{N}_2} \quad x_{\hat{N}_3}) \\
 \underline{b}_1 &= (y_{\hat{N}_1} \quad y_{\hat{N}_2} \quad y_{\hat{N}_3})
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 \underline{a}_2 &= (x_{\hat{N}_1} \quad x_{\hat{N}_3} \quad x_{\hat{N}_4}) \\
 \underline{b}_2 &= (y_{\hat{N}_1} \quad y_{\hat{N}_3} \quad y_{\hat{N}_4})
 \end{aligned} \tag{12}$$

Figure 8 shows the definition of the barycentric coordinates c_1 , c_2 and c_3 .

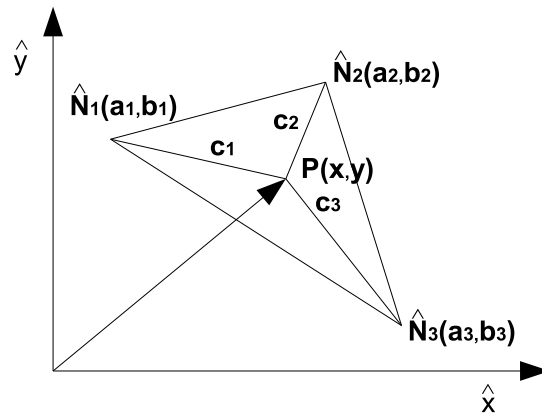


Figure 8: Barycentric coordinates in a triangle

They can be used to determine, whether a point P is inside or outside a given triangle. The solution of the system of equations (eq. 13) provides c_* . If all barycentric coordinates are non-negative, the point P is inside the considered triangle.

$$\begin{aligned}
 c_1 + c_2 + c_3 &= 1 \\
 a_1 c_1 + a_2 c_2 + a_3 c_3 &= x \\
 b_1 c_1 + b_2 c_2 + b_3 c_3 &= y
 \end{aligned} \tag{13}$$

This system of equation is solved for both triangles defined in equation 11 and 12 with $P(x, y) = \hat{C}(\hat{x}, \hat{y})$ and \hat{C} within the reference radius (*inside.m/inside.c*). If the center point is inside one of those triangles or on the edge of one, the element is considered as shaded.

The algorithm is performed for all other visible elements as well. Afterwards the appearance of the example satellite changes to figure 9. The shaded elements are colored in red.

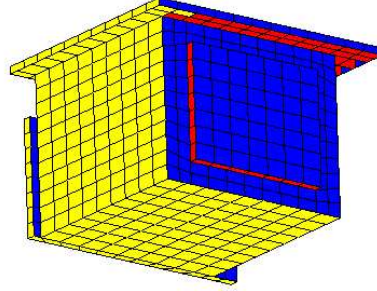


Figure 9: Example satellite after shadow algorithm

4 Force and torque calculation

After determining the illumination conditions, the last step contains the calculation of the specific radiation force \underline{f}_i for all illuminated elements (*radforce.m/radforce.c*):

$$\underline{f}_i = -A_i[(1 - c_{si}) \cdot \underline{e}_{sun} + 2(c_{si} \cos \alpha + \frac{1}{3}c_{di}) \cdot \underline{e}_N] \cos \alpha \quad (14)$$

The sun vector \underline{e}_{sun} , the normal unit vector \underline{e}_N of the element i and the angle between them $\alpha = \underline{e}_{sun} \cdot \underline{e}_N$ were already introduced above. Besides, the element area A_i from *et.txt* and the coefficient of specular c_{si} and diffuse c_{di} reflexion respectively taken from *ct.txt* (cf. [2]) are needed. The solar pressure

$$P_{sol} = \frac{S_0}{c \cdot |r|^2} \quad , \quad (15)$$


where S_0 denotes the solar constant at 1 AU, c is the speed of light and r is the distance to the sun, is equal for each element. Therefore it is not taken into account in equation 14, but multiplied with the sum of the element forces afterwards:

$$\underline{F}_{sp} = P_{sol} \cdot \sum_{i=1}^n \underline{f}_i \quad (16)$$

The torque calculation involves the reference point vector \underline{r}_i of each element, given in *rt.txt* (cf. [2]). The computation takes place in *torque.m/torque.c*. According to the force, the total torque is achieved by summing up all element torques:

$$\underline{T}_{sp} = P_{sol} \cdot \sum_{i=1}^n \underline{f}_i \times \underline{r}_i \quad (17)$$

The programmes *sat_lighting* and *sp_lookup* mentioned in section 2, compute $\frac{\underline{F}_{sp}}{P_{sol}}$ and $\frac{\underline{T}_{sp}}{P_{sol}}$. Thus it is possible to use different kinds of solar pressure models without repeating the whole computation process. This approach is very helpful for simulations where time-dependency is considered.

	Modeling of solar radiation force and torque	Doc.No.: FLK-SIM-TN-ZAR-009 Issue: 1.1 Page: 12 of 12
---	--	---

References

- [1] M. Cicek. Entwicklung und Validierung einer Methode zur Berechnung der Oberflächenkräfte auf Satelliten. Master's thesis, Hochschule Reutlingen, June 2005.
- [2] S. Grotjan. Modeling of surface forces - Tutorial. Technical Report FLK-SIM-TU-ZAR-001, ZARM, University of Bremen, 2006.
- [3] J.R. Wertz, editor. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978.

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 2 of 30
---	---------------------------------	--

Document Change Record

Issue	Date	Changed Pages / Changed Chapters	Remarks	Done
1.0	23/05/06	all	Initial Version by Ivanka Pelivan	✓
1.1	16/08/06	4.2	Updated Version by Stefanie Grotjan	✓
1.2	30/10/06	new Chapter 3 (Installation)	Update by Ivanka Pelivan	✓
1.3	15/02/07	2, added 3.3 and 7.1	Update by Ivanka Pelivan	✓
1.4	09/05/07	3.2	Update on Windows installation	✓
1.5	18/03/08	chapters 4, 6 and 7	Update by Stefanie Bremer	✓
1.5	11/04/08	update of figure 2	Update by Stefanie Bremer	✓

Contents

1	Introduction	4
2	DF Simulator Features	5
3	Installation of the DF Simulator	7
3.1	Linux Installation (Default)	7
3.2	Windows Installation	8
3.3	DF Simulator Make Commands	10
4	DF Simulator Structure	12
4.1	DF Simulator Versions	12
4.2	DF Simulator Directory Structure	13
5	Interface Definition	17
6	Interface for the Drag-Free Simulator Core	20
6.1	Matlab/Simulink Simulation Environment	20
6.2	Interface Layout	24
6.3	Wrapper for the Core	24
7	DF Simulator Application	26
7.1	Matlab Script Options - Dynamics Core	26

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 4 of 30
---	---------------------------------	--

1 Introduction

In this document the drag-free simulator (DF simulator) functionality, organisation and application is described in chapters 2, 4 and 7.

For programmers, this document also defines the interface to be used for all modules of the drag-free simulator (chapter 5). For use with Matlab/Simulink, interfaces are commonly S-functions coded in C. The interface definition presented in this document provides a common structure for all modules of a simulator.

2 DF Simulator Features

The features of the DF simulator core include:

- Simulation of full satellite and test mass dynamics in six degrees of freedom by numerical integration of the equations of motion (13 states: attitude rate, quaternion, position, velocity)
- Up to four differential accelerometers utilizing two test masses each (8 (TMs) + 1 (Satellite) = 9x13 states = 117 states)
- Consideration of linear and nonlinear coupling forces and torques between satellite and test masses as well as between test masses
- Modelling of cross-coupling interaction
- Earth gravity model choice: EGM96 and EIGEN-GL04C up to 360th degree and order, GGM02C up to degree and order 200, patched with EGM96 coefficients for higher degrees
- Gravitational influence of Sun, Moon and planets can be included
- Gravity-gradient torques and forces
- Integration methods: 5th order Runge-Kutta, Bulirsch-Stoer, Euler-Cauchy (small step size only)
- up to version v-1-2: 128 bit numerical precision ('quad precision') on an ALPHA processor

Several error sources are considered in the model:

- + misalignment and attitude errors
- + coupling biases
- + displacement errors

From version v-1-3 on the `libgravity_alpha.a` has been removed since the ALPHA is not working properly anymore. It is anticipated that in the near future a version with 128 bit numerical precision on an INTEL will be available.

The rough structure of the dynamics core is displayed in figure 1. The function `df_sim_init` initializes the states, parameters and options. In `df_sim.loop` the equations of motion for the satellite and test masses are integrated and the states are returned. For a description of the equations of motion incorporated in the DF simulator core see [5].

Within the integration loop the gravity library is called. The gravity model is described in detail in [7] for EGM96 and in [6] for the newer models EIGEN-CG03C, EIGEN-GL04C and GGM02C. The overall structure of the modular DF simulator is displayed in the next chapter.

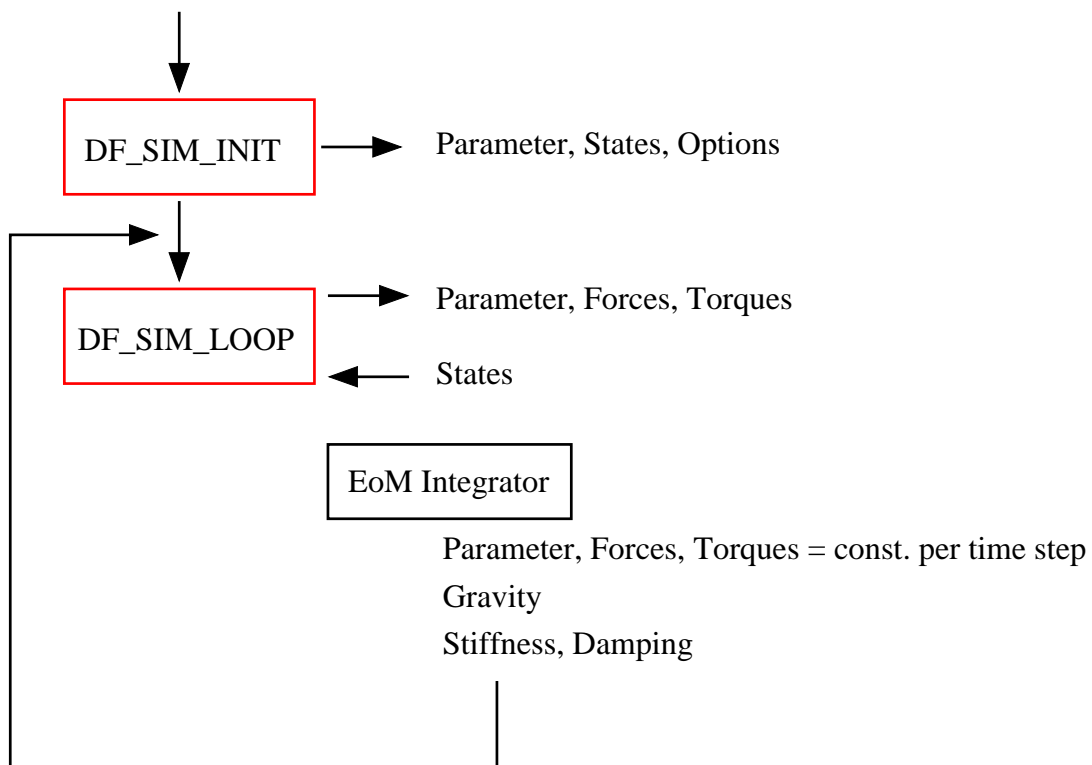


Figure 1: DF Simulator Core Structure

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 7 of 30
---	---------------------------------	--

3 Installation of the DF Simulator

The DF Simulator is developed on a Linux platform. Installation under Linux is described below. Installation under Windows principally is the same as under Linux after a few additional steps have been carried out. The Windows installation is described in section 3.2.

3.1 Linux Installation (Default)

The main part of the simulator including the dynamics core is written in Fortran or C/C++ with a user interface in Matlab/Simulink. As compiler for the Fortran code *gfortran* is used and for C/C++ *gcc/g++*. To install *gfortran* (as root) carry out the following steps:

1. Download binary from GCCWiki - GFortran:
<http://gcc.gnu.org/wiki/GFortranBinaries>
2. Unpack the downloaded tar.gz (e.g. *gfortran-linux.tar.gz*) in directory */usr* (or a directory of your choice):

```
gunzip gfortran-linux.tar.gz
```

```
tar -xvf gfortran-linux.tar
```

The *gfortran* directory is called *irun*.

3. Create a symbolic link from */usr/irun* (or whatever you want to rename it, change the following command accordingly):

```
ln -s /usr/irun/bin/gfortran /usr/bin/gfortran
```

4. Add your library path */usr/irun/lib* to */etc/ld.so.conf*, then update with:
`ldconfig -v`

The DF Simulator is under version control using SVN. With

```
svn co https://spacetec/svn/df_simulator
```

the complete project including tagged versions from the former SVN repository can be checked out (use: `svn co https://spacetec/svn/df_simulator/trunk` to get the newest version only).

In order to compile the core, run `make top` inside folder *df_simulator/trunk*. If the libraries need to be re-compiled, type `make libs`. For a list of make options including the explanation of their use simply type `make` or see below (section 3.3).

3.2 Windows Installation

In order to compile the DF Simulator on Windows, a Linux-like environment for Windows is necessary.

In this section the prerequisites and instructions how to generate a Windows executable of the df_simulator (software status of 2007/05) are given.

The prerequisites are:

1. Cygwin (or another Linux-like environment for Windows)
 2. gcc.4.1.1 (or higher), including gfortran
1. The cygwin installation has to be carried out in the following way:
- download the cygwin tarball from <http://cygwin.com/>
 - install by executing setup.exe with gcc-, gmp-, libgmp-devel-, mpfr-, libmpfr-devel-, make-, mingw-, and perl-packages (select under categories devel, libs, mingw, perl)

As of 2007/05 this version includes gcc-3.4.4 but not gfortran. The gfortran version available from <http://gcc.gnu.org/wiki/GFortranBinaries> uses gcc-4.2.0. The df_simulator project uses gfortran libraries and libstg++.a which is not included in the gcc package that comes with gfortran. The use of libstg++.a results in an error message declaring undefined references which are probably caused by having two libgcc.a's (one being used by gcc and one for gfortran). The newer libgcc is necessary for the gfortran libs but does not include some of the procedures called by libstg++. To avoid these difficulties download a newer version of gcc (4 and up) that includes gfortran (therefore referring to the same libgcc).

Note: A newer Cygwin may include gcc.4. But you may have to re-build gcc anyway to set it up for mingw (see 2. below).

2. The built of gcc (reminder: gcc.4... snapshot and higher required → includes gfortran) has to be set up for mingw since the built of dll's using the cygwin gcc results in a dll that is calling the cygwin1.dll which apparently is not compatible with MATLAB71 (maybe later versions will be). The following links have to be set:

- in folder /usr/local/bin
 - ln -s /usr/bin/ar.exe ar.exe
 - ln -s /usr/bin/ld.exe ld.exe
- in folder /usr/local/gcc-4.1.1/bin
 - ln -s /usr/bin/ar.exe ar.exe



- ln -s /usr/bin/as.exe as.exe
- ln -s /usr/bin/ld.exe ld.exe
- ln -s /usr/bin/nm.exe nm.exe

For higher versions you might also need to set the corresponding mingw32-links, i.e.

- in folder /usr/local/bin
 - ln -s /usr/bin/ar.exe mingw32-ar.exe
 - ln -s /usr/bin/ld.exe mingw32-ld.exe
- in folder /usr/local/gcc-4.1.1/bin
 - ln -s /usr/bin/ar.exe mingw32-ar.exe
 - ...

Execute in gcc folder (here gcc-4.1.1):

```
1) ./configure --prefix=/usr/local/gcc-4.1.1 --target=mingw32 --with-head  
ers="/usr/include/mingw" --with-headers="/usr/include/w32api" --with-libs  
="/usr/lib/mingw" --with-headers="host-i686-pc-cygwin/gcc"
```

(Note: `--with-headers` copies necessary header files for the mingw installation in a mingw- folder. Only the last `--with-headers` path is executed, that means that `./configure` currently has to be executed three times

- a) `./configure --prefix=/usr/local/gcc-4.1.1 --target=mingw32 --with-head
ers="/usr/include/mingw" --with-libs="/usr/lib/mingw"`
- b) `./configure --prefix=/usr/local/gcc-4.1.1 --target=mingw32 --with-head
ers="/usr/include/w32api" --with-libs="/usr/lib/mingw"`
- c) `./configure --prefix=/usr/local/gcc-4.1.1 --target=mingw32 --with-head
ers="host-i686-pc-cygwin/gcc" --with-libs="/usr/lib/mingw")`

If folder `host-i686-pc-cygwin` does not exist, continue with 2) `make`. The folder `host-i686-pc-cygwin` should be created now. If `make` does not run through, execute again

```
c) ./configure --prefix=/usr/local/gcc-4.1.1 --target=mingw32 --with-head  
ers="host-i686-pc-cygwin/gcc" --with-libs="/usr/lib/mingw")
```

Then go on with:

- 2) `make`
- 3) `make install`

In folder /usr/local/bin set the link (version example 4.1.1):

```
ln -s ../gcc-4.1.1/bin/mingw32-gcc.exe mingw32-gcc.exe (to not overwrite default gcc compiler)
```

3. Make an import library from the dll's libmex, libmat and libmx:

```
cd $MATLAB_HOME (set your matlab home directory first)
dlltool -l bin/win32/libmex.a -D bin/win32/libmex.dll -d extern/include/libmex.def (same for libmat and libmx)
```

If you want to test a simple example first: the matlab example timestwo.c can be translated the following way:

```
mingw32-gcc -v -g -O2 -shared -fexceptions -otimestwo.dll -I$MATLAB_HOME/extern/include -I$MATLAB_HOME/simulink/include -L$MATLAB_HOME/bin/win32 timestwo.c -lmat -lmex -lmx -DMATLAB_MEX_FILE
```

The df_simulator.dll is generated with

1) make libs

2) make top

using the same make options as for the Linux installation.

Note: The cygwin installation has been carried out with gcc versions 4.1.1 and 4.1.2. Newer versions might bring new challenges...

3.3 DF Simulator Make Commands

The simulator make options include:

- libcoeff - creates default input coefficients for the gravity library (the current default is EGM96)
- libs - creates libraries for the simulator
- top - builds the complete simulator from the top directory
- clean - deletes all object files in the current directory and all subdirectories
- realclean - deletes all object files, backup files, and all dependency files in the current directory and all subdirectories
- coreclean - realclean for the directories common, dynamics and run
- envclean - realclean for the directories common, and environment

In the subdirectories the following targets can be used:

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 11 of 30
---	---------------------------------	---

- this - compiles the binaries of the current directory
- module - compile the current branch of the source tree

4 DF Simulator Structure

4.1 DF Simulator Versions

Up to June 2006 the DF Simulator development has been under CVS (Concurrent Versions System) control. Thereafter SVN is used for revision control. The tags assigned to subversions of the DF simulator under CVS control are saved in a folder named *tags*. There are several versions of the DF simulator available, the highest increment specifying the latest version:

- v-1-0: Initial version as taken over from GDFS source tree
- v-1-1: Updated version with (1) cross-coupling between translational and rotational motion, (2) a mechanical reference frame which allows the change of the center of mass position and (3) the gravitational quadrupole effect of the spherical Earth
- v-1-2: Updated version with the force and torque due to the gravity gradient effect of the oblate Earth
- v-1-3: Updated version with improved ECI to ECEF transformation containing (1) earth rotation, (2) IAU 1976 precession theory, (3) IAU 1980 nutation theory, and (4) polar motion
- v-1-4: New MATLAB/Simulink-to-Core interface and structure definition
- v-1-5: New directory structure including source code for dynamics core as well as environment modelling
- v-2-0: Updated version to utilize gfortran for compilation
- v-2-1: Atmospheric model added (NRLMSISE-00 + HWM93), new ephemeris calculation algorithm
- v-2-2: Backup tag, final version where ephemeris and transformation matrix calculation is done inside the `df_simulator` kernel
- v-3-0: Ephemeris and ECI to ECEF transformation matrix calculation is removed from the kernel and takes place in separate s-functions

Changes in the Matlab parameter structure from version v-1-3 to v-1-4 are documented in section 6.1.

Newer versions have not been tagged specifically because each check-in of a modified file is accompanied by a new revision number for the whole simulator project (use

`svn log <file>` to see its history). Major changes however should be tagged in a more human-friendly way (and saved in subfolder *tags*). It is recommended to use the latest version (in folder *trunk*).

4.2 DF Simulator Directory Structure

In figure 2 the DF simulator directory structure is displayed. Up to version v-1-3 the directories *common* and *environment* have not been part of the DF simulator project. In *common*, source code accessed by various modules is stored. The directory *dynamics* contains the source code for the DF simulator core. In *environment* (cf. figure 3) the *gravity* source code to generate the library `libgravity_intel.a` (placed in folder *lib*) is checked in. The gravity library uses the transformation routines available in directory *transformation*. The directory *magneticfield* provides the source code for the calculation of the Earth's magnetic field. The source code for the ephemeris calculation is placed in the *ephemeris* directory. The folder *atmospheric_drag* provides source code for the atmospheric model. The *environment* folder will be updated if new environment models are available. In folder *disturbances* pre-processing tools for environmental data are provided (for details cf. [1] and [2]). The *run* folder (cf. figure 4) contains the Matlab/Simulink environment for mission simulation. The interface between Matlab/Simulink and the dynamics core is described in chapter 6. The general interface structure is outlined in the next chapter.

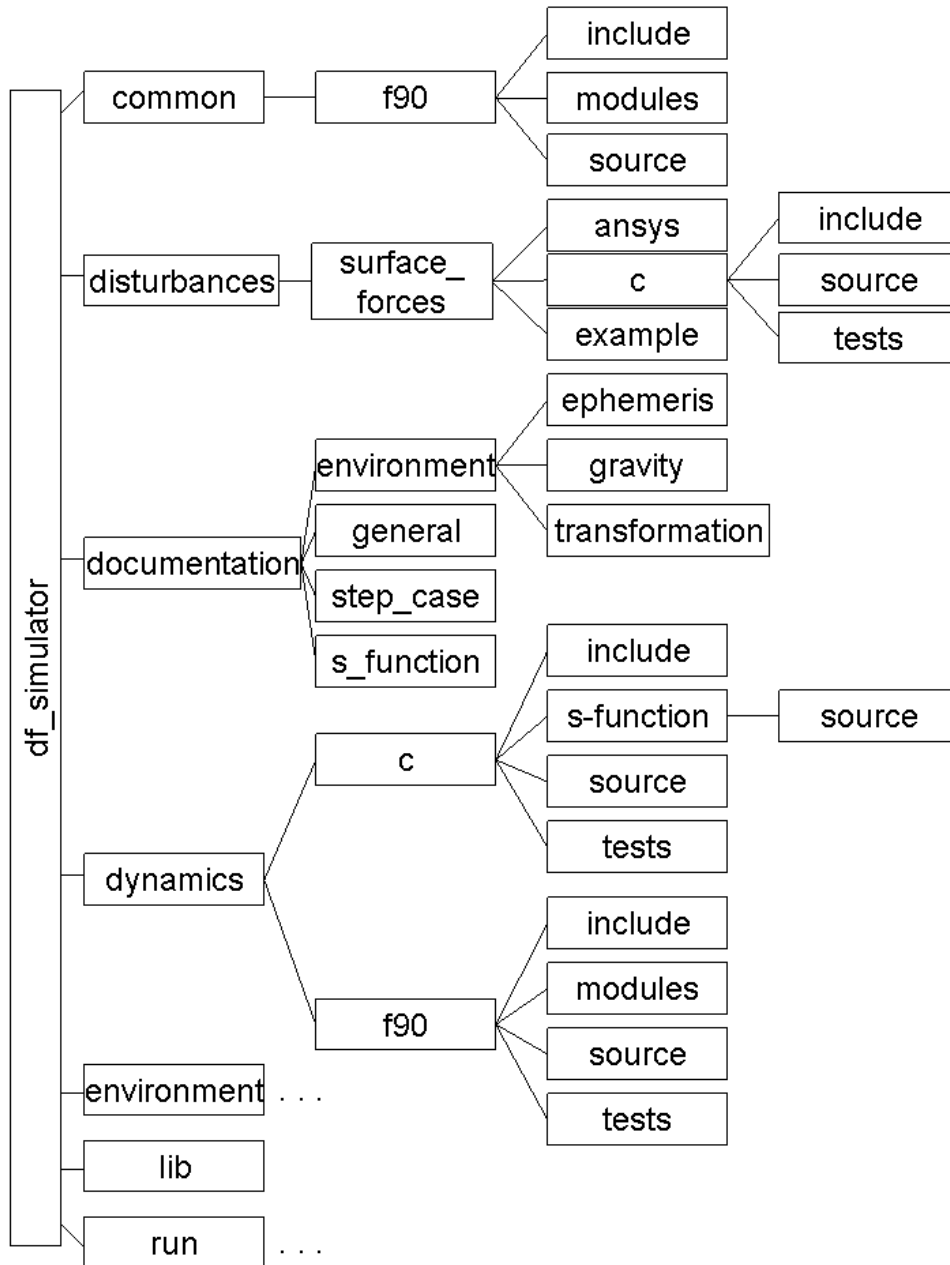


Figure 2: DF Simulator Directory Structure

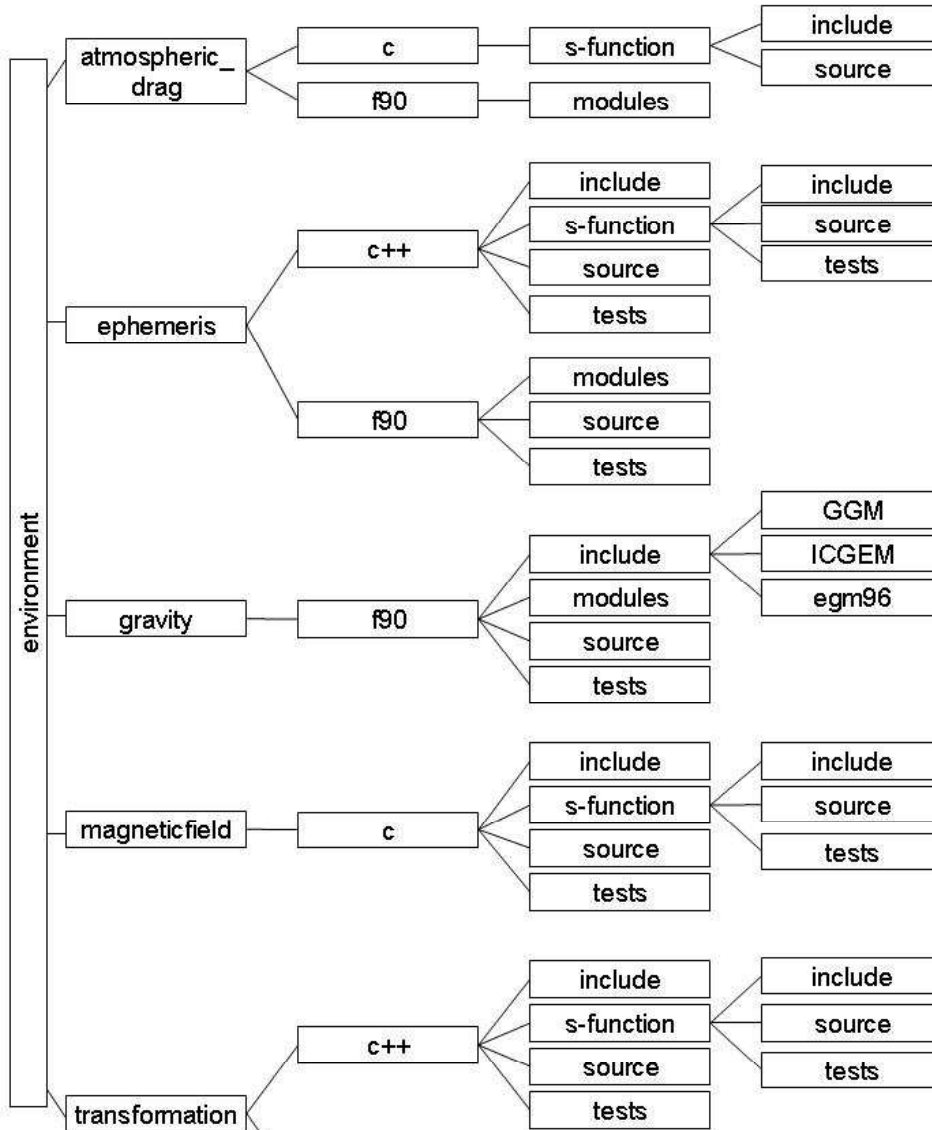


Figure 3: Environment Directory Structure

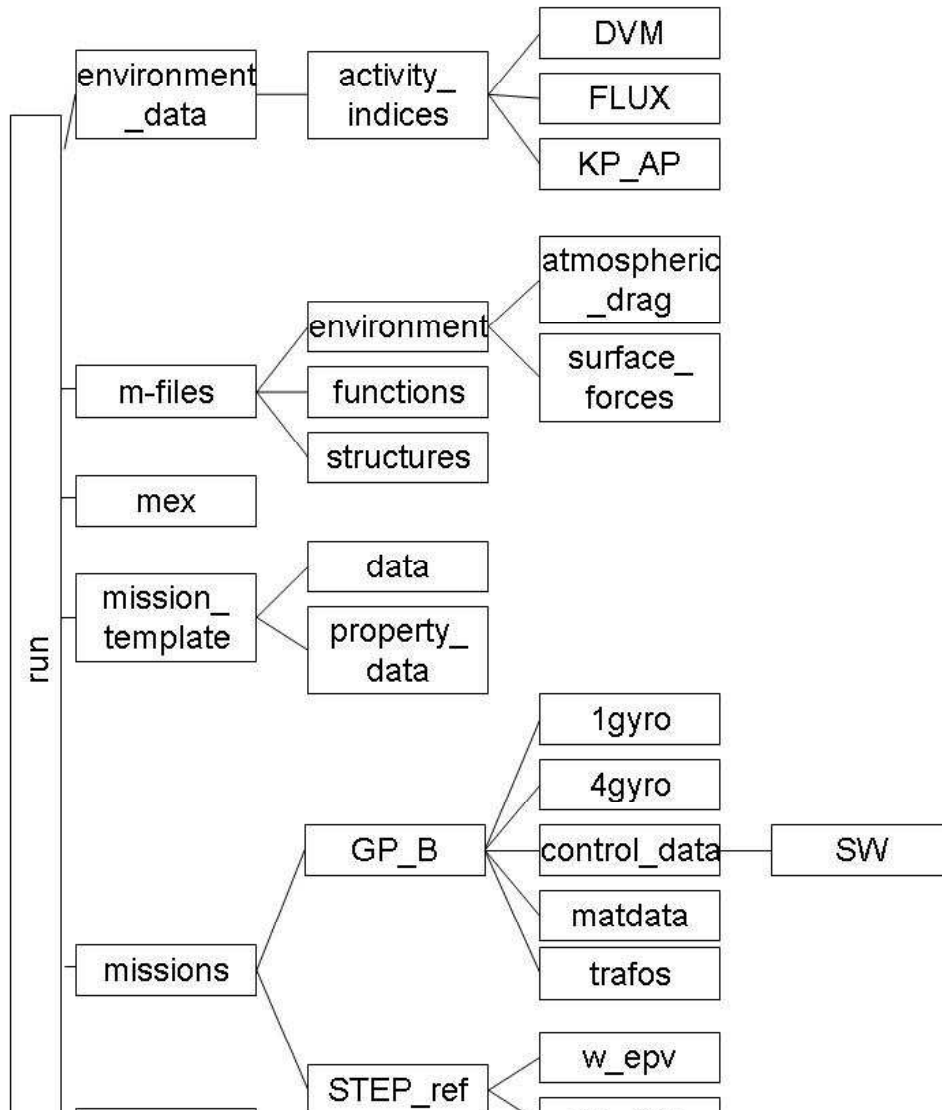


Figure 4: Run Directory Structure

5 Interface Definition

The DF simulator interface should have an initialization, output, update and termination part. The Matlab/Simulink interface is represented by an S-function. Within the S-function, inputs, parameters, states and outputs have to be stored in structures. The structures are passed to void functions for initialization etc. as follows:

- Init (Parameters,States)
- Output(Inputs,Parameters,States,Outputs)
- Update(Inputs,Parameters,States)
- Terminate(void)

It is self-explanatory that function Init shall contain all initialization necessary for parameters and states, Output defines the output variables and Terminate is called to carry out code termination actions. The actual computation is done in function Update.

The simulation module can be provided as a Matlab/Simulink library. An example is shown in figure 5.

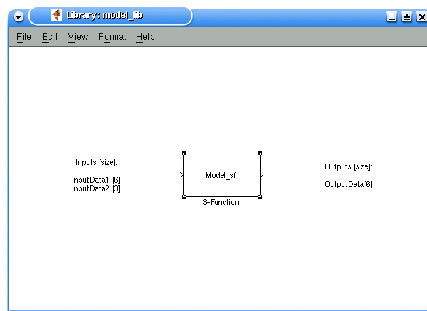


Figure 5: Simulink Model Library

The parameter and state structs are passed over as S-function parameters. Within Matlab, only the parameters and states are stored in structures. Internally, i.e. inside the

S-function, also inputs and outputs are handled as structures. For an S-function written in C code, a header file must be provided in which the interface functions and structures are declared. An example header file could look like this:

```
#ifndef _Interface_
#define _Interface_

#include "includes.h"

// Parameters
struct Parameters
{
    double parameter1;
    double parameter2;
    ...
};

typedef struct Parameters Parameters;

// Inputs
struct Inputs
{
    double input1;
    ...
};

typedef struct Inputs Inputs;

// Outputs
struct Outputs
{
    double output1;
    ...
};

typedef struct Outputs Outputs;
```

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 19 of 30
---	---------------------------------	---

```
// States
struct States
{
    double states;
};

typedef struct States States;

// Prototypes

extern void Init(struct Parameters*, struct States*);
extern void Output(struct Inputs*, struct Parameters*, struct States*, struct Outputs*);
extern void Update(struct Inputs*, struct Parameters*, struct States*);
extern void Terminate(void);

#endif /* _Interface_ */
```

6 Interface for the Drag-Free Simulator Core

The interface *df_simulator_sf.c* for the drag-free simulator core calls the following functions:

- Core_Init (CoreParam,CoreState)
- Core_Output(CoreIn,CoreParam,CoreState,CoreOut)
- Core_Update(CoreIn,CoreParam,CoreState)
- Core_Terminate()

6.1 Matlab/Simulink Simulation Environment

Directory Structure

The directory structure for a Matlab/Simulink simulation is the following:

(DRAG_FREE_SIMULATOR_HOME)/run/

with subfolders *environment_data*, *m-files* (containing *functions*, *structures* and *environment*), *mex*, *simulink*, *missions*, *mission_template*, and *tools*. Folder *mex* contains the s-functions needed for a simulation, *simulink* provides libraries for model build, in *missions* satellite applications are placed like the STEP reference case, *mission_template* contains everything needed to start a simulation (the how-to is lined out in chapter 7) and *tools* contains related non-Matlab utilities. In folder *environment_data* coefficient and other environmental data files are stored. The Matlab file *start_df_sim* sets the scenario path for the data files to be loaded, initializes the structures and opens the Simulink file for the simulation. It should be placed at the top directory of every mission, e.g. it can be found under *missions/STEP_ref* and the default is placed in *mission_template*. Among the Matlab files in *m-files* is the m-file *load_df_sim_params* which loads the simulation specific parameter data from m-file *load_sim_data*, the satellite specific data from *load_sat_data* and accelerometer data from files *load_acc_data_[i]*, $i=1..4$, also placed in subfolder *data*. The default *load_sim_data.m*, *load_sat_data.m* and *load_acc_data_[i].m*, $i=1..4$ are stored in subfolder *mission_template/data*. In subfolder *mission_template/property_data* example data for geometry or other properties are stored. This data is generated by pre-processing tools, e.g. those in (DRAG_FREE_SIMULATOR_HOME)/disturbances.

Matlab Structures

Compared to previous versions of the DF simulator the S-function for the simulator core is reorganized to conform with the current guidelines established for interface construction. For the same reason the Matlab structures for the drag-free simulator core are

reshaped.

Since the structure definition within the Matlab simulation files up to version v-1-3 is not consistent with the structures defined above, several changes are introduced. These changes also address the inconsistency in terminology prior to version v-1-4: Satellite parameters should have been gathered in a satellite structure whereas they are saved as satellite structure parameters and also as simulation structure parameters. An example for that is *df_sim_struct.d_mass* which should have been *satellite_df_struct.d_mass* for constant mass. Recently the drag-free simulator has been updated to include the option to introduce a varying mass due to fuel consumption etc. (see [5]) such that the satellite mass becomes a time-variable input and should be placed in the input structure.

The tables below summarize the parameters in the data files. For comparison, also the notation used in former versions (until version v-1-3) is given.

Parameters in *load_sim_data*:

<i>v-1-4 and higher versions</i>	<i>previous versions</i>
<i>df_sim.const_links</i>	<i>df_sim_struct.d_const_links</i>
<i>acc_opt.acc_id</i>	<i>acc_opt.acc_id</i>
<i>df_core_param.d_MJD</i>	<i>df_sim_struct.d_MJD</i>
<i>df_core_param.d_t_steps</i>	<i>df_sim_struct.d_t_step</i>
<i>df_core_param.d_eps_dp</i>	<i>df_sim_struct.d_eps_dp</i>
<i>df_core_param.d_h_min_dp</i>	<i>df_sim_struct.d_h_min_dp</i>
<i>df_core_param.d_h_1_dp</i>	<i>df_sim_struct.d_h_1_dp</i>
<i>df_core_param.i_option_g</i>	<i>df_sim_struct.i_option_g</i>
<i>df_core_param.i_option_g_tide</i>	N/A
<i>df_core_param.i_option_g_order</i>	<i>df_sim_struct.i_option_g_order</i>
<i>df_core_param.i_option_g_degree</i>	<i>df_sim_struct.i_option_g_degree</i>
<i>df_core_param.i_option_g_cog</i>	<i>df_sim_struct.i_option_g_cog</i>
<i>df_core_param.i_option_g_planets</i>	<i>df_sim_struct.i_option_g_planets</i>
<i>df_core_param.i_option_gg_model</i>	<i>df_sim_struct.i_option_gg_model</i>
<i>df_core_param.i_option_ggt</i>	<i>df_sim_struct.i_option_ggt</i>
<i>df_core_param.i_option_gg_acc</i>	<i>df_sim_struct.i_option_gg_acc</i>
<i>df_core_param.i_option_ggt_tm</i>	<i>df_sim_struct.i_option_ggt_tm</i>
<i>df_core_param.i_option_refsys</i> (only used up to v-2-2)	N/A
<i>df_core_param.i_option_intmeth</i>	N/A

The parameter *df_sim.const_links* is fed in directly as parameter to the s-function block like *acc_opt.acc_id*. The value of *df_sim.const_links* determines whether the coupling

forces and torques between satellite and test masses are constant or changing with time. The original option to input variable spring stiffness, damping and DC forces and torques is eliminated. Instead forces da_F_trans and torques da_T_rot can be input to model e.g. nonlinear coupling. These inputs are internally added to the DC parts of the coupling forces and torques.

Parameters in *load_sat_data*:

<i>v-1-4 and higher versions</i>	<i>previous versions</i>
	df_sim_struct.d_mass
	df_sim_struct.da_moi
	df_sim_struct.da_r_m_mb
df_core_param.d_num_acc	satellite_df_struct.d_num_acc
df_core_param.i_option_sat_dof	df_sim_struct.i_option_sat_dof
df_core_state.da_y	df_sim_struct.da_y

The inputs d_mass , da_moi and $da_r_m_mb$ are fed in directly, e.g. as inputs through Simulink Constant blocks. In the S-function they are placed in the input structure.

Parameters in *load_acc_data_1* (dito for *load_acc_data_2,3,4*):

<i>v-1-4 and higher versions</i>	<i>previous versions</i>
df_core_state.da_y	df_sim_struct.da_y
df_core_param.s_accelero1.d_num_tm	satellite_df_struct.s_accelero1.d_num_tm
df_core_param.s_accelero1... da_r_m_ma	satellite_df_struct.s_accelero1... da_r_m_ma
da_T_a_b	da_T_a_b
sa_test_mass1(2).d_mass	sa_test_mass1(2).d_mass
sa_test_mass1(2).d_eta	sa_test_mass1(2).d_eta
sa_test_mass1(2).d_moi	sa_test_mass1(2).d_moi
sa_test_mass1(2).d_moi_inv	sa_test_mass1(2).d_moi_inv
sa_test_mass1(2).da_r_a_asens	sa_test_mass1(2).da_r_a_asens
sa_test_mass1(2).da_T_sens_a	sa_test_mass1(2).da_T_sens_a
sa_test_mass1(2).da_switch_trans_DOF	sa_test_mass1(2).da_switch_trans_DOF
sa_test_mass1(2).da_switch_rot_DOF	sa_test_mass1(2).da_switch_rot_DOF
sa_link1(2,3).da_k_trans	sa_link1(2,3).da_k_trans
sa_link1(2,3).da_kr_trans	sa_link1(2,3).da_kr_trans
sa_link1(2,3).da_d_trans	sa_link1(2,3).da_d_trans
sa_link1(2,3).da_dr_trans	sa_link1(2,3).da_dr_trans
sa_link1(2,3).da_f_dc_trans	sa_link1(2,3).da_f_dc_trans
sa_link1(2,3).da_r_offset	sa_link1(2,3).da_r_offset
sa_link1(2,3).da_k_rot	sa_link1(2,3).da_k_rot
sa_link1(2,3).da_kt_rot	sa_link1(2,3).da_kt_rot
sa_link1(2,3).da_d_rot	sa_link1(2,3).da_d_rot
sa_link1(2,3).da_dt_rot	sa_link1(2,3).da_dt_rot
sa_link1(2,3).da_t_dc_rot	sa_link1(2,3).da_t_dc_rot

An example Simulink model is shown in figure 6. The first two inputs are external forces and torques expressed in the inertial, orbital and mechanical reference frame (see [5] for a definition of reference frames and [8] for a more detailed input description). The third Simulink block provides satellite mass, moments of inertia and the vector specifying the coordinates of the mechanical reference frame w.r.t. the satellite body-fixed frame (originating in the satellite's center of mass). The ECI to ECEF transformation matrix, here calculated by the *ECI2ECEF_matrix_sf* block, is supplied by input port number four. The fifth block contains the *ephemeris_sf* which provides the position of the celestial bodies. Finally, for each accelerometer (only one is present in this example) links specifying coupling forces and torques can be input (this example has only one test mass, therefore three force and three torque vector components can be input). This input can be used to model non-linear coupling interaction.

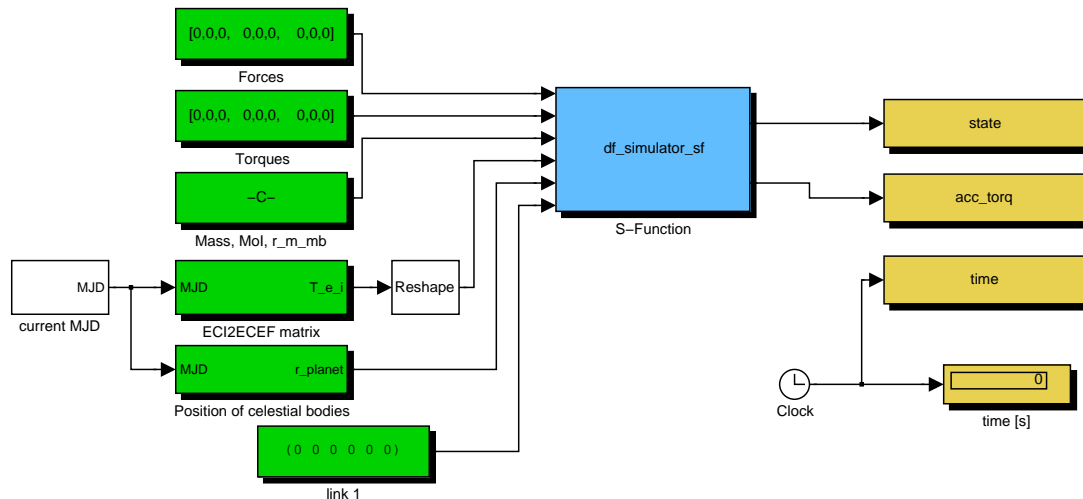


Figure 6: Simulink Model

6.2 Interface Layout

As described in the beginning of this chapter the interface *df_simulator_sf* calls the void functions *CORE_Init*, *CORE_Output*, *CORE_Update* and *CORE_Terminate*.

6.3 Wrapper for the Core

The interface defined previously has been introduced to provide a common structure for all modules included in a simulation. However it is not the purpose to re-code the drag-free simulator core module itself. After reshaping of the Matlab structures the core interface cannot address the core directly anymore. A wrapper has been coded to connect the interface with the core. In this wrapper the interface structures are assigned to the core structures and variables. The interface *df_simulator_sf* calls the void functions *CORE_Init*, *CORE_Output*, *CORE_Update* and *CORE_Terminate* within the wrapper *df_simulator_front.c*. The wrapper calls the Fortran core routines *df_sim_init* and *df_sim_loop*. The wrapper functions are described below.

CORE_Init(CoreParam,CoreState)

In this function the core parameter structures and states handed over to the core function *df_sim_init* are initialized. In previous versions of the DF simulator (up to v-1-3) some variables belonging to the input structure were initialized in *df_sim_init*. Since the *CoreIn* struct is not passed over to *CORE_Init*, and since these variables are not needed in

	The Modular Drag-Free Simulator	Doc.No.: FLK-SIM-TN-ZAR-006 Issue: 1.6 Page: 25 of 30
---	---------------------------------	---

CORE_Output (see below) the initialization is not necessary at this point since it is done for every time step in CORE_Update.

CORE_Output(CoreIn,CoreParam,CoreState,CoreOut)

in fact: CORE_Output(CoreState,CoreOut)

This function is called after CORE_Init. It only assigns the states within CoreState to the output CoreOut.

CORE_Update(CoreIn,CoreParam,CoreState)

This function calls df_sim_loop, the core function that calls the EoM (equations of motion) integrator. The interface parameters, states and inputs within CoreIn, CoreParam and CoreState are assigned to the core parameters, states and inputs. In this routine the externally provided coupling forces and torques are added to the DC coupling forces and torques for every time step.

CORE_Terminate()

This function is a void function.

7 DF Simulator Application

To start a mission including the DF Simulator modules, a new subdirectory can be placed in the missions directory. In the *mission_template* directory the Matlab files necessary for initialization and set-up of parameters are included. In the form provided they can be copied directly to the new mission and modified where appropriate. The modification options for the Matlab script parameters are described below. With the help of the Simulink libraries provided in directory *df_simulator/run/simulink* a new Simulink model file can be created easily. The new model can be loaded through parameter *df_simulator_model* in the input data file *load_sim_data.m*. The file *start_df_sim.m* initializes the simulation. In this file all paths used for a simulation need to be entered.

NOTE: The functions and structure definitions in the *m-files* directory should never be changed.

The *STEP_ref* simulation in the *missions* folder serves as a reference case for the STEP test mass dynamics. It can also be studied to get familiarized how the DF simulator core module is used. The STEP reference simulation is documented in [9]. The corresponding Simulink model includes the s-function blocks for the calculation of the transformation matrix from ECI to ECEF and the ephemeris of the celestial bodies. These blocks are only added to illustrate their usage. As this information is not needed for the STEP reference case, it would be sufficient to provide zero-vectors to the s-function input ports.

A description of the DF simulator core S-function can be found in [8]. For information on the transformation library and the ephemeris library see [3] and [4].

7.1 Matlab Script Options - Dynamics Core

The options *df_core_param.i** in the matlab script files that can be modified can be grouped in satellite, gravity and further simulator options.

Satellite Options

- **i_option_sat_dof:** Option for choosing which DOFs of the satellite will be fixed and which will be free and thus integrated:
 - I_ALL_FREE: all satellite DOFs are free
 - I_X_Y_TRANS_Z_ROT: only x and y are free, z is fixed (equatorial orbit), rotation around z only
 - I_NO_TRANS_ALL_ROT: no translational movement, all rotational DOFs free
 - I_NO_TRANS_X_ROT: no translational movement, only rotation around x

- I.NO_TRANS_Y_ROT: no translational movement, only rotation around y
- I.NO_TRANS_Z_ROT: no translational movement, only rotation around z
- I.NONE_FREE: all satellite DOFs are fixed

Gravity Options

- **i.option_g:** Option for choosing the gravity model applied in the simulation to calculate the acceleration due to gravity forces:
 - I.G_OPT_SIMPLE: This option uses a simplified model of the gravitational field up to degree 6 (J_n -terms of equation (1)).
 - I.G_OPT_EGM: full Earth Gravity Model (EGM) from gravitational potential function:

$$\Phi = \frac{\mu_E}{r} \left[1 + \sum_{n=2}^{\infty} \left(\frac{R_E}{r} \right)^n J_n P_{n0}(\sin\Theta) \right] + \frac{\mu_E}{r} \sum_{n=2}^{\infty} \sum_{m=1}^n \left(\frac{R_E}{r} \right)^n (C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)) P_{nm}(\sin\Theta) \quad (1)$$

with:

R_E	Earth radius.
r	Position vector.
J_n	Earth zonal harmonic coefficients of degree n .
P_{nm}	Legendre polynomials of degree n and order m .
Θ	Geocentric latitude.
C_{nm}, S_{nm}	Tesseral harmonic coefficients for $n \neq m$, sectoral harmonic coefficients for $n = m$.
λ	Geocentric longitude.

- I.G_OPT_GFZ: Model up to degree 360 based on GRACE data from GFZ described in [6]
- I.G_OPT_CSR: Model up to degree 360 based on GRACE data from CSR described in [6]
- **i.option_g_tide:** Option for global gravity model specifying in which tide system the spherical harmonic coefficients shall be provided.
 - I.TIDE_FREE: In the tide-free system all tidal effects are removed (EGM96 default).
 - I.TIDE_ZERO: In the zero-tide system the permanent tidal deformation due to external bodies is retained. This affects the C_{20} coefficient only.

- **i_option_g_degree:** Option setting the degree of the gravity model. This option can be set from 1 to 360.
- **i_option_g_order:** Option setting the order of the gravity model. This option can be set from 0 to 6 if I_G_OPT_SIMPLE is used and from 0 to i_option_g_degree if I_G_OPT_EGM, I_G_OPT_GFZ or I_G_OPT_CSR is used.
- **i_option_g_planets(1:10):** Option switching on(1)/off(0) the effects of Sun, Moon and planets on the gravitational acceleration (i_option_g_planets(1): Sun, (2): Mercury, (3): Venus, (4): Mars, (5): Jupiter, (6): Saturn, (7): Uranus, (8): Neptune, (9): Pluto, (10): Moon). This option is only active if I_G_OPT_EGM, I_G_OPT_GFZ or I_G_OPT_CSR is used.
- **i_option_g_cog:** Option switching on(1)/off(0) the correction for the gravity force acting in the satellite's center of gravity (COG) instead of its center of mass (COM).
- **i_option_gg_model:** Option for choosing the model to be applied for calculation of the gravity-gradient matrix and the gravity-gradient induced forces and torques. For the latter only options I_GG_OPT_SPHERE and I_GG_OPT_J2 apply. If I_GG_OPT_EGM, I_G_OPT_GFZ or I_G_OPT_CSR is chosen, the calculation will be carried out as if I_GG_OPT_J2 was set.
The gravity-gradient matrix can be used alternatively for the computation of the gravity-gradient torque and the gravity-gradient acceleration between the COMs of satellite and test masses. Here all options apply. However, the gravity-gradient matrix is calculated by linearizing the gravitational field at the center of mass of the satellite.
 - I_GG_OPT_SPHERE: gravity-gradient matrix based on a spherical field.
 - I_GG_OPT_J2: gravity-gradient matrix including the J_2 zonal harmonic term of the gravitational potential (1).
 - I_GG_OPT_EGM: gravity-gradient matrix of the Earth Gravity Model. In this case the option i_option_g must be set to I_G_OPT_EGM, I_G_OPT_GFZ or I_G_OPT_CSR.
- **i_option_ggt:** Option for choosing the gravity model applied in the simulation for gravity gradient torques:
 - I_GGT_OPT_OFF: no gravity-gradient torque is calculated.
 - I_GGT_OPT_SIMPLE: gravity-gradient torque calculated from spherical potential if I_GG_OPT_SPHERE is used, including J_2 -term if I_GG_OPT_J2 is used.

- I_GGT_OPT_MATRIX: gravity-gradient torque calculated using the gravity-gradient matrix chosen by option `i_option_gg_model`.
- **i_option_gg_acc:** Option for choosing the way to compute the gravity gradient acceleration between test mass and satellite:
 - I_GG_ACC_OPT_OFF: no gravity-gradient acceleration is calculated.
 - I_GG_ACC_OPT_SIMPLE: difference of accelerations from spherical gravitational potential
 - I_GG_ACC_OPT_EGM: difference of accelerations from Earth Gravity Model. In this case the option `i_option_g` must be set to `I_G_OPT_EGM`, `I_G_OPT_GFZ` or `I_G_OPT_CSR`.
 - I_GG_ACC_OPT_SPHERE: difference of accelerations from spherical gravitational potential expanding fractions to avoid numerical problems
 - I_GG_ACC_OPT_LIN: The gravity-gradient acceleration is computed by linearizing the gravity field at the satellite COM using the gravity-gradient matrix.

NOTE: Use last two options (`I_GG_ACC_OPT_SPHERE`, `I_GG_ACC_OPT_LIN`) on 32-bit machines to avoid numerical problems!

- **i_option_ggt_tm:** Option for choosing the way to compute the gravity gradient torque for the test masses:
 - I_GGT_OPT_OFF: no gravity-gradient torque is calculated.
 - I_GGT_OPT_SIMPLE: gravity-gradient torque calculated from spherical potential if `I_GG_OPT_SPHERE` is used, including J_2 -term if `I_GG_OPT_J2` is used.
 - I_GGT_OPT_MATRIX: gravity-gradient torque calculated using the gravity-gradient matrix chosen by option `i_option_gg_model`.

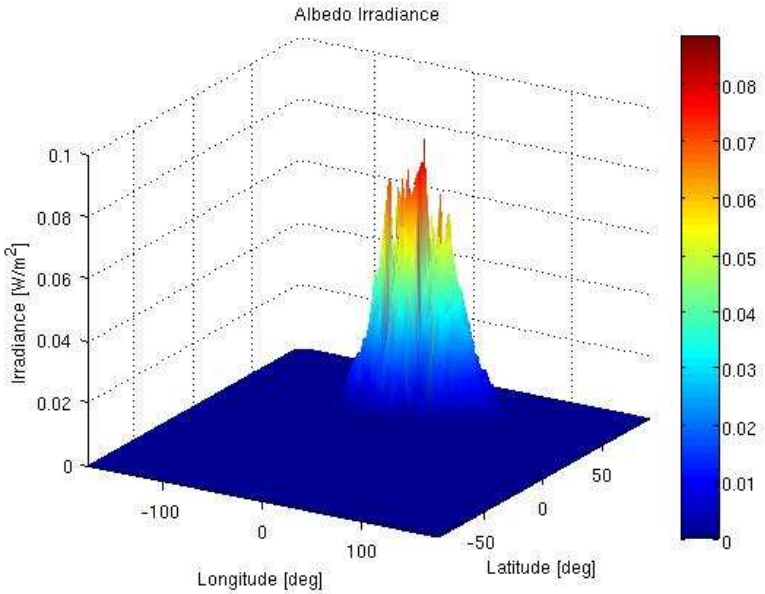
Further Simulator Options

- **i_option_intmeth:** Option for choosing the integration method
 - I_INT_RK: 5th order Runge-Kutta
 - I_INT_BS: Bulirsch-Stoer
 - I_INT_EUL: Euler-Cauchy (for small step size only!)

References

- [1] S. Bremer. Modeling of solar radiation force and torque. Technical Report FLK-SIM-TN-ZAR-009, ZARM, University of Bremen, 2008.
- [2] S. Grotjan. Modeling of surface forces - Tutorial. Technical Report FLK-SIM-TU-ZAR-001, ZARM, University of Bremen, 2006.
- [3] S. Grotjan. Transformation Library. Technical Report FLK-SIM-TN-ZAR-005, ZARM, University of Bremen, 2006.
- [4] R. Keil and I. Pelivan. Ephemeris Calculation in the Drag-Free Simulator. Technical Report FLK-SIM-TN-ZAR-008, Center of Applied Space Technology and Microgravity (ZARM), April 2007.
- [5] I. Pelivan. Equations of Motion for Satellite and Test Mass Dynamics. Technical Report FLK-SIM-TN-ZAR-001, ZARM, University of Bremen, 2005.
- [6] I. Pelivan. Earth Gravity Model Update for the Generic Drag-Free Simulator. Technical Report FLK-SIM-TN-ZAR-007, Center of Applied Space Technology and Microgravity (ZARM), February 2007.
- [7] S. Scheithauer. Earth Gravity Model 96 – Implementation. Technical Report SIM-ENV-TN-ZAR-001, ZARM, University of Bremen, 2006.
- [8] S. Theil. Generic Drag-Free Simulator: S-Function Description. Technical Report SIM-DYN-TN-ZAR-002, ZARM, University of Bremen, 2006.
- [9] S. Theil. STEP Reference Simulation Case for the Generic Drag-Free Simulator. Technical Report STP-SIM-TN-ZAR-003, ZARM, University of Bremen, 2006.

The Earth Albedo Model



Project: FIRST LOOK	Document No.: FLK-ENV-RP-ZAR-001									
<p>ZARM –Center of Applied Space Technology and Microgravity</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 40%;">Am Fallturn</td> <td style="width: 20%;">Phone:</td> <td style="width: 40%;">+49-421-218-3551</td> </tr> <tr> <td>D28359 Bremen</td> <td>Fax:</td> <td>+49-421-218-4356</td> </tr> <tr> <td>Germany</td> <td>E-Mail:</td> <td>evandro@spacetec.zarm.uni-bremen.de evandro@dem.inpe.br</td> </tr> </table>		Am Fallturn	Phone:	+49-421-218-3551	D28359 Bremen	Fax:	+49-421-218-4356	Germany	E-Mail:	evandro@spacetec.zarm.uni-bremen.de evandro@dem.inpe.br
Am Fallturn	Phone:	+49-421-218-3551								
D28359 Bremen	Fax:	+49-421-218-4356								
Germany	E-Mail:	evandro@spacetec.zarm.uni-bremen.de evandro@dem.inpe.br								

Doc. No.:	ATP-NAV-TN-ZAR-005	Date:	June 20 th 2008
Issue:	1.0	Date:	
Written:	Evandro Marconi Rocco	Date:	
Approved:		Date:	
Approved:		Date:	–

Table of Contents

1 Introduction.....	4
2 The Earth Albedo Mathematical Model.....	5
3 Implementation of the Model.....	9
3.1 Description of the Sub-Systems	9
3.2 Input Variables.....	12
3.3 Output Variables.....	13
3.4 Variables Saved in the Workspace.....	14
4 Some Results Provided by the Earth Albedo Model.....	15
4.1 The Reflectivity of the Earth Surface.....	15
4.2 The Full Albedo Irradiance.....	16
4.3 The Instantaneous Albedo Irradiance.....	17
4.4 The Incident Angle at the Grid Point.....	18
4.5 The Incident Angle at the Satellite.....	19
4.6 The Albedo Irradiance at the Satellite.....	20
4.7 The Albedo Vector Centered in the Satellite Position.....	21
4.8 Satellite Illumination Condition for the Entire Day.....	21
4.9 Number of Illuminated Visible Cells for the Entire Day.....	22
4.10 Total Albedo for the Entire Day.....	22
5 Results Obtained Considering Different Resolutions.....	23
5.1 Reduction Factor 1.....	24
5.2 Reduction Factor 2.....	26
5.2.1 Reduction Factor 1 x Reduction Factor 2.....	28
5.3 Reduction Factor 3.....	29
5.3.1 Reduction Factor 1 x Reduction Factor 3.....	31
5.4 Reduction Factor 4.....	32
5.4.1 Reduction Factor 1 x Reduction Factor 4.....	34
5.5 Reduction Factor 5.....	35
5.5.1 Reduction Factor 1 x Reduction Factor 5.....	37
5.6 Reduction Factor 6.....	38
5.6.1 Reduction Factor 1 x Reduction Factor 6.....	40
5.7 Reduction Factor 7.....	41
5.7.1 Reduction Factor 1 x Reduction Factor 7.....	43
5.8 Reduction Factor 8.....	44
5.8.1 Reduction Factor 1 x Reduction Factor 8.....	45
5.9 Reduction Factor 9.....	47
5.9.1 Reduction Factor 1 x Reduction Factor 9.....	49
5.10 Reduction Factor 10.....	50
5.10.1 Reduction Factor 1 x Reduction Factor 10.....	52
5.11 Reduction Factor 15.....	53
5.11.1 Reduction Factor 1 x Reduction Factor 15.....	55
5.12 Reduction Factor 20.....	56

5.12.1	Reduction Factor 1 x Reduction Factor 20.....	58
5.13	Directions of the Albedo Vector Considering Different Resolutions.....	59
6	Results Obtained Considering Different Altitudes	59
6.1	The Full Albedo Irradiance for the Altitude Around 100 km.....	60
6.2	The Full Albedo Irradiance for the Altitude Around 200 km.....	60
6.3	The Full Albedo Irradiance for the Altitude Around 300 km.....	61
6.4	The Full Albedo Irradiance for the Altitude Around 500 km.....	61
6.5	The Full Albedo Irradiance for the Altitude Around 700 km.....	62
6.6	The Full Albedo Irradiance for the Altitude Around 1000 km.....	62
6.7	The Full Albedo Irradiance for the Altitude Around 36000 km.....	63
7	Evaluation of the Terrestrial Albedo Applied to Some Scientific Missions...	63
7.1	Gravity Probe B.....	63
7.1.1	Evaluation of the Trajectory Deviation for the GP-B.....	69
7.2	Microscope Mission.....	72
7.2.1	Evaluation of the Trajectory Deviation for the Microscope Mission.....	78
7.3	Step Mission.....	81
7.3.1	Evaluation of the Trajectory Deviation for the Step Mission.....	87
8	Conclusion.....	90
	References.....	91

1- Introduction

Albedo is the fraction of solar energy reflected diffusely from the planet back into space. It is a measure of the reflectivity of the planet’s surface. Thus, the Earth albedo can be defined as the fraction of the incident solar radiation returned to space from Earth’s surface.

$$albedo = \frac{radiation\ reflected\ back\ to\ space}{incident\ radiation}$$

The range of possible values for the albedo is from 0 (completely dark) to 1 (completely bright).

The Earth’s surface can be considered as a diffusely reflecting surface. The reflecting radiation represents approximately 34% of the incident radiation. However, this factor depends on the surface conditions. Ice, especially with snow on top of it, has a high albedo. Most sunlight hitting the ice surface bounces back towards space. Water is much more absorbent and less reflective. So, if there is a lot of water, more solar radiation is absorbed by the ocean than when ice dominates. Following there are some usual values of the albedo related to the surface's condition.

Clouds:0.4- 0.8 green areas: 0.05 - 0.25 average planetary: 0.34

The percentage of diffusely reflected Sun light in relation to various surface conditions of the Earth surface is shown in the next figure.

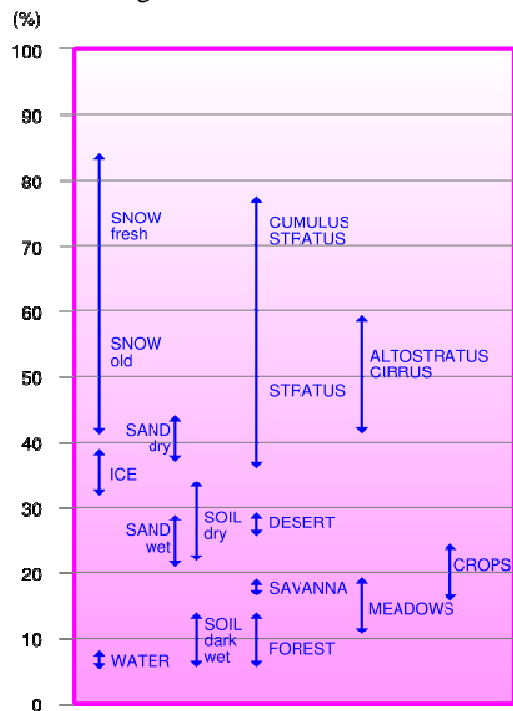


Fig. 1 – Percentage of diffusely reflected Sun light.

The effect of the albedo in the satellite depends on the following factors:

- Position of the satellite;
- Attitude of the satellite;
- Optical properties of the satellite surfaces:
 - incident radiation is absorbed;
 - incident radiation is reflected specularly;
 - incident radiation is reflected diffusely;
 - some combination of absorbed, specularly and diffusely cases.

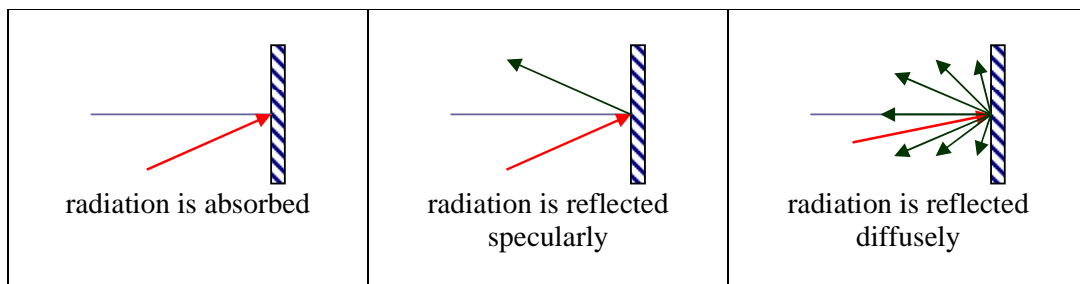


Fig. 2 - Optical properties of the surfaces.

2- The Earth Albedo Mathematical Model

Solar radiation includes all electromagnetic waves emitted by the Sun. Solar radiation falling at right angles on an area of 1 m^2 at a solar distance of 1 AU ($149598200 \pm 500 \text{ km}$) is $1371 \pm 5 \text{ W/m}^2$, which is called solar constant. The solar radiation for a distance d is given by:

$$J_s = \frac{P}{4\pi d^2}$$

where P is the total power output from the Sun ($.8 \times 10^{25} \text{ W}$) and d is the distance from the Sun.

The model of the Earth albedo is based on the reflectivity data measured by NASA's Earth Probe satellite, which is part of the TOMS project (Total Ozone Mapping Spectrometer). In this project, the reflectivity data is available daily, on line at TOMS website: <http://toms.gsfc.nasa.gov/>.

The reflectivity data fluctuates because of changes in clouds and ice coverage and seasonal changes. The data resolution partitions the Earth surface into a number of cells. The incident irradiance on each cell is used to calculate the total radiant flux from the cell. From the radiant flux from each cell, the irradiance at the satellite is calculated using a model based in the model developed by Bhanderi, 2005.

The satellite data is given in a resolution of $\Delta\phi_g = 1 \text{ deg}$ in latitude times $\Delta\theta_g = 1.25 \text{ deg}$ in longitude (180×288 data points) that result in 51840 cells in the Earth's surface.

The two-dimensional data space D is defined as a grid of data points $\Phi \times \Theta$, where Φ represents the latitude and Θ the longitude:

$$D = \begin{cases} \Phi = (0, \Delta\phi_g, 2\Delta\phi_g, \dots, 179\Delta\phi_g) \\ \Theta = (0, \Delta\theta_g, 2\Delta\theta_g, \dots, 287\Delta\theta_g) \end{cases}$$

To each data point (ϕ_g, θ_g) in the data space D , the mean reflectance of a cell, $\phi_g \pm \phi_g/2$ and $\theta_g \pm \theta_g/2$, on the Earth surface is available in the TOMS data product. The reflectivity of a cell is calculated by measuring the irradiance received E_{meas} by the satellite. The irradiance measurement E_{meas} is modeled by:

$$E_{meas} = \frac{\rho f}{1 - S\rho} E_{dd} + E_{dd0}$$

where:

- ρ is the reflectivity of the reflecting surface;
- f is the fraction of reflected irradiation reaching the satellite;
- S is the fraction of reflected irradiance scattered back to the reflecting surface;
- E_{dd} is the amount of direct and diffuse irradiance reaching the reflecting surface;
- E_{dd0} is the amount of atmospheric scattered irradiance reaching the satellite.

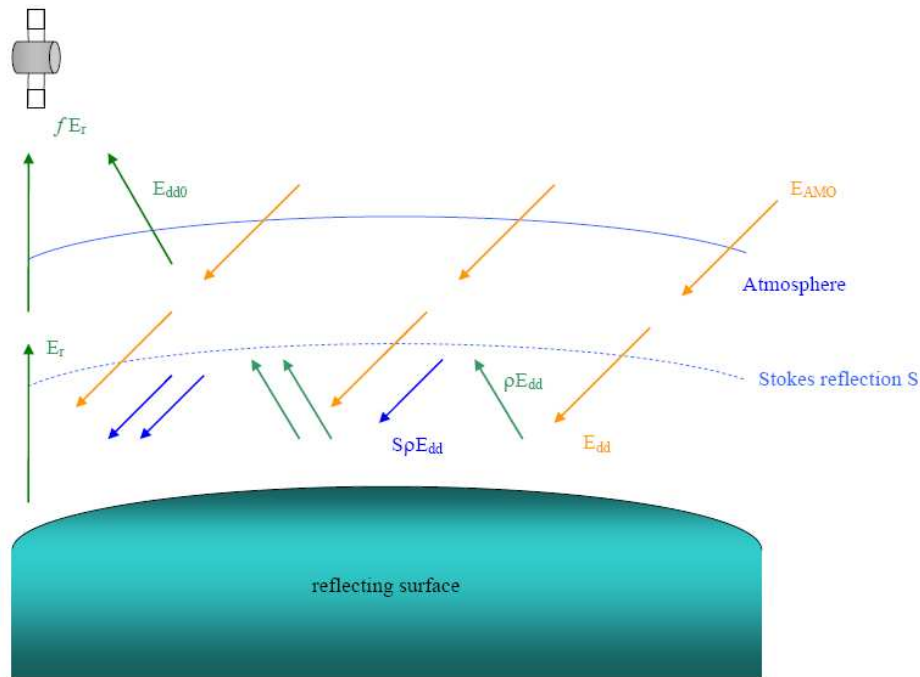


Fig. 3 – Irradiance reflected towards the satellite.

The total irradiance reaching the surface is the sum of E_{dd} and the infinite sum of irradiance reflected between the atmosphere and the surface.

$$E_r = \rho E_{dd} (1 + S\rho + S^2\rho^2 + \dots + S^n\rho^n + \dots) = \frac{\rho E_{dd}}{1 - f\rho}$$

Due to atmospheric scattering and absorption only a fraction f of the reflected irradiation reaches the satellite. The incident solar irradiance E_{AMO} (measured considering air mass zero) reaches the cell at grid point (ϕ_g, θ_g) , at an incident angle ϕ_m to the cell normal. The amount of radiant flux reflected by the cell is given by the irradiance and the area of the cell $A_C(\phi_g, \theta_g)$. The Earth albedo contribution of the cell E_C that reaches the satellite, and the density of the radiant flux are dependent on the angle ϕ_{out} .

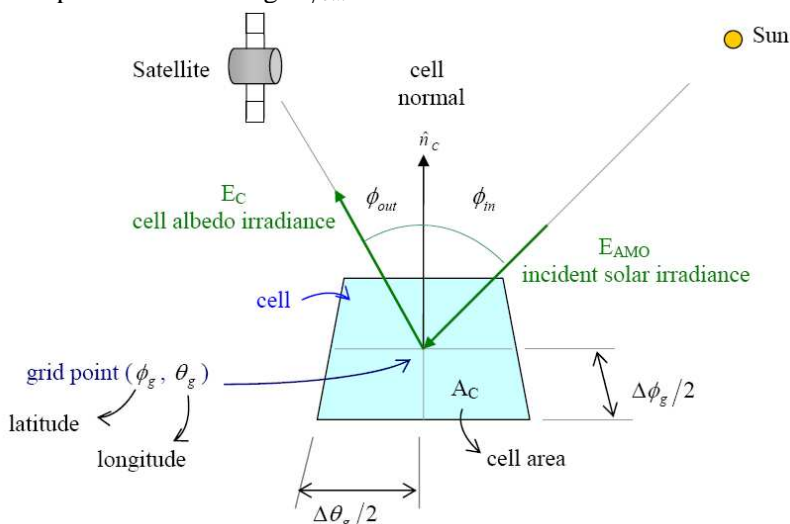


Fig. 4 - Cell at grid point (ϕ_g, θ_g) in the surface of the Earth.

The area of the cell $A_C(\phi_g, \theta_g)$ is found using surface of revolutions.

$$A_C(\phi_g, \theta_g) = \theta_g r_E^2 \left[\cos\left(\phi_g - \frac{\Delta\phi_g}{2}\right) - \cos\left(\phi_g + \frac{\Delta\phi_g}{2}\right) \right]$$

where r_E is the Earth mean radius.

The incident radiant flux $P_C(\phi_g, \theta_g)$ on a single cell at grid point (ϕ_g, θ_g) , is given by:

$$P_C(\phi_g, \theta_g) = E_{AMO} A_C(\phi_g, \theta_g) \hat{r}_{Sun} \cdot \hat{n}_C$$

where \hat{n}_C is the cell normal and \hat{r}_{Sun} is the Sun line of sight vector.

The reflected flux $P_R(\phi_g, \theta_g)$ is given by:

$$P_r(\phi_g, \theta_g) = \rho(\phi_g, \theta_g) P_C(\phi_g, \theta_g)$$

where $\rho(\phi_g, \theta_g)$ is the reflectivity of the grid point.

Assuming Lambertian reflectivity (Ryer, 1997), the reflected irradiance $E_r(\phi_g, \theta_g)$ of the cell can be calculated by:

$$E_r(\phi_g, \theta_g) = \frac{P_r(\phi_g, \theta_g)}{\pi}$$

The reflected irradiance at satellite distance $E_{sat}(\phi_g, \theta_g)$ is given by:

$$E_{sat}(\phi_g, \theta_g) = \frac{E_r(\phi_g, \theta_g)}{\|\hat{r}_{sat}\|^2}$$

where \hat{r}_{sat} is the satellite line of sight vector.

The albedo irradiance $E_C(\phi_g, \theta_g)$ from a single cell is found combine the previous equations.

$$E_C(\phi_g, \theta_g) = \frac{P_r(\phi_g, \theta_g) \hat{r}_{sat} \cdot \hat{n}_C}{\pi \|\hat{r}_{sat}\|^2} = \frac{\rho(\phi_g, \theta_g) E_{AMO} A_C (\hat{r}_{sun} \cdot \hat{n}_C) (\hat{r}_{sat} \cdot \hat{n}_C)}{\pi \|\hat{r}_{sat}\|^2}$$

And the total albedo irradiance is given by the following expression:

$$E_a = \sum_{V_{Sun} \cap V_{sat}} E_C(\phi_g, \theta_g)$$

where $V_{Sun} \cap V_{sat}$ is the set of sunlit grid points visible from the satellite, V_{Sun} are the grid points visible from Sun and V_{sat} are the grid points visible from satellite.

The reference system used in the model is presented in the next figure.

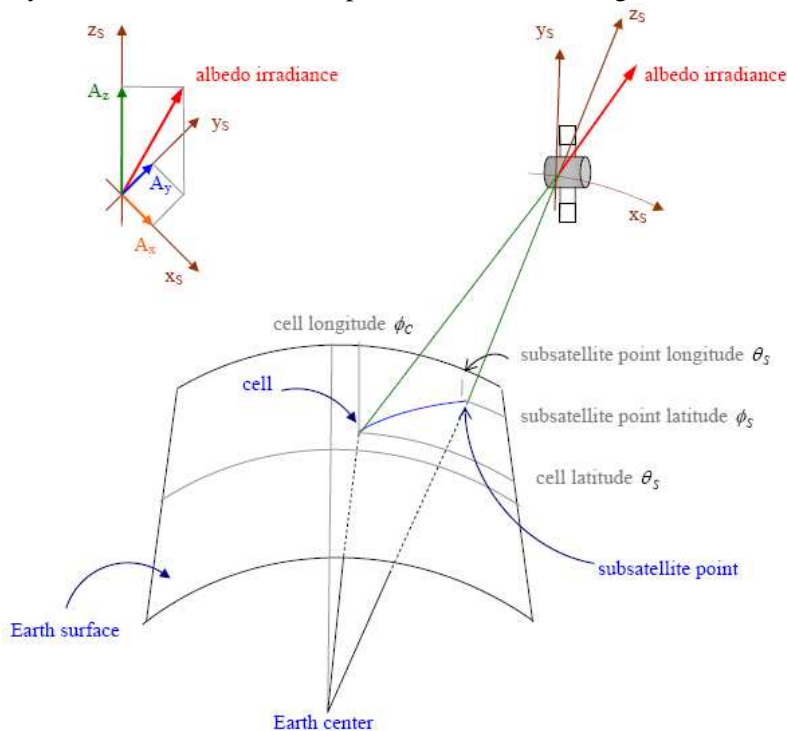


Fig. 5 – The reference system.

		Doc.No.: FLK-ENV-RP-ZAR-001 Issue: 2.0 Page: 9 of 91
---	--	--

The x_S axis is parallel to the Earth equatorial plane. The y_S axis is in the plane that contains the meridian of the subsatellite point. The z_S axis is in the direction of the radius.

3- Implementation of the model

The model was implemented in Matlab/Simulink using S-Functions generated from routines written in C code. The main window of the model is shown in Figure 6. The item 3.1 presented a description of the main subsystems. Figure 7 presents the structure of the Earth Albedo Model

3.1- Description of the Sub-Systems

To calculate the albedo, the state vector of the satellite must be determined. This vector is obtained in the subsystem *State Vector*, from the initial keplerian elements of the satellite. Beyond the state vector it is necessary to determine the sun vector, so this vector is calculated in the subsystem *Sun Vector* considering the modified julian date related to 1950.0 as input.

With the state vector and the sun vector, the illumination condition is calculated in the subsystem *Illumination*. These vectors were calculated considering the geocentric inertial frame referred to vernal equinox (ECI) related to J2000 equator and equinox. To calculate the albedo it is necessary to transform these coordinates into geocentric terrestrial coordinates (ECEF). This is made in the subsystem *Geocentric Inertial Frame to Geocentric Terrestrial Frame*. With these new coordinates the albedo, the incident angle at each cell and the incident angle at the satellite are calculated in the subsystems *Albedo Irradiance*, *Incident Angle at Grid Point* and *Incident Angle at the Satellite*.

The subsystem *Albedo Irradiance* provides a map (Figures 21 and 22) that shows the albedo that reaches the satellite from each cell visible by the satellite in the surface of the Earth. However, to determine the total albedo vector it is necessary to sum all albedo vectors from all cells. Thus some calculations must be done.

The subsystem *Subsatellite Point* provides the longitude and the latitude of the subsatellite point in the Earth surface. Then, in the next subsystem, the longitude and the latitude of each cell are determined and the result is saved in the matrix *cell_longitude* and *cell_latitude*. With these matrixes and with the subsatellite coordinates the matrixes *cell_albedo_x*, *cell_albedo_y* and *cell_albedo_z* are calculated in the subsystem *Albedo of Each Cell*. Finally, the total albedo vector is calculated in the subsystem *Albedo Sum*.

The main structure of the Earth Albedo Model is shown in Figure 6. The input variables of the model are related in Figure 7, and the output variables in Figure 8. The variables saved in the workspace during the execution of the model are shown in Figure 9.

To calculate forces and torques applied to the satellite due to the albedo the matrixes *cell_albedo_x*, *cell_albedo_y* and *cell_albedo_z* can be used. But a model describing the satellite shape and a description of the optical properties referred to the satellite surfaces are necessary. However the albedo model is not dependent of the satellite model and can be applied to any mission, considering different satellite models. Meanwhile the development of these satellite models is not the goal of the albedo model.

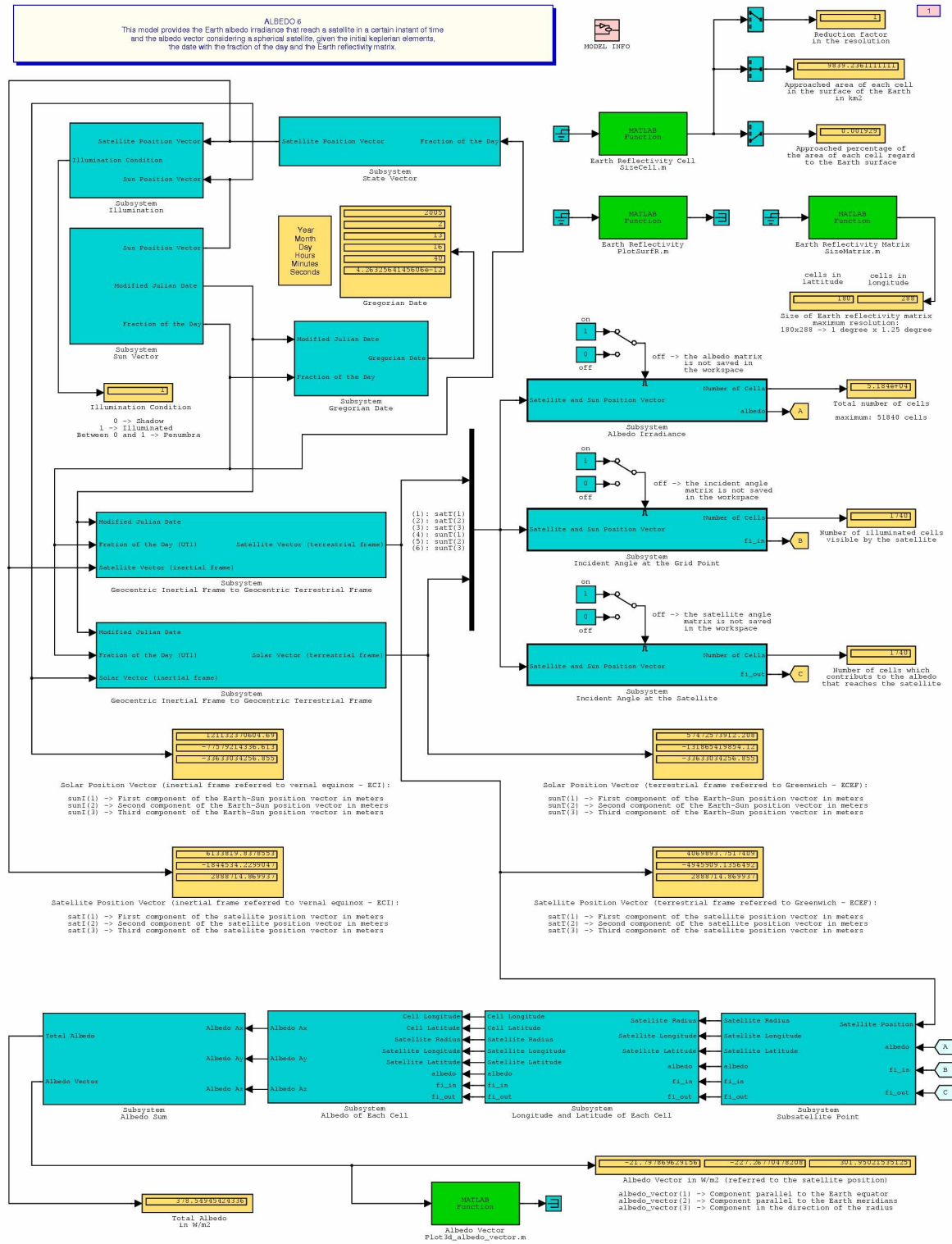


Fig. 6- Main window of the Earth Albedo Model.

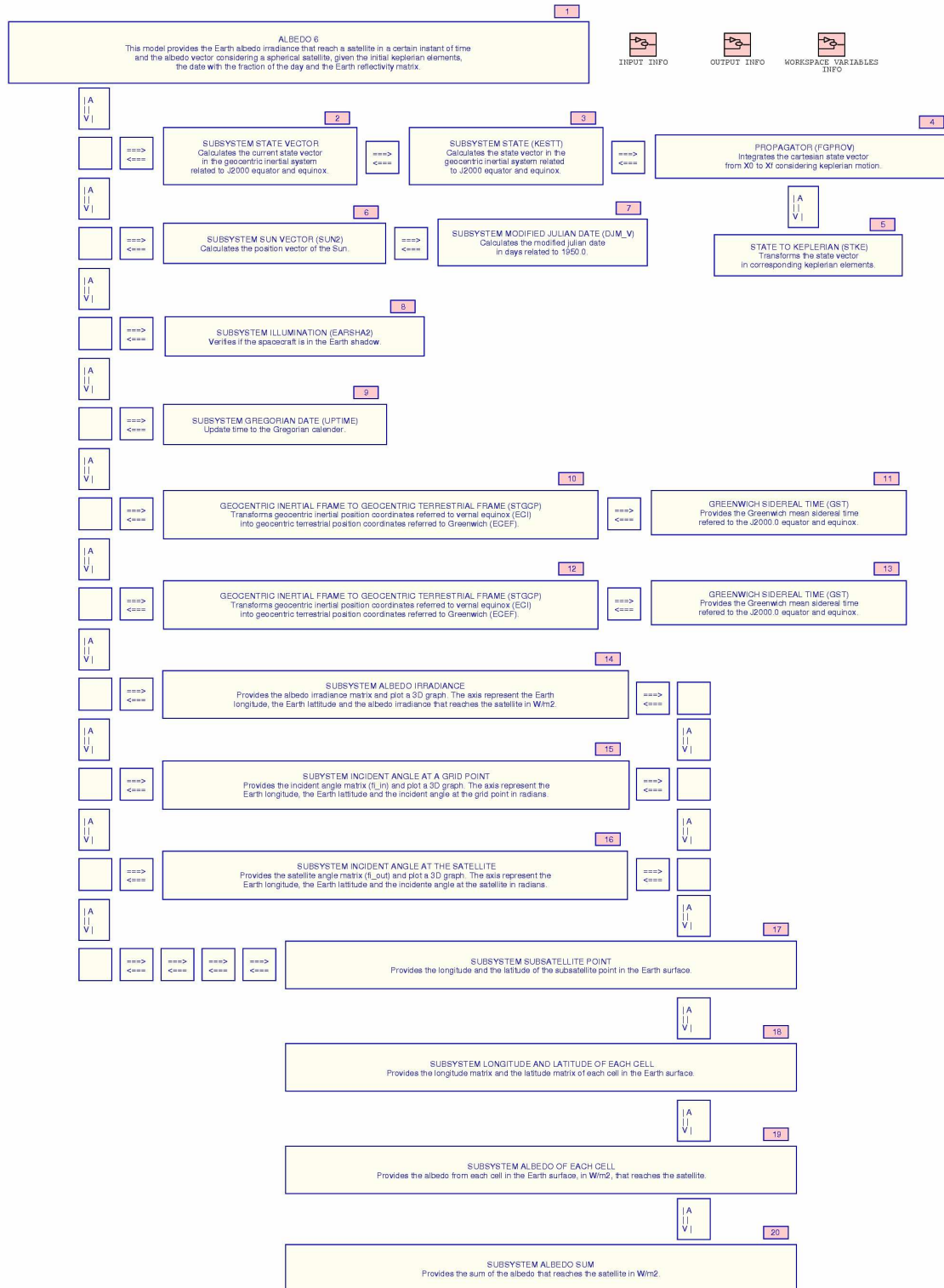


Fig. 7 – Structure of the Earth Albedo Model.

3.2- Input Variables

The input variables for the model are presented in Figure 8. The figure shows the input variables and the sub-system where the variables must be set.

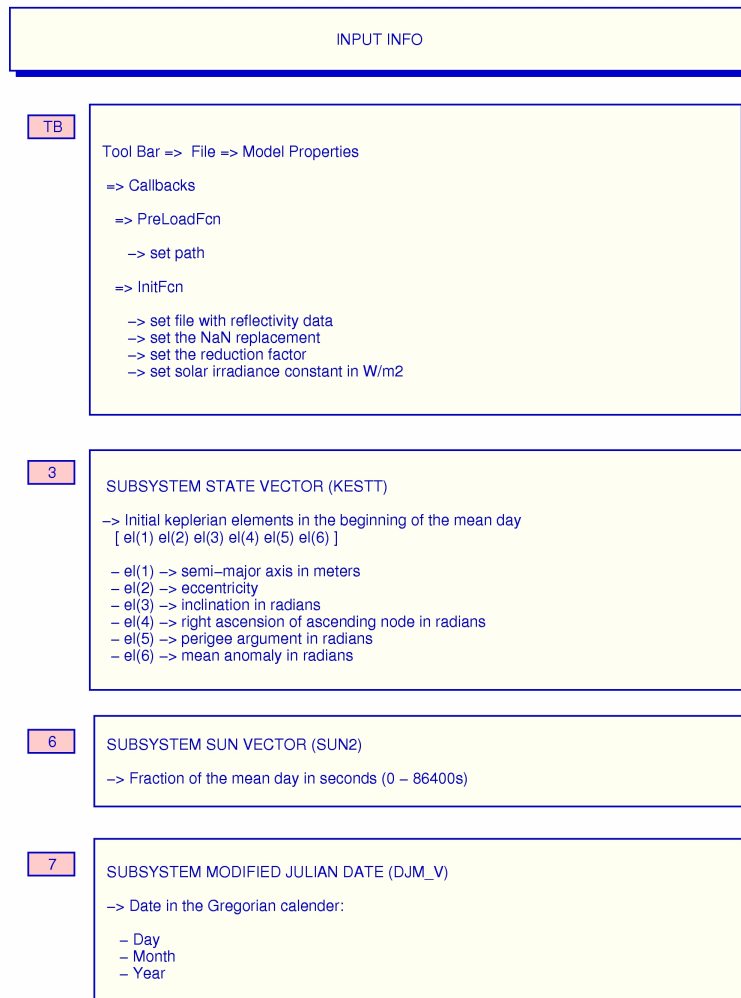


Fig. 8 – Input variables of the Earth Abedo Model.

3.3- Output Variables

The output variables are presented in Figure 9.

OUTPUT INFO	
1	ALBEDO 6 -> Reduction factor applied in the resolution -> Area of each cell in the surface of the Earth in km ² -> Percentage of the area of each cell regard to the total area of the Earth surface -> Size of Earth reflectivity matrix
3	SUBSYSTEM STATE (KESTT) -> State vector in the initial of the simulation (beginning of the mean day) - X } - Y } -> position in the inertial frame (meters) - Z } - Xp } - Yp } -> velocity in the inertial frame (meters/seconds) - Zp } -> Rotation matrix from the inertial frame to the orbital frame -> Initial eccentric anomaly in radians -> Initial true anomaly in radians
4	SUBSYSTEM PROPAGATOR (FGPROV) -> Current state vector: - X } - Y } -> position (meters) - Z } - Xp } - Yp } -> velocity (meters/seconds) - Zp }
8	SUBSYSTEM STATE TO KEPLERIAN (KEORV) -> Current keplerian elements [e1(1) e1(2) e1(3) e1(4) e1(5) e1(6)]: - e1(1) -> semi-major axis in meters - e1(2) -> eccentricity - e1(3) -> inclination in radians - e1(4) -> right ascension of ascending node in radians - e1(5) -> perigee argument in radians - e1(6) -> mean anomaly in radians -> Current mean anomaly in radians -> Current mean anomaly in degrees
6	SUBSYSTEM SUN VECTOR (SUN2) -> Solar vector (inertial frame) - sun(1) -> First component of the Earth-Sun position vector in meters - sun(2) -> Second component of the Earth-Sun position vector in meters - sun(3) -> Third component of the Earth-Sun position vector in meters
7	SUBSYSTEM MODIFIED JULIAN DATE (DJM_V) -> Modified julian date in days related to 1950.0
8	SUBSYSTEM ILLUMINATION (EARSHA2) -> Illumination Condition 0 -> Shadow 1 -> Illuminated Between 0 and 1 -> Penumbra
9	SUBSYSTEM GREGORIAN DATE (UPTIME) -> Time in the Gregorian calendar: - Year - Month - Day - Hours - Minutes - Seconds
10	SUBSYSTEM GEOCENTRIC INERTIAL FRAME TO GEOCENTRIC TERRESTRIAL FRAME (STGCP) -> Geocentric terrestrial position coordinates (referred to Greenwich)
11	SUBSYSTEM GREENWICH SIDEREAL TIME (GST) -> Greenwich sideral time in radians
12	
13	
14	SUBSYSTEM ALBEDO IRRADIANCE -> Albedo irradiance matrix -> Albedo irradiance 3D graph -> Total number of cells
15	SUBSYSTEM INCIDENT ANGLE AT THE GRID POINT -> Incident angle matrix (fi_in) -> Incident angle 3D graph -> Number of illuminated cells visible by the satellite
16	SUBSYSTEM INCIDENT ANGLE AT THE SATELLITE -> Satellite angle matrix (fi_out) -> satellite angle 3D graph -> Number of cells which contributes to the albedo that reaches the satellite
17	SUBSYSTEM SUBSATELLITE POINT -> Satellite radius in meters -> Satellite longitude matrix in radians -> Satellite latitude matrix in radians
18	SUBSYSTEM LONGITUDE AND LATITUDE OF EACH CELL -> Cell longitude matrix in radians -> Cell latitude matrix in radians
19	SUBSYSTEM OF EACH CELL -> Cell albedo_x matrix in W/m ² (component parallel to the Earth equator) -> Cell albedo_y matrix in W/m ² (Component parallel to the Earth meridians) -> Cell albedo_z matrix in W/m ² (component in the direction of the radius)
20	SUBSYSTEM ALBEDO SUM -> Albedo vector referred to the satellite position in W/m ² (sum of the albedo coming from all cells in the Earth surface) - albedo_vector(1) -> component parallel to the Earth equator - albedo_vector(2) -> component parallel to the Earth meridians - albedo_vector(3) -> component in the direction of the radius -> Total albedo that reaches the satellite in W/m ² (absolute value of the albedo vector)

Fig. 9 – Output variables of the Earth Albedo Model.

3.4- Variables Saved in the Workspace

The variables saved in the Matlab workspace are presented in Figure 10.

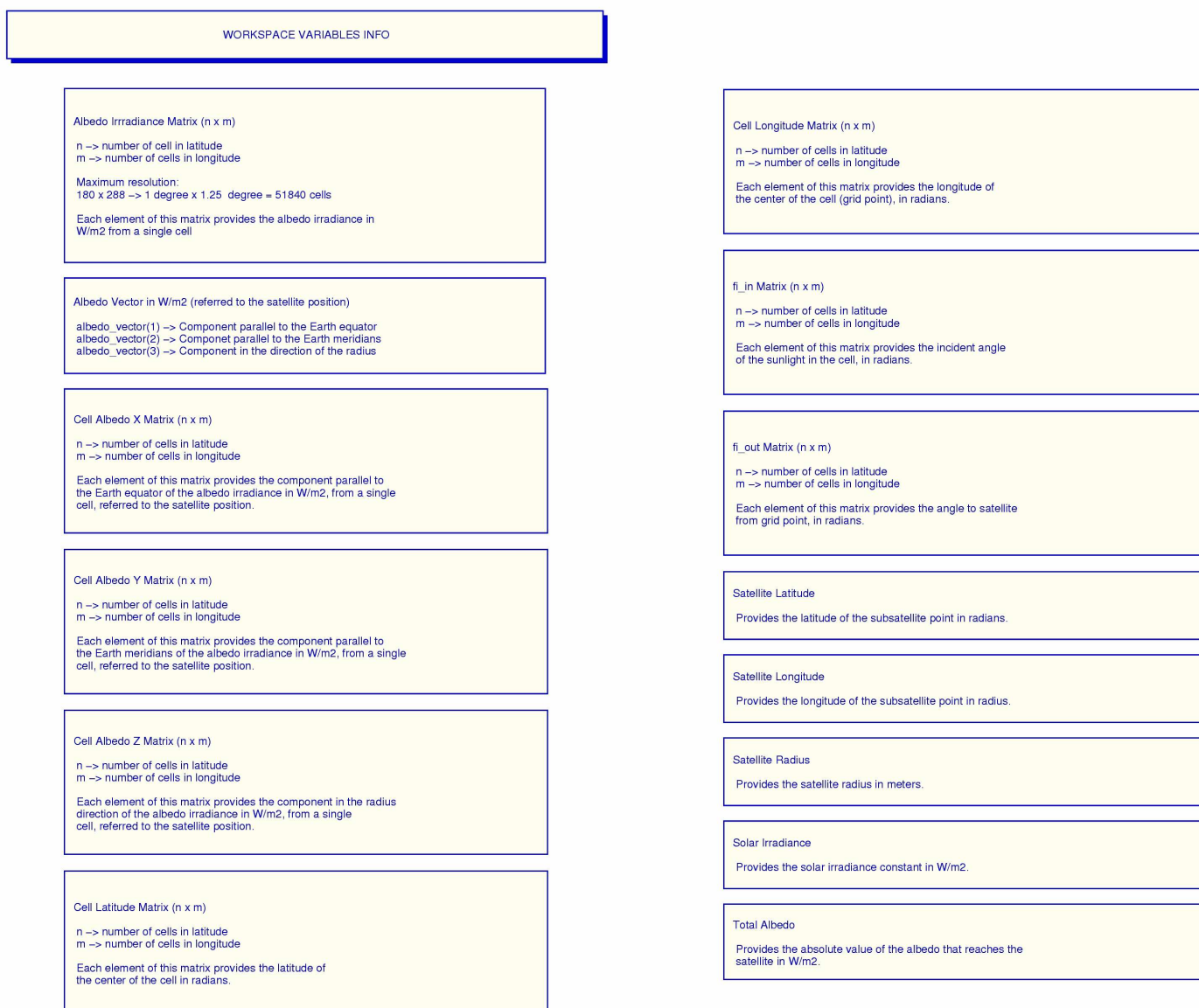


Fig. 10 – Variables saved in the workspace.

4- Some Results Provided by the Earth Albedo Model

4.1- The Reflectivity of the Earth Surface

The reflectivity of the Earth considering the date 13.02.2006 is shown in Figures 11 and 12. Figure 11 shows the reflectivity replacing the gaps in the data measured by zero. Figure 12 shows the reflectivity replacing the gaps by one.

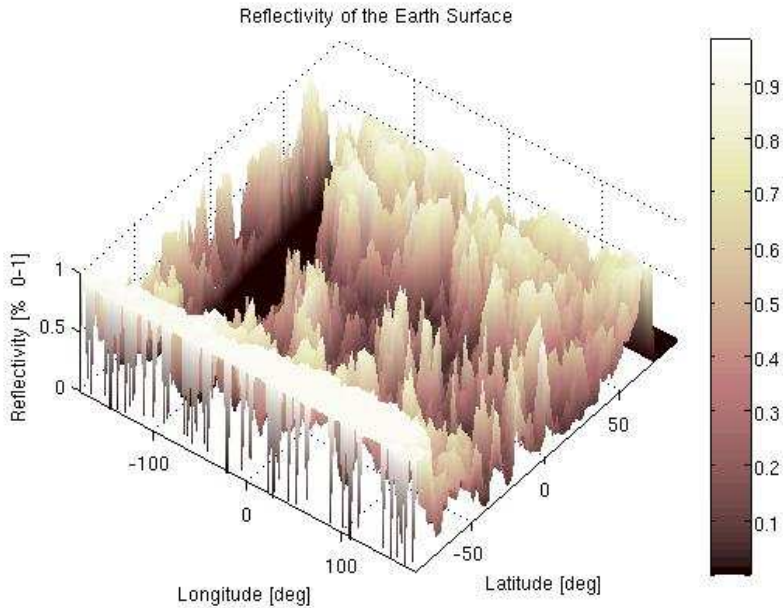


Fig. 11 – Reflectance of the Earth replacing the gaps by zero.

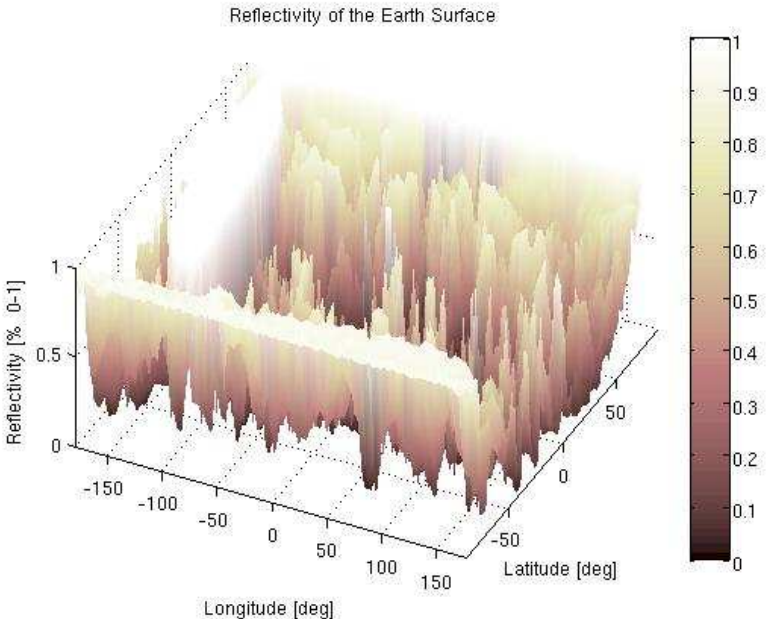


Fig. 12 – Reflectance of the Earth replacing the gaps by one.

4.2- The Full Albedo Irradiance

In Figures 13 and 14 the full albedo irradiance is presented. This represents a hypothetical situation that considers the satellite and the Sun at the zenith of each cell in the surface of the Earth. The full albedo can be considered the maximum value for the albedo in W/m^2 for a given altitude.

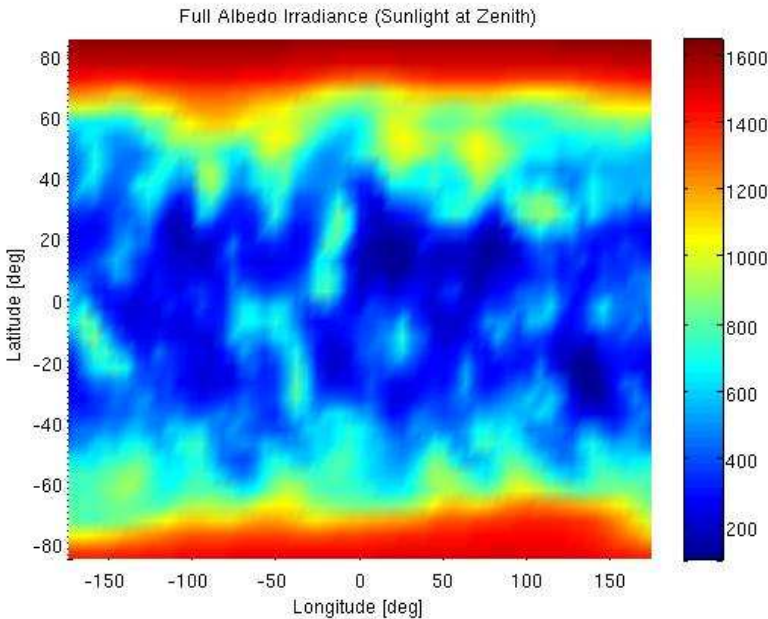


Fig. 13 – Full albedo irradiance.

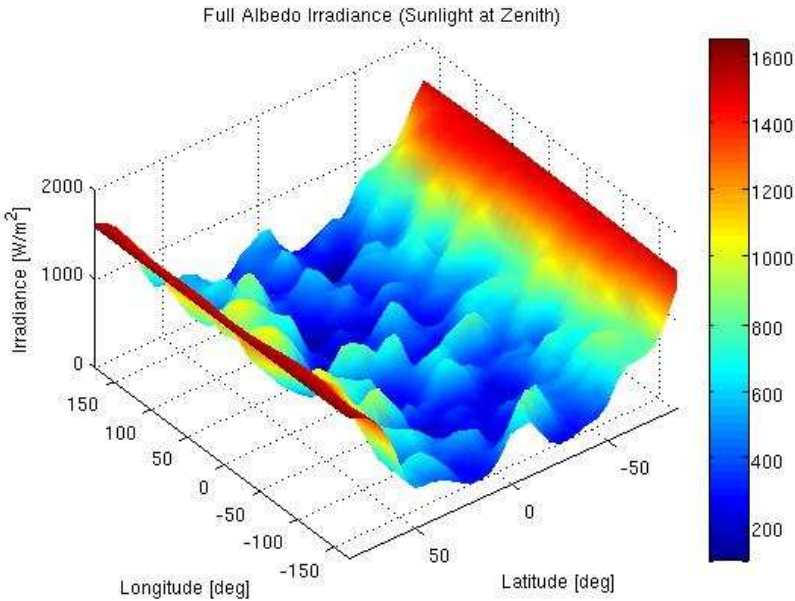


Fig. 14 – Full albedo irradiance 3D.

4.3- The Instantaneous Albedo Irradiance

The instantaneous albedo irradiance is presented in Figures 15 and 16. This represents a hypothetical situation that considers the satellite at the zenith of each cell and the Sun in a fixed predefined position (dependent of the time). The instantaneous albedo can be considered the maximum albedo in W/m^2 of each cell for a given altitude at a specific instant of time.

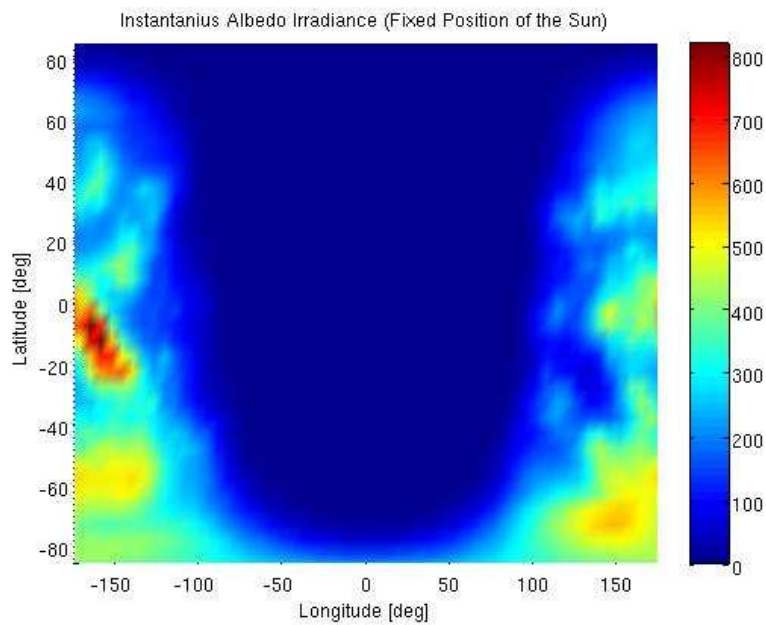


Fig. 15 – Instantaneous albedo irradiance.

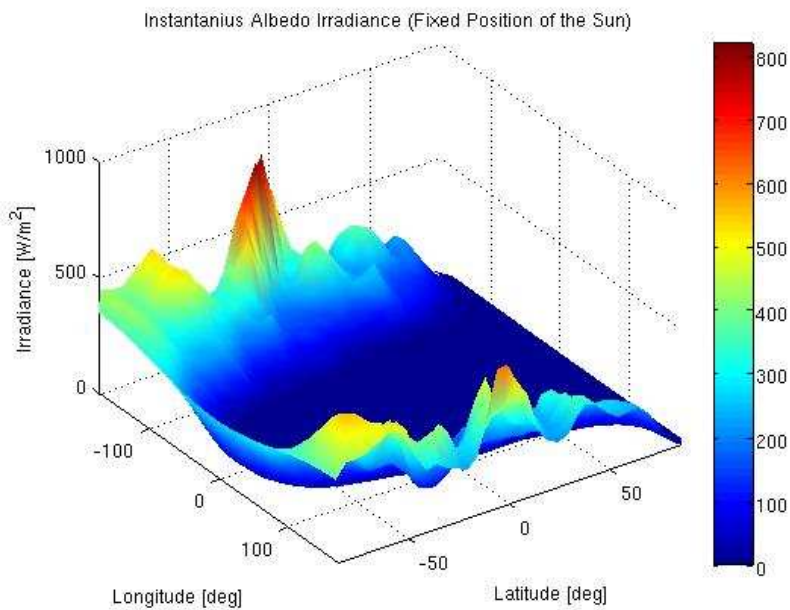


Fig. 16 – Instantaneous albedo irradiance 3D.

4.4- The Incident Angle at the Grid Point

In Figures 17 and 18 the incident angle at the grid point ϕ_m is presented in radians.

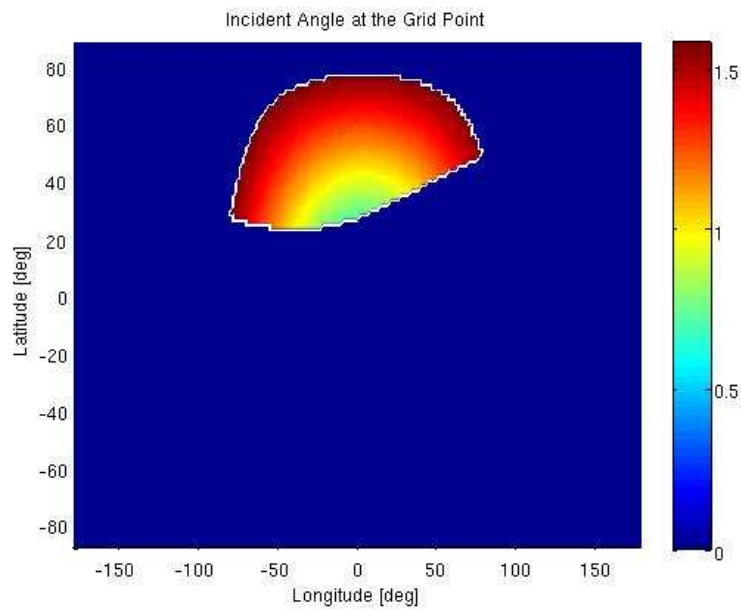


Fig. 17 – Incident angle at the grid point.

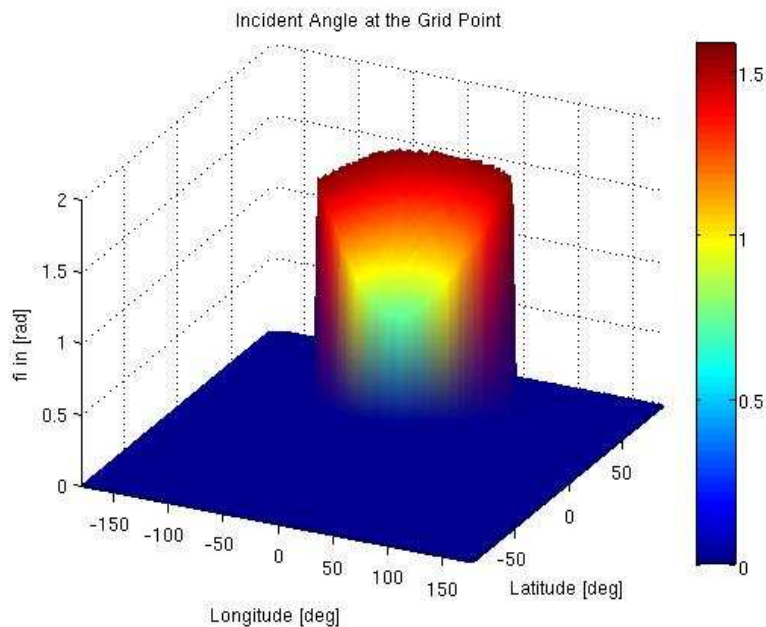


Fig. 18 – Incident angle at the grid point 3D.

4.5- The Incident Angle at the Satellite

In Figures 19 and 20 the incident angle at the satellite ϕ_{out} is presented in radians.

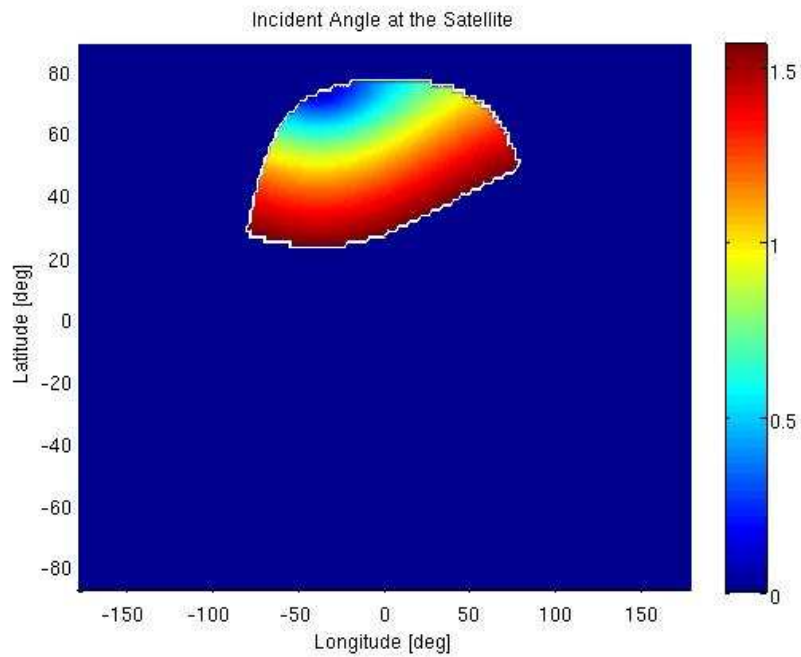


Fig. 19 – Incident angle at the satellite.

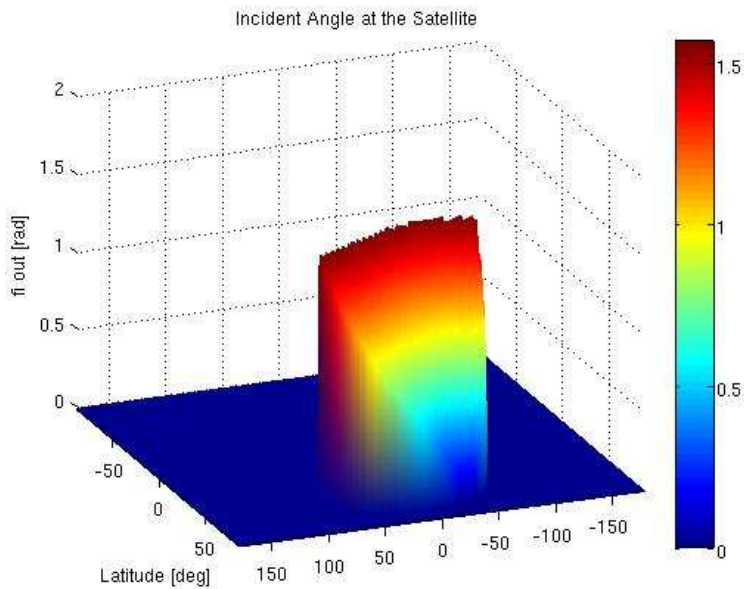


Fig. 20 – Incident angle at the satellite 3D.

4.6- The Albedo Irradiance at the Satellite

The albedo irradiance is presented in Figures 21 and 22. This represents the albedo regard to the set of sunlit grid points visible from the satellite for a specific position of the satellite in a specific instant of time.

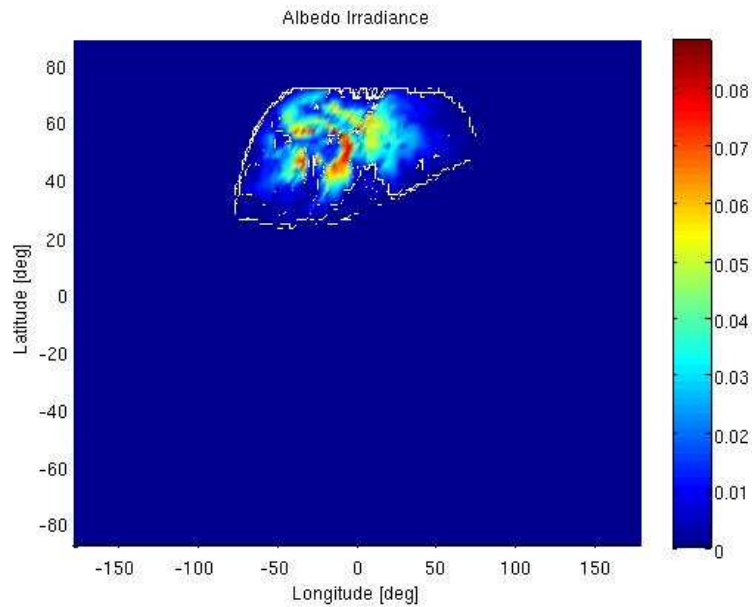


Fig. 21 – Albedo irradiance in W/m^2 .

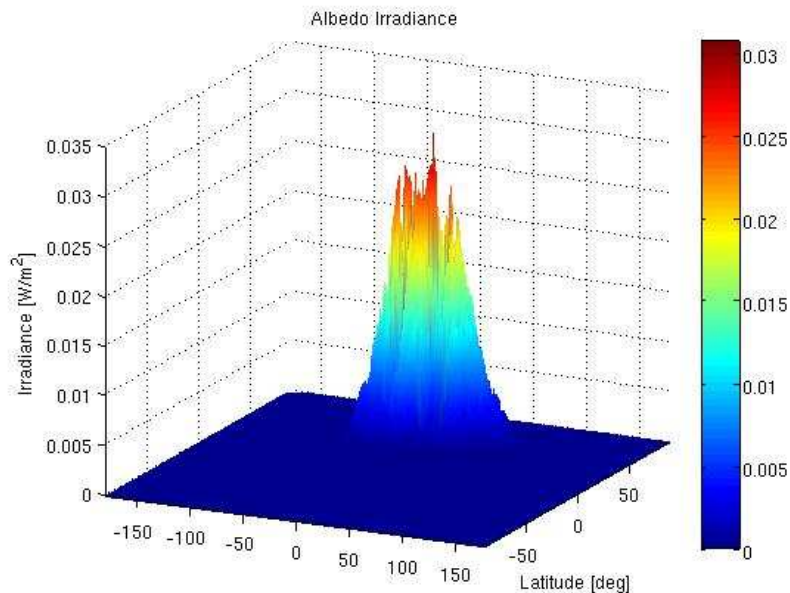


Fig. 22 – Albedo irradiance in W/m^2 3D.

4.7- The Albedo Vector Centered in the Satellite Position

Figure 23 shows the albedo vector that reaches the satellite in W/m^2 .

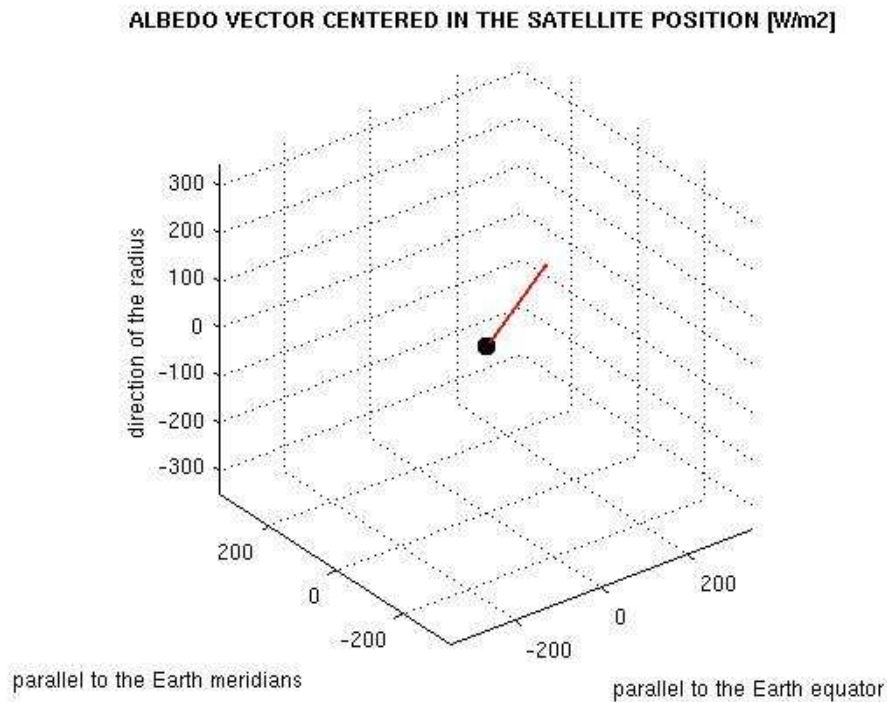


Fig. 23 – Albedo vector in W/m^2 .

4.8- Satellite Illumination Condition for the Entire Day

The satellite illumination condition is presented in Figure 24.

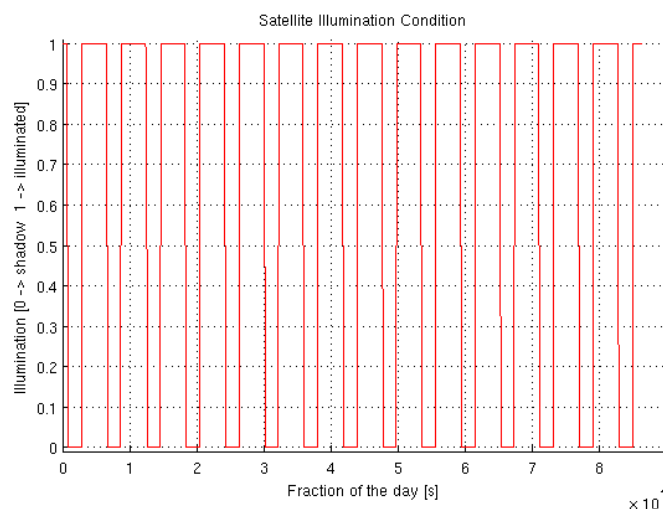


Fig. 24 – Satellite illumination condition for the entire day.

4.9- Number of Illuminated Visible Cells for the Entire Day

The number of illuminated cells visible by the satellite is presented in Figure 25.



Fig. 25 – Number of illuminated visible cells for the entire day.

4.10- Total Albedo for the Entire Day

The total albedo in W/m² for the entire day at satellite position is presented in Figure 26.

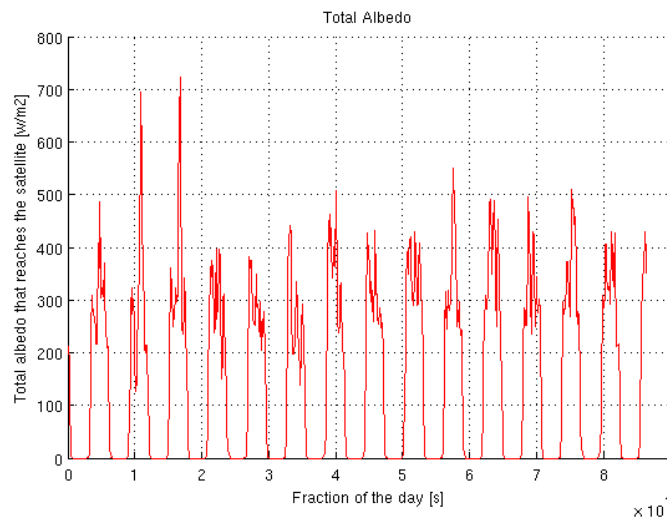


Fig. 26 – Total albedo for the entire day.

The total albedo vector in W/m^2 for the entire day is presented in Figure 27.

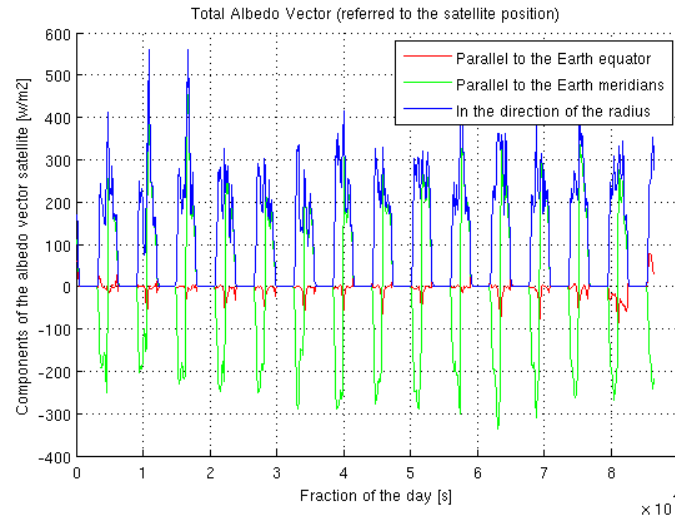


Fig. 27 – Total albedo vector for the entire day.

5- Results Obtained Considering Different Resolutions

The satellite data is given in a maximum resolution of $\Delta\phi g = 1$ deg in latitude times $\Delta\theta g = 1.25$ deg in longitude. This represents 180×288 data points that results in 51840 cells in the maximum resolution. Thus, the number of elements in the matrixes used to calculated the albedo vector is very big (51840 elements). A possibility to reduce the computer effort in the calculations could be obtained reducing of the number of elements, considering the average of the reflectivity of a group of cells in the surface of the Earth. Table 1 shows the number of cells related to a reduction factor that can be set in the Earth Albedo Model.

Table 1- Reduction Factor

Reduction Factor	Size of the Reflectivity Matrix	Number of Cells
1	180 x 280	51840
2	90 x 144	12960
3	60 x 96	5760
4	45 x 72	3240
5	36 x 57	2052
6	30 x 48	1440
7	25 x 36	1025
8	22 x 36	792
9	20 x 32	640
10	18 x 28	504
15	12 x 19	228
20	9 x 14	126

A study of the effect in the results obtained with the Albedo Model using different resolutions is presented in Figures 28 to 98

5.1- Reduction Factor 1

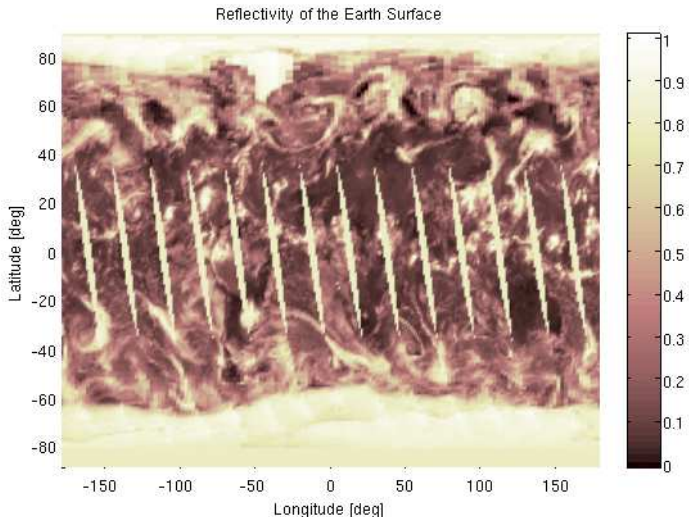


Fig. 28 – Reflectivity of the Earth Surface (red. factor 1).

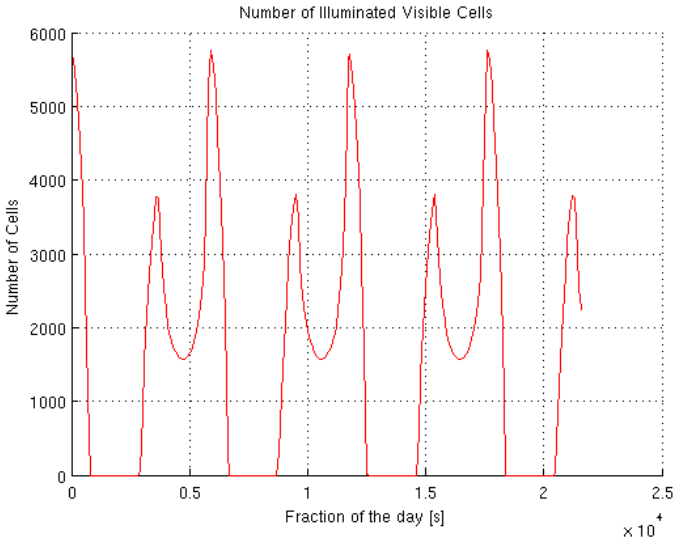


Fig. 29 – Illuminated Visible Cells (red. factor 1).

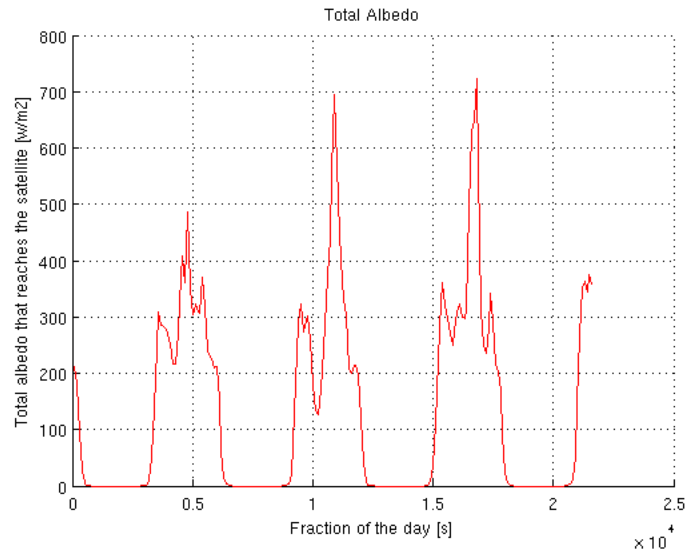


Fig. 30 – Total albedo (red. factor 1).

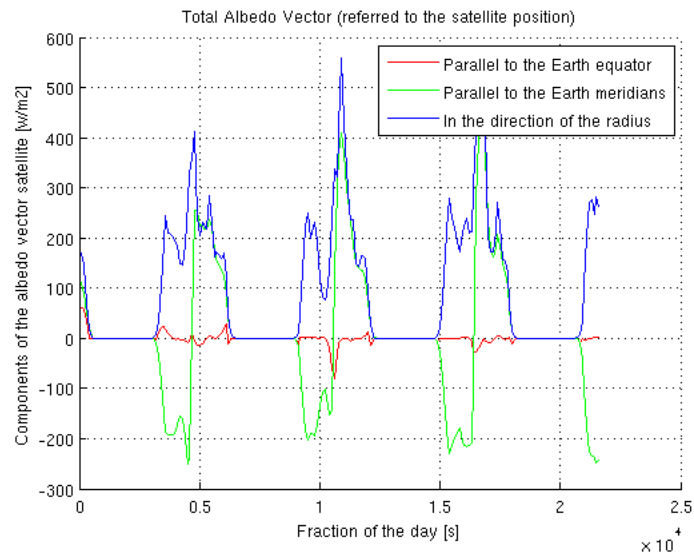


Fig. 31 – Total albedo vector (red. factor 1).

5.2- Reduction Factor 2

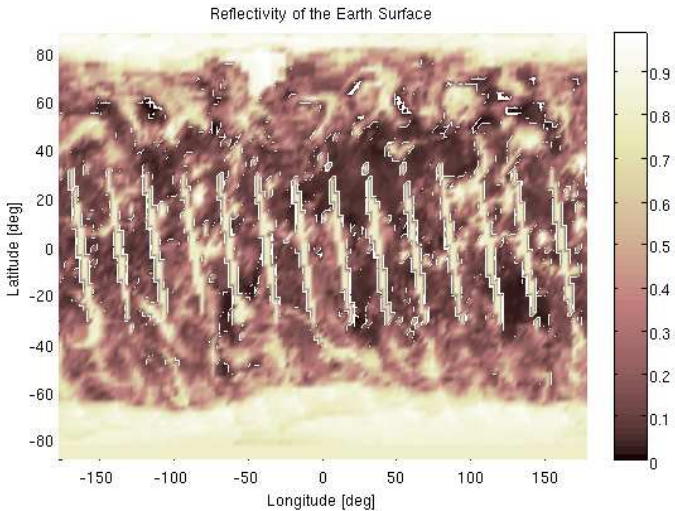


Fig. 32 – Reflectivity of the Earth Surface (red. factor 2).

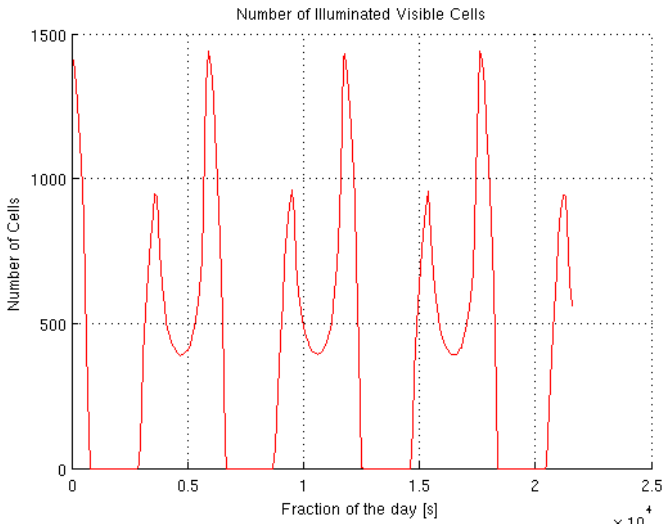


Fig. 33 – Illuminated Visible Cells (red. factor 2).

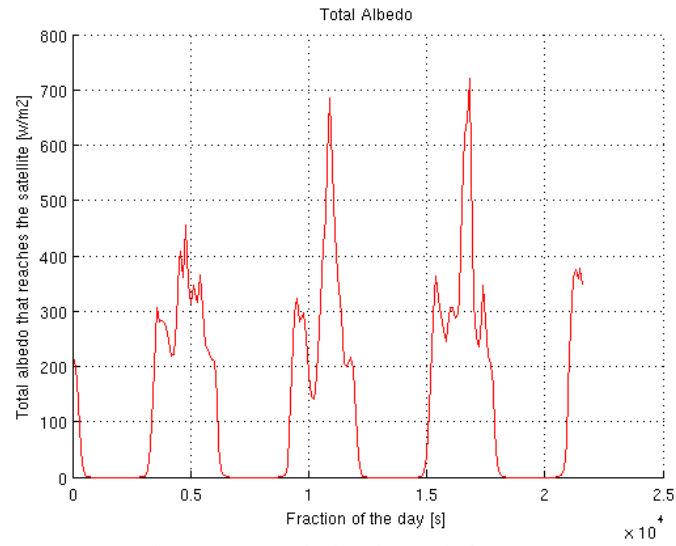


Fig. 34 – Total albedo (red. factor 2).

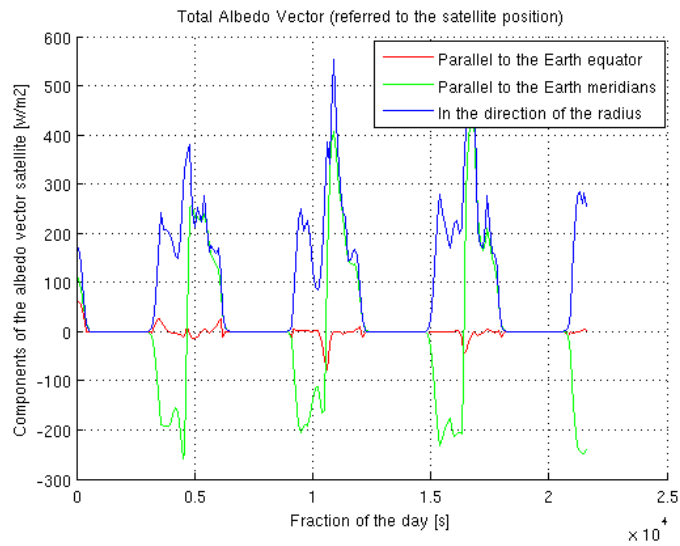


Fig. 35 – Total albedo vector (red. factor 2).

5.2.1- Reduction Factor 1 x Reduction Factor 2

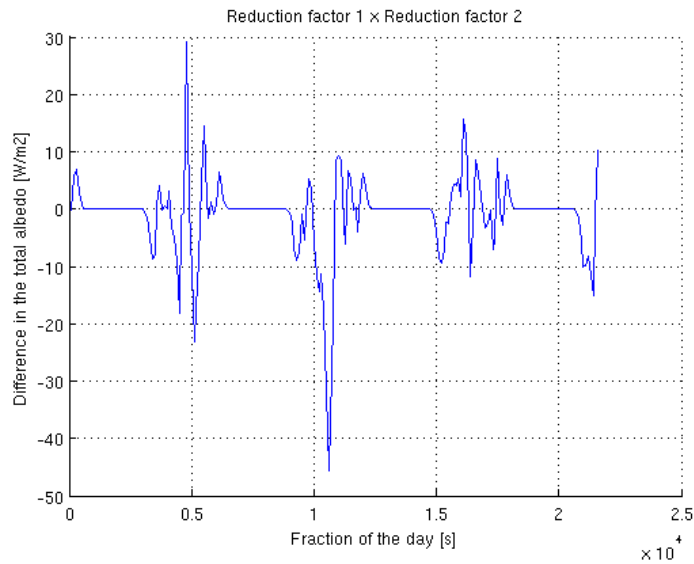


Fig. 36 – Difference in the total albedo (red. factor 1 x red. factor 2).

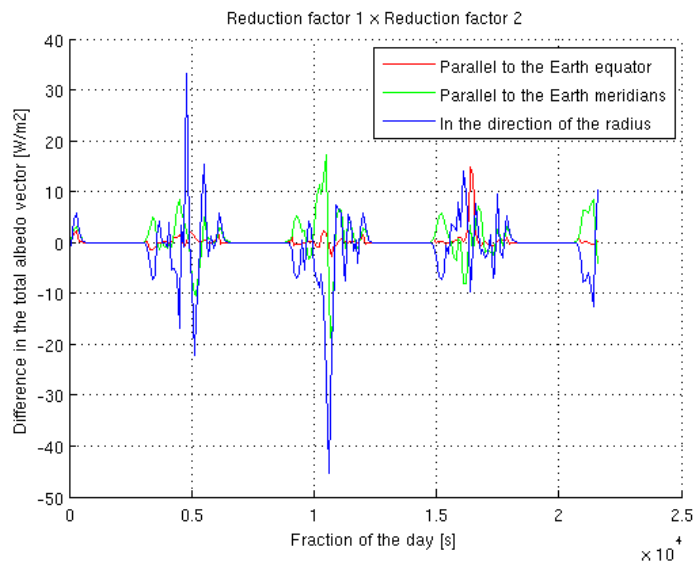


Fig. 37 – Difference in the total albedo vector (red. factor 1 x red. factor 2).

5.3- Reduction Factor 3

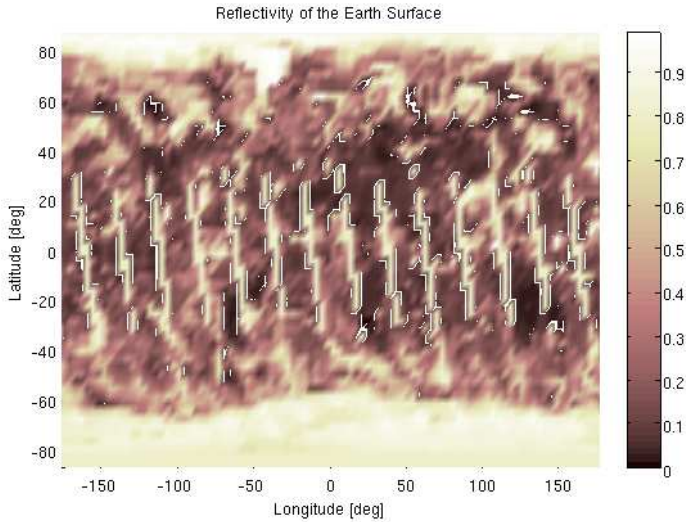


Fig. 38 – Reflectivity of the Earth Surface (red. factor 3).

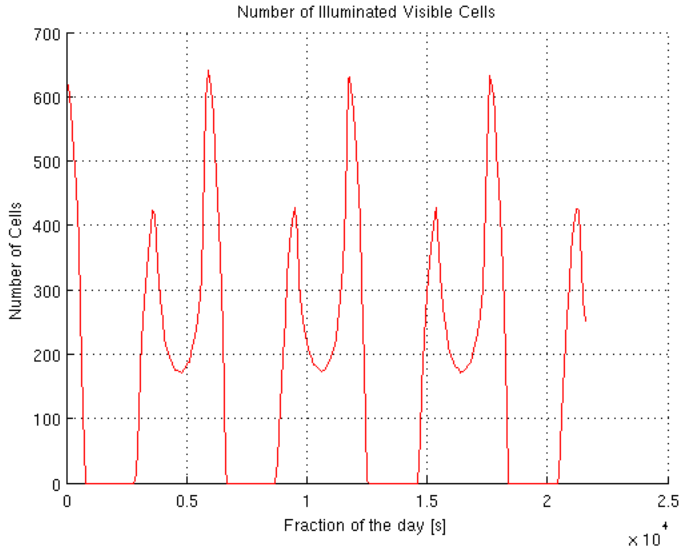


Fig. 39 – Illuminated Visible Cells (red. factor 3).

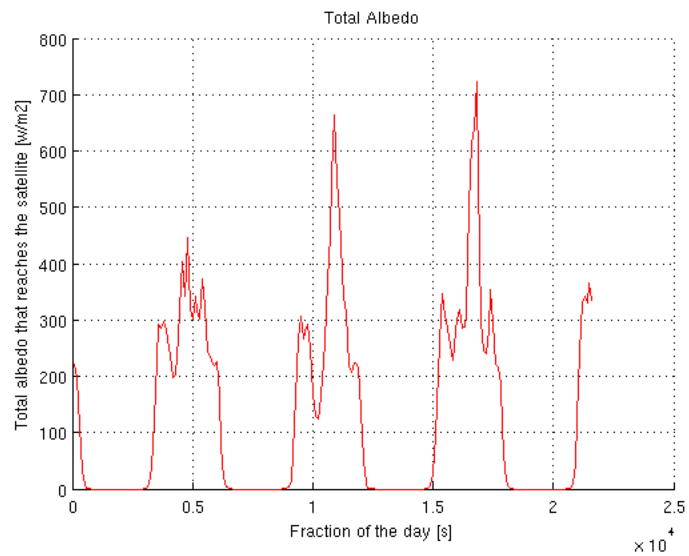


Fig. 40 – Total albedo (red. factor 3).

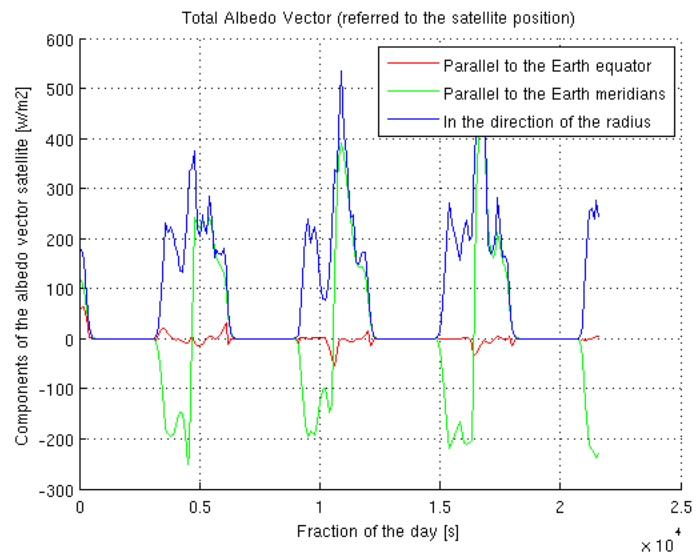


Fig. 41 – Total albedo vector (red. factor 3).

5.3.1- Reduction Factor 1 x Reduction Factor 3

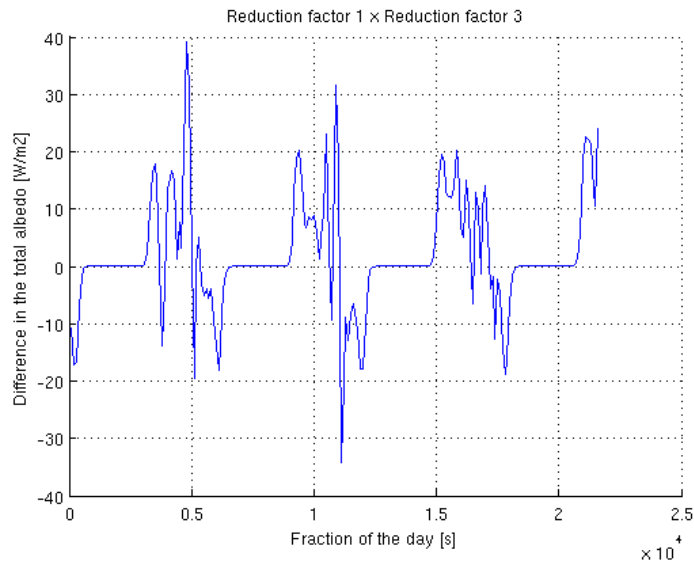


Fig. 42 – Difference in the total albedo (red. factor 1 x red. factor 3).

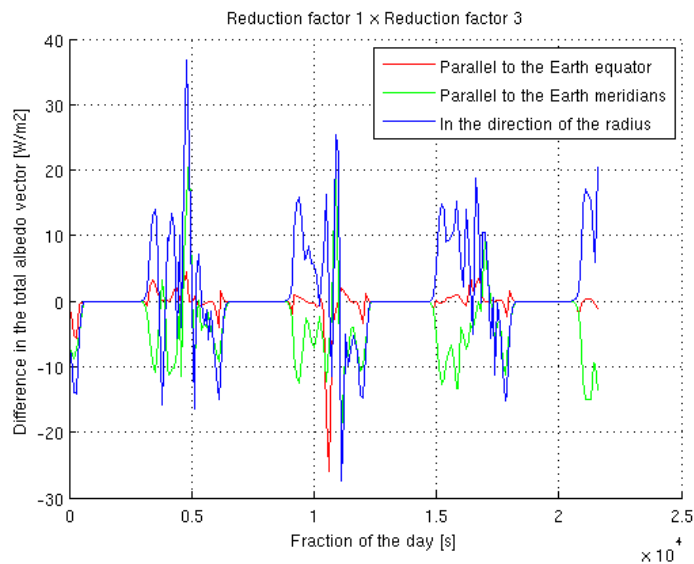


Fig. 43 – Difference in the total albedo vector (red. factor 1 x red. factor 3).

5.4- Reduction Factor 4

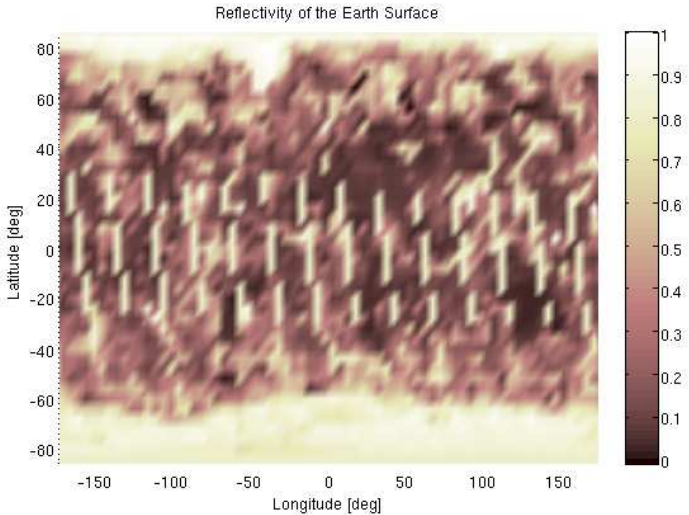


Fig. 44 – Reflectivity of the Earth Surface (red. factor 4).

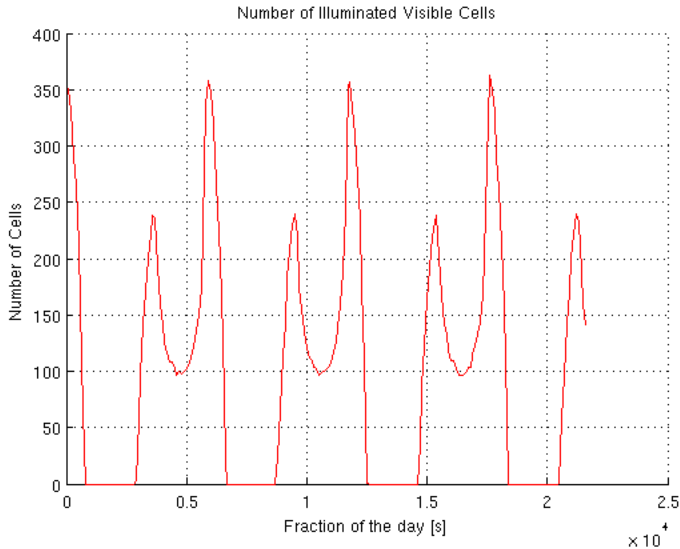


Fig. 45 – Illuminated Visible Cells (red. factor 4).

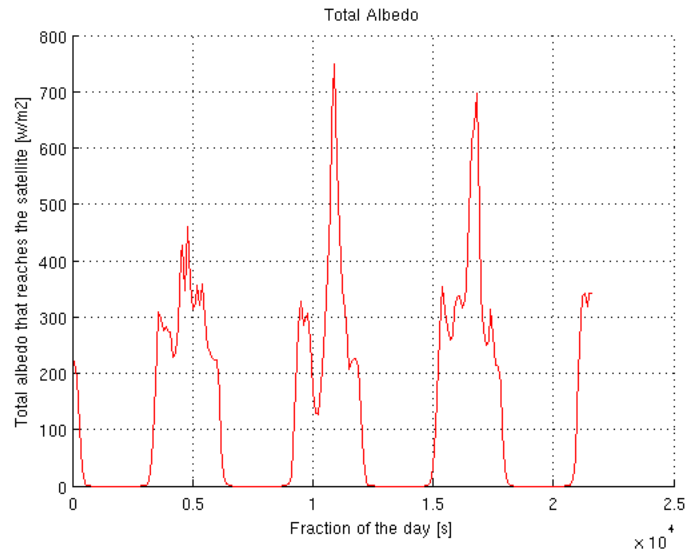


Fig. 46 – Total albedo (red. factor 4).

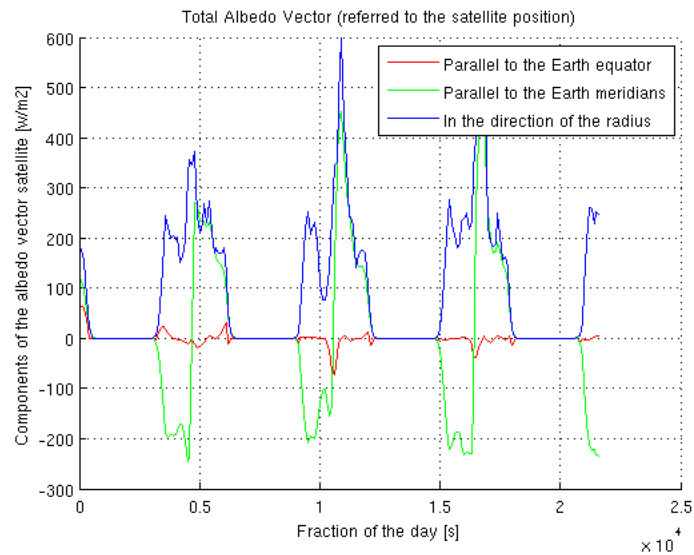


Fig. 47 – Total albedo vector (red. factor 4).

5.4.1- Reduction Factor 1 x Reduction Factor 4

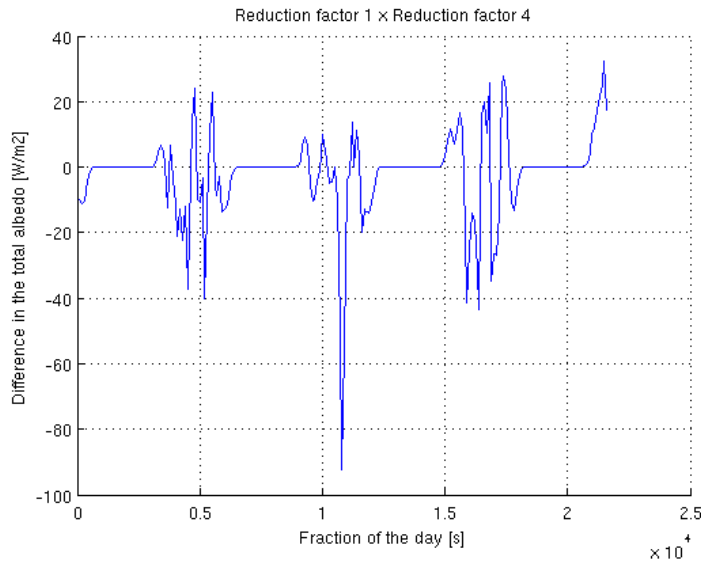


Fig. 48 – Difference in the total albedo (red. factor 1 x red. factor 4).

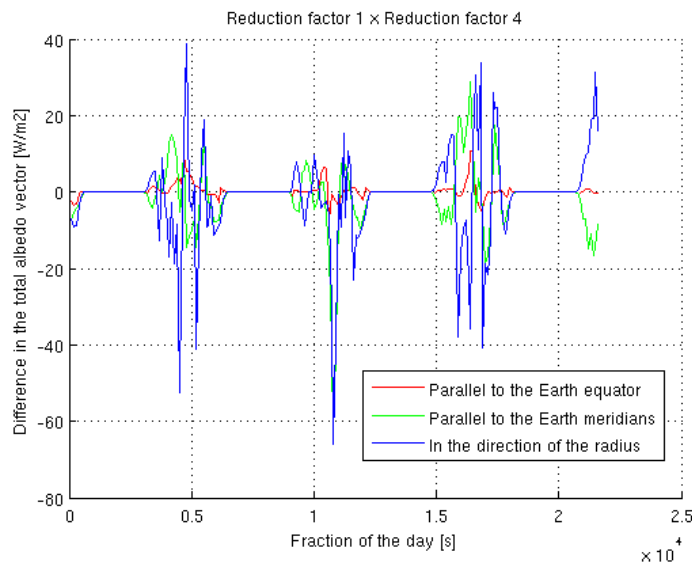


Fig. 49 – Difference in the total albedo vector (red. factor 1 x red. factor 4).

5.5- Reduction Factor 5

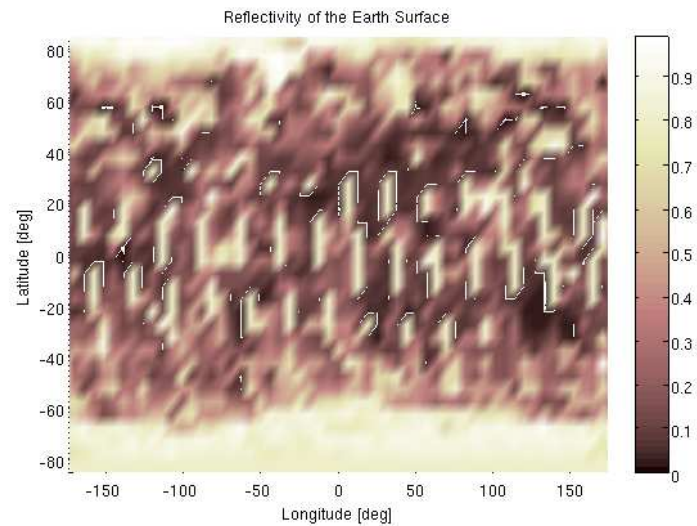


Fig. 50 – Reflectivity of the Earth Surface (red. factor 5).

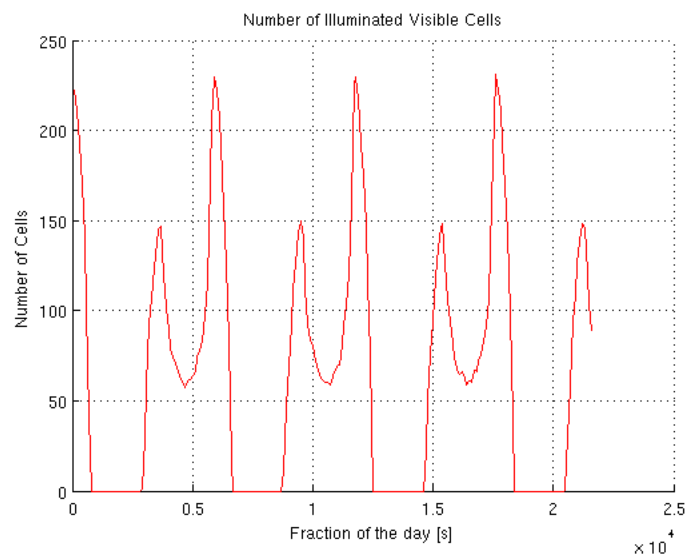


Fig. 51 – Illuminated Visible Cells (red. factor 5).

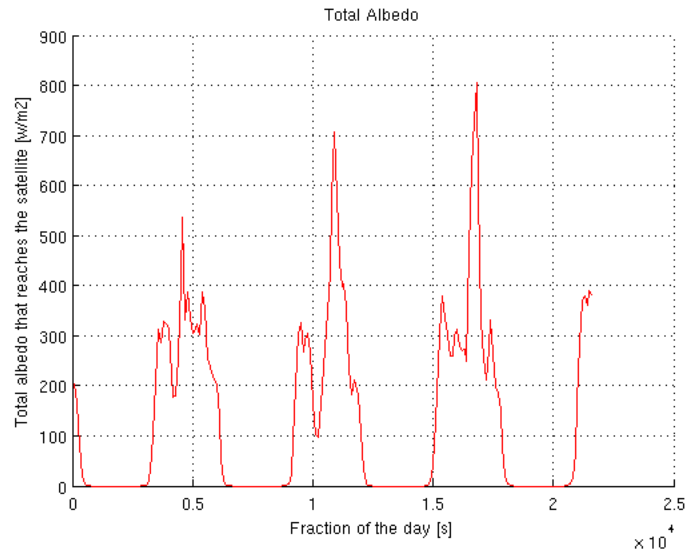


Fig. 52 – Total albedo (red. factor 5).

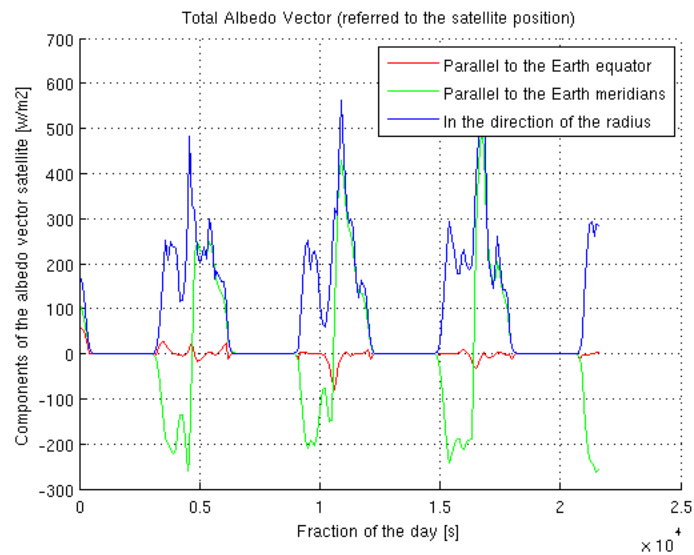


Fig. 53 – Total albedo vector (red. factor 5).

5.5.1- Reduction Factor 1 x Reduction Factor 5

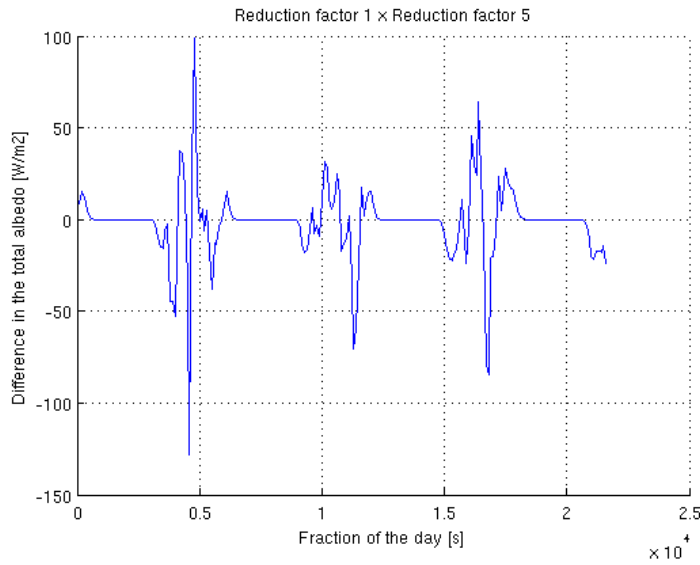


Fig. 54 – Difference in the total albedo (red. factor 1 x red. factor 5).

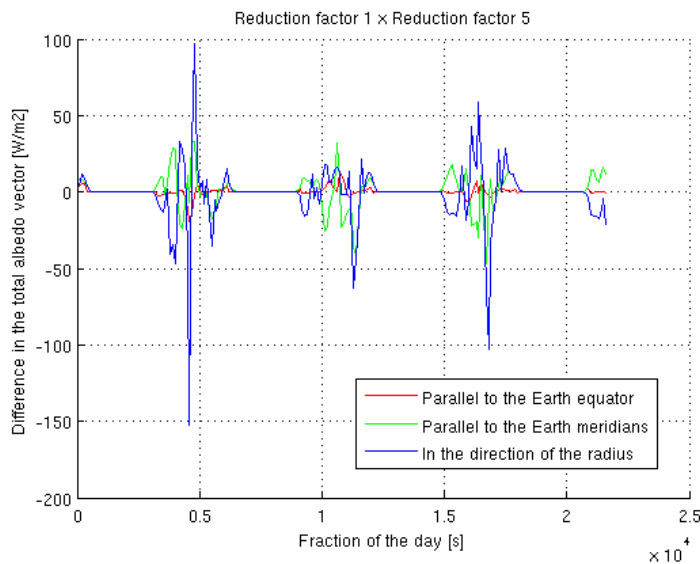


Fig. 55 – Difference in the total albedo vector (red. factor 1 x red. factor 5).

5.6- Reduction Factor 6

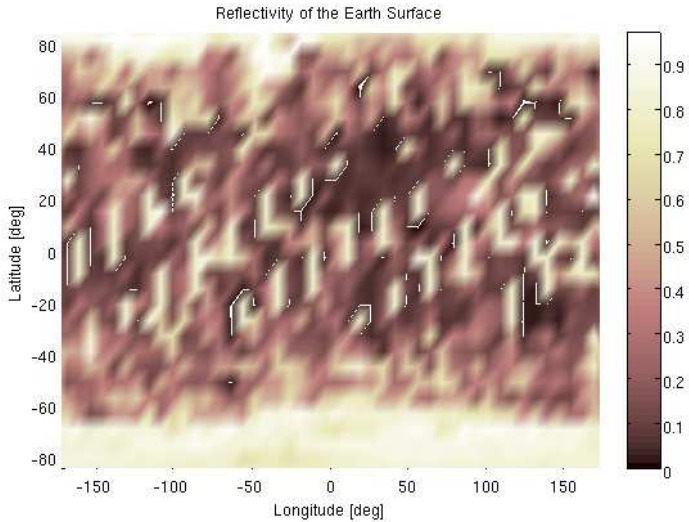


Fig. 56 – Reflectivity of the Earth Surface (red. factor 6).

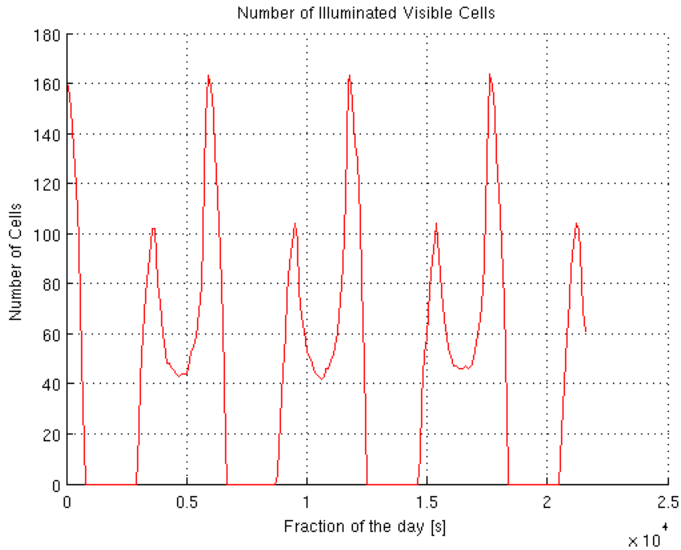


Fig. 57 – Illuminated Visible Cells (red. factor 6).

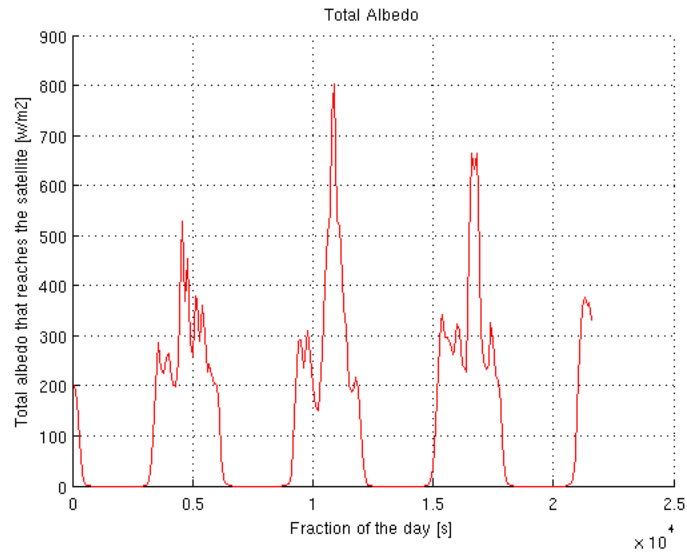


Fig. 58 – Total albedo (red. factor 6).

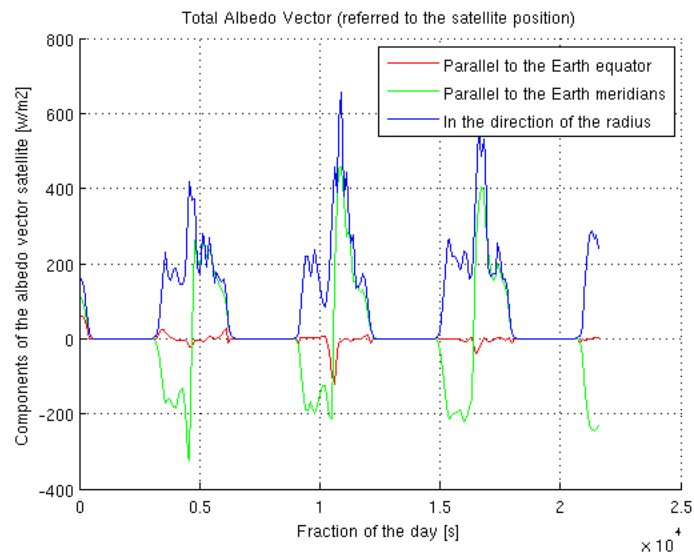


Fig. 59 – Total albedo vector (red. factor 6).

5.6.1- Reduction Factor 1 x Reduction Factor 6

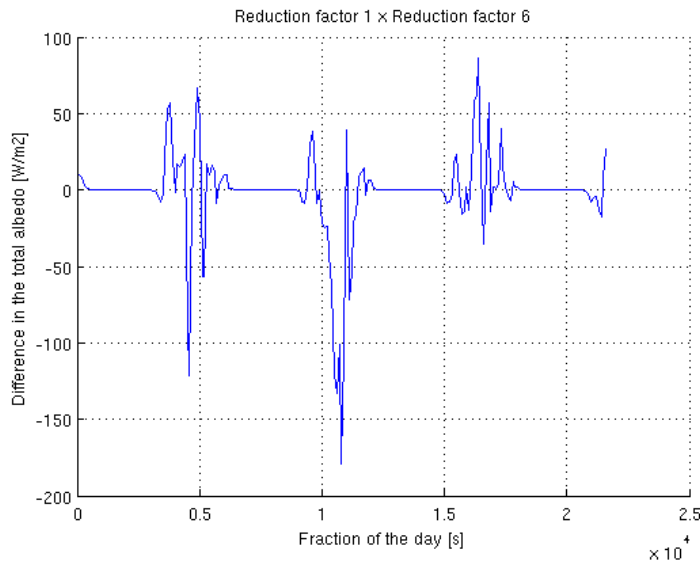


Fig. 60 – Difference in the total albedo (red. factor 1 x red. factor 6).

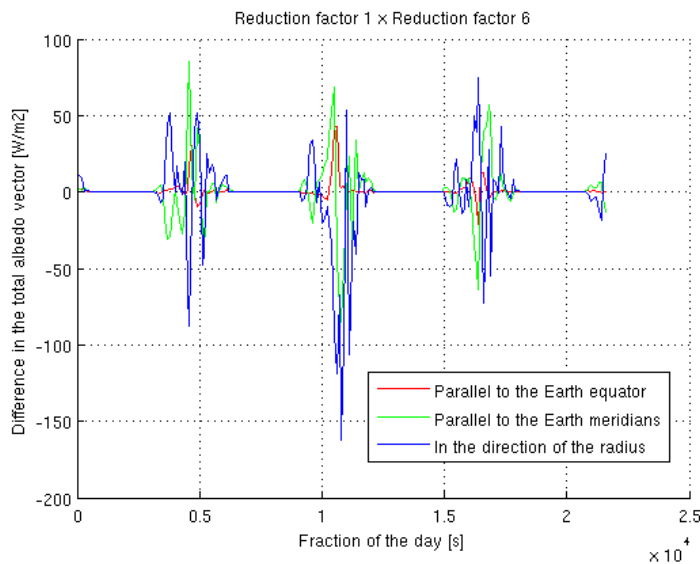


Fig. 61 – Difference in the total albedo vector (red. factor 1 x red. factor 6).

5.7- Reduction Factor 7

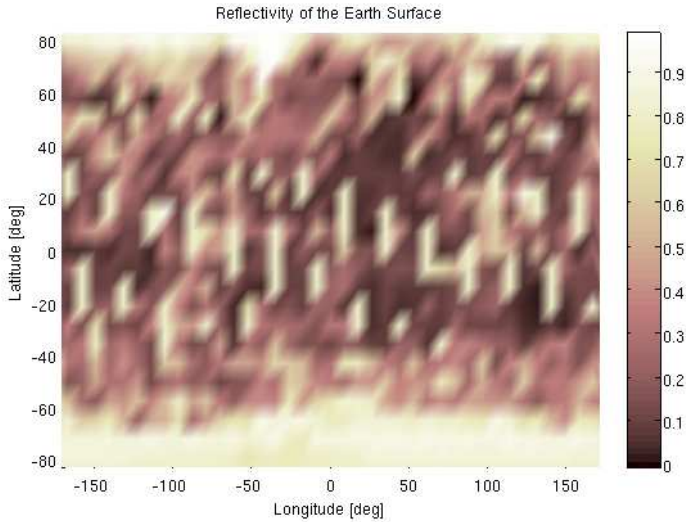


Fig. 62 – Reflectivity of the Earth Surface (red. factor 7).

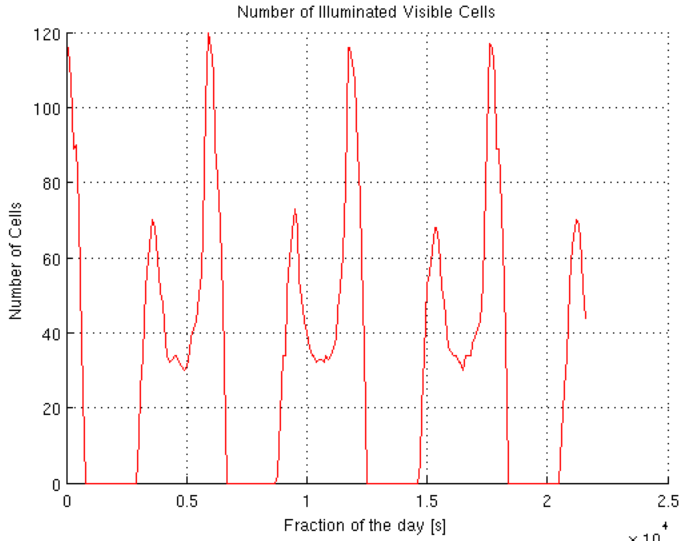


Fig. 63 – Illuminated Visible Cells (red. factor 7).

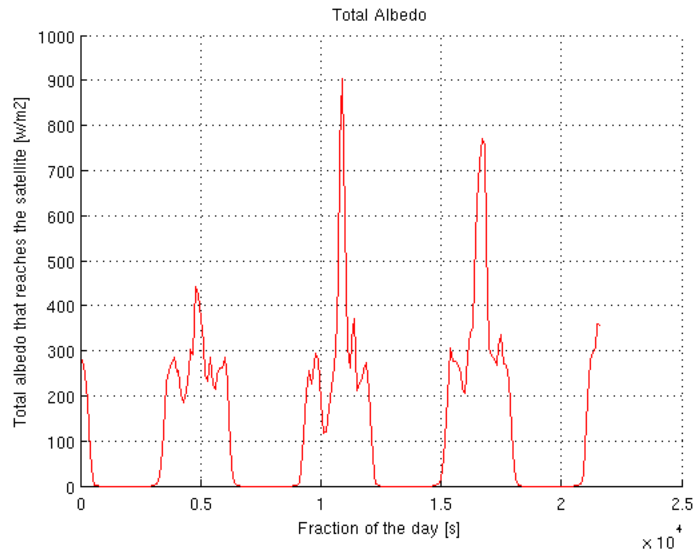


Fig. 64 – Total albedo (red. factor 7).

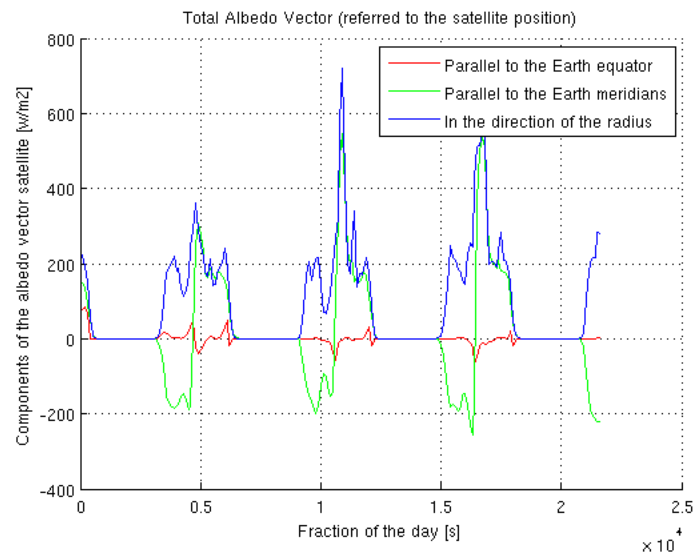


Fig. 65 – Total albedo vector (red. factor 7).

5.7.1- Reduction Factor 1 x Reduction Factor 7

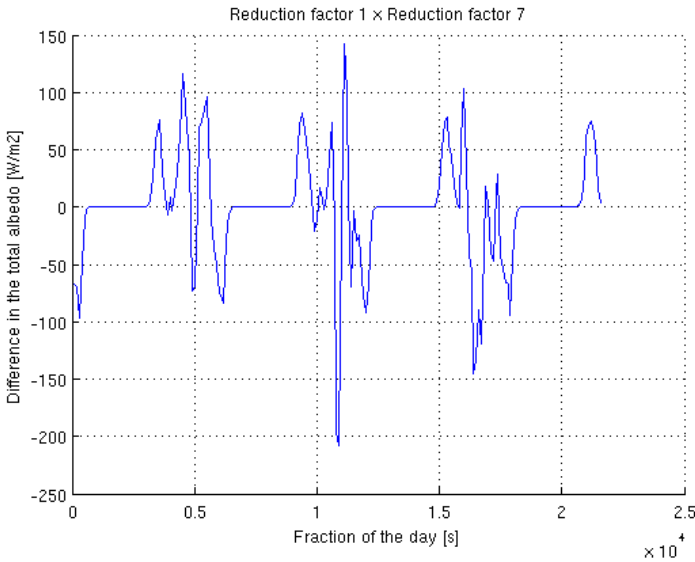


Fig. 66 – Difference in the total albedo (red. factor 1 x red. factor 7).

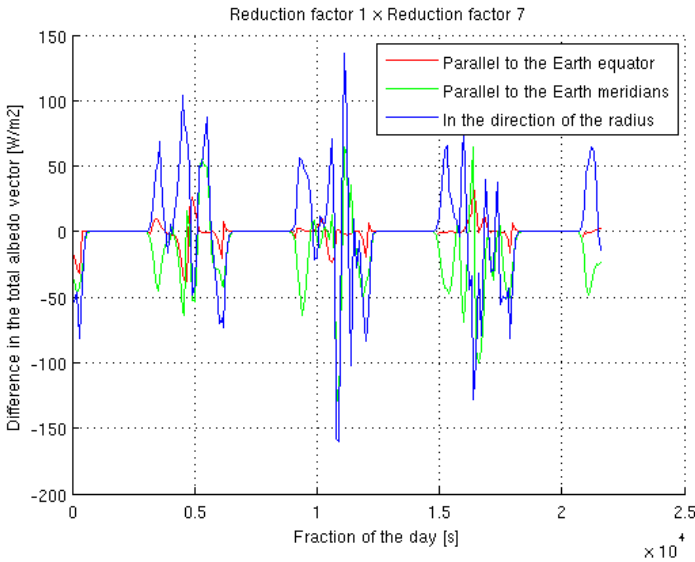


Fig. 67 – Difference in the total albedo vector (red. factor 1 x red. factor 7).

5.8- Reduction Factor 8

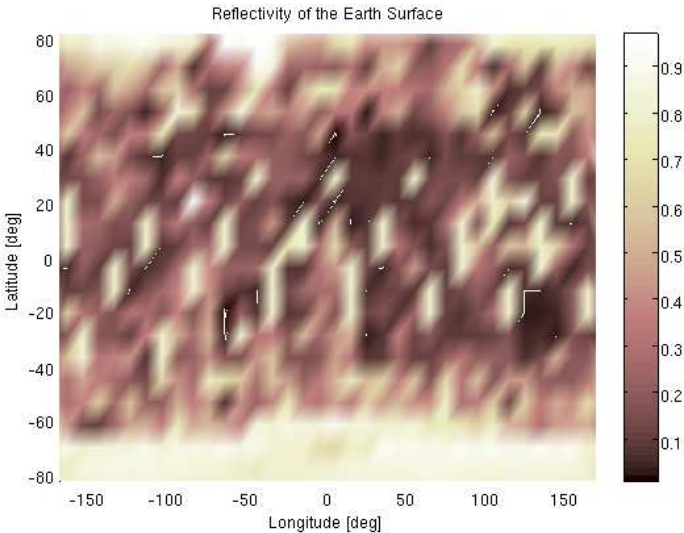


Fig. 68 – Reflectivity of the Earth Surface (red. factor 8).

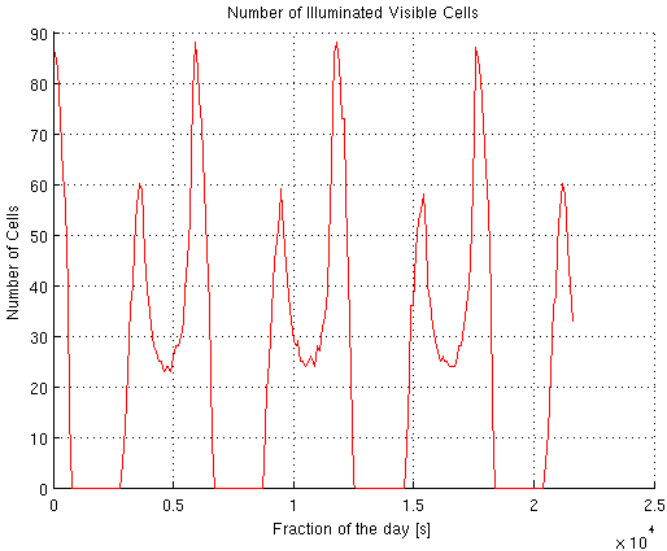


Fig. 69 – Illuminated Visible Cells (red. factor 8).

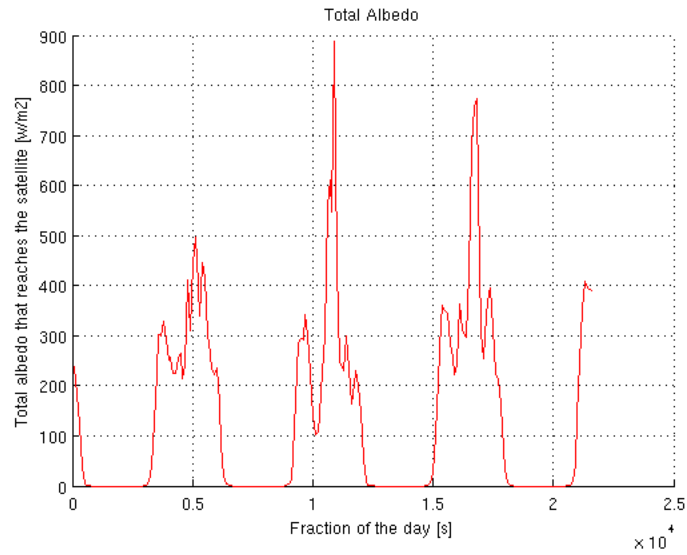


Fig. 70 – Total albedo (red. factor 8).

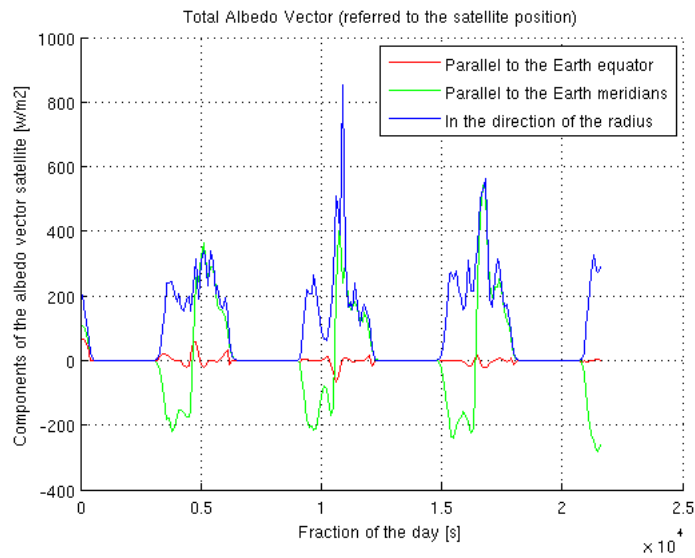


Fig. 71 – Total albedo vector (red. factor 8).

5.8.1- Reduction Factor 1 x Reduction Factor 8

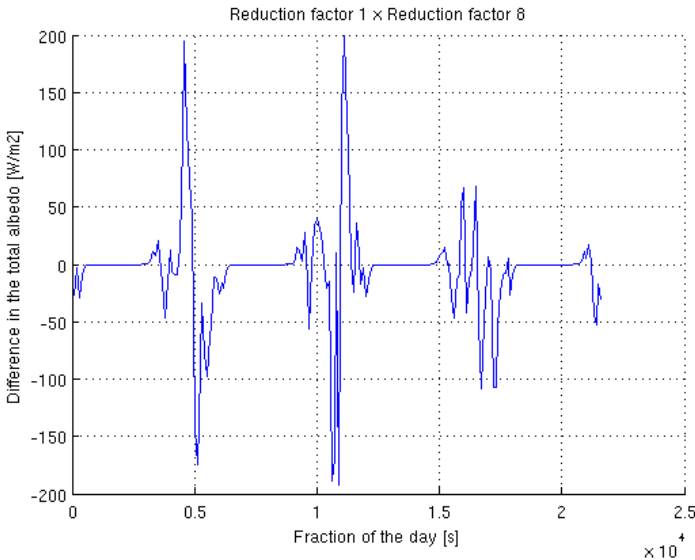


Fig. 72 – Difference in the total albedo (red. factor 1 x red. factor 8).

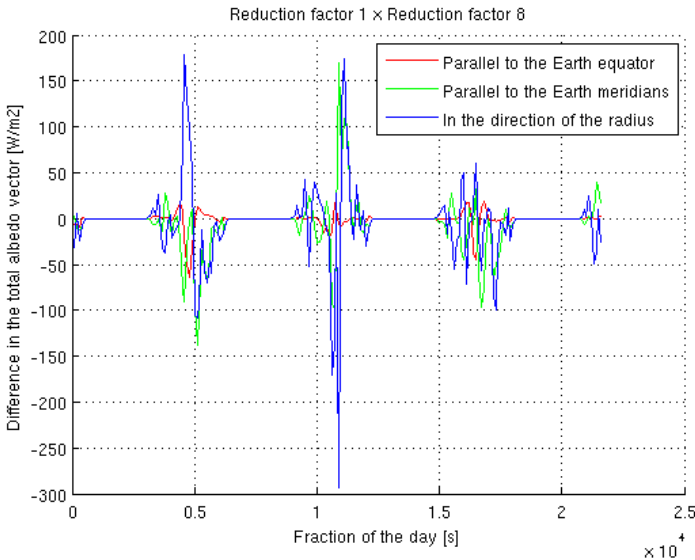


Fig. 73 – Difference in the total albedo vector (red. factor 1 x red. factor 8).

5.9- Reduction Factor 9

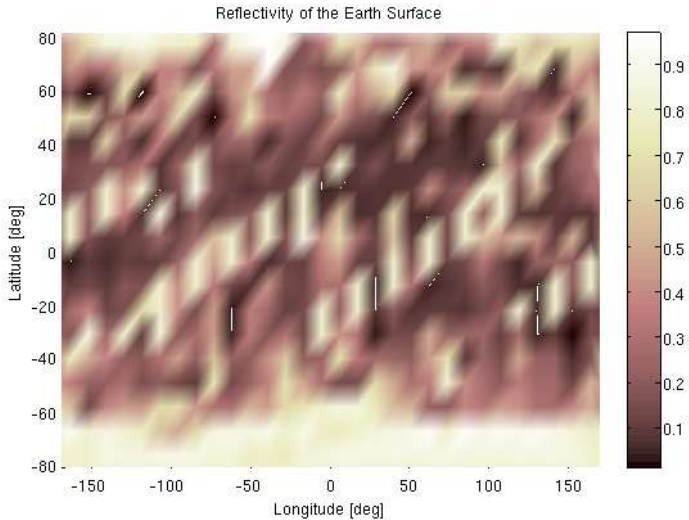


Fig. 74 – Reflectivity of the Earth Surface (red. factor 9).

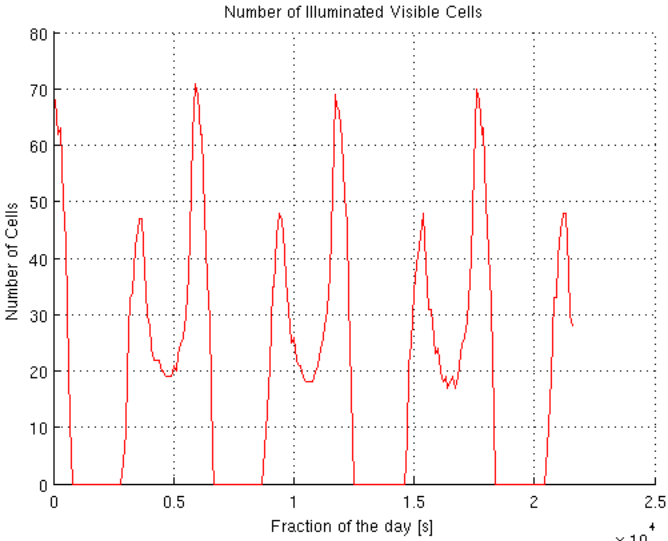


Fig. 75 – Illuminated Visible Cells (red. factor 9).

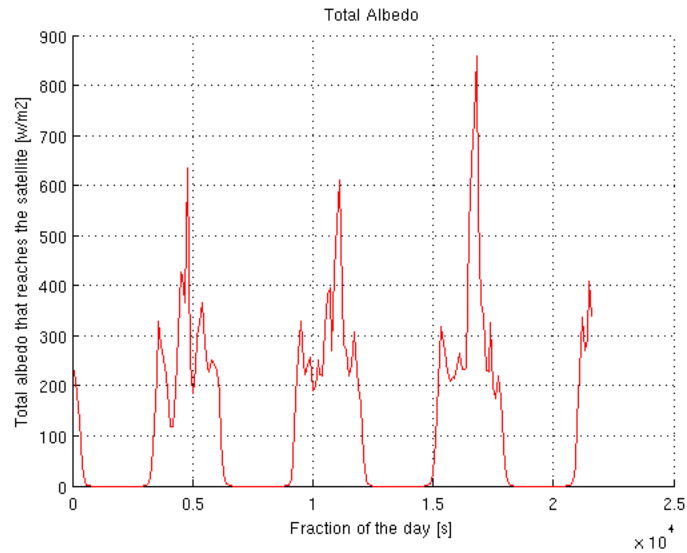


Fig. 76 – Total albedo (red. factor 9).

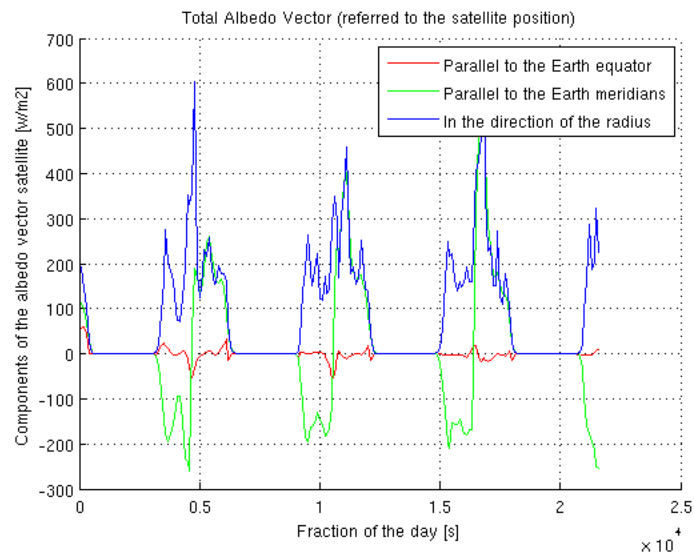


Fig. 77 – Total albedo vector (red. factor 9).

5.9.1- Reduction Factor 1 x Reduction Factor 9

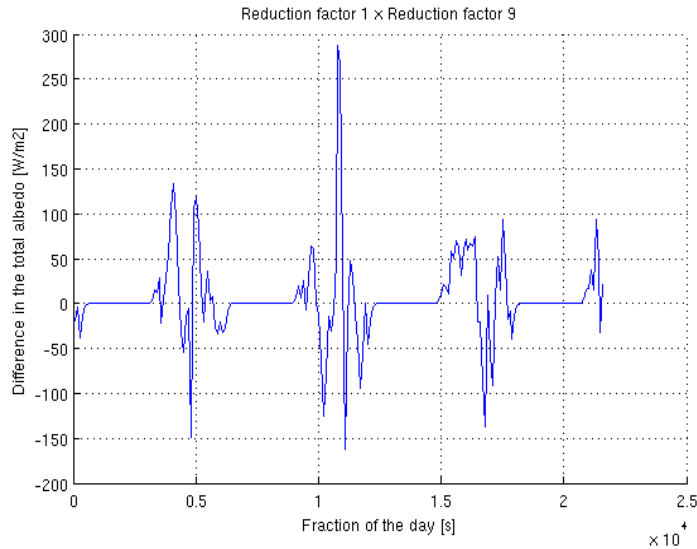


Fig. 78 – Difference in the total albedo (red. factor 1 x red. factor 9).

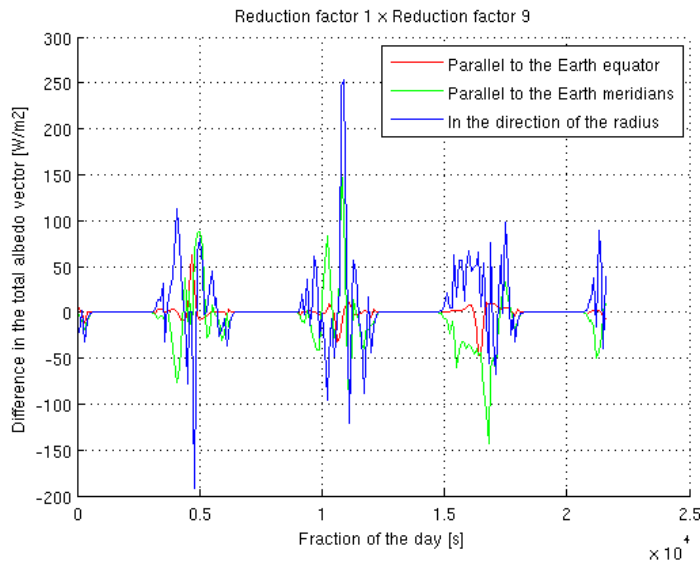


Fig. 79 – Difference in the total albedo vector (red. factor 1 x red. factor 9).

5.10- Reduction Factor 10

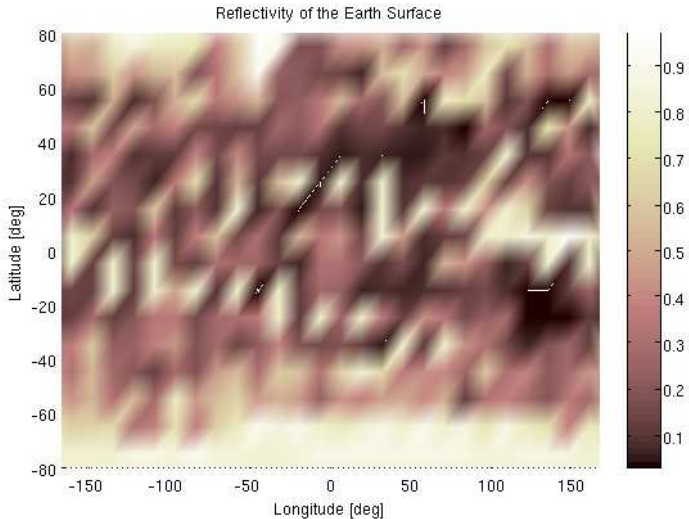


Fig. 80 – Reflectivity of the Earth Surface (red. factor 10).

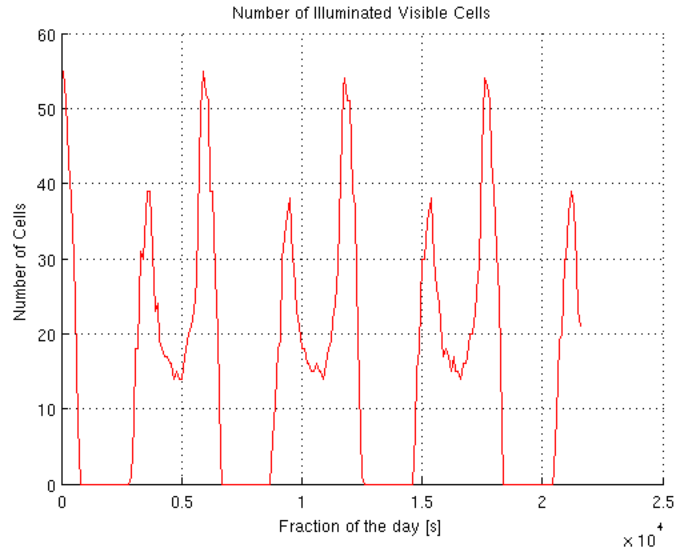


Fig. 81 – Illuminated Visible Cells (red. factor 10).

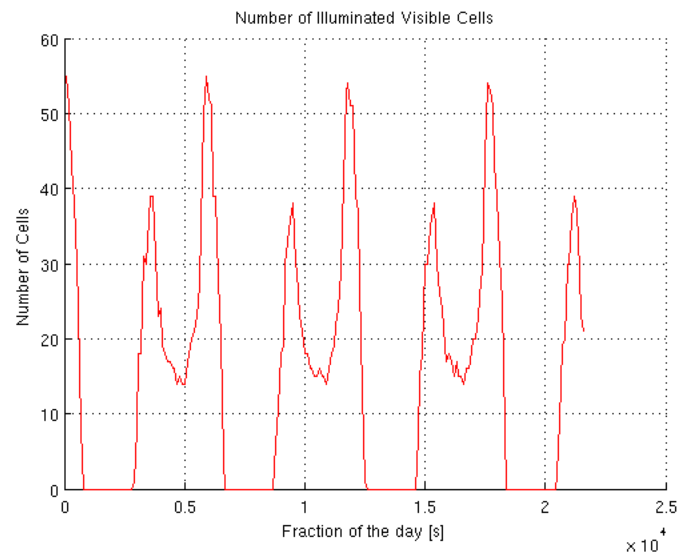


Fig. 82 – Total albedo (red. factor 10).

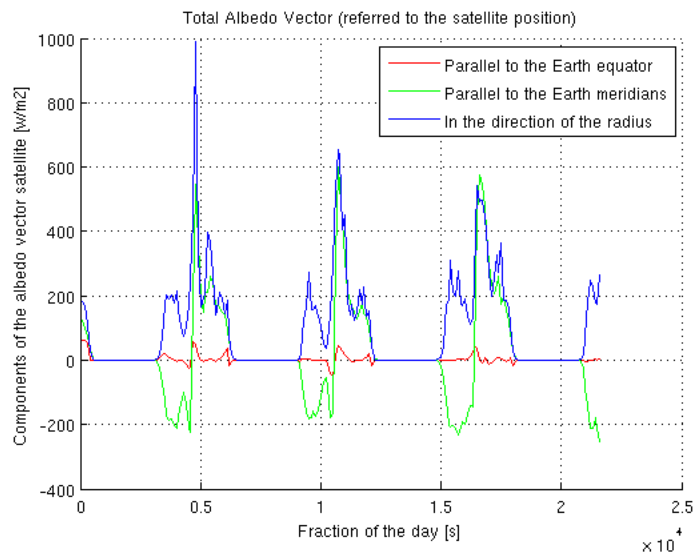


Fig. 83 – Total albedo vector (red. factor 10).

5.10.1- Reduction Factor 1 x Reduction Factor 10

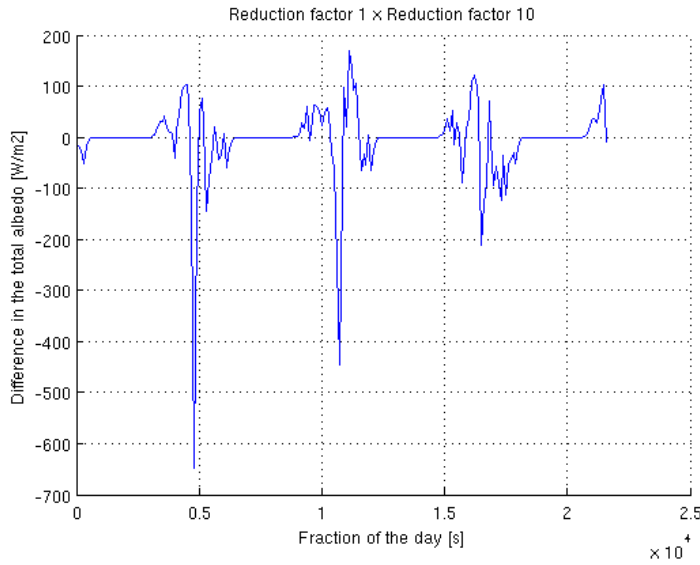


Fig. 84 – Difference in the total albedo (red. factor 1 x red. factor 10).

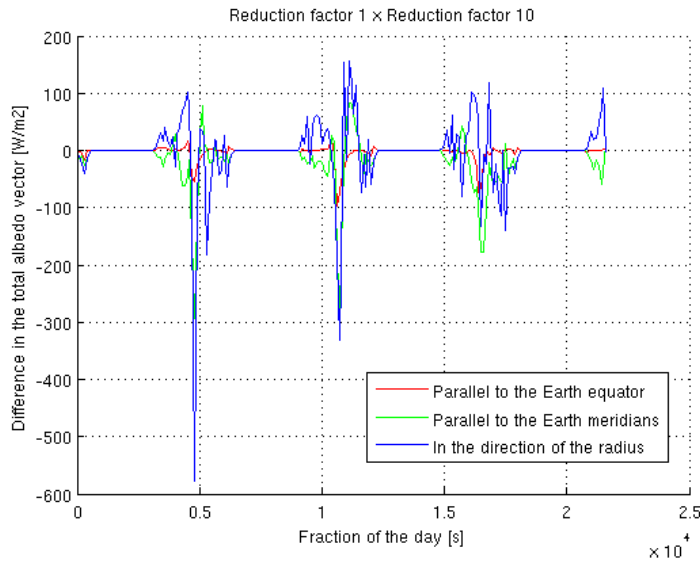


Fig. 85 – Difference in the total albedo vector (red. factor 1 x red. factor 10).

5.11- Reduction Factor 15

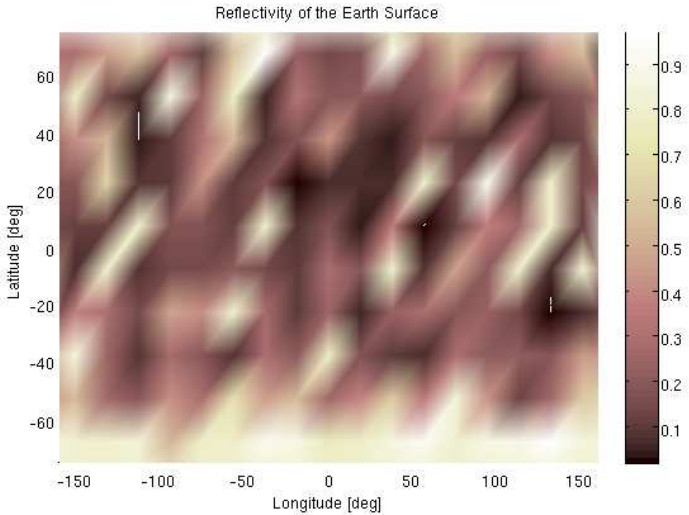


Fig. 86 – Reflectivity of the Earth Surface (red. factor 15).

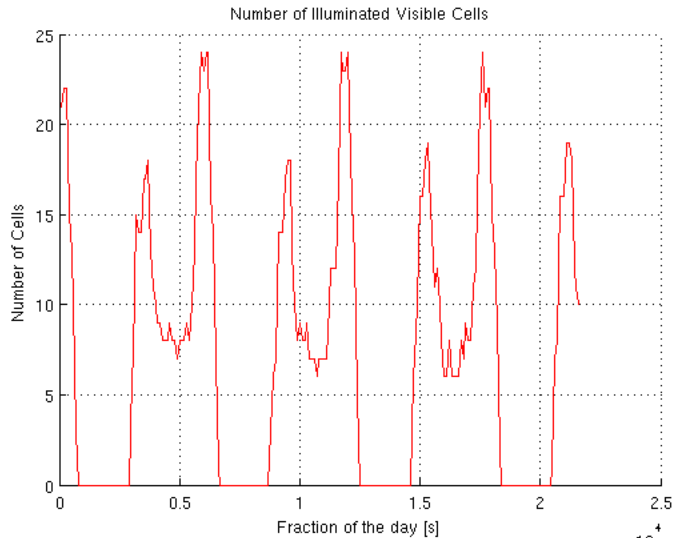


Fig. 87 – Illuminated Visible Cells (red. factor 15).

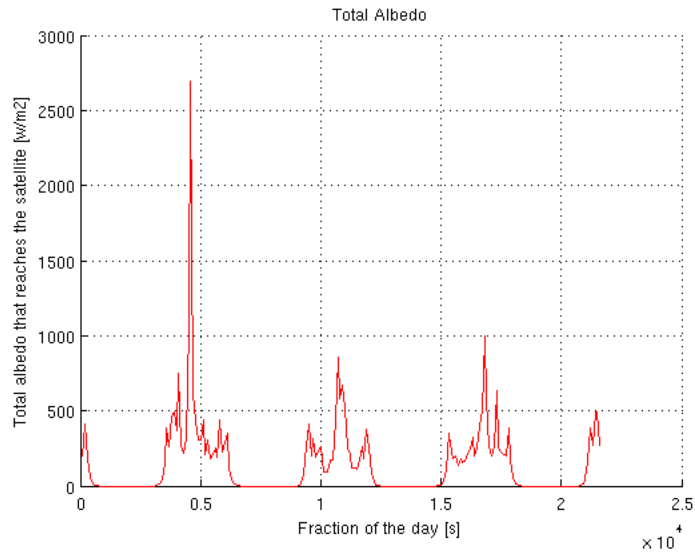


Fig. 88– Total albedo (red. factor 15).

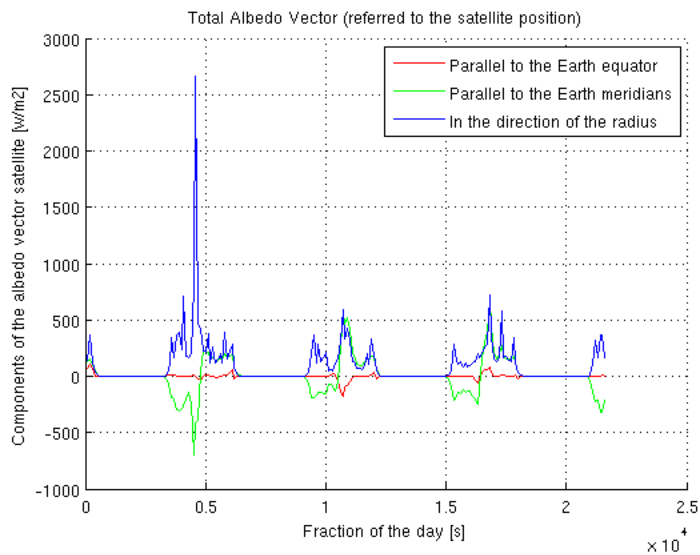


Fig. 89 – Total albedo vector (red. factor 15).

5.11.1- Reduction Factor 1 x Reduction Factor 15

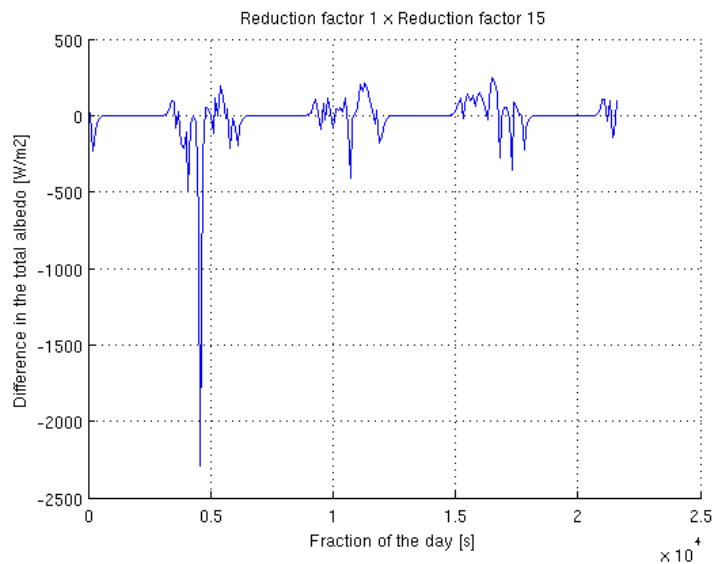


Fig. 90 – Difference in the total albedo (red. factor 1 x red. factor 15).

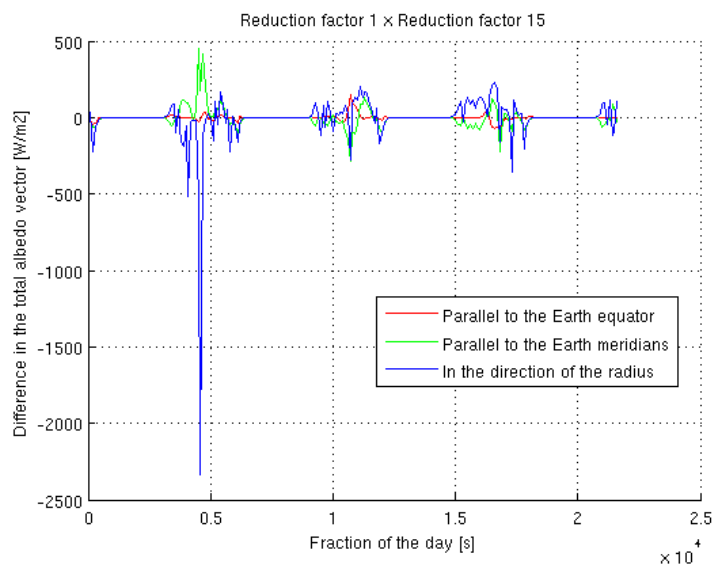


Fig. 91 – Difference in the total albedo vector (red. factor 1 x red. factor 15).

5.12- Reduction Factor 20

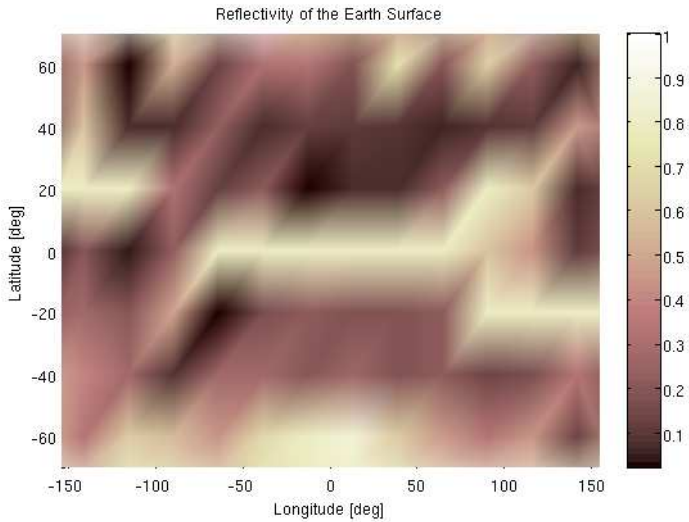


Fig. 92 – Reflectivity of the Earth Surface (red. factor 20).

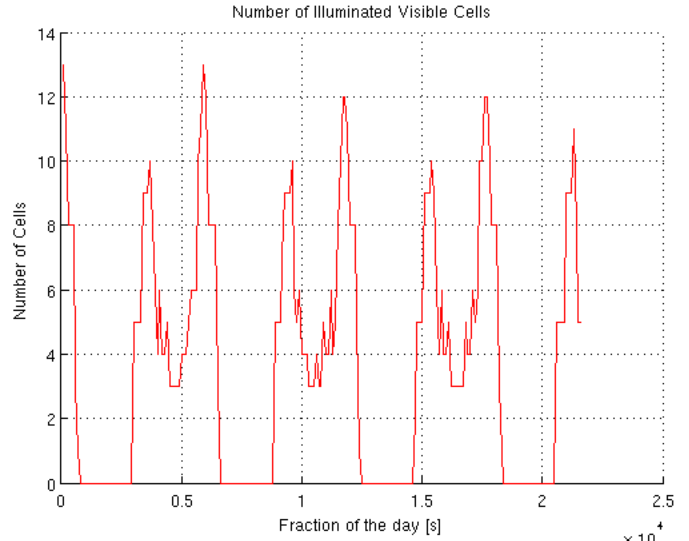


Fig. 93 – Illuminated Visible Cells (red. factor 20).

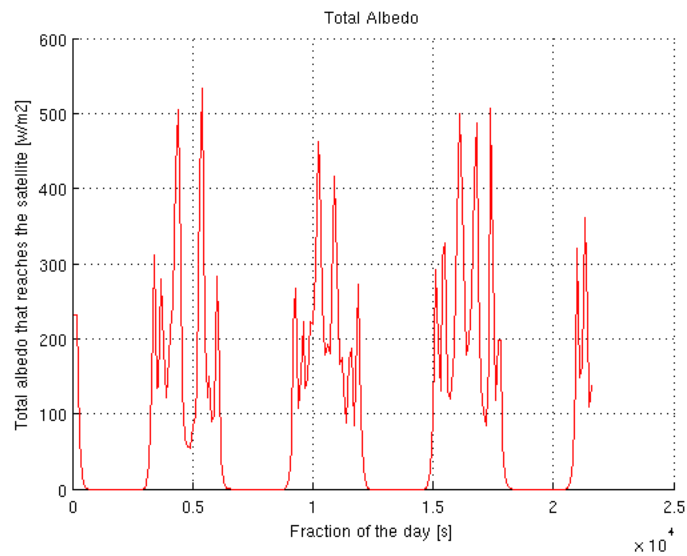


Fig. 94– Total albedo (red. factor 20).

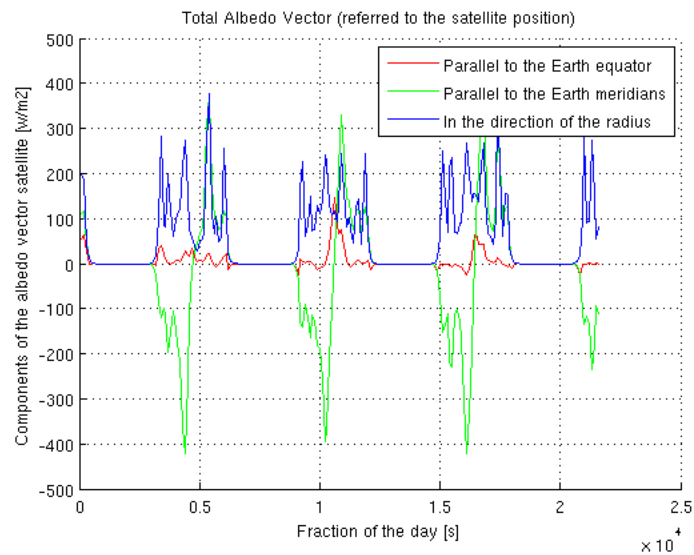


Fig. 95 – Total albedo vector (red. factor 20).

5.12.1- Reduction Factor 1 x Reduction Factor 20

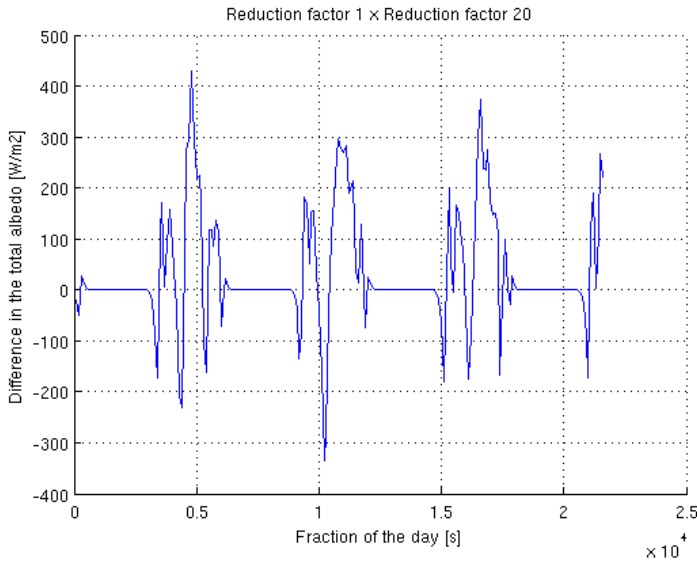


Fig. 96 – Difference in the total albedo (red. factor 1 x red. factor 20).

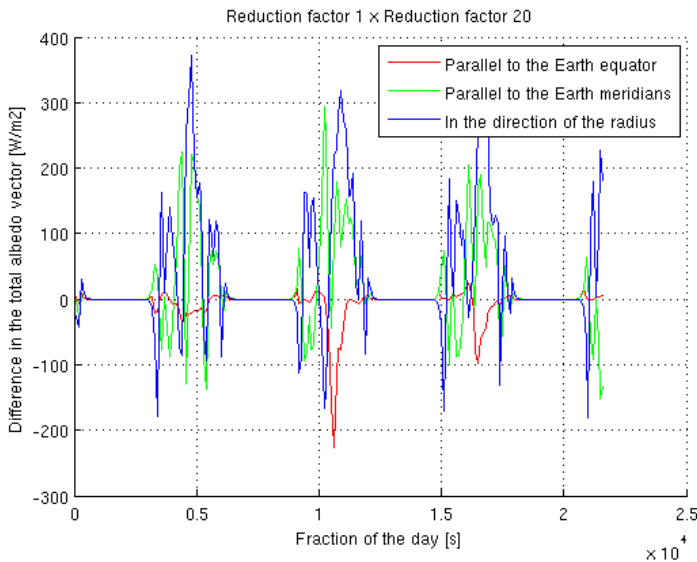


Fig. 97 – Difference in the total albedo vector (red. factor 1 x red. factor 20).

5.13- Directions of the Albedo Vector Considering Different Resolutions

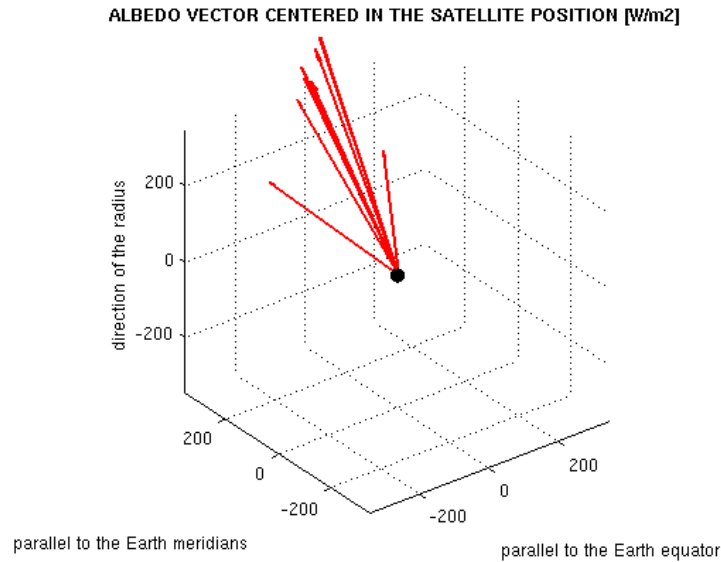


Fig. 98 – Albedo vector considering different resolutions.

Therefore, it is clear that it is not possible to use other resolution than the maximum if the goal is to obtain a good accuracy. Otherwise the results obtained with the model could be corrupted. The variation of the albedo vector using different resolutions is not only in the absolute value of the vector but also in the direction of the albedo that reaches the satellite.

6- Results Obtained Considering Different Altitudes

A study shown the full albedo irradiance considering different altitudes of the satellite is presented in Figures 99 to 106

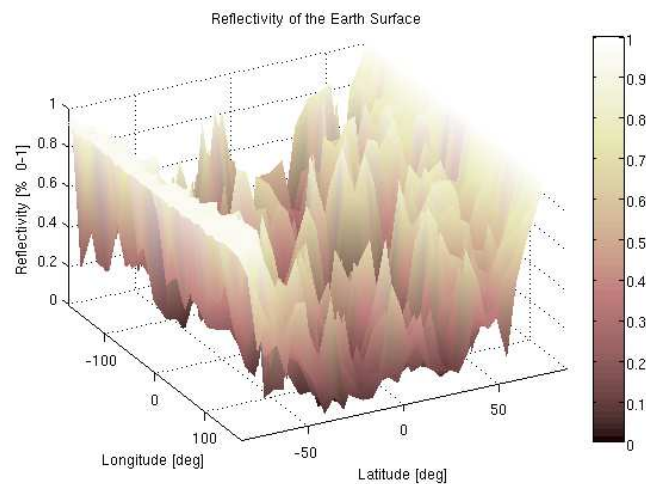


Fig. 99 – Reflectivity.

6.1- The Full Albedo Irradiance for the Altitude Around 100 km

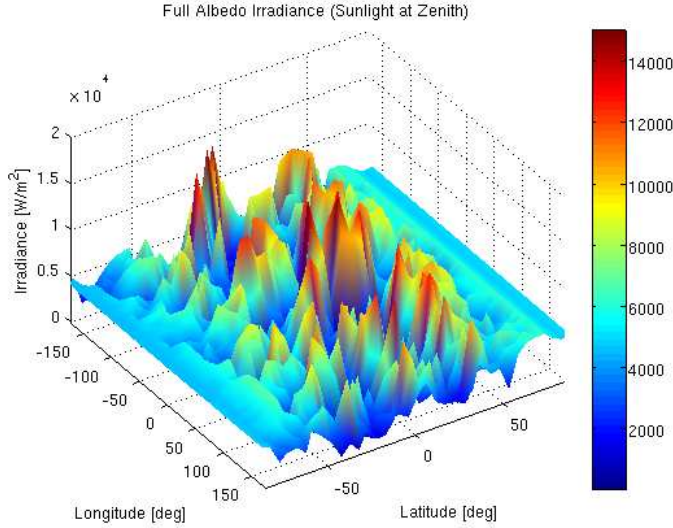


Fig. 100 – Full albedo irradiance around 100 km.

6.2- The Full Albedo Irradiance for the Altitude Around 200 km

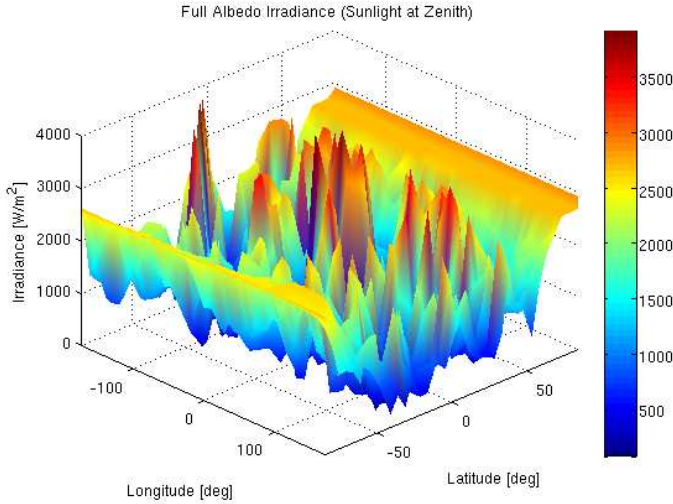


Fig. 101 – Full albedo irradiance around 200 km.

6.3- The Full Albedo Irradiance for the Altitude Around 300 km

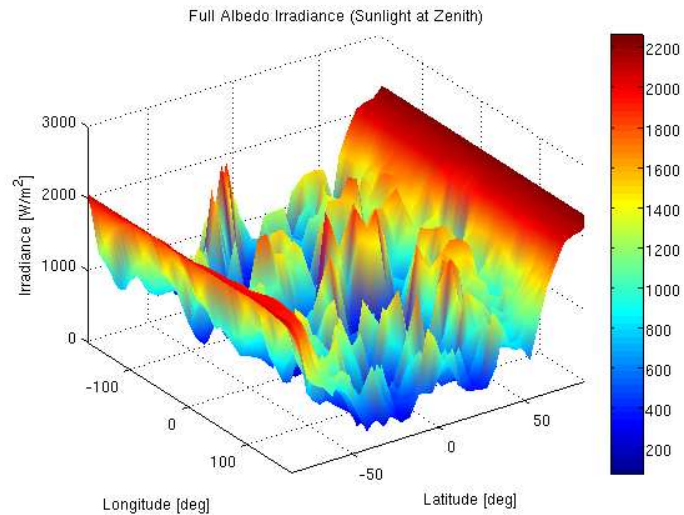


Fig. 102 – Full albedo irradiance around 300 km.

6.4- The Full Albedo Irradiance for the Altitude Around 500 km

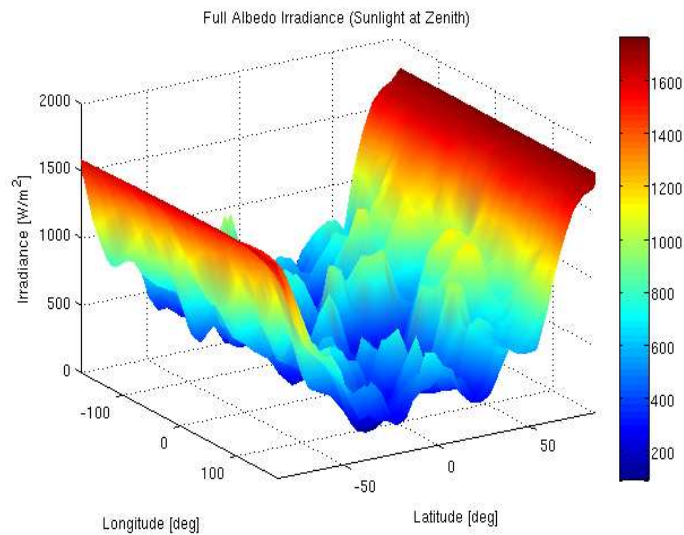


Fig. 103 – Full albedo irradiance around 500 km.

6.5- The Full Albedo Irradiance for the Altitude Around 700 km

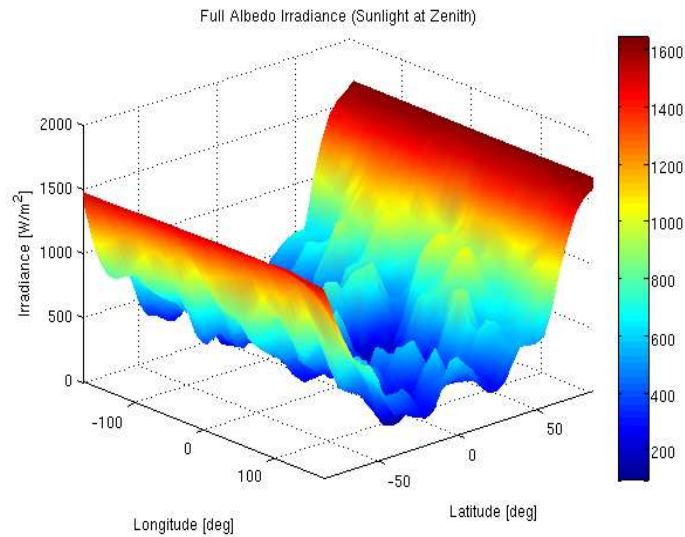


Fig. 104 – Full albedo irradiance around 700 km.

6.6- The Full Albedo Irradiance for the Altitude Around 1000 km

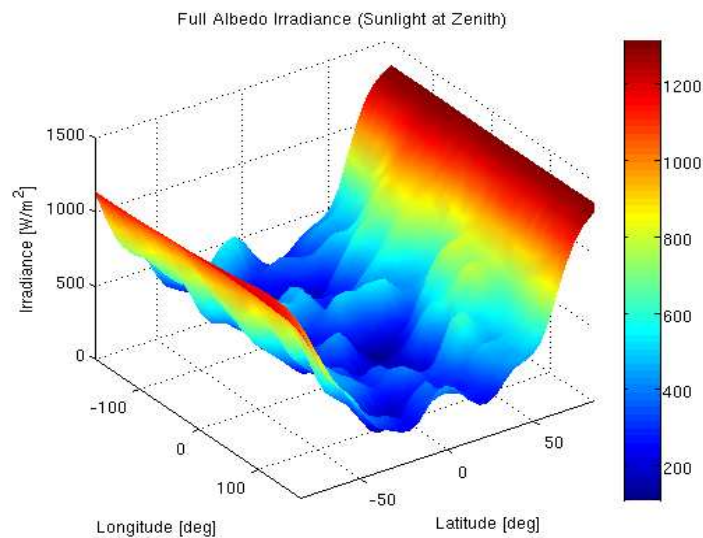


Fig. 105 – Full albedo irradiance around 1000 km.

6.7- The Full Albedo Irradiance for the Altitude Around 36000 km

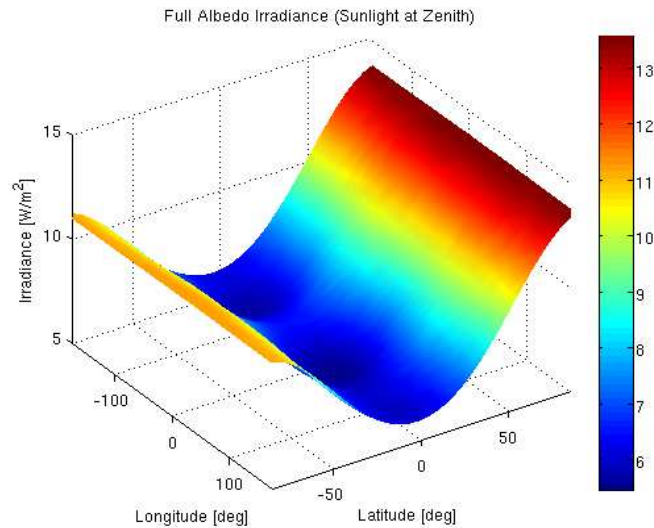


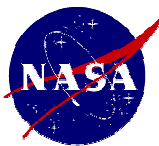
Fig. 106 – Full albedo irradiance around 36000 km.

7- Evaluation of the Terrestrial Albedo Applied to Some Scientific Missions

7.1- Gravity Probe B

The Gravity Probe B (GP-B) is a drag-free satellite designed to measure two predictions of Einstein's general theory of relativity by monitoring the orientations of ultra-sensitive gyroscopes relative to a distant guide star. The spacecraft was launched on April 20, 2004.

The graphs (Figures 107 to 123) were obtained considering the nominal orbit of the Gravity Probe B in April 20, 2004. The graphs show the simulation for the entire day.



Nominal Orbit:

$$a = 7027.4 \text{ km}$$

$$e = 0.0014$$

$$i = 90.007^\circ$$

$$\omega = 71.3^\circ$$

$$\Omega = 163.26^\circ$$

GRAVITY PROBE B

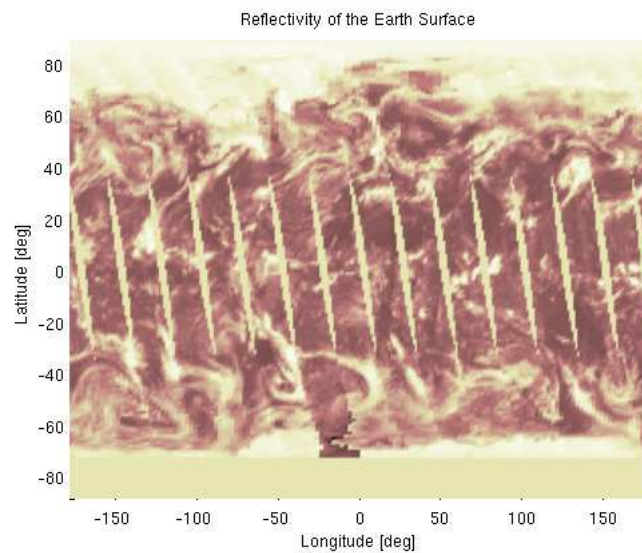


Fig. 107 – Reflectivity of the Earth in 20/04/2004.

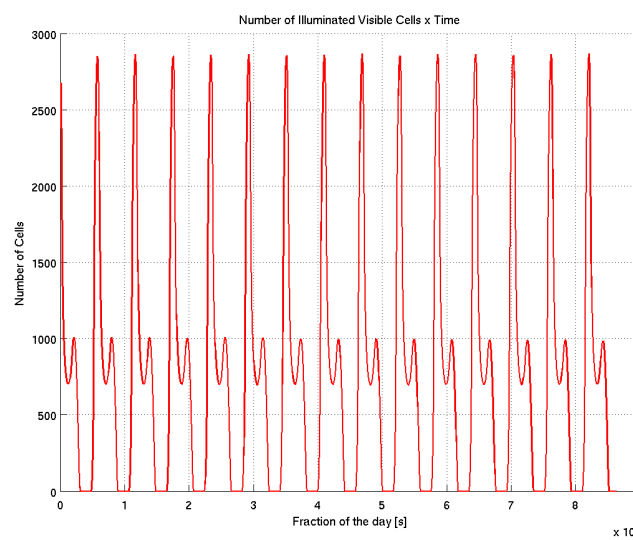


Fig. 108 – Number of illuminated visible cells (GP-B).

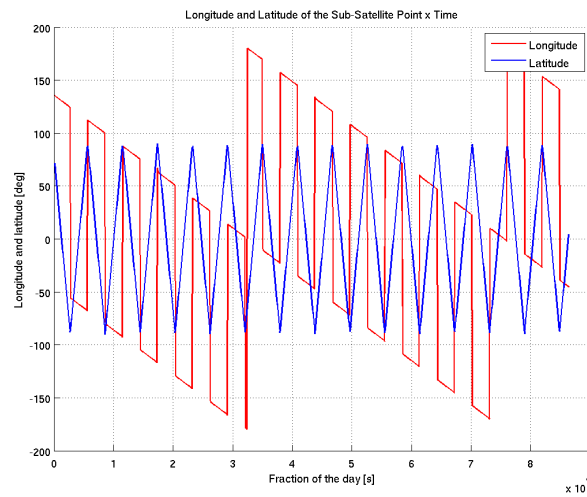


Fig. 109 – Longitude and latitude of the sub-satellite point (GP-B).

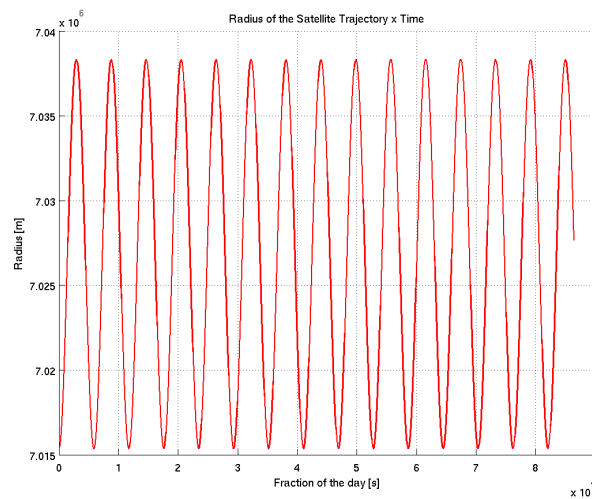


Fig. 110 – Radius of the satellite trajectory (GP-B).

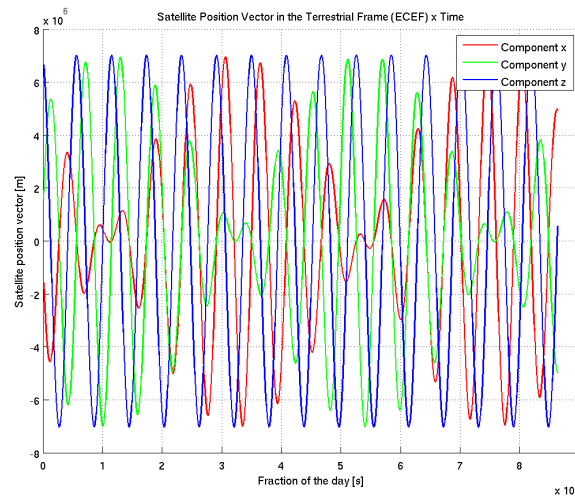


Fig. 111 – Satellite position vector in the terrestrial frame (GP-B).

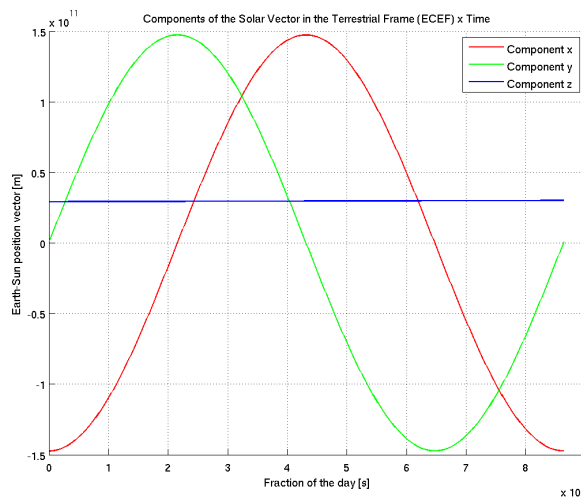


Fig. 112 – Sun position vector in the terrestrial frame.

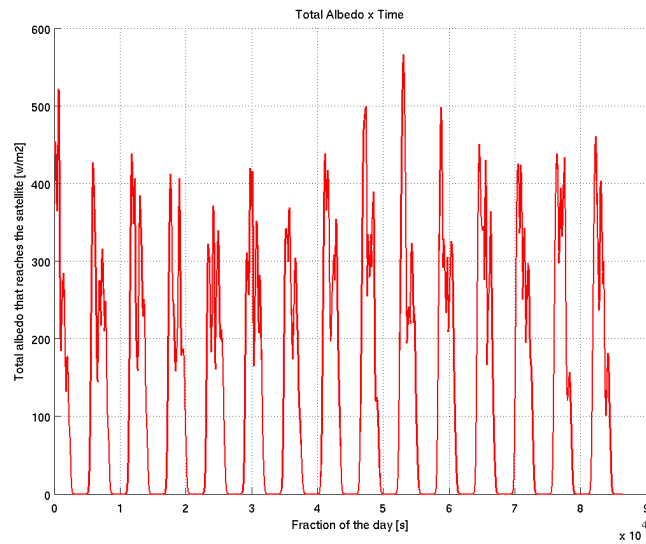


Fig. 113 – Total albedo (GP-B).

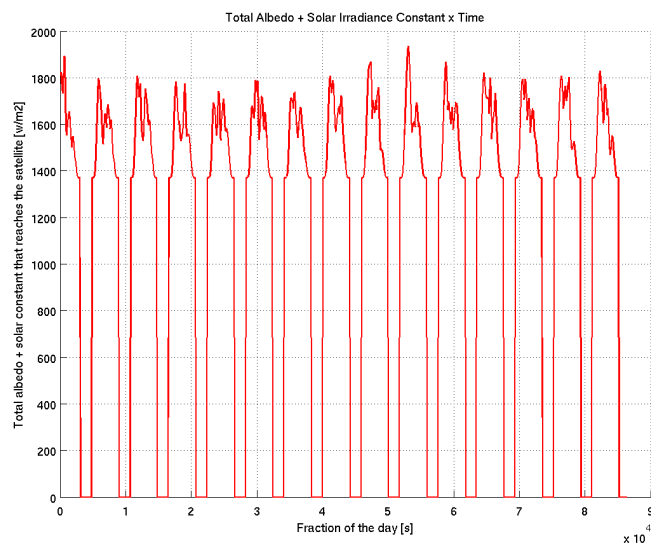


Fig. 114 – Total albedo + solar constant (GP-B).

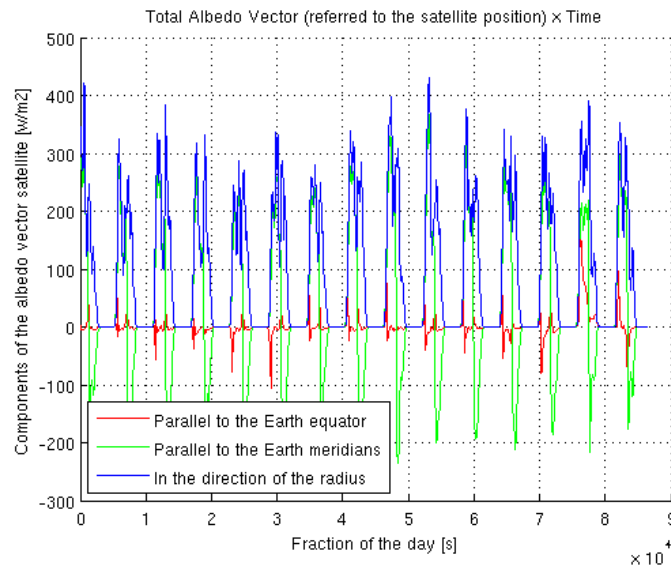


Fig. 115 – Total albedo vector (GP-B).

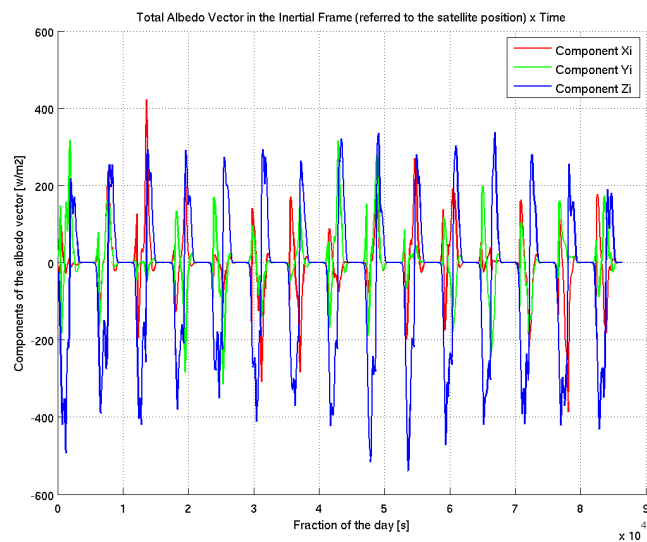


Fig. 116 – Total albedo vector in the inertial frame (GP-B).

7.1.1- Evaluation of the Trajectory Deviation for the GP-B

To evaluate the deviation of the trajectory, a model describing the satellite shape and a description of the optical properties of the surfaces are necessary. In this work was used a simple model for the optical properties and was assumed a spherical satellite shape with a mass of 1000 kg, just to test the Earth Albedo Model. Thus, considering that 30% of the energy that reaches the satellite is absorbed, 30% is reflected specularly, 40% is reflected diffusely, and using the model presented in Harris et al. 1969 (NASA SP-8027) the force applied can be calculated. The effect in the trajectory is shown in Figures 118 to 123.

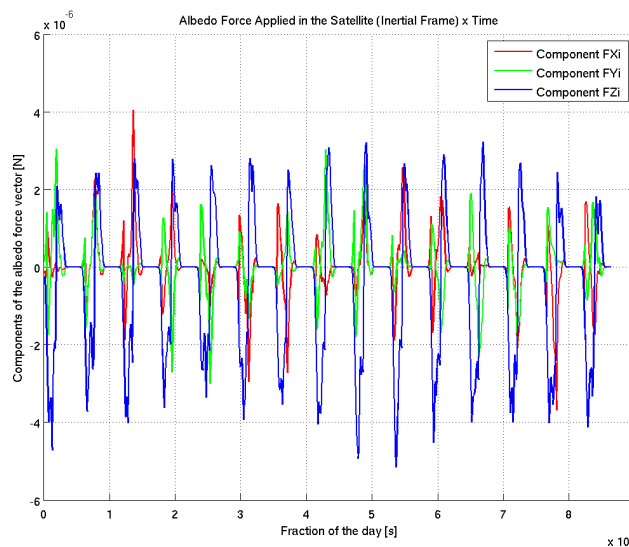


Fig. 117 – Albedo force vector applied in the satellite (GP-B).

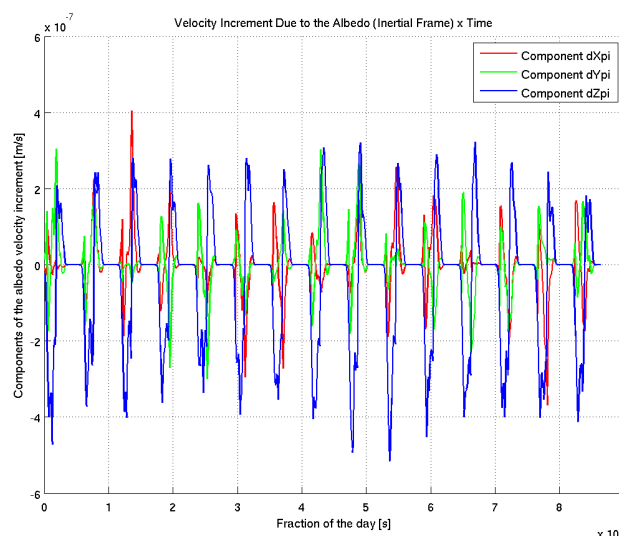


Fig. 118 – Velocity increment due to the albedo (GP-B).

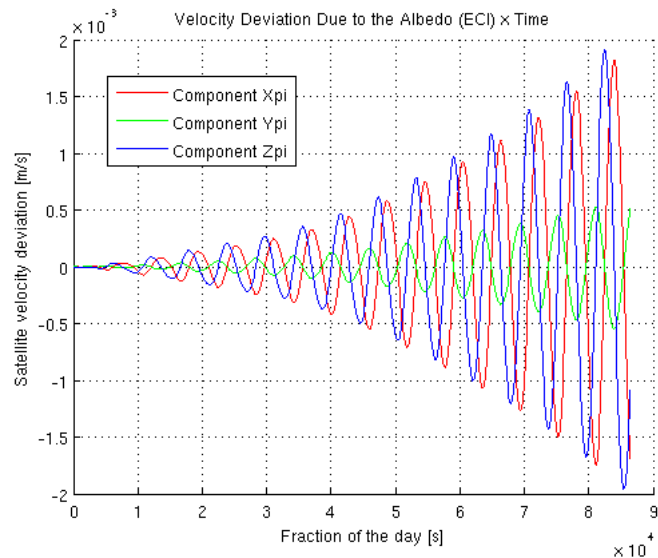


Fig. 119 – Velocity deviation for the entire day (GP-B).

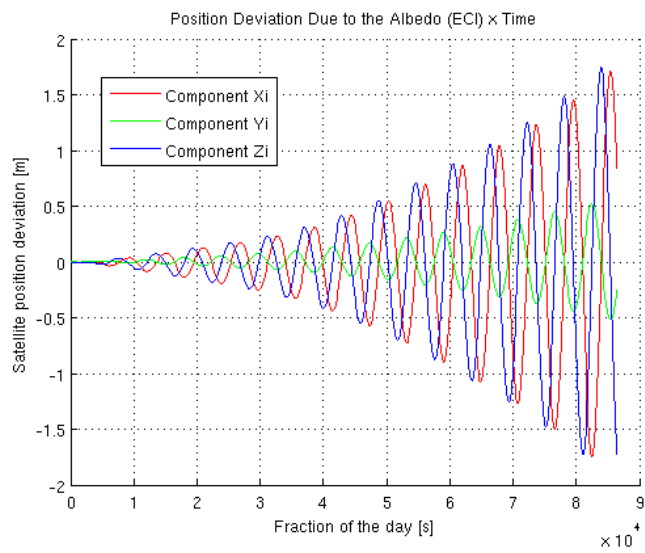


Fig. 120 – Position deviation for the entire day (GP-B).

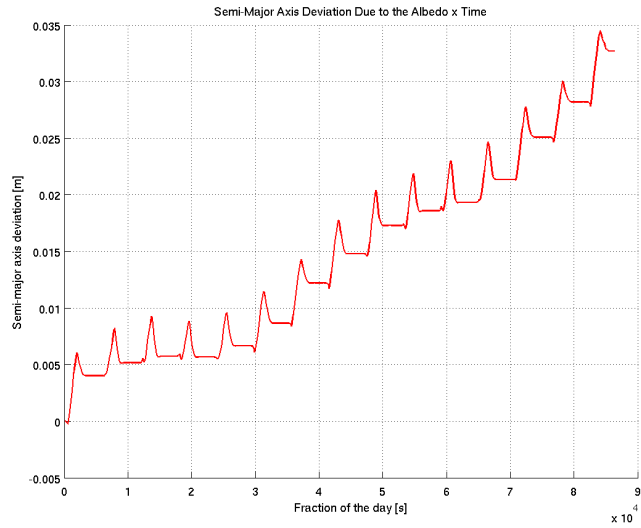


Fig. 121 – Semi-major axis deviation for the entire day (GP-B).

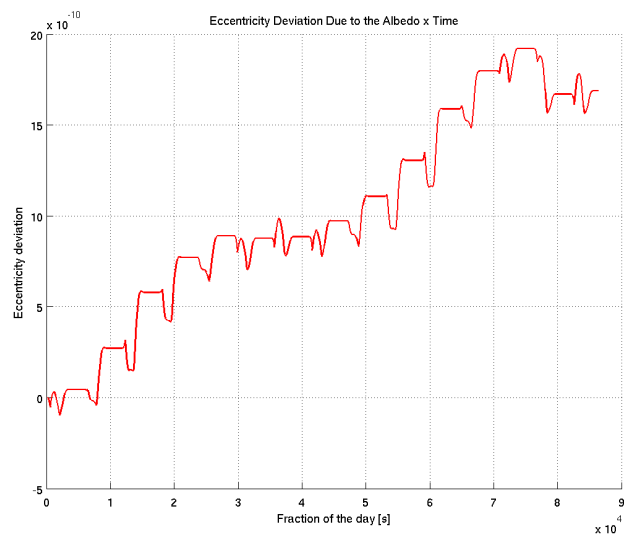


Fig. 122 – Eccentricity deviation for the entire day (GP-B).

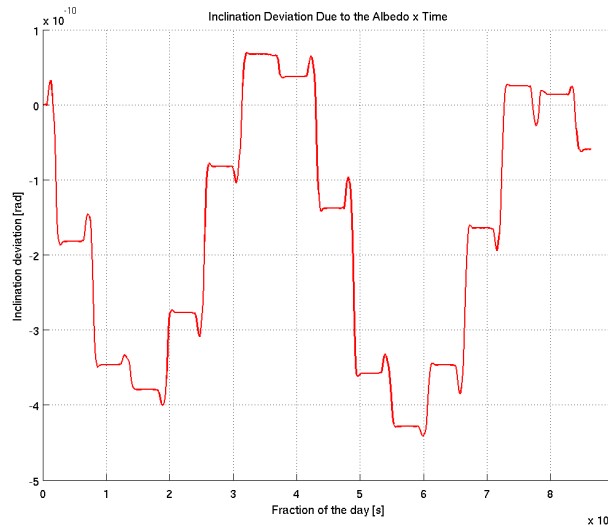


Fig. 123 – Inclination deviation for the entire day (GP-B).

7.2- Microscope Mission

The MICROSCOPE mission was designed to test the Equivalence Principle with a high accuracy (10^{-15}) using two proof-masses of different material on precisely the same orbit and measure any difference in the forces required to maintain the common orbit.

The graphs (Figures 124 to 140) were obtained considering the nominal orbit of the Microscope Mission and the Earth reflectivity in April 20, 2004. The graphs show the simulation for the entire day.



Nominal Orbit:

$$a = 7108 \text{ km}$$

$$e = 0.0012$$

$$i = 98.2^\circ$$

$$\omega = 0$$

$$\Omega = 110^\circ$$

MICROSCOPE

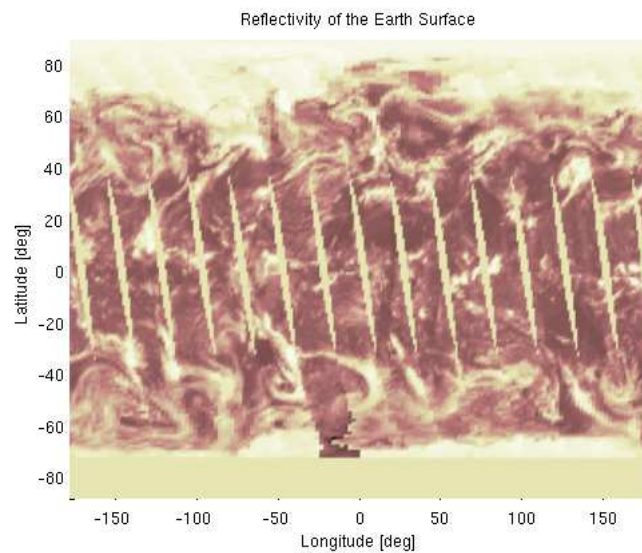


Fig. 124 – Reflectivity of the Earth in 20/04/2004.

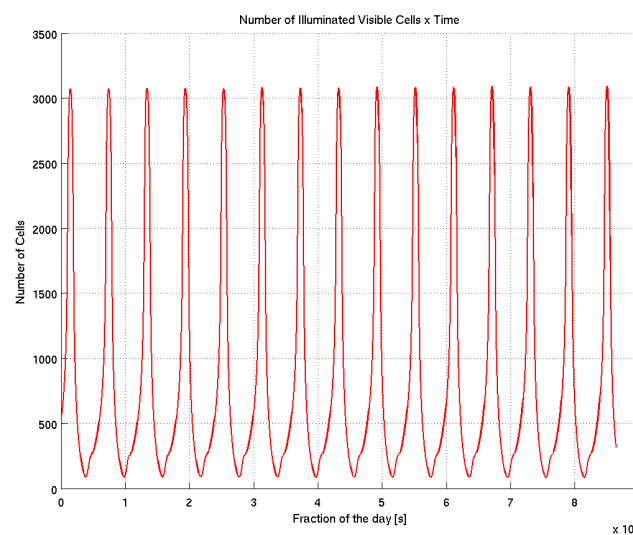


Fig. 125 – Number of illuminated visible cells (Microscope).

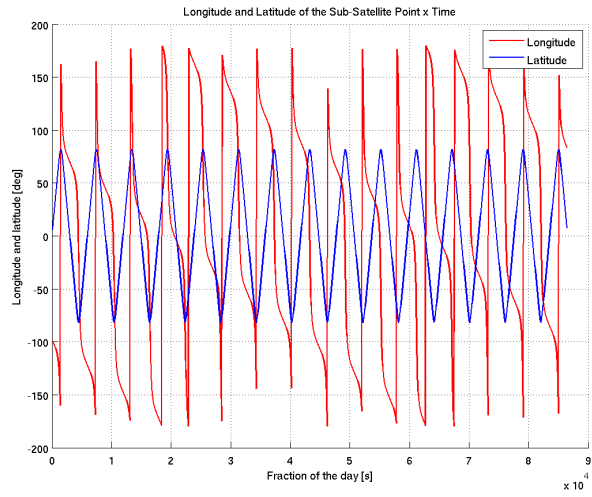


Fig. 126 – Longitude and latitude of the sub-satellite point (Microscope).

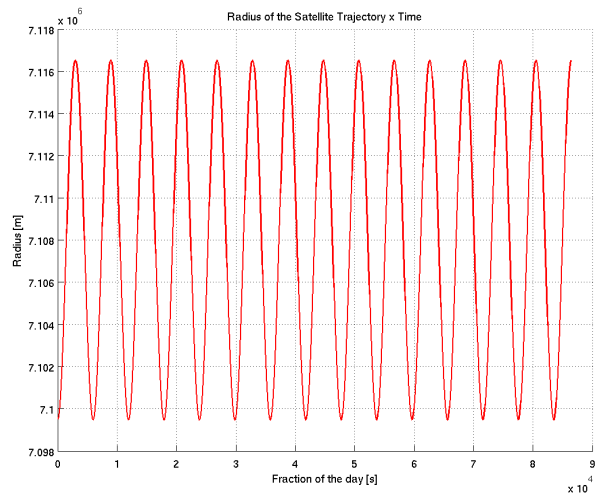


Fig. 127 – Radius of the satellite trajectory (Microscope).

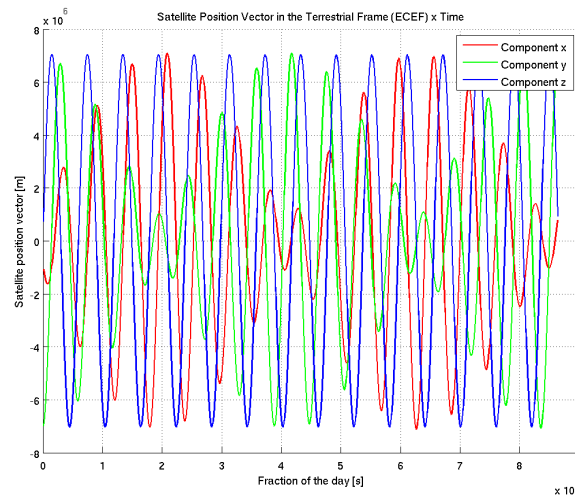


Fig. 128 – Satellite position vector in the terrestrial frame (Microscope).

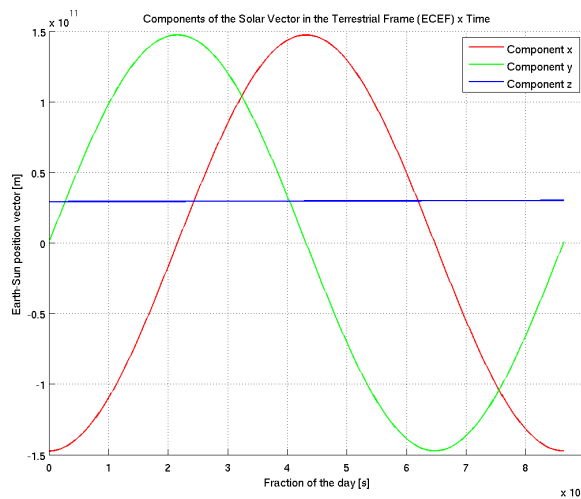


Fig. 129 – Sun position vector in the terrestrial frame.

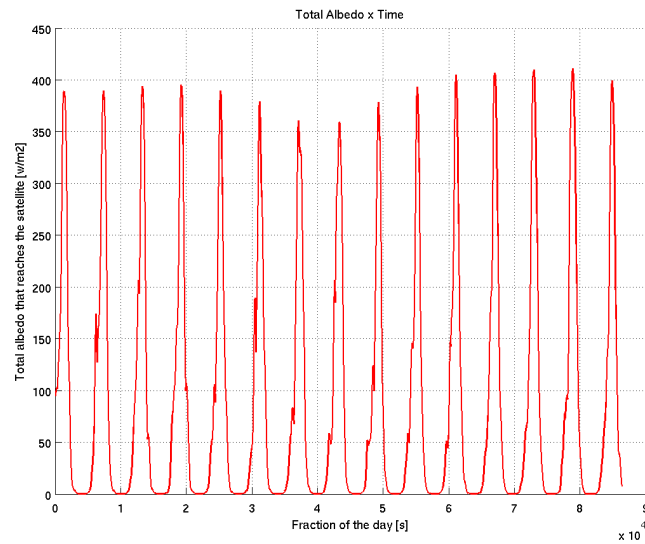


Fig. 130 – Total albedo (Microscope).

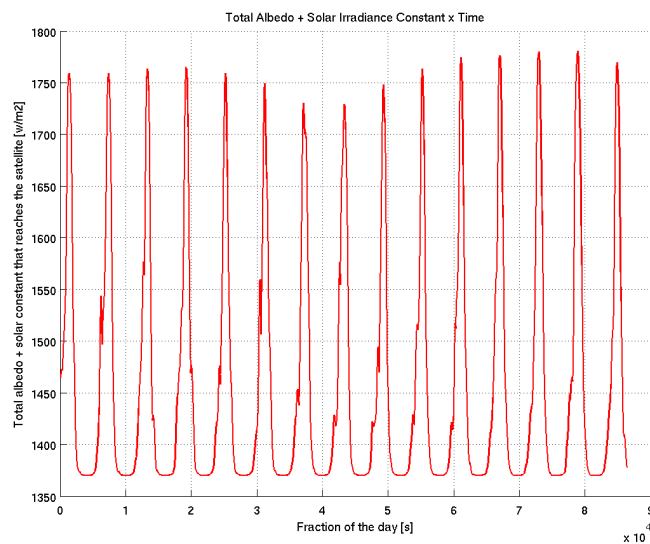


Fig. 131 – Total albedo + solar constant (Microscope).

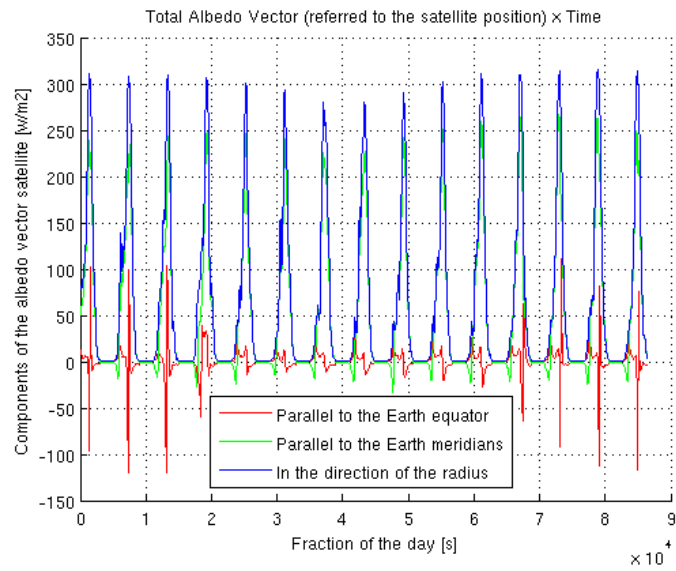


Fig. 132 – Total albedo vector (Microscope).

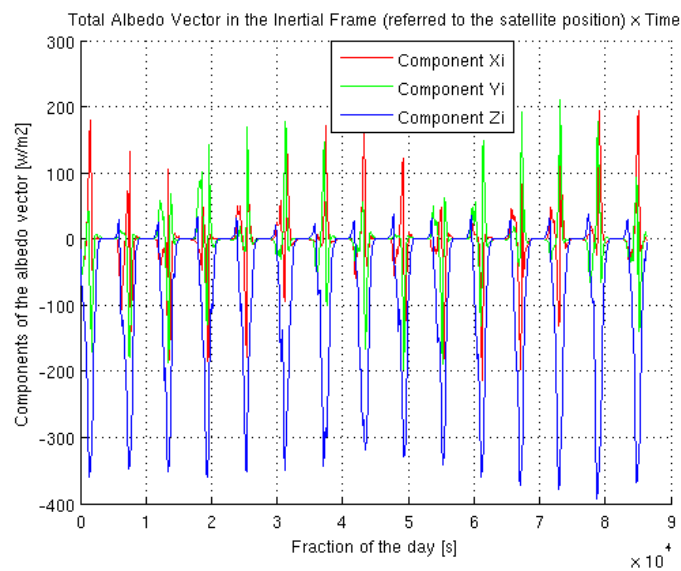


Fig. 133 – Total albedo vector in the inertial frame (Microscope).

7.2.1- Evaluation of the Trajectory Deviation for the Microscope Mission

The same assumption made in item 7.1.1 was used to calculate the force applied for the Microscope: spherical satellite shape with a mass of 1000 kg, 30% of the energy that reaches the satellite is absorbed, 30% is reflected specularly, and 40% is reflected diffusely. The effect in the trajectory is shown in Figures 135 to 140.

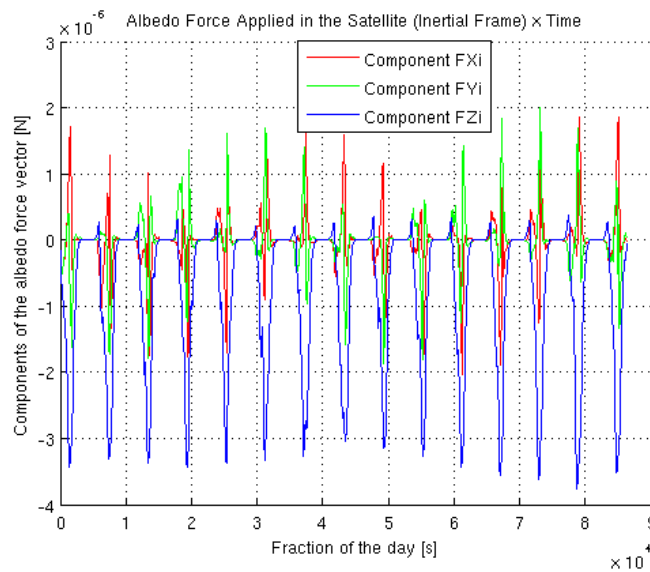


Fig. 134 – Albedo force vector applied in the satellite (Microscope).

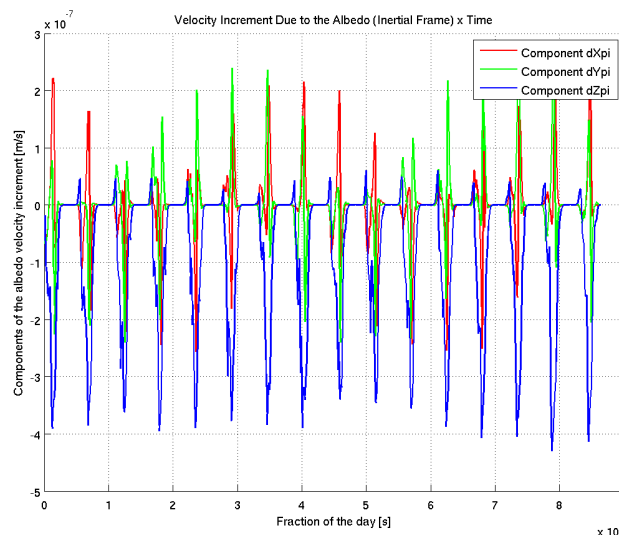


Fig. 135 – Velocity increment due to the albedo (Microscope).

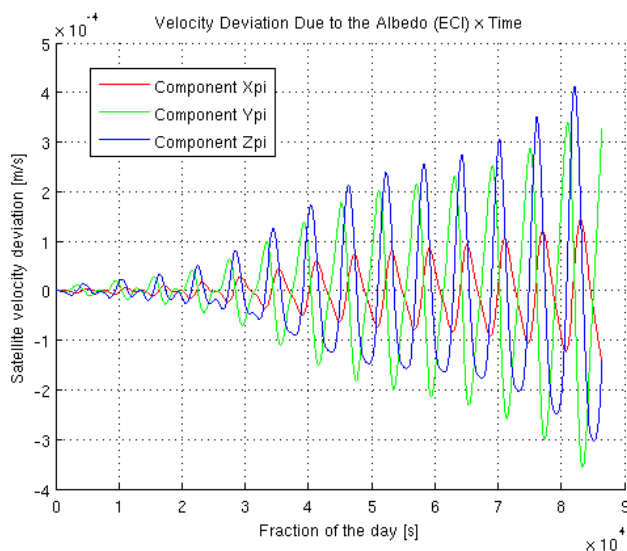


Fig. 136 – Velocity deviation for the entire day (Microscope).

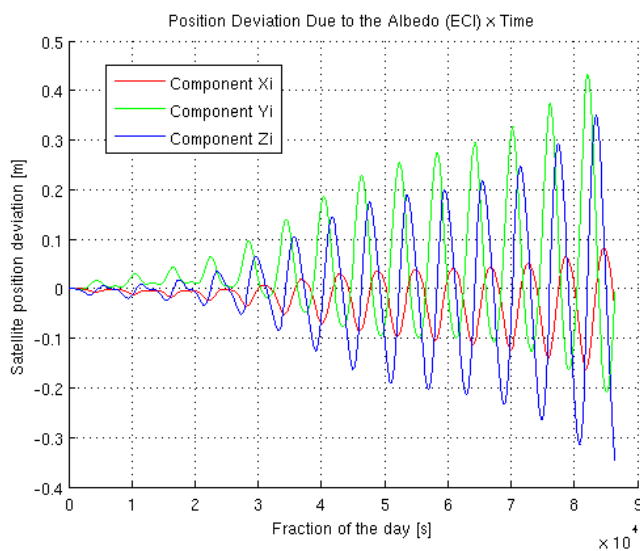


Fig. 137 – Position deviation for the entire day (Microscope).

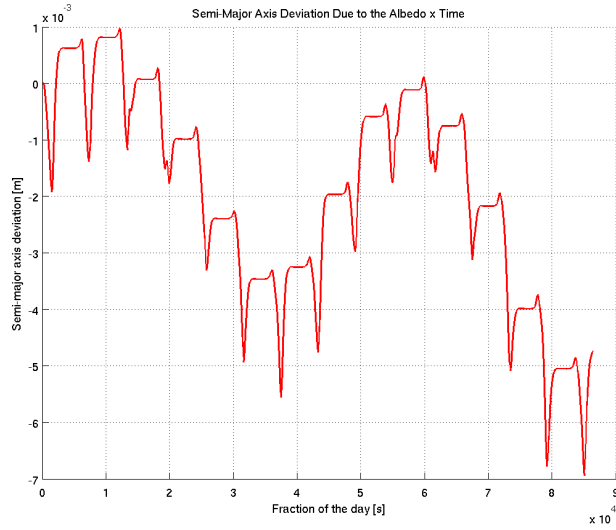


Fig. 138 – Semi-major axis deviation for the entire day (Microscope).

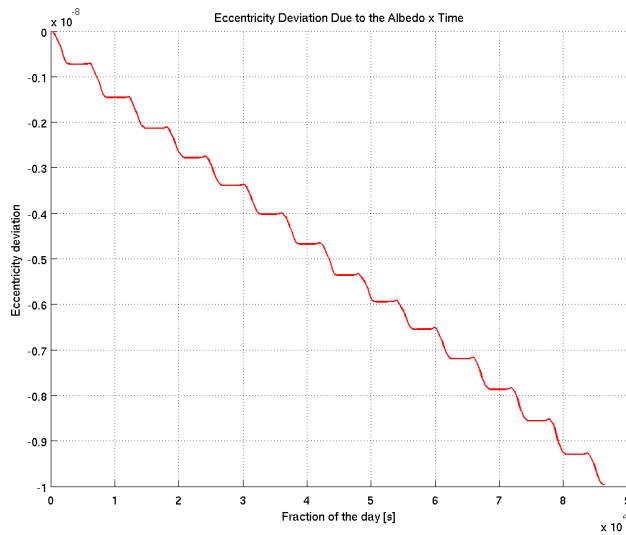


Fig. 139 – Eccentricity deviation for the entire day (Microscope).

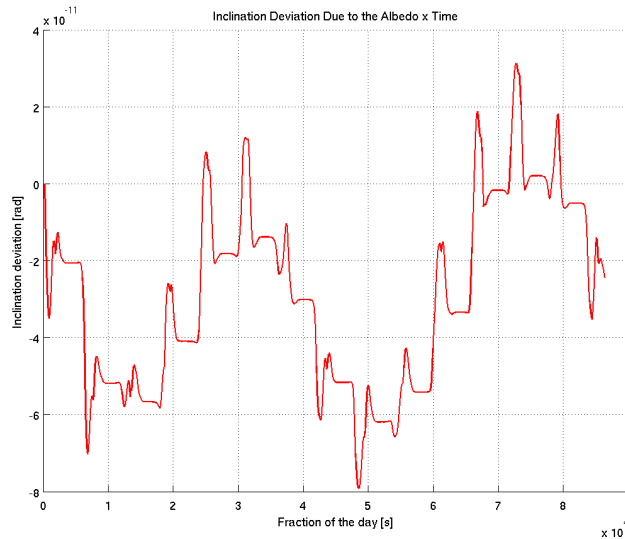


Fig. 140 – Inclination deviation for the entire day (Microscope).

7.3- Step Mission

The STEP mission was designed to test the Equivalence Principle with a high accuracy (10^{-17}) using pairs of concentric free-falling proof-masses of different material. The accuracy of 10^{-17} represents an unprecedented improvement of five orders of magnitude over the most precise experiments performed to date.

The graphs (Figures 141 to 157) were obtained considering the nominal orbit of the Step Mission and the Earth reflectivity in April 20, 2004. The graphs show the simulation for the entire day.



STEP

Nominal Orbit:

$a = 6778 \text{ km}$

$e = 0.0012$

$i = 97.03^\circ$

$\omega = 0$

$\Omega = 110^\circ$

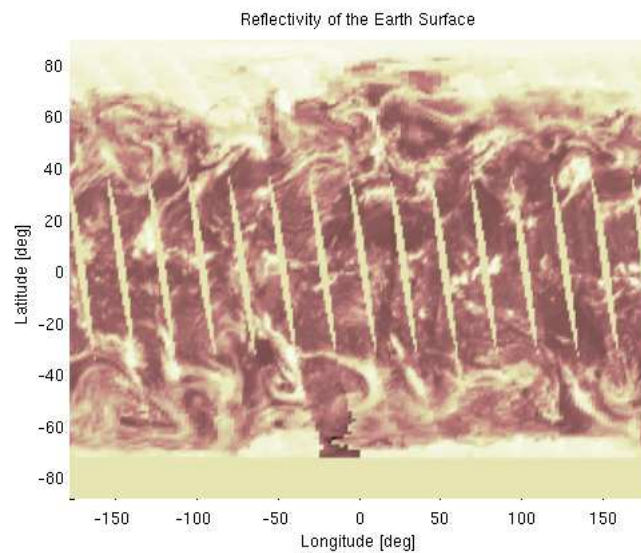


Fig. 141 – Reflectivity of the Earth in 20/04/2004.

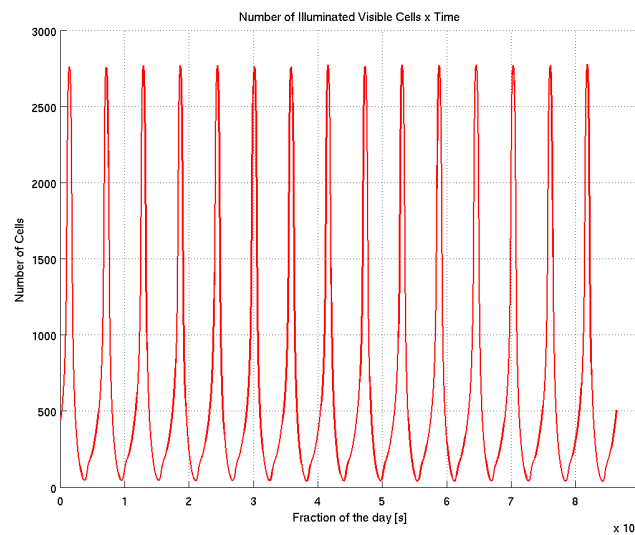


Fig. 142 – Number of illuminated visible cells (Step).

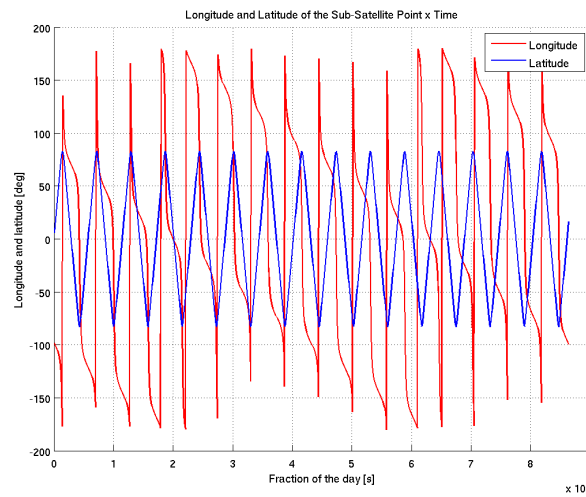


Fig. 143 – Longitude and latitude of the sub-satellite point (Step).

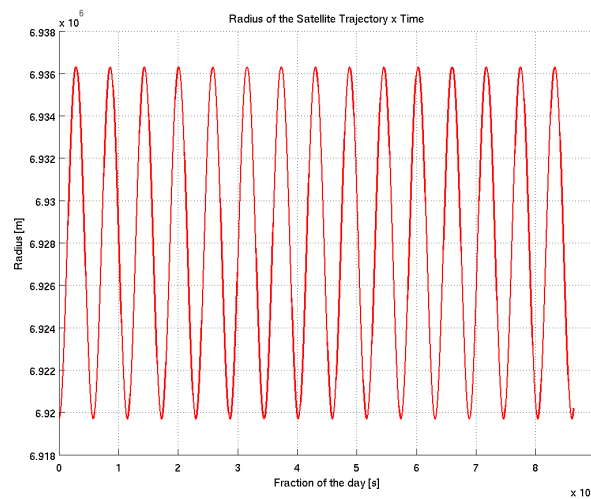


Fig. 144 – Radius of the satellite trajectory (Step).

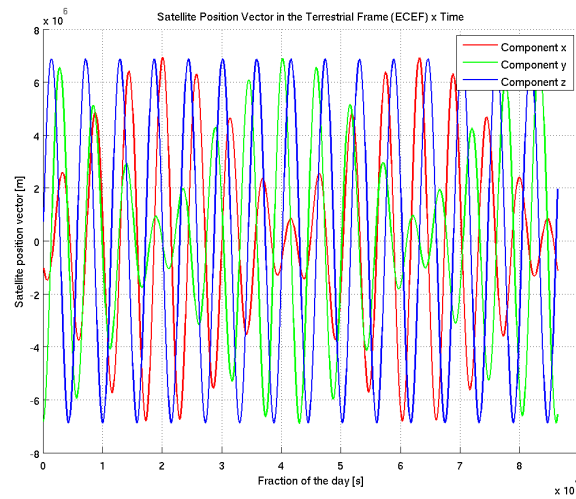


Fig. 145 – Satellite position vector in the terrestrial frame (Step).

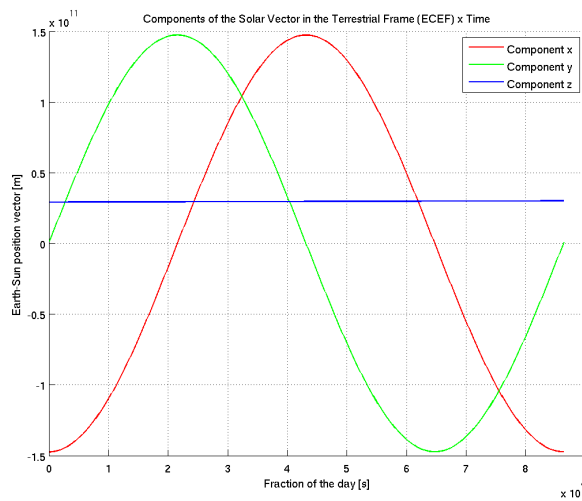


Fig. 146 – Sun position vector in the terrestrial frame.

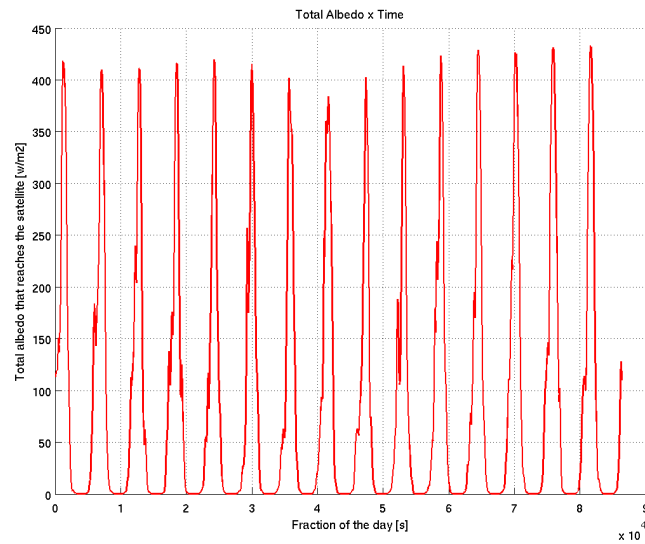


Fig. 147 – Total albedo (Step).

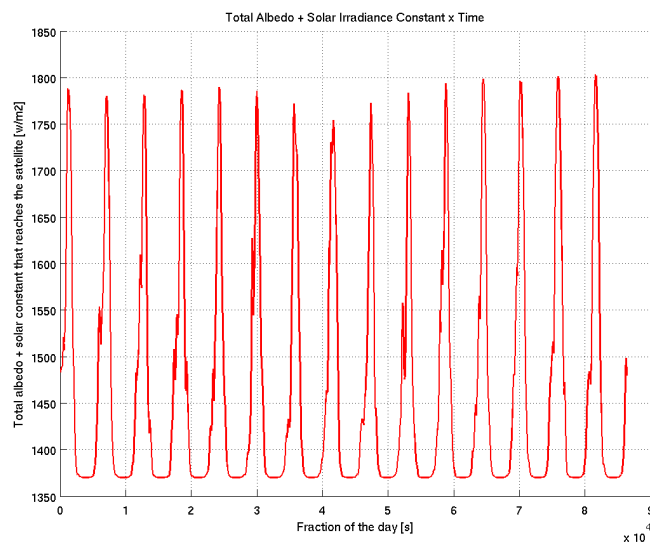


Fig. 148 – Total albedo + solar constant (Step).

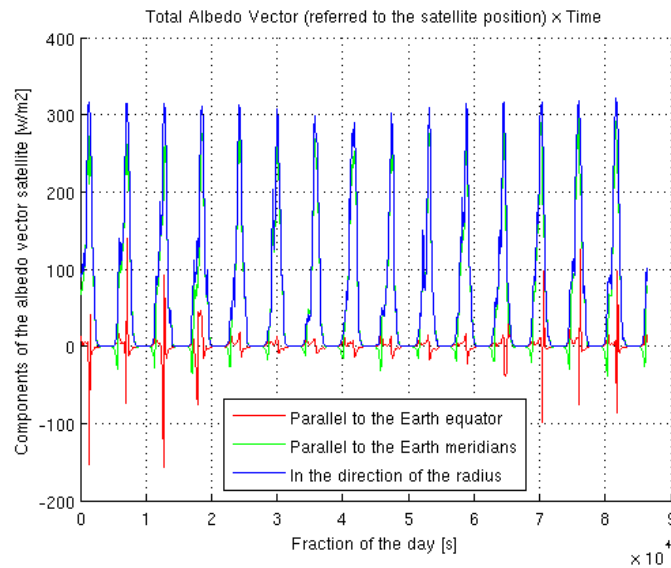


Fig. 149 – Total albedo vector (Step).

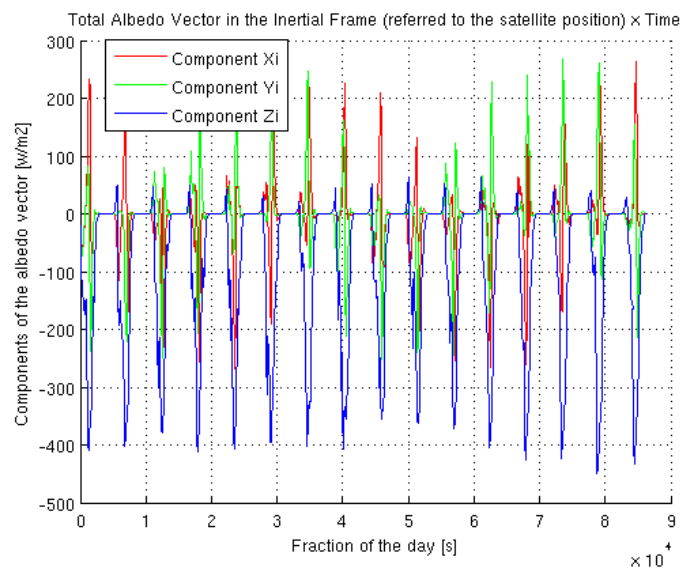


Fig. 150 – Total albedo vector in the inertial frame (Step).

7.3.1- Evaluation of the Trajectory Deviation for the Step Mission

The same assumption made in item 7.1.1 was used to calculate the force applied for the Step: spherical satellite shape with a mass of 1000 kg, 30% of the energy that reaches the satellite is absorbed, 30% is reflected specularly, and 40% is reflected diffusely. The effect in the trajectory is shown in Figures 152 to 157.

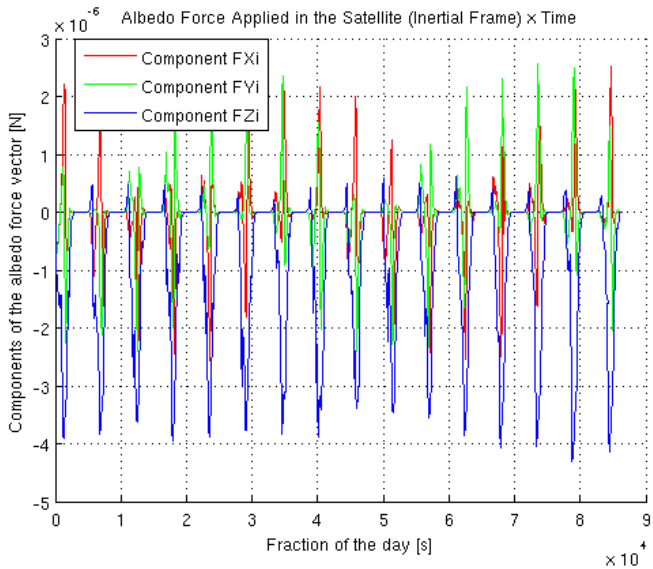


Fig. 151 – Albedo force vector applied in the satellite (Step).

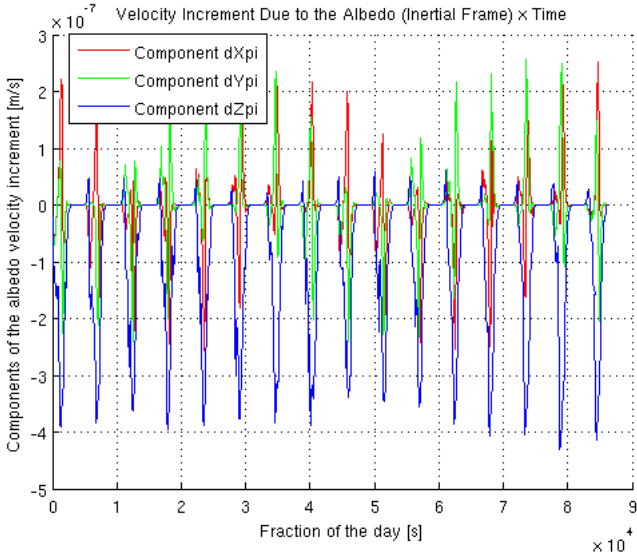


Fig. 152 – Velocity increment due to the albedo (Step).

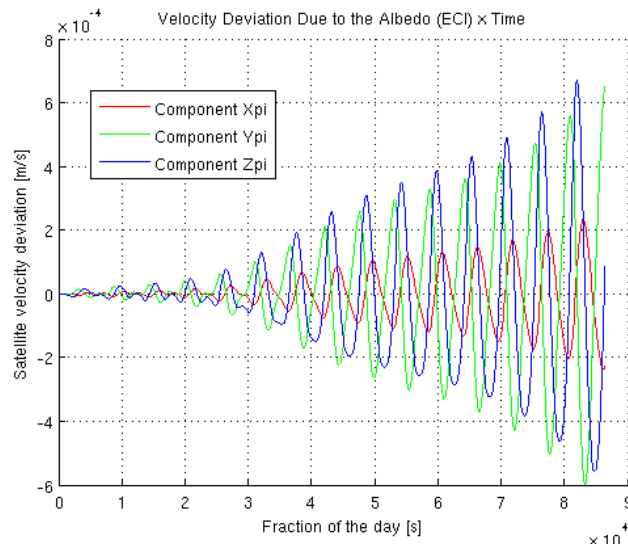


Fig. 153 – Velocity deviation for the entire day (Step).

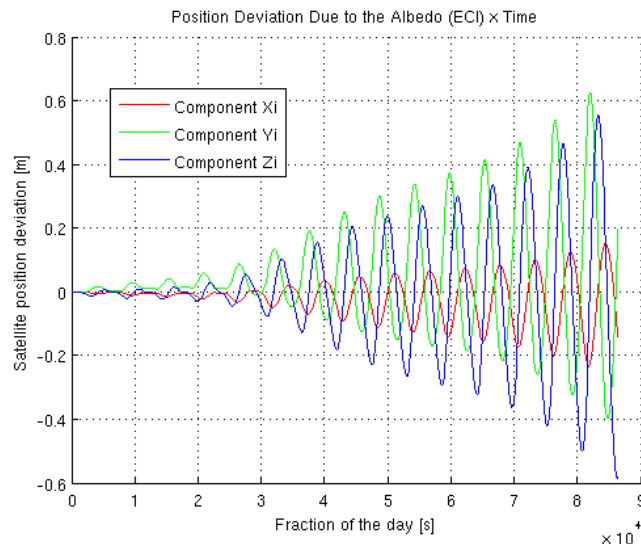


Fig. 154 – Position deviation for the entire day (Step).

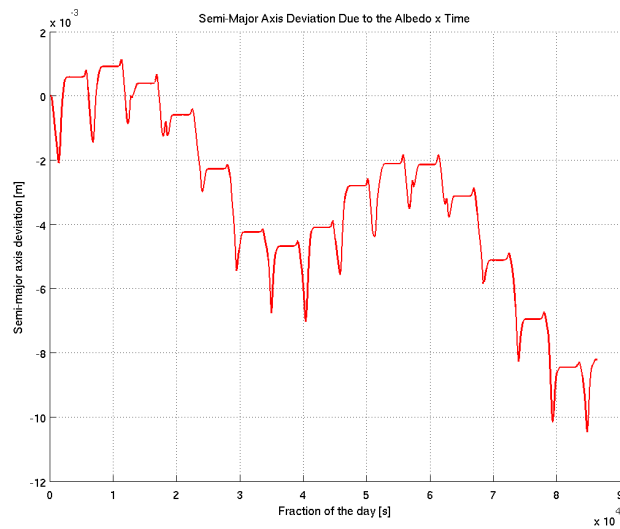


Fig. 155 – Semi-major axis deviation for the entire day (Step).

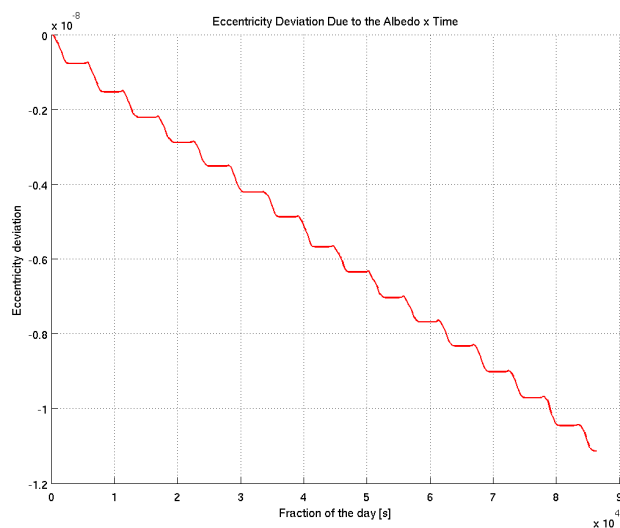


Fig. 156 – Eccentricity deviation for the entire day (Step).

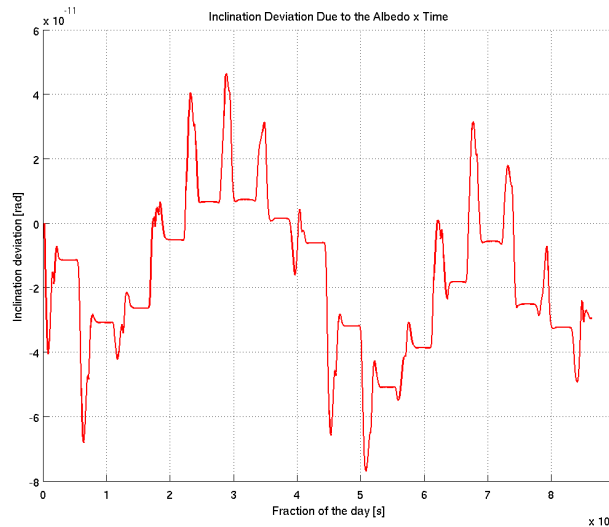



Fig. 157 – Inclination deviation for the entire day (Step).

8- Conclusion

The albedo model was implemented and tested with success. The results shown that the model is capable to determine the albedo vector that reaches the satellite. With this vector the forces and torques can be calculated. In this work a simple model for the optical properties of the satellite surfaced was used to test the albedo model and evaluate the force applied. But others more sophisticated optical models can be used, considering specific missions as Gravity Probe B, Microscope, Step and some others scientific missions. The deviation of the trajectory due to the albedo is small. Considering one day of simulation for the Gravity Probe B, the deviation in the position is almost 2 meters. However, in a drag-free mission this deviation must be corrected to reach the accuracy required in this kind of mission. To calculate the torque applied to the satellite the attitude of the satellite must be known. Beyond the attitude, the constructive characteristics of the satellite must be known. However, these constructive characteristics and the attitude dynamics of the satellite are not the goal of this work. The Earth Albedo Model can be used as a toll in the analysis of the disturbing applied to a satellite in orbit around the Earth. The model is not set to just one specific mission but can be used in any mission, simply setting the orbital parameters, the date, and the Earth reflectivity.

		Doc.No.: FLK-ENV-RP-ZAR-001 Issue: 2.0 Page: 91 of 91
---	--	---

References

Bhanderi, D., *Spacecraft Attitude Determination with Earth Albedo Corrected Sun Sensor Measurements*, Ph.D. Thesis, Aalborg University, Denmark, August 2005.

Wertz, J.R., editor, *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, 1978.

Appel, P.; Theil, S.; Winkler, S.; Schleicher, A., *Attitude Estimation from Magnetometer and Earth-Albedo-Corrected Coarse Sun Sensor Measurements*, 5th International ESA Conference on Spacecraft Guidance, Navigation and Control Systems, October 2002, pp. 613-616.

Harris, M; Lyle, R., *Spacecraft Radiation Torques*, NASA Space Vehicle Design Criteria (Guidance and Control) NASA SP-8027, October 1969.

Stark, J.; Fortescue, P., *Spacecraft Systems Engineering*, John Wiley & Sons Ltd., England, September 1994.



Modelling the attitude noise – A simplified approach

prepared by: Ralf Keil, Stephan Theil
approved by:
reference: GAIA-C2-TN-ZARM-RK-001-1
issue: 1
revision: 0
date: 28. November 2008
status: Issued

1 Introduction

This technical note deals with the modelling of the attitude noise of the Gaia spacecraft. Many disturbance torques affect the dynamics of the Gaia spacecraft. These torques include external (environmental) as well as internal (thrusters jets) disturbances. In order to perform and design an accurate model for the attitude noise these torques should be taken into consideration. The presented model considers only deviations from the Nominal Scanning Law (NSL, (3)). It contains the solar pressure noise as the major contribution of these external disturbance torque models. On the other hand, we consider the control torque as an actuator noise. The proposed simulation is designed as a closed-loop configuration with a Kalman-Filter (KF) (5) for the state estimation and a linear-quadratic optimisation algorithm (LQR) for the state feedback. KF and LQR shall represent the on-board attitude control system.

Please refer to the documents (8) and (9) for a mathematical model of the spacecraft's attitude.

2 Objectives and Requirements

The main objective of this study is to simulate the noise of the attitude and rates of the Gaia spacecraft to be as realistic as possible. The NSL describes the motion of the spacecraft at the L2 position. It defines the reference pointing attitude for the measurements of the two astrometric telescopes fields of view on the sky. This measurement principle relies on the systematic and repeating observations of star positions in the two fields of view. In this study, however, we will assume that the spacecraft dynamics will not exactly follow the NSL and will investigate all the disturbances which deviates the spacecraft motion from the NSL.

The measurement and pointing requirements for the on-board *Operational-Mode* attitude determination and control of the Gaia satellite is described in some detail in (1), and partly in (2). The most recent determination and control performance requirements are summarized in the "ESA Gaia Mission Requirements Document" (1), section 4.7. Table 1 (on the next page) lists the principal mission- and system-level requirements imposed on the attitude pointing and rate performances as provided. These requirements are applicable to both ASTRO telescope Lines-Of-Sight (LOS). The Gaia astrometric measurement principle and CCD operations make that a distinction must be made in the accuracy requirements for quantities measured in the along-scan and those along the across-scan directions.

The definitions in table 1 (according to Ref. (2)) are:

- The **Attitude Measurement Error (AME)** refers to the instantaneous 3-axis angular separation between the estimated satellite attitude (as well as the ASTRO-LOS) and the actual one,
- the **Rate Measurement Error (RME)** refers to the mean difference between the

Parameter	Requirement (99.73 % probability level)
AME - Attitude Measurement Error	< 20 arcsec
RME1 - Rate Measurement Error (AL)	< 0.9 mas/s
RME2 - Rate Measurement Error (AC)	< 2.7 mas/s
APE - Absolute Pointing Error	< 60 arcsec
RPE1 - Relative Pointing Error (AL)	< 5 mas
RPE2 - Relative Pointing Error (AC)	< 10 mas
MRE1 - Mean Rate Error (AL)	< 2 mas/s
MRE2 - Mean Rate Error (AC)	< 10 mas/s

Table 1: Summary of AOCS performance requirements

estimated spacecraft scan rate and the actual one,

- the **Absolute Pointing Error (APE)** refers to the instantaneous 3-axis angular separation between the desired satellite attitude (as well as the ASTRO-LOS) and the actual one,
- the **Relative Pointing Error (RPE)** at any given time t is defined as the standard deviation of the absolute pointing error over the Astrometric Field (AF) CCD integration time τ around t , and
- the **Mean Rate Error (MRE)** is defined as the mean difference between the desired satellite scan rate attitude and the actual one; the averaging time is the time spent by an object from being detected to being confirmed in the AF.

The AME, RME, RPE and MRE in along-scan direction are of particular interest for the simulation of the attitude noise.

3 Simulator Model Structure

The simulator used is based on a modular design (see Fig. 5). Its core is the module “*Dynamics*” (Fig. 6) that contains the satellite’s equations of motion. We consider the spacecraft dynamics as a rigid body dynamics with simplifications (4). The equations of motion are integrated by means of a widely-used solver for ordinary differential equations (ODE). The output of this module is the state vector containing the (real) state of the spacecraft (currently: its rate and angle for one axis). It propagates to the model block “*Sensors*” (Fig. 7) to simulate the behavior of the two sensor devices, the AF sensor, and the AST sensor, respectively. The result is the noise-induced output of the sensors. To control the satellite’s attitude motion its state has to be determined by

an observer. This is done using the Kalman-Filter “*KF*” block (Fig. 8). The estimated state of the system serves as the input for the controller that on its part put commands to the “*Actuators*” (Fig. 9) module with the FEEP model in order to converge the model behavior to that of the plant. The external “*Disturbances*” (Fig. 10) module contains currently a noise model of the solar pressure, being the chief disturbance source for Gaia spacecraft. All mandatory parameters for the simulation to be executed are defined in certain initialization files and these are called once the simulator environment is started. This allows smooth adjustments and clear extensions (see section 5).

4 The state observer and state feedback

The aim of the observer is to estimate the states of a real system, while it depicts a copy of the system model and makes use of the given measurement of the input and output of this system. The observation model predicts a state and the observer performs a procedure to minimize the deviation of this estimated state from a measured state. This estimation error is fed back to the model via an estimator gain. Following the Luenberger approach, the “*KF*” module resembles the system model in order to adjust the estimator gain. The KF algorithm computes this gain using minimization criterion (including solving the appropriate Riccati equation) with predefined parameter values for the process and measure noise covariances. The regulator LQR provides the information to design a state feedback controller, viz the computation of the feedback matrix gain in the “*Controller*” block. The required values for the weighting matrices are listed in (6).

5 The blocks in brief

5.1 Actuators: Thrusters

One of the major noise generators are the thrusters. The simulation block contains the noise model of FEEP thrusters, although they are not going to be used for Gaia. Since the slope of their PSD diagram, however, can be treated as a simplified representation of the noise characteristics within a steady state model of the MPS (see (7)), they have been implemented in the simulator structure in a rather straightforward way. Fig. 1 shows the FEEP noise block being the input block for the “*Actuators*” block (left side of Fig. 9).

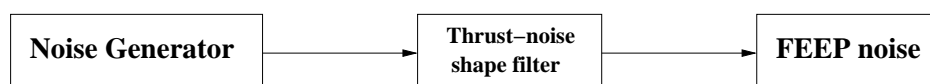


Figure 1: The FEEP noise simulator block

VARIABLE	DESCRIPTION	VALUE	UNIT
noise_power	white noise power	$1 \cdot 10^{-7}$	$\text{N}/\sqrt{(\text{Hz})}$
noise_seed	start seed of noise	random	–
noise_dt	sample time	0.1	s
noise_shape	TF	A,B,C,D	–

Table 2: Parameter of the FEEP thruster noise block

The input noise is generated by a band-limited white noise with a mean amplitude of 1. The thrust-noise shape filter is represented by a transfer function (TF) of a discrete state space model (with coefficients A,B,C,D and the sample time) together with a scaled white noise power (Table 2). The resulting level of thruster noise can be seen in the spectral distribution of the noise according to Fig. 2.

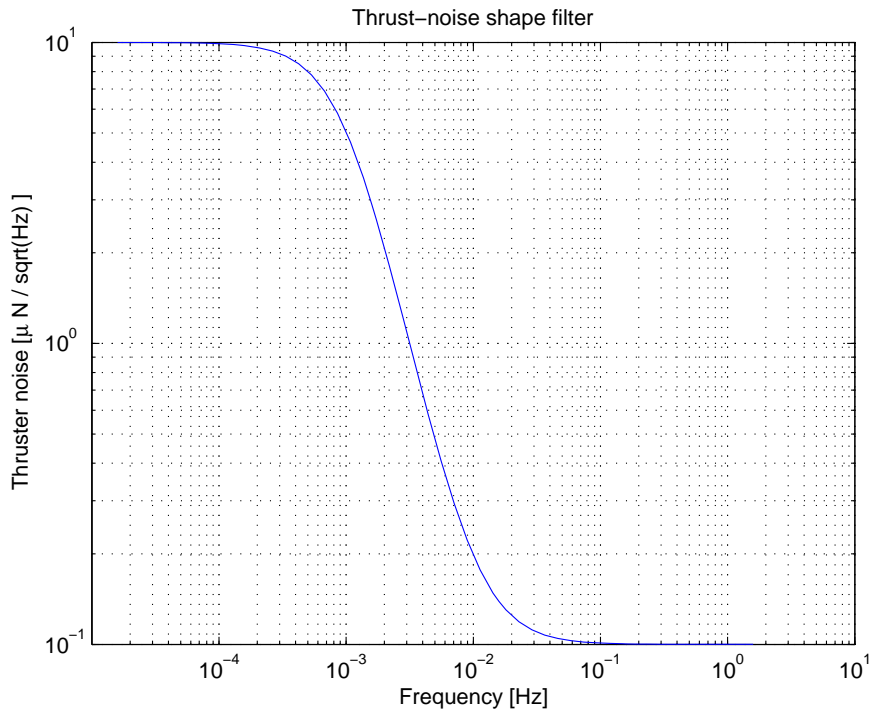


Figure 2: Thruster noise spectrum

5.2 Disturbances: Solar pressure

The simulation of the solar pressure disturbance makes use of the same noise generation as for the thrusters following the plant in Fig. 3. The form filter $H_{\text{sol}}(f)$ is scaled with a gain to reach the value of $1.59 \cdot 10^{-5} \text{ N}/\sqrt{\text{Hz}} @ f = 10^{-4} \text{ Hz}$. This value is the product of the solar pressure and the area of the sun shield array (compare with Fig. 10). It includes dependencies on the solar incidence angle, reflection, etc.

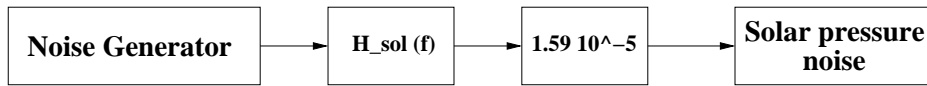


Figure 3: The solar pressure disturbance plant

$H_{\text{sol}}(f)$ on its part consists of two components: a smooth broad-band-slope and a 5-min-oscillation-peak. The corresponding transfer function $H_{\text{sol}}(s)$ can be written as

$$H_{\text{sol}}(s) = \frac{0.1233 \cdot s^{10} + 0.4471 \cdot s^9 + 0.526 \cdot s^8 + 0.2215 \cdot s^7 + 0.0223 \cdot s^6 + 0.0004 \cdot s^5 + 8.698 \cdot 10^{-6} \cdot s^4}{s^{10} + 4.698 \cdot s^9 + 7.007 \cdot s^8 + 3.252 \cdot s^7 + 0.0725 \cdot s^6 + 0.0015 \cdot s^5 + 2.404 \cdot 10^{-5} \cdot s^4}$$

The lever arm for the solar pressure force is scaled with the cosine of the solar aspect angle of 45 deg with $1.288 \cdot \cos(45/180 * \pi) \text{ m} = 0.911 \text{ m}$.

Fig. 4 (left) shows the power spectrum density of the solar pressure perturbation measured with the VIRGO instrument on-board the SOHO satellite. The normalization resembles the constant solar pressure value. On the right side one can see the calculated optimised form filter $H_{\text{sol}}(f)$.

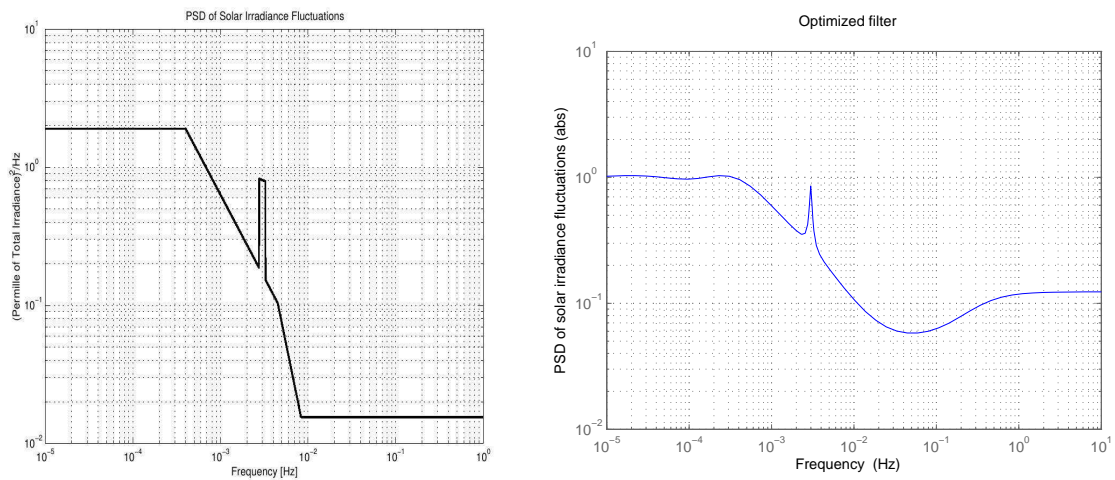


Figure 4: PSD slope from SOHO/VIRGO – comparison data vs. optimised form filter

5.3 Dynamics

The approach to this simulation model originates from the intention to move the spacecraft according to a prescribed course defined by the NSL. This means, the attitude and the rates of the spacecraft can be represented by the quaternions \tilde{q}_{NSL} and $\vec{\omega}_{\text{NSL}}$. The aim of the simulation run, however, is to provide real quaternions \tilde{q}_{real} and $\vec{\omega}_{\text{real}}$, in which both quantities differ only slightly from the NSL-values:

$$\begin{aligned}\tilde{q}_{\text{real}} &= \Delta\tilde{q} \otimes \tilde{q}_{\text{NSL}} \\ \vec{\omega}_{\text{real}} &= \vec{\omega}_{\text{NSL}} + \Delta\vec{\omega}\end{aligned}$$

with $\Delta\tilde{q} \approx (\Delta\varphi_x/2, \Delta\varphi_y/2, \Delta\varphi_z/2, 1)^T$ and $\Delta\vec{\omega} \approx (\Delta\omega_x, \Delta\omega_y, \Delta\omega_z)^T$.

According to the equations of motion the attitude motion can be described by the Euler equation

$$\dot{\vec{\omega}}_{i,b}^b = (\mathbf{I}_b^b)^{-1} [\mathbf{T}_{\text{control,dist}}^b - \vec{\omega}_{i,b}^b \times (\mathbf{I}_b^b \vec{\omega}_{i,b}^b)],$$

with the indices b=body frame (S/C frame), and i=inertial frame; $\mathbf{T}_{\text{control,dist}}^b$ are the control, and disturbance torques, respectively. \mathbf{I}_b^b is the inertia matrix containing only diagonal elements.

In a first approximation, $\omega_x = \omega_y \approx 0$, i.e. the cross product vanishes; only $\omega_z = 2\pi/6 \text{ h} = 1 \text{ arcmin/s}$ remains. Thus, the three equations are decoupled and individual simulations of the three DOF can be performed. This yields to completely independent simulation loops when investigating attitude and rates deviations from the NSL (see Fig. 6).

5.4 Sensors: AF and AST

The sensors currently used in this simulation model are a sensor related to the Astrometric Field (AF; see Fig. 7) and a sensor linked to the Autonomous Star Tracker (AST). The AST-sensor consists of an attitude noise simulated by a band-limited white noise with the power of $0.1 \cdot (\sigma_{\text{AST}}/\text{deg})^2 \text{ W} = 0.1 \cdot (15 \text{ arcsec}/\text{deg})^2 \text{ W} = 5.29 \cdot 10^{-10} \text{ W}$ within a sample time of 0.1 s and a random seed. The AF-sensor combines a band-limited white noise of power $0.1 \cdot (\sigma_{\text{AF}}/\text{deg} * \text{s})^2 = 0.1 \cdot (1 \text{ mas}/\text{s}/\text{deg} * \text{s})^2 \text{ W} = 2.35 \cdot 10^{-18} \text{ W}$ and a TF of

$$\text{TF}_{\text{AF}} = \frac{1}{1 + \frac{T_{\text{av,AF}}}{2\pi}},$$

where $T_{\text{av,AF}}$ is the averaging CCD integration time of 5 s of the AF.

5.5 Kalman-Filter (KF)

The number of different states of interest is given by two quantities (ω and \tilde{q}) plus $T_{av,AF}$, resulting in a three-dimensional implementation of the KF block (Fig. 8, left). The KF approach allows to “observe” and estimate the current state of the system and to act as the input source for the controller.

The state space model for the attitude estimation is given by the equations

$$\begin{aligned}\dot{\vec{x}} &= \mathbf{A} \cdot \vec{x} + \mathbf{B} \cdot \vec{u} \\ \vec{y} &= \mathbf{C} \cdot \vec{x} + \mathbf{D} \cdot \vec{u}\end{aligned}$$

where $\vec{x} = (\omega, \tilde{q}, T_{av,AF})^T$ is the state vector, \vec{u} the input vector and \vec{y} the output vector (all of dimension 3). The system matrix A, the input matrix B, the output matrix C, and the feedforward matrix D are defined according to (for the z-axis)

$$\begin{aligned}\mathbf{A} &= \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & -\frac{T_{av,AF}}{2\pi} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1/\text{Sat.I}_{zz} \\ 0 \\ 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\ \mathbf{G} &= \mathbf{I}_{[3 \times 3]}, \text{ and } \mathbf{H} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.\end{aligned}$$

where Sat.I_{zz} is the z-component of the momentum of inertia. G and H are auxiliary matrices.

The system design of the KF makes use of the state space model as given above. The KF takes the fact into consideration that the system as well as the output signal \vec{y} hold a noise contribution, of which the covariances of the steady state are given by the matrices Q (for the system noise), R (for the output signal noise), and N as the covariance matrix for the product of both noise signals:

$$\mathbf{Q} = \begin{pmatrix} \sigma_{\dot{\omega}}^2 & 0 & 0 \\ 0 & \sigma_{\dot{\varphi}}^2 & 0 \\ 0 & 0 & \sigma_{\dot{\omega}_f}^2 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \sigma_{\omega}^2 & 0 \\ 0 & \sigma_{\varphi}^2 \end{pmatrix},$$

with $\sigma_{\omega} = \sigma_{AF}$, $\sigma_{\varphi} = \sigma_{AST}$, $\sigma_{\dot{\omega}} = 8.1 \cdot 10^{-9}$, $\sigma_{\dot{\varphi}} = 10^{-20}$, $\sigma_{\dot{\omega}_f} = 10^{-20}$ and

$$\mathbf{N} = 0_{[3 \times 2]}, \vec{x}_{\text{est},0} = (0, 0, 0)^T \text{ as the initial KF state.}$$

Given all these information, the KF is computed, resulting in various parameters of the filter, like the Kalman gain, and the steady state covariance.

The next step is to use an optimisation algorithm in order to find the best estimation for the state vector for the given system.

5.6 Controller

The goal of the optimisation algorithm is to estimate the state vector \vec{x}_{est} in such a way that the quadratic mean value of the error $\vec{x} - \vec{x}_{\text{est}}$ finds its minimum. Thus, for KFs like the one presented here it is customary to use the method of the linear-quadratic optimisation algorithm (LQR). The KF output parameters are used as input parameter for the LQR-algorithm.

The state space model for the controller is 2-dimensional, with ω and \tilde{q} as the free variables and given by the matrices

$$AC = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, BC = \begin{pmatrix} 1/\text{Sat.I}_{zz} \\ 0 \end{pmatrix},$$

and for the weighting matrices

$$QC = \begin{pmatrix} w_{\omega}^2 & 0 \\ 0 & w_{\tilde{q}}^2 \end{pmatrix}, RC = w_u^2,$$

with the weighting factors $w_{\omega} = 9.6963 \cdot 10^{-9}$ (MRE-AL), $w_{\tilde{q}} = 2.9089 \cdot 10^{-4}$ (APE) and

$w_u = 10^{-6}$ as the weighting factor for the input of the LQR algorithm.

The LQR algorithm computes a feedback of the state in form of a matrix gain K (Fig. 8, right) that modifies the state vector \vec{x}_{est} to its optimal value.

6 The simulation

The whole simulation is created and performed within the ‘‘Matlab/Simulink’’ environment, a technical computing package for solving numerical problems in combination with modelling of time-relevant technical systems (version 7.1.0.183 (R14) SP 3, August 02, 2005).

The simulation runs over a complete spin period of the Gaia spacecraft of 6 hours. The solver of the ordinary differential equations was chosen to be the ode45-algorithm (Dormand-Prince; based on a Runge-Kutta-formalism) using a variable step size and a relative tolerance of 0.1%, being the upper limit of the computed state to be treated as accurate.

The results of the simulation are presented in Fig. 11. The upper two plots show the error of the rate ω , and the attitude parameter ϕ , based on the estimation calculations, whereas in the plots at the bottom the control errors of both quantities are displayed. One can see that both errors are within the limits given by the mission requirements.

7 Preview

This note comprises the treatment of different noise contributors to the total attitude noise expected for the Gaia spacecraft. The presented model shall demonstrate the basic principle to handle noise signals within a simulation loop in order to understand the dynamics of the satellite in terms of physical effects. Much further work could be done for this model, but at this point, we decided to mark this note as the preliminary endpoint of the work on this model.

As the Gaia project community is proceeding efficiently in developing the Java-based software package GaiaTools for most of the tasks related to this mission, we decided to start transferring our model to this programming environment. The “Gaia Attitude Model” will contain all modules essential for a high-precision attitude simulation and will make much use of the modules developed for this noise model.

References

- [1] ESA, “Gaia Mission Requirement Document”, Gaia-EST-RD-00553, Nov. 2006.
- [2] Astrium, “Gaia System Level Technical Reassessment Study, Final Report”, EF5/FR/PC/038.02, 27 June 2002, Chapter 3.5.
- [3] L. Lindegren, “Attitude Parameterization for Gaia”, SAG-LL-30, 1 July 2000.
- [4] Wertz, J. R., “Spacecraft Attitude Determination and Control”, Kluwer Academic Publishers, the Netherlands, 1990.
- [5] Gelb, A., “Applied Optimal Estimation”, MIT Press, Cambridge, MA, 1974.
- [6] Astrium, “Spacecraft Requirements Specification”, GAIA.ASF.SP.SAT.00001, Issue 2, 2 Aug. 2006
- [7] Astrium, “Hypotheses for AOCS analyses”, GAIA.ASF.TCN.SAT.00009, Issue 3, 15 April 2008
- [8] Lindegren, L., & Luri, X., “A model of noisy attitude for cycle 3 simulations”, GAIA-C2-TN-LU-LL-073-1, Issue 1, 19 July 2007
- [9] Hobbs, D. & Lindegren, L., “Attitude Processing for Gaia”, GAIA-C3-TN-LU-DH-002-01, Issue 1, 19 June 2008

8 Figures

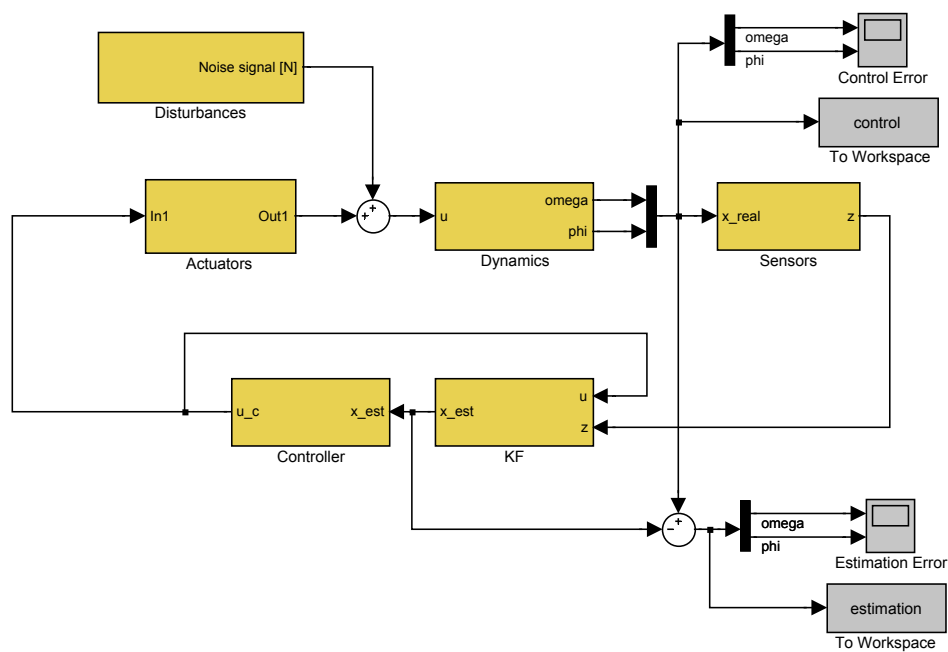


Figure 5: Generic simulator structure for a closed-loop configuration

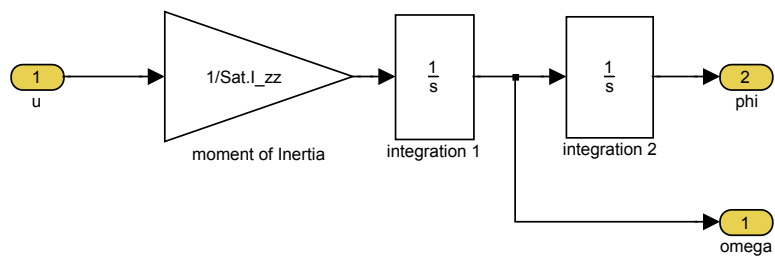


Figure 6: Simulator module “Dynamics”

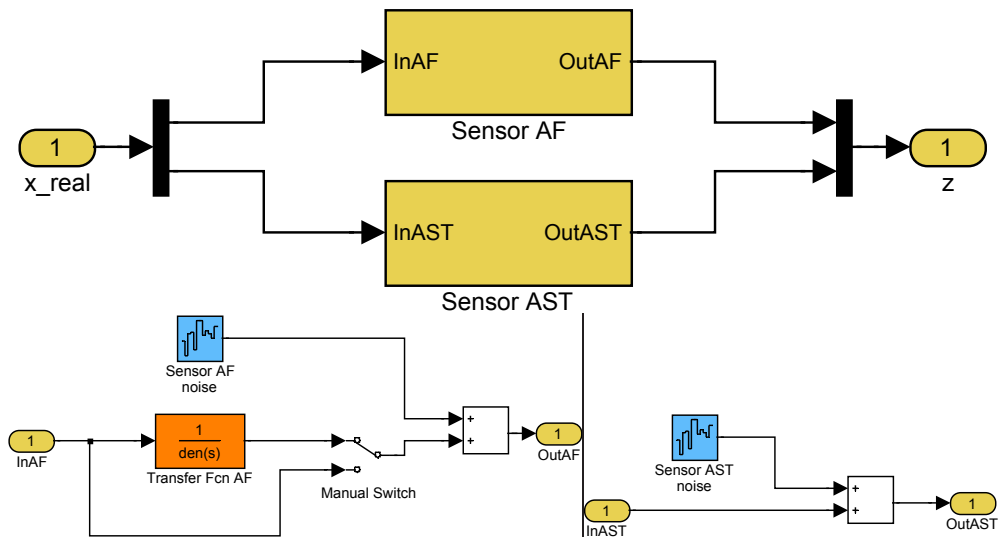


Figure 7: Simulator module “Sensors” and its components “Sensor AF” (left) and “Sensor AST” (right)

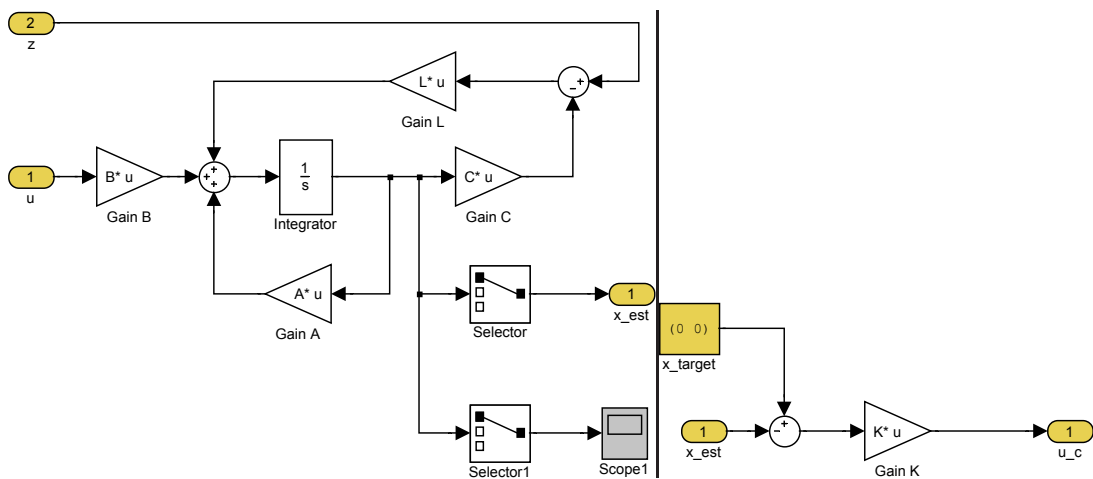


Figure 8: Simulator modules “KF” (left) and “Controller” (right)

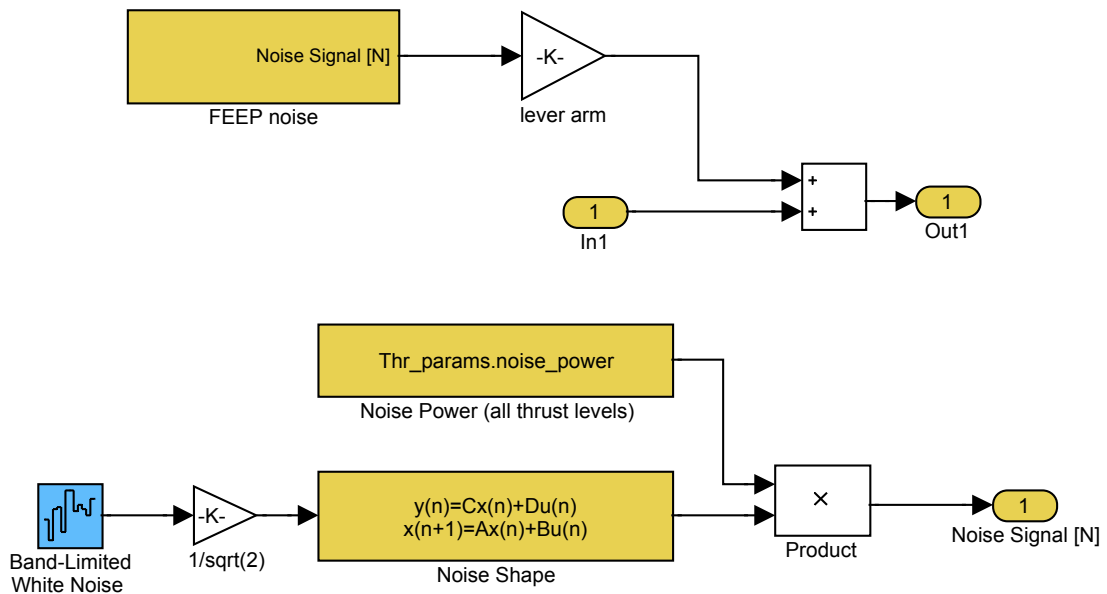


Figure 9: Simulator module “Actuators”. The block “FEEP noise” in the upper plot contains all blocks from the lower plot.

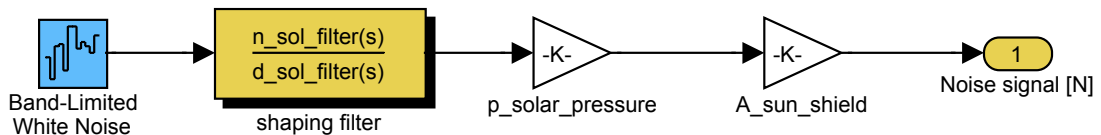


Figure 10: Simulator module “Disturbances”. Here: the solar pressure

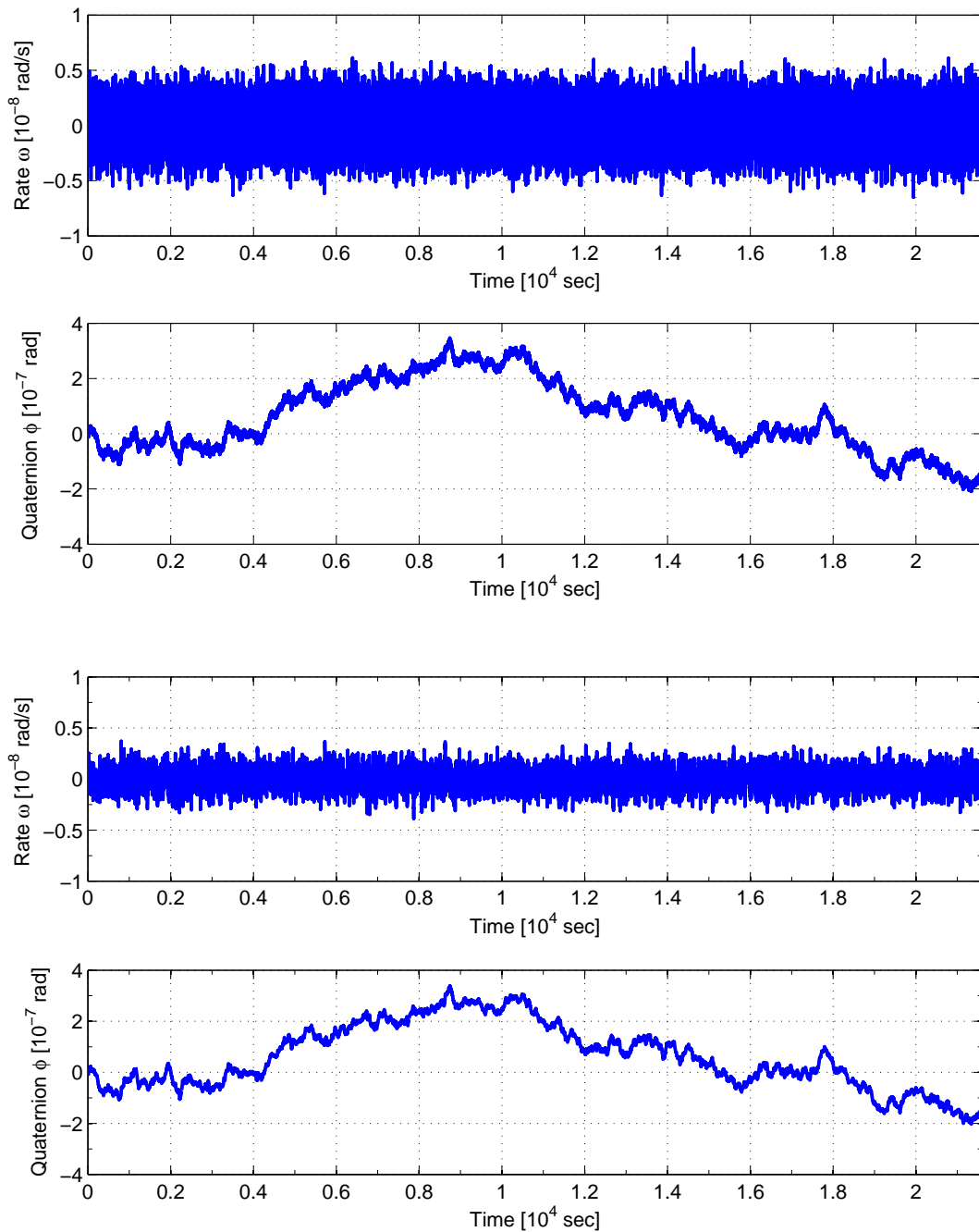


Figure 11: Estimation error (the two upper plots) and Control error (bottom plots) of the rate ω and the attitude parameter, the quaternion ϕ



Simulating Gaia's Attitude: Description and realisation of the Gaia Attitude Model (GAM)

prepared by: R. Keil
affiliation : ZARM, University of Bremen
approved by:
reference: GAIA-C2-TN-ZARM-RK-002-D
issue: 1
revision: 0
date: 2009-12-16
status:

Abstract

This technical note describes the Gaia attitude model (GAM) and its fundamental idea to develop a sophisticated model of the dynamics of the Gaia satellite entirely based on physical considerations. The modelling comprises of all essential parts for a spacecraft simulator and are combined to the simulation package GAM as part of the tasks covered by the CU2 "Data simulations" within the Gaia project community. The goal of this work is to have a simulation software available to achieve a detailed understanding of the satellite's real behaviour during its operational phase.

Document History

Issue	Revision	Date	Author	Comment
0	0	2009-11-02	RK	Creation

Acronym List

Acronym	Description
AC	ACross scan (direction)
AF	Astrometric Field (in Astro)
AL	ALong scan (direction)
AOCS	Attitude and Orbit Control Sub-system
CCD	Charge-Coupled Device
CL	Closed Loop
CoM	Centre of Mass
DPAC	Data Processing and Analysis Consortium
DSA	Deployable Sunshield Assembly
EADS	European Aeronautic Defence and Space company (EADS Astrium is the Gaia prime contractor)
FoV	Field of View (also denoted FOV)
GAM	Gaia Attitude Model
HW	Hardware (also denoted H/W)
ICRS	International Celestial Reference System
IoA	Institute of Astronomy (Cambridge; also denoted IOA)
KF	Kalman Filter
LEI	Leiden Observatory (Sterrewacht Leiden)
LOS	Line of Sight
MPS	Micro-Propulsion Sub-system
MRD	Mission Requirements Document
NDA	Non-Disclosure Agreement
NM	Normal Mode (AOCS)
NSL	Nominal Scanning Law
ODE	Ordinary Differential Equation
OOP	Object Oriented Programming
RK4	Runge-Kutta integrator algorithm of 4th order
SCRS	SpaceCraft Reference System
SM	Sky Mapper
SRP	Solar Radiation Pressure
SRS	Scanning Reference System
STR	Star TRacker

SW	Software (also denoted S/W)
ZARM	Centre for Applied Space Dynamics and Microgravity, Bremen (Germany)

Acknowledgements

This research project has been supported by a *Marie Curie Transfer of Knowledge* Fellowship of the European Community's Sixth Framework Programme under contract number MTKD-CT-2004-014188.

I want to thank Floor van Leeuwen (IoA), Daniel Riquez (LEI), and Anthony Brown (LEI) for their valuable contributions and always helpful comments on many issues and questions.

The GAM project is part of the DPAC software development for the Gaia mission and can be found in the subversion-repository at the ESAC-server site
<http://gaia.esac.esa.int/dpacsvn/DPAC/CU2/software/Experimental/GaiaAttitudeModel/>

Contents

1	Introduction	5
1.1	Pointing control definitions	5
1.2	Reference systems	6
2	History and simulator's cornerstones	8
3	Global structure and design	10
3.1	Model description	10
3.2	Dynamics/Kinematics	12
3.3	Disturbances	14
3.3.1	External disturbances	15
	The SRP.	15
	The hits generator.	15
3.3.2	Internal disturbance: The MPS	16
3.4	Sensors	17
3.5	AOCS	18
4	Example simulation	20
5	Summary	24
6	References	25

1 Introduction

Modelling the behaviour of a satellite in orbit becomes more and more an essential part of the data processing chain for a space-based mission. The larger the sensitivity of the carried payload (i. e. CCDs, optics, test masses) is to perform a measurement the better the knowledge of the satellite's dynamics has to be in order to reach the desired level of accuracy in the measurement. For Gaia, this level and, hence, the final scientific accuracy of the data, are only achievable if its motion can be understood very precisely.

This technical note explains the fundamental idea of the Gaia Attitude Model (GAM), a software package to model the behaviour of Gaia based on a physical approach in order to simulate the satellite's dynamics to a high degree. In combination with an emulated version of the on-board software for the control of the satellite in orbit, this software is capable to export high-fidelity simulation data of Gaia's attitude and rates. The main goal of this work is to develop a satellite model to be as realistic as possible that can be used and integrated in the software developments of other areas within the Gaia community.

1.1 Pointing control definitions

Before presenting the idea of the GAM, it is worth to recall that the simulation of a rigid body like Gaia pointing towards a specific direction at the sky, is based on three different terms of attitude. According to Fig. 1 they can be defined as follows:

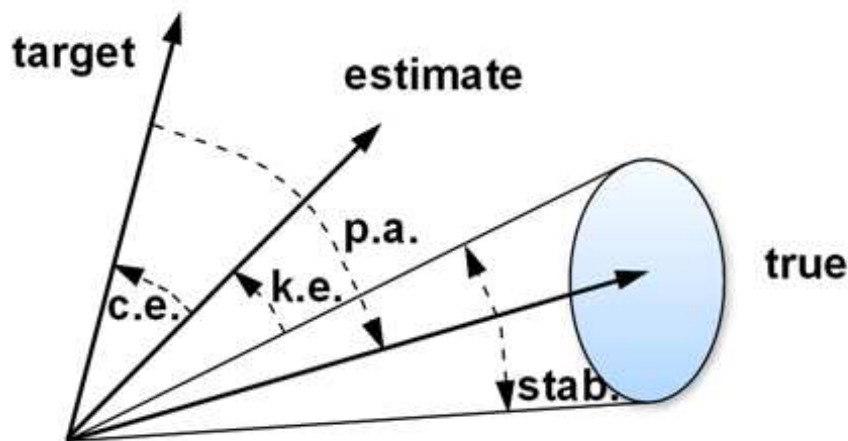


FIGURE 1: Pointing control definitions. Three different terms of attitude are shown, the target, the estimated, and the true attitude, respectively. Likewise, one can define a difference between two of them, as it is described in the text. For the rates, the same classification can be made. The GAM is based on the usage of these definitions throughout its entire development process.

- The **target** attitude describes the desired pointing direction given by any kind of (semi-)analytical description, here the NSL,
- the **true** or **actual** attitude is the (mean) attitude value of the true pointing direction of the LoS,
- the **estimate** attitude indicates the instantaneous estimate of the true attitude, based on a simplified representation of the satellite's dynamics.

They, moreover, allow to define certain error terms by two of the attitude terms. The difference between the target and the true attitude is called the pointing-inaccuracy, or "attitude error" ("p.a." in the sketch) and is usually determined over a long period of time. The deviation of the estimated attitude from the true one represents the (maybe poor) quality of the knowledge of the latter attitude and is thus called the "knowledge error" ("k.e."), because the more information one can get from measurements the better is the estimated attitude deduced from the AOCS, and finally, the closer is the estimated value at the true value. In addition, the "control error" ("c.e.") quantifies the deviation of the estimated attitude from the target value, providing essential information to the attitude control in order to correct the satellite's orientation (see section 3.5). And finally, a jittering motion of the satellite can be monitored by the peak-to-peak motion of the LOS ("stab."), keeping track of the stability of the pointing direction and, thus, of the true attitude.

1.2 Reference systems

The interaction between scientifically motivated and technologically used terms and quantities makes it inevitably to clearly state the difference between reference systems representing different aspects of the satellite. The group of astronomical reference systems, as described in section 4.2.4 and shown in Fig. 4.1 of Bastian (BAS-003), contains the "Scanning Reference System" (SRS), a system that is fixed to the rigid body of the spacecraft during its motion and rotation. Its origin is the centre of Mass (CoM) and the principal axes x , y , and z (see Fig. 2) are defined according to their introduction in Bastian's document. The SRS is fully aligned to the "Mechanical Spacecraft Reference System" (SCRS), i. e. the mechanical equivalent to the SRS, with its zero point C_S (the center of the circular satellite interface with the launch vehicle adapter, see EADS Astrium (GAIA.ASF.SP.SAT.00001); *not* the CoM) and the axes $-X_S$, Y_S , and Z_S . Please notice the **difference in the orientation of the principal axes in the SRS and the axes in the SCRS with respect to the satellite**. The nominal rotation axis of the satellite, as defined by the astronomical community, is usually written as the 'z' axis, whereas the engineering side is used to call this long side axis the ' $-X_S$ ' direction (due to considerations related to the launch vehicle configuration). In addition, the 'x' axis corresponds to the '+ Z_S ' axis. In agreement with the 'Gaia Mission Requirements Document' (MRD, ESA Gaia Project (GAIA-EST-RD-00553)), both definitions are valid and are used according to their operational field. Any kind of information that deals with the satellite's attitude or angular rate in this tech-

nical note, will be given as SRS quantities with respect to the primary reference system, the ICRS.

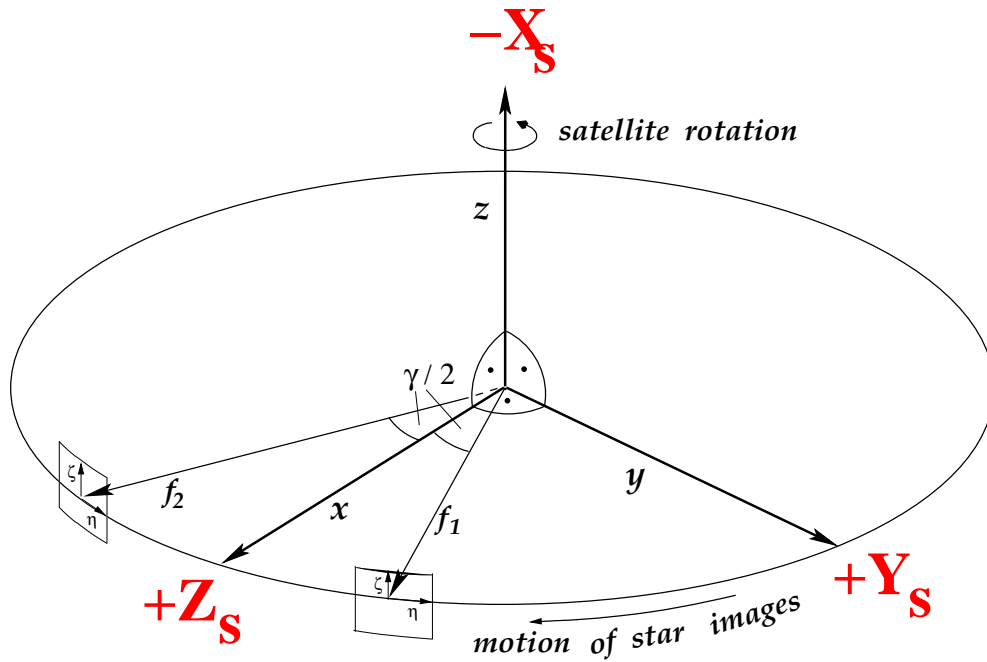


FIGURE 2: The "Scanning Reference System" (SRS) with its axes x , y , and z , respectively, superposed by the mechanical equivalent, the "Mechanical Spacecraft Reference System" (SCRS) with its axes $+Z_S$, $+Y_S$ and $-X_S$. Notice, that due to the launch vehicle configuration system engineers prefer the ' $-X_S$ '-axis of the spacecraft to point along its long axis, whereas the astronomers call this rotation axis usually the ' z ' axis. In addition, the ' x ' axis corresponds to the ' $+Z_S$ ' axis. In this document, the satellite's attitude and angular rate are given as SRS components with respect to the ICRS. (Image taken from Bastian (BAS-003).)

2 History and simulator's cornerstones

In a meeting in early 2007, members of different CUs discussed some aspects of the Gaia attitude (Theil & Keil (STH-001)) as this would be important in various areas of the data processing, for example in AGIS, in the IDT, and for simulation purposes. All participants agreed on the need to start developing a simulation software right from the beginning of the mission's planning. It turned out that, however, the proposed simulation software would have to be modified profoundly and so, it was later decided to start developing a simulation software from scratch, although sophisticated work on Gaia's attitude in a different simulation environment has been done in the meantime, dealing with the interaction of noise sources contributing to the overall dynamics of the satellite, and adaptable to any kind of simulation model (Keil & Theil (RK-001)).

As a part of the instrument modelling a first implementation of an attitude model for the satellite has been developed and used from cycle 3 simulations and later (Lindgren & Luri (LL-073)). This approach is based on a stochastic model, generating white noise in all three satellite axes independently (from a stationary Markov process) and superimpose it on the target attitude as given by the NSL to compute data representing the actual attitude. This idea may be a fast way to create a simple attitude model, but it eventually lacks of physical principles and effects that are of essential importance for every high-precise motion of a spacecraft.

Any kind of adaptive software for attitude simulation has to comprehend a list of general items, extracted from the selection process of the software (environment) or based on criteria for the simulator design. The software, therefore, should contain

- A high-precise propagation of the attitude within a diversity of different simulation runs, i. e. mission scenarios with specific objectives or with formulations of a certain problem,
- Highly flexible implementations of perturbation models, and their characteristics, changeable by freely chosen parameters,
- Simulating the spacecraft sensors, here based on SM-AF1 and Star Tracker measurement data,
- Integrating modules simulating hardware and on-board software components (for testing and research purposes),
- Extensions of various nature to be implemented smoothly into the current software stage,
- Usage of simulation techniques in a transparent manner.

For orbital simulation, this list needs to be modified and extended. But this is not the subject of this technical note.

The attitude model for Gaia's spacecraft is currently being developed to build up a high-fidelity software package for the simulation of the satellite's dynamics, as its understanding is a key component to reach the highest available level of accuracy of the measurements and, finally, of the scientific output. Because both Gaia's attitude and angular rate are closely linked to the scientific measurement principle, this software allows a detailed modelling of these quantities from an entirely physical point of view, hence, to be as realistic as possible. It takes into account disturbances of various physical origin acting on the satellite (see section 3.3). The overall goal of the GAM is to provide precise simulated data of the satellite's orientation on its orbit as a function of time.

3 Global structure and design

The framework of the GAM is a collection of classes and libraries as they are currently set up by the Gaia community: the GaiaTools (see Angeli (2005) & GaiaTools Committee (MTL-015)) and its accompanying libraries, the Gaia Parameter Database (Lammers & Joliet (UL-001)) and a plotting library, currently developed at the Leiden Observatory.

The GAM as well as the other classes and libraries are written in Java, a widely used, taught, and supported object-oriented programming (OOP) language. DPAC preferred Java over C++ mainly because of a faster progress in the code development, higher code reliability, and 100% portability within the heterogeneous coding networks. Its functional syntax is similar to that of the numeric calculations and simulations package MATLAB with its extension SIMULINK for simulating dynamical systems, which both are intensively used in the engineering sciences. In addition, the Gaia community makes use of the fact that there exist many supporting libraries written in Java for advanced mathematical computations, analysis, graphics, etc.

3.1 Model description

The GAM is based on a modular design (Fig. 3) containing several blocks, each of them providing a specific functionality (cf. the sections beginning with 3.2). The four major blocks are incorporated by a closed loop (CL) configuration to represent the iterative nature of the simulation process. The core module “**Dynamics/Kinematics**” symbolizes the formulated satellite’s equations of motion. The dynamics of the Gaia spacecraft is considered to be that of a rigid-body with simplifications (Wertz (Editor)). The equations of motion are integrated by means of a widely-used solver for ordinary differential equations (ODE). The output of this module is the state vector containing the real state of the spacecraft, i. e. the 3-component actual rate value plus the 4-component actual attitude in form of a quaternion (see subsection 3.2). This state propagates to the model block “**Sensors**” (subsection 3.4) to simulate the behaviour of the two sensor devices, the SM-AF1 sensor for the rate measurements, and the STR sensor for the attitude determination, respectively. The simulated measurements from both devices are processed by including noise-induced data superposed to the measurements and by taking an ‘internal’ time-delay into account. Both data streams are then combined (using a second time-delay line) to the sensor’s output data in form of simulated measurements of the spacecraft’s attitude and rate. This output propagates to the third major block, the “**AOCS**” (subsection 3.5). It represents the part of the on-board software that is important for the control of the spacecraft’s manoeuvring during its various operational phases in orbit, particularly in the normal mode (NM), in which the scientific measurements are taken. In this sense, it processes the measurements from the sensors to estimate the current state of the spacecraft according to a predefined algorithm. In addition, the control of the satellite’s attitude motion is ensured by a minimization algorithm for the state using a Kalman-Filter (KF) description of the system. In the next step, the estimated state of the spacecraft will be compared with the state as given by the NSL at the specific instant of time. Any deviation in one (or more) of the components larger than

the allowed range of values, as given in the MRD (ESA Gaia Project (GAIA-EST-RD-00553)), will instantaneously be transformed into a requirement for a thrust to carry out. The definition of a thrust request as well as the determination of the optimal configuration of the thrusters that can execute this thrust in magnitude and direction is in the scope of the block “**Thrusters**” (subsection 3.3.2). Here, the activation algorithm is implemented and a thruster-specific noise model adds a noise component to the final output, the thruster torque. This torque is applied to the equation of motion in the dynamics block as one perturbation acting on the spacecraft.

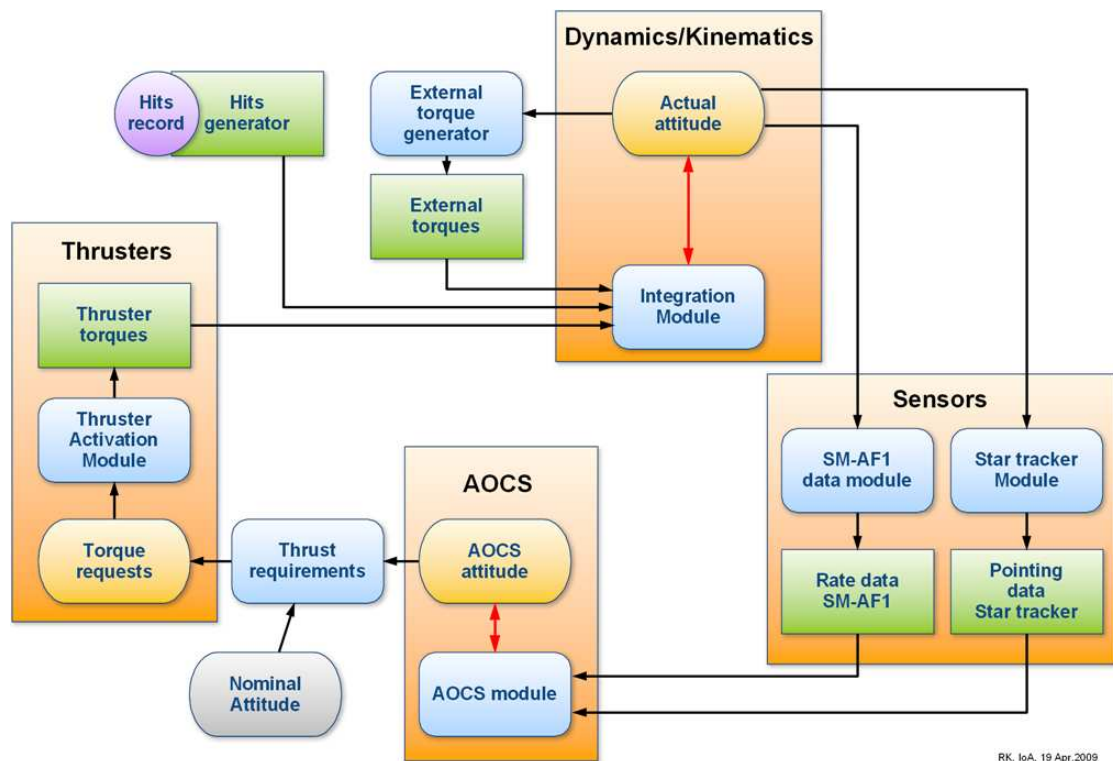


FIGURE 3: Global structure of the GAM designed as a closed loop (CL) configuration to visualize the iterative process of a simulation run. The essential parts are the “External torque generator” and the “Dynamics/Kinematics” blocks with the output ‘actual attitude/rate’, the “Sensors” blocks with the processing of measurement data to allow the third block “AOCS” to determine the ‘estimated attitude/rate’, and finally, the “Thrusters” block with the functionality of thruster activation based on deviations of the estimated attitude/rate from the target, the ‘NSL attitude/rate’. Due to its modular design this model can easily be extended according to specific needs.

Other, external perturbations, are summarised in the block “**External torque generator**” (subsection 3.3). Currently, this block contains the code of the solar radiation pressure model, representing the largest disturbance source for Gaia, a stochastic model of impacts caused by (micro-)meteoroids and a second stochastic model of ‘clanks’, i. e. discontinuities in the scan-phase of the satellite of yet unknown origin.

The simulation software allows one or more disturbances to be activated/deactivated simply by boolean flags. The initialization procedure defines mandatory parameters for a simulation scenario, such as the simulation duration, the simulation time step, etc and for the spacecraft configuration, such as the matrix of inertia. In addition, several initialization files list adjustable parameters for all models used in the GAM and enable the associated method to generate event tables according to the activation flag of a certain type of disturbance. These can be called once the simulator environment is started. In general, the modular design of the GAM allows smooth adjustments and clear extensions to investigate specific effects.

The next sections will describe the major blocks in more detail, but it will also guide to the technical notes written for the so far implemented external disturbance models.

3.2 Dynamics/Kinematics

The state vector x of the spacecraft at any instant of time during the simulation run is given by a combination of its orientation, formulated in quaternion notation (\mathbf{q}), and its angular velocity (or rate vector ω) with respect to the SRS (see Bastian (BAS-003), Lindegren (SAG-LL-030) and appendix A of Hobbs & Lindegren (DH-002) for further information on quaternions):

$$x = \begin{pmatrix} \mathbf{q} \\ \omega \end{pmatrix} \quad (1)$$

The first four components represent the attitude quaternion $\mathbf{q} = (q_x, q_y, q_z, q_w)$ and the latter three describe the spin rate vector $\omega = (\omega_x, \omega_y, \omega_z)$.

The spacecraft's behaviour is fully described by a superposition of a translational and a rotational motion written as two sets of ODEs, the first deals with the kinematic aspects of this motion, and the last with the dynamics of the spacecraft's body.

Kinematics. The following ODE describes the satellite's attitude in the quaternion formalism:

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\Omega(\omega)\mathbf{q}(t), \quad (2)$$

where $\Omega(\omega)$ is defined by

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}, \quad (3)$$

or, rewritten by its components,

$$\dot{\mathbf{q}}_{1..3}(t) = \frac{1}{2}(\mathbf{q}_4\omega - \omega \times \mathbf{q}(t)) \quad (4)$$

and

$$\dot{\mathbf{q}}_4(t) = -\frac{1}{2}\omega^T \mathbf{q}(t), \quad (5)$$

with the cross product $\omega \times \mathbf{q}$ given by

$$\omega \times \mathbf{q} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (6)$$

as shown by Wie (2008), p. 326ff.

Dynamics. The ODE containing the dynamics of the spacecraft as a rigid-body makes use of the Euler's equations

$$\dot{\omega}(t) = I_{SC}^{-1}(T_{total} - \omega \times I_{SC} \cdot \omega), \quad (7)$$

where I_{SC} is the moment of inertia of the satellite and T_{total} is the total sum of disturbance and control torques acting on the S/C (e. g. Wie (2008), p. 340f).

Both sets of ODE are implemented in the associated block (see Fig. 4), together with the integrator algorithms for solving them numerically, the Euler algorithm and the Runge-Kutta-algorithm of 4th order, respectively (selectable by a flag).

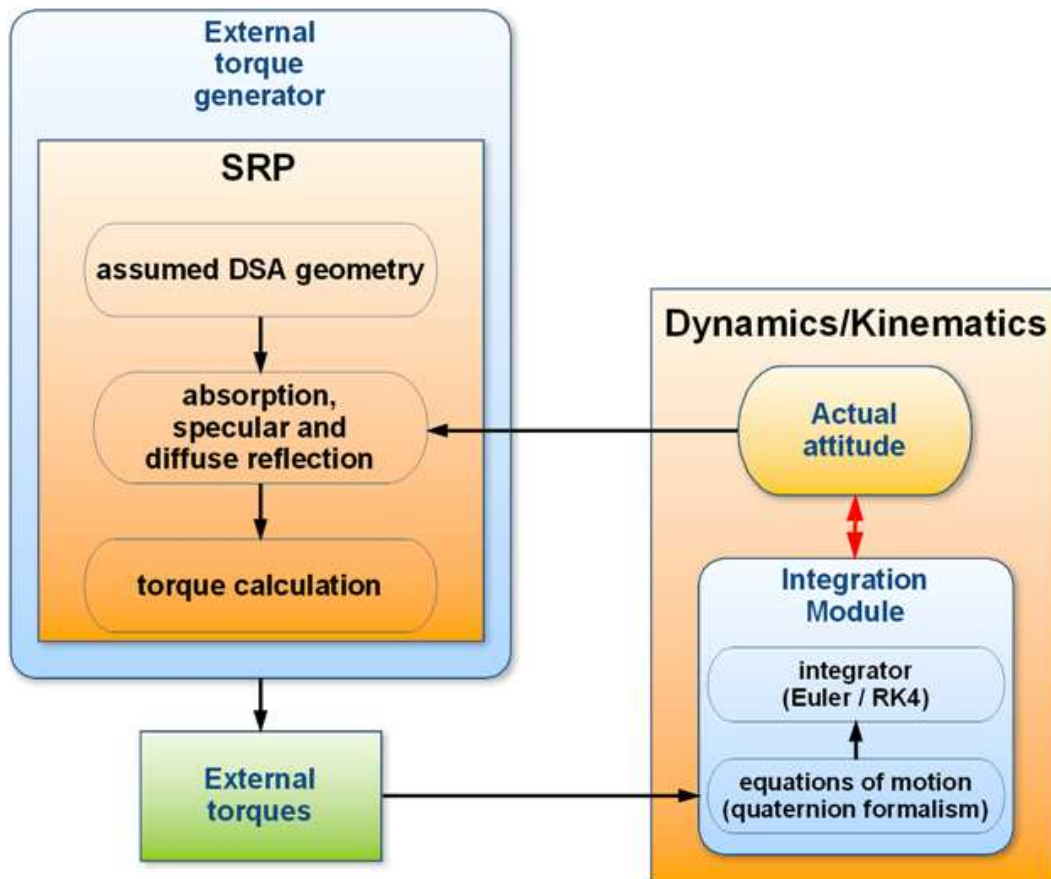


FIGURE 4: The interaction of the “Dynamics/Kinematics” block with its two sets of ODE describing the motion of the spacecraft with the block “External torque generator”, containing the general description of the calculation of the solar radiation pressure (SRP) acting on the SC.

3.3 Disturbances

Many perturbations will affect the orbital motion of the Gaia spacecraft. The larger the desired accuracy of the spacecraft’s attitude is, the larger is the number of perturbations one has to take into consideration. These include external (environmental) as well as internal perturbations (e.g. from the micro-propulsion system MPS in form of thruster firings). The software GAM is able to handle torques of arbitrary origin. The currently developed model contains the solar radiation pressure (SRP) as the major contribution of an external disturbance torque model plus stochastic models for (micro-)meteoroid impacts and discontinuities in the satellite’s scan-phase as detected for the first time in the HIPPARCOS data (van Leeuwen (2007)). The largest internal torque originates from actuation processes due to the micro-propulsion system MPS for continuously controlling the attitude.

3.3.1 External disturbances

The SRP. As already mentioned before, primarily the SRP contributes to the total torque budget of external disturbances. The modelling of the SRP for the GAM was divided into three parts, as it is indicated in Fig. 4. The geometry of the spacecraft’s DSA is modelled by a grid of triangles, representing the surface elements. As part of the initialisation process, this grid can be adjusted simply by changing one or more free parameters (e. g. the number of triangles). For each element, the force acting on it is calculated based on three different optical effects: absorption, specular and diffuse reflection. Afterwards, the associated torque is determined. Finally, the torques of all sun-shield elements are summed up to result in the total SRP-torque. Its quantity is given by $T_{\text{SRP}} = (129.4 \pm 0.1) \mu\text{Nm}$. In the next step, this will be added up with torques from other disturbance sources to propagate to the equations of motion in the “Dynamics/Kinematics” block. For a more detailed study on the theoretical background and further results particularly on different DSA geometries, see Risque (DRO-001).

The hits generator. This block represents the modelling of two different processes, the stochastic models for (micro-)meteoroid impacts as well as for discontinuities in the satellite’s scan-phase (van Leeuwen (2004FVLNOCODE)).

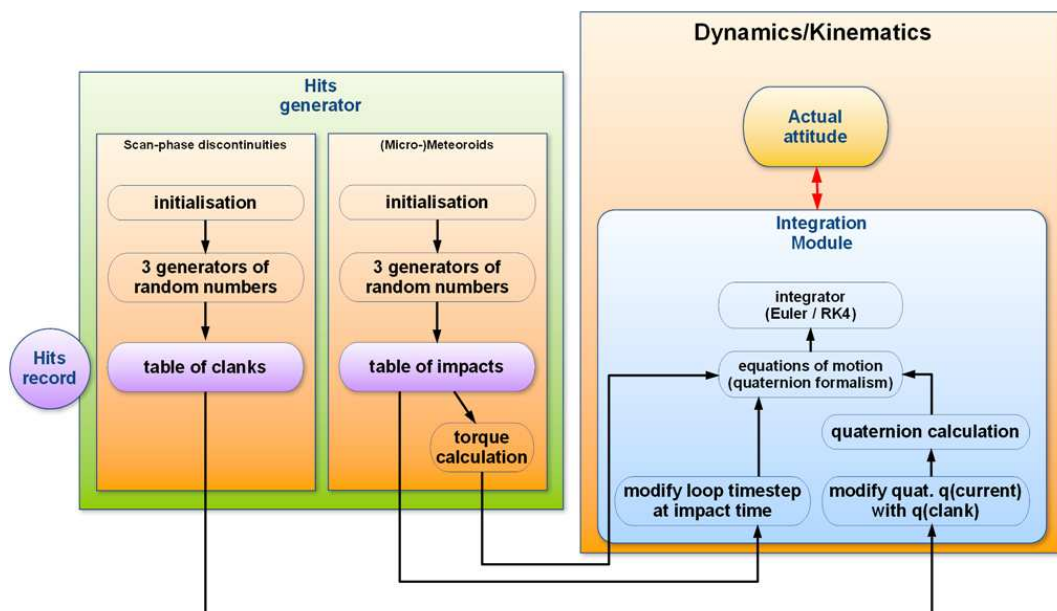


FIGURE 5: The hits generator with its current two components: the (micro-)meteoroid impacts model and the clank model to simulate discontinuities in the satellite’s scan-phase. Both models are based on tables, created through the initialisation process, each listing the information on time, direction, and magnitude of an impact and clank, respectively. Both data tables are fed into the “Integration Module” of the “Dynamics/Kinematics” block to be analysed.

Both make use of three independent random processes to generate a list of events, one event per line, as the initialising set up of each model. It contains the following information: the instants of time for the hit/clank to occur, the direction information, and the magnitude of each event. In case of the clanks, an event is given by the three components of a unit vector and its magnitude. Due to the assumption that clanks will only affect the attitude of the spacecraft, the quaternion associated with a clank, will modify the current quaternion in the integration loop (Fig. 5) before solving the equations of motion. For the (micro-)meteoroids, the table data affect the loop in two ways. Firstly, they allow to calculate the resulting torque that proceeds directly to the equations of motion, but secondly, the time instant of an impact will split the simulation time step into two smaller time steps, one before the impact, and the other one thereafter, resulting in one more iteration step inside the integration loop (see Risquez (2009) for a closer look on the computation of the magnitude of a (micro-)meteoroid hit and on the integration of this model into the GAM; see also Lindegren (SAG-LL-031) and Hobbs & Lindegren (DH-003)). As a consequence, this model will only disturb the rate of the spacecraft in a stochastic way, but ignore its attitude values.

3.3.2 Internal disturbance: The MPS

The thruster module masks the MPS to be used in the NM for keeping the satellite in a state of acceptable accuracy (with respect to the numerous simultaneous requirements on pointing and stability). The MPS consists of six active thruster devices (and another six for redundancy) with a maximum nominal thrust level of $500 \mu\text{N}$ each to enable displacements on small angular scales. As can be seen from the global GAM structure (Fig. 3) the “Thrusters” block receives its input, the requirement for executing a thrust of certain level, from a comparison of the estimated values of the spacecraft’s state (from the AOCS module) with the expected values from the NSL. As a function of this difference a torque is calculated and requested. The underlying problem is to find the optimal configuration of thrusters to be switch on in order to provide this torque in magnitude and direction, together with a minimum of propellant consumption, according to the following equation

$$\tau_{\mathbf{r}} = A \cdot \mathbf{F}_{\mathbf{C}}, \quad (8)$$

where

$\tau_{\mathbf{r}}$ is the torque request [Nm], in SCRS,

A is the thruster activation matrix, containing information about the thruster moment arms [m], and

$\mathbf{F}_{\mathbf{C}}$ is the commanded thrust vector [N].

Because the positions and directions of the thrusters are given by the spacecraft design, thus the matrix A is known, it is straightforward to calculate the requested torque coming from the thrusters as soon as the set of thrusters to provide this torque is known (“Torque requests”). (Here, we neglect any time-dependent effects, such as an expected drift of the CoM during the mission’s lifetime, and parameter drifts of thruster characteristics.) Therefore, the mathematical problem is to invert the non-square matrix A in the equation (8) to gain the thrust firings command information. With the help of pseudo-inverse matrices a solution can be found with an optimal linear combination of thrusters to be activated (“Thruster Activation Module”). At this stage, the commanded thrust are converted into produced forces (“Thruster torques”), taking care of technical details and limitations of the thruster devices, such as the thruster saturation limit, the quantization of thruster firings, minimum allowed thrust, and, last but not least, the thruster noise model. All free parameters for this calculation, mandatory or not, are changeable within certain limits at the beginning of each simulation run.

The total domain of actuator systems, including the MPS, as it is presented here, is subject to the willingness of the manufacturer to provide and exchange information and data with the scientific community. On the one hand, the development of the GAM benefits greatly from this opportunity. On the other hand, it should be clear that not every single information or algorithm can be distributed or published to others not involved in such a work. The same is true for the implementation of the sensors and AOCS modules (see the next two sections).

3.4 Sensors

This block comprises the activities to simulate measurements coming from the SM-AF1 device and the Star Tracker (STR), respectively. The idea is to mimic the procedures the industry is developing for the real sensor devices in order to create a model for the rate and the attitude determination based on (simulated) measurement data.

The rate measurement principle is based on the detection of an object in each of the two telescopes by one of the SM-CCDs and afterwards confirmed by the AF1-CCDs. The precise position of an object’s image in the focal plane depends largely on the accuracy of the distance and geometry calibration of this plane. If this is known to a high degree, then the position of the object will be measured in both detectors to result in an ‘estimated’ position at the time of the CCD read-out. This gives directly an estimated value of the speed of the object across and along the CCDs. Finally, the rate components of the spacecraft can be determined as a measured quantity by means of the speed measurements and the telescope geometry.

The purpose of the STR is to determine a quaternion that represents the attitude of the spacecraft (more precisely, the STR measurement reference system) with respect to the ICRS. The FoV of the STR allows the identification of stars by comparing their mutual positions on its focal plane with data from an on-board star catalogue. These measurements are sent to various internal algorithms to accurately determine the 3-axis attitude of Gaia at any time, when the telescopes

are pointing to the celestial sphere.

In brief, the steps can be summarised as follows:

- input the actual attitude quaternion (in ICRS) from the “Dynamics/Kinematics” block,
- calculate the corresponding quaternion of the STR orientation in order to simulate the real measurement process of the STR,
- apply a time-delay to incorporate two effects: the integration time of a CCD and the data processing time of the STR,
- generate measurement noise in the axes of the STR following the available noise model and calculate the error quaternion accordingly to apply the latter to the attitude quaternion of the STR axes,
- process the measurements to simulate the STR output,
- determine the STR axes quaternion (in ICRS),
- output the measured spacecraft attitude quaternion in the ICRS.

Finally, the data streams of both devices have to be combined to allow a smooth transition to the next module. An additional time-delay needed to be coded in order to account for the time shifts.

Again, this section can not be described in a comprehensive manner, because the developments of industry-specific HW and SW are not open for general usage. Because of the NDA between the prime contractor EADS Astrium and the group working on GAM, however, we have insights into the elements of the herein before mentioned aspects and, thus, can implement them as necessary in the GAM to continue the progression of the GAM coding as realistic as possible.

3.5 AOCS

The “AOCS” block represents all on-board algorithms that are of important to determine and control the attitude and rate of the spacecraft. It is an essential component of every simulation software, because it will judge by its design, if the spacecraft’s behaviour (within the given limits) allows a smooth operational phase with long observation periods or the motion of the SC has to be corrected frequently to fulfil the requirements.

The on-board realisation SW of the AOCS takes as an input the speeds of stars, as determined from measurements by the SM-AF1 sensor unit and the measured attitude quaternion of the

spacecraft from the STR operations (Fig. 3). Both are processed in the algorithms of the AOCS to primarily compute the (unbiased) estimated rate vector and attitude quaternion of the spacecraft at 1 Hz-resolution.

The estimation process comprises of basically four blocks of different functionality:

1. In the initialisation phase a **pre-processing of the measured star rate data** has to be performed, combining up to four different measurements (depending on the stars availability): for each of the FoV one rate measurement in the AL direction and another one in the AC direction. Including time-delay information between both data streams, this processed data is going to be sent to the rate estimation algorithm.
2. The design of the **rate estimator algorithm** is based on a Kalman-Filter (KF) approach to determine estimated values for the spacecraft's rate and for the disturbance torque (in the SCRS). The first step is to calculate this torque based on a simple rigid-body model of the dynamics of the spacecraft. In the next step, the estimated value of the disturbance torque is added to the control torque (from the controller block output as a torque request; see below) to deduce an estimated rate (the propagation step). After a measurement prediction step for the rate in all three axes of the SCRS with a subsequent error determination between the measured and the predicted values, the last step consists of the computation of the correction values for the rate estimation values as well as for the torque estimation value, making use of the four different rate measurements and the estimated values from the previous call of the algorithm (the update step).
3. The rate estimate from the rate estimator and the measured attitude of the spacecraft (from the STR) propagates to the next estimator, the **attitude estimator** to gain an estimated value for the attitude of the spacecraft. In its propagation step, the rate estimate allows to determine the current attitude estimate, whereas the STR attitude is used in the update step to correct the attitude estimate according to newly incoming measured attitude data.
4. The **controller** block represents the functionality of the AOCS to produce torques of certain magnitude and direction based on an analytical description, the control law. This control torque can be considered as a correcting torque to the estimated disturbance torque in the rate estimator algorithm and, thus, it takes over the task to control the satellite within certain limits of its motion. At the designing stage of the controller structure, special care had been required to fulfil the rate requirements for the satellite (ESA Gaia Project (GAIA-EST-RD-00553)), because they are much more strict than the requirements on the attitude. It is, however, in any case a very demanding task to find the optimal solution between a simple and robust design. For this reason, the controller design, as it was done by EADS Astrium, is also subject to the NDA, and the mentioned statements at the end of the previous two sections are also valid for the implementation of this part into the GAM.

4 Example simulation

The following text gives a brief overview of the steps necessary to initialise and run a simulation scenario.

(1) Initialise the simulation environment/propagation

1. assign/modify simulation parameters
2. choose integrator (Euler / RK4)
3. set/change initial conditions
(Note: the initial time $t_{\text{initial}} = 0.0$ (s) corresponds to the initial GaiaTime 2010-01-01T00:00:00.000 in TCB)
4. set tolerance(s) and step size(s)

(2) Set up the spacecraft environment

1. assign/modify spacecraft parameters
2. set/change initial conditions
3. modify matrix of inertia (elements)

(3) Set up perturbation models

1. set/unset any perturbation model(s)
2. assign perturbation model parameters
3. create event tables for perturbation model(s)

(4) Analyse the simulation output

1. export to external file
2. plot states or other parameters
3. extract state histories

As an example, the following configuration has been chosen to give the output table listed on the next page:

(1) Initialise the simulation environment/propagation

1. $t_{\text{initial}} = 0.0$, $t_{\text{final}} = 10000.0$, $\text{timeStep} = 1.0$ (all seconds)
2. chosen integrator: Euler
3. initial conditions:
 $\mathbf{q} = (q_x, q_y, q_z, q_w) = (0.0, 0.0, 0.0, 1.0)$, $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z) = (0.0, 0.0, \omega_{\text{spin}})$, with
 $\omega_{\text{spin}} = 2.91 \cdot 10^{-4}$ rad/s

(2) Set up the spacecraft environment

1. mass = 2120.0 kg
2. full inertia matrix of the Gaia spacecraft
3. CoM position [m] (in SRS)

(3) Set up perturbation models

1. switched on: SRP, (micro-)meteoroids, clanks
2. a) SRP parameters: simple DSA geometry (one triangle for each of the three sun-shield materials), coefficients of all three optical effects for all three sun-shield materials
b) (micro-)m.: $\text{EVENTS_PER_SECOND} = 0.001$ (s^{-1}), $\text{DISC_SURFACE} = 82.355$ (m, Gaia's cross section), derived parameters from Ernst (Matr.236604) study
c) clanks: $\text{CLANK_EVENTS_PER_SECOND} = 0.001$ (s^{-1}),
 $\text{DEVIATION_1_SIGMA_SEC} = 600$ (s; clank events at 1σ time deviation),
 $\text{JUMP_SLOPE} = 1.5$, $\text{JUMP_MIN} = 20$ (mas), $\text{JUMP_MAX} = 120$ (mas)
(power law parameters for the clanks magnitude distribution)
3. event tables: micrometeoroidsTable.dat, clanksTable.dat

(4) Analyse the simulation output

1. export to external file: output_actualattitude.dat (see next page)

Time of the simulation [s]	actual rate X [1e-6rad/s]	actual rate Y [1e-6rad/s]	actual rate Z [1e-6rad/s]	actual quat. 1 [-]	actual quat. 2 [-]	actual quat. 3 [-]	actual quat. 4 [-]	event flag
0.000000	-0.023435	0.036294	290.885504	0.00000000	0.00000000	0.00014544	0.99999999	0
1.000000	-0.046870	0.072586	290.882799	-0.00000001	0.00000002	0.00029089	0.99999999	0
2.000000	-0.070297	0.108885	290.880093	-0.00000004	0.00000005	0.00043633	0.99999999	0
3.000000	-0.093714	0.145190	290.877388	-0.00000007	0.00000011	0.00058177	0.99999998	0
4.000000	-0.117123	0.181500	290.874683	-0.00000012	0.00000018	0.00072721	0.99999997	0
5.000000	-0.140523	0.217817	290.871977	-0.00000018	0.00000027	0.00087264	0.99999996	0
6.000000	-0.163914	0.254139	290.869272	-0.00000025	0.00000038	0.00101808	0.99999994	0
7.000000	-0.187296	0.290468	290.866567	-0.00000033	0.00000051	0.00116351	0.99999992	0
8.000000	-0.210669	0.326802	290.863862	-0.00000042	0.00000065	0.00130895	0.99999991	0
9.000000	-0.212484	0.329625	290.863652	-0.00000043	0.00000067	0.00132025	0.99999991	0
9.077684	-0.234024	0.363149	290.861158	-0.00000053	0.00000082	0.00145438	0.99999989	1
10.000000	-0.257379	0.399496	290.858454	-0.00000065	0.00000100	0.00159981	0.99999987	0
11.000000	-0.280725	0.435848	290.855749	-0.00000077	0.00000120	0.00174524	0.99999984	0
12.000000	-0.304063	0.472207	290.853045	-0.00000091	0.00000142	0.00189067	0.99999981	0
13.000000	-0.327391	0.508571	290.850340	-0.00000107	0.00000165	0.00203609	0.99999979	0
14.000000	-0.350711	0.544941	290.847636	-0.00000123	0.00000191	0.00218152	0.99999976	0
15.000000	-0.374022	0.581317	290.844931	-0.00000141	0.00000218	0.00232694	0.99999972	0
16.000000	-0.397324	0.617700	290.842227	-0.00000159	0.00000247	0.00247236	0.99999969	0
17.000000	-0.420617	0.654088	290.839523	-0.00000179	0.00000278	0.00261778	0.99999965	0
18.000000	-0.443900	0.690482	290.836819	-0.00000200	0.00000310	0.00276320	0.99999961	0
19.000000	-0.467142	0.726828	290.834119	-0.00000223	0.00000345	0.00290841	0.99999957	0
19.998541	-0.467175	0.726882	290.834115	-0.00000223	0.00000345	0.00290862	0.99999957	0
20.000000	-0.490836	0.763163	290.831060	-0.00000246	0.00000381	0.00305404	0.99999953	0
21.000000	-0.514093	0.799575	290.828356	-0.00000271	0.00000419	0.00319945	0.99999948	0
22.000000	-0.537341	0.835993	290.825653	-0.00000296	0.00000459	0.00334487	0.99999944	1
[...]								



The output file shows one simulation time step in each row. The columns contain the following information:

- First column: current time of the simulation. Shown here are the first 26 seconds of a 10000 second simulation run,
- Second - fourth columns: all three components of the actual rate vector, in units of 10^{-6} rad/s,
- Fifth - Eighth column: all four components of the quaternion representing the actual attitude of the Gaia spacecraft.
- Ninth column: event flag. '0' means no event at that time, '1' (currently) indicates a (micro-)meteoroid impact or a clank event.

5 Summary

The GAM software allows to simulate the dynamics the Gaia spacecraft from a purely physical point of view, as its design shows the way to be as realistic as possible. The global programming structure is able to handle external and internal perturbations of arbitrary origin acting on the satellite. This includes environmental disturbances, such as the solar radiation pressure (SRP), (micro-)meteoroid impacts on the spacecraft's surface, and discontinuities in its scan-phase. In addition, GAM can deal with thruster firings from the MPS, appearing as internal disturbance source. The high flexibility of the GAM in terms of parameter adjustments or extensions of blocks makes it in ideal tool for any comparison between real measurements and simulation data – for the benefit of the accuracy of the scientific results.

Within a short time, the work on the last major blocks, the sensor's and the AOCS modules, will be finished. As soon as both are fully functional, all major parts are available to be incorporated into the global structure in order to accomplish the simulation loop. Even at the current stage, this simulation software can produce useful information for other simulation tools within the Gaia project. Although there are still a lot of tasks to be tackled, the simple handling of the software in terms of improvements and extensions, however, as well as the first results, show a clear way of progression towards an even more sophisticated modelling of the Gaia spacecraft.

6 References

- Angeli, F.D., 2005, *The Gaia Software Toolbox - User guide*, Tech. rep., IoA,
http://www.rssd.esa.int/SA-general/Projects/GAIA/wiki/index.php?title=CU1:_GaiaTools
- [**BAS-003**], Bastian, U., 2007, *Reference systems, conventions and notations for Gaia*,
GAIA-CA-SP-ARI-BAS-003,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/358698>
- [**GAIA.ASF.SP.SAT.00001**], EADS Astrium, 2007, *Satellite Requirements Specification (SRS)*,
GAIA.ASF.SP.SAT.00001,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2722565>
- [**Matr.236604**], Ernst, R., 2007, *Momentum transfer to orbiting satellites by micrometeoroid impacts*,
Matr.236604,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2814741>
- [**GAIA-EST-RD-00553**], ESA Gaia Project, 2006, *Mission Requirements Document (MRD)*,
GAIA-EST-RD-00553,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/463164>
- [**MTL-015**], GaiaTools Committee, 2008, *GaiaTools Software User Manual*,
GAIA-C1-UG-ESAC-MTL-015,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2858963>
- [**DH-002**], Hobbs, D., Lindegren, L., 2008, *Attitude Processing for Gaia*,
GAIA-C3-TN-LU-DH-002,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2843088>
- [**DH-003**], Hobbs, D., Lindegren, L., 2008, *Handling of micro-meteoroid impacts in the Gaia attitude update*,
GAIA-C3-TN-LU-DH-003,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2848392>
- [**RK-001**], Keil, R., Theil, S., 2008, *Modelling the attitude noise - A simplified approach*,
GAIA-C2-TN-ZARM-RK-001,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/2864753>
- [**UL-001**], Lammers, U., Joliet, E., 2009, *Gaia parameter database user manual*,
GAIA-CA-UG-ESAC-UL-001,
URL <http://www.rssd.esa.int/l1ink/liveliink/open/358718>
- van Leeuwen, F., 2007, *Hipparcos, the New Reduction of the Raw Data*, Springer, astrophysics and space science library. vol. 350 edn.

- [SAG-LL-030], Lindegren, L., 2000, *Attitude parameterization for GAIA*,
SAG-LL-030,
URL <http://www.rssd.esa.int/llink/livelink/open/358607>
- [SAG-LL-031], Lindegren, L., 2000, *Effects of micrometeoroids on GAIA attitude*,
SAG-LL-031,
URL <http://www.rssd.esa.int/llink/livelink/open/358616>
- [LL-073], Lindegren, L., Luri, X., 2007, *A model of noisy attitude for cycle 3 simulations*,
GAIA-C2-TN-LU-LL-073,
URL <http://www.rssd.esa.int/llink/livelink/open/2770181>
- [DRO-001], Riquez, D., 2009, *Physical Effects in the Gaia Attitude Simulation; Solar Radiation*,
GAIA-C2-TN-LEI-DRO-001,
URL <http://www.rssd.esa.int/llink/livelink/open/2881765>
- Riquez, D., 2009, *Physical Effects in the Gaia Attitude Simulation. Micro-Meteoroids*, Tech. rep., Leiden Observatory
- [STH-001], Theil, S., Keil, R., 2007, *Minutes of the joined CUs meeting on Gaia Attitude Modelling and Simulation*,
GAIA-C2-MN-ZARM-STH-001,
URL <http://www.rssd.esa.int/llink/livelink/open/2742870>
- [2004FVLNOCODE], van Leeuwen, F., 2004, *Scan phase discontinuities in the Hipparcos data*,
2004FVLNOCODE,
URL <http://www.rssd.esa.int/llink/livelink/open/391772>
- Wertz(Editor), J.R., 1978, *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, 1 edn.
- Wie, B., 2008, *Space Vehicle Dynamics and Control*, American Institute of Aeronautics and Astronautics, Reston, VA, 2nd edn.

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart Veröffentlichung		
3a. Titel des Berichts First Look - Fast Initial In-Orbit Identification of Scientific Satellites			
3b. Titel der Publikation High Performance Satellite Dynamics Simulator for Scientific Missions			
4a. Autoren des Berichts (Name, Vorname(n)) Keil, Ralf Maat, Matthias		5. Abschlussdatum des Vorhabens 30.06.2009	
		6. Veröffentlichungsdatum	
4b. Autoren der Publikation (Name, Vorname(n)) Bindel, Daniel; Pelivan, Ivanka; Bremer, Stefanie; List Meike; Theil, Stephan; Romagnoli, Daniele		7. Form der Publikation Konferenzband	
8. Durchführende Institutionen(en) (Name, Adresse) ZARM – Zentrum für angewandte Raumfahrttechnologie und Mikrogravitation Universität Bremen Am Fallturm 28359 Bremen		9. Ber.Nr. Durchführende Institution	
		10. Förderkennzeichen 50 OY 0502	
		11a. Seitenzahl Bericht 535	
		11b. Seitenzahl Publikation 5	
13. Fördernde Institutionen (Name, Adresse) Deutsches Zentrum für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn		12. Literaturangaben	
		14. Tabellen	
		15. Abbildungen	
16. Zusätzliche Angaben Vortrag am 8. September 2009, KonferenzNr. DLRK 2009-1265			
17. Vorgelegt bei (Titel, Ort, Datum) Deutscher Luft- und Raumfahrtkongress 2009, Aachen, 8.-10. September 2009			
18. Kurzfassung Das Projekt "First Look - Fast Initial In-Orbit Identification of Scientific Satellites" wurde am Zentrum für angewandte Raumfahrt-technologie und Mikrogravitation (ZARM) durchgeführt. Die Laufzeit war von 01/07/2005 bis 30/06/2009. Ziel des Projekts war die Entwicklung von Algorithmen und Modellen zur In-Orbit-Identifikation von Satelliten und Experiment-parametern bei wissenschaftlichen Missionen, um damit die Qualität der Ergebnisse sicherzustellen. Mit der voranschreitenden Entwicklung von Technologien verwenden Wissenschaftsmissionen immer sensiblere Sensoren, die eine extrem störungsfreie Umgebung benötigen. Beispiele hierfür sind STEP, Gaia oder die Gravity Probe B Mission (GP-B). Infolge dieser Entwicklung sind zum einen die Messgeräte sehr stark mit dem Satellitenbus verbunden. Zum anderen benötigt die Prozessierung der Messdaten einen langen Zeitraum, ehe die wissenschaftlichen Ergebnisse vorliegen. Man muss daher zu Beginn einer Mission sicherstellen, dass die heute aufgenommenen Messwerte nicht mit unbekanntem nicht-detektierbaren Fehlern behaftet sind, die das Endergebnis in mehreren Jahren verfälschen. Dazu werden Methoden zur schnellen In-Orbit Identifikation benötigt, die sehr schnell erkennen können, ob und wie das Gesamtsystem funktioniert. Genau an diesem Punkt setzt das Vorhaben "First Look" an. Es soll eine Grundlage schaffen, die für alle zukünftigen Missionen angewendet werden kann. Dazu werden die Erfahrungen aus vergangenen und aktuellen Missionen ausgewertet und die Synergien zwischen diesen Missionen genutzt, um allgemeine Methoden und Ansätze abzuleiten.			
19. Schlagwörter HIPPARCOS, Gaia, GP-B, STEP, Drag-Free-Satellit, Simulation, Störkräfte			
20. Verlag		21. Preis	

Document Control Sheet

1. ISBN or ISSN	2. Type of Report Publication
3a. Report Title First Look - Fast Initial In-Orbit Identification of Scientific Satellites	
3b. Title of Publication High Performance Satellite Dynamics Simulator for Scientific Missions	
4a. Author(s) of the Report (Family Name, First Name(s)) Keil, Ralf Maat, Matthias	5. End of Project 30.06.2009
	6. Publication Date
4b. Author(s) of the Publication (Family Name, First Name(s)) Bindel, Daniel; Pelivan, Ivanka; Bremer, Stefanie; List Meike; Theil, Stephan; Romagnoli, Daniele	7. Form of Publication Proceedings issue
8. Performing Organization(s) (Name, Address) ZARM – Zentrum für angewandte Raumfahrttechnologie und Mikrogravitation Universität Bremen Am Fallturm 28359 Bremen	9. Originator's Report No.
	10. Reference No. 50 OY 0502
	11a. No. of Pages Report 535
	11b. No. of Pages Publication 5
13. Sponsoring Agency (Name, Address) Deutsches Zentrum für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn	12. No. of References
	14. No. of Tables
	15. No. of Figures
16. Supplementary Notes Presentation on September 8, 2009, KonferenzNr. DLRK 2009-1265	
17. Presented at (Title, Place, Date) Deutscher Luft- und Raumfahrtkongress 2009, Aachen, September 8-10, 2009	
18. Abstract The project "First Look - Fast Initial In-Orbit Identification of Scientific Satellites" has been realized. The duration was from 01/07/2005 to 30/06/2009. The purpose of the project was the development of algorithms and models for the in-orbit-identification of satellites and experimental parameters for scientific missions in order to assure the quality of the results. With the progress of the development of technologies scientific space missions are using more sensitive sensors, for which an extremely undisturbed environment is required. Examples are STEP, Gaia, or the Gravity Probe B Mission (GP-B). Due to this development, on the one hand, the on-board-instruments are linked very closely to the satellite bus. On the other hand, the processing of the data takes a long time until the scientific results are available. Therefore, with the development of a new mission it has to be guaranteed that the recorded data are not influenced by undetectable errors affecting the results for years. Thus, methods for a fast in-orbit-identification are required, i.e. to allow a fast identification of the system, whether and if so, how it is working. With the "First Look"-project a foundation shall be established that can be applied to future missions. For this reason, the experiences and knowledge of past and current missions are analysed. Synergy effects between the missions are used to derive general methods and approaches.	
19. Keywords HIPPARCOS, Gaia, GP-B, STEP, drag-free-satellite, simulation, disturbance forces	
20. Publisher	21. Price