

**Gemeinsamer Abschlussbericht
des
Forschungsverbundprojektes**



GUSMA

**„Gekoppelte Unternehmensübergreifende Simulation
Mobiler Arbeitsmaschinen
zur Virtualisierung der Produktentstehung“ (GUSMA)**



GEFÖRDERT VOM



**Bundesministerium
für Bildung
und Forschung**

BETREUT VOM



**PTKA
Projektträger Karlsruhe**
Karlsruher Institut für Technologie

Dieses Forschungs- und Entwicklungsprojekt wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmenkonzept „Forschung für die Produktion von morgen“ gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Autoren

**Karlsruher Institut für Technologie (KIT),
Institut für Fahrzeugsystemtechnik (FAST),
Lehrstuhl für Mobile Arbeitsmaschinen (Mobima)**

Prof. Dr. Marcus Geimer
Andreas Rüdener
Song Han
Dr. Lars Völker

AGCO GmbH / Fendt

Dr. Heinz Böhler
Thomas Wendler

HYDAC System GmbH

Dr. Frank Bauer
Peter Huth
Rüdiger Honsbein

Fluidon GmbH

Ronald Kett
Oliver Breuer

LMS Deutschland GmbH

Dennis Berghaeger
Anthony Domi

SIMPACK AG

Steven Mulski
Dr. Jennifer Paulin

Inhaltsverzeichnis

Kurzzusammenfassung	1
Abstract	1
1. Herausforderungen der Produktentwicklung im Kontext der unternehmensübergreifenden Simulation mobiler Arbeitsmaschinen.....	2
1.1 Produktkomplexität am Beispiel Komfortsysteme im Traktor	2
1.2 Simulation in der Produktentwicklung.....	3
1.3 Co-Simulation als Mittel zur unternehmensübergreifenden Simulation.....	5
2. Das Projekt GUSMA	9
2.1 Vorstellung und Kennzahlen	9
2.2 Ziele und Durchführung	9
3. Anforderungsanalyse.....	11
3.1 Schnittstellenanalyse bei Softwareherstellern	13
3.2 Anforderungen an eine zentrale Plattform	13
4. GUSMA-Plattform - Die drei Säulen der Standardisierung	17
4.1 Der Plattformgedanke	17
4.2 Standardinterface.....	18
4.3 Standardisierte Vorgehensweise	21
5. Auswahl eines geeigneten Modelldetaillierungsgrads.....	24
5.1 Vorgehen für Auswahl eines geeigneten Modelldetaillierungsgrad.....	25
6. Durchführung einer gekoppelten unternehmensübergreifenden Simulation am Beispiel einer hydropneumatischen Traktorvorderachsfederung mit Niveauregulierung.....	26
6.1 Funktionsweise der hydropneumatischen Traktorvorderachsfederung mit Niveauregulierung	26
6.2 Aufteilung der Modellierung entsprechend Hersteller – Zulieferer	29
6.3 Beschreibung der Teilmodelle und Aufbau der Systemsimulation	31
6.4 Belastungen am Realbauteil.....	62
6.5 Abgleich Prüfstand und Simulation.....	70
7. Zusammenfassung.....	75
8. Literaturverzeichnis	77
9. Anhang	78
9.1 Ist-Stand bei Software-Herstellern im GUSMA Arbeitskreis: Beschreibung der bestehenden Co-Simulations-Schnittstelle.....	78
9.2 Beispielhafte Implementierung des GUSMA-Standards.....	90
9.3 Erweiterte Beschreibung der Simulationsmodelle und –ergebnisse	117
9.4 Untersuchung relevanter Verluste in der hydropneumatischen Achsfederung	121
9.5 Erfahrungen aus der Simulation	131

Kurzzusammenfassung

Im Verbundforschungsprojekt „Gekoppelte Unternehmensübergreifende Simulation Mobiler Arbeitsmaschinen zur Virtualisierung der Produktentstehung“ (GUSMA) wurde in den Jahren 2008 bis 2011 das Ziel verfolgt, den unternehmensübergreifenden Aufbau virtueller Prototypen durchgängig zu ermöglichen und damit die Effizienz im Produktentwicklungsprozess maßgeblich zu erhöhen. Hierbei wurde die Technologie der gekoppelten Simulation aufgegriffen und unter Aspekten der Schnittstellenstandardisierung, sowie des Know-How-Schutzes eine funktionsfähige Co-Simulationsplattform unter Matlab/Simulink¹ aufgebaut. Mit diesen Maßnahmen wurde technisch die unternehmensübergreifende Produktentwicklung mit Hilfe der standardisierten Co-Simulation erstmalig nutzbar gemacht.

Dieses Forschungs- und Entwicklungsprojekt wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmenkonzept „Forschung für die Produktion von morgen“ gefördert und vom Projektträger Karlsruhe (PTKA) betreut (Förderkennzeichen 02 PC 1100 – 02 PC 1105). Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Abstract

In the years 2008 to 2011, the joint research project GUSMA, which is a German acronym for “Coupled Simulation of Mobile Machines between different business partners for the virtualization of the product design”, pursued the objective of enabling the effective cross-company build-up of virtual prototypes and in that way, to increase the efficiency of the product development process significantly. Therefore, the technology of coupled simulation was used and a fully functional co-simulation platform based on Matlab/Simulink¹ was developed taking the aspect of interface standardization, as well as know-how protection into account. Applying these measures, for the first time, cross-company product development was realized technically by using a standardized co-simulation.

This research and development project was funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept “Research for Tomorrow’s Production and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

¹ vgl. Fa. Mathworks, <http://www.mathworks.de/>

1. Herausforderungen der Produktentwicklung im Kontext der unternehmensübergreifenden Simulation mobiler Arbeitsmaschinen

Bereits seit mehreren Jahrzehnten werden computergestützte Simulationen eingesetzt, um reale technische Systeme nachzubilden. Der Zweck liegt hierbei insbesondere darin, das Systemverständnis zu erhöhen, Optimierungen am System durchzuführen oder dessen Verhalten vorherzusagen [1]. Dieser Mehrwert an Informationen soll im Produktentwicklungsprozess zu einer Reduktion von Entwicklungszeit und –kosten bei gleichzeitiger Steigerung der Produktqualität führen.

Um ein reales System in der Simulation computergerecht darzustellen, gilt es, dieses in einem Ersatzmodell abzubilden. Die hierfür notwendige Beschreibung der Modelle erfolgt in einem mehrstufigen Prozess. Im Folgenden soll der Leser über die Herausforderungen der Produktentwicklung sensibilisiert werden.

1.1 Produktkomplexität am Beispiel Komfortsysteme im Traktor

Die Branche der mobilen Arbeitsmaschinen ist gekennzeichnet durch die Entwicklung hoch komplexer Produkte. Hohe Vielfalt an Produktvarianten und geringe Stückzahlen, welche bis hin zur individuellen Einzelfertigung reichen können, setzen hohe Ansprüche an die Effizienz der stark von kleinen und mittleren Unternehmen (KMU) geprägten Branche. Am Beispiel der Traktoren der Marke Fendt der AGCO GmbH lässt sich hierbei das Spannungsfeld verdeutlichen, das auch viele andere Hersteller und Zulieferer von Mobilien Arbeitsmaschinen betrifft [2]. Auf der einen Seite werden hochkomplexe Systemfahrzeuge entwickelt, die viele Technologien aus den Bereichen Automatisierung, Hydraulik und Pneumatik mit der Fahrzeugmechanik verbinden, und dies bei immer kürzeren Entwicklungszeiten. Auf der anderen Seite zwingen die im Vergleich zum On Road Bereich (PKW, Truck) kleineren Stückzahlen und die dadurch kleineren Entwicklungsbudgets den Fahrzeughersteller dazu, die häufig noch im Aufbau befindlichen eigenen Simulationskapazitäten möglichst effektiv zu nutzen und dabei auch die Kompetenz der Zulieferer mit einzubinden.

Die Komplexität der Fahrzeuge soll beispielhaft an den Komfortsystemen eines Fendt-Großtraktors aus der Vario 900 Baureihe (**Abbildung 1.1**) vorgestellt werden [3].

- Hydropneumatische Einzelradfederung an der Vorderachse mit einer fahrgeschwindigkeitsabhängigen Wankabstützung (FSC: Fendt Stability Control) und mit automatischer Niveauregulierung
- Drei-Punkt-Kabinenlagerung mit Luftfederelementen mit automatischer Niveau-regulierung
- Fahrersitz mit semiaktiver Luftfederung und mit automatischer Niveauregulierung
- Aktive Schwingungstilgung des Krafthebers beim Gerätetransport im Heck
- Passive Federung des Krafthebers bei Gerätetransport in der Front
- Hohe Automatisierung, z.B. Motor-Getriebe-Management-System TMS mit dem stufenlosen VARIO Getriebe, Vorgewendemanagement Variotronic, GPS basiertes automatisches Spurführungssystem Varioguide

Diese Ausstattung ergibt einen Federungs- und Lenkkomfort auf LKW-Niveau und ermöglicht erstmals eine Höchstgeschwindigkeit von 60 km/h bei Standardtraktoren. Das Gesamtfederungspaket muss dabei auf unterschiedliche Fahrprofile und Einsätze abgestimmt werden.

Zukünftige Entwicklungen werden verstärkt aktiv regelnde Systeme beinhalten, so dass eine effektive Abstimmung z.B. des Fahrkomforts und eine Erprobung der dazugehörigen Software ohne Simulation kaum mehr möglich sein wird.



Abbildung 1.1 - Fendt Großtraktor der Serie 900 Vario (265 kW)

1.2 Simulation in der Produktentwicklung

Bereits seit mehreren Jahrzehnten werden im Produktentwicklungsprozess in der Industrie Simulationswerkzeuge eingesetzt, um reale technische Systeme nachzubilden. Der Zweck liegt hierbei insbesondere darin, das Systemverständnis zu erhöhen, Optimierungen am System durchzuführen oder dessen Verhalten vorherzusagen ([1], [4]). Insbesondere in frühen Phasen der Produktentwicklung hat sich dieser Mehrwert an Wissen als ein Erfolgsfaktor zur Reduktion von Entwicklungszeit und –kosten erwiesen, wie **Abbildung 1.2** in verdeutlicht wird. Denn mit zunehmendem Fortschritt im Produktentstehungsprozess sinken die Freiheitsgrade für Entscheidungen. Dementsprechend ist es von großer Bedeutung, das vorhandene Wissen über ein zu entwickelndes Produkt so früh wie möglich zu maximieren.

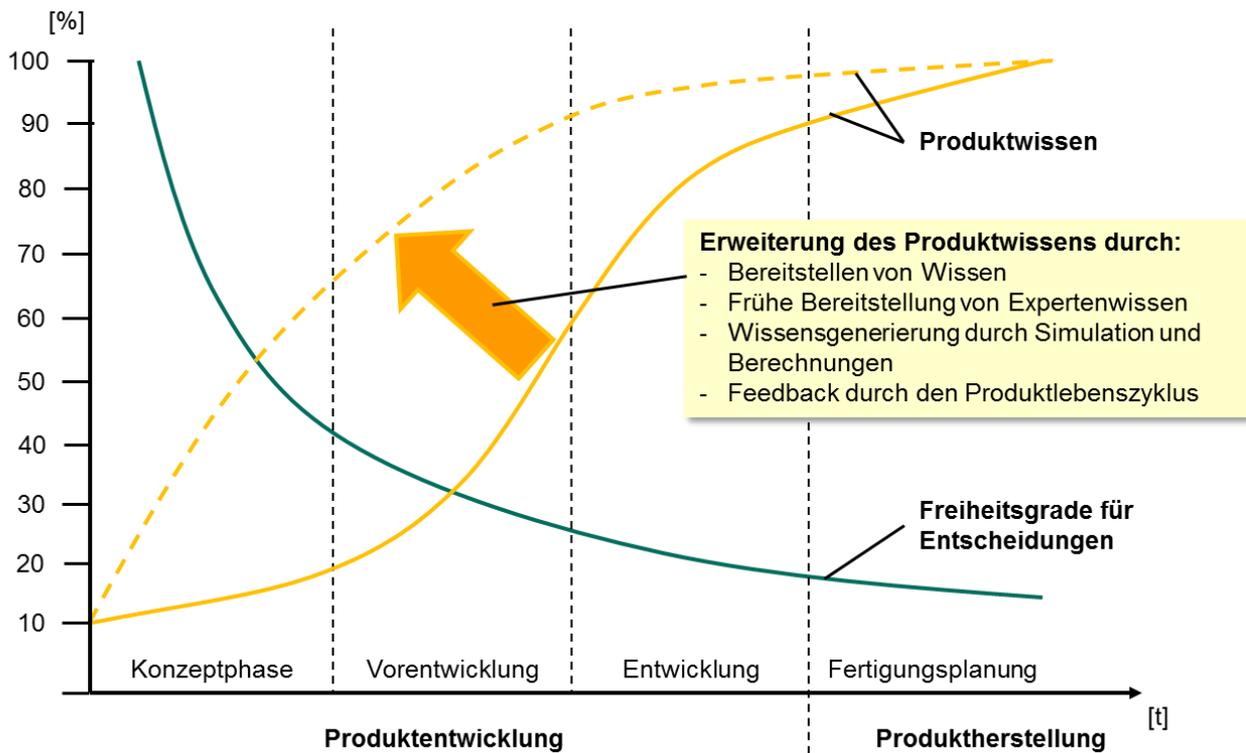


Abbildung 1.2 - Frühe Bereitstellung des Produktwissens (Quelle: L.-H. Krause, IPK Berlin)

Die rasche Durchdringung dieser sogenannten CAE-Technologien (Computer Aided Engineering) in der Automobil- und Luftfahrtindustrie bereits in den 1990er Jahren unterstreicht deren Wirtschaftlichkeit. Zunehmend haben auch die Unternehmen des Maschinen- und Anlagenbaus das Potenzial der computergestützten Berechnung für ihren Bereich erkannt und die Effizienz im Entwicklungsprozess steigern können. Denn gerade der stark mittelständisch geprägte Maschinen- und Anlagenbau stellt als größter industrieller Arbeitgeber und seinen vielen „Hidden Champions“ für Deutschland einen wesentlichen Motor für Wachstum und Wohlstand dar [5].

Die Branche der Mobile Arbeitsmaschinen repräsentiert hierbei ein typisches Anwendungsgebiet dieses Bereichs, an dem sich die Herausforderungen im Produktentstehungsprozess darstellen lassen. Neben der durch kleine und mittlere Unternehmen (KMU) geprägten Firmenlandschaft, erfordert das Produktspektrum effiziente und effektive Prozesse bei der Produktentwicklung. Einer hohen, an die Kundenbedürfnisse angepassten Variantenvielfalt stehen hierbei geringe Stückzahlen gegenüber, die bis hin zur Einzelfertigung reichen können. Mobile Arbeitsmaschinen sind hoch komplexe Systeme, in denen Komponenten der Mechanik, Hydraulik, Elektronik und zunehmend auch der Elektrik kombiniert sind. Hoch moderne Steuerungsalgorithmen, die bereits heute schon den autonomen Betrieb der Maschinen ermöglichen (z.B. [6]), oder der Trend zur Effizienzsteigerung durch neuartige Hydraulikkonzepte (z.B. [7], [8]) steigern die Komplexität der Maschinen enorm.

Bei der Entwicklung mobiler Arbeitsmaschinen müssen die Komponenten der entsprechenden Fachgebiete optimal aufeinander abgestimmt werden. Diese Vernetzung einzelner Komponenten erfordert daher die Betrachtung der Maschine als Gesamtsystem. Der Einsatz von Simulationswerkzeugen gilt auch in diesem Zusammenhang als viel versprechend und der positive Effekt des sogenannten „Frontloading“ – der Verlagerung von Handlungen in frühe Phasen des Entwicklungsprozesses und der daraus folgenden Kostensenkung [9] - soll somit auf Gesamtfahrzeu-

gebene in der Branche Einzug halten. Um aber eine mobile Arbeitsmaschinen in ihrer Gesamtheit als virtuellen Prototypen abbilden zu können, müssen die verschiedenen Fachdisziplinen – Mechanik, Hydraulik, Steuerungstechnik, etc. – abgebildet werden können und dies in einem bedarfsgerechten Detaillierungsgrad.

1.3 Co-Simulation als Mittel zur unternehmensübergreifenden Simulation

Um ein reales System in der Simulation computergerecht darzustellen, gilt es, dieses in einem Ersatzmodell abzubilden. Die hierfür notwendige Beschreibung der Modelle erfolgt in einem mehrstufigen Prozess und kann je nach Anforderung und Betrachtungsweise in verschiedene Modellierungsarten eingeteilt werden. Eine Möglichkeit ist die Unterscheidung von mathematischer und physikalischer Modellbeschreibung. Gemäß **Abbildung 1.3** wird als erster Schritt vom Anwender eine Modellbeschreibung durchgeführt. Hierfür wird zumeist ein physikalisches Ersatzmodell vom realen System abgeleitet. Auf Basis des vereinfachten (physikalischen) Modells kann das Modell auf mathematische Weise beschrieben werden.

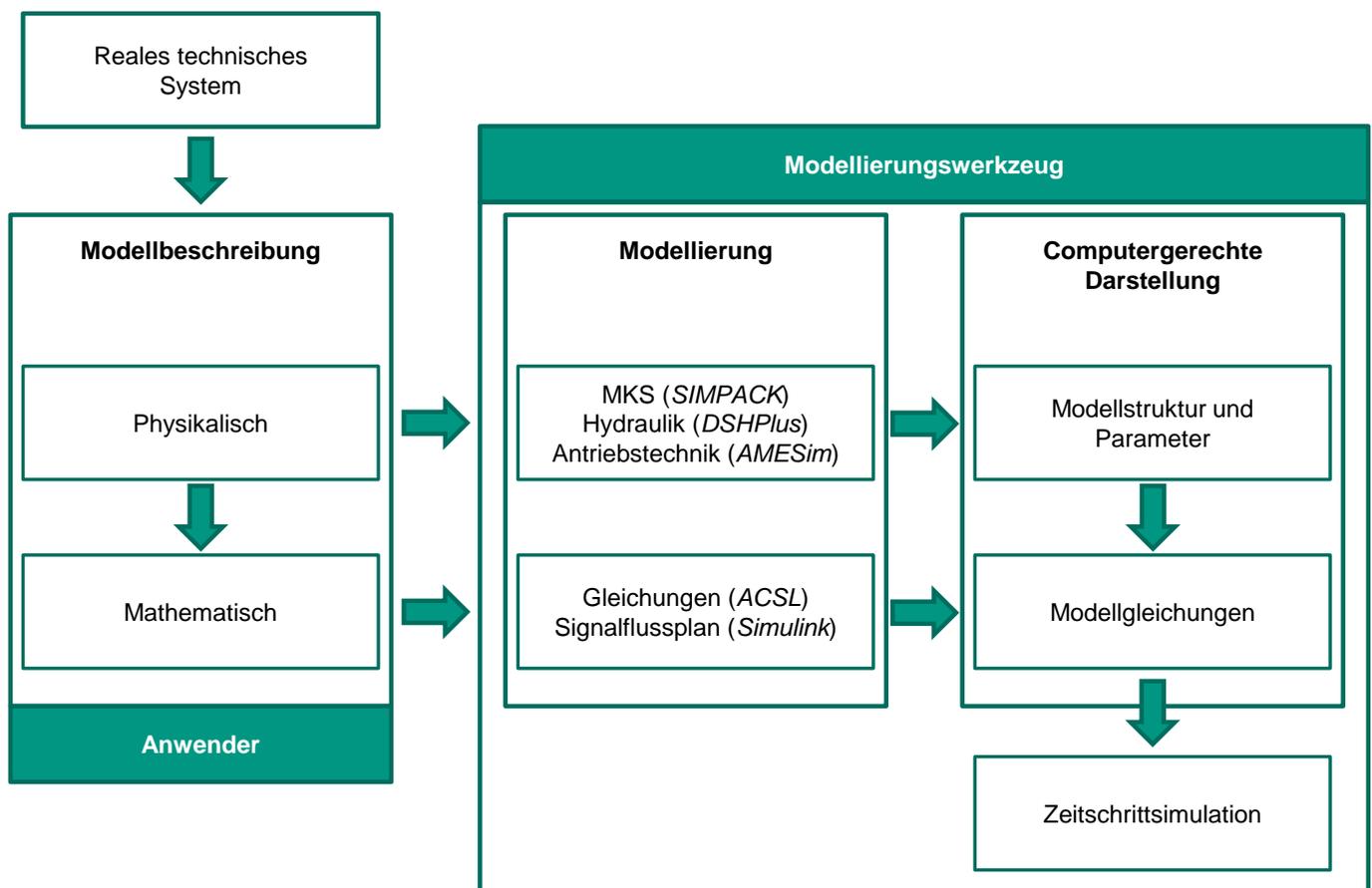


Abbildung 1.3 - Beschreibungsformen und Modellierungsprozess (in Anlehnung an [1])

Eine wesentliche Herausforderung hierbei ist es, die Modellierungsart in Abhängigkeit vorhandener Ressourcen zu wählen, unter welchen insbesondere die vorhandene Software, Hardware, wie auch das Know-How der Ingenieure genannt seien. Anschließend muss das Ersatzmodell in Abhängigkeit des Modellierungswerkzeuges und der vo-

rangegangenen Modellbeschreibung implementiert werden. Letztendlich erzeugen Software-interne Algorithmen in Anschluss an die Modellierung eine computergerechte Darstellung des Systems. Insofern das Modell in einer physikalischen Modellierungsumgebung erstellt wurde, stellt sich diese in Form einer Software-spezifischen Modellstruktur mit entsprechenden Parametern dar. Die für die Berechnung notwendigen Modellgleichungen werden intern von dieser Struktur oder im Falle der mathematischen Modellierung direkt beispielsweise aus den Gleichungen, beziehungsweise dem Signalflussplan abgeleitet.

Von diesen Modellgleichungen ausgehend können Analysemethoden durchgeführt werden, beispielsweise in Form von einer Zeitschrittsimulation. Auf diese Weise können die gewünschten Effekte in der Simulation für ein (Teil-) System dargestellt werden.

1.3.1 Modellierung im Kontext einer Gesamtsystembetrachtung

Aus Sicht eines mechatronischen Gesamtsystems ergibt sich die oben genannte Anforderung, die spezifischen Eigenschaften der jeweiligen Domänen Mechanik, Hydraulik, Steuerung et cetera entsprechend der Bedürfnisse des Benutzers zu modellieren. Neben den wie in **Abbildung 1.3** gezeigten mathematisch basierten Modellbeschreibungen, welche zum Beispiel in Form von Gleichungssystemen oder als Signalflussplan dargestellt werden können, existiert eine Vielzahl an spezialisierten Softwaretools zur physikalischen Beschreibung. Viele davon haben sich aus dem spezialisierten Fachgebiet heraus im Laufe der Zeit zu sogenannten Multidomain-Tools weiterentwickelt. Diese bieten die Möglichkeit komplette mechatronische Systeme in einem Simulationswerkzeug abzubilden. Ebenso erwähnt seien die Softwareprogramme, welche explizit zur Multidomain-Modellierung geschaffen wurden. Als ein Beispiel seien hier auf der Modelica-Sprache basierte Systeme genannt.

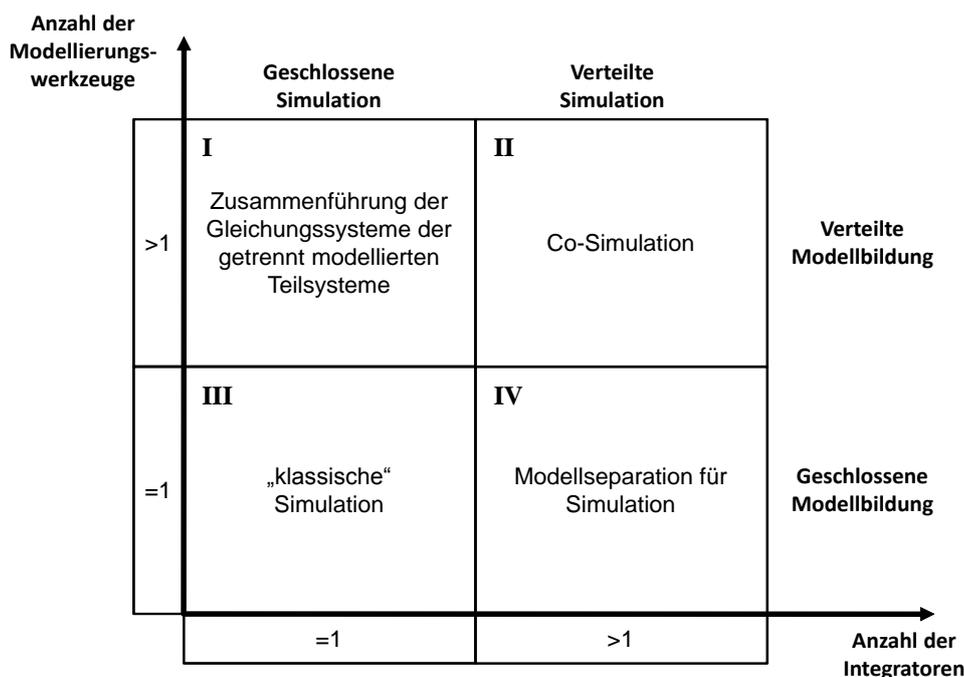


Abbildung 1.4 - Modellbildungsvarianten nach Geimer (siehe [10])

Die bisher genannte Modellbildungsform bezieht sich auf die Modellierung mit einem Modellierungswerkzeug unter Benutzung eines darauf optimierten Integrators. Nach Geimer (siehe [10]) wird dies in Abhängigkeit der beiden genannten Bezugsgrößen als „klassische“ Simulation bezeichnet. Für die Gesamtsystemsimulation bestehen allerdings weitere Modellierungsvarianten. Diese werden in **Abbildung 1.4** veranschaulicht. Abhängig von der Anzahl der Modellierungswerkzeuge unterscheidet man zwischen geschlossener und verteilter Modellbildung, wobei bei ersterer ein Modellierungswerkzeug und bei zweiterer mehrere Werkzeuge verwendet werden. Je nach Anwendungsgebiet werden für ein Gesamtsystem die Integratoren beziehungsweise die Solver von nur einem oder auch mehreren Tools verwendet. So unterscheidet man analog zur Anzahl der Modellierungswerkzeuge zwischen geschlossener und verteilter Simulation. Sehr häufig werden unter den vier möglichen Varianten die „klassische“ Simulation – Fall III – und die Co-Simulation - Fall II - verwendet.

1.3.2 Chancen und Herausforderungen der gekoppelten Simulation

Geimer definiert in [10] die Co-Simulation darüber, dass mindestens zwei Softwaretools miteinander gekoppelt werden und dabei mindestens zwei unterschiedliche Integratoren verwendet werden. Ein Gesamtsystem wird dann durch Kopplung der Teilsysteme simuliert. Hierbei werden die Teilmodelle aus ihren jeweiligen Simulationsumgebungen exportiert und auf einer gemeinsamen Plattform zu einem Gesamtmodell gekoppelt. Die Co-Simulation erfolgt in einer Master-Slave-Beziehung mit der Plattform als Master und den beteiligten Simulationsprogrammen als Slaves. Die Teilmodelle kommunizieren über entsprechende Schnittstellen mit der Plattform und liefern festgelegte Austauschgrößen zentral an den Master.

In Bezug auf eine „klassische“ Simulation birgt die gekoppelte Simulation als Modellbildungsvariante gewisse Herausforderungen. Eine Grundvoraussetzung für die Co-Simulation ist die Kompatibilität der Teilmodelle. Bereits mit der Wahl der Simulationsumgebung muss sichergestellt sein, dass sich alle Teilmodelle in der Co-Simulation untereinander oder zumindest mit der als Master identifizierten Umgebung verständigen können. Des Weiteren müssen die Teilmodelle in den zu übergebenden Ein- und Ausgangsgrößen aufeinander abgestimmt sein. Dies impliziert sowohl die Anzahl der Parameter und Variablen, als auch deren Übergabe in einem einheitlichen Format. Als Beispiel hierfür wäre die Einheit der Variablen oder der Faktor einer Einheit zur entsprechenden SI-Einheit zu nennen. Insbesondere beim Modellaustausch über Unternehmensgrenzen hinweg spielt dies eine wichtige Rolle. An dieser Stelle sei auch der Know-How-Schutz der Teilmodelle erwähnt.

Während des Simulationsprozesses selbst bedarf es insbesondere einer zeitlichen Koordination der beteiligten Softwareprogramme. Der Kommunikationsaufwand, sowie die in vielen Fällen zu verwendende feste Schrittweite bedingt als Konsequenz meist höhere Rechenzeiten, als dies bei Multi-Domain-Umgebungen der Fall ist.

Für den Engineering-Prozess jedoch bietet die Anwendung einer gekoppelten Simulation Vorzüge, welche nach Schweizer (siehe [11]) als Modellierungs-, Engineering- und Effizienzvorteil zusammengefasst werden. Diese sind wie folgt beschrieben:

- *Modellierungsvorteil:* Mit Hilfe einer Co-Simulation können unterschiedliche physikalische Disziplinen in dafür optimierten Simulationstools aufgebaut und bearbeitet werden. Die Spezialisierung der Programme gestattet es zudem, Modelle mit hohem Detaillierungsgrad aufzubauen.
- *Effizienzvorteil:* Jedes Teilmodell kann in der jeweiligen Modellierungsumgebung mit einer optimierten Schrittweite integriert werden. Dies ist insbesondere dann von Vorteil, wenn die Teilmodelle stark unterschiedliche Frequenzen aufweisen.
- *Engineeringvorteil:* Der Aufbau eines Gesamtsystems aus verschiedenen Teilmodellen bietet eine hervorragende Grundlage für die Arbeitsteilung über Abteilungs- oder gar Unternehmensgrenzen hinweg. Durch die parallele Modellierung der unterschiedlichen Subsysteme trägt die Co-Simulation im Sinne des Simultaneous Engineering, wie auch des Collaborative Engineerings seinen Teil zur Optimierung von Entwicklungsprozessen bei.

Abgesehen davon, dass für jedes Fachgebiet ein hierfür spezialisiertes Simulationsprogramm eingesetzt werden kann, ist es dem Anwender weiterhin möglich in seiner gewohnten Simulationsumgebung zu arbeiten. Das Einarbeiten in weitere Softwarepakete ist somit hinfällig.

Obwohl gewisse Herausforderungen beim Durchführen einer Co-Simulation bestehen, erscheint diese als Modellbildungsvariante viel versprechend. Insbesondere zeigen die unter dem Modellierungs- und Engineeringvorteil genannten Aspekte das Potenzial auf, die gekoppelte Simulation unternehmensübergreifend im Entwicklungsprozess einzubinden.

Die zunehmende Systemkomplexität stellt vor dem Hintergrund des internationalen Wettbewerbs und den an vielen Stellen fehlenden Fachkräften im Ingenieursbereich eine wachsende Herausforderung dar. Daher wird an vielen Stellen eine engere Verknüpfung der Kompetenzen verschiedener Unternehmen insbesondere im Bereich der KMU als erfolgreicher Ansatz gesehen [12]. Die unter dem Modellierungs- und Engineeringvorteil genannten Aspekte zeigen hierbei auf, dass die Co-Simulation großes Potenzial als Werkzeug für die unternehmensübergreifende Zusammenarbeit im Produktentstehungsprozess bietet. Damit ist es prinzipiell möglich, Simulationsmodelle unternehmensübergreifend auszutauschen und das Wissen von Zulieferer und Hersteller in frühen Phasen der Produktentwicklung gezielt zusammen zu führen. Auf diese Weise besteht für die Unternehmen insbesondere die Möglichkeit, der zunehmenden Dynamisierung der Produktlebenszyklen gerecht zu werden und sich nachhaltig einen Innovationsvorteil aufzubauen. Die Zulieferer stärken gleichzeitig ihre Position gegenüber dem Systemlieferanten. Diese haben dann die Möglichkeit, frühzeitig auf die anwendungsspezifischen Wünsche einzugehen und können so ihre Produkte besser auf die Kundenwünsche abstimmen.

2. Das Projekt GUSMA

Mit dem Hintergrund, dass sich die unternehmensübergreifende Co-Simulation als Modellierungsvariante im Bereich mobiler Arbeitsmaschinen bisher noch nicht durchgesetzt hat, wurde am Lehrstuhl für Mobile Arbeitsmaschinen (Mobima) des Karlsruher Instituts für Technologie (KIT) das Verbundprojekt GUSMA initiiert. Ein zentrales Ziel war es, diese für die genannte Branche nutzbar zu machen. GUSMA steht für „Gekoppelte Unternehmensübergreifende Simulation Mobiler Arbeitsmaschinen zur Virtualisierung der Produktentstehung“.

2.1 Vorstellung und Kennzahlen

Das Projekt wurde vom Bundesministerium für Bildung und Forschung (BMBF) im Rahmenkonzept „Forschung für die Produktion von morgen“ gefördert und vom Projektträger Karlsruhe (PTKA) im Karlsruher Institut für Technologie (KIT), Campus Nord, betreut. Das Projekt begann am 01.08.2008 und endet am 31.12.2011.

2.2 Ziele und Durchführung

Im Projekt GUSMA sollte eine standardisierte Vorgehensweise für die gekoppelte Simulation erarbeitet werden, um deren Anwendung zu vereinfachen und somit eine breitere Nutzung z.B. auf dem Gebiet der mobilen Arbeitsmaschinen zu ermöglichen.

Zentraler Ansatzpunkt der Projektidee war der Gedanke einer Plattform, auf der die gekoppelte Simulation durchgeführt werden sollte. Das zu simulierende System sollte dabei in Teilmodelle unterschiedlicher Fachdisziplinen bzw. in Teilmodelle unterschiedlicher Hersteller / Zulieferer aufgeteilt werden. Die einzelnen Teilmodelle wurden in fachgebietsspezifischen Simulationsprogrammen erzeugt und in einem, zur Plattform kompatiblen Format exportiert. Durch den Export auf die Plattform und die damit verbundene Umwandlung des Teilmodells wurde der Know-how-Schutz realisiert. Neben dem geschützten Teilmodell sollte auch der passende Solver mitexportiert werden, so dass jedes Teilmodell autark lauffähig war.

Auf der Plattform wurde für den jeweiligen Anwender die Möglichkeit geschaffen, über eine grafische Benutzeroberfläche durch Vernetzung der einzelnen Teilmodelle das Gesamtmodell aufzubauen. Zusätzlich wurden eine Parameterverwaltung sowie eine Bestimmung der Startwerte für die Simulation realisiert. Eine Möglichkeit zum Speichern der Datensätze rundete die Anforderungsliste an die Plattform ab. Auf diese Weise wurde eine Plattform zur intuitiven und effizienten Durchführung einer gekoppelten Simulation implementiert und somit zur Nutzbarmachung der Co-Simulation im Bereich mobiler Arbeitsmaschinen beitragen.

Aus Sicht der Industrie soll der Einsatz der gekoppelten Simulation anhand der standardisierten Vorgehensweise dem Fahrzeughersteller den Aufbau eines virtuellen Prototypen erleichtern, da er validierte Teilmodelle der Zulieferer einsetzen und somit deren Expertenwissen nutzen kann, ohne dass deren Know-how abfließt. Dadurch kann er ein Teil seiner Entwicklungs- und Testarbeit in der Simulation erledigen.

Die Zulieferer wiederum stärken durch die Lieferung von Hardware und Simulationsmodell ihre Wettbewerbsposition gegenüber anderen Marktbegleitern. Darüber hinaus sind sie selbst in der Lage, ihre Komponenten am virtuellen Prototyp zu testen und sparen ihrerseits Entwicklungszeit und –kosten. Das offene Konzept der Plattform soll des Weiteren auch ermöglichen, dass ein Fahrzeughersteller unabhängig von der Simulationsumgebung des Zulie-

ferers das eingebrachte Teilmodell nutzen kann und umgekehrt. Auf diese Weise soll letztlich auch die Anwendung von Simulationsmethoden im Entwicklungsprozess für KMUs in effektiver Weise nutzbar gemacht werden.

Die Durchführung des Projekts erfolgte in einem Zusammenschluss aus

- einem Forschungsinstitut, dem Lehrstuhl für Mobile Arbeitsmaschinen, KIT,
- einem OEM, der AGCO GmbH/Fendt,
- einem Zulieferer, der HYDAC SYSTEM GmbH,
- und mehreren Software-Unternehmen, der Fluidon GmbH, LMS Deutschland GmbH und der SIMPACK AG.

Die Kooperation der genannten Projektteilnehmer und der ständige Austausch trugen zur durchgängigen Validierung der Praxistauglichkeit der erzielten Forschungsfortschritte bei. Durch die Einbindung der erwähnten Software-Unternehmen wurde darüber hinaus die Umsetzbarkeit der standardisierten Vorgehensweise geprüft. Gleichzeitig wurde die Integration der notwendigen Software-Funktionalitäten sichergestellt.

Die standardisierte Vorgehensweise für die gekoppelte Simulation wurde am Beispiel einer hydropneumatischen Vorderachsfederung eines Traktors validiert. Dies fand sowohl virtuell auf der entwickelten Plattform, als auch durch praktische Versuche am Prüfstand statt. Weiterhin wurde im Rahmen des Projekts eine Richtlinie für den Modelldetaillierungsgrad erstellt.

3. Anforderungsanalyse

Die gekoppelte Simulation wird bisher in der Industrie nur vereinzelt durchgeführt. Die Ursachen dafür können an fehlenden Lizenzen liegen, an der Tatsache, dass bisher fachgebietspezifisch nur mit einer Simulationssoftware Teil- und nicht Gesamtsysteme simuliert werden oder dass dem Anwender das Know-how auf dem Gebiet der gekoppelten Simulation fehlt

Der Fahrzeughersteller bindet zur Simulation gekoppelter Systeme andere Fachgebiete als Subsystem, z.B. über eine in Fortran geschriebene Subroutine, in das eigene Simulationsmodell ein. Diese, für zwei Fachbereiche gangbare, Methode stößt an Grenzen, wenn mehrere Fachdisziplinen miteinander kombiniert werden müssen.

Auf Seiten des Zulieferers wird die Simulation gekoppelter Systeme durchgeführt, indem Komponenten aus anderen Fachdisziplinen aus entsprechenden Programmbibliotheken des Softwaretools verwendet werden.

Die Softwarehersteller haben ihre Produkte den Wünschen der Kunden angepasst. Es bestehen Möglichkeiten der Co-Simulation, des Gleichungs- und Codeimports/-exports, der Einbindung von Subroutinen oder der Abbildung und Einbindung anderer Fachdisziplinen aus speziellen Programmbibliotheken. Die Programme unterschiedlicher Hersteller sind dabei hinsichtlich ihrer Schnittstellenmöglichkeiten vergleichbar.

Im Bereich der Co-Simulation wird heute mit einer Master-Slave-Beziehung gearbeitet. Für die Co-Simulation selbst stehen zwei unterschiedliche Strategien bzw. Arten zur Verfügung, die Simulation mit zwei parallel geöffneten Programmen und die Simulation im eingebetteten Modus.

Für die Co-Simulation, bei dem zwei Programme parallel rechnen, müssen beide Programme zur Verfügung stehen. Das bedeutet, dass entweder beide Programme auf einem Rechner installiert oder über eine Netzwerkverbindung miteinander verbunden sein müssen. In beiden Fällen benötigt der Anwender für jedes an der Co-Simulation beteiligte Programm eine Lizenz.

Demgegenüber gibt es die Möglichkeit der Co-Simulation im eingebetteten Modus. Technisch gesehen handelt es sich beim eingebetteten Modus immer noch um eine Co-Simulation. Bei der Erstellung des Teilmodells zur Simulation im eingebetteten Modus wird der Gleichungslöser mit exportiert. Aufgrund dessen besitzt das Teilmodell die volle Programmfunktionalität, was die numerische Berechnung des Systems betrifft. Allerdings muss zur Simulation das Programm, mit dem das Teilmodell erstellt wurde, nicht geöffnet bzw. nicht einmal auf dem entsprechenden System installiert sein. Diese Art der Co-Simulation eröffnet die Möglichkeit, mehrere Modelle einer Fachdisziplin in einem Gesamtsimulationsmodell einzubauen, da jedes Teilmodell autark rechnet. Darüber hinaus reduziert diese Co-Simulationsmöglichkeit Lizenzkosten, da zur Simulation des Systems lediglich die Lizenz des Masterprogramms vorhanden sein muss.

Somit unterstützt die Co-Simulation im eingebetteten Modus den Gedanken der unternehmensübergreifenden Simulation. Mit ihr wäre es beispielsweise einem Fahrzeughersteller möglich, ein Gesamtsystem zu simulieren, bei dem Komponenten einer Fachdisziplin von verschiedenen Herstellern stammen. Er kann mit der Systemreaktion der Teilmodelle rechnen und benötigt für die einzelnen Programme keine Lizenz, was die Kosten deutlich senkt. Die Tatsache, dass bei dieser Art der Co-Simulation keine Veränderungen an den exportierten Teilmodellen vorge-

nommen werden kann, ist zudem eine Voraussetzung für Maßnahmen des Know-How-Schutzes. Für Veränderungen werden das Originalmodell sowie die entsprechende Software benötigt.

Auf Basis dieser Erkenntnis wurde eine Anforderungsliste für die zu erstellende Methodik generiert. Die in diesem Zusammenhang durchgeführte Anforderungsanalyse bestand aus dem Erfassen des Standes der Technik im Bereich Co-Simulationsplattformen sowie aktueller Forschungsvorhaben. In Zusammenarbeit mit dem Projektarbeitskreis – insbesondere den Software-Herstellern – wurde der Ist-Zustand vorhandener Co-simulationsschnittstellen fixiert. Darauf aufbauend wurde eine Umfrage über den Einsatz von Simulationswerkzeugen bei den Stiftungsfirmen des MOBIMA e.V. durchgeführt. Der Förderverein MOBIMA e.V. unterstützt den Lehrstuhl für Mobile Arbeitsmaschinen fachlich. Ihm gehören folgende namhafte Industrieunternehmen der Branche an: AGCO, Argo Hytos, Bosch Rexroth, Bucher Hydraulics, Claas, Liebherr, Daimler Chrysler, Hawe Hydraulics, Hydac System GmbH, Linde und ZF. Unter Berücksichtigung der Zielbedingung, dass die Gesamtsystemsimulation plattformbasiert (siehe Abschnitt 4.1) ablaufen und die Umsetzung möglichst einfach applizierbar sein sollte, wurden die Software-Schnittstellen der Projektpartner unter dem wirtschaftlichen Aspekt der höchsten Kundennachfrage untersucht. Hierbei fiel insbesondere die Schnittstelle zu Mathworks' Matlab/Simulink ins Gewicht, welche aufgrund der hohen Verbreitung von Matlab/Simulink in der Industrie auf hohe Nachfrage trifft.

3.1 Schnittstellenanalyse bei Softwareherstellern

Im ersten Schritt wurden bestehende Softwareschnittstellen der Softwarepartner im GUSMA Arbeitskreis analysiert. Diese werden im Anhang 9.1 umfassend beschrieben. Wie dort zu erkennen ist, verfügen alle drei Simulationswerkzeuge bereits Co-Simulationsschnittstellen zu Matlab/Simulink. Darüber hinaus bestätigt eine Recherche bei weiteren Simulationswerkzeugen, dass ein Großteil ebenfalls Co-Simulationsschnittstellen zu Matlab/Simulink anbieten. Auch aus einer im Rahmen des Projekts durchgeführten Umfrage des Lehrstuhls bezüglich der Verwendung von Softwaretools in Firmen kann das Fazit gezogen werden, dass die Software Matlab/Simulink eine weite Verbreitung hat. Für den weiteren Verlauf im Projekt erscheint die Übernahme von Matlab/Simulink als Grundlage für eine Plattform als viel versprechend. Die Einschränkung aufgrund einer Abhängigkeit zu Matlab wird dabei von den Vorzügen der weiten Verbreitung von Matlab in der Industrie und den bereits von vielen anderen Softwarewerkzeugen bereit gestellten Schnittstellen zu Matlab überwogen. An den bisher dargestellten Schnittstellen ist hierbei sehr gut zu erkennen, dass die individuelle Gestaltung der Interfaces zu Matlab jedoch stark abweichen kann. Für die Nutzbarmachung und Durchsetzung der Co-Simulation als Werkzeug für die unternehmensübergreifende Produktentwicklung, ist eine für den Nutzer einfache und effiziente Anwendung von Teilmodellen in der gekoppelten Simulation erforderlich. Das Mittel der Standardisierung sollte im Projekt GUSMA daher ein zentrales Element werden.

3.2 Anforderungen an eine zentrale Plattform

Auf Basis der vorangegangenen Analyse bestehender Softwareschnittstellen für die Co-Simulation wurde ersichtlich, dass für die unternehmensübergreifende Gesamtsystemsimulation eine zentrale Softwareplattform erforderlich sei. Dies, um auf der einen Seite das Ziel der einfachen Anwendbarkeit für den Nutzer zu verfolgen und auf der anderen Seite, um die wirtschaftliche Umsetzbarkeit für die beteiligten und zukünftigen Softwarepartner zu gewährleisten.

Eine Plattform sollte in diesem Zusammenhang die Möglichkeit bieten, Modelle aus unterschiedlichen Fachdisziplinen aufzunehmen sowie eine einfache und für alle Programme gleiche Handhabung sicherzustellen. Die Software zur Erstellung der Teilmodelle ist beliebig. Lediglich das Exportformat muss zur Plattform kompatibel sein. Um aus den Teilmodellen ein Gesamtsimulationsmodell zu generieren, müssen die Teilmodelle auf der Plattform miteinander vernetzt werden. Eine Definition und eine einheitliche Form für die Austauschgrößen sind erforderlich. Hierfür sollte zunächst eine Nomenklatur eingeführt werden, die die einzelnen Größen in der Simulation beschreibt. Für die Simulation wurden folgende Begriffe identifiziert:

- Zustandsvariablen: zeitabhängige, während der Simulation veränderliche Größen; Die Zustandsgrößen stellen die Input- und Outputgrößen der Teilmodelle dar. Zustandsgrößen zum Zeitpunkt $t = 0$ werden Initialisierungsvariable genannt.

- **Parameter:** zeitunabhängige, während der Simulation unveränderliche Größen; Mit Parametern werden beispielsweise Geometrien beschrieben. Die Parameter lassen sich in drei Untergruppen unterteilen:
 - *Geschützte Parameter:* Sie können nur vom Modellersteller gelesen und verändert werden. Mit ihrer Hilfe soll das Know-how geschützt werden.
 - *Änderbare Parameter:* Der Modellersteller kann Parameter definieren, die der Anwender ändern darf (z.B. Blendengröße). Dadurch sollen Optimierungsrechnungen beim Anwender ermöglicht werden.
 - *Gemeinsame Parameter:* Bei der Verwendung mehrerer Teilmodelle besteht die Möglichkeit, dass ein Parameter von verschiedenen Teilmodellen verwendet werden muss (z.B. Kolbendurchmesser).
- **Prozessgrößen:** Visualisierte Größen aus den Teilmodellen, die nicht für die Berechnung verwendet werden.

Jedes Teilmodell in der Simulation besitzt Zustandsgrößen, die so genannten Zustandsvariablen. Diese Zustandsvariablen ergeben sich aus der mathematischen Beschreibung der jeweiligen Fachdisziplin und beschreiben die Änderung der Zustände des Modells während des Simulationsprozesses. Sie bilden die Ein- und Ausgabegrößen der jeweiligen Modelle. Mit Hilfe der Zustandsvariablen können einzelne Teilmodelle miteinander verknüpft werden. Die Verknüpfung der Teilmodelle soll grafisch erfolgen und durch den Anwender durchgeführt werden. Eine Plattform muss daher die vorhandenen Zustandsvariablen visualisieren können. **Abbildung 3.1** zeigt eine Struktur, mit der im Bereich der gekoppelten Simulation gearbeitet wird am Beispiel eines fiktiven Systems, das aus einem mechanischen, zwei hydraulischen und zwei regelungstechnischen Teilmodellen besteht [15].

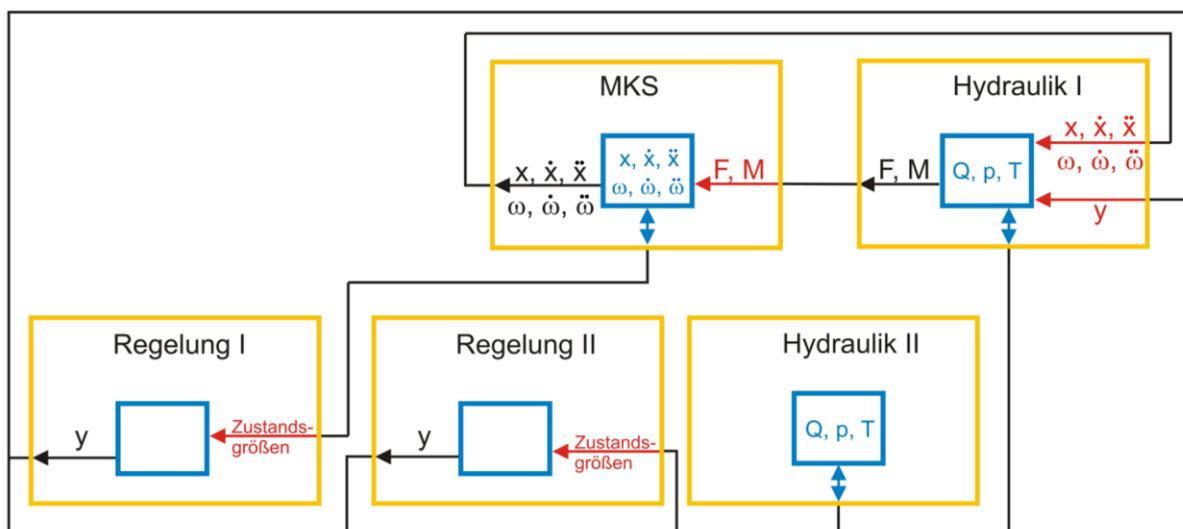


Abbildung 3.1 - Struktur bei der gekoppelten Simulation (siehe [15])

Die Grundlage dieser Struktur sind die Material-, Energie- und Informationsflüsse zwischen den Teilmodellen. In der Struktur ist deutlich die Möglichkeit erkennbar, sowohl Teilmodelle aus unterschiedlichen wie auch aus gleichen Fachdisziplinen miteinander zu koppeln. Die Austauschgrößen lassen sich somit noch in interne (innerhalb einer Fachdisziplin, blaue Pfeile) und externe (fachdisziplinübergreifend, schwarze Pfeile) unterscheiden.

Die Zustandsvariablen der einzelnen Teilmodelle sowie die zugehörigen Parameter sollen in je einer separaten Datei abgelegt werden, sobald ein Teilmodell auf die Plattform gebracht wird. Alle Zustandsgrößen und Parameter werden dadurch gesammelt und stehen auf einen Blick zur Verfügung.

Mit Hilfe der Zustandsvariablendatei können in einem weiteren Schritt die Initialisierungsvariablen bestimmt werden, sofern dies vom Anwender gewünscht wird. Hierfür soll die Plattform eine Möglichkeit bieten, die Startwerte der Zustandsvariablen zu berechnen. Die Berechnung soll anhand von Hilfsfunktionen durchgeführt werden. Mit Hilfe der dadurch erzeugten Initialisierungsvariablen soll der Einschwingvorgang bzw. die Einschwingzeit in der Simulation reduziert werden. Daneben besteht weiterhin die Möglichkeit, die Initialisierungsvariablen durch die Simulation selbst zu erzeugen, indem das Gesamtsystem aus den Teilmodellen zusammengebaut und ohne weitere Initialisierung die Simulation gestartet wird. Dies kann allerdings zu verfälschten Startwerten führen.

Die Parameterdatei soll bei der Identifikation gemeinsamer Parameter aus verschiedenen Teilmodellen helfen. Die Datei mit allen Parametern soll vom Anwender auf gleiche Parameter überprüft werden. Gleichen Parametern soll der gleiche Wert zugewiesen werden und sie sollen den gleichen Namen erhalten. Eine Umbenennung anderer Parameter soll ebenfalls ermöglicht werden. Hierfür müssen die bestehenden Parameterdateien der Teilmodelle durch den neu erzeugten Datensatz ergänzt bzw. überschrieben werden.

Die erzeugten Initialisierungsvariablen sowie der erzeugte Parametersatz werden im Initialisierungsprozess der einzelnen Teilmodelle vor dem Start der Simulation den einzelnen Teilmodellen zugewiesen. Die Erzeugung der Initialisierungsvariablen, der Parameterdatei sowie die Erstellung des Gesamtsimulationsmodells durch Vernetzung der einzelnen Teilmodelle wird als Pre-Processing bezeichnet. Dieser Pre-Process ist beispielhaft in **Abbildung 3.2** dargestellt.

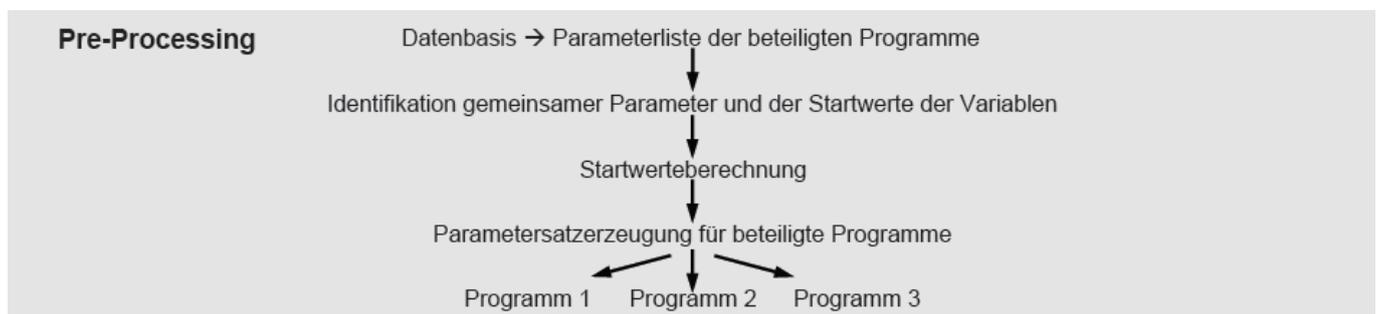


Abbildung 3.2 - Beispielhafter Initialisierungsprozess

Abbildung 3.3 zeigt noch einmal zusammenfassend die Anforderungen an die Plattform und den Ablauf der Simulation. Nachdem die Teilmodelle auf die Plattform gebracht und miteinander vernetzt wurden, wird der erläuterte Pre-Process durchlaufen. Im anschließenden Initialisierungsprozess werden die neuen Daten auf die Teilmodelle übertragen. Anschließend kann die Simulation gestartet werden. Die Simulationsdaten werden fortlaufend in eine

Datei geschrieben und können auch visualisiert werden. Mit dem Simulationsende werden die Daten an einem zentralen Speicherort gespeichert und stehen anschließend für den Post-Prozess zur Verfügung.

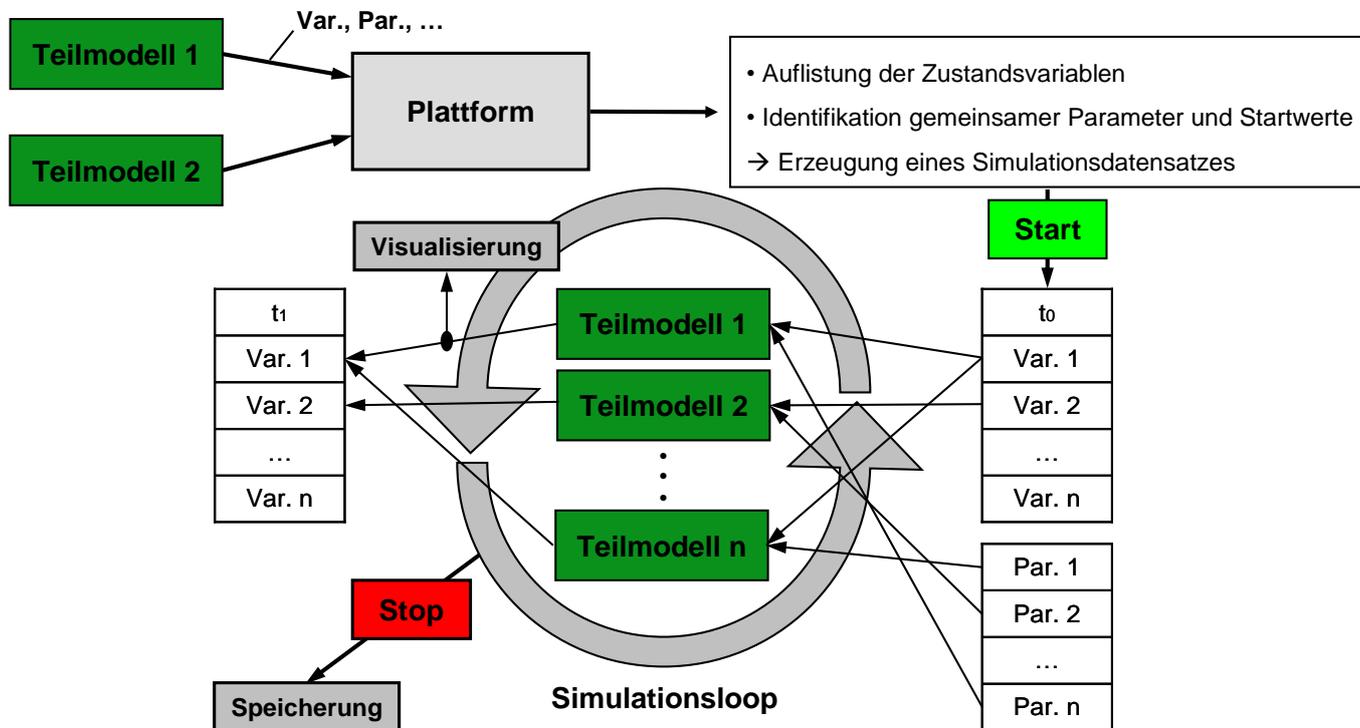


Abbildung 3.3 - Möglichkeit einer standardisierten Co-Simulation

4. GUSMA-Plattform - Die drei Säulen der Standardisierung

Um die Co-Simulation als Werkzeug für die unternehmensübergreifende Zusammenarbeit im Produktentstehungsprozess einsetzen zu können, stand als zentrales Element die Standardisierung im Fokus. Die Standardisierung beruhte im Projekt auf drei Säulen – einem Plattformgedanke, einer Standard-Datenschnittstelle, sowie eine Standard-Vorgehensweise. Weiterhin wurde ein Schwerpunkt auf dem Schutz des Know-Hows beim Modellaustausch gelegt. Im Folgenden werden die genannten Elemente näher erläutert.

4.1 Der Plattformgedanke

Ausgangspunkt im Projekt war der Gedanke einer Plattform, auf welcher die gekoppelte Simulation durchgeführt werden kann (siehe **Abbildung 4.1**). Das zu simulierende System sollte dabei in Teilmodelle unterschiedlicher Fachdisziplinen bzw. in Teilmodelle unterschiedlicher Hersteller / Zulieferer aufgeteilt werden. Die einzelnen Teilmodelle können in fachgebietsspezifischen Simulationsprogrammen erzeugt und in einem zur Plattform kompatiblen Format exportiert werden. Durch den Export auf die Plattform und die damit verbundene Umwandlung des Teilmodells wurde der Know-how-Schutz realisiert. Entsprechend der festgelegten Eingangs- und Ausgangsgrößen werden die Teilmodell miteinander verbunden. Idealerweise sollte neben dem geschützten Teilmodell auch der passende Solver mitexportiert werden, so dass jedes Teilmodell autark lauffähig ist.

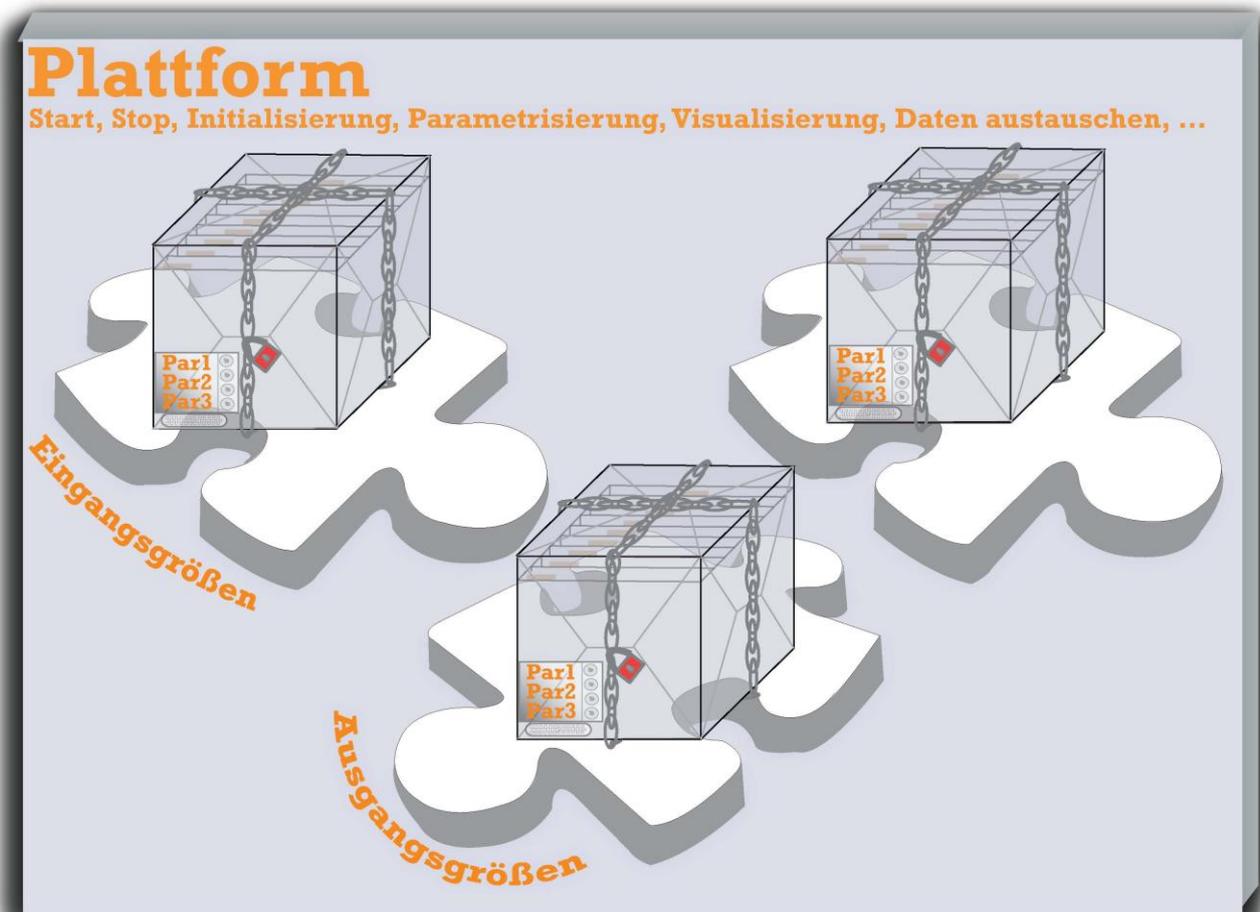


Abbildung 4.1 – Plattformgedanke für die unternehmensübergreifende Zusammenarbeit

Die Teilmodelle sollen daher auf der Plattform lauffähig sein, ohne dass weitere Lizenzen der Erstellungsprogramme erforderlich werden. Auf dieser Basis soll die Gesamtsystemsimulation mit den erforderlichen Funktionalitäten für den Anwender nutzbar sein. Dem entsprechend soll die Plattform die erforderlichen Funktionen für das erfolgreiche Ausführen im PreProcess, der Simulation und im PostProcess bieten. Dies schließt u.a. Funktionen für Start, Stop, Initialisierung, Parametrisierung freigegebener Daten, Visualisierung, Datenaustausch u.v.m. ein.

Als Basis für die Plattform wurde MATLAB/Simulink der Firma Mathworks gewählt. Die weite Verbreitung dieses Softwarepakets in der Industrie und die Tatsache, dass zahlreiche kommerzielle Simulationsprogramme bereits eine Schnittstelle zu MATLAB/Simulink anbieten (siehe Kapitel 3.1), trugen wesentlich zu deren Auswahl bei. Weiterhin wird der Vorteil der Co-Simulation zu Nutzen gemacht, dass der Anwender in gewohnten Softwareumgebungen arbeiten und für die Gesamtsystemsimulation keine neue Software erlernen muss. Ebenso können die Möglichkeiten der Visualisierung von Matlab/Simulink mitgenutzt werden. Dies soll die einfache Anwendbarkeit der GUSMA Plattform garantieren und eine Gelegenheit besonders für kleine und mittlere Unternehmen darstellen, die Co-Simulation unternehmensübergreifend im Entwicklungsprozess einsetzen zu können.

4.2 Standardinterface

Um den Datenaustausch standardisiert zu ermöglichen, wurden Anforderungen für ein Standarddateninterface formuliert und in drei Prozessschritte beim Aufbau des Co-Simulationsmodells eingeteilt. Ein Simulationswerkzeug muss diese Anforderungen erfüllen, um gemäß dem GUSMA-Standard eine Co-Simulation durchführen zu können.

A) Beim Modellaufbau im Simulationswerkzeug:

■ *Festlegen von geschützten Parametern.*

Nach Aufbau des Teilmodells im Simulationswerkzeug ist dieser Schritt das wesentliche Kernelement und die Voraussetzung, um den Know-How-Schutz bei der unternehmensübergreifenden Modellübergabe zu gewährleisten.

■ *Export nach Matlab/Simulink inklusive Solver.*

Ziel ist es hierbei, dass das Teilmodell beim Kunden autark lauffähig ist und für diesen keine weiteren Lizenzgebühren anfallen.

■ *Empfehlung A1: Export der Modell-Dokumentation aus Simulationswerkzeug.*

Viele Simulationstools bieten die Möglichkeit, direkt in der Software ein Teilmodell zu beschreiben oder dieses auf andere Arten zu dokumentieren. Als nicht feste Bedingung und Empfehlung für den GUSMA-Standard ist es, diese Beschreibung des Teilmodells zu exportieren.

B) Beim Modellimport / Öffnen des exportierten Modells in Simulink:

■ *Daten in einheitlichem Format in den Matlab Workspace laden.*

Für den Datenaustausch wurde der Workspace von Matlab als Datenspeicher festgelegt (**Abbildung 4.2**). Als zentrales Element in Matlab gewährleistet dies insbesondere eine Versions-unabhängige sowie nahezu wartungsfreie Lösung. Alle für eine Simulation freigegebenen Daten – Parameter und Initialisierungsvariablen – werden beim Import eines Teilmodells in den Workspace geschrieben bzw. von dort ausgelesen. Im

Workspace werden die genannten Austauschdaten nach einem Standardformat als Structure eingeschrieben. In der Structure werden die Informationen Value, Unit, Factor und Marker übergeben. Diese enthalten neben dem Namen die Information über den Wert der Größe, die Einheit, den Faktor zur entsprechenden SI-Einheit und einen so genannten Marker. Letzterer legt fest, ob es sich um einen Parameter oder eine Initialisierungsvariable handelt und kann entsprechend die Werte „p“ oder „iv“ einnehmen. Abbildung 15 zeigt dies exemplarisch für die Größe „Durchmesser_Kolben3_AME“. Hierbei handelt es sich um einen Parameter (marker = 'p') mit dem Wert „13“. Dieser wird in der Einheit Millimeter ('mm') übergeben, welche den Faktor 0,001 zur SI-Einheit – in diesem Fall „Meter“ – hat.

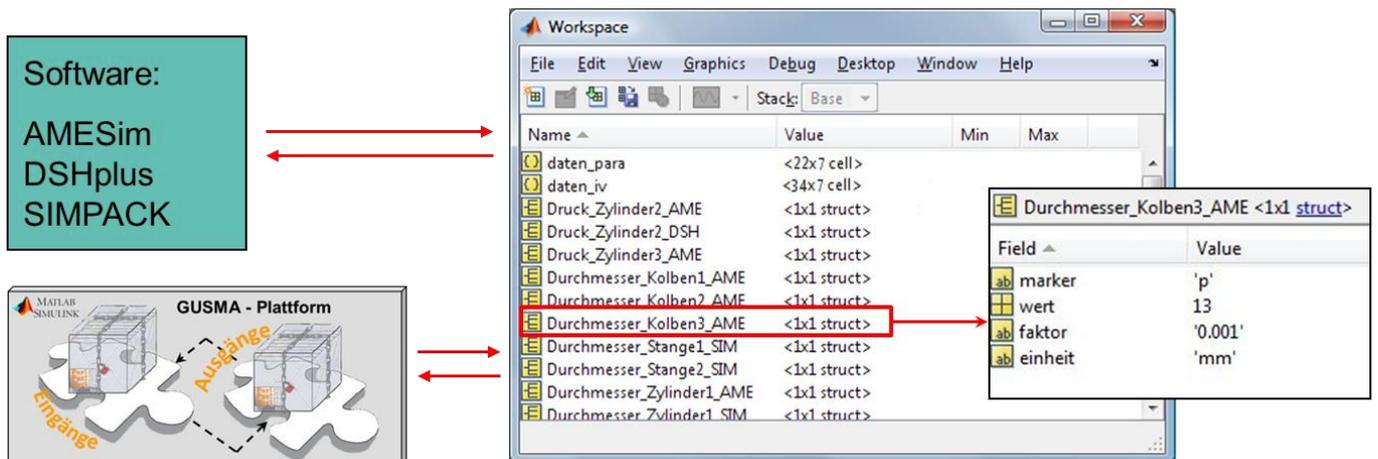


Abbildung 4.2 – Definition des Dateninterface nach GUSMA-Standard

- Automatisches Laden der Daten in Matlab Workspace.**
 Die freigegebenen Daten sollen beim Import des Teilmodells auf die GUSMA-Plattform automatisch in den Matlab Workspace geschrieben werden. Dies kann mit Hilfe sogenannter Callback-Funktionen realisiert werden.
- Anzeigen der In- und Outputvariablen am Modellblock in Simulink.**
 Die Zustandsgrößen am Simulink-Block eines Teilmodells sollen für den Nutzer direkt sichtbar sein. Durch das Anzeigen der Anzahl und der jew. Namen der Größen wird das Verbinden der unterschiedlichen Teilmodelle erheblich erleichtert.
- Abrufen des Parameters für das Kommunikationsintervall „t_c“.**
 Der Zeitpunkt zum Austausch der Zustandsgrößen wird vom Simulationsmaster, in diesem Fall Simulink, vorgegeben und als Kommunikationsintervall bezeichnet. Falls das Teilmodell die Vorgabe eines Kommunikationsintervalls erfordert, soll dieses mit dem Wert „t_c“ aus dem Workspace abgerufen werden.
- Empfehlung B1: Abrufbarkeit der Modellbeschreibung gewährleisten.**
 Entsprechend der Empfehlung A1 soll die exportierte Modellbeschreibung über den „Help“-Button der Blockmaske in Simulink abrufbar sein. Neben der textuellen Beschreibung ist eine Abbildung des Teilmodells aus der grafischen Erstellungsfläche wünschenswert.

■ *Empfehlung B2: Automatische Benennung der Daten im Workspace.*

Die GUSMA-Plattform unterstützt den Benutzer beim Gesamtsystemaufbau insbesondere durch Programmfunktionen für das Parameterhandling. In Hinblick auf ein einheitliches Verständnis und eine Wiederauffindbarkeit von Parametern und Initialisierungsvariablen wird eine einheitliche, automatisch generierte Nomenklatur der Daten im Workspace empfohlen. Vorgeschlagen wird.

[Teilmodellname]_[Komponentenname]_[Parametername]

Hierbei ist noch zu beachten, dass der Unterstrich als Trennzeichen zwischen den Namen, hier „_“, in der Anzahl der Unterstriche variiert werden kann. Im Anschluss daran folgt durch Punktationszeichen getrennt („.“) die Angabe des oben genannten Standardformat (Marker, Value, Unit, SI-Factor).

C) Während der Co-Simulation

Als Austauschdatenspeicher wurde der Matlab Workspace festgelegt. Daher muss das Ein- und Auslesen aller Daten der an der Co-Simulation beteiligten Modelle über den Workspace erfolgen.

■ *Daten von [Software] in Matlab Workspace schreiben*

Wichtig ist hierbei vor allem, dass vor dem ersten Integrationszeitschritt bzw. bei der Initialisierung der Co-Simulation die Parameter/Variablen direkt aus dem Workspace ausgelesen werden müssen. Ansonsten kann es zu ungewünschten Einschwingvorgängen im Gesamtsystem kommen.

■ *Daten von Workspace in [Software] zurückschreiben.*

Die gerade erwähnten Anforderungen an die Co-Simulationsschnittstelle wurde von den im Projekt GUSMA beteiligten Softwarefirmen beispielhaft umgesetzt. Die Ergebnisse sind im Anhang 9.2 dargestellt.

4.3 Standardisierte Vorgehensweise

Um den Aufbau einer Co-Simulation auf der GUSMA-Plattform effizient zu gestalten, wurde eine standardisierte Vorgehensweise in einer grafischen Benutzeroberfläche (auch: Graphical User Interface - GUI) implementiert. Diese sieht sechs Schritte vor, die vom Benutzer ausgeführt werden müssen, um von den exportierten Teilmodellen zu einer funktionsfähigen Co-Simulation zu gelangen (siehe auch [16]). **Abbildung 4.3** zeigt dieses GUI.

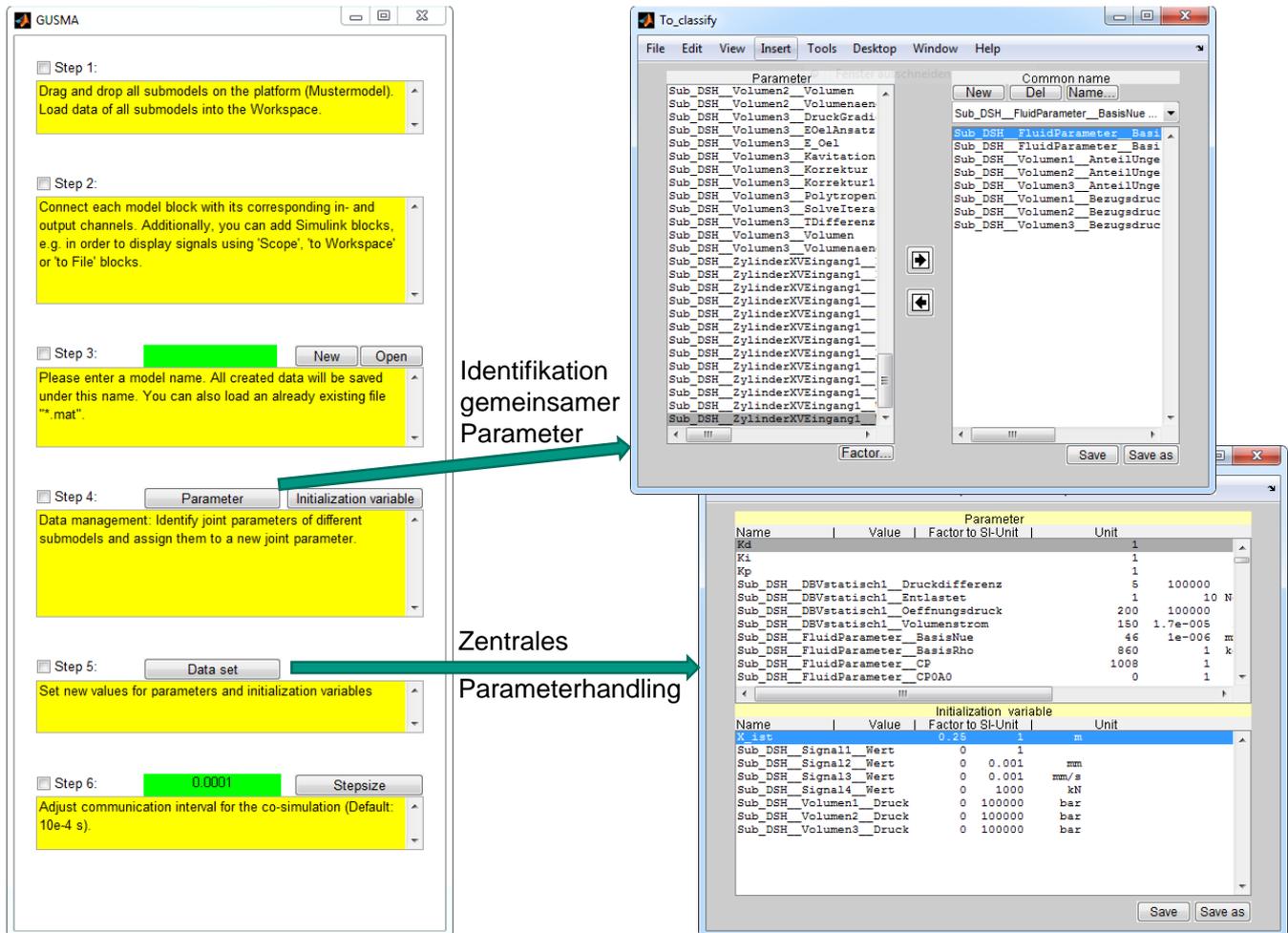


Abbildung 4.3 – Grafische Benutzeroberfläche der GUSMA-Plattform

Schritt 1: Ziehen Sie alle Teilmodelle auf die Plattform (Mustermodell). Laden Sie die Daten aller Teilmodelle in den Workspace!

In diesem Schritt werden alle Teilmodelle auf die Plattform gezogen. Die Parameter der Teilmodelle werden automatisch im Hintergrund in den Matlab-Workspace eingelesen. Der Entwickler eines Teilmodells hat beim Modelaufbau schon die Möglichkeit festzulegen, welche Parameter auf der Plattform sichtbar oder unsichtbar sind. Dadurch kann das Know-how des Modellerstellers geschützt werden.

Schritt 2: Verbinden Sie die einzelnen Modellblöcke mit den entsprechenden Ein- und Ausgangskanälen. Sie können zusätzlich Simulink-Blöcke hinzufügen, um beispielsweise relevante Größen zu beobachten, ``Scope``, ``to Workspace``, ``to File``.

Jede ausgetauschte Zustandsgröße wird im Modellierungs-GUI als separater Kanal angezeigt, inklusive des vorgegebenen Variablennamen. Blöcke, wie z.B. ``Mux`` oder ``Demux`` sind daher nicht mehr nötig. Jede ausgetauschte Zustandsgröße wird somit übersichtlich und klar dargestellt. Der Anwender soll in diesem Schritt die Teilmodelle im Simulink-Fenster entsprechend ihrer Ein- und Ausgangsgröße miteinander verbinden. Die Funktionsblöcke aus der Simulink-Bibliothek, wie beispielsweise ``Scope`` oder ``to Workspace``, können hinzugefügt werden, um die Simulationsergebnisse zu visualisieren oder zu speichern. Ein wichtiger Punkt für den Anwender ist, dass das Abspeichern der in diesem Schritt im System hinterlegten Parameter im nachfolgenden Schritt 3 stattfindet.

Schritt 3: Geben Sie einen Modellnamen ein. Alle entstehenden Daten werden unter diesem Namen gespeichert. Oder wählen Sie eine schon existierende Datei ``*.mat`` aus.

Die Daten im Workspace können in diesem Schritt gesichert werden. Ebenso hat der Benutzer an dieser Stelle die Möglichkeit, die schon gespeicherten Daten in den Workspace zu laden. Diese Daten im Workspace können anschließend auf dem programmierten GUI weiter bearbeitet werden. Somit können auf einfache Weise Parameterstudien durchgeführt und virtuelle Prototypen mit unterschiedlichen Konfigurationen bedatet werden.

Schritt 4: Datenverwaltung: Sie haben die Möglichkeit, gemeinsame Parameter aus unterschiedlichen Teilmodellen zu identifizieren und diese einem gemeinsamen neuen Parameter zuzuordnen.

Dieser Schritt ist das zentrale Element des GUI. Die Buttons ``Parameter`` und ``Initialisierungsvariable`` dienen dazu, alle im Workspace befindlichen Daten aufzulisten und gemeinsame Parameter oder Initialisierungsvariablen einzuteilen. ``Parameter`` sind Werte, die während der Simulation nicht geändert werden, wie beispielsweise eine Zylinderlänge. ``Initialisierungsvariable`` steht für die Zustandsgrößen, welche zu Beginn einer Simulation als Startwert festgelegt werden. Dies kann beispielsweise eine Anfangsposition oder ein Vorspanndruck in einem hydraulischen System sein, welcher an mehreren hydraulischen Knoten anliegt. Sobald der Benutzer einen dieser Buttons anwählt, taucht jeweils ein Sub-GUI auf. Auf der rechten Seite in **Abbildung 4.3** ist das Sub-GUI für die Parameterverwaltung zu sehen. Die klassifizierten Parameter werden im linken Fenster auf dem Sub-GUI angezeigt. Zum Identifizieren gemeinsamer Parameter kann der Benutzer durch Doppelklicken auf einen Parameter im linken Fenster oder Klicken auf die Pfeile in der Mitte die zutreffenden Parameter in das rechte Fenster schieben und diese somit einem gemeinsamen Parameter zuordnen. Genauso kann die Zuordnung eines Parameters rückgängig gemacht werden. Das Sub-GUI für die Initialisierungsvariablen ist in der Handhabung identisch.

Schritt 5: Sie haben die Möglichkeit, neue Werte für die Parameter und Initialisierungsvariablen zu vergeben.

Im Schritt fünf besteht die Möglichkeit, Werte für änderbare und gemeinsame Parameter, sowie Initialisierungsvariablen festzulegen oder zu ändern. Dieser Vorgang wird durch das GUI, wie in **Abbildung 4.3** rechts unten angezeigt, abgeschlossen.

Schritt 6: Stellen Sie das Kommunikationsintervall für die Simulation ein (Standardwert: $10e-4$ s).

In diesem Schritt kann der Benutzer eine Variable namens ``t_c`` mit Hilfe des GUIs ändern. Die Variable ``t_c`` wird im Workspace von Matlab als Kommunikationsintervall eingerichtet. Dieses beschreibt bei einer Co-Simulation die Dauer zwischen zwei Zeitpunkten, in welchen die Teilmodelle miteinander kommunizieren und Daten austauschen. Wenn eine an der Co-Simulation beteiligte Software das Kommunikationsintervall während des Simulationslaufs benutzen muss, kann sie die ``t_c`` direkt verwenden.

Nach der Durchführung der beschriebenen Schritte kann die Simulation mit dem Start-Knopf der Simulink-Plattform ausgeführt werden.

5. Auswahl eines geeigneten Modelldetaillierungsgrads

Mit Hilfe der Simulation ist es möglich, dynamische Prozesse eines zu untersuchenden Systems abzubilden. Kosten- und Zeitersparnis stehen als Gründe für den Einsatz von Simulationswerkzeugen in der Industrie an erster Stelle. Diese Ersparnis ergründet sich in den Tatsachen, dass mit intensivem Aufwand erstellte Prototypen oder Versuchseinrichtungen nicht in vollem Umfang oder aber mit einem bereits höheren Wissensstand über das zu entwickelnde Produkt aufgebaut werden können. In sehr sicherheitskritischen Branchen, wie z.B. der Luftfahrtindustrie, werden Produkte bereits heute zu einem Großteil allein mit Hilfe virtueller Prototypen bis zur Serienreife geführt.

Jedoch ist zu beachten, dass reale Prozesse aus wirtschaftlichen Gesichtspunkten nicht beliebig detailliert in der Simulation nachgebildet werden können. Denn trotz steigender Rechenleistung können Simulationen mehrere Tage andauern und sowohl Infrastruktur als auch Personal für längere Zeit in Anspruch nehmen. Auch gilt die Vorstellung in Frage zu stellen, dass ein beliebig detailliertes Simulationsmodell in beliebig genauen Resultaten resultiert. Denn hierbei spielen auch Faktoren der Parametrierbarkeit eine wesentliche Rolle.

Um also auch in diesem Bereich die Wirtschaftlichkeit des Entwicklungsprozesses zu gewährleisten, müssen Rechenzeiten klein gehalten werden. Daher gilt es in der Simulation stets einen Kompromiss zwischen Modelldetaillierungsgrad und Ergebnisgüte der Modelle einzugehen.

Im Zusammenhang mit der unternehmensübergreifenden Produktentwicklung wurde im Rahmen des Projekts GUSMA daher auch die Auswahl eines geeigneten Modelldetaillierungsgrads untersucht. Ein wesentlicher Bestandteil in der gekoppelten Simulation stellt hierbei insbesondere die Festlegung einer Grenzfrequenz in der Simulation dar. Die Qualität der Simulationsergebnisse einer gekoppelten Simulation lässt sich nach heutigem Wissensstand an der Wahl des Kommunikationsintervalls festmachen. Dies ist speziell bei der Kopplung unterschiedlich steifer Systeme, z.B. Mechanik und Hydraulik, wichtig.

Im Rahmen des Projekts GUSMA untersuchte Völker in [13] das Kommunikationsintervall bei der gekoppelten Simulation. Dabei beschreibt er die Grundlagen der Co-Simulation im Allgemeinen. Im Speziellen wurden Co-Simulationen mit MATLAB/Simulink als Master untersucht. Dies entspricht ebenso der Konstellation im Projekt GUSMA. Die Kopplung bei der Co-Simulation erfolgt durch Austausch von Zustandsgrößen der einzelnen Teilm Modelle. Der Zeitpunkt zum Austausch der Zustandsgrößen wird vom Simulationsmaster vorgegeben und als Kommunikationsintervall bezeichnet. Entgegen anderer Programme ist eine automatische Anpassung des Kommunikationsintervalls unter Simulink nur in Sonderfällen möglich. Aus diesem Grund wurde für eine gekoppelte Simulation unter MATLAB/Simulink eine Methodik zur Bestimmung des Kommunikationsintervalls erarbeitet. Dieses wird wie folgt ermittelt

Aus der höchsten im System ermittelten Frequenz bzw. kleinsten Zeitkonstante T_{Klein} wird zunächst ein Zeitschritt durch Kehrwertbildung ermittelt. Ein Zehntel dieses Kehrwerts ergibt dann das für die gekoppelte Simulation zu verwendende Kommunikationsintervall h .

$$h = \frac{T_{\text{Klein}}}{10}$$

Aus Sicht der Co-Simulation muss der Detaillierungsgrad der Modelle so eingestellt werden, dass bei einem vorgegebenen Kommunikationsintervall konsistente, konvergente und stabile Gesamtsystemsimulation durchgeführt werden kann. Diese werden nachfolgend in Kürze erläutert (siehe [13]).

Konsistenz:

Die Grundlage der näherungsweise Lösung der Differentialgleichung ist, dass die Approximation der Differentialgleichung und der Anfangswerte Lösungen der zugehörigen Gleichungen liefern, die wiederum eine Approximation der gesuchten Lösung der Differentialgleichung ist. Der Begriff der Konsistenz präzisiert dabei die Approximation der Anfangswertaufgabe.

Zur Festlegung der Konsistenz muss der Fehler betrachtet werden, der bei der Diskretisierung entsteht, wie auch die Fehlerfortpflanzung.

Konvergenz:

Diese liegt bei Einschrittverfahren vor, wenn das Verfahren konsistent zur Anfangswertaufgabe und Lipschitz-stetig ist. Die Lipschitzstetigkeit gewährleistet darüber hinaus die Stabilität des Einschrittverfahrens.

5.1 Vorgehen für Auswahl eines geeigneten Modelldetaillierungsgrad

Im Sinne des unternehmensübergreifenden Aufbaus eines Gesamtsystems in einer Co-Simulation ist der Detaillierungsgrad abhängig von den Frequenzen, die in der Co-Simulation abgebildet werden sollen. Dementsprechend gilt es in einem ersten Schritt festzulegen, welche Frequenzen dargestellt werden sollen und davon den Detaillierungsgrad für das Gesamtmodell abzuleiten.

Hierbei müssen alle theoretisch auftretenden Frequenzen mit einbezogen werden. Diese können auftreten in Form von

- Anregungsfrequenzen,
- Eigenfrequenzen oder
- Taktzeiten von Teilmodellen bzw. Abtastfrequenzen von Regelungselementen.

Nach der Festlegung der maximal auftretenden Frequenzen und damit des Modelldetaillierungsgrads für das Gesamtsystem, kann das Maß der Detaillierung für die Teilmodelle abgeleitet werden.

6. Durchführung einer gekoppelten unternehmensübergreifenden Simulation am Beispiel einer hydropneumatischen Traktorvorderachsfederung mit Niveauregulierung.

6.1 Funktionsweise der hydropneumatischen Traktorvorderachsfederung mit Niveauregulierung

Für das Projekt GUSMA wurde das niveauregulierte Vorderachs-Federungssystem der Fendt Baureihe 700Vario als Beispielobjekt gewählt. Das System ist hydropneumatisch ausgeführt - die Federungswirkung wird durch Membranspeicher erzielt, die verdrängtes Ölvolumen der Federzylinder aufnehmen bzw. abgeben. Durch die Niveauregulierung wird dieses passive hydraulisch-mechanische System zu einem mechatronischem.

Ein Lagesensor vergleicht die mittlere Ist-Position mit der Vorgabe. Bei Abweichung wird durch Anpassung des Drucks im Federungssystem auf die Soll-Position geregelt. Unabhängig von der Achslast bleibt der zur Verfügung stehende Federweg gleich und damit der Federungskomfort auf hohem Niveau. Durch Eingriff des Fahrers via Betätigung der Folientaster am Fahrerplatz (siehe **Abbildung 6.1**) lässt sich die Vorderachse auch manuell anheben oder absenken, was etwa beim Anbau von Anbaugeräten vorteilhaft ist. Auch ein Sperren der Vorderachsfederung ist möglich. Hierbei wird die Achse in den Einfederanschlag verfahren und dort gehalten. Benötigt wird eine gesperrte Federung beispielsweise für ein exakte Tiefenführung von Anbaugeräten. In allen Zuständen des Federungssystems kann die Achse um die Fahrzeuglängsachse frei pendeln.



Abbildung 6.1 - Folientaster zur Betätigung der Vorderachsfederung

Die Achse der Baureihe 700Vario ist als gefederte und angetriebene Starrachse ausgeführt. Sie ist mittels einer Längsschwinge (2 in **Abbildung 6.2**) mit dem Rahmen des Fahrzeugchassis (1) verbunden. Über Gummi-Metall-Lagerbuchsen kann die Achse eine Drehbewegung um die Fahrzeugquerachse ausführen (Federbewegung der Achse), aber auch kleine Radial- und Axialverschiebungen sind im Rahmen der Lagersteifigkeit möglich. Zwischen Längsschwinge und Fahrzeugrahmen sind zwei Hydraulikzylinder (4) angebracht. Der Öldruck auf deren Kolbenflächen stützt die vertikalen Radlasten ab. Der maximale Federweg beträgt $\pm 50\text{mm}$ aus der Neutrallage. Die Maxi-

malpositionen werden durch Anschläge in den Zylindern (beim Ausfedern) und zwischen Schwinge und Rahmen (Einfedern) festgelegt. Der Metallanschlag beim Einfedern ist durch Elastomerpuffer gedämpft.

Der Achskörper (3) selbst ist in der Schwinge mit zwei Gleitlagern gelagert und kann um die Fahrzeuglängsachse pendeln. Anschlagflächen an Achskörper und Schwinge begrenzen den Pendelwinkel auf ca. 8°.

Der Antrieb der Achse erfolgt über eine zentrale Kardanwelle vom Getriebe. Das Antriebsmoment wird über ein sperrbares Differential auf die beiden Außenplaneten in der Radnabe verteilt. Die Lenkung der Achse mit Lenkeinschlägen bis zu 50° erfolgt hydraulisch. Sowohl Antrieb als auch Lenkung sind für das Projekt GUSMA von untergeordneter Bedeutung.

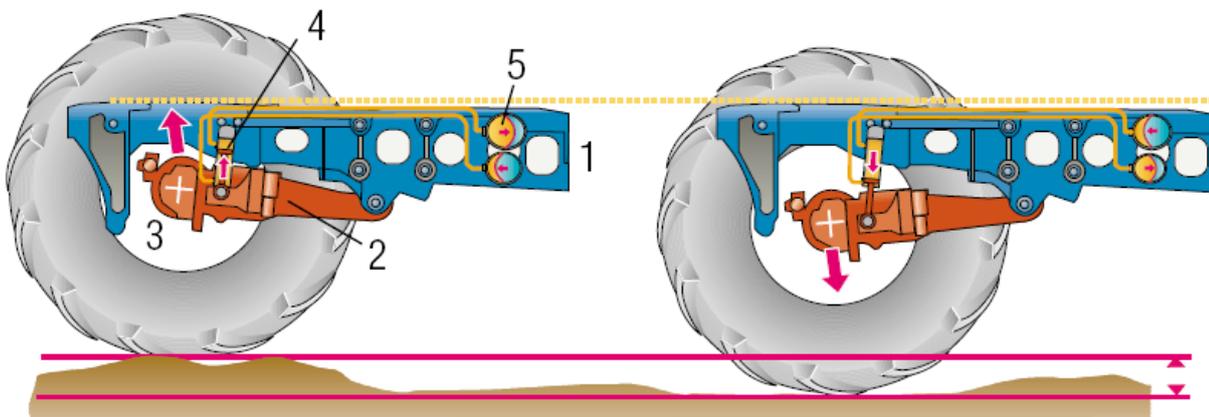


Abbildung 6.2 - Komponenten und schematische Funktionsweise der Vorderachsfederung der Baureihe 700Vario

Die Achslast wird wie bereits erwähnt von zwei Hydraulikzylindern aufgenommen. Sowohl Ring- als auch Kolbenflächen beider Zylinder sind hydraulisch verbunden. Beim Einfedern (links in **Abbildung 6.2**) bewegt sich der Achskörper nach oben und schiebt über die Kolben Volumen aus dem Zylinder in zwei Stickstoff-Membranspeicher (5). Gleichzeitig wird Ölvolumen aus einem dritten, ringseitig angeordnetem, Speicher in die Zylinder geholt. Beim Ausfedern (rechts) sind die Volumenströme entsprechend umgekehrt. Regel- und Steuereingriffe von Fahrzeug-Steuergerät und Fahrer, in dieses, bis jetzt als passiv beschriebenes, System erfolgen über ein magnetbetätigtes 4/2-Wege-Schieberventile und ein 3-Wege-Proportional-Druckregelventil. Die notwendige Druckversorgung erfolgt über die regelbare Kolbenpumpe des Fahrzeugs, die etwa auch Kraftheber und die Ventile zur Versorgung externer Verbraucher versorgt und 200bar liefert. Ventile und Speicher sitzen an einem Ventilblock auf der rechten Seite des Fahrzeugrahmens und sind über Rohr- und Schlauchleitungen mit den Zylindern verbunden. Die Hydraulikpumpe ist am Getriebegehäuse angeflanscht.

Bei einem Eingriff der Niveauregulierung, aber auch beim manuellen Heben oder Senken durch den Fahrer, öffnet das 4/2-Wege-Ventil auf der Kolbenseite entweder zur Druckversorgung oder zum Tank. Das Druckregelventil regelt gleichzeitig auf der Ringseite den Druck auf 100bar. Eine Ausnahme ist der Zustand „Federung gesperrt“. Hier werden die Zylinderringflächen kurzzeitig mit 200bar beaufschlagt, um ein sicheres Erreichen der Endposition zu gewährleisten.

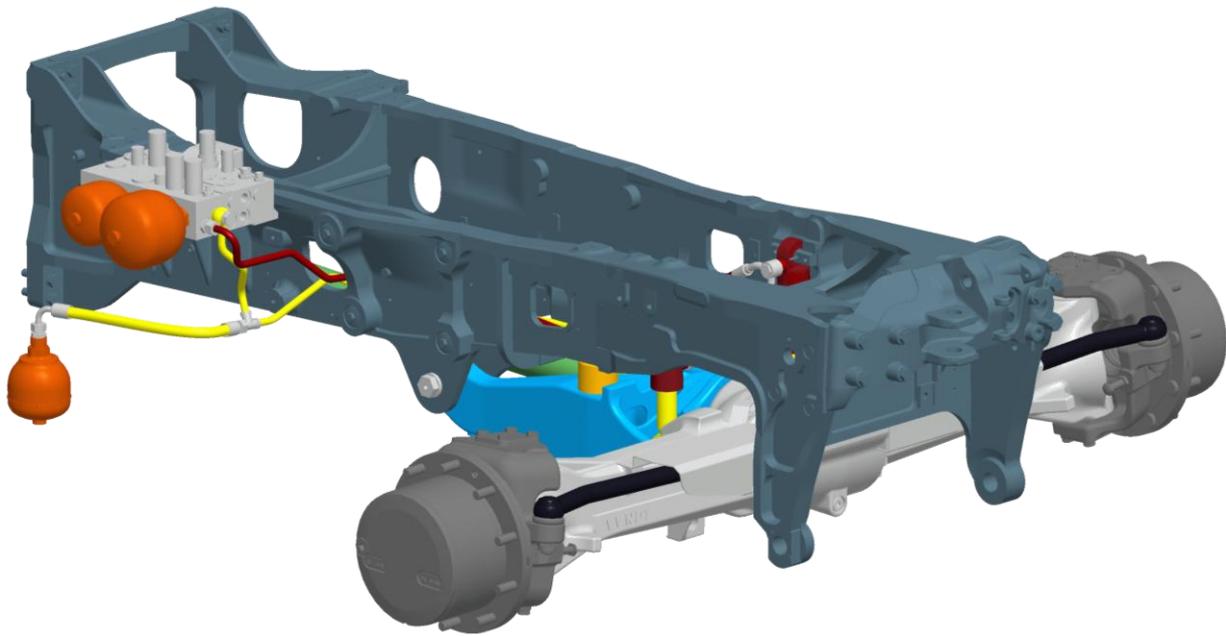


Abbildung 6.3 - Vorderachse mit Fahrzeugrahmen und Federung der Baureihe 700Vario; nicht dargestellt ist die regelbare Kolbenpumpe

6.2 Aufteilung der Modellierung entsprechend Hersteller – Zulieferer

Für die Simulation wird die hydropneumatische Vorderachsfederung entsprechend der realen Aufteilung von Zulieferer und Fahrzeughersteller und somit auch anhand der beteiligten Ingenieursfachdisziplinen in Teilmodelle zerlegt. **Abbildung 6.4** zeigt das Vorgehen im Projekt GUSMA für den standardisierten unternehmensübergreifenden Modellierungsprozess.

Es existieren ein Steuerungs-, ein Mechanik- und zwei Hydraulikmodelle. Zwei Hydraulikmodelle entstehen, da die hydraulische Druckversorgung und die hydropneumatische Federung von unterschiedlichen Zulieferern stammen. In Anlehnung an den in Abbildung 1.3 gezeigten Modellierungsprozess erstellt die Firma AGCO das Mechanik-, sowie das Steuerungsmodell. Dies erfolgt in den jeweiligen Fachabteilungen mit den Programmen SIMPACK und Simulink. Die Firma HYDAC liefert das in AMESim erstellte Hydraulikmodell für den Federungskreis und die Firma Fluidon als fiktiver Zulieferer in diesem Fall das Hydraulikmodell für die Druckversorgung.

Jedes Unternehmen kann hierbei den Detaillierungsgrad entsprechend der Anforderungen an das Gesamtsystem anpassen und die zu übergebenden beziehungsweise die zu schützenden Parameter und Variablen auswählen. Die computergerechte Darstellung läuft Software-spezifisch innerhalb jedes Modellierungswerkzeuges ab.

Anschließend wird ein Modellexport nach Matlab/Simulink nach dem oben beschriebenen GUSMA-Standard durchgeführt. Auf diese Weise werden funktionsfähige und parametrisierte Teilmodelle erzeugt, die direkt per Drag & Drop auf der GUSMA-Plattform eingefügt werden können. Im Anschluss werden am Mobima - in der Rolle als Fahrzeughersteller - gemäß dem oben beschriebenen Vorgehen die Teilmodelle zum Gesamtsimulationsmodell der Vorderachsfederung zusammen gebaut.

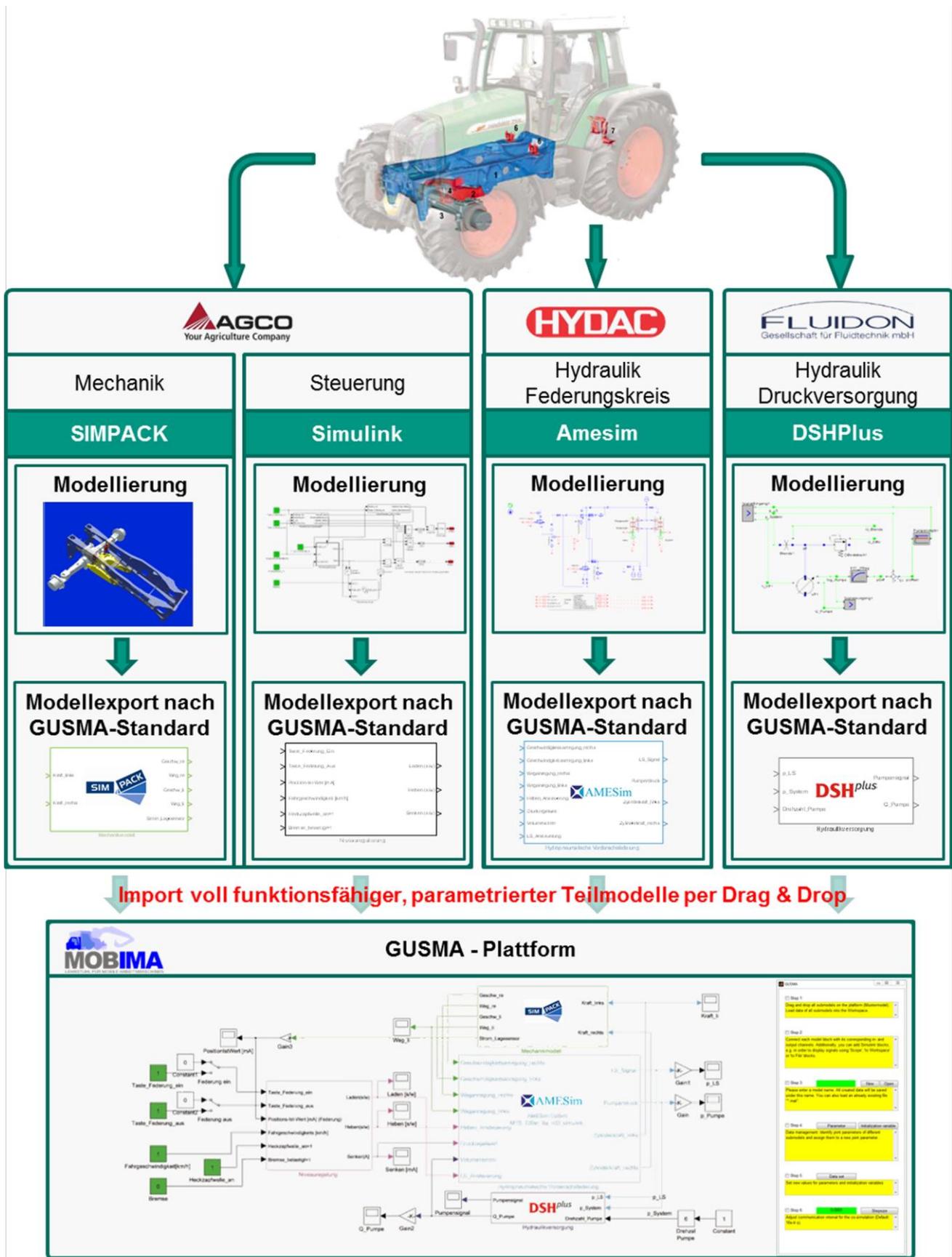


Abbildung 6.4 - Vorgehen bei standardisiertem unternehmensübergreifendem Modellierungsprozess

6.3 Beschreibung der Teilmodelle und Aufbau der Systemsimulation

Der Aufbau der Systemsimulation erfolgte entsprechend der in Kapitel 6.2 vorgestellten Arbeitsteilung zeitlich parallel. Für das Verständnis der Gesamtsystems simulation werden im Folgenden nun die Funktionsweise und der Aufbau der einzelnen Teilmodelle erläutert. Dabei soll weiterhin für die unterschiedlichen Modellierungsvorgehensweisen in den jeweiligen Simulationswerkzeugen sensibilisiert werden.

6.3.1 Hydropneumatischer Federungskreis: Das AMESim-Modell

Die hydropneumatische Federung wurde von der Fa. HYDAC in AMESim modelliert und hat zum einen die Aufgabe elektrische Signale in hydraulische Stellgrößen zu übersetzen, um somit zum einen die Niveauregulierung und zum anderen das Sperren der Federung zu realisieren. Zum anderen ist ein Leakage freies Absperrn der Federung gegenüber dem restlichen System ebenfalls Teil des Lastenheftes, so dass auch nach längerem Stillstand die Niveaulage der Federung beibehalten wird.

Das hydropneumatische Federungsmodell besteht im Wesentlichen aus den Elementen der Load Sensing-Steuerung (LS-Steuerung), dem Hebenventil, der Systemdruckbegrenzung, der Signalaufbereitung für den Druckregelwert, der Druckversorgung und der Schnittstelle zur GUSMA-Plattform via Simulink. Diese Elemente und deren Funktionsweise im AMESim werden nachfolgend beschrieben

LS-Steuerung

Die in **Abbildung 6.5** dargestellte Baugruppe besteht aus einem Schaltventil, das den Pumpendruck auf die LS-Leitung kurzschließt, um somit das Druckniveau der Regelpumpe auf deren Maximum anzuheben.

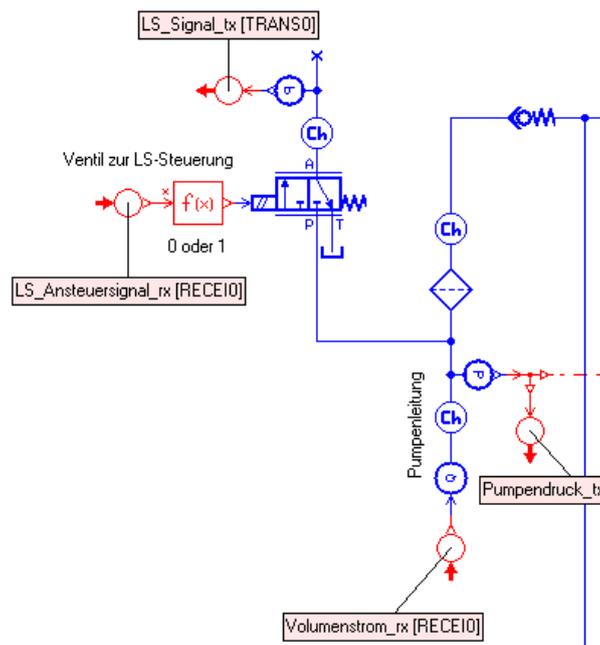
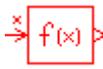


Abbildung 6.5 – LS-Steuerung

Wird nur das LS-Signal aktiviert, wird das Niveau über das entsperrbare Rückschlagventil wie in **Abbildung 6.10** dargestellt abgelassen.

Zum Anheben des Niveaus muss das Schaltventil, dargestellt im Detail B - **Abbildung 6.10** -, zusätzlich geschaltet werden. Die Niveaufahrt wird im Einzelnen später beschrieben.

Der Funktionsblock  stellt mit dem eingestellten Verstärkungsfaktor des Ventils sicher, dass ein digitales Signal [0...1] das Ventil vollständig öffnet bzw. schließt. Das Eingangssignal darf polarisiert sein, also [-1...0...1], weil bei dem eingehenden Signal das Vorzeichen entfernt wird. Die Schaltschwelle beträgt 0,5.

Durch die Aktivierung des LS-Ventils wird auch der Druckregler aktiviert. Der Druckregler hält den stangenseitigen Druck bei einer Niveaufahrt immer konstant. Somit ist der Druck der Stangenseite unabhängig von dem Befüllungszustand der Bodenseite. Dies ist ein sehr wirkungsvolles Mittel, um die Differenzierung der Drucksignale zu vermeiden.

Hebenventil

Das Hebenventil (**Abbildung 6.6**) ist nur wirksam, wenn gleichzeitig die LS-Ansteuerung erfolgt. Die Gründe sind zum einen, dass ein genügend hohes Druckpotential zur Verfügung stehen muss und zum anderen, wie bereits bei der LS-Ansteuerung beschrieben, der Druck der Stangenseite konstant gehalten werden muss.

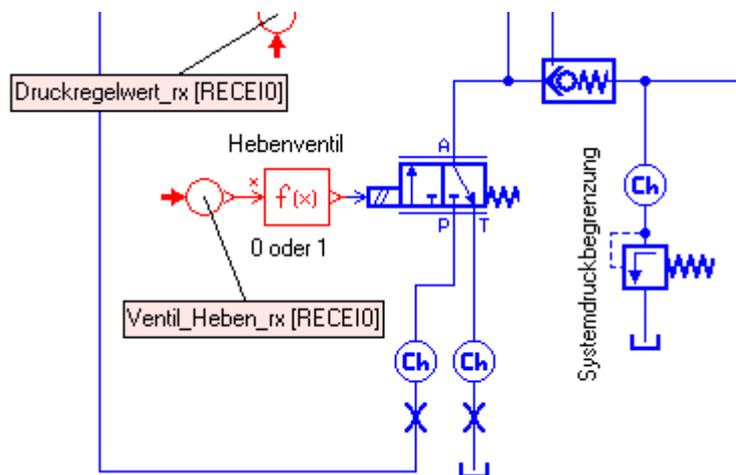


Abbildung 6.6 - Hebenventil

Systemdruckbegrenzung

Diese Druckbegrenzung (**Abbildung 6.6**) begrenzt nicht den Primärdruck des hydraulischen Systems, sondern den Systemdruck der Federung. Diese Maßnahme ist wirksam gegen mechanische Beschädigungen, die auftreten können, wenn die Federung in die Überlastung geht. Durch das vorgeschaltete Rückschlagventil ist die Federung von der Gesamtdruckbegrenzung des Systems abgeschnitten und fordert daher eine eigene Absicherung. Der Vorteil liegt darin, dass ein anderes Druckniveau wirksam werden darf.

Druckreglerwert – Signalaufbereitung

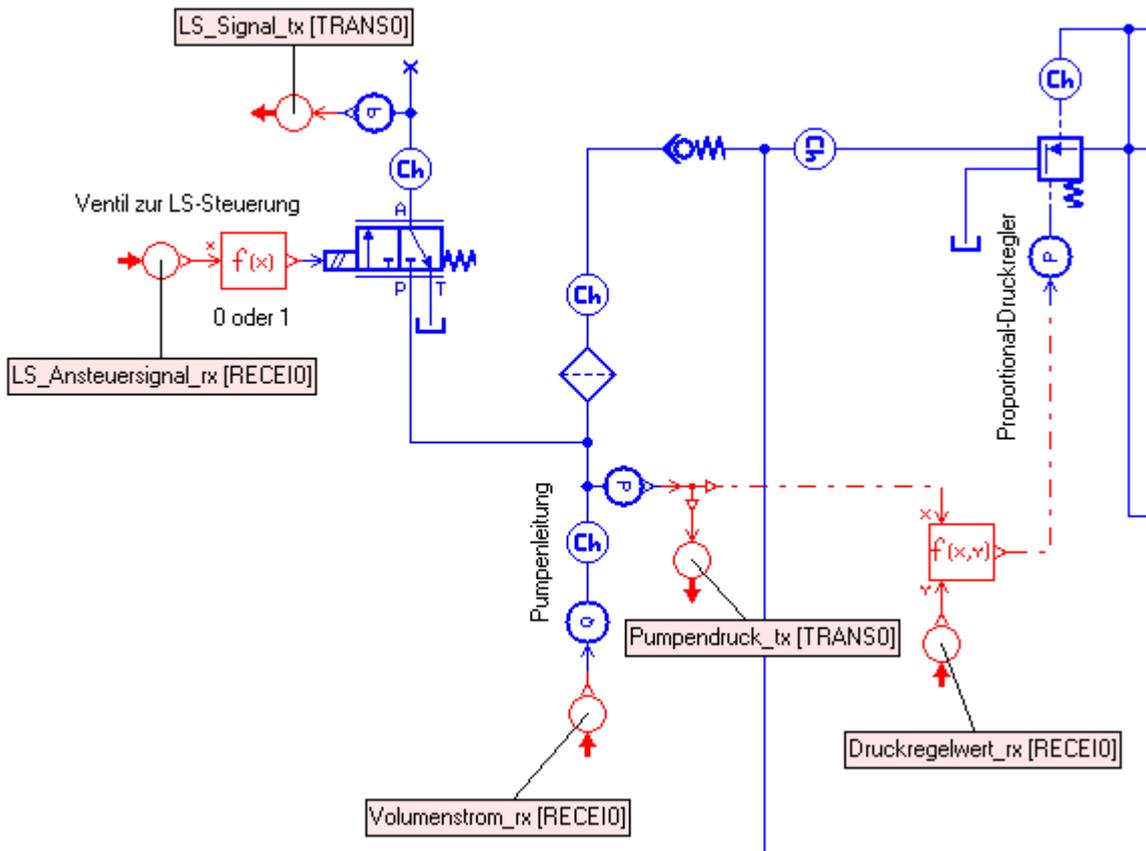
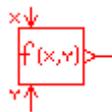


Abbildung 6.7 - Proportional-Druckreglerkreis

Die Problematik besteht darin, dass der Druckregelwert dem System nicht immer seinen Einstellwert zur Verfügung stellen kann. Daher ist es erforderlich, eine Fallentscheidung zu erzwingen. Fall 1 – Der Versorgungsdruck liegt niedriger als der Einstellwert – der Druckregler kann nur den Versorgungsdruck bereitstellen. Fall 2 – Der Versorgungsdruck liegt höher als der Einstellwert – der Druckregler nimmt seine Arbeit auf.



Diese Fallentscheidung wird im $f(x,y)$ -Block (**Abbildung 6.7**) getroffen. Ferner wird in demselben Block noch der äquivalente Vorspanndruck der Feder des Druckreglers berücksichtigt. Die Kennlinie des Druckreglers wurde mit Hilfe der zwei für das System interessanten Arbeitspunkte angenähert

Druckversorgung

Die Druckversorgung ist in **Abbildung 6.7** und **Abbildung 6.8** dargestellt. Es wird ein Volumenstrom angeliefert. In der nachgeschalteten Kapazität C_h wird der zugehörige Druck berechnet.

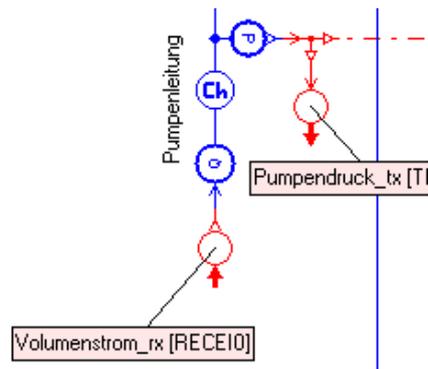


Abbildung 6.8 - Druckmittelversorgung

Übergabewerte an das Gesamtsystem

Über die Schnittstelle (**Abbildung 6.9**) wird aus der Simulink-Umgebung mit AMESim bidirektional kommuniziert. Auf der linken Seite befinden sich die Größen, die zu Simulink hin kommuniziert werden, an der rechten Seite sind alle Signale angetragen, die das AMESim-Modell empfängt. Sämtliche Größen, die kommuniziert werden, sind als SI-Einheiten geführt und werden im Anhangskapitel 9.3.1 beschrieben. Jedes Signal wird intern in AMESim zu den typischen hydraulischen Einheiten gewandelt.



Abbildung 6.9 - AMESim-Schnittstellenblock

■ Output:

- Das *LS-Signal* beschreibt die Höhe des in dem hydraulischen Modell vorherrschenden Druckes. Die interne Einheit ist das Bar, das als Pascal übermittelt wird.
- Der *Pumpendruck* beschreibt die Höhe des in dem hydraulischen Modell vorherrschenden Versorgungsdruckes. Dieser Druck entsteht aus einer volumenstromgetriebenen Versorgung. Die interne Einheit ist das Bar, das als Pascal übermittelt wird.
- Die *Zylinderkräfte* (links bzw. rechts) werden getrennt ausgegeben, weil das Modell entsprechend aufgebaut worden ist. Diese Konstellation begünstigt, dass mit diesem Modell auch ein hydraulisches Pendeln über die Zylinder verwirklicht werden könnte. Die interne Einheit ist das Newton, das auch als solches übermittelt wird.

- Der *kolbenseitige* und der *stangenseitige Druck* werden nur zur Rückmeldung an das System ausgegeben. Diese Drücke beschreiben den internen Zustand der Federung. Die interne Einheit ist das Bar, das als Pascal übermittelt wird.

■ Input:

Vorab: Binäre Signale werden als 0 und 1 dargestellt, wobei 0 bedeutet „AUS“ und 1 bedeutet „EIN“. In diesem Modell sind alle binären Signale mit einem Schwellenwert von 0,5 belegt. Zudem ist die Verarbeitung des eingehenden Signals vorzeichenunabhängig, das heißt, $-0,5 < x < 0,5$ wird als 0 gewertet und alle anderen Signale werden als 1 gewertet.

- Die *LS-Ansteuerung* verlangt ein binäres Signal und schaltet das LS-Signal auf das Druckversorgungssystem.
- Der *Volumenstrom* verlangt die Einheit [m³/s] und wird intern als [L/min] verarbeitet.
- Der *Druckregelwert* beschreibt den Strom, der nötig ist, um den Proportional-Druckregler auf seinen Sollwert einzustellen.
- Die *Heben-Ansteuerung* verlangt ein binäres Signal und schaltet das Ventil zum Ausfahren der Federungszyylinder.
- Die *Wegeanregung* verlangt nach Meter.
- Die *Geschwindigkeitsanregung* verlangt nach [m/s].

Sowohl die Weg- als auch die Geschwindigkeitsanregung für dieses Modell ist, aus Gründen des hydraulischen Pendelns, ebenfalls für links und rechts getrennt ausgeführt.

Gesamtschaltbild des Hydraulikschemas

Das Schema zeigt symbolhaft den Aufbau des Systems. Für das Niveauregelsystem sind vorgefertigte Modelle verwendet worden, da die damit erreichbaren Ergebnisse ausreichend genau sind.

Für das Federungssystem wurden Individualmodelle verwendet. Einerseits steht die Kontrolle des Systems im Vordergrund, andererseits ermöglicht dies eine detaillierte Modellierung der Reibung der Federungszyylinder.

Beim Leitungssystem wurde absichtlich sehr stark vereinfacht, weil die Berechnung der Leitungsdynamik sehr zeitintensiv ist. Diese Vereinfachung bewirkt allerdings, dass induzierte Druckwellen, ebenso wie die Viskositätsänderung, nicht berücksichtigt werden können. Sollte sich auf dem Prüfstand erweisen, dass an dieser Stelle doch nicht zu vernachlässigende Einflüsse auftauchen, muss das Modell dementsprechend nachgebessert werden!

Die Kräfte der Hydraulikzylinder werden mittels spezieller Sensoren gemessen und zurückgemeldet. Diese sollen die reinen Kraftkomponenten liefern. Würden die herrschenden Differenzdrücke benutzt, so wären die Verlust erzeugenden Kräfte mit enthalten und könnten nicht sauber getrennt aufgeführt werden.

Die beiden Rückschlagventile in den Details C und D der **Abbildung 6.10** sind nicht Teil des in Realität ausgeführten hydraulischen Blocks. In der Simulation hat sich gezeigt, dass bei der Initialisierung durch unzulässig hohe Unterdrücke die Berechnung zur Singularität neigt. In der Simulation funktionieren diese beiden Rückschlagventile als Nachsaugventile. Sobald der Druck im System positiv wird, sind die Nachsaugventile ohnehin geschlossen und spielen somit keine Rolle für die weitere Simulierung.

Auf diese Weise können die drei zentralen Funktionen des hydropneumatischen Federungssystem erfüllt werden. Diese sind „Niveau absenken“, „Niveau anheben“ und „Federung sperren“ und werden im Folgenden kurz beschrieben.

Niveau absenken

Zur Niveauabsenkung muss lediglich das LS-Ventil geschaltet werden. Der Standby-Druck wird dann direkt und nicht lastproportional in die LS-Kette gemeldet. Die Schwenkpumpe fördert somit das Fördermedium unter maximalem Druck in das Niveauregelsystem, wodurch das entsperbare Rückschlagventil im Detail A der **Abbildung 6.10** geöffnet wird. Gleichzeitig hält der Druckregler (Detail F) den Druck auf der Ringseite konstant. Das überschüssige Öl der Kolbenseite fließt über das geöffnete Rückschlagventil zum Tank hin ab.

Das Fahrzeug senkt sich.

Niveau anheben

Wird zusätzlich zum LS-Ventil auch das 3/2-Wegeventil (Detail B) geschaltet, so wird auch das Rückschlagventil im Detail E geöffnet. Somit kann das, durch das Anheben auf der Stangenseite, überschüssige Öl über den Druckregler (Detail F) zum Tank hin abfließen. Auch hier hält der Druckregler den Druck auf der Stangenseite konstant.

Das Fahrzeug hebt sich.

Federung sperren

Um die Federung zu sperren, wird zunächst das Niveau wie soeben beschrieben abgesenkt. Ist die Federung einmal bis auf Block abgesenkt, so kann durch Erhöhung des Druckregelwertes sichergestellt werden, dass die Federungszyylinder sicher in ihrer eingezogenen Endanschlagstellung gehalten werden.

Das Fahrzeug ist komplett abgesenkt und die Federung ist durch die eingeschlossene Ölmenge der Stangenseite gesperrt. Ein Ausfedern kann nicht stattfinden.

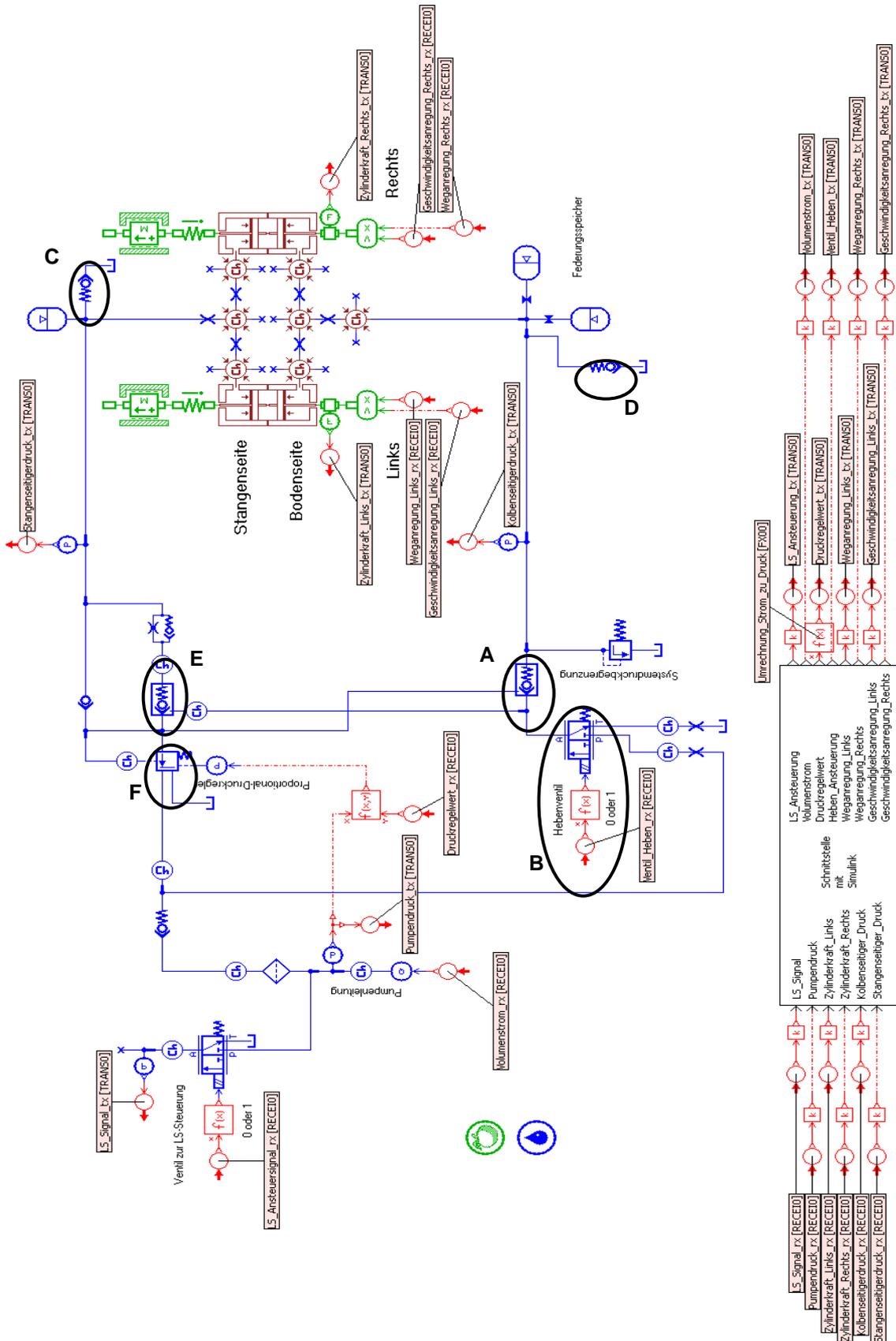


Abbildung 6.10 – Gesamtschaltbild des hydraulischen Federungskreises

6.3.2 Hydraulische Druckversorgung: Das DSHPlus-Modell

Mit dem auf die Modellierung von fluidtechnischen Systemen spezialisierten Programm *DSHplus* wird die Druckversorgung der Hydraulik im Traktor modelliert. Hierbei handelt es sich um eine Load-Sensing-Pumpe (LS-Pumpe). Bei diesem System ist die Pumpe mit einem Regler ausgerüstet, der es ermöglicht, die Pumpe in einen Stand-By-Zustand zurück zu schwenken, wenn vom System kein Volumenstrom benötigt wird, also z. B. Steuerventile in Neutralstellung sind. In diesem Zustand fördert die Pumpe kein bzw. sehr wenig Öl und baut nur einen geringen Druck auf. Wird Volumenstrom vom System benötigt, so stellt der Pumpenregler durch Vergleich von Lastdruck und Pumpendruck den Energiebedarf des Systems fest. Pumpendruck und –förderstrom werden so automatisch an den jeweiligen Bedarf angepasst.

Die LS-Pumpe versorgt in der mobilen Arbeitsmaschine die gesamte Hydraulik von der Lenkung über die Federung bis zur Arbeitshydraulik. In der Beispielanwendung „hydropneumatische Traktorvorderachsfederung“ ist neben der Pumpe nur die Achsfederung als hydraulisches System abgebildet. Ein kritischer Betriebszustand, wie er z. B. durch Volumenstrombedarf mehrerer Verbraucher bei unterschiedlichen Lastdrücken entsteht, ist nicht zu erwarten.

Das Modell der LS-Pumpe ist in *DSHplus* vereinfacht abgebildet (**Abbildung 6.11**), da zum einen nur eine geringe Detaillierungstiefe gefordert ist und zum anderen die Rechenzeit nicht zu lang werden soll.

Zentrales Element ist die Triebwerkseinheit *VP1*, eine Pumpe, deren Fördervolumen einerseits aus einer Eingangsdrehzahl und dem Fördervolumen und andererseits aus der Verstellung des Pumpenschwenkwinkels berechnet wird. Auf der Pumpenausgangsseite schließen sich ein Druckbegrenzungsventil, das den maximalen Systemdruck begrenzt, und eine Blende an, die als Hilfselement zur Ermittlung von hydraulischen Schnittstellendaten eingefügt wird.

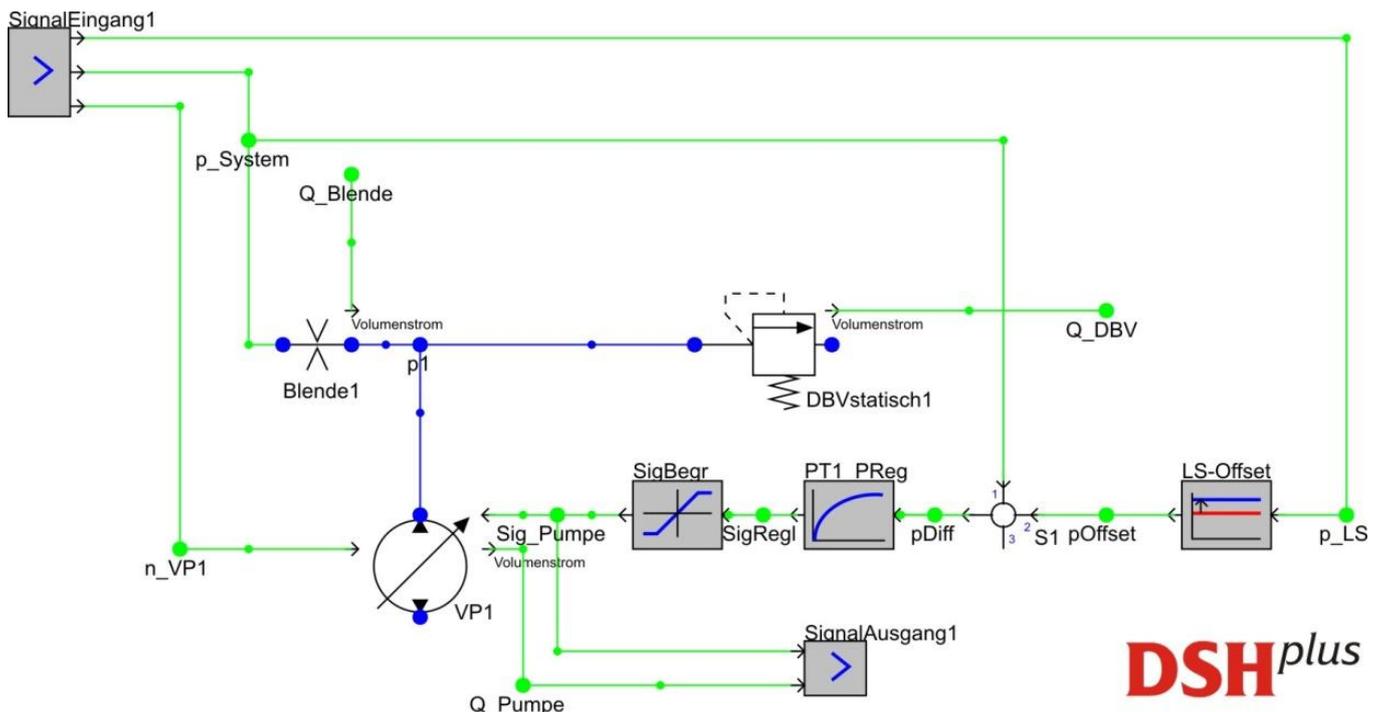


Abbildung 6.11 - Teilmodell LS-Pumpe mit Schnittstellenelementen

Der Pumpenregler setzt sich aus mehreren Elementen zusammen. Der LS-Druck (p_{LS}) wird mit einem Offsetdruck versehen ($LS-Offset$), der den Mindestsystemdruck darstellt. Der Vergleich von Lastdruck und Pumpendruck erfolgt mit $S1$. Aus dem berechneten Differenzdruck wird mit einer Verzögerung erster Ordnung das Stellsignal der Pumpe bestimmt ($PT1_PReg$). Die Verzögerung lehnt sich an die Geschwindigkeit des realen Stellsystems der Pumpe an. Eine Signalbegrenzung ($SigBegr$) sorgt dafür, dass die Pumpe nur im real möglichen Verstellbereich verschwenkt wird.

Im Modell werden zahlreiche Größen erfasst, die zur Simulation der Pumpe im Gesamtsystem nicht notwendig sind. Mit diesen Größen ist aber eine Beurteilung des Pumpenverhaltens während der Evaluierung des Modells einfach möglich. Die Parameter und Startwerte der Variablen des Modells sind im Kapitel 9.3.2 im Anhang dargestellt.

Zum hydraulischen Anschluss der Pumpe an die Traktorhydraulik sind zwei Größen notwendig, ein Druck als Ausgang und ein Volumenstrom als Eingang. Der Volumenstrom ist in diesem Fall als der Volumenstrom definiert, der vom System abgenommen wird. Die Druckänderung bestimmt sich aus der Differenz des abgenommenen Volumenstroms und dem Fördervolumenstrom der Pumpe.

Übergabewerte an das Gesamtsystem

Zur Interaktion des Pumpenmodells mit dem Gesamtsystem sind 3 Eingangsgrößen und 2 Ausgangsgrößen definiert. Alle anderen Größen sind unidirektionale, in SI-Einheiten übertragene Signale, die gelesen oder vorgegeben werden.

■ Input:

- LS-Druck [Pa],
- Systemdruck [Pa],
- Pumpendrehzahl [1/s].

■ Output:

- Ansteuersignal Pumpe [-],
- Volumenstrom [m^3/s].

Freigegebene / geschützte Parameter

Im Beispiel sind keine Parameter explizit geschützt oder freigegeben, um eine hohe Flexibilität während der Simulation zu haben. Im produktiven Einsatz wäre es sinnvoll, alle Parameter bis auf den abgesicherten Systemdruck (DBV) und den LS-Offset zu schützen. Dies entspricht den physikalischen Einstellmöglichkeiten an einer LS-Pumpe.

6.3.3 Mechanik: SIMPACK-Modell

Das Teilmodell "Mechanik" wurde im MKS-Werkzeug SIMPACK der am Projekt beteiligten Firma SIMPACK AG abgebildet.

Entsprechend der üblichen Vorgehensweise bei komplexeren Modellen wurde die Prüfstandsmechanik mit verschiedenen eigenständigen Untermodellen, sog. Substrukturen, abgebildet. In der Praxis gewährleistet dies die Wiederverwendbarkeit sowie den einfachen Austausch einzelner Baugruppen.

Das GUSMA Beispielmodell besteht aus den Teilen *Vorderachse*, *Rumpf* und *Prüfstands Aufbau*, die in einem Hauptmodell miteinander verknüpft sind. Für ein schnelles, problemloses Zusammensetzen des Gesamtmodells sorgen Verbindungskörper, deren Aufgabe es ist die Freiheitsgrade der einzelnen Teilmodelle gegenüber dem Raum bzw. anderen Teilmodellen aufzunehmen und somit einen starren Einbau zu ermöglichen.

In **Abbildung 6.12** ist die 3D-Darstellung des Gesamtmodells zu sehen.

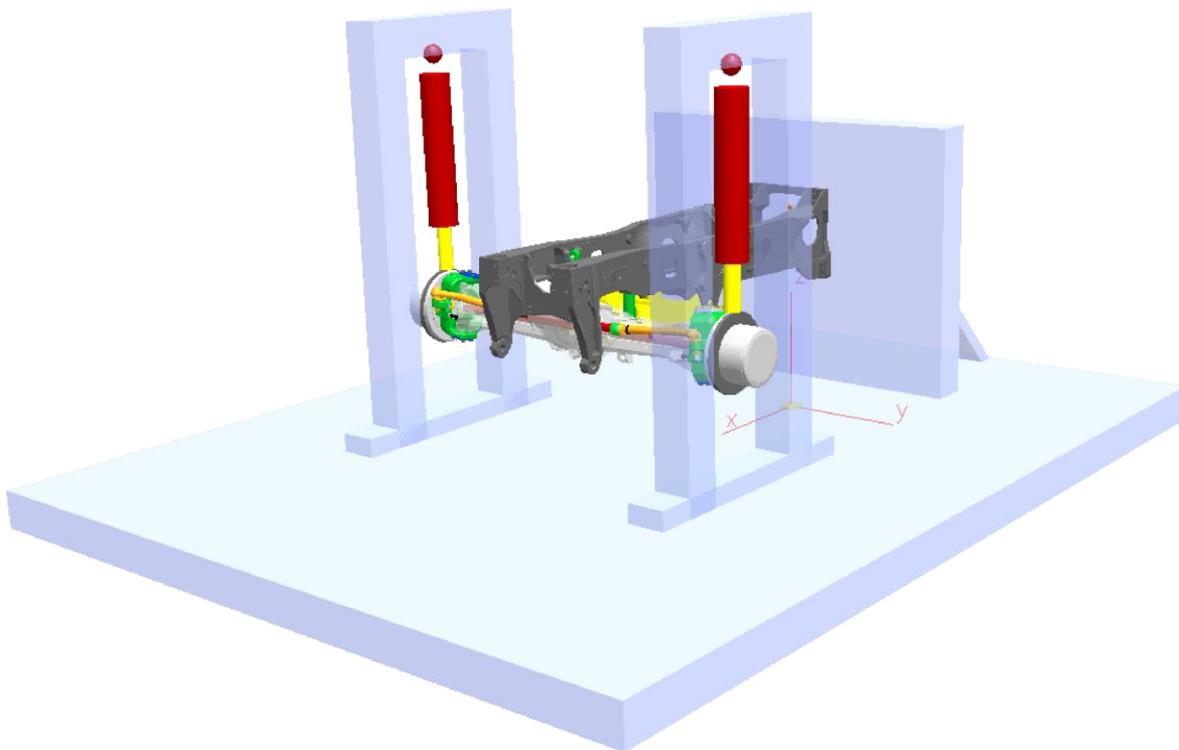


Abbildung 6.12 - Teilmodell Mechanik - 3D-Darstellung

Substruktur: Vorderachse

Das Subsystem *Vorderachse* wurde in einer größeren Detailtiefe aufgebaut als für die Beispielsimulationsaufgabe Vorderachsfederung nötig wäre. Um die Verwendbarkeit in anderen Modellen vorzuhalten, etwa einem Modell des Gesamtfahrzeugs, wurde die Lenkkinematik und ein einfaches Modell des Antriebsstranges mit modelliert.

Im Modell wurden ausschließlich starre Körper verwendet. Für den vorliegenden Verwendungszweck bedeutet dies keine Einschränkung der Ergebnisgüte, andere Randbedingungen könnten allerdings eine elastische Modellierung einzelner Bauteile, beispielsweise der Längsschwinge nötig werden lassen.

Vor dem eigentlichen Modellaufbau in SIMPACK wurden im CAD-System ProEngineer aus einem 3D-Modell die Positionen und Winkellagen der Koppelpunkte zu anderen Bauteilen (Körper in der MKS-Umgebung) und Kraftangriffspunkten für jeden Körper des Modells ermittelt und in eine Parameterdatei geschrieben. Es empfiehlt sich hier das Referenzsystem identisch zum körperfesten System im MKS-Modell zu wählen. Zu erwähnen ist hier, dass von der am Prüfaufbau des Lehrstuhls für mobile Arbeitsmaschinen verwendeten Achse keine ausreichenden 3D-Daten existieren, so dass das Modell einer ähnlichen, aber in ihren Abmessungen identischen Achse genutzt wurde. Die Gesamtmasse der Original-Vorderachse ist aus einer Wägung bekannt. Die Aufteilung der Masse auf die einzelnen Körper, deren Schwerpunktlagen und Trägheitstensenoren wurden ebenfalls mit Hilfe der CAD-Daten in guter Näherung ermittelt. Die Massen nicht modellierter Bauteile, wie etwa des Differentials, wurden den modellierten Körpern entsprechend der Einbausituation sinnvoll zugeschlagen.

Allgemein ist zur Bestimmung der Masseigenschaften eines komplexeren Systems, wie der Vorderachse eines Schleppers mit diversen Teilen unterschiedlicher Zulieferer oft experimenteller Aufwand nötig, da in der Regel die CAD-Daten nicht in der nötigen Tiefe vorliegen um allein auf deren Basis eine ausreichende Berechnung durchführen zu können. Für das Einsatzspektrum des Beispielmodells mit den vor allem translatorischen Bewegungen des Achskörpers sind die vorhandenen Daten und getroffenen Annahmen absolut ausreichend.

Mit den generierten Parametern wurde in SIMPACK das eigentliche Modell der Vorderachse erstellt. Der schematische Aufbau ist in **Abbildung 6.13** zu sehen.

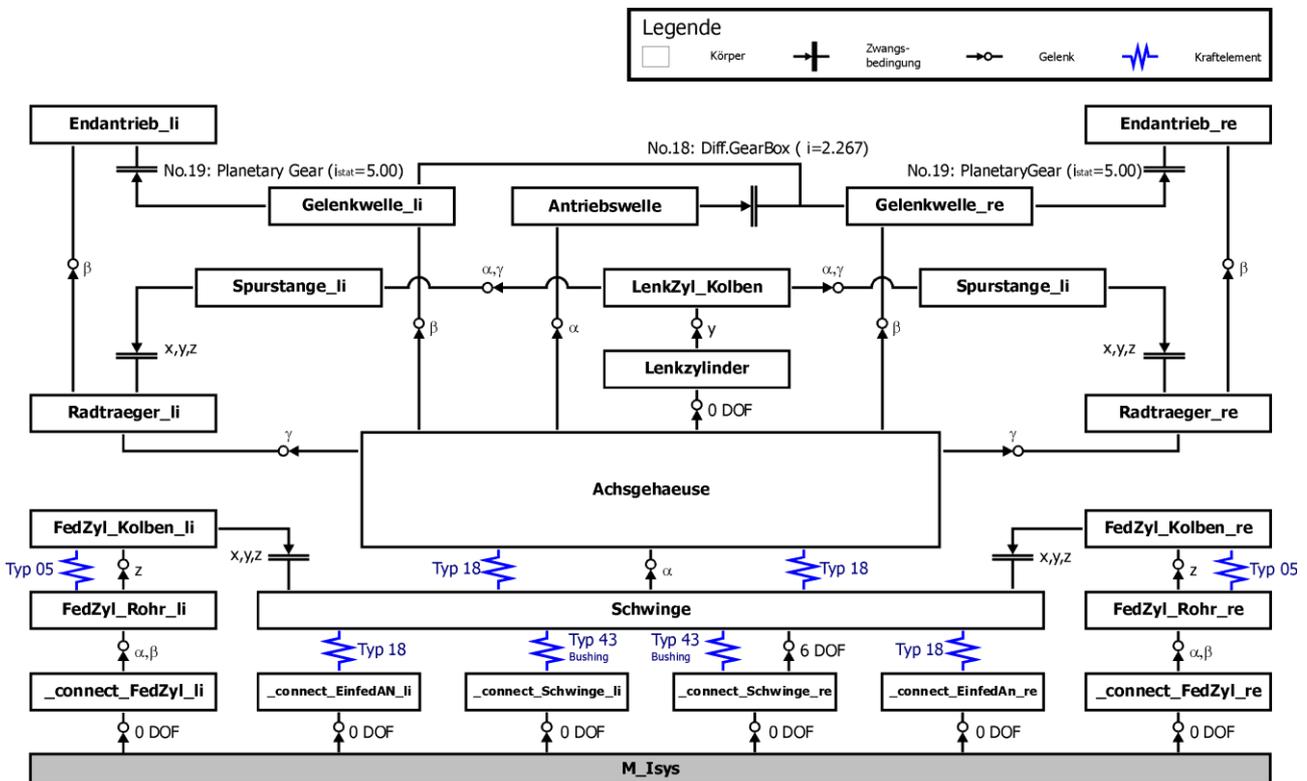


Abbildung 6.13 - Schematischer Aufbau des MKS-Modells der Substruktur Vorderachse

Dem Modell verbleiben zehn unabhängige Freiheitsgrade (*independent states*), die mit Startwerten vorbelegt werden müssen. Dies sind:

- Schwinge - Translation x, y, z und Rotation um x und z
- Federungszyylinder rechts - Translation z
- Achsgehäuse - Rotation um x
- Antriebswelle - Rotation um x
- Gelenkwelle rechts - Rotation um y
- Lenkzylinder Kolben - Translation y

Alle weiteren Positionen ergeben sich aus oben aufgeführten. Die drei Freiheitsgrade von Lenkung und Antriebsstrang sind für die Verwendung im Beispiel *Prüfstand* unerheblich und werden im Hauptmodell gesperrt.

Die Elastomerbuchsen der Verbindung Rumpf/Längsschwinge sind als 6-dimensionales Kraftelement abgebildet. Als Steifigkeiten sind momentan die statischen Werte der Herstellerzeichnung hinterlegt. Zur Berücksichtigung der Dynamik sind hier Prüfstandsversuche in Vorbereitung.

Die mechanischen Endanschläge der Federungszyylinder sind durch eine Kraft-Weg-Kennlinie mit der Zylinderlänge als Eingangsgröße enthalten. Bevor die Federzylinder allerdings in den Einfederanschlag kommen steht bereits die Schwinge am Halbrahmen des Schlepper an. Dieser Gussanschlag ist als einseitig wirkendes Feder-Dämpfer-Element modelliert. Ebenso der Pendelanschlag, der die Drehbewegung der Achse um die Längsachse begrenzt. Die Steifigkeiten und Dämpfungen aller Anschläge sind Schätzwerte.

Zur anschaulichen Darstellung des Modells wurden in SIMAPCK so genannte Primitivkörper erzeugt bzw. CAD-Geometrien importiert. Um die Animation handhabbar zu lassen wurden die CAD-Daten zuvor vereinfacht.

Substruktur: Rumpf

Das Teilmodell des Fahrzeugrumpfs hat lediglich die Funktion den Prüfling mit dem Prüfstand starr zu verbinden. Entsprechend besteht es hier nur aus einem Körper, dem Halbrahmen des Fahrzeugs, der mit 0 Freiheitsgraden am Prüfstand fixiert wird. Für die Animation wurde eine CAD-Geometrie importiert.

Substruktur: Prüfstandsaufbau

Die Substruktur *Prüfstandsaufbau* bildet die Prüfstandsumgebung am Lehrstuhl für mobile Arbeitsmaschinen nach. Die Funktion besteht wie am realen Prüfstand in der Aufnahme des Prüflings, hier Rahmen und Achse, sowie dessen Anregung. In **Abbildung 6.14** ist der schematische Aufbau zu sehen.

Das Teilmodell hat eine offene Baumstruktur, ohne Schleifenschlüsse mit den folgenden Freiheitsgraden:

- Prüfzylinder Rohr rechts & links - Rotation um x, y und z
- Prüfzylinder Kolben rechts & links - Translation z
- Achsaufnahme rechts & links - Rotation um x

Beim Einbau in das Gesamtmodell der Mechanik werden alle Freiheitsgrade gesperrt, bis auf die Translation des Kolbens, über den die Anregung eingeleitet wird. Im folgenden Kapitel über die Testumgebung ist dies näher beschrieben.

Für die 3D-Animation wurde der Aufbau mit Primitivkörpern nachgestellt.

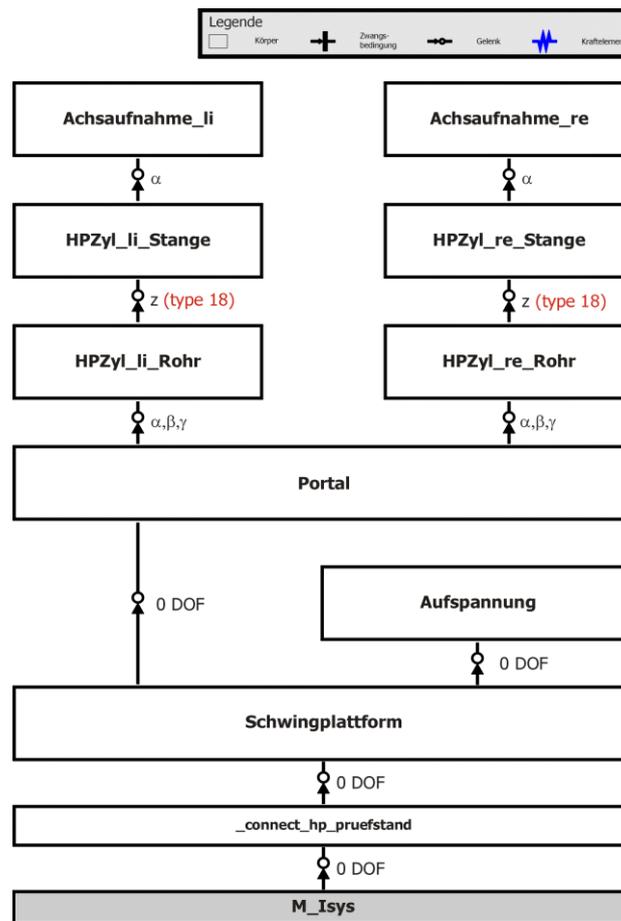


Abbildung 6.14 - Schematischer Aufbau der Substruktur *Prüfstandsaufbau*

Parameter

Name und Wert (oder Formel zur Berechnung) der Modellparameter des virtuellen Prüfstands sind fast ausschließlich in so genannten Parameterdateien definiert. Jedes der drei Teilmodelle besitzt zwei dieser Dateien, die beim Einbau mit ins Hauptmodell übernommen werden.

In der ersten Datei sind die allgemeinen modellspezifischen Parameter enthalten, etwa geometrische Abmessungen, Massen oder Werte der Kraftelemente. Diese Dateien sind vom Hauptmodell unabhängig und müssen nur bei einer Anpassung des Teilmodells geändert werden.

Im Gegensatz dazu können die zweiten Dateien untereinander Abhängigkeiten besitzen. In diesen sind die Parameter definiert, die sich mit der Simulationsaufgabe ändern, wie beispielsweise die Startwerte. Über die Anordnung in der <Hauptmodell>.dbi - Datei, in der alle zu ladenden Parameterdateien stehen, ist eine Rangfolge gegeben. Auch das Hauptmodell an sich besitzt zwei Parameterdateien. Diese stehen in der Hierarchie oben. Hier wird etwa die Art der Prüfstandsanzregung vorgegeben. Durch formelmäßige Verknüpfungen leiten sich daraus dann die Werte weiter unten stehender Dateien ab, z. B. die Längen von Federungs- oder Hydropulszylinder. In **Abbildung 6.15** sind diese Abhängigkeiten schematisch dargestellt.

Durch die Verwendung dieser Struktur wird zum Einen die Austauschbarkeit der Teilmodelle unterstützt zum Anderen verhindert sie vor allem eine redundante Definition von abhängigen Parametern, was mit einer größeren Übersichtlichkeit und geringeren Fehleranfälligkeit gleichzusetzen ist.

Um mit den Teilmodellen auch separat arbeiten zu können, ist es allerdings nötig, die hierarchisch übergeordneten Dateien mit einzulesen oder diese durch Dateien mit Blindwerten zu ersetzen.

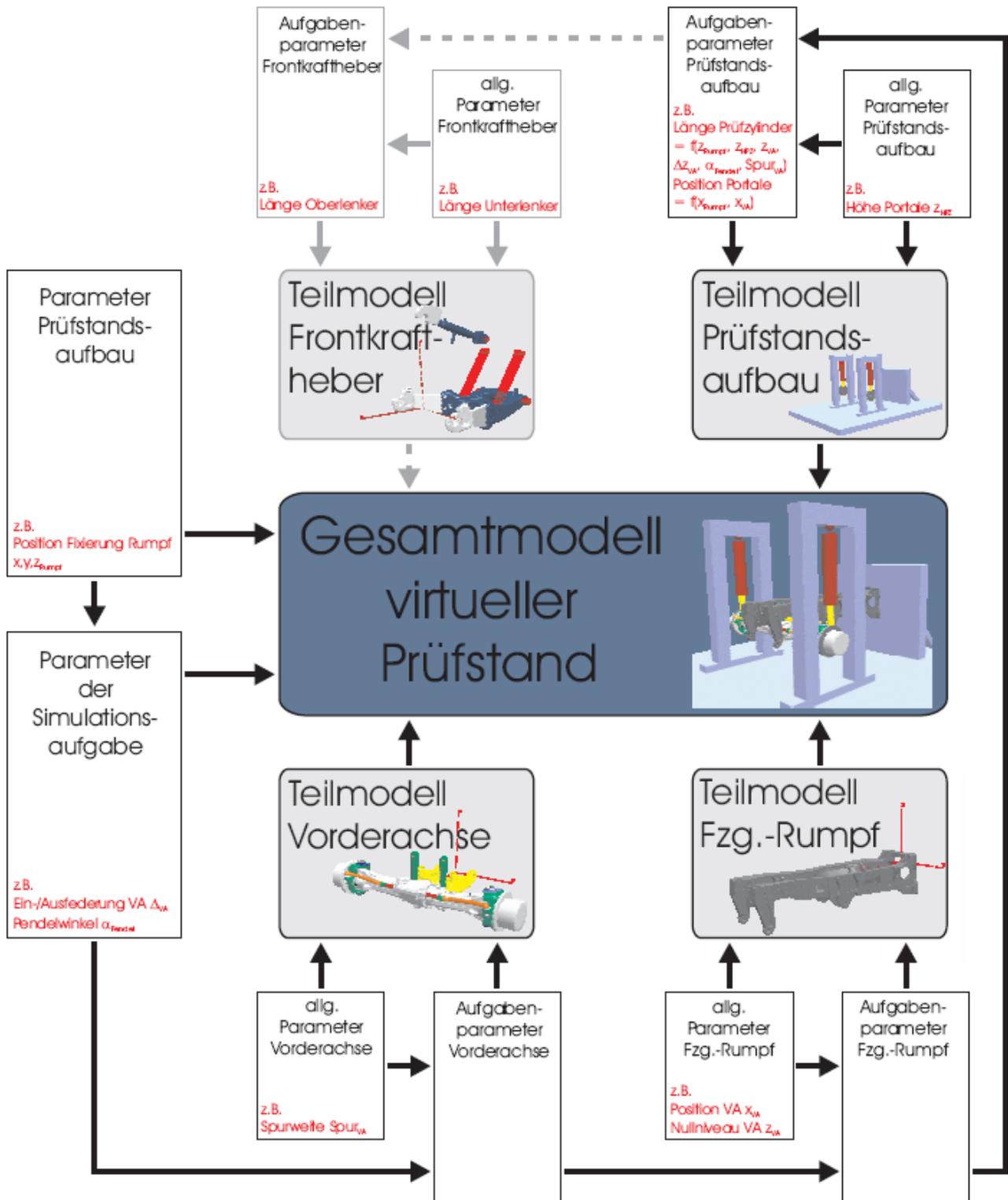


Abbildung 6.15 - Übersicht Parameterstruktur des mechanischen Modells

Übergabewerte an das Gesamtsystem

Das mechanische Modell des Prüfstandsbaus steht mit den Teilmodellen der Hydraulik und der Regelungstechnik über folgende Austauschgrößen in Verbindung:

- Input:
 - Hydraulische Kraft Federungszyylinder rechts/links (vom Hydraulikmodell der Federung)
- Output:
 - Einfederung Federungszyylinder rechts/links
 - Einfederungsgeschwindigkeit der Federungszyylinder rechts/links (jeweils an das Hydraulikmodell der Federung)
 - Stromwert des Lagesensors (an das Modell der Regelung)
 - Fahrgeschwindigkeit
 - Zustand Heckzapfwelle

Der Austausch erfolgt über die $u(t)$ -input bzw. die $y(t)$ -output Vektoren.

Die Einfederung sowie die Geschwindigkeit werden direkt dem Zustand der translatorischen Gelenke beider Kolben entnommen. Der Stromwert des Lagesensors wird über eine Kennlinie aus der Einfederung berechnet. Die Übergabe lediglich des Schwingenwinkels mit anschließender Berechnung im Modell der Regelung wäre hier allerdings eine sauberere Trennung der Disziplinen.

Vom hydraulischen Modell der Federung wird die Zylinderkraft zurückgemeldet und dient als Eingang für zwei Kraftelemente die zwischen Kolben und Rohr der Zylinder wirken.

Testumgebung

Wie in diesem Abschnitt ersichtlich wird, sind dem Hauptmodell der Mechanik zwei Parameterdateien zugeordnet. Mittels der Datei, die die aufgabenspezifischen Werte enthält kann der Nutzer die Anregung des Prüfstands vorgeben. Es sind momentan folgende Szenarien auswählbar:

- Wegvorgabe
 - synthetisches Sinussignal (Frequenz, Amplitude, Phase einstellbar)
 - Zeitverlauf aus Messung am Realsystem
- Kraftvorgabe
 - vordefinierte oder freie Achslast (jeweils konstant)
 - Zeitverlauf aus Messung am Realsystem

Die Umsetzung im Modell erfolgt zum einen durch eine zusätzliche Datei <Hauptmodell>.add, die die Parameter des Gelenks der Kolbenstangen der Hydropulszylinder überschreibt, die eigentlich im Teilmodell definiert sind. Zum anderen wurde im Hauptmodell pro Pulszylinder eine Kraft eingefügt mit einem Betrag der Vorgabe entsprechend bzw. mit Betrag Null bei Wegvorgabe.

Die Messungen aus dem Realsystem entstammen beispielsweise dem Fahrbetrieb beim Fahren über Feld- und Waldwege, auf denen je nach Wegprofil Stöße in das Fahrzeug eingeleitet werden. In **Abbildung 6.16** ist das

Wegprofil einer solchen Feldwegfahrt exemplarisch dargestellt. Auf der Hochachse ist dabei die Wegamplitude in Metern und auf der Querachse die Zeit in Sekunden aufgetragen.

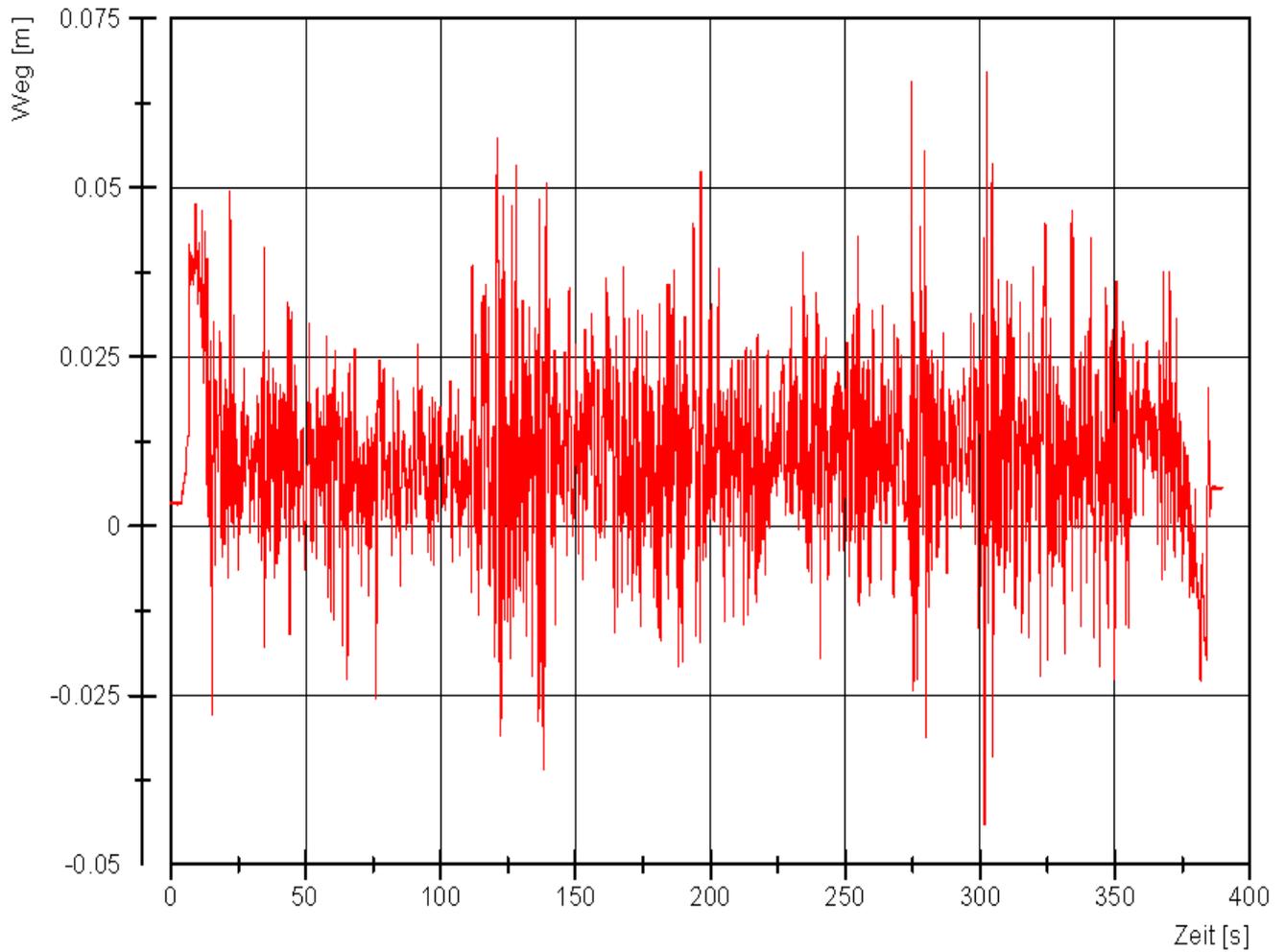


Abbildung 6.16 - Wegprofil einer Feldwegfahrt eines Traktors

6.3.4 Steuerung: Simulink-Modell

Das regelungstechnische Teilmodell der Niveauregulierung der Traktorvorderachse wurde mit Hilfe des Simulationswerkzeugs Simulink erstellt. Simulink ist integriert in die Entwicklungsumgebung MATLAB der Firma Mathworks (nicht am Projekt GUSMA beteiligt). Innerhalb des Projekts findet Simulink auch Verwendung als Plattform für die Verknüpfung der einzelnen Teilmodelle.

Das Teilmodell der Niveauregulierung wird als eigenständiges Modell auf die GUSMA-Plattform „gezogen“ und mit den anderen Teilmodellen verknüpft. Aus der Auswertung der Informationen über den aktuellen Zustand des Gesamtsystems „Fahrzeug“ (Eingangsgrößen des Teilmodells) werden Vorgaben (Ausgangsgrößen) für dessen Beeinflussung gemacht. Als Besonderheit kommen hier Eingangsgrößen auch aus der Definition der Simulationaufgabe (z.B. Niveauregulierung aktiv oder inaktiv) auf der Plattform selbst und nicht nur aus einem anderen Teilmodell.

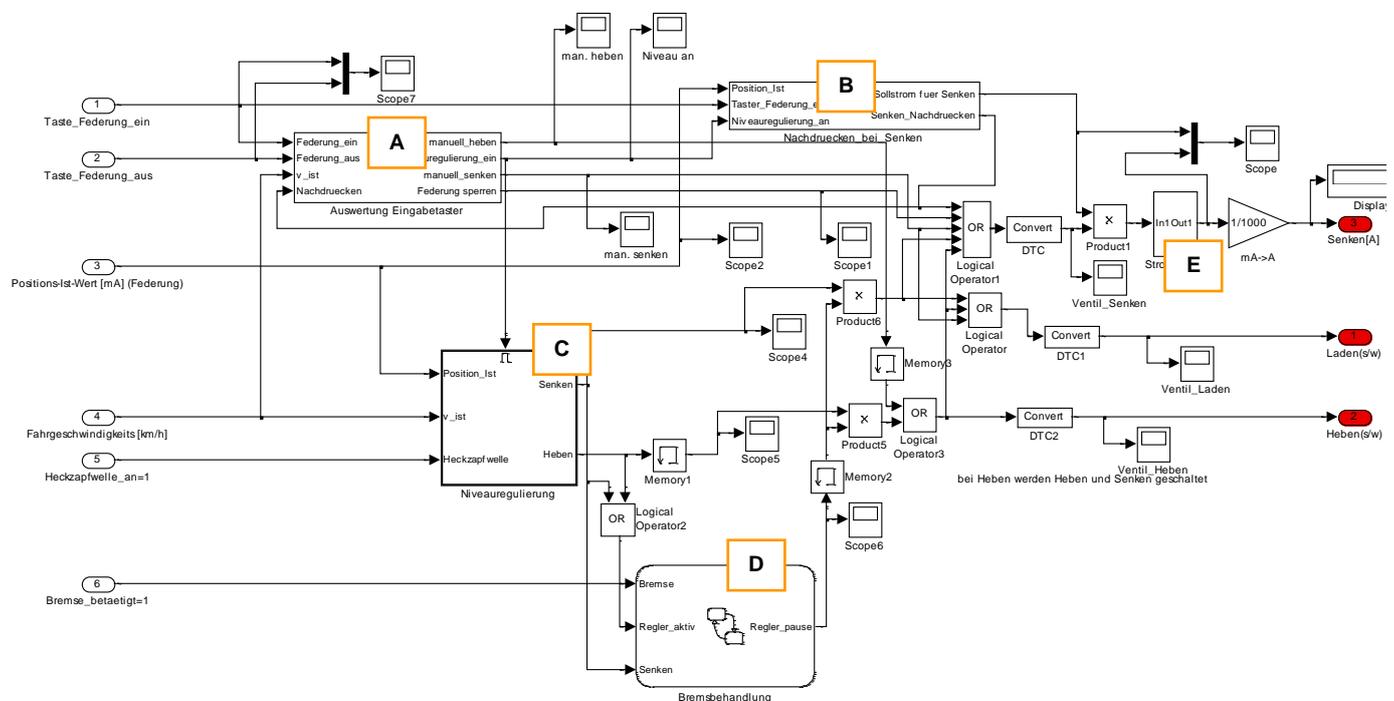


Abbildung 6.17 - Simulink-Modell der Niveauregulierung (Hauptebene)

Das Modell Niveauregulierung gliedert sich in fünf Submodelle (siehe **Abbildung 6.17**):

- **A – Auswertung Eingabetaster:** Verarbeiten der Signale aus der Definition der Simulationaufgabe, der Fahrzeuggeschwindigkeit und der Sonderfunktion „Nachdrücken bei Senken“ (siehe B) zur Bestimmung des Systemzustands (1. Niveauregulierung ein; 2. Federung sperren; 3. manuelles Heben und 4. manuelles Senken).
- **B – Nachdrücken bei Senken:** Erkennen des Zustands „Federung sperren“ und Sicherstellen der definierten Achsposition.
- **C – Niveauregulierung:** Erkennen und Reagieren auf Lageabweichungen der Vorderachse unter Berücksichtigung von Fahrgeschwindigkeit und Zustand der Heckzapfwelle

- **D – Bremsbehandlung:** Überstimmen von Regeleingriffen aus dem Subsystem Niveauregulierung am bremsenden Fahrzeug
- **E – Stromregler:** Bestimmen des Stromwerts zur Ansteuerung des Proportionalventils der Ringseite des Federzylinders

Regelbetrieb

Über kurzzeitiges Drücken der Taste „Federung betätigt“ wird die Niveauregulierung aktiviert. Ist die Fahrgeschwindigkeit größer als v_1 (sofern die Zapfwelle angeschaltet ist, beträgt dieser Wert v_2), wird die Position der Vorderachse auf einen festgelegten Wert, der ca. der Mitte entspricht, geregelt. Bewegt sich der ungefederte Teil der Vorderachse nach Einregelung der Mittellage nach oben oder unten, müssen die beiden Totbänder (für unten und oben getrennt parametrierbar) zusätzlich zur Fahrgeschwindigkeit überschritten bzw. unterschritten sein. Die Stromwerte des Positionsgebers sind 4...20mA, wobei der kleinere Wert für „Federung ganz ausgehoben“ steht. Zusätzlich muss die Abweichung zum Sollwert für einen einstellbaren Wert (mit t_1 eingestellt) anliegen, bevor der Regler die Magnetventile bestromt.

Wird die Taste „Federung aus“ kurzzeitig betätigt, wird das Fahrzeug abgesenkt und der Modus „Federung sperren“ aktiv. Federung sperren heißt allerdings, dass die Längsschwinge am Anschlag ansteht. Um die Lage am Anschlag sicher zu gewährleisten, wird ab einem Wert nahe des Anschlags der Druck auf der Ringseite vom Normaldruck bei aktivierter Federung auf den maximalen Systemdruck erhöht und nach Erreichen des Anschlags für eine Zeitspanne t_2 auf 200 bar gehalten.

Bei Betätigung der Bremse und anschließendem Verlassen des Totbandes werden die Magnetventile nicht bestromt. Sofern die Bremse während eines Regelvorgangs betätigt wird, wird ein Absenken unterbunden, während ein Abheben weiter durchgeführt wird.

Manueller Betrieb

Werden die Taster „Federung betätigt“ oder Taster „Federung aus“ länger als drei Sekunden gedrückt, wird der Federungsweg solange durchfahren, bis entweder die Anschläge erreicht sind oder die Taste losgelassen wird.

Initialisierung

Um den Zustand der Niveauregulierung zu Beginn der Simulation festzulegen, wurde ein Initialisierungsblock (Initial Condition) eingefügt. Die Belegung dieses Initialisierungsblocks erfolgt über den Block „Anfangszustand Niveauregulierung“. Je nach dem hier eingetragenen Wert (0=„inaktiv“, 1=„aktiv“), beginnt die Simulation mit „inaktiver“ oder „aktiver“ Niveauregulierung.

Übergabewerte an das Gesamtsystem

Das regelungstechnische Modell steht mit den Teilmodellen der Hydraulik und der Mechanik über folgende Austauschgrößen in Verbindung:

- Input:
 - Zustand Taste Federung ein [-] (aus Definition der Simulationsaufgabe; input1 in Abbildung 1.1)
 - Zustand Taste Federung aus [-] (aus Definition der Simulationsaufgabe; input2)

- Ist-Position der Längsschwinge [mA] (aus Teilmodell Mechanik; input3)
- Fahrgeschwindigkeit [km/h] (aus Teilmodell Mechanik; input4)
- Zustand Heckzapfwelle [-] (aus Teilmodell Mechanik; input5)
- Zustand Bremse [-] (aus Teilmodell Mechanik; input6)

■ Output:

- Strom für Magnetventil Senken [mA] (output3)
- Zustand Magnetventil Heben [-] (output2)
- Zustand Magnetventil Laden [-] (output1; alle für Teilmodell Hydraulik „Federung“)

6.3.5 Aufbau des Gesamtsystems auf der GUSMA-Plattform

Das Gesamtsimulationsmodell der Traktorvorderachsfederung wurde aus den von den Firmen gelieferten Teilmodellen unter Simulink aufgebaut. Die hierfür notwendigen Vorbereitungen der Teilmodelle wurden bereits ausführlich in den vorangegangenen Abschnitten erläutert. Der nächste Schritt ist somit das Zusammenfügen der Teilmodelle. Der Aufbau des Gesamtmodells lässt sich jedoch nicht ohne weitere Vorbereitungen und durch einfaches Verbinden der Teilmodelle bewerkstelligen. Daher wurden Zwischenschritte durchgeführt, die nachfolgend erläutert werden sollen.

Überprüfungen der einzelnen Teilmodelle

Jedes Teilmodell wurde sowohl vom Ersteller als auch vom Lehrstuhl auf seine Funktionssicherheit hin überprüft, d.h. die Teilmodelle wurden auf Plausibilität getestet. Hierfür wurden verschiedene Testszenarien entwickelt. Am Lehrstuhl wurde vor allem das Verhalten der Teilmodelle bei der Co-Simulation untersucht. Lediglich das hydraulische Teilmodell Federung mit Niveauregulierung, welches das zentrale Element des Gesamtsimulationsmodells bildet, wurde auch in AMESim getestet.

Für diesen Test wurden vom Modellersteller alle Elemente der Traktorvorderachsfederung in AMESim abgebildet. Das AMESim-Modell zeigt **Abbildung 6.18**. Der Aufbau und die Funktionsweise des Modells wurden bereits in der Dokumentation beschrieben. Prinzipiell enthält das Modell aus **Abbildung 6.18** das Teilmodell für die Gesamtsystems simulation. An dieser Stelle sollen daher nur die einzelnen Elemente genannt werden, ohne weiter auf ihre Funktion einzugehen.

Auf der rechten Seite der Darstellung sind in dunkelrot die beiden hydraulischen Federungszyylinder (1) abgebildet, die mit der in grün dargestellten Federungsmechanik (2), bestehend aus der Masse der Elemente sowie der kinematischen Übersetzung der Federungskinematik, verbunden sind. Im oberen Bereich befindet sich die Stangenseite der Federungszyylinder, im unteren Bereich die Kolbenseite. Die eigentliche Niveauregulierung (3) ist über Signalglieder abgebildet. Die hydraulische Versorgung der Traktorvorderachsfederung (4) ist links unten als vereinfachte LS-Pumpe dargestellt. Angesteuert wird sie über das LS-Ventil (5). Vervollständigt wird das Modell durch das Druckregelventil (6), das Heben-/Senken-Ventil (7) und den stangenseitigen (8) sowie die kolbenseitigen Speicher (9).

Beim Funktionstest sinkt die Vorderachse aufgrund ihres Eigengewichts in den Federzylinderendanschlag. Diese Auslenkung aus der Mittellage wird ermittelt und durch die Niveauregulierung durch Ansteuerung der entsprechenden Ventile ausgeregelt, so dass die Vorderachse in ihre Mittellage zurückkehrt. Die Funktion des Modells ist daher nachgewiesen.

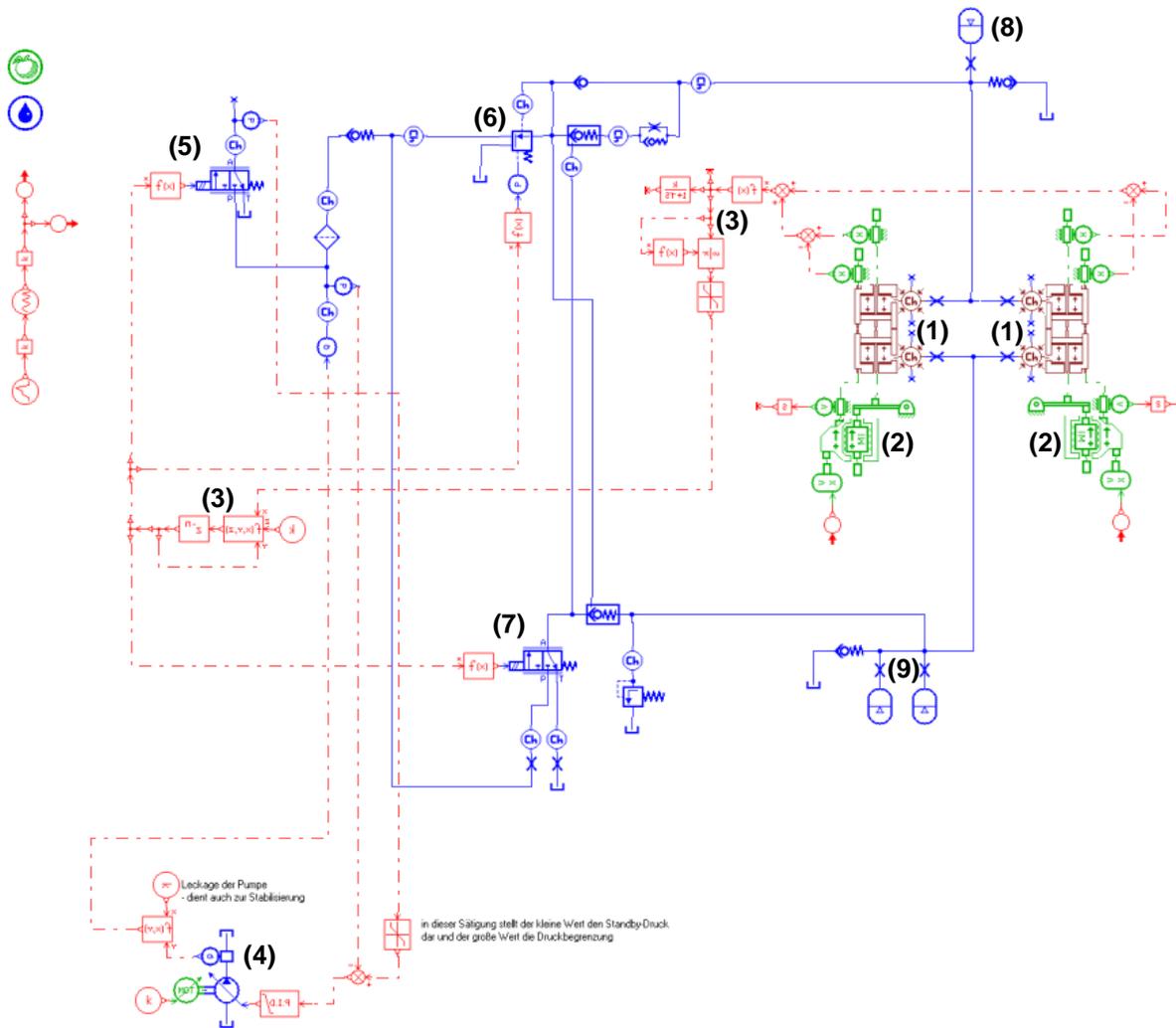


Abbildung 6.18 -Traktorvorderachsffederung in AMESim

Die Überprüfung des zweiten Hydraulikmodells erfolgte in Simulink, indem die Co-Simulation von Simulink mit DSHplus anhand des Modells der LS-Pumpe getestet wurde. Hierfür wurde ein einfaches Simulinkmodell aufgebaut, welches in **Abbildung 6.19** zu sehen ist. Der LS-Pumpe (DSHplus-Block in Abbildung 6.18) wurden die Variablen Drehzahl, Systemdruck und LS-Druck übergeben und der entsprechende Volumenstrom der Pumpe sowie das Pumpensignal wurden ausgegeben. Die Co-Simulation von DSH und Simulink funktionierte und das Modell lieferte plausible Werte.

Des Weiteren musste die Co-Simulation zwischen SIMPACK und Simulink sowie die Funktion des Mechanikmodells getestet werden. Hierfür wurde ein weiteres Modells erstellt, welches in **Abbildung 6.20** dargestellt ist. Das Mechanikmodell besitzt als Eingangsgrößen die Kräfte der Federungszyylinder und gibt die Lage der Federung sowie deren Geschwindigkeit aus. Zur Überprüfung der Funktionssicherheit wurden daher die „äußeren“ Kräfte auf das Mechanikmodell eliminiert (Konstanten mit dem Wert 0). Dadurch sinkt das Modell aufgrund des Fahrzeuggewichtes in seinen oberen Endanschlag. Diese Bewegung konnte durch den Plot (Weg_li oder Strom Lage-sensor) überprüft werden. Die Co-Simulation von SIMPACK und Simulink funktionierte und das Modell lieferte plausible Werte.

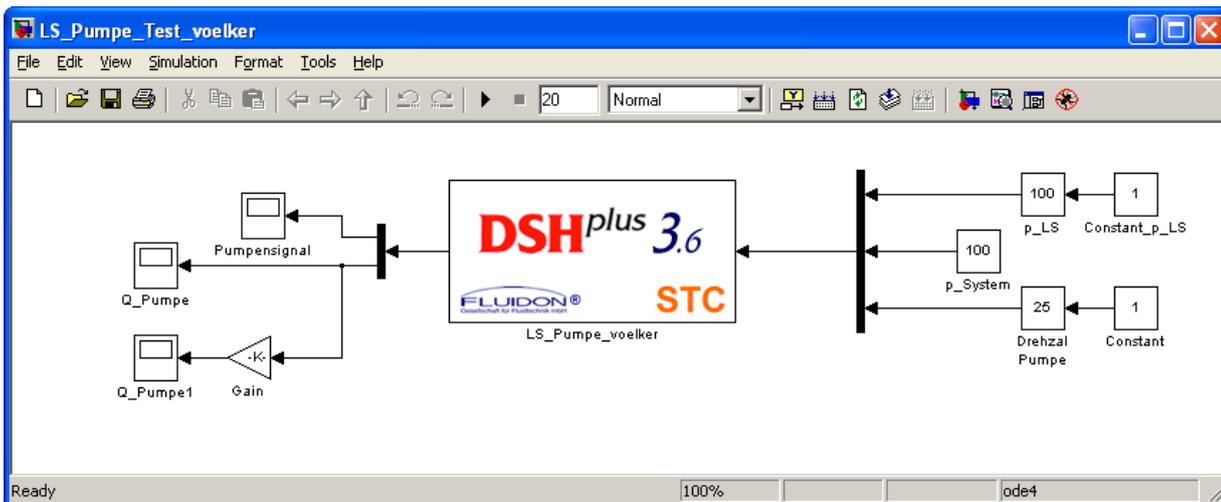


Abbildung 6.19 - Überprüfung des Teilmodells LS-Pumpe

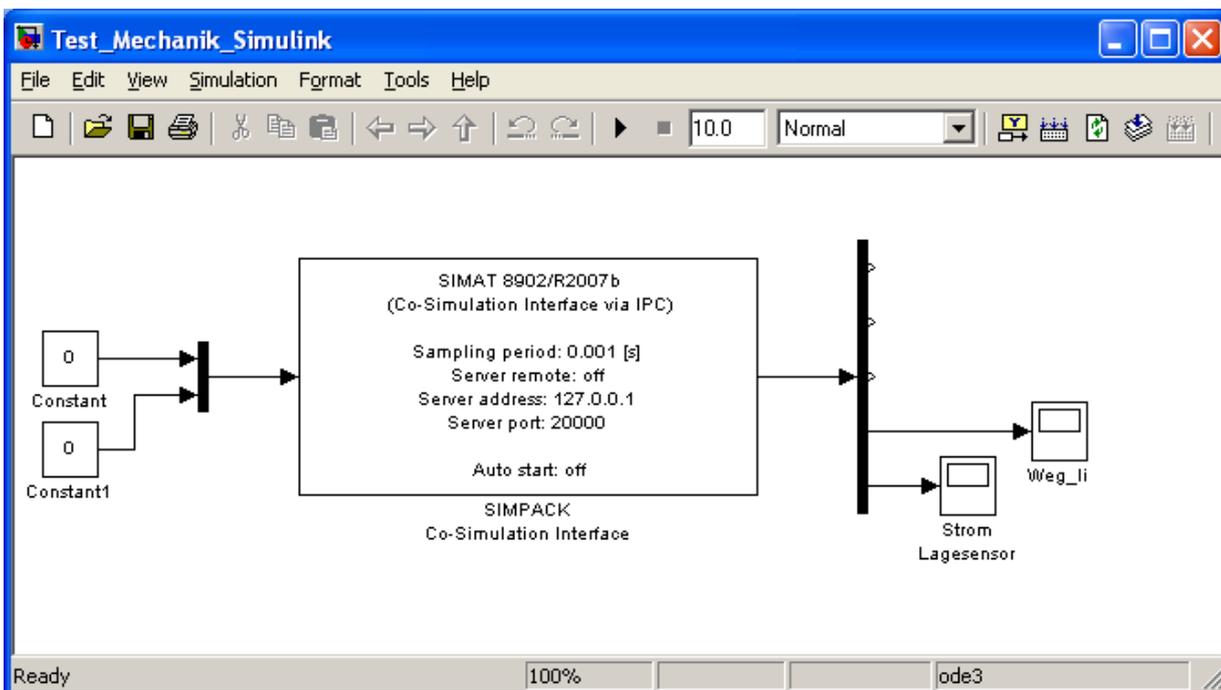


Abbildung 6.20 - Überprüfung des Teilmodells Mechanik

Für das Steuerungsmodell wurden keine separaten Testszenarien entwickelt, da das Modell nicht in Co-Simulation betrieben wird. Es ist bereits in Simulink aufgebaut und wurde lediglich in Kombination mit den anderen Modellen getestet, was nachfolgend beschrieben werden soll.

Überprüfung der Teilmodelle in Kombination zueinander

Für den Aufbau des Gesamtsimulationsmodells wurden im Anschluss an die Einzeltest Kombinationen verschiedener Teilmodelle getestet. Zentrales Element bei diesen Tests war jeweils das Teilmodell der Firma Hydac, das AMESim-Modell der Traktorvorderachsfederung mit Niveauregulierung. Es wurden Kombinationen dieses Teilmodells mit der Steuerung, mit der hydraulischen Versorgungseinheit sowie mit der Mechanik durchgeführt.

Für den in **Abbildung 6.21** dargestellten Test der Hydraulik mit der Steuerung wurden die Steuerungselemente aus dem Hydraulikmodell entfernt und hierfür entsprechende Übergabevariablen definiert.

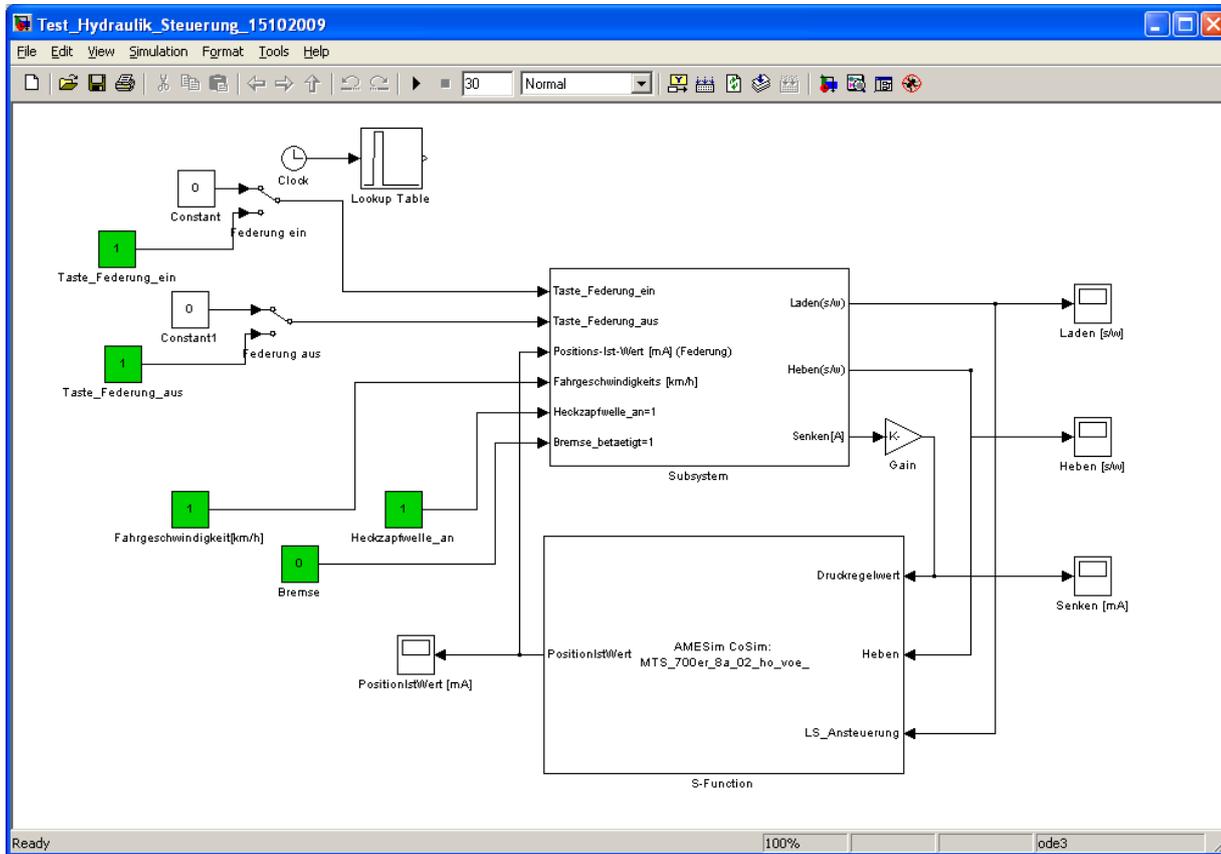


Abbildung 6.21 - Überprüfung der Kombination aus Mechanik und Hydraulik

Im oberen Bereich der Abbildung 6.21 ist das Regelungsmodell dargestellt. Die Ventilansteuersignale werden an die Hydraulik übergeben und stellen demnach die Ausgangsgrößen des Teilmodells dar. Als Eingangsgröße wird dem Teilmodell der Weg aus dem hydraulischen Modell übergeben. Hinzu kommen Eingangsgrößen, die vom Fahrzeug bzw. vom Benutzer stammen. Zum einen werden die Bremse, die Heckzapfwelle sowie die Fahrgeschwindigkeit überwacht. Unterhalb spezifischer Geschwindigkeitsgrenzen, die wiederum an den Betrieb oder Nichtbetrieb der Heckzapfwelle geknüpft sind, wird die Niveauregulierung deaktiviert. Ebenso wird überwacht, ob die Bremse aktiviert ist, um eine Überlagerung der Bremsbewegung mit einem eventuellen Absenken durch die Niveauregulierung zu vermeiden. Im linken oberen Bereich sind die beiden Taster dargestellt, mit deren Hilfe der Fahrer die Federung aktivieren oder deaktivieren kann.

Mit Hilfe dieses Simulationsmodells konnte die Funktionssicherheit der Kombination dieser beiden Teilmodelle nachgewiesen werden. Hierfür waren mehrere Simulationsdurchläufe nötig, in denen die Modelle angepasst wurden. Letztlich konnte die Freigabe des Steuerungsmodells für das Gesamtsimulationsmodells gegeben werden. Im nächsten Schritt wurden die beiden Hydraulikmodelle miteinander kombiniert. **Abbildung 6.22** zeigt das Testmodell von AMESim und DSHplus unter Simulink. In diesem Fall liefert das Modell der Federung den System- und LS-Druck an die hydraulische Versorgungseinheit und bekommt den Volumenstrom der Pumpe rückgemeldet.

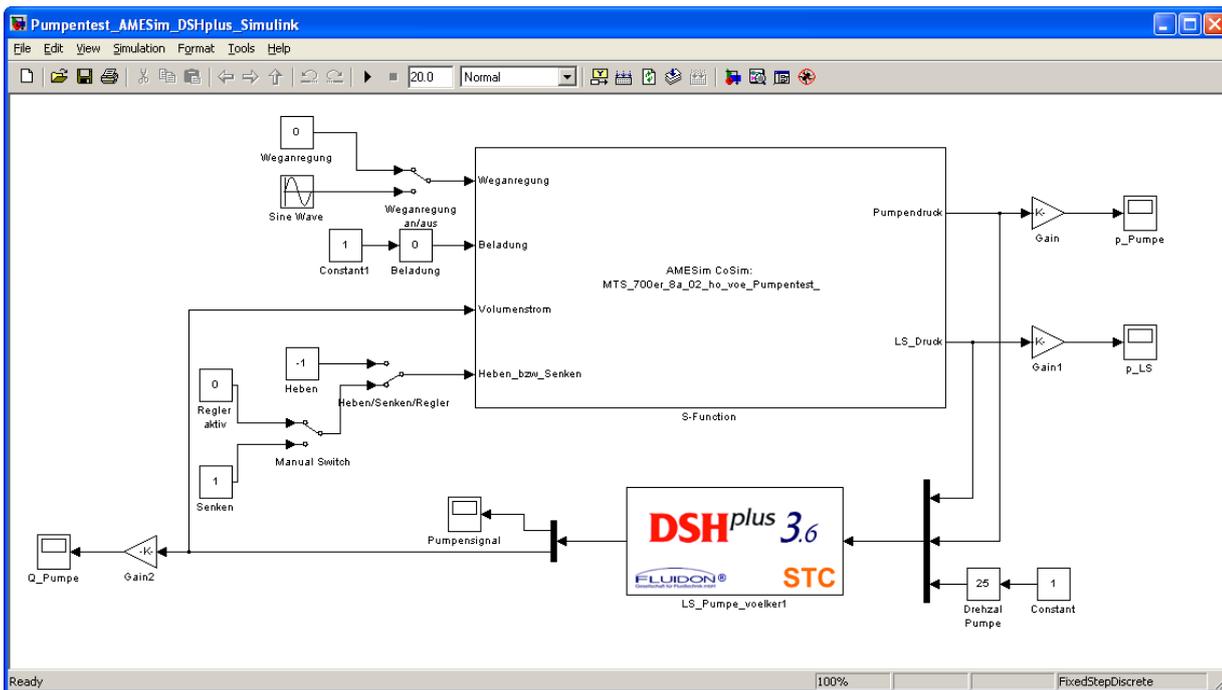


Abbildung 6.22 - Überprüfung der Kombination aus Hydraulik und Hydraulik

Das Federungsmodell besitzt darüber hinaus noch weitere Eingänge. Durch den Benutzer können von außen die Beladung des Fahrzeugs sowie eine Weganregung vorgegeben werden. Darüber hinaus kann der Benutzer die Federung aktivieren oder deaktivieren.

Auch in diesem Fall konnte mit Hilfe des Modells die Funktionssicherheit dieser beiden Teilmodelle nachgewiesen werden. Sie wurden ebenfalls im Zuge der Simulation angepasst, so dass letztlich das Pumpenmodell für das Gesamtsimulationsmodell freigegeben werden konnte.

Abschließend musste nun noch die Überprüfung der Kombination Mechanik mit Hydraulik durchgeführt werden. Auch hierfür wurde erneut ein spezielles Federungsmodell erzeugt, welches die Pumpe, den Regler und die Federung beinhaltet, jedoch keine mechanischen Komponenten. **Abbildung 6.23** zeigt das Simulationsmodell dieser beiden Teilmodelle. Im oberen Bereich ist erneut das AMESim-Modell der Federung und im unteren Bereich das mechanische Modell dargestellt.

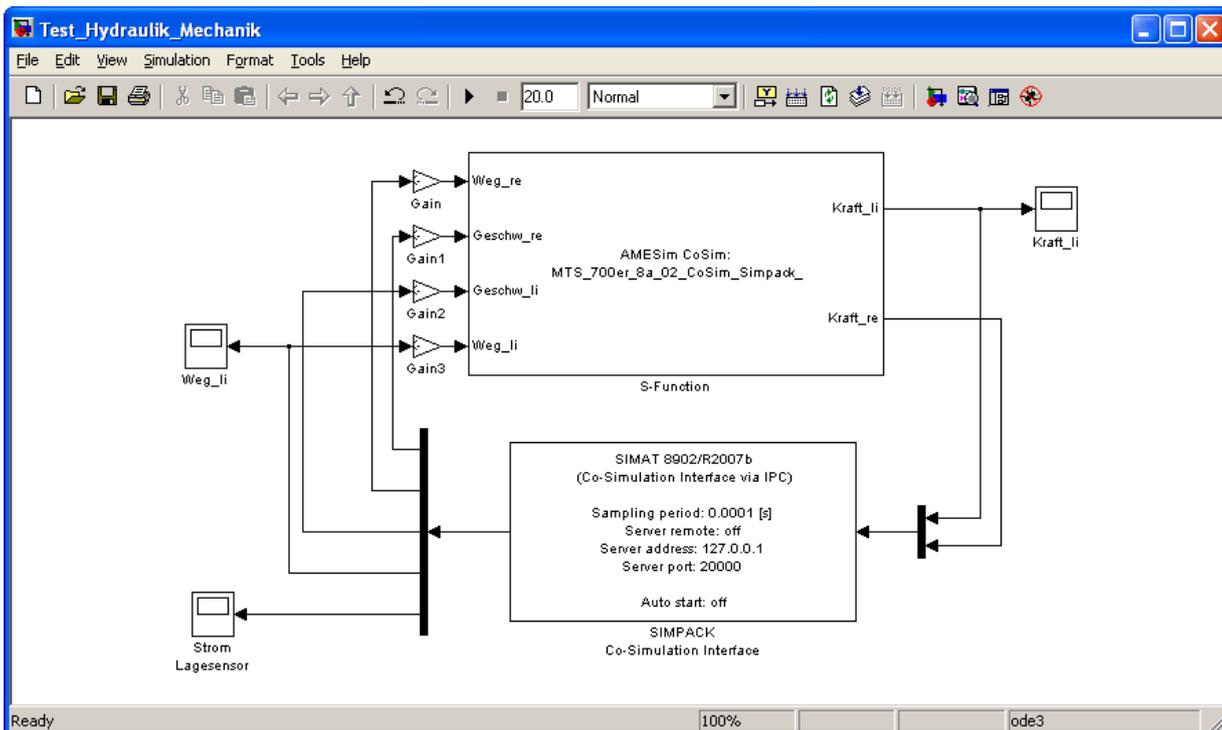


Abbildung 6.23 - Überprüfung der Kombination aus Mechanik und Hydraulik

Das hydraulische Modell übergibt die Kräfte der Federungszyylinder an das mechanische Modell und erhält hierfür Wege und Geschwindigkeiten der Komponenten zurück. Darüber hinaus gibt das mechanische Modell einen Stromwert der Lage aus, welcher für das Teilmodell der Regelung benötigt wird.

Nach mehreren Simulationsdurchläufen und Anpassungen an den Modellen konnte das mechanische Modell für die Gesamtsimulation freigegeben werden. Im Zuge dessen konnte auch das hydraulische Teilmodell der Vorderachsfederung für die Gesamtsimulation freigegeben werden, da die Funktionssicherheit in Kombination mit den jeweiligen Einzelmodellen nachgewiesen wurde. Für den Aufbau des Gesamtsimulationsmodells der Traktorvorderachsfederung musste jedoch ein weiteres Modell der Federung erstellt werden, welches nur noch die hydraulischen Komponenten sowie den Block mit den Übergabevariablen enthielt. Der Aufbau des Gesamtsimulationsmodells soll nachfolgend beschrieben werden.

6.3.6 Aufbau des Gesamtsimulationsmodells

Für die Erstellung des Gesamtsimulationsmodells wurden alle zuvor getesteten Teilmodelle in ein Simulink-File übertragen und anschließend die entsprechenden Ein- und Ausgabegrößen miteinander verbunden. **Abbildung 6.24** zeigt das Gesamtmodell der Traktorvorderachsfederung mit Niveauregulierung unter Simulink, die graphische Verknüpfung der einzelnen Modelle sowie auch Elemente zur Visualisierung (Scopes).

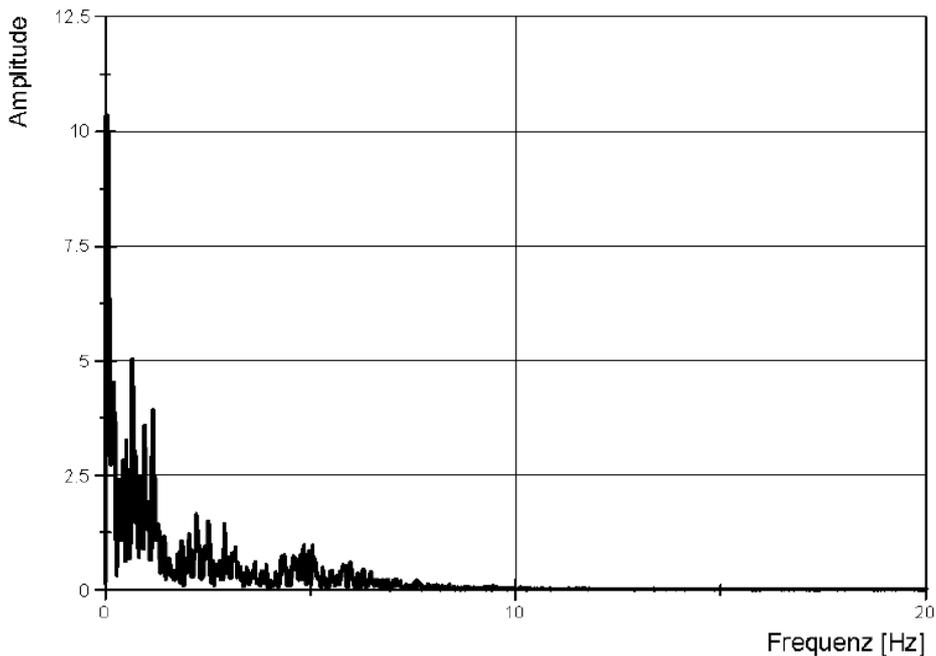


Abbildung 6.25 - FFT-Analyse des Wegprofils einer Prüfbahnfahrt nach ISO 5008

Nach der Ermittlung der Anregungsfrequenzen werden die Eigenwerte des Systems untersucht. Hierfür wird ein Modell benötigt, welches sowohl die mechanischen als auch die hydraulischen Komponenten sowie die Regelung beinhaltet. Dieses wurde von der Fa. HYDAC aufgebaut und ist oben bereits in Abbildung 6.18 dargestellt. Für die Simulation wird von einer nicht befüllten Vorderachsfederung ausgegangen. Aus diesem Grund sinkt zu Beginn der Simulation die Vorderachse aufgrund ihres Eigengewichtes in ihren Endanschlag. Mit Hilfe der Niveauregelung wird die Traktovorderachse an den oberen Rand des Toleranzbandes der Mittellage gebracht. Anschließend werden im Wechsel die Grenzen des Toleranzbandes angefahren. Ausgehend von der oberen Toleranzbandgrenze beginnt ein so genannter Sweep, ein Sinus mit sich ändernder Frequenz, mit einer Frequenz zwischen 0,5 Hz und 2,5 Hz sowie einer Amplitude zwischen 38 mm und 50 mm. Bei ca. 210 s wird ein Offset zunächst von 25 mm in ausfedernder Richtung und anschließend der gleiche Offsetbetrag in einfedernder Richtung aufgebracht. Es folgen stochastische Anregungen mit einer Amplitude von bis zu 84 mm. Abschließend werden im Wechsel noch einmal die Toleranzgrenzen angefahren. Das Ergebnis der Simulation zeigt **Abbildung 6.26**. Über der Zeit ist der Weg eines Federungszyllinders aufgetragen.

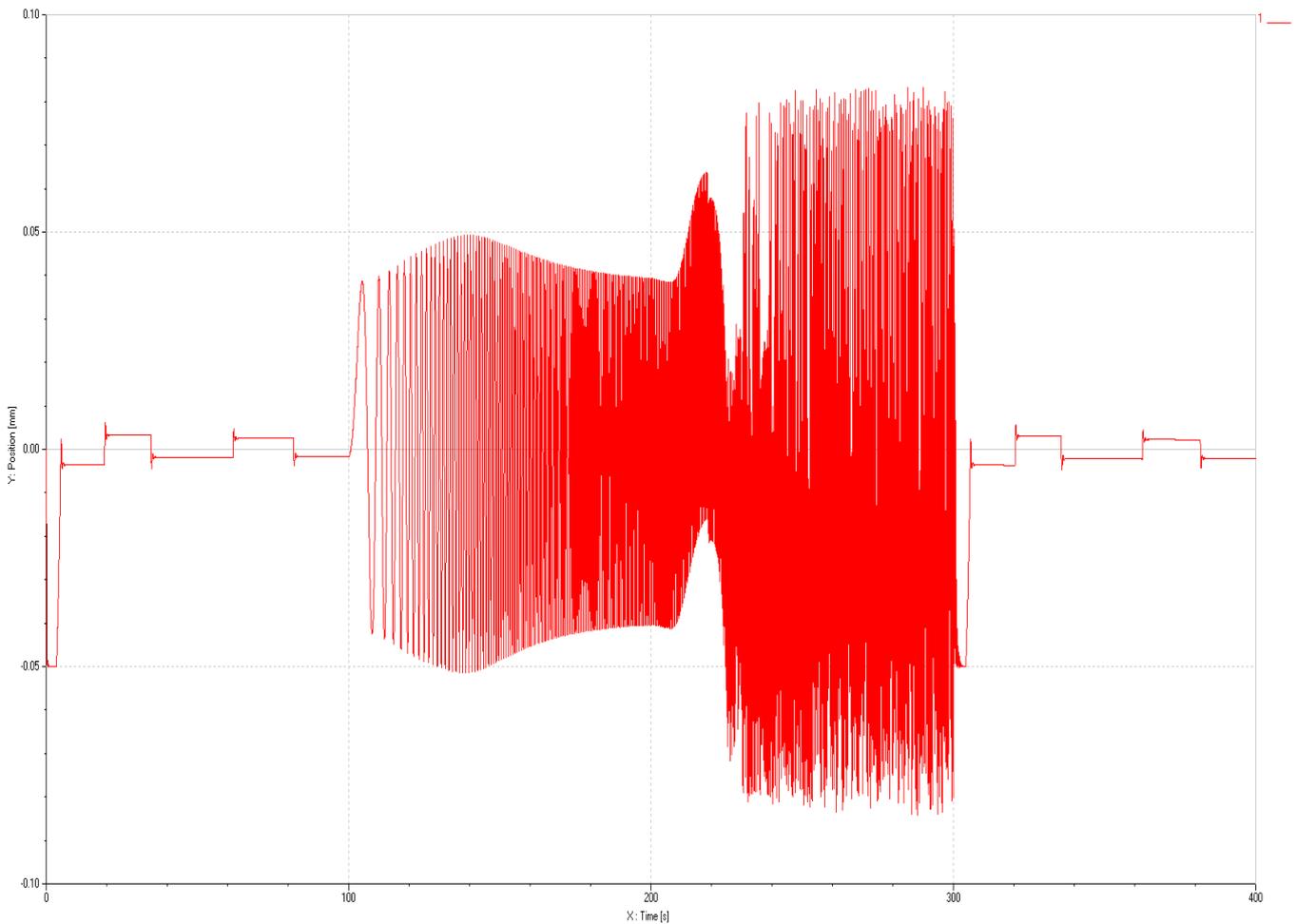


Abbildung 6.26 - Simulationsergebnis des Gesamtsimulationsmodells

Für die Ermittlung der Eigenwerte stellt AMESim ein Analysewerkzeug (Linear Analysis – Eigenvalues) zur Verfügung. Zur Nutzung dieses Analysewerkzeugs muss der Anwender vor Beginn der Simulation Linearisierungszeitpunkte festlegen. Zu diesen Zeitpunkten wird das Gesamtsystem linearisiert und die Eigenwerte werden berechnet. Das Ergebnis kann der Anwender entweder tabellarisch wie in **Abbildung 6.27** oder als Plot in der komplexen Zahlenebene einsehen.

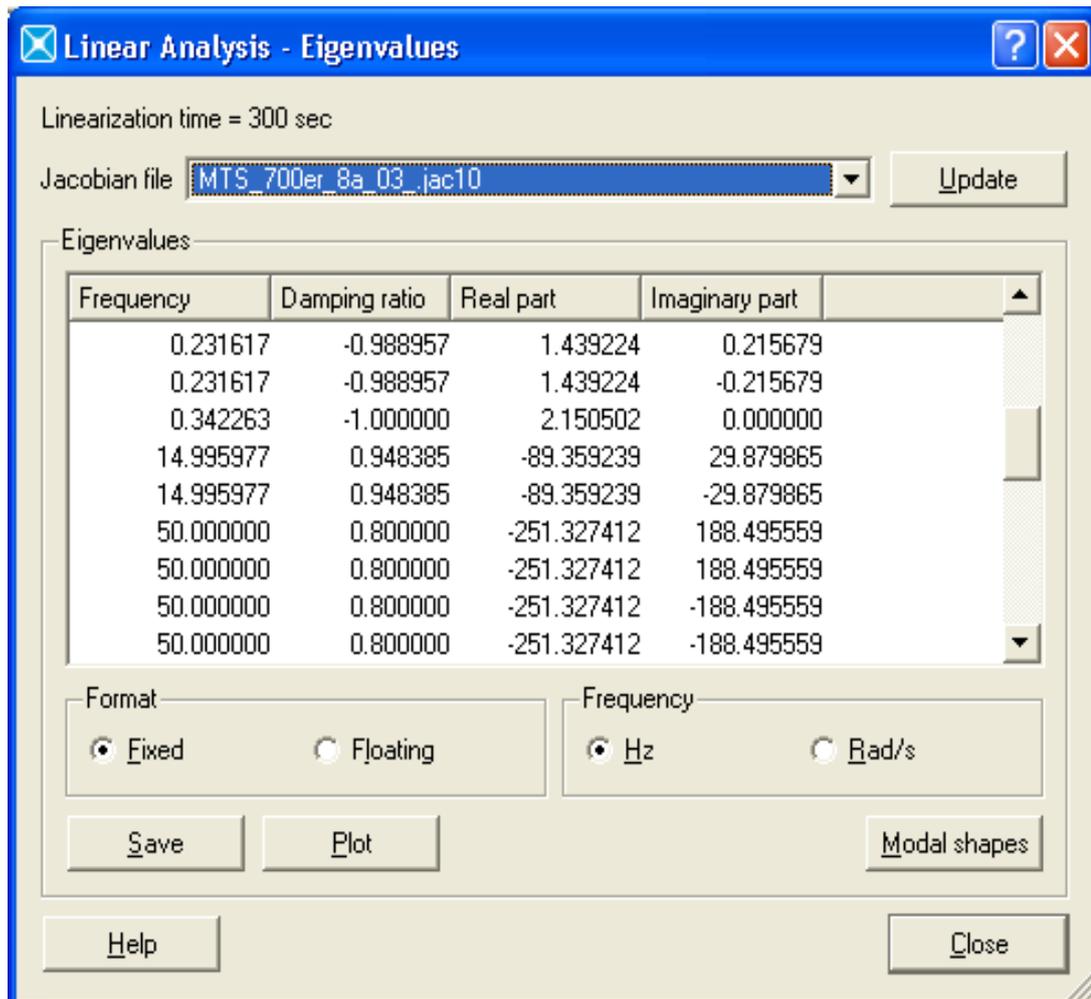


Abbildung 6.27 - Ergebnis der Eigenwertberechnung aus AMESim

Die Tabelle in **Abbildung 6.27** ist auf die wesentlichen Ergebnisse gekürzt. Es werden deutlich mehr Werte angezeigt.

Der Anwender muss unter all diesen Werten die Eigenwerte herausfiltern. Dabei kann er sich an den Imaginärteilen der einzelnen Ergebnisse orientieren. Besitzt ein Ergebnis keinen Imaginärteil, handelt es sich nicht um eine Eigenfrequenz. Die Eigenfrequenzen werden in der ersten Spalte der Tabelle aufgeführt. Die höchste im Modell auftretende Eigenfrequenz beträgt 50 Hz. Dies führt zu einer Zeitschrittweite von 20 ms. Nach der Regel

$$h = \frac{T_{\text{Klein}}}{10}$$

ergibt dies ein Kommunikationsintervall von 2 ms. Damit verhält sich das Modell grundsätzlich erwartungsgemäß. Allerdings sind die Reaktionszeiten der Regelung länger als erwartet. Dies hängt damit zusammen, dass der Niveauregler intern auf 1 ms getaktet ist und dadurch eine Verzögerung bei den Reaktionszeiten des Reglers auftritt, wenn nur alle 2 ms ein neuer Wert geliefert wird. Das heißt also, dass das Kommunikationsintervall für diese Simulation sich an der Taktzeit des Niveaureglers von 1 ms orientieren sollte, da in diesem System dies der kleinste Zeitschritt ist.

6.3.8 Ergebnisse der Simulationsläufe

Die Simulation wurde sowohl mit allen vier Programmen parallel offen, als auch mit DSH $plus$ im Embedded-Modus durchgeführt und verursachte dabei keine Probleme. Das Ergebnis eines Simulationslaufs zeigt **Abbildung 6.28**.

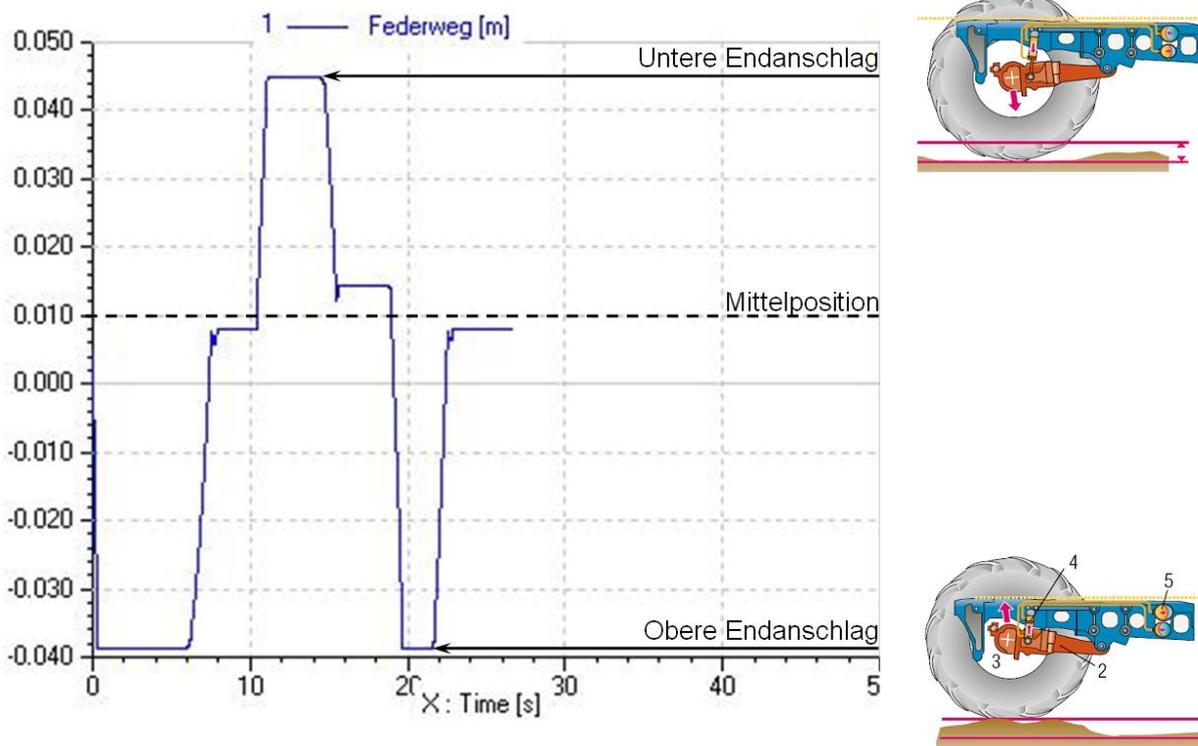


Abbildung 6.28 - Simulationsergebnis der Traktorvorderachsfederung mit Niveauregelung

Zu Beginn der Simulation befindet sich das mechanische Teilmodell in Mittelstellung der Federungszyylinder, dies entspricht der Auslenkung 0 m bei 0 s. Aufgrund der nicht befüllten Federung und des Eigengewichtes des Fahrzeugs sinkt die Federung in ihren oberen Endanschlag (siehe Prinzipskizze rechts unten in **Abbildung 6.28**), wobei ein Einfedern negatives und ein Ausfedern einen positives Wegsignal bedeutet. Nach ca. 2 s wird die Niveauregelung über den Taster im Regelungsmodell aktiviert. Die Regelung erkennt die Positionsabweichung von der Sollposition (Mittellage der Federung) und beginnt, die Abweichung auszuregulieren, indem die Pumpe auf Druck gefahren und das Hebenventil aktiviert wird. Nach Erreichen der in der Regelung hinterlegten Mittelposition nach ca. 7 s (diese weicht von der Nulllage ab) verbleibt die Federung in dieser Position. Die eigentliche Mittelposition wird aufgrund des Toleranzbandes des Reglers nicht erreicht. Daher ist in **Abbildung 6.28** eine Differenz zwischen IST-Position und Mittelposition zu erkennen.

Nach Erreichen der Mittelposition wird die Lageregelung deaktiviert und die Federung manuell ausgefahren, so dass sie nach 11,5 s den unteren Endanschlag erreicht. Dies entspricht einer voll ausgefederten Federung. Bei 12,5 s wird die Lageregelung reaktiviert, wodurch bei 14 s eine Einfederung des Fahrzeugs beginnt. Diese Einfederung stoppt oberhalb der Mittelposition, da wiederum das Toleranzband des Reglers erreicht ist.

Bei 18 s wird die Lageregelung deaktiviert. Das Fahrzeug sinkt erneut in den oberen Endanschlag. Nach Erreichen dieser Endlage wird die Lageregelung reaktiviert und die Federung kehrt bei 22 s in ihre Mittelposition zurück.

Wie das Ergebnis dieser ersten Simulationsläufe zeigt, existiert ein funktionierendes Gesamtmodell der Traktorvorderachsfederung. Die Simulationsergebnisse erscheinen plausibel. In den nächsten Schritten muss nun untersucht werden, inwieweit die Simulationsergebnisse die reale Traktorvorderachsfederung wiedergeben. Hierzu soll der Endzustand, wie er in Abbildung 6.28 dargestellt ist, abgespeichert und als Anfangszustand für nachfolgende Simulationen verwendet werden. Für den Vergleich mit Prüfstandsmessungen der realen Vorderachse soll in der Simulation eine sinusförmige Weganregung aufgebracht werden. Die Weganregung soll eine Amplitude von 20 mm haben und um drei Lagen aufgebracht werden (Mittellage, Mittellage + 25 mm, Mittellage - 25 mm). Die Weganregung wird im mechanischen Teilmodell auf das System aufgebracht. Der Vergleich der Messergebnisse mit der Simulation erfolgt anschließend in einer Software zur Messdatenauswertung (DIADEM der Firma NI).

Die Niveauregelung der Traktorvorderachsfederung stellt im Betrieb des Traktors sicher, dass das Fahrzeug nach Möglichkeit immer den vollen Federweg zur Verfügung hat, unabhängig des Beladungszustandes des Fahrzeugs. Diese Beladungszustände äußern sich in unterschiedlichen Drücken auf der Kolbenseite der Federungszyylinder. Um diese Beladungszustände in der Simulation abzubilden, müssen die Radlasten im mechanischen Teilmodell angepasst werden. Nach einer Anpassung der Radlasten muss die Simulation wie bereits beschrieben gestartet werden, d.h. dass zunächst mit einer drucklosen Federung simuliert wird. Dadurch werden die an die neuen Randbedingungen (Radlasten) angepassten Startwerte der Federung erzeugt, mit denen anschließend die sinusförmige Weganregung gerechnet werden kann.

Die Verbesserung der Simulationsmodelle ist ein wichtiger Bestandteil dieses Arbeitspaketes. Der weitaus wichtigere Bestandteil ist jedoch die Co-Simulation und die unternehmensübergreifende Zusammenarbeit bei der Co-Simulation. Auf Basis der an diesem Gesamtmodell gewonnenen Erkenntnisse sollen Maßnahmen für die standardisierte Vorgehensweise abgeleitet werden. Dies bezieht sich zum einen auf die Modellerstellung, die Handhabung der Modelle auf der Plattform sowie die Handhabung der beteiligten Simulationsprogramme. Aus diesem Grund sollen die erarbeiteten Erfahrungen aus diesem Simulationsmodell nachfolgend beschrieben werden.

6.4 Belastungen am Realbauteil

Der Prüfstand zur Untersuchung der Traktorvorderachsfederung erfüllt drei grundsätzliche Funktionen:

1. Positionierung und Fixierung des zu untersuchenden sowie des anregenden Systems und der Sensorik auf dem am Lehrstuhl vorhandenen Prüffeld
2. Anregung der Traktorvorderachsfederung
3. Messung und Speicherung der Sensorsignale

Nachfolgend werden der Prüfstand, die Anregung sowie die Messung und Speicherung der Sensorsignale erläutert.

6.4.1 Aufbau des Prüfstands

Zur Durchführung von Messversuchen wurde ein Prüfstand wie in **Abbildung 6.12** konstruiert, der eine Anregung der Traktorvorderachsfederung über zwei hydraulische Pulszylinder ermöglicht, um daraus die Systemreaktion der Federung zu ermitteln. Die Anregung erfolgt im Fahrzeug entweder durch die Fahrbahn oder durch Anbaugeräte oder durch eine Kombination aus beidem. Dabei wird die Anregung über die Reifen und das Felgenlochbild auf die Achse übertragen.

Zur Durchführung der Messversuche wurde der vordere Halbrahmen über Aufspannmöglichkeiten im vorderen und hinteren Bereich fest mit dem Prüffeld verbunden, so dass sich die Achse frei zum Rahmen bewegen konnte. Die beiden Pulszylinder wurden je an einem Portal hängend direkt über der Achse positioniert. Die Anbindung der Pulszylinder an die Vorderachse erfolgte über ein selbst entwickeltes Anschlusselement und geschah über das Felgenlochbild, an dem normalerweise die beiden Vorderräder montiert werden. **Abbildung 6.29** zeigt ein CAD-Bild des Prüfstands, wie er am Lehrstuhl aufgebaut wurde.

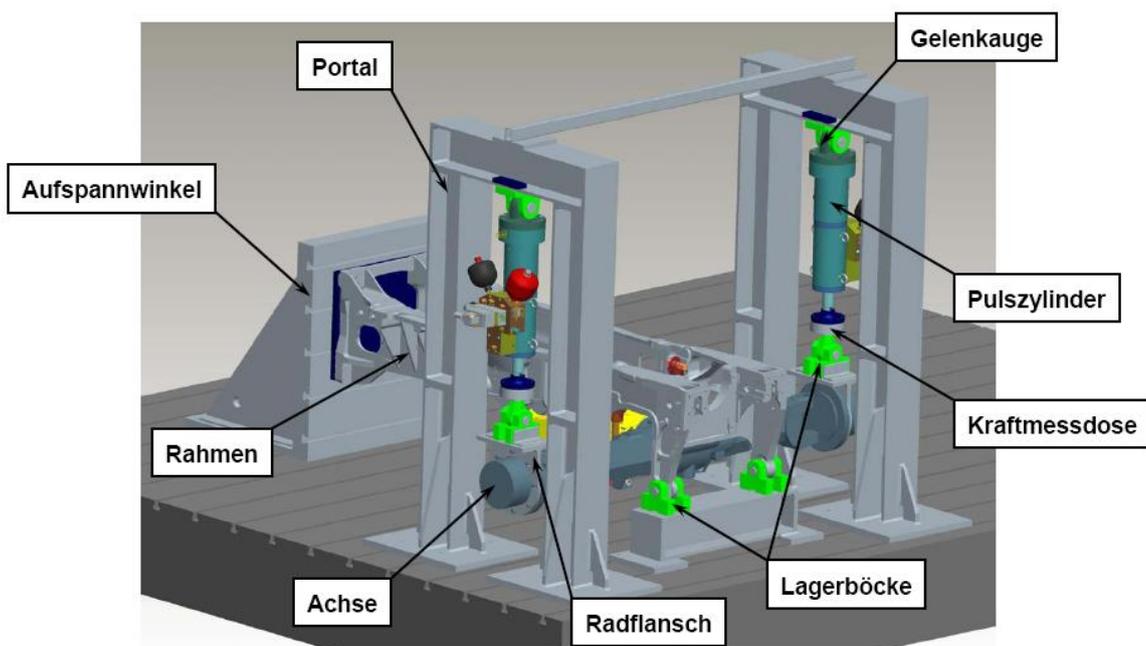


Abbildung 6.29 - CAD-Darstellung des Prüfstands

Die eigentliche Vorderachsfederung besteht aus den Federungszyclindern, den zugehörigen Leitungen und Hydraulikspeichern. **Abbildung 6.30** zeigt die Komponenten im Hydraulikschaltplan, **Abbildung 6.31** als CAD-Darstellung. Die Federung ist Teil des Hydrauliksystems des Fahrzeugs. **Abbildung 6.30** ist daher nur ein Ausschnitt aus dem Gesamtsystem.

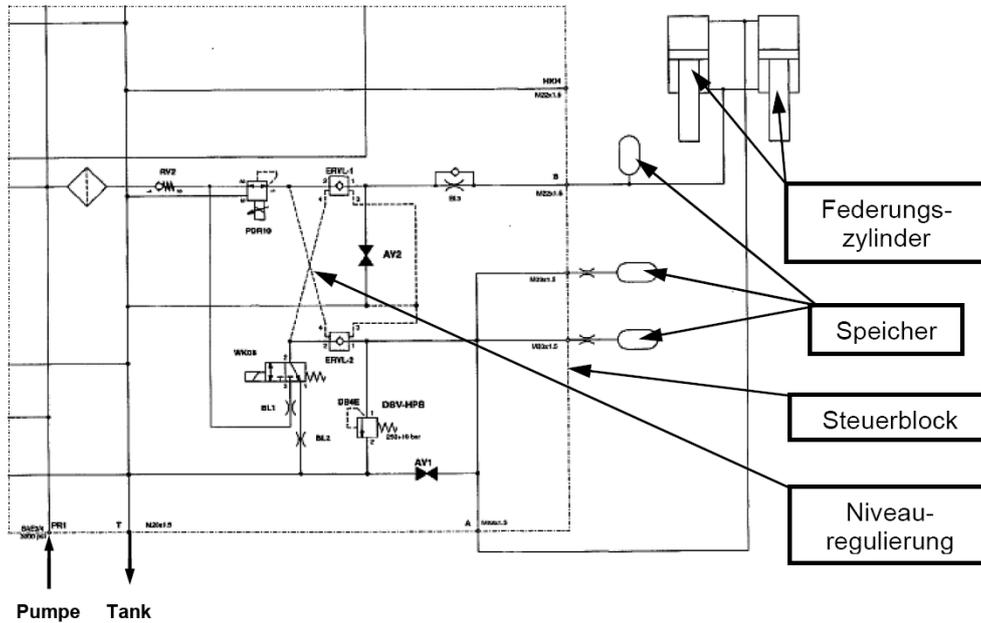


Abbildung 6.30 - Darstellung des Hydraulikkreislaufs mit Niveauregulierung

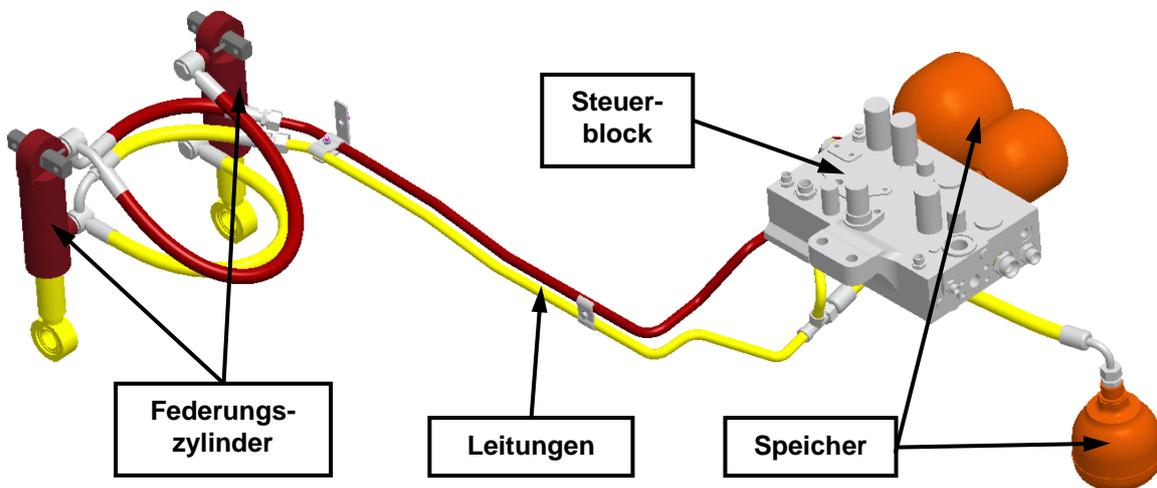


Abbildung 6.31 - CAD-Darstellung der Vorderachsfederungskomponenten

Zentrales Element dieses Hydrauliksystems ist ein Steuerblock (grau), an den u.a. die Federungszyclinder (dunkelrot + gelb) angeschlossen sind bzw. der als Träger für die Hydraulikspeicher (orange) dient. Ein solcher Steuerblock wurde auch am Prüfstand verbaut. Die Funktionen des Steuerblocks, welche nicht die Vorderachsfederung betrafen, wurden deaktiviert; auch die automatische Niveauregulierung wurde außer Kraft gesetzt. Das System Vorderachsfederung wurde in den ersten Messreihen als abgeschlossenes System betrachtet. **Abbildung 6.32**

zeigt ein Bild des Prüfstands in der Draufsicht. Auf diesem Bild ist auch eine Vorrichtung zur Befüllung der Vorderachsfederung über eine Handpumpe zu sehen.

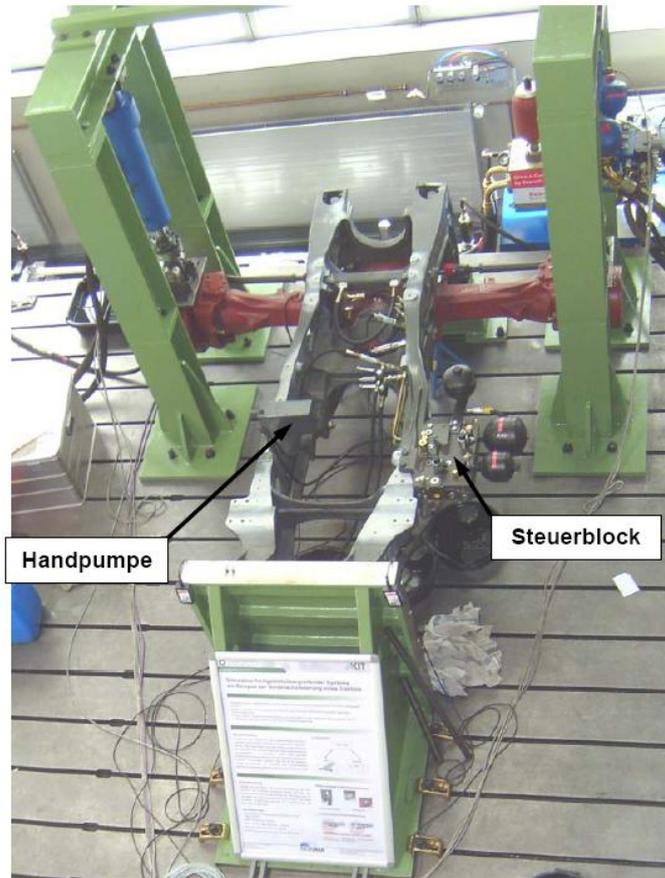


Abbildung 6.32 - Draufsicht der Vorderachsfederungsprüfstands

6.4.2 Anregung der Vorderachsfederung und verbaute Sensorik

Die Anregung am Prüfstand erfolgt durch so genannte Pulszylinder, die an der Vorderachse angreifen. Zunächst wurde im Projekt von einer Sinusförmigen Anregung ausgegangen. Die Frequenzen lagen im Bereich zwischen 0,5 und 2,5 Hz, bei reduziertem Hub auch bis 5 Hz. Als maximale Amplitude für den Hub wurden 20 mm angesetzt. Darüber hinaus sollten drei Lagen der Federung gemessen werden, die Nulllage und dann ein Offset von +/-25 mm aus der Nulllage. Diese Anregung entspricht dem Prüfzyklus, wie er auch bei der Firma AGCO gefahren wird. Für die Erzeugung der Messwerte wurde die Vorderachsfederung entsprechend der realen Beladungs-/Belastungszustände vorgespannt. Die Vorspanndrücke waren der nachfolgenden **Abbildung 6.33** entnommen. Die Abbildung enthält darüber hinaus noch Angaben zu den Vorspanndrücken der Speicher sowie zu den geometrischen Größen der Federung.

Berechnung 700Com3

2 Zyl 40 x 50 ASP 1x 1,4 ltr+ 1x 0,75 ltr P-Null 45 bar, VSP 1x0,75 ltr P-Null 80 bar ringseitiger Systemdruck 100 bzw. 200 bar

Systemdaten

Drücke / Kappawerte		Federwege		zu federnde Lasten (Mittellage)	
Systemdruck	200 bar	Federweg an der Achse:	+/- 50,0 mm	minimal	650 daN
EXE (Kappa)	1,3	Zylinderhub:	+/- 39,2 mm	leere Maschine	1750 daN
1/Kappa	0,7892308	Hebelarm Achse	750 mm	maximal n. LH	3350 daN
EXL	1	Hebelarm Zylinder	587 mm	ungefederte Masse	650 kg
		Übersetzung i:	1,2768	FL-Achslast leer	3750 daN
DW- Zylinder		Vorspann-Speicher stangenseitig		Arbeitsspeicher kolbenseitig	
Stangen- Durchmesser	40 mm	Volumen V _s	750 cm ³	Volumen 1.ASP	1400 cm ³
Kolben- Durchmesser	50 mm	Vorspanndruck	80 bar	Volumen 2. ASP	750 cm ³
Anzahl Zylinder	2 Stück	zul. max. Druck	250 bar	Zusatz-Speicher Summe	0 cm ³
Kolbenfläche (Summe)	39,27 cm ²	zul min Druck	88 bar	Summe ASP-Vol	2150 cm ³
Stangenfläche (Summe)	25,13 cm ²	P1Z Nulllage	100 bar	Vorsp-Druck ASP	45 bar
Ringfläche(Summe)	14,14 cm ²	V1Z Nulllage	600,0 cm ³	P2 _{ASP} Nulllage	bar
				V2 _{ASP} Nulllage	cm ³

Kolbenseitiger Speicher

P _{max} -stat	200,00 bar
P _{max} -dyn	373,53 bar
P _{min} -dyn	49,94 bar
i	7,48

ringseitiger Speicher

P _{max} -stat	100,00 bar
P _{max} -dyn	118,48 bar
P _{min} -dyn	87,24 bar
i	1,34

Minimal

Wege		stangenseitiger Speicher				Arbeitsspeicher kolbenseitig			Kräfte		
Federweg [mm]	Zylinderweg [mm]	Diff-VolSt [cm3]	V2Z [cm3]	P2Z [bar]	FRing [daN]	Diff-VolKol [cm3]	V2Asp [cm3]	P2Asp [bar]	FZylres [daN]	FAchse res [daN]	Achslast [daN]
0	0	0,00	600,00	100,0	1414	0,00	1693,49	57,13	830	650	1300

Leere Maschine

Wege		stangenseitiger Speicher				Arbeitsspeicher kolbenseitig			Kräfte		
Federweg [mm]	Zylinderweg [mm]	Diff-VolSt [cm3]	V2Z [cm3]	P2Z [bar]	FRing [daN]	Diff-VolKol [cm3]	V2Asp [cm3]	P2Asp [bar]	FZylres [daN]	FAchse res [daN]	Achslast [daN]
0	0	0,00	600,00	100,0	1414	0,00	1041,56	92,89	2234	1750	2400

Maximal 180bar

Wege		stangenseitiger Speicher				Arbeitsspeicher kolbenseitig			Kräfte		
Federweg [mm]	Zylinderweg [mm]	Diff-VolSt [cm3]	V2Z [cm3]	P2Z [bar]	FRing [daN]	Diff-VolKol [cm3]	V2Asp [cm3]	P2Asp [bar]	FZylres [daN]	FAchse res [daN]	Achslast [daN]
0	0	0,00	600,00	100,0	1414	0,00	612,71	157,91	4787	3750	4400

Maximal 200bar

Wege		stangenseitiger Speicher				Arbeitsspeicher kolbenseitig			Kräfte		
Federweg [mm]	Zylinderweg [mm]	Diff-VolSt [cm3]	V2Z [cm3]	P2Z [bar]	FRing [daN]	Diff-VolKol [cm3]	V2Asp [cm3]	P2Asp [bar]	FZylres [daN]	FAchse res [daN]	Achslast [daN]
0	0	0,00	600,00	100,0	1414	0,00	483,75	200,00	6440	5045	5695

Abbildung 6.33: Systemkonfigurationen für Messversuche

Zur Erfassung der Messdaten sind am Prüfstand Weg-, Kraft- und Drucksensoren verbaut. Der Weg wird induktiv in den Pulszylindern erfasst. Die Kraft wird zwischen den Pulszylindern und der Vorderachse über Kraftmessdosen gemessen. Die Drucksensoren wurden sowohl auf Kolben- wie auch auf Stangenseite im Kreislauf der Federungs- zylinder platziert. Die Messdaten werden mit Hilfe der Software LabView (Version 8.5.1) erfasst. Hierfür wurde eine Messdatenerfassung programmiert, welche die analogen Messdaten erfasst, skaliert und in Dateien abspeichert.

Die LabView-Oberfläche bot darüber hinaus die Möglichkeit, Messdaten online zu visualisieren. Der Prozess der Messdatenerfassung und Verarbeitung wird nachfolgend beschrieben.

6.4.3 Messdatenerfassung am Prüfstand und Verarbeitung

Zur Erfassung der Messwerte wurde die in **Abbildung 6.34** dargestellte Messkette aufgebaut. Die Signale der Weg- sowie Kraftsensoren werden in eine Steuerungs- und Regelungseinheit, die so genannte HCE, gespeist. Die HCE steuert und regelt die Bewegung der Pulszylinder und benötigt für die Betriebsmodi Kraft- bzw. Wegregelung die entsprechenden Sensorsignale. Die Signale stehen am Frontpanel der HCE an BNC-Abschlüssen zur Verfügung und können dort abgegriffen werden. Gleichzeitig erhalten die Sensoren von der HCE ihre Speisespannung.

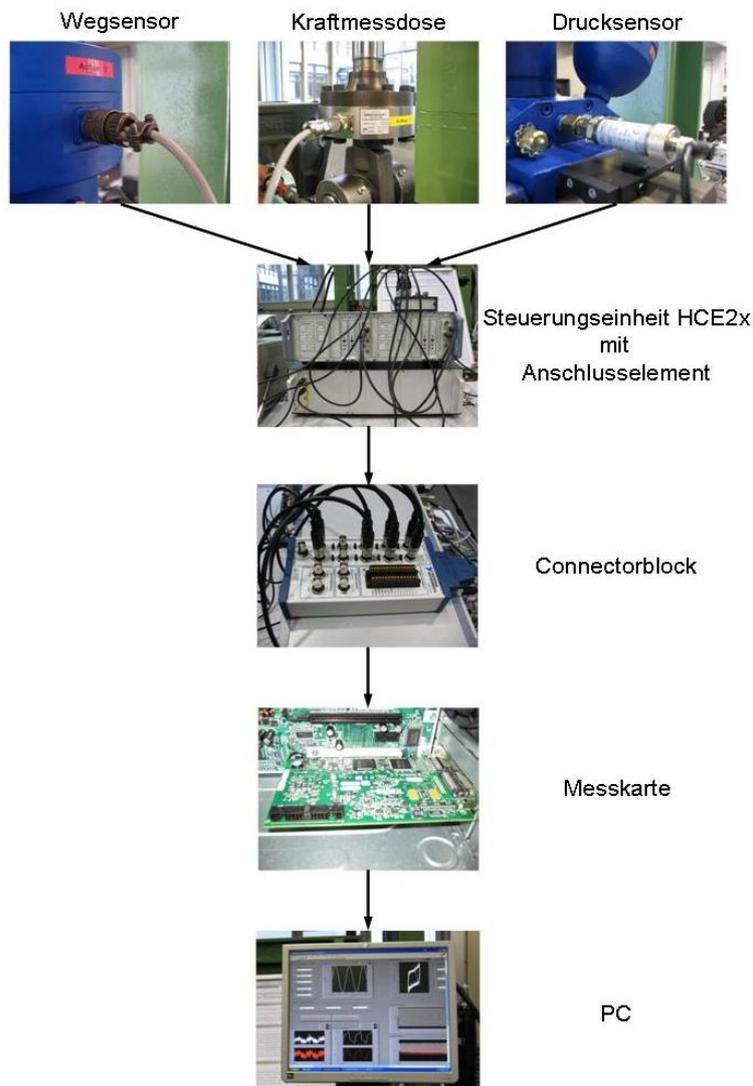


Abbildung 6.34: Darstellung der Messkette

Die Drucksensoren mussten ebenfalls mit einer Speisespannung versorgt und gleichzeitig die Messsignale ausgelesen werden. Da eine direkte Anbindung an die HCE nicht vorgesehen war, wurde ein spezielles Anschlusselement für die Drucksensoren konzipiert. Das Anschlusselement versorgt die Sensoren mit einer 24 V Gleichspannung.

nung. Die Messsignale werden durchgeschleust und stehen am Frontpanel des Anschlusselements an BNC-Anschlüssen zur Verfügung

Die an den Frontpanels der HCE und des Anschlusselements zur Verfügung stehenden Messsignale werden mittels BNC-Kabel in einen so genannten Connectorblock der Firma National Instruments NI geleitet. Von dort werden die Messdaten über ein Anschlusskabel in eine Messkarte der Firma NI übertragen.

Das Einlesen von der Messkarte und Abspeichern der Messdaten erfolgt mit der Software LabView der Firma NI. LabView ist eine grafische Benutzeroberfläche, mit der Mess-, Steuer- und Regelhardware an einen PC angeschlossen werden können. Mit Hilfe der grafischen Programmierung können ankommende Messsignale erfasst und in Dateien abgespeichert werden. Die Abtastrate der abzutastenden Kanäle kann eingestellt und eine Skalierung der Messdaten vorgenommen werden. **Abbildung 6.35** zeigt ein Beispiel einer solchen Programmierung.

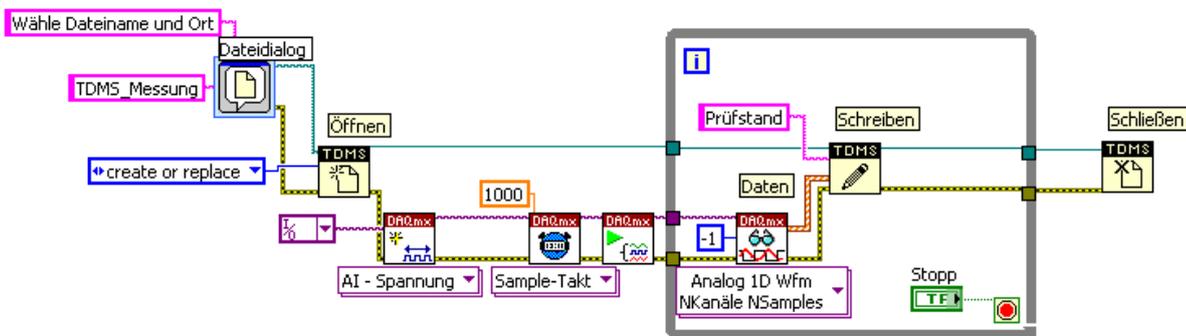


Abbildung 6.35 - Messdatenerfassung mit LabView

LabView bietet neben der Datenerfassung und –bearbeitung die Möglichkeit, online die Messwerte zu visualisieren. Die **Abbildung 6.36** und **Abbildung 6.37** sind zwei Beispiele einer solchen Online-Visualisierung. In **Abbildung 6.36** sind in rot und weiß je das Kraftsignal einer Kraftmessdose dargestellt. Das Signal ist über der Zeit aufgetragen. Es wurde mit einer Frequenz von 1 kHz abgetastet. Im rechten Teil der **Abbildung** sind ebenfalls die beiden Kraftsignale zu sehen. Allerdings wurden hier die Eingangswerte durch eine Mittelwertbildung geglättet. Hierfür wurden in jedem Abtastschritt 30 Werte links und rechts der Abtaststelle zur Mittelwertbildung herangezogen.

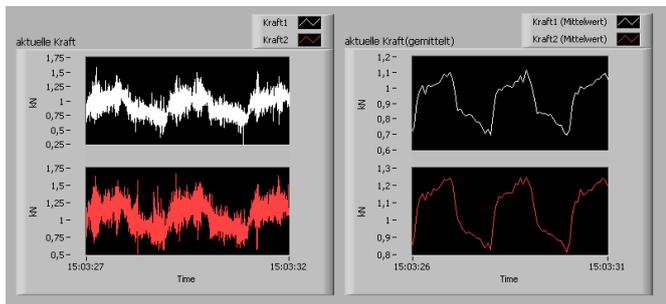


Abbildung 6.36 - Kraftsignal orig. und gemittelt

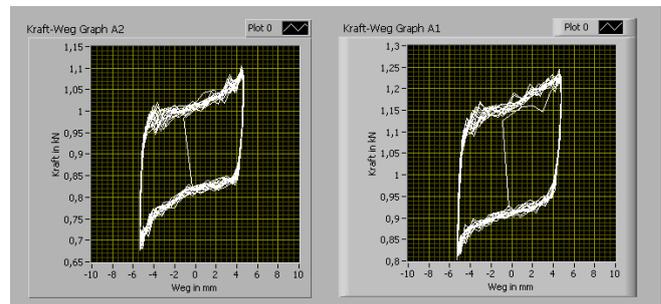


Abbildung 6.37 - Kraft-Weg-Hysterese

In **Abbildung 6.37** ist ein Kraft-Weg-Diagramm dargestellt. Dabei wurde das Signal der Kraftmessdosen über dem Signal der Wegsensoren sowohl für die linke wie auch für die rechte Seite des Traktors aufgetragen. Mit Hilfe dieser Darstellung lässt sich das charakteristische Verhalten der Federung zeigen. Letztlich stellt das Diagramm die Steifigkeit der Federung dar.

Neben der Erfassung und Visualisierung der Messwerte ermöglicht die LabView-Programmierung eine Sollwertgenerierung. Dieser Sollwert wird für die Steuerung der Pulszylinder benötigt. Gemäß dem im Kapitel 6.5.2 beschriebenen Prüfprogramm erzeugt der Sollwertgenerator ein Signal, dessen Signalform, Frequenz, Amplitude und Offset über eine grafische Oberfläche einstellbar ist. Die grafische Benutzeroberfläche ist in **Abbildung 6.38** dargestellt. Im linken Bereich können die Grenzen des Sollsignals, seine Abtastrate und die Rate der Speicherwerte eingestellt werden. Im rechten Bildbereich ist das Signal selbst zu sehen. Im unteren Bereich sind die Einstellmöglichkeiten bezüglich Signalform, Frequenz, Amplitude und Offset dargestellt.

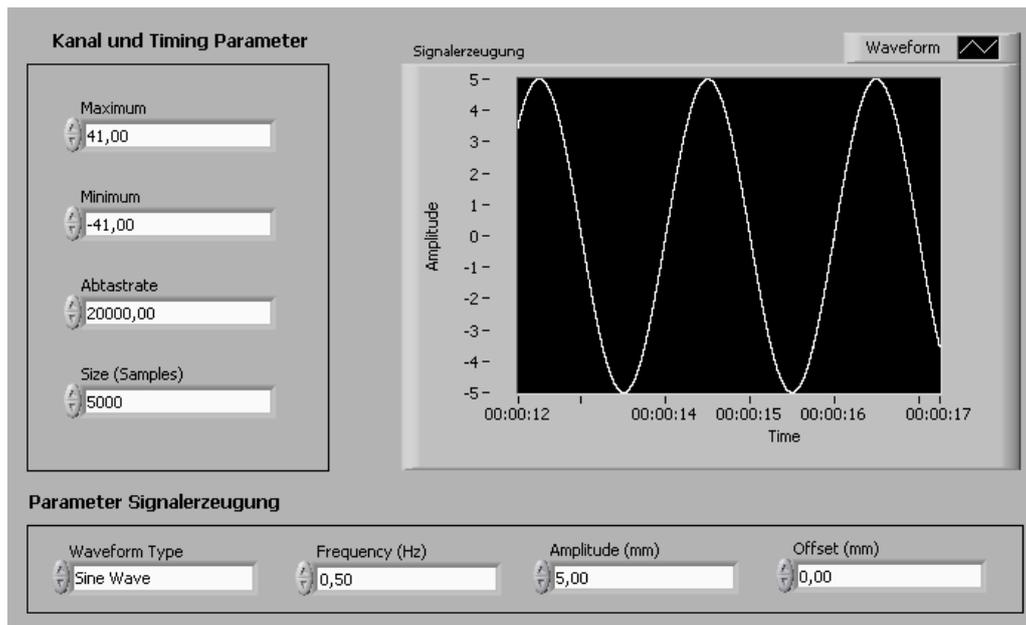


Abbildung 6.38 - Sollwertgenerierung in LabView

Die Auswertung der Messdaten erfolgte mit der Software DIADEM (Version 10.2) der Firma NI. DIADEM bietet die Möglichkeit, Daten zu verwalten, zu analysieren sowie einen Bericht aus den Daten zu erstellen. Bei der Datenanalyse lassen sich Filter- und Glättungsfunktionen verwenden, die die Messdaten grafisch aufwerten. Dies ist notwendig, da die Ausgangsdaten der Messungen prinzipbedingt (Sensor mit analogem Signal) mit einem Rauschen behaftet sind.

Abbildung 6.39 zeigt das Ergebnis einer Messung. Simuliert wurde bei dieser Messung ein leeres Fahrzeug in der Nulllage der Federung. Als Anregung wurden eine Frequenz von 0,5 Hz (linke Darstellung) und 2,5 Hz (rechte Darstellung) und eine Amplitude von 20 mm gefahren.

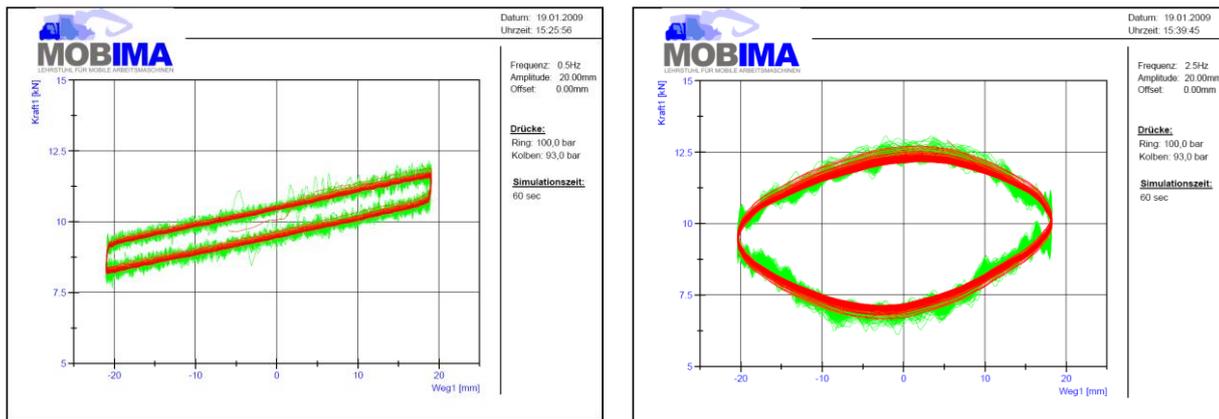


Abbildung 6.39 - Geglättetes (rote Kurve) und ungeglättetes Messsignal (grüne Kurve) bei 0,5 Hz (linkes Diagramm) und 2,5 Hz Anregung (rechtes Diagramm)

Dargestellt sind in den Diagrammen auf der x-Achse das Wegsignal und auf der y-Achse das Kraftsignal. Durch das Auftragen der Kraft über dem Weg kann in den Diagrammen die Steifigkeit der Federung abgelesen werden. Die Reaktion der Federung auf die Anregung zeigt deutlich eine Hysterese. Die Hysterese verstärkt sich bei zunehmenden Frequenzen. Dies ist auf den Einfluss der Dämpfungskraft zurückzuführen, die aufgrund der höheren Fließgeschwindigkeiten des Öls erzeugt wird. Die Federung wird somit steifer, je schneller die Anregung erfolgt. Dies ist eine erwünschte Eigenschaft.

Die Messdaten wurden sowohl als Rohdaten (grüne Kurve) als auch geglättet (rote Kurve) dargestellt. Geglättet wurde in diesem Fall durch Mittelwertbildung. Hierzu wird ein Messwert als Ausgangspunkt ausgewählt. Ausgehend von diesem Punkt werden 30 Messwerte links und rechts dieser Stelle betrachtet und der Mittelwert über diese Messwerte gebildet. Das Ergebnis ist eine geglättete Kurve.

Der Vergleich der Messdaten mit den Simulationsergebnissen wird ebenfalls in DIADEM erfolgen.

6.5 Abgleich Prüfstand und Simulation

Im Folgenden werden die oben generierten Messergebnisse mit den Simulationsergebnissen verglichen. Dies bezieht sich insbesondere auf das Hydraulik Modell der hydropneumatischen Federung. Hierbei werden insbesondere die Verlustquellen der Reibung und der Leitungs- und Speichermodellierung untersucht. Im Anschluss findet der Abgleich des Gesamtsystems mit der Simulation statt.

6.5.1 Abgleich des Teilmodells des hydropneumatischen Federungskreises in AMESim

Bedeutung der Reibung für die Simulation

Von der Firma Fendt wurden zwei neue, unbenutzte Federungszyylinder zur Verfügung gestellt. Diese Zylinder wurden vermessen und bewertet. Diese Versuchsreihe soll klären, wie sich die Eigenschaften der Zylinder, bezüglich der Reibung, bei Alterung ändern. Ein weiterer Aspekt ist es die korrekten Reibwerte zu erhalten, um diese in die Simulation einfließen zu lassen und somit das Modell besser abstimmen zu können. Die Ergebnisse der Reibungsmessung sind im Anhang 9.4 dargestellt.

Das Simulationsmodell wurde mit diesen neuen Erkenntnissen erweitert, um zu prüfen, ob sich der benötigte Rechenaufwand signifikant ändert. In **Abbildung 9.44** sind typische Werte der Simulationen gezeigt. Bei etwas genauerer Betrachtung derselbigen liegt es nahe, das Simulationsmodell möglichst zu vereinfachen, um den Übergang zum Echtzeitmodell zu ermöglichen.

Bei der Vereinfachung muss im Gegenzug damit gerechnet werden, dass die Genauigkeit der Ergebnisse kritisch zu betrachten ist. Dies heißt im Umkehrschluss, dass die Modelle jeweils nur für einen Betriebspunkt abgeglichen werden können bzw. in engen Grenzen ihre Gültigkeit haben.

Zu diesem Zeitpunkt ist in dem Simulationsmodell ein fester Reibwert für die Federungszyylinder hinterlegt. Für die Haftreibung wurden 370N und für die Gleitreibung wurden 360N pro Zylinder gewählt. In dem Reibmodell wurden keine besonderen Modellschaltungen hinterlegt, d.h. dass das Modell nur zwischen Haftreibung und Gleitreibung schaltet.

Einfluss unterschiedlicher Speicher- bzw. Leitungsmodelle

Beim Speicher- bzw. Leitungsmodell war die Vorgehensweise und Simulationsrandbedingung identisch mit der des Reibmodells. Es wurden im Wesentlichen vier Modelle verwendet. Bei den Speichern wurde unterschieden zwischen Speichern mit Idealgasgleichung und Speichern mit einer Realgasgleichung. Bei den Leitungen wurde ein vollständiges Leitungsmodell mit einer komplexen Wellen- und Schallgleichung hinterlegt sowie dessen Ersatzmodell, welches die Verluste über eine Reihe von Blenden, siehe **Abbildung 6.40**, darstellte.

Die unter diesen Bedingungen resultierten Rechenzeiten ergaben sich durch die Berechnung auf einem Simulations-PC. Die Zeiten sollen als qualitative Bewertungsgrundlage dienen.

Leitungsmodell mit Realgas-Speichern \Leftrightarrow 44 min 5 sek.

Leitungsmodell mit Idealgas-Speichern \Leftrightarrow 40 min 13 sek.

Blendenmodell mit Realgas-Speichern \Leftrightarrow 1 min 56 sek.

Blendenmodell mit Idealgas-Speichern \Leftrightarrow 1 min 20 sek.

Aus den Ergebnissen konnte abgeleitet werden, dass der Berechnungsfehler bei einem einzelnen Betriebspunkt des abgestimmten Simulationsmodells mit Blendenersatzmodell und Realgasgleichung vernachlässigbar ist. Der Einfluss auf die aufgewendete Berechnungszeit ist erheblich gegenüber einem vollständigen Leitungsmodell. Die Realgas- bzw. Idealgasgleichung haben einen nur untergeordneten Stellenwert, da bei beiden Gleichungen jeweils ein analytischer Ansatz zugrunde liegt.

In das jetzige Simulationsmodell sind deshalb ein Blendenersatzmodell und Speicher mit Realgasgleichung implementiert worden.

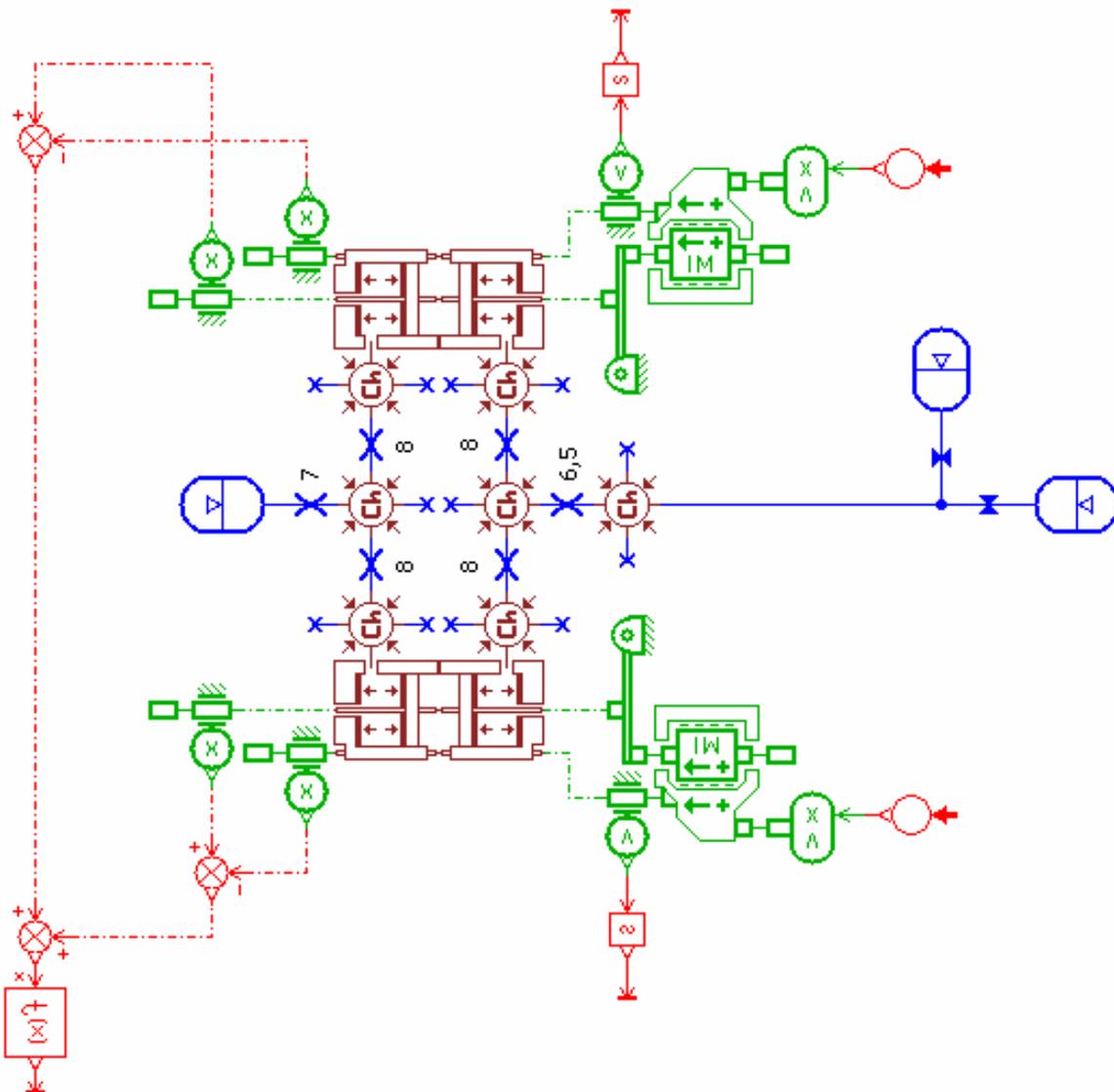


Abbildung 6.40 - Leitungsmodell als Blenden abgebildet

6.5.2 Abgleich des Gesamtmodells

Für die Abstimmung des Simulationsmodells wurde eine reale Wegmessung eines Federungszyinders zugrunde gelegt. Dieses Signal wurde ungefiltert in die Simulation eingelesen und verarbeitet.

Die Signale stammen aus einer Messung, bei der das Fahrzeug an der Vorderachse mit 4000kg belastet wurde. Hierbei handelt es sich um eine Feldwegfahrt bei einer Geschwindigkeit von 40km/h (siehe **Abbildung 6.41**).

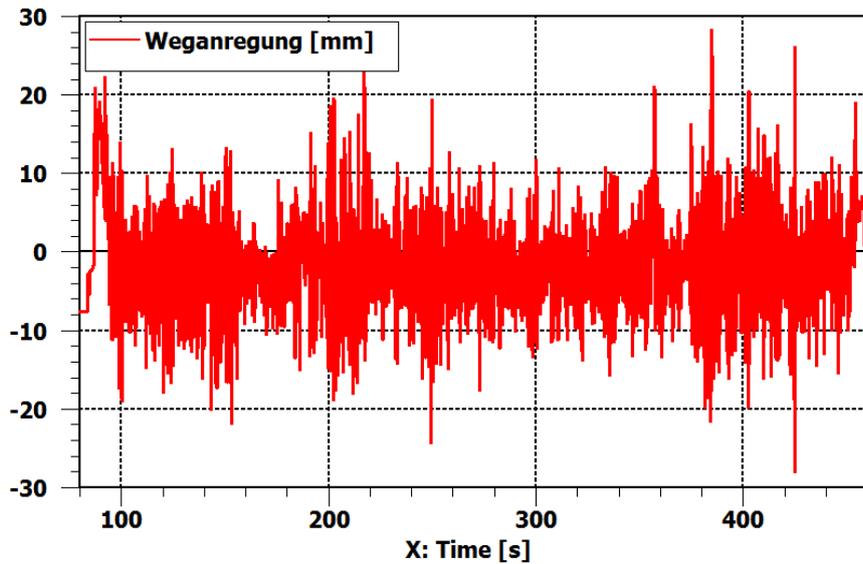


Abbildung 6.41 – Signal der Weganregung bei Feldwegfahrt

Abbildung 6.42 zeigt beispielhaft das Ergebnis der gemessenen und simulierten Druckverläufe auf der Kolbenseite der Federungszyylinder. **Abbildung 6.43** zeigt die Differenz beider Signale für den gewählten Ausschnitt.

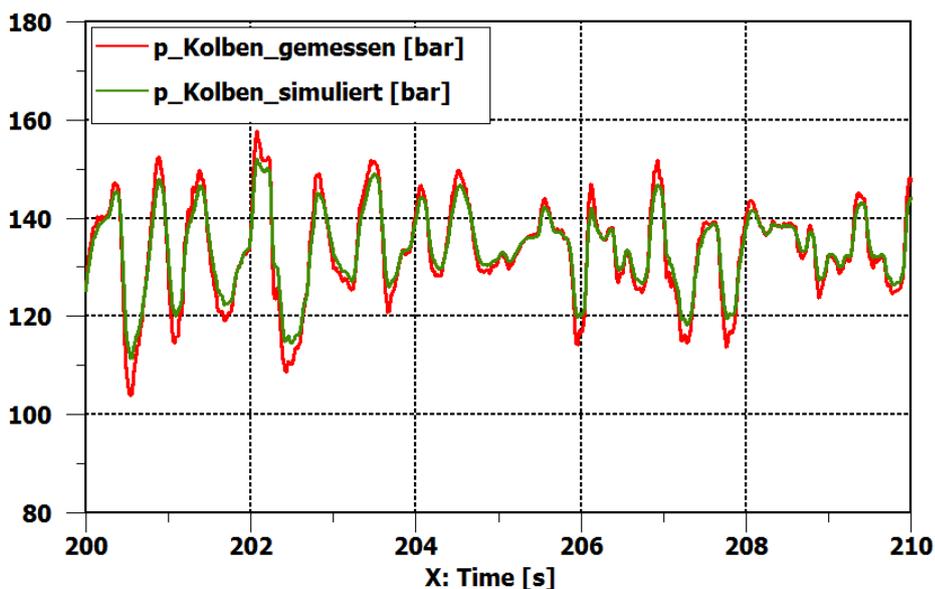


Abbildung 6.42 – Ausschnitt aus dem simulierten und gemessenen Druckverlauf an Kolbenseite

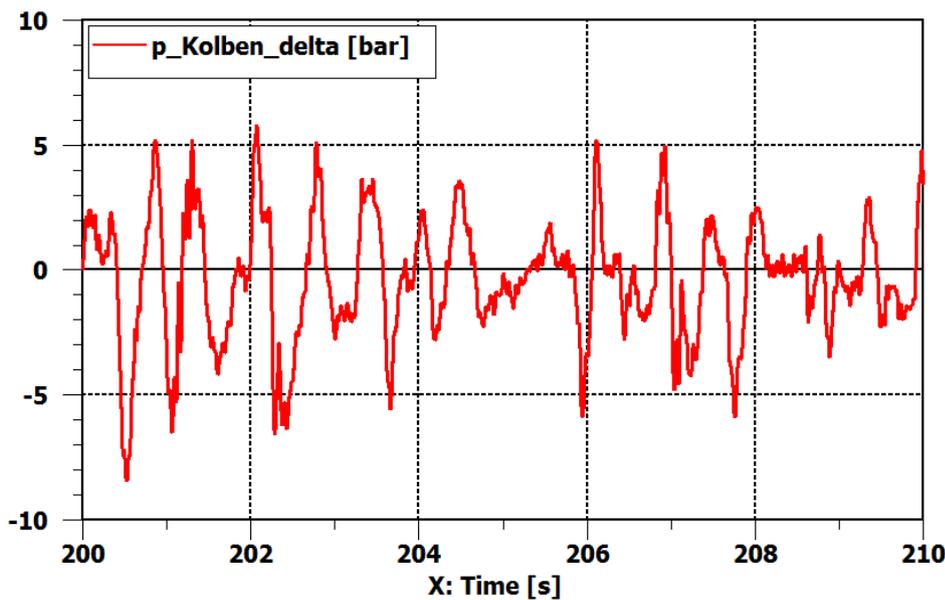


Abbildung 6.43 – Ausschnitt der Differenz zwischen simulierten und gemessenen Druckverlauf auf Kolbenseite

Die Abweichung der Simulation zur Messung wird wie folgt beschrieben:

$$\eta = \frac{(F_{real} - F_{simuliert})}{F_{real}} \cdot 100\%$$

Aus der **Abbildung 6.44** geht hervor, dass das gesetzte Ziel erreicht wurde. - Soll die Abweichung zwischen Simulation und Realität noch geringer werden, so ist ein wesentlich höherer Aufwand zu betreiben. Letzteres betrifft insbesondere das Modell des hydro-pneumatischen Federungskreises. Unter anderem kann es dann erforderlich werden, die Blenden-Ersatzmodelle wieder durch Leitungsmodelle zu ersetzen. Zudem wird es unabdingbar, sich den Reibungsverhältnissen des Systems zu widmen. Alle diese Maßnahmen bedingen, ohne Ausnahme, einen höheren Rechen- und Zeitaufwand, da das Simulationsmodell noch mehr Freiheitsgrade erhält.

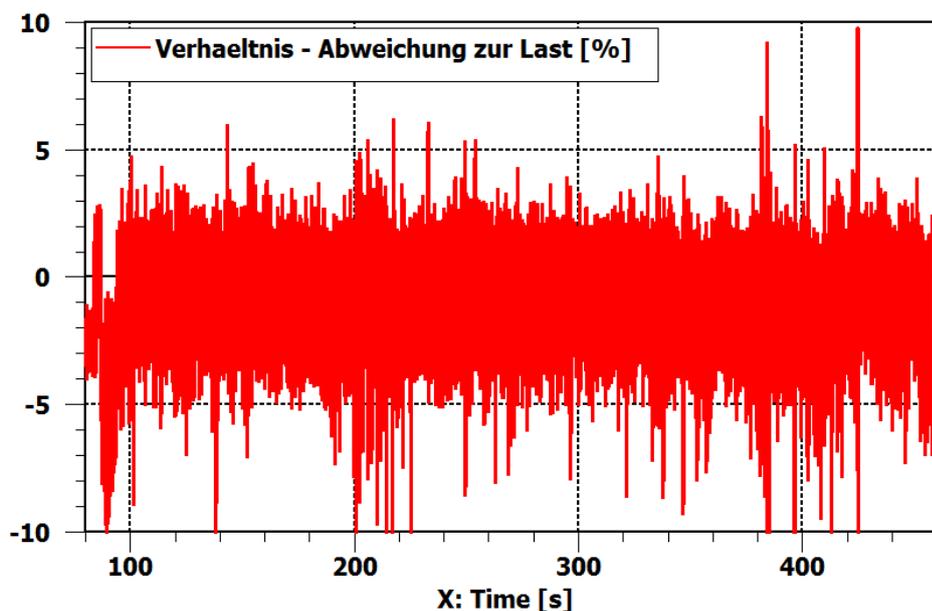


Abbildung 6.44 - Prozentuale Abweichung Messung – Simulation

Da die reellen Leitungsmodelle durch Blenden-Ersatzmodelle ersetzt wurden, wurde eine Abstimmung des Modells unabdingbar und so ergab sich eine Abweichungstoleranzgrenze zwischen Simulation und Messung von $\pm 5\%$. In gewissen Anwendungsgebieten ist diese Grenze durchaus beachtlich unter Berücksichtigung der Tatsache, dass dies allein durch die Verknüpfung von Know-How in einem Entwicklungsnetzwerk erfolgt und hierfür prinzipiell kein physischer Prototyp vorhanden sein muss. Eine derartige Abweichung ist für den ersten Schritt daher tolerabel. Für beispielsweise den virtuellen Entwurf von Komfortregelsystemen wären weitere Optimierungen erforderlich. Dies ließe sich z.B. durch eine noch höhere Abdeckung der dargestellten Frequenzen realisieren. Mit Hilfe weiterer Untersuchungen für die Modellierung ließe sich diese Abweichung noch verringern.

7. Zusammenfassung

In den Jahren 2008 bis 2011 wurde am Lehrstuhl für Mobile Arbeitsmaschinen am Karlsruher Institut für Technologie (KIT) das Forschungsverbundprojekt "Gekoppelte Unternehmensübergreifende Simulation Mobiler Arbeitsmaschinen zur Virtualisierung der Produktentstehung" (GUSMA) durchgeführt. GUSMA hat sich zum Ziel genommen, die Effizienz im Produktentstehungsprozess durch eine enge Verzahnung unternehmensübergreifender Entwicklungsnetzwerke zu steigern. Der Einsatz der Simulationstechnik soll dabei ermöglichen, das Know-How aller beteiligten Partner frühzeitig in den Entwicklungsprozess einzubringen, ohne dies aber offen zu legen, und somit unternehmensübergreifend virtuelle Prototypen aufzubauen. Die Durchführung des Projekts erfolgte in einem Zusammenschluss aus einem Forschungsinstitut, dem Lehrstuhl für Mobile Arbeitsmaschinen (Mobima) am KIT, einem OEM, der AGCO GmbH/Fendt, einem Zulieferer, der HYDAC SYSTEM GmbH und mehreren Software-Unternehmen, der Fluidon GmbH, LMS Deutschland GmbH und der SIMPACK AG. Die Kooperation der genannten Projektteilnehmer sicherte durchgehend die Praxistauglichkeit der erzielten Forschungsfortschritte, sowie - durch die beteiligten Softwareunternehmen - die softwaretechnische Umsetzbarkeit.

Grundlage für das Vorgehen im Projekt ist die Idee, die Technik der gekoppelten Simulation (oder auch: Co-Simulation) für den Zweck der abteilungs- oder unternehmensübergreifenden Modellerstellung nutzbar zu machen. Dies stützt sich insbesondere auf die Erkenntnis, dass die Komplexität mechatronischer Produkte zu einer Vielfalt an sowohl multidisziplinären als auch fachgebietsspezifischen Simulationswerkzeugen auf dem Markt geführt hat und deren Einsatz im Unternehmen entsprechend des Entwicklungs- und Optimierungsziel gewählt wird. Die Kopplung von Simulationsmodellen aus unterschiedlichen Simulationstools bietet sich daher als Hilfsmittel für den unternehmensübergreifenden Aufbau von virtuellen Prototypen an. Insbesondere in der mittelständisch geprägten Branche der mobilen Arbeitsmaschinen ist dieser Ansatz zur Darstellung und Validierung von Gesamtsystemen vielversprechend.

Im Verbundprojekt GUSMA wurde eine Möglichkeit geschaffen, um die Co-Simulation als unternehmensübergreifendes Werkzeug zu nutzen. Dies wurde im Projekt im Wesentlichen durch die Standardisierung der Co-Simulation erreicht. Diese basierte im Projekt auf drei Elementen:

- Der *Plattformgedanke*: Es wurde eine Simulationsplattform auf Basis von Matlab/Simulink entwickelt, welche den Benutzer beim Aufbau der Co-Simulation unterstützt und Möglichkeiten zum Datenhandling des Gesamtmodells integriert.
- Das *Standard-Interface*: Die Festlegung einer standardisierten Datenschnittstelle ermöglicht einen für den Benutzer einfachen Aufbau des Gesamtsystems. Die Kommunikation zwischen der Plattform und den jeweiligen Simulationstools erfolgt über einen zentralen Datenaustausch-Speicher, dem Matlab-Workspace. Hierfür wurde ebenso ein standardisiertes Datenformat definiert.
- Die *standardisierte Vorgehensweise*: Für einen effektiven Aufbau des Gesamtsystems wird der Benutzer anhand von sechs festgelegten Schritten durch den Modellaufbau geleitet. Dieses Vorgehen ist in eine Benutzeroberfläche implementiert.

Weiterhin wurde im Projekt der Know-How-Schutz beim Austausch von Teilmodellen als wesentliches Element betrachtet. Zum Schutz von Teilmodellen wurde ein Parameterschutz eingearbeitet, welcher den Export von sensiblen Daten des Modells verhindert.

Im Rahmen des Projekts wurden weiterhin Untersuchungen zur Festlegung eines geeigneten Modelldetaillierungsgrads durchgeführt. Diese haben ergeben, dass im Voraus – vor der unternehmensspezifischen Modellierung von Teilsystemen - eine Festlegung auf die maximal darzustellenden Frequenzen im Gesamtmodell erfolgen muss. Davon kann letztendlich auch die Einstellung des Kommunikationsintervalls für die Co-Simulation abgeleitet werden.

Die Validierung der Software-Plattform erfolgte am Beispiel einer hydropneumatischen Traktorvorderachsfederung mit Niveauregulierung. Referenzfahrzeug war ein Fendt-Traktor der 700er Serie. Dies fand sowohl virtuell auf der entwickelten Plattform, als auch durch praktische Versuche am Prüfstand statt. Die Aufteilung der Teilmodellentwicklung für die Simulation orientierte sich an der im Projekt gewählten Einteilung in Hersteller und Zulieferer. Auf diese Weise ergaben sich vier Teilmodelle, die abteilungs- bzw. unternehmensübergreifend erstellt und am Mobima, in der Rolle als Fahrzeughersteller, zu einem Gesamtsystem auf der GUSMA-Plattform aufgebaut wurden. Bei dem im Projekt gewählten Modelldetaillierungsgrad ergaben sich Abweichungen der Simulation zu Messergebnissen am Prüfstand von maximal +/- 5 Prozent. Dieses Ergebnis basierte auf der initialen Zusammensetzung validierter Teilmodelle. Mit entsprechenden Maßnahmen kann die Güte der Simulation noch erhöht werden. Jedoch ist dann eine verringerte Simulationsperformance und höherer Modellierungsaufwand zu erwarten.

Durch die im Projekt GUSMA erarbeitete Plattform können Teilmodelle abteilungs- und unternehmensübergreifend ausgetauscht und unabhängig von der ursprünglichen Modellierungsumgebung weiterverwendet werden. Dieser Aspekt eröffnet neue Möglichkeiten, um die Expertise von Hersteller und Zulieferer bereits in frühen Phasen des Produktentstehungsprozesses zu bündeln. Auf diese Weise kann die unternehmensübergreifende Simulation in hohem Maß zur Zeit- und Kostenoptimierung im Entwicklungsprozess beitragen.

Der GUSMA Standard wird als VDMA-Richtlinie veröffentlicht werden. Dieser befindet sich derzeit in der Ausarbeitung und wird ab Mitte des Jahres 2012 zur Verfügung stehen. Interessenten können sich an das genannte Institut wenden.

8. Literaturverzeichnis

1. Dronka, S.: *Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation*, 2004, TU Dresden.
2. Böhler, H.; Völker, L.: *Gekoppelte, unternehmensübergreifende Simulation mobiler Arbeitsmaschinen in BMBF-Kongress "10. Karlsruher Arbeitsgespräche Produktionsforschung 2010"*. 2010. Karlsruhe.
3. Heisler, R.; Brenninger, M.; Kraus, S.; Hofmann, R.: *Der Fendt 936 Vario*. ATZoffhighway, 2009. 08/2009: p. S. 38 - 49.
4. Frick, M.: *Erfolgsfaktoren der Produktentwicklung : ein interkultureller Vergleich in mittleren Unternehmen des Hochtechnologie-Sektors*, 2008: Hamburg.
5. *VDMA Maschinenbau in Zahl und Bild 2011*, 2011: Frankfurt am Main.
6. Jahnke, B; Geimer, M.; Noack, P.; Happich, G.: *Folgsame Traktoren - Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen*. KEM, 2012. 08/2012: p. 14-15.
7. Dengler, P.; Geimer, M.: *Zwischen den Drücken lesen - Effizienzsteigerung durch ein Konstantdrucksystem mit Zwischendruckleitung O + P* Zeitschrift für Fluidtechnik, 2011. 1-2: p. 24-27.
8. Nagel, P; Geimer, M; Roos, L.; Lang, T.: *Antriebsstrang mit Energierückgewinnung – Entwicklungsmethodik und Betriebsstrategien mobiler Arbeitsmaschinen*. in *Vollversammlung des Forschungsfonds Fluidtechnik im VDMA*. 2012. Frankfurt.
9. Thomke, S.; Fujimoto, T: *The effect of "front-loading" problem-solving on product development performance*1999, Harvard University. Graduate School of Business Administration. Division of Research: Division of Research, Harvard Business School.
10. Geimer, M.; Linsel, P.: *Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? Ein Versuch der Begriffsvereinheitlichung O + P* Zeitschrift für Fluidtechnik, 2006. 50(11-12): p. 572 - 576.
11. Schweizer, M.; Busch, M.: *Numerische Ansätze zur gekoppelten Simulation*, in *Fluidon Konferenz 2009*: Aachen.
12. Abele, E.; Reinhart, G.: *Zukunft der Produktion : Herausforderungen, Forschungsfelder, Chancen*2011, München: Hanser.
13. Völker, L., *Untersuchung des Kommunikationsintervalls bei der gekoppelten Simulation*, 2011: Karlsruhe.
14. *AMESim and Simulink Interface Manual*; , December 2008.
15. Völker, L.; Han, S.; Geimer, M.: *Standardisation of the co-simulation of mobile machines with participation of different business partners*. in *International Conference on Agricultural Engineering "Land.Technik"*. 2009. Hannover.
16. Han, S; Rüdener, A.; Geimer, M.: *Die GUSMA-Plattform: In sechs Schritten zum virtuellen Produkt*. Mobile Maschinen, 2011. 02/2011: p. 32-34.

9. Anhang

9.1 Ist-Stand bei Software-Herstellern im GUSMA Arbeitskreis: Beschreibung der bestehenden Co-Simulations-Schnittstelle

9.1.1 Die STC-Schnittstelle von DSHplus

Mit DSHplus gibt es grundsätzlich zwei Wege, Programme durch das DSHplus-STC Interface (Simulation Tool Chain) zu verbinden. Beide Varianten haben Vor- und Nachteile, zur Auswahl der geeigneten Variante ist es daher wichtig, die Prinzipien hinter den beiden Varianten zu kennen.

Die zwei Kopplungsmöglichkeiten

Die folgende Tabelle gibt einen Überblick darüber, warum es zwei Möglichkeiten zur Kopplung der DSHplus-STC mit anderen Simulationsprogrammen gibt.

Co-Simulation

Vorteile

- alle Größen im DSHplus-Modell können ohne Vorbereitung ausgewertet werden
- Modell ist transparent und einsehbar
- Analysemöglichkeiten von DSHplus verwendbar
- Schutz von Parametern möglich

Nachteile

- langsamer als das eingebettete Verfahren
- Bedienung mehrerer Programme notwendig

Embedded-Mode

Vorteile

- benötigt weniger Speicher und Rechenzeit
- Black-Box-Modell, Inhalt vor Anwender verborgen
- Technologie geschützt
- Schutz von Parametern möglich
- benötigt nur eine Runtime-Lizenz

Nachteile

- wenn interne Größen ausgewertet werden sollen, muss ihre Übergabe im Quelltext codiert sein
- Verständnis des Modells nur durch Dokumentation möglich

Für jeden Anwendungsfall bzw. abhängig vom aktuellen Projektstadium wird die eine oder andere Variante der Kopplung von Vorteil sein. Wichtig für den Anwender der DSHplus-STC ist, dass zu jedem Zeitpunkt ein Wechsel zwischen den beiden Kopplungsarten möglich ist.

Die Berechnung des Embedded-Moduls kann in diesem Zusammenhang wie eine Co-Simulation betrachtet werden, bei der aber der komplette Verwaltungsaufwand des zweiten Simulationsprogramms (Programmoberfläche, Online-Grafik, etc.) fehlt. Hierdurch ergeben sich selbstverständlich Geschwindigkeitsvorteile für das Embedded-Modul bei sonst gleichbleibender Berechnungsqualität.

Sind verschiedene Simulationszeiten in DSHplus und dem Zielprogramm eingestellt, dominiert das Zielprogramm den Ablauf. Ist z. B. in DSHplus eine Simulationszeit von 10 s und im Zielprogramm 5 s gewählt, ist die Co-Simulation nach 5 s beendet.

Funktionen der DSHplus-STC

Die DSHplus-STC ist ein Interfacewrapper um eine Basisschnittstelle von DSHplus, den **Open-Interface-Funktionen (OIF)**. Diese sind ein Programmierinterface (API), das aus fünf Funktionen besteht, mittels derer eine Interaktion mit den im Simulationsmodell integrierten Signaleingangs- und Signalausgangsbauteilen möglich ist.

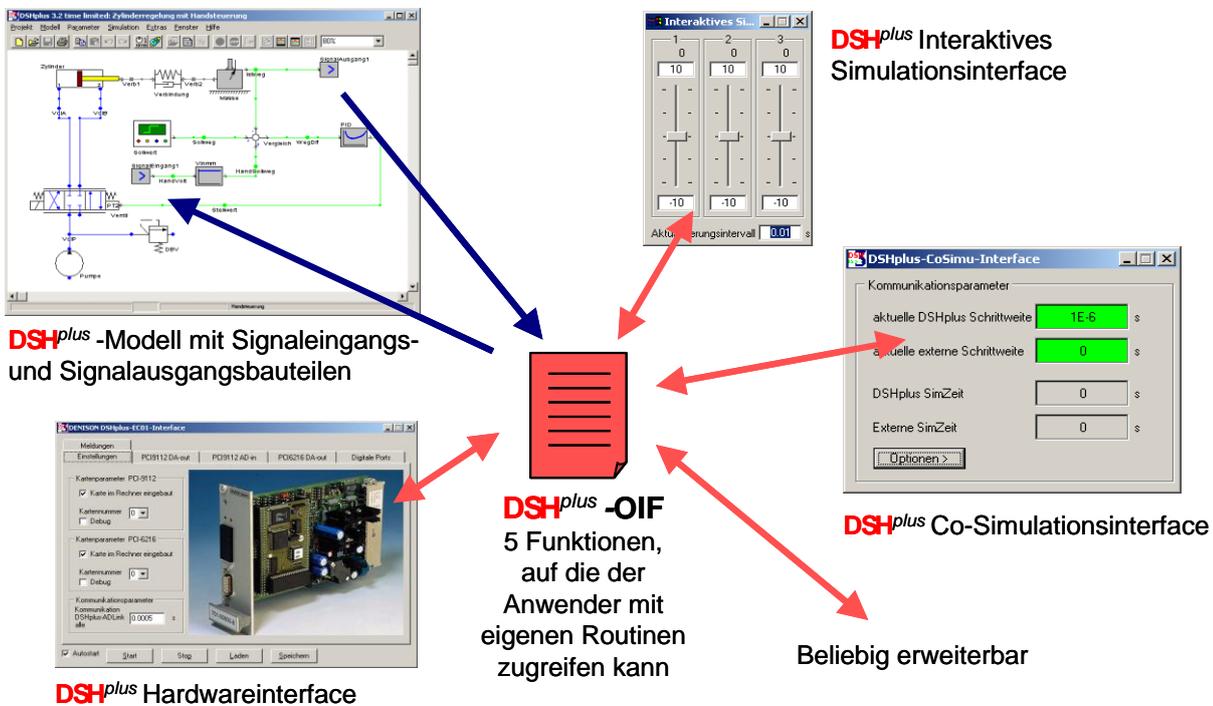


Abbildung 9.1 - Beispiele für Anwendungen der DSHplus-OIF

Anwendungsmöglichkeiten der DSHplus-OIF sind die interaktive Beeinflussung der Simulation über Schieberegler, die Anbindung eines externen Programms über das Co-Simulationsinterface (DSHplus-STC) oder die Ankopplung von Messtechnikhardware (DSHplus-Hil-Box) an das Simulationsmodell, siehe auch **Abbildung 9.1**.

Die DSHplus-STC-Modulfunktionen selber bilden die Schnittstelle des exportierten DSHplus-Moduls zu externen Programmen (**Abbildung 9.2**).

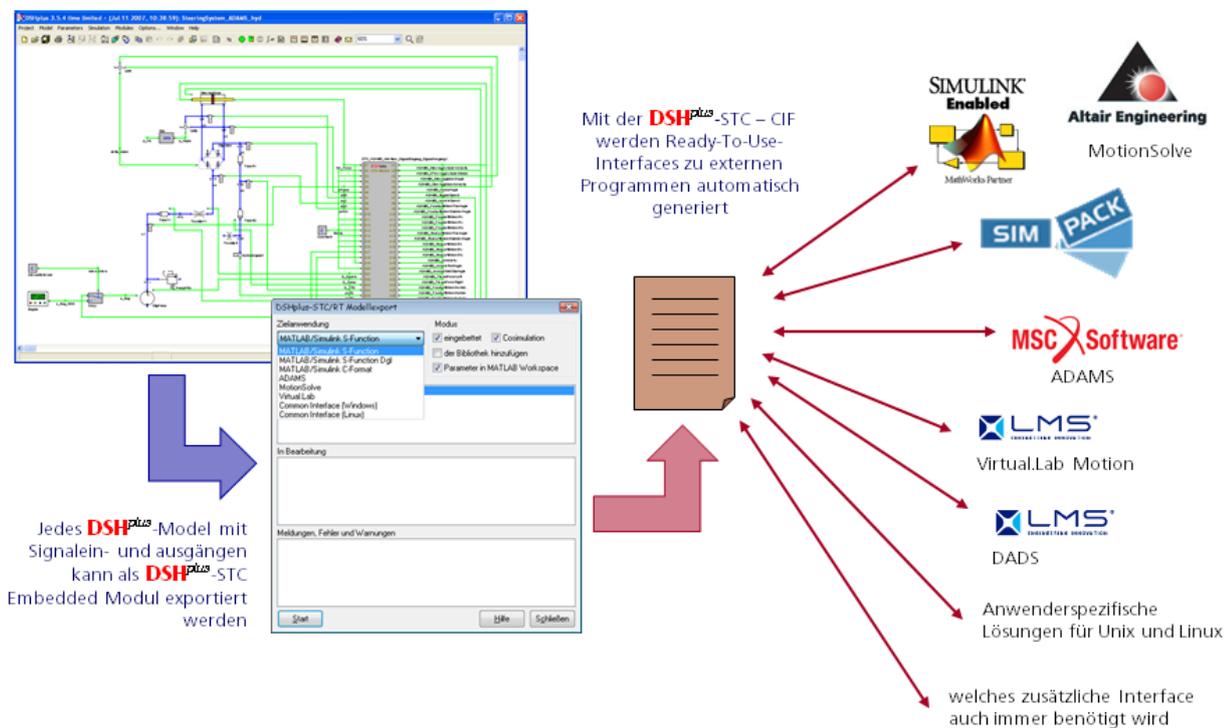


Abbildung 9.2 - Mögliche Zielprogramme für DSHplus-STC Module

Über die Modulfunktionen, die vom Anwender in das Zielprogramm eingebunden werden, erfolgt sowohl die Kommunikation mit dem DSHplus-Modul im Embedded-Modus als auch die Kommunikation mit dem Co-Simulationsinterface, das über die DSHplus-OIF wiederum die Verbindung zum DSHplus-Simulationsmodell herstellt.

Die Kopplungsprozedur

Abbildung 9.3 verdeutlicht den Ablauf der Funktionsaufrufe bei der Kopplung eines DSHplus-STC-Moduls oder der Co-Simulation mit einem externen Simulationsprogramm. Die Kopplung gliedert sich in die drei Betriebsbedingungen „Start“, „Simulation“ und „Stop“.

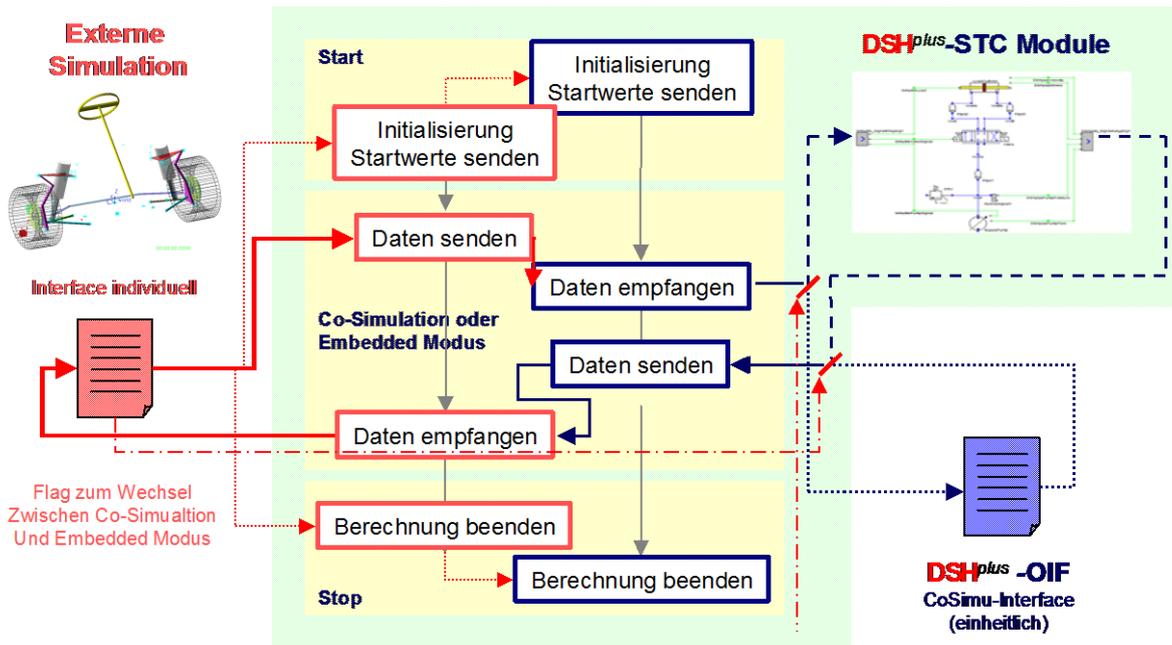
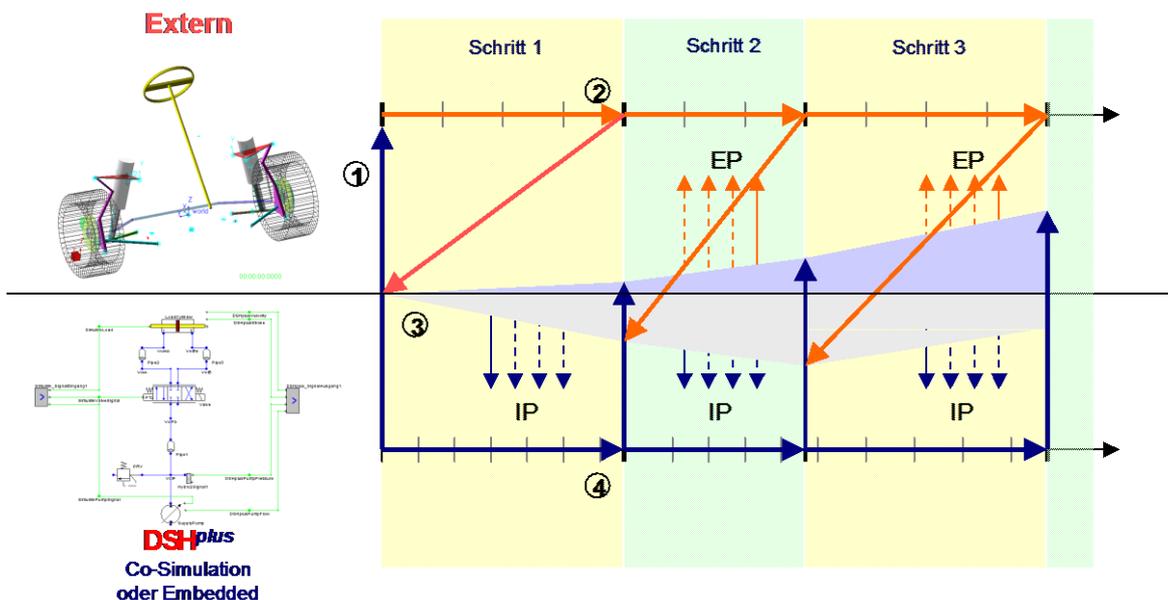


Abbildung 9.3 - Schematische Darstellung der DSHplus-STC-Kopplung

Beim Start der Kopplung (Initialisierung) wird im Fall der Co-Simulation zunächst DSHplus gestartet. DSHplus durchläuft die Initialisierung des Simulationsmodells und verfällt in einen „standby“ Modus. Im Embedded-Modus wird direkt das externe Programm gestartet, das seinerseits während der Initialisierung das DSHplus-STC-Modul initialisiert.

Das externe Programm empfängt im Anschluss an die Initialisierung die Startwerte des DSHplus-Modells und beginnt mit der Simulation des eigenen Modells, siehe **Abbildung 9.4**, (1).

Nach jedem verifizierten Simulationsschritt des externen Programms (2) werden die Werte der Eingangssignale in das DSHplus-Modell und die Zeitdauer des Simulationsschrittes an das DSHplus-STC-Modul übergeben (3).



Zweck: Vermeidung von sprunghaften Übergabewerten, die zu langen Rechenzeiten führen.

Abbildung 9.4 - Inter- und Extrapolation beim Datenaustausch mit dem STC-Modul

Das DSH*plus*-STC-Modul startet jetzt die Berechnung des vorgegeben Zeitintervalls. Um sprungförmige Änderungen der Übergabewerte des externen Programms zu verhindern, werden diese während der Berechnung interpoliert.

Nach der Berechnung des Zeitintervalls erfolgt die Übergabe der Ausgangssignale des DSH*plus*-STC-Moduls an das externe Programm (4), das mit dem nächsten Berechnungsschritt startet. Hierbei kann eine Extrapolation der DSH*plus*-Daten erfolgen, um während der Berechnung eine Veränderung fluidtechnischen Größen zu berücksichtigen.

Ist die Simulation des externen Programms beendet oder wird sie gestoppt, beendet der Aufruf der Stop-Funktion des externen Programms auch das DSH*plus*-STC-Modul oder die Co-Simulation.

9.1.2 Interface von LMS Imagine.Lab AMESim

LMS Imagine.Lab AMESim bietet eine komplette 1D-Simulationsplattform zur Modellierung und Analyse multidisziplinärer, intelligenter Systeme und zur Berechnung ihrer Performance in unterschiedlichen Disziplinen. Mittels validierter analytischer Modelle werden detaillierte Komponenten oder komplexe multidisziplinäre Systeme beschrieben, die das hydraulische, pneumatische, thermische, elektrische und mechanische Verhalten eines Systems abbilden können. Aus validierten Bibliotheken werden lediglich vordefinierte Bauteile zusammengesetzt, um das System realistisch zu simulieren. Damit ist LMS Imagine.Lab AMESim in der Lage, lange bevor detaillierte CAD-Geometrien zur Verfügung stehen, das System zu simulieren und somit Aussagen über das zukünftige Systemverhalten zu machen.

LMS Imagine.Lab AMESim bietet spezielle Lösungen für folgende Bereiche:

- Antriebsstrang und Getriebe (*Powertrain Transmission*)
- Verbrennungsmotoren (*Internal Combustion Engine*)
- Gesamtfahrzeugdynamik (*Vehicle System Dynamics*)
- Fahrwerke und Steuersysteme für Luft- und Raumfahrt (*Ground Loads and Flight Controls*)
- Fahrzeugwärmemanagement (*Vehicle Thermal Management*)
- Fluid Systeme (*Fluid Systems*)
- Elektromechanik (*Electromechanical*)
- Energie (*Energy*)

In LMS Imagine.Lab AMESim ist die Kopplung zwischen elementaren und anwendungsorientierten Bibliotheken einfach. Anwender bauen aus unterschiedlichsten physikalischen Bereichen, wie Hydraulik, Pneumatik, Thermik, Mechanik und Elektromechanik Komponenten oder Systemmodelle zusammen. Diese können wiederum sehr einfach kundenspezifisch angepasst, erweitert und in eigenen Bibliotheken archiviert werden. Darüber hinaus bietet die offene Simulationsplattform eine große Anzahl von ausgewählten Schnittstellen zu anderen Softwareprogrammen und Softwaresprachen an, z.B. auch für Simulink.

Durch Systemsimulation können Modelle im Zeit- und Frequenzbereich analysiert werden und dadurch ein besseres Systemverständnis abgeleitet werden. Sie wird inzwischen sowohl bei der Entwicklung hydraulischer Komponenten (z.B. Pumpen, Ventile) als auch kompletter Systeme von der Common-Rail-Einspritzung über die Bagger-Aktuatorik bis zur Kühlerauslegung erfolgreich eingesetzt.

Das Simulationsprogramm AMESim besitzt für die Kopplung mit anderen Programmen ein Schnittstellenelement. Das Schnittstellenelement wird im Skizziermodus erstellt. **Abbildung 9.5** zeigt das Konfigurationsfenster des Schnittstellenelements.

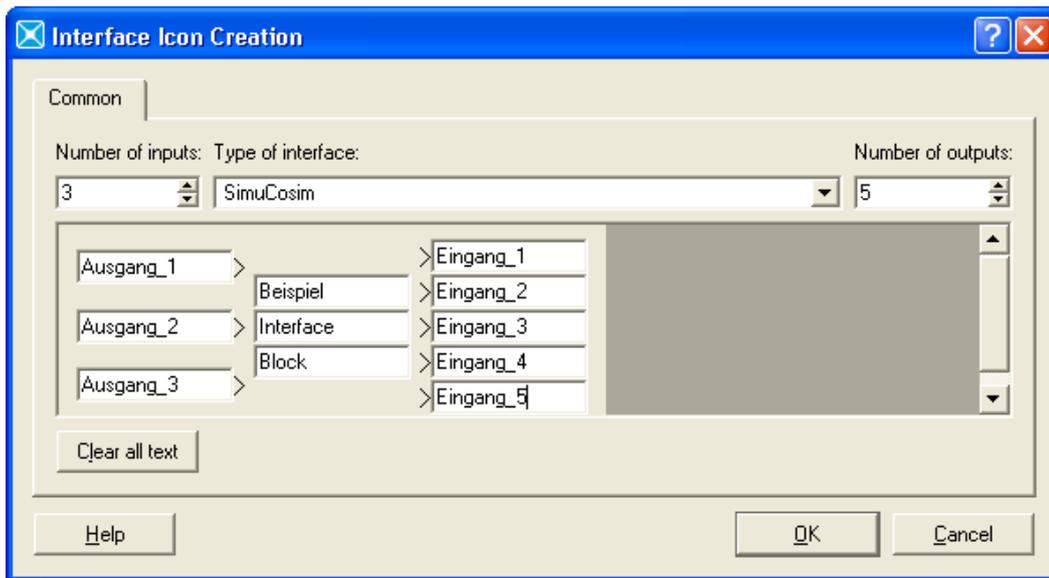


Abbildung 9.5 - Konfigurationsfenster des AMESim-Interface-Blocks [13]

Für die Kopplung mit anderen Programmen muss zunächst der Schnittstellentyp (Type of Interface) angegeben werden, d.h. es wird festgelegt, mit welchem Programm gekoppelt wird. AMESim ermöglicht die Kopplung zu Simulink, Adams, VLMotion und LabView. Darüber hinaus kann auch der Schnittstellentyp „User“ oder „UserCosim“ gewählt werden, die durch den Anwender selbst programmiert werden können. Für die Kopplung mit MATLAB/Simulink kann zwischen dem Schnittstellentyp „Simulink“ und „SimuCosim“ gewählt werden. Bei Schnittstellentyp „Simulink“ wird ein Modellexport durchgeführt, indem die Gleichungen des AMESim-Modells exportiert, in Simulink eingebunden und durch Simulink bzw. dessen Solver gelöst werden. Beim Schnittstellentyp „SimuCosim“ wird eine Co-Simulation zwischen Simulink und AMESim durchgeführt. Der Unterschied zwischen beiden Möglichkeiten ist in **Abbildung 9.6** noch einmal grafisch dargestellt.

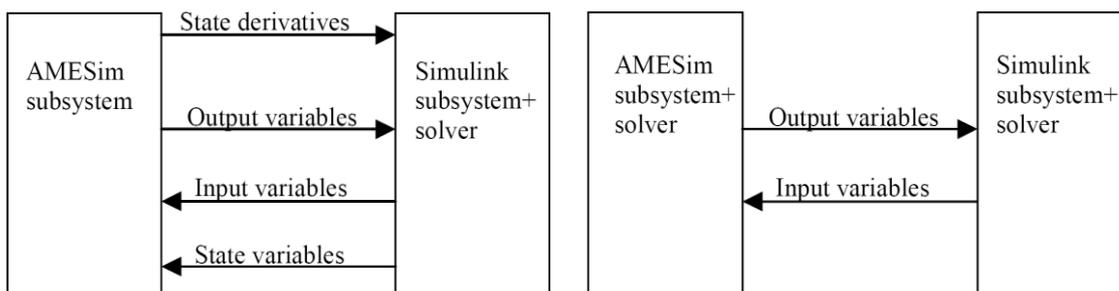


Abbildung 9.6 – AMESim Code-Export (links), Co-simulation AMESim und Simulink [14]

Nach der Festlegung des Schnittstellentyps wird die Anzahl an Eingangs- und Ausgangsgrößen festgelegt. Maximal sind hier 99 Ein- und Ausgänge zulässig. Es können jedoch mehrere Schnittstellenblöcke eingebaut werden. Diese müssen dann jedoch vom gleichen Schnittstellentyp sein. Die Ein- und Ausgänge sowie das Schnittstellenelement selbst können mit Namen versehen werden. Das Schnittstellenelement sieht dann entsprechend **Abbildung 9.7** aus. Bei diesem Beispiel wurden so genannte Empfängerblöcke (Receiver) und Übertragungsblöcke (Transmitter) integriert, damit alle Anschlüsse des Schnittstellenelements verbunden sind.

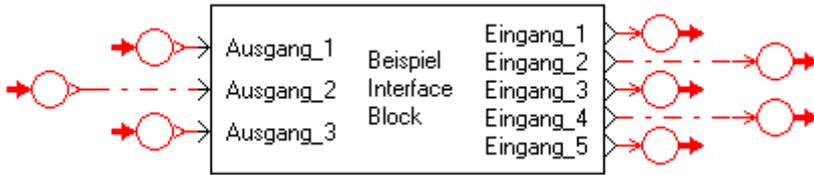


Abbildung 9.7 - SimuCosim-Schnittstellenelement mit Reciever- und Transmitterblöcken

Die Einbindung des AMESim-Modells unter Simulink erfolgt über die S-Function aus der Simulink –Toolbox „User-defined function“. In die S-Function wird der Name des AMESim-Modells mit Unterstrich am Ende des Namens eingetragen. In dem in **Abbildung 9.8** dargestellten S-Function-Konfigurationsfenster lautet der Name Beispiel_Interface_Block. Bei den S-Function-Parametern werden die Einstellungen für den AMESim-Solver eingegeben. Die erste Zahl gibt das Kommunikationsintervall (Sampling Period) an. Im Beispiel ist dies 1 ms. Die zweite Zahl gibt die Zeitpunkte an wann AMESim Werte speichert (Communication Interval). In diesem Fall alle 0,01 s. Die dritte Zahl gibt die Fehlertoleranz an, die bei 10⁻⁵ s liegt. Dies sind die drei wichtigsten Einstellungen, die getätigt werden müssen. Darüber hinaus kann die maximale Schrittweite und die Zeitspanne (Time Range) übergeben werden sowie Einstellungen, ob die Ergebnisse, die Unstetigkeiten oder das Zeitsignal dargestellt werden sollen.

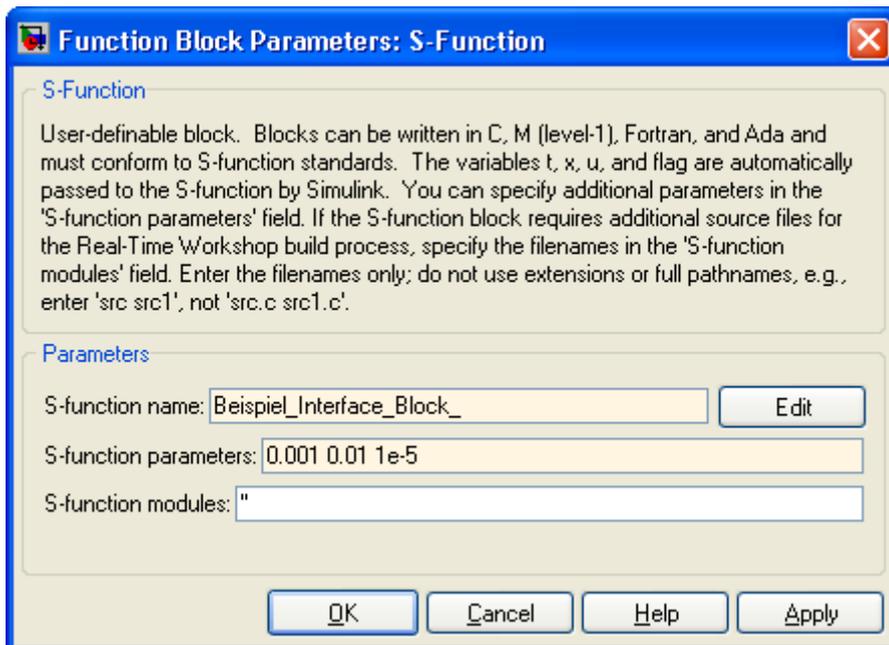


Abbildung 9.8 - Blockmaske des AMESim – S-Function [13]

Wird nun die Simulation unter Simlink gestartet, wird die SFunction initialisiert und an das AMESim-Modell angepasst und es ergibt sich die S-Function in der in **Abbildung 9.9** dargestellten Form. Es werden die Ein- und Ausgänge mit ihren Namen angezeigt. Allerdings wird die Reihenfolge im Vergleich zum Schnittstellenblock in AMESim gedreht. Zusätzlich wird der Name des AMESim-Modells angezeigt, welches dem S-Function-Block hinterlegt ist.

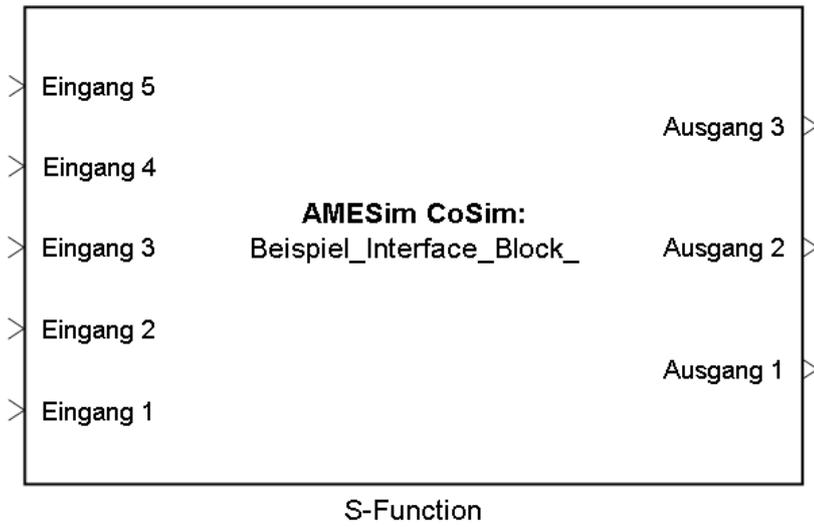


Abbildung 9.9 – Initialisierter AMESim S-Function Block [13]

9.1.3 Interface der Software SIMPACK

SIMPACK Kinematics & Dynamics

Das Grundmodul „SIMPACK Kinematics and Dynamics“ ist das Herzstück der MKS-Software SIMPACK. Mit diesem Modul können Modelle generiert, gerechnet und ausgewertet werden. Die Grundfunktionalitäten umfassen u.a. die Parametrierung sowie die Datenbankverwaltung. Alle weiteren SIMPACK Module, Schnittstellen und andere Add-On's können an SIMPACK Kinematics and Dynamics angeschlossen werden. Speziell für SIMPACK ist der High-End Solver SODASRT entwickelt worden, welcher die Bewegungsgleichungen mit großer Schnelligkeit und Stabilität löst. Es können Rechnungen im Zeit- und Frequenzbereich durchgeführt werden. Die Auswertung der Ergebnisse erfolgt im SIMPACK Post Prozessor, welcher ebenfalls mit SIMPACK Kinematics and Dynamics mitgeliefert wird. Die Oberfläche erlaubt die visuelle Darstellung der Modell-Animation, sowie 2- und 3-D Plots. Eine umfangreiche Filterbibliothek steht ebenso zur Verfügung.

SIMPACK Automotive

Das SIMPACK Automotive Zusatzmodul dient der Modellierung und Analyse von Straßenfahrzeugen. Ebenso werden Modelle für Trassierung und Fahrbahnanregung, eine Bibliothek mit parametrisierten SIMPACK Substrukturmodellen von Fahrzeugkomponenten (diverse Standard-Radaufhängungen, Lenkung, Triebstrang, etc.) sowie zwei Reifenmodelle (*Pacejka Curve fit* sowie *Pacejka Similarity Method*) zur Verfügung gestellt.

FEMBS

FEMBS ist eine SIMPACK-FEM Schnittstelle zur Einbindung elastischer Körper in SIMPACK. FEMBS beinhaltet Zusatzmodule zur Berücksichtigung lokaler elastischer Verformungen (Frequency Response Modes) und zur Erstellung einer SIMPACK-3D-Grafik aus dem FE-Modell. Das Zusatzmodul ist erhältlich für folgende FE-Codes: ABAQUS®, ANSYS®, MSC.Nastran®, NX Nastran, I-DEAS® und PERMAS.

SIMBEAM

SIMBEAM ist ein SIMPACK Modul zur Bereitstellung elastischer, räumlicher balkenförmiger Strukturen für SIMPACK. Starr-elastische Koppelterme und nichtlineares Verformungsverhalten werden automatisch berücksichtigt.

SIMPACK Code Export

Mit SIMPACK Code Export kann von SIMPACK Anwendermodellen der zugehörige Rechencode der nichtlinearen Bewegungsdifferentialgleichung erzeugt und exportiert werden. Dieser Code kann mit SIMPACK ausgeführt werden. Zur Ausführung des Codes in anderen Simulationsumgebungen wird dann die Funktionalität *SIMPACK Code Execution* benötigt, die in zwei unterschiedlichen Varianten angeboten werden.

SIMPACK Inter Process Communication (IPC)

Die Schnittstelle zur Kopplung von SIMPACK mit anderen Simulatoren durch Co-Simulation über TCP/IP – basierte *Sockets*.

MatSIM

Das MatSIM Modul dient dem Export von *MATLAB Simulink* Modellen nach SIMPACK. MATSIM konvertiert die mittels des *Realtime Workshop* exportierten Simulink-Modelle in SIMPACK User-Routinen. MATSIM wird bereits zum Export der Simulink Modelle benötigt, da es als Target anzugeben ist. Zur Nutzung von MATSIM ist das Modul SIMPACK Control erforderlich.

SIMAT

SIMAT ist eine zweiteilige Schnittstelle zwischen SIMPACK und *MATLAB/Simulink*. Zum einen können lineare Systemmatrizen ausgetauscht werden (*Linear System Interface*), zum anderen stellt SIMAT eine Co-Simulationsschnittstelle bereit. Die Co-Simulationsschnittstelle ermöglicht die gleichzeitige Simulation von SIMPACK- und *Simulink*-Modell mit dem jeweiligen *Solver* sowie dem Austausch der Ergebnisse.

Der mit der Version SIMPACK 8904 standardmäßig ausgelieferte SIMAT Block für die Co-Simulation zwischen *MATLAB* und SIMPACK ist in **Abbildung 9.10** dargestellt.

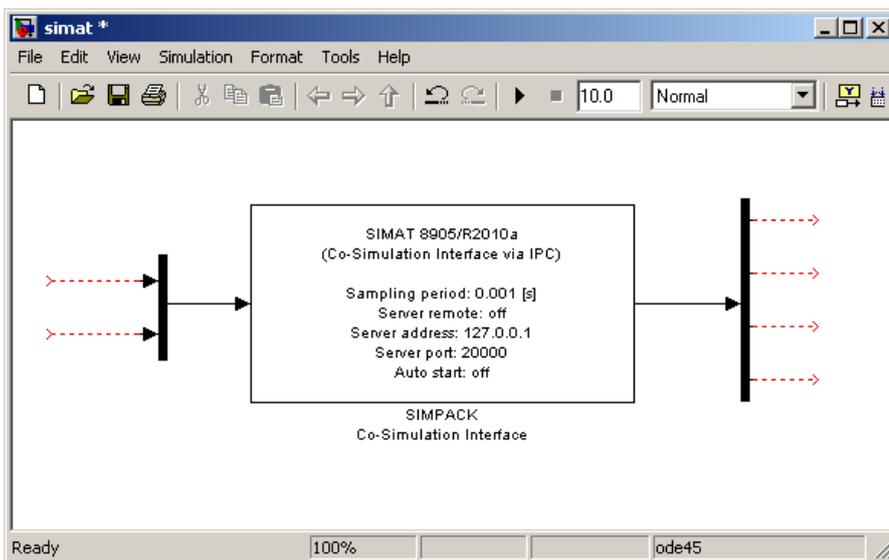


Abbildung 9.10 – SIMAT Block in SIMPACK 8904

Der Block verfügt lediglich über einen Eingang und einen Ausgang. Die Zahl der in SIMPACK definierten Ein- und Ausgänge ist daher aus dem Block nicht ersichtlich. Der Anwender muss manuell einen „Mux“ Block aus der *MATLAB/Simulink* Bibliothek erstellen. Der „Mux“ Block (links im Bild) sorgt für das Bündeln aller Eingänge, die in den SIMAT Block fließen sollen. Analog dazu wird ein „Demux“ Block zum Aufsplitten aller Ausgänge aus dem SIMAT Block benötigt (rechts im Bild).

Nachteil dieser Umsetzung ist, dass der Anwender des SIMAT Blocks nicht über folgende grundlegende Informationen verfügt:

- Anzahl der Eingänge
- Anzahl der Ausgänge
- Namen der Ein- und Ausgänge (für eine vereinfachte Zuordnung im MATLAB Modell)

Eine Verwechslung der Zuordnung sowie eine falsche Anzahl von Ein- und Ausgangskanälen in den Mux- bzw. Demux-Blöcken ist leicht möglich. Der Anwender des SIMAT Blocks muss also vom Ersteller des SIMPACK Modells diese Informationen erhalten, was nicht immer unproblematisch ist, insbesondere wenn MATLAB und SIMPACK Anwender bei unterschiedlichen Firmen angestellt sind.

Des Weiteren verfügt der SIMAT Block über folgende Eigenschaften:

- Vorgabe des Kommunikationsintervalls (default = 0.001)
- Angabe des Ports (default = 20000)
- Optional: Angabe eines Remote Servers (für die Co-Simulation über ein Netzwerk bzw. das Internet)
- Optional: Verwenden der Auto-Start Option (für das automatisierte Starten der Co-Simulation)

Abbildung 9.11 zeigt diese Eigenschaften des SIMAT Blocks und wie sie sich vom Anwender einstellen lassen.

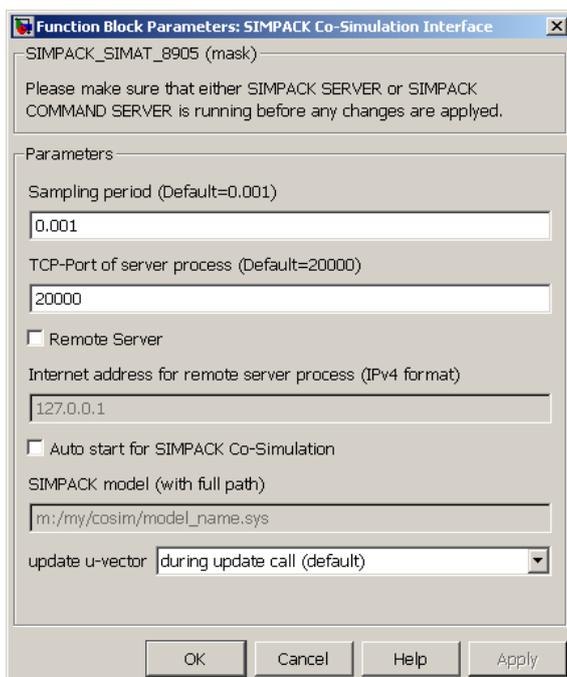


Abbildung 9.11 – Eigenschaften des SIMAT-Block

9.2 Beispielhafte Implementierung des GUSMA-Standards

9.2.1 Beispielhafte Implementierung der Schnittstelle in AMESim

Während des GUSMA Projektes wurden unterschiedliche Skripte auf Matlab-Basis entwickelt, welche es erlauben

- Parameter für die GUSMA Plattform zu exportieren: Skript `readgusma.m`

```
>> help readgusma
Matlab file to read global parameters from an AMESim model to
the GUSMA platform

This script will read AMESim global parameters of an AMESim file
and create Matlab variables corresponding to these global parameters.
This Matlab variables are structures containing the following fields:
- 'value': contains the value of the parameter or init. variable
- 'marker': - if the title of the AMESim global parameter
              starts with a '#': 'iv' (corresponds to the start
              value of a state variable)
              - else: 'p' (standard parameter)
- 'unit': contains the unit of the global parameter

Before using this script it is necessary to create a variable
with the name "amefile" which value is the name of the AMESim file
without extension (.ame)

see also writegusma

Anthony Domi <anthony.domi@lmsintl.com>
Copyright (c) 2010-2011 by LMS Deutschland GmbH
```

Abbildung 9.12 – Das Skript `readgusma.m`

Mit dem in **Abbildung 9.12** gezeigten Skript „`readgusma.m`“ ist es möglich, globale AMESim Parameter aus einem AMESim-Modell bzw. der dazugehörigen AMESim-Datei auszulesen und Matlab-Daten entsprechend dem GUSMA-Standardformat zu erstellen.

- Geänderte Parameter, die innerhalb der GUSMA Plattform geändert wurden in dem LMS Imagine.Lab AMESim Modell zu laden: Skript `writgusma.m`

```
>> help writegusma
Matlab script to write global parameters from the GUSMA platform to
an AMESim model

This script will look at all variables located in the Matlab workspace
and if they corresponds to an AMESim parameter defined by the GUSMA
platform, it will write the value of the parameters in the AMESim file.

Before using this script it is necessary to create a variable
with the name "amefile" which value is the name of the AMESim file
without extension (.ame)

Example:
    amefile = '3PistonPump';
    readgusma;
    Ppist.value = 10;
    writegusma;

see also readgusma

Anthony Domi <anthony.domi@lmsintl.com>
Copyright (c) 2010-2011 by LMS Deutschland GmbH
```

Abbildung 9.13 – Das Skript writegusma.m

Das in **Abbildung 9.13** dargestellte Skript „writegusma.m“ ermöglicht das Zurückschreiben geänderter Daten aus dem Matlab Workspace in die AMESim-Datei.

Diese beiden Skripte müssen dann in Verbindung mit der Co-Simulation Schnittstelle zwischen LMS Imagine.Lab AMESim und Matlab Simulink benutzt werden und stellen somit die Kommunikation zwischen GUSMA-Plattform und AMESim her.

Während des Projektes wurde auch die LMS Imagine.Lab AMESim Blackbox Schnittstelle weiterentwickelt. Diese Schnittstelle erlaubt eine LMS Imagine.Lab AMESim Modell nach Simulink zu exportieren, ohne dass es notwendig ist auf die Lizenzierung von LMS Imagine.Lab AMESim zugreifen zu können. Das exportierte Modell enthält aber nicht den LMS Imagine.Lab AMESim Solver. Es handelt sich dabei also um einen Modellexport und keine Co-Simulation.

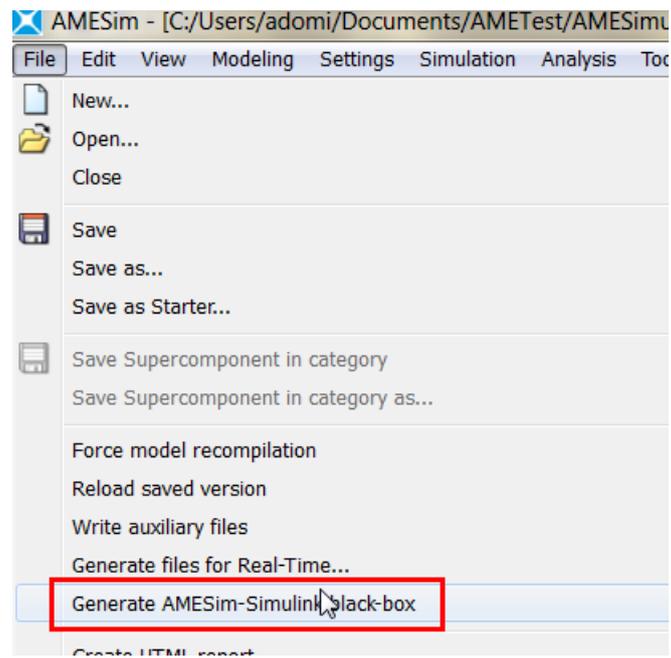


Abbildung 9.14 – Screenshot des Startmenüs in AMESim

9.2.2 Beispielhafte Implementierung der Schnittstelle in DSHPlus

Der GUSMA-Standard stellt verschiedene Anforderungen an Modell und Modellexport bzw. –import. Im Folgenden ist beschrieben, wie diese Anforderungen in der Software DSHplus im Einzelnen erfüllt sind.

Festlegen von geschützten/nicht zu exportierenden Parametern

Bei der Parametrierung der Komponenten in DSHplus kann für jeden einzelnen Parameter festgelegt werden, ob er schreibgeschützt und/oder verborgen sein soll. Diese Eigenschaft gilt sowohl bei der Simulation in DSHplus selber als auch im exportierten Modell.

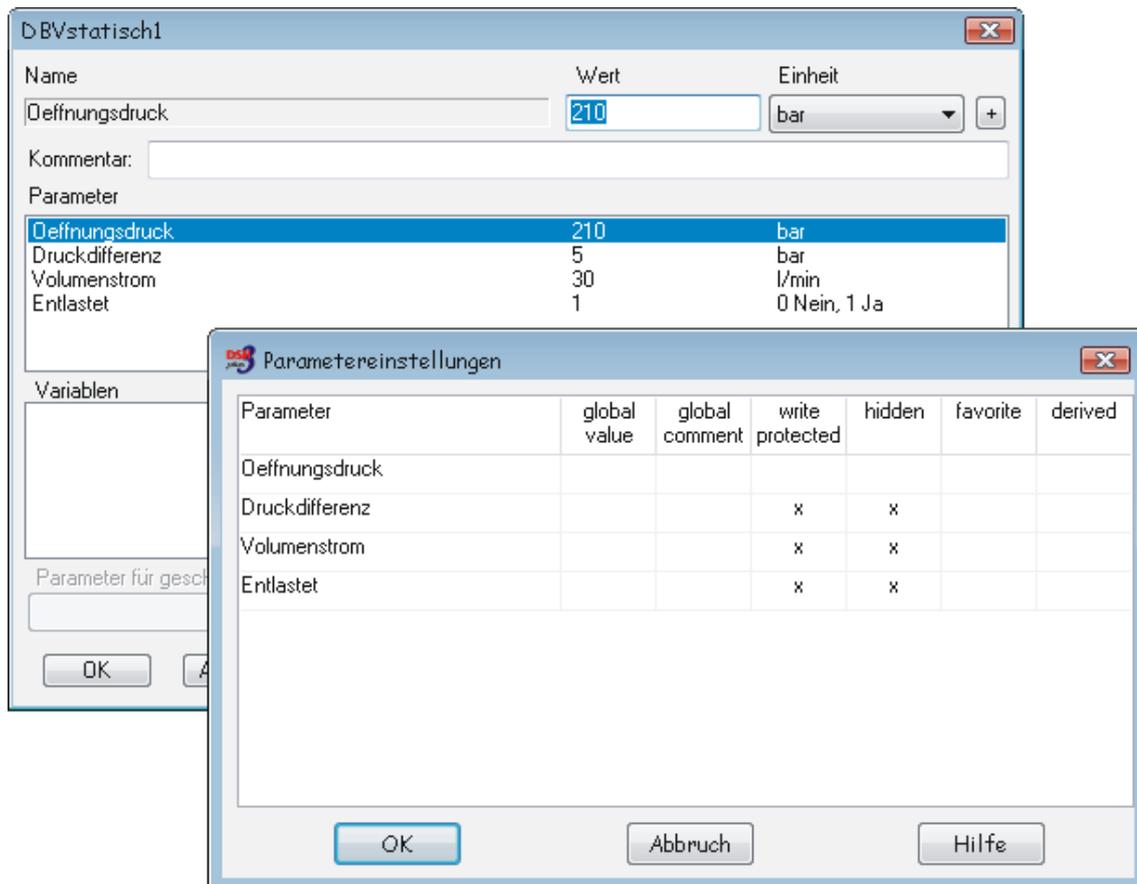


Abbildung 9.15 - Parameterdialog des DBVs mit Eigenschaftsdialog

Verborgene Parameter sind im exportierten Parametersatz nicht vorhanden, sondern werden fest in das Modell geschrieben, schreibgeschützte Parameter sind sichtbar, können aber nicht geändert werden.

Export nach MATLAB/Simulink

Mit dem Export eines DSHplus-Modells als MATLAB/Simulink S-Function im Modus „embedded“ oder „Co-Simulation“ und der Zusatzoption „Parameter in MATLAB Workspace“ (siehe unten) steht ein Paket zur Verfügung, das den GUSMA-Standard erfüllt und mit der GUSMA-Plattform verwendet werden kann.

Folgende Dateien werden beim Modelleport erzeugt:

Datei	Bedeutung
LSPumpe.dll	DSH <i>plus</i> -Simulationsmodell
LSPumpeModell.dll	DSH <i>plus</i> -Interface
LSPumpe.txt	Parameter und Variablen des Simulationsmodells
LSPumpe.ken.txt	Kennfelder des Simulationsmodells
LSPumpe.mdl	Simulink Blockdefinition (mit Ein-/Ausgangsvektoren)
LSPumpe_SinglePort.mdl	Simulink Blockdefinition (mit aufgelösten Ein-/Ausgängen)
LSPumpe_dll.mexw32	MATLAB S-Function (MATLAB-seitiges Interface)
LSPumpe_par.m	Parameterstruktur für MATLAB-Workspace
LSPumpe_save.m	Makros zur Parameterbehandlung in MATLAB

Das DSH*plus*-Modell wird in Simulink durch einen Block repräsentiert, der alle Ein- und Ausgänge entsprechend des ursprünglichen DSH*plus*-Modells enthält.

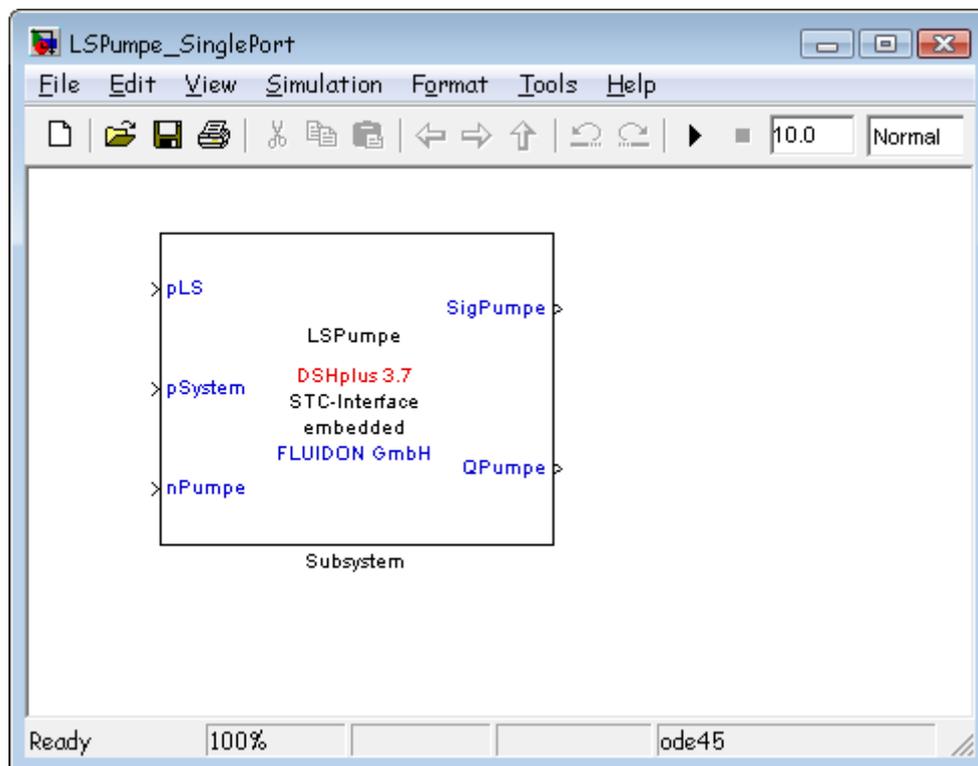


Abbildung 9.16 - S-Function-Block des DSH*plus*-Modells in Simulink

Eine begleitende Dokumentation erläutert dem Anwender die Funktion des Modells, beschreibt die Schnittstellen-Größen und listet die Parameter des Modells auf. Hier liegt es in der Verantwortung des Modell-Lieferanten, eine aussagekräftige Dokumentation zu erstellen.

Import in MATLAB/Simulink

Durch die Auswahl der Exportoption „Parameter in MATLAB Workspace“ werden die zur Parameterverarbeitung im MATLAB-Workspace notwendigen m-Files erzeugt.

Die in MATLAB sichtbaren Parameter sind gemäß der Konvention benannt, das m-File stellt die Verbindung zwischen den MATLAB-Parametern und den DSHplus-Modellparametern her. Als Trennzeichen wird „_“ verwendet. Durch die Verwendung von Callback Funktionen im Simulink-Block ist sichergestellt, dass die Parameter beim Einladen des Blocks in den Workspace eingeladen werden und dass die Parameter beim Speichern und bei Simulationsstart vom Workspace ins Modell zurückgeschrieben werden.

Einführung in Softwareversion

Dieses Kapitel beschreibt die Schritte, die für den Export eines Modells aus DSHplus für die GUSMA-Plattform durchgeführt werden müssen. Der Aufbereitung und dem Test des Modells, bei denen die Modelländerung, die Parametrierung und die Interfaceauswahl beschrieben werden, folgt die Darstellung des Exportvorgangs selbst. Abschließend wird die Verwendung im externen Zielprogramm erläutert.

Aufbereiten und Testen des Modells

Ausgangspunkt für den Modellexport durch die DSHplus-STC ist ein vollständig parametrisiertes Simulationsmodell. Als Beispiel wird das Modell einer einfachen hydraulischen Achse, bestehend aus Pumpe, Ventil, DBV, Zylinder und Leitungen verwendet.

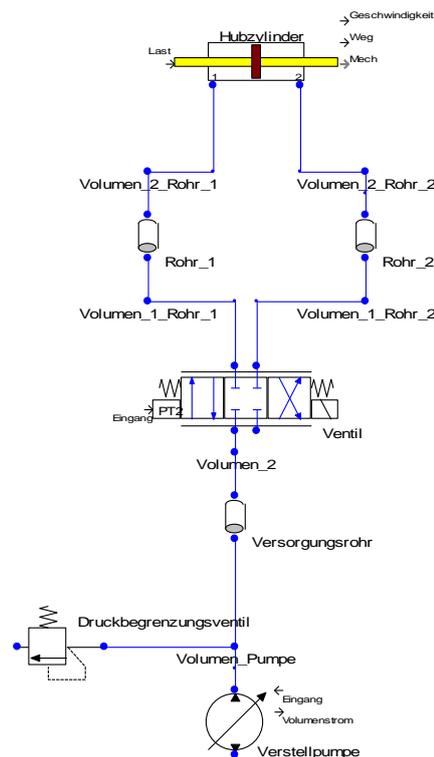


Abbildung 9.17 - DSHplus-Modell einer einfachen hydraulischen Achse

Die Vorgabe der Zylinderlast, die Ansteuerung der Pumpe und des Ventils sowie die Lageregelung des Systems sollen von außen erfolgen. Hierzu müssen zuerst die Schnittstellenelemente zur Kopplung von DSHplus-Modellen mit externen Programmen hinzugefügt und parametrisiert werden. Ein erster Test mit einer einfachen Ansteuerung

der Schnittstellen durch Schieberegler – dem Standarddialoginterface der DSHplus-OIF – hilft, die Funktion des Modells sicherzustellen. Anschließend erfolgt der Export zur Zielanwendung.

Im Folgenden werden die einzelnen Schritte detailliert erläutert.

Exportvorbereitung

Nach dem Aufbau des hydraulischen Teils des Modells folgt die Einbindung der Schnittstellenelemente im Bearbeitungsmodus von DSHplus. Die Schnittstellenelemente sind Grundvoraussetzung für die Einbindung des Modells ins Zielsystem.

Dabei ist darauf zu achten, dass stets der Name SignalEingang mit einer laufenden Nummer zu verwenden ist (z. B. SignalEingang1, SignalEingang2, etc.). Der Text vor diesem Namen ist frei wählbar (z. B. TestelementSignalEingang1). Wird dies bei der Modellerstellung missachtet, wird das eigentliche DSHplus-Modell zwar ordnungsgemäß erstellt, der Datenaustauschvektor aber nicht richtig erzeugt.

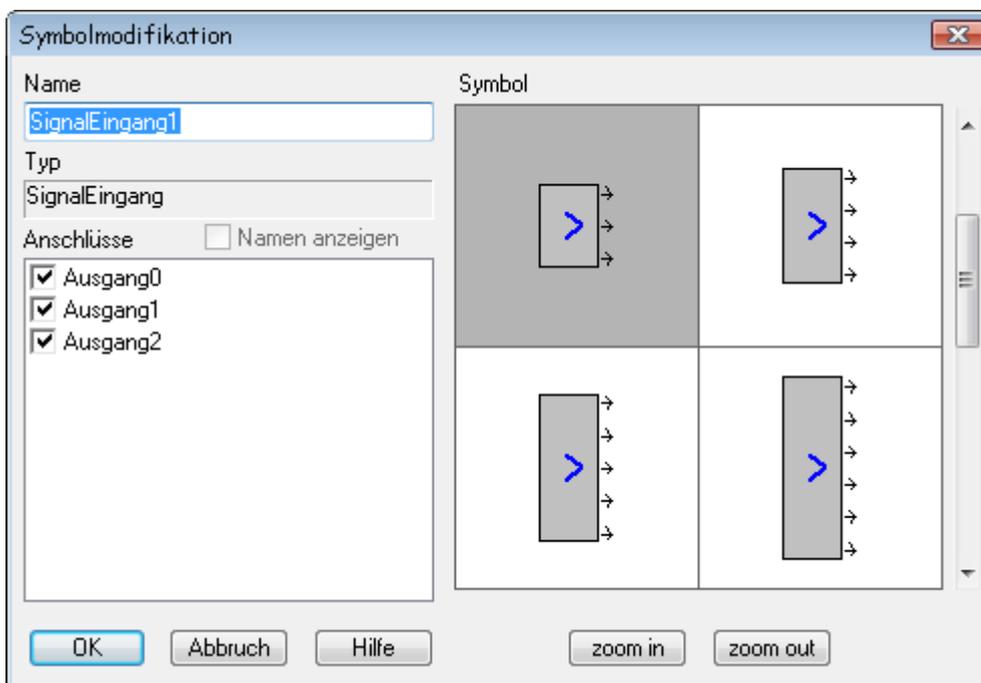


Abbildung 9.18 - Symbolmodifikation SIGNALEINGANG

Gleiche Regeln gelten für das Element SignalAusgang, das zur Datenübergabe vom DSHplus-Modell an das externe Programm gebraucht wird und in der Bibliothek unter derselben Gruppe zu finden ist.

Es sind beliebig viele dieser beiden Bausteine in das Simulationsmodell integrierbar. Die Bausteine besitzen jeweils maximal neun, minimal einen Signalanschluss.

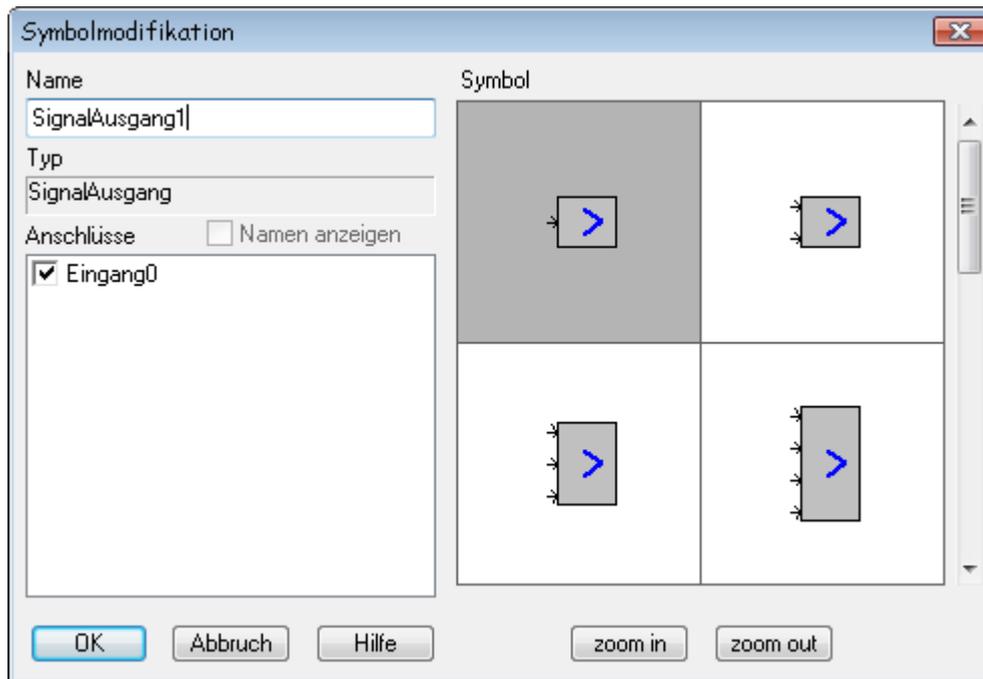


Abbildung 9.19 - Symbolmodifikation SignalAusgang mit 9 Ausgängen

Die Aus- bzw. Eingänge der Bauteile SIGNALEINGANG und SIGNALAUSGANG werden dann mit den Ein- bzw. Ausgängen des Modells verbunden.

Die Anschlüsse eines jeden Signaleingangs- und Signalausgangsbauteils sollten möglichst lückenlos belegt werden. Das Modell wird sonst zwar ordnungsgemäß erstellt, die ausgelassenen Verbindungen werden jedoch nicht im Datenaustauschvektor mit dem externen Programm berücksichtigt. Die Positionen der Signale auf DSH*plus*-Seite stimmen hierdurch dann nicht mehr mit der Signalposition im Datenaustauschvektor, der fortlaufend besetzt ist, überein.

Die Erstellung des kompletten Modells ist nun abgeschlossen.

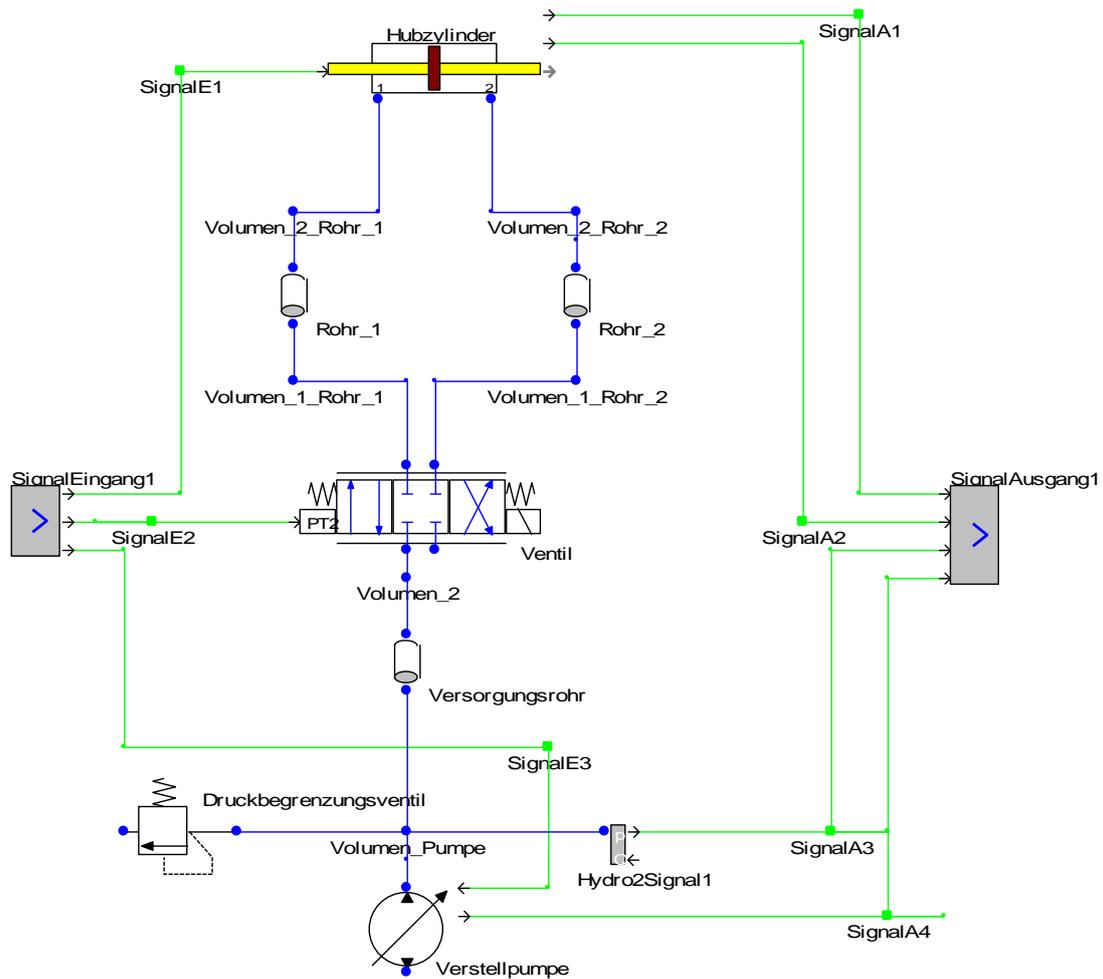


Abbildung 9.20 - Kompletmodell für den Export

Modellparametrierung

Nach dem Beenden des Bearbeitungsmodus und der Erstellung des Modells identifiziert DSHplus das Vorhandensein von Signaleingangs- und Signalausgangsbauteilen im Simulationsmodell und das Dialogfenster **Hardware Interface Auswahl** öffnet sich. Hier wird das gewünschte Interface gewählt und das Fenster zunächst minimiert.

Als nächstes erfolgt die Parametrierung der neuen Bauteile und eventuell der Knoten. Dabei besitzen die Bauteile SIGNAL EINGANG und SIGNAL AUSGANG einen eigenen Parametrierungsdialog.

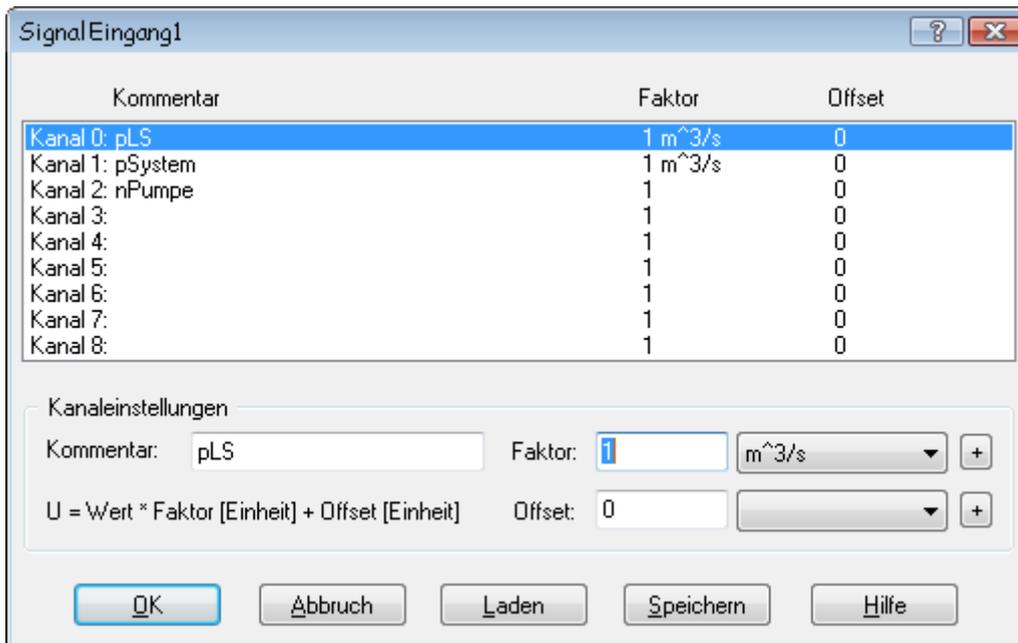


Abbildung 9.21 - Parametrierungsdiallog Bauteile SIGNALEINGANG und SIGNALAUSGANG

Der Dialog zeigt alle Kanäle (Ein- bzw. Ausgänge) an. Der Faktor ist immer mit 1 und ohne Einheit vorbelegt, der Offset immer mit 0 und ebenfalls ohne Einheit. Eine Änderung der Offsetwerte ist angebracht, wenn z. B. Koordinatenunterschiede zwischen dem externen System und dem DSHplus-Modell vorhanden sind. Faktoränderungen dienen zur Skalierung der Signale. Beide Parameter sind in allen verfügbaren Einheiten definierbar. Zudem können kanalspezifische Kommentare hinzugefügt werden, die eine schnelle Identifizierung der verschiedenen Signale ermöglichen.

Nach Abschluss der Bauteil- und eventuellen Knotenparametrierung und Validierung der eingegebenen Daten kann entweder eine interaktive Simulation oder ein direkter Datenexport an ein Zielprogramm erfolgen.

Auswahl eines Hardware-Interfaces

Mit der Wahl des auf den DSHplus-OIF basierenden Standardinterfaces kann die interaktive Simulation gestartet werden. Hierzu wird das DSHplus-Projekt geöffnet, und es erscheint automatisch der Dialog **HARDWARE INTERFACE AUSWAHL**.

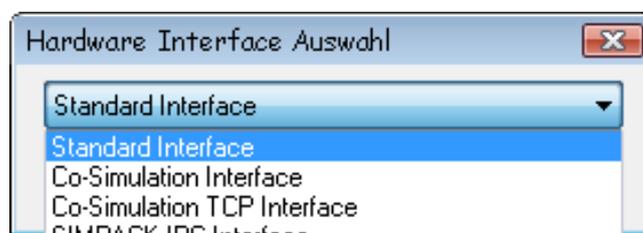


Abbildung 9.22 - Dialogfenster Hardware Interface Auswahl

Das *Standardinterface* ist generell in Form eines Schiebereglers implementiert.

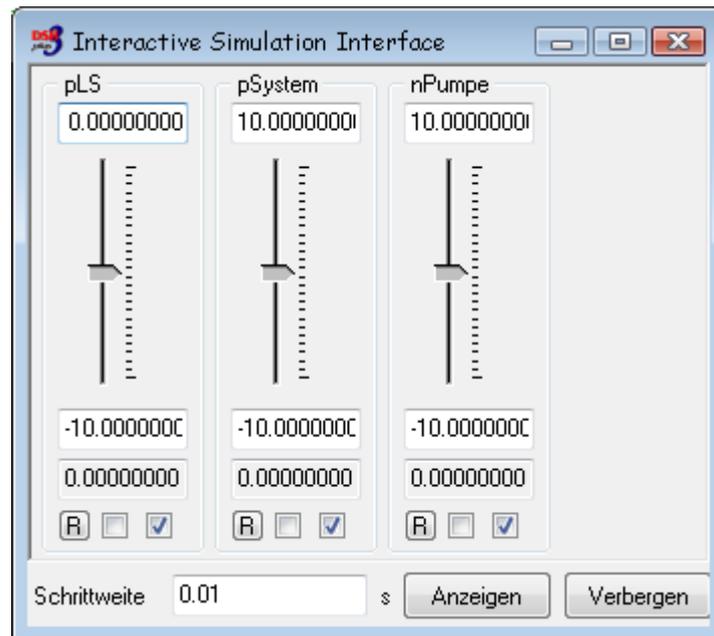


Abbildung 9.23 - Interaktives Simulations-Interface (Standardinterface)

Mittels der Schieberregler ändert der Benutzer einfach und schnell die in das Simulationsmodell eingelesenen Größen. Die Anzahl der Schieber ist identisch mit der Anzahl der verbundenen Signalausgänge der Bauteile SIGNALEINGANG.

Das **Aktualisierungsintervall** gibt die Zeitspanne wieder, nach der die Schiebereglerstellungen an das Modell übermittelt werden. Standardmäßig ist eine automatische Skalierung von -10 bis $+10$ für alle Regler eingestellt, die jedoch leicht zu ändern ist. Sie wird auch beim Öffnen des Projektes automatisch eingestellt.

Die anderen Interfaces werden benutzt, wenn das Modell in *DSHplus* berechnet wird und über die Schnittstelle von außen Werte vorgegeben oder abgegriffen werden (Co-Simulation).

Test des aufbereiteten Modells

Beim Test des Modells mit den voreingestellten Parametern der Simulationsrechnung ist der Ablauf der Simulation in der Regel so schnell, dass ein interaktives Eingreifen kaum möglich ist. Die Simulation ist oft schon vorbei, bevor der erste Schieberegler betätigt wird.

Der Anwender kann zur Verlangsamung der Berechnung durch Verändern der Simulationsparameter eine für ihn optimale Simulationsgeschwindigkeit einstellen. Dabei ist es zweckmäßig, einen Integrator mit fester Schrittweite zu wählen und dessen Schrittweite relativ klein ($1.0e-6$) zu parametrieren. Der Simulationsablauf verlangsamt sich und ein sinnvolles Eingreifen ist jetzt möglich.

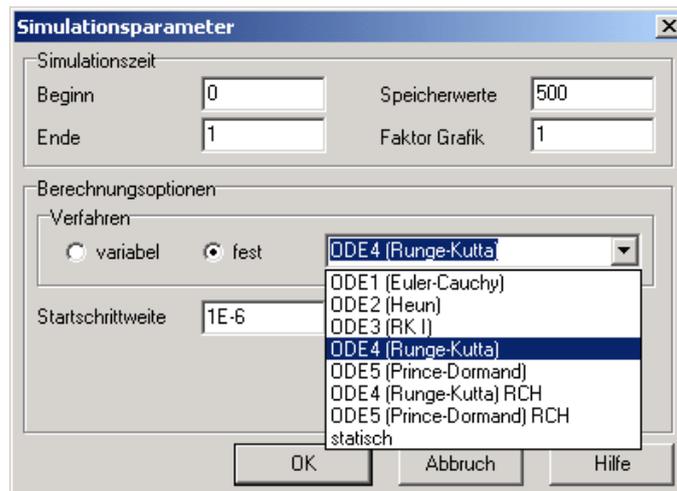


Abbildung 9.24 - DSHplus Simulationsparameter bei einer interaktiven Berechnung

Lassen sich dadurch immer noch nicht genug Systemveränderungen während der Simulation testen, so kann entweder eine längere Simulationszeit eingestellt werden oder eine *kontinuierliche Simulation* gestartet werden. Eine kontinuierliche Simulation erlaubt es dem Benutzer, ohne Abbruch der Simulation fortlaufend aktiv einzugreifen und somit über die Veränderungen der Schieberegler unterschiedlichste Systembedingungen zu testen. Nach dem Start wird die Zeitachse der Ergebnisgrafiken zunächst mit den eingestellten Simulationsparametern Start- und Endzeit initialisiert. Ist die eingestellte Zeit abgelaufen, verschiebt sich die Zeitachse um den voreingestellten Betrag und die Ergebnisse werden wieder neu angezeigt.

Export des DSHplus-Modells

Der Export des DSHplus-Simulationsmodell ist nötig, um entweder gekoppelte Simulationen im Embedded-Modus oder als Co-Simulationen zwischen zwei Programmen durchzuführen. Das Modell wird zusammen mit seinen Parametern aus der DSHplus-Struktur herausgelöst, die Schnittstellenfunktionen werden ergänzt und alles zusammen separat abgelegt.

Der Modellexport wird über das Menu **MODELL-MODELL EXPORTIEREN** aufgerufen. Hier wird nun die Zielanwendung und der für den Export gewünschte Parametersatz gewählt.

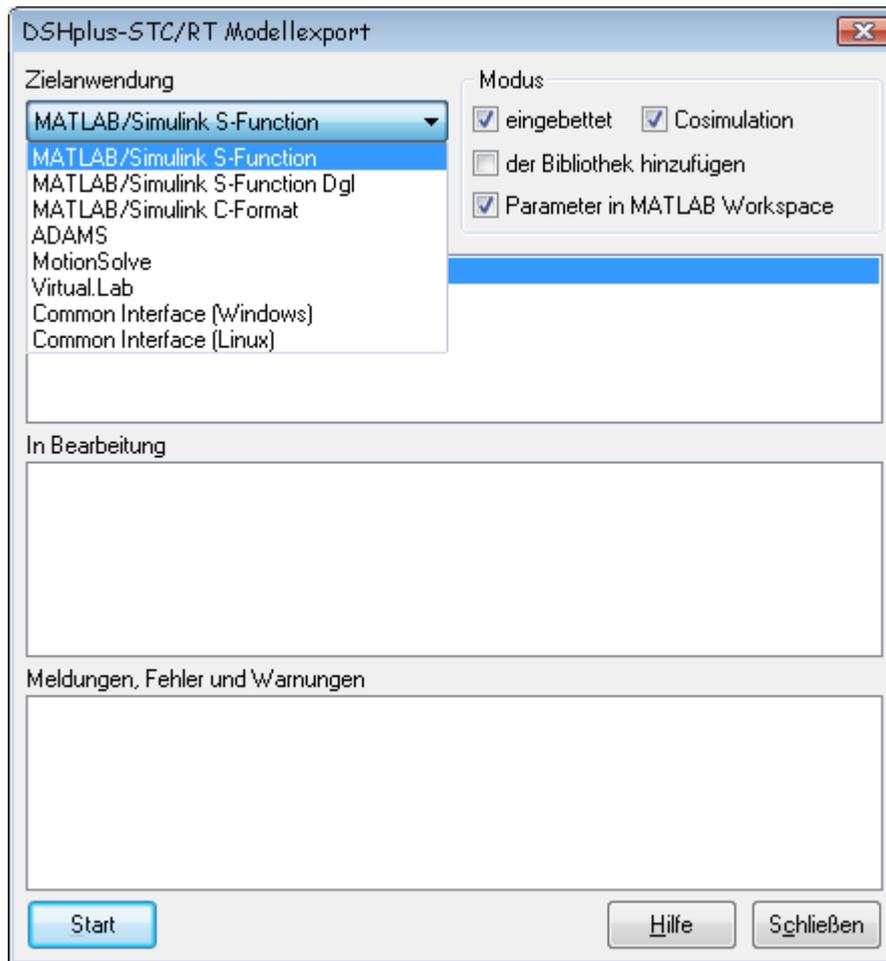


Abbildung 9.25 - Dialogfenster Modellexport vor dem Export

In der Liste **In Bearbeitung** werden Ablaufmeldungen des Exportvorganges angezeigt; eventuelle Meldungen, Warnungen oder Fehler des Compilers und des Linkers werden in der Liste **Meldungen, Fehler und Warnungen** angezeigt. Nach dem Ende des Exportierens zeigt das Modellexportfenster in der Liste **In Bearbeitung** an, ob der Vorgang erfolgreich war.

9.2.3 Beispielhafte Implementierung der Schnittstelle in SIMPACK

Umgesetzte Anforderungen (SIMAT_2)

Im Rahmen des GUSMA-Projekts wurden vielfältige Anforderungen gemeinschaftlich erarbeitet. Die Anforderungen an die Softwarepakete wurden, im Falle von SIMPACK, fast vollständig umgesetzt. Bei den implementierten Erweiterungen der Software handelt es sich um folgende Punkte:

- Festlegen der Freigegebenen Parameter (SubVars) in SIMPACK
- Anzeigen der Ein- und Ausgangskanäle im SIMAT_2 Block
- Anzeigen des SIMPACK Logos im SIMAT_2 Block
- Anzeigen und Modifizieren der SubVars im SIMAT_2 Block
- Schreiben der SubVars in den MATLAB Workspace als Struktur „SPCK“
- Optional: Schreiben der SubVars in das MATLAB Arbeitsverzeichnis als .m-Datei
- Optional: Zurücksetzen der SubVars auf die Default-Werte aus dem SIMPACK Modell
- Optional: Zurückschreiben der SubVars nach SIMPACK
- Optional: Automatisches Durchführen der „Measurements“ nach der Co-Simulation

Diese umgesetzten Erweiterungen werden im Folgenden präsentiert und ausführlich beschrieben.

Vorbereitung des Beispiel-Modells in SIMPACK

Anhand eines einfachen Beispielmodells werden die im Rahmen des GUSMA-Projekts durchgeführten Erweiterungen vorgestellt. Die folgende Abbildung zeigt das Beispielmodell in SIMPACK.

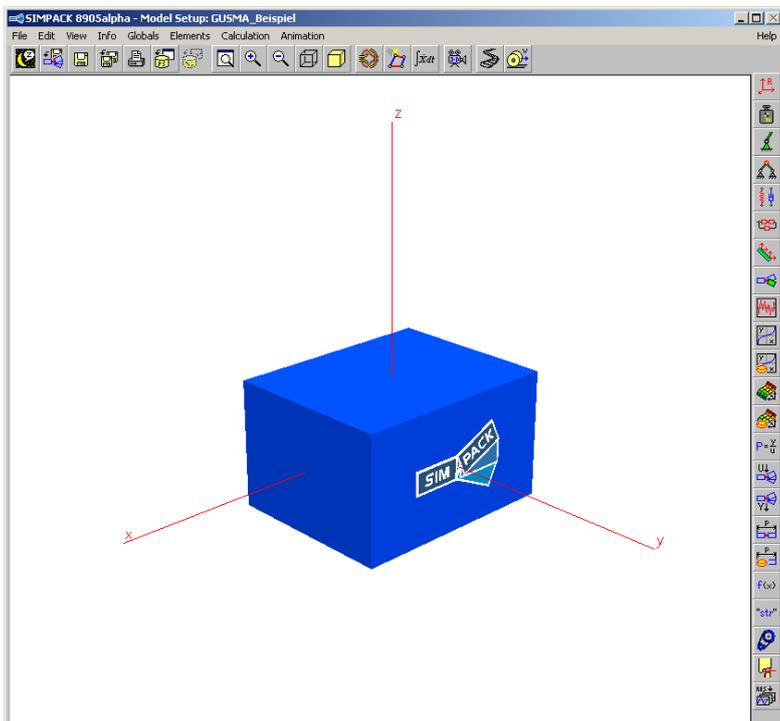


Abbildung 9.26 – Simpack Beispielmodell

Es handelt sich um den Standardkörper in Form von dem blauen Würfel. Dem Körper wurde ein Freiheitsgrad gegeben: Freie Bewegung in z-Richtung.

Damit eine Co-Simulation mit MATLAB durchgeführt werden kann, sind mindestens ein Eingangs- und ein Ausgangskanal zu definieren. Die folgende Abbildung zeigt den definierten Eingangskanal „\$U_Force“, welcher z.B. mit einem Kraftelement auf den Würfel wirken könnte. Der Wert der Kraft würde von MATLAB bereitgestellt werden.

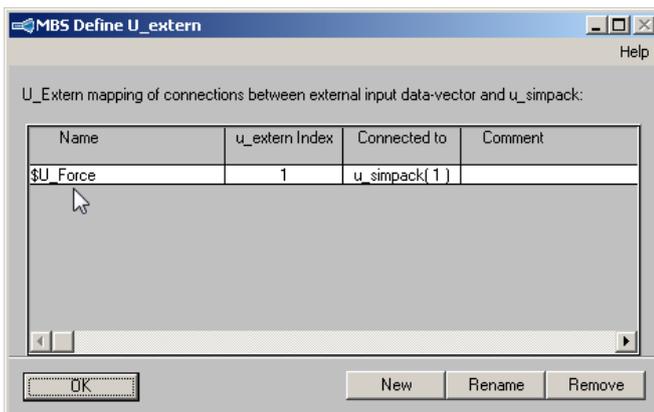


Abbildung 9.27 - simpack_uextern

Die zwei beispielhaft definierten Ausgangskanäle „\$Y_Position“ und „\$Y_Velocity“ sind in den folgenden Abbildungen dargestellt.

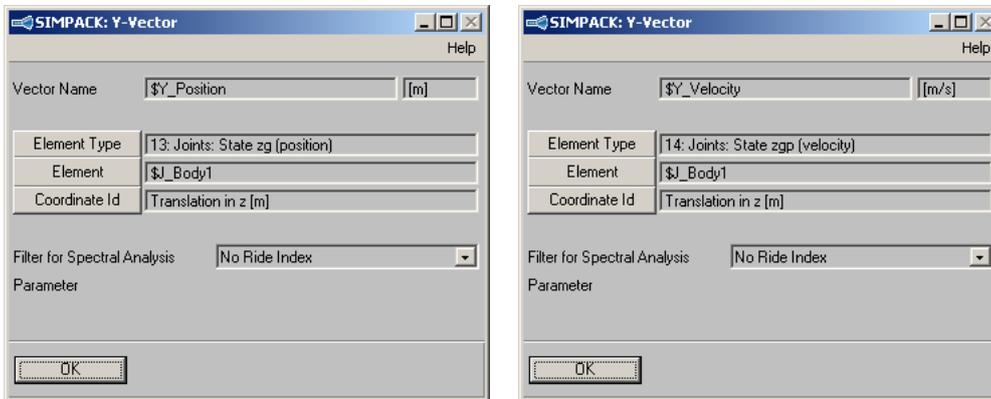


Abbildung 9.28 – Simpack yvector1 (links) und Simpack yvector2 (rechts)

Die Ausgangskanäle übergeben in diesem Fall die Position und Geschwindigkeit des Körpers an MATLAB.

Des Weiteren wurden in dem SIMPACK Beispielmmodell drei Parameter, genannt „Substitution Variables“ oder kurz „SubVars“, definiert. Auf diese wird im nächsten Kapitel näher eingegangen.

Festlegen der Freigegebenen Parameter in SIMPACK

Wie bereits gesagt, werden in der SIMPACK Fachsprache Parameter „SubVars“, eine Abkürzung für „Substitution Variables“, genannt.

Beim dem, mit der Version SIMPACK 8094 standardmäßig ausgelieferten SIMAT Block, können keine SubVars an den SIMAT Block übergeben werden.

Mit der im Rahmen des GUSMA Projekts weiterentwickelten Version SIMPACK 8905 können beliebig viele SubVars im Mechanikmodell durch den SIMPACK Anwender freigeschalten werden. Diese freigegebenen Parameter/SubVars stehen dann im SIMAT_2 Block zur Verfügung.

Nach SIMAT_2 exportiert werden folgende Eigenschaften der SubVars:

- Name
- Wert
- Kommentar

Die nachstehende Abbildung zeigt drei in SIMPACK definierte SubVars.

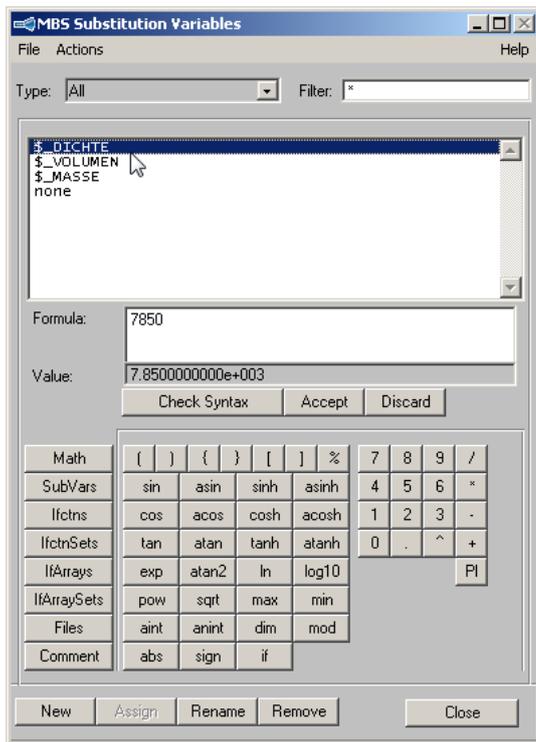


Abbildung 9.29 – Subvars

Es handelt sich dabei um die Parameter „\$_DICHTE“, „\$_VOLUMEN“ und „\$_MASSE“.

Zum Freischalten der SubVars muss der „Type:“ auf „Exportable SubVars“ umgeschaltet werden. Alle für die Freigabe zulässigen SubVars werden dann in der Liste aufgeführt, wie in der folgenden Abbildung ersichtlich.

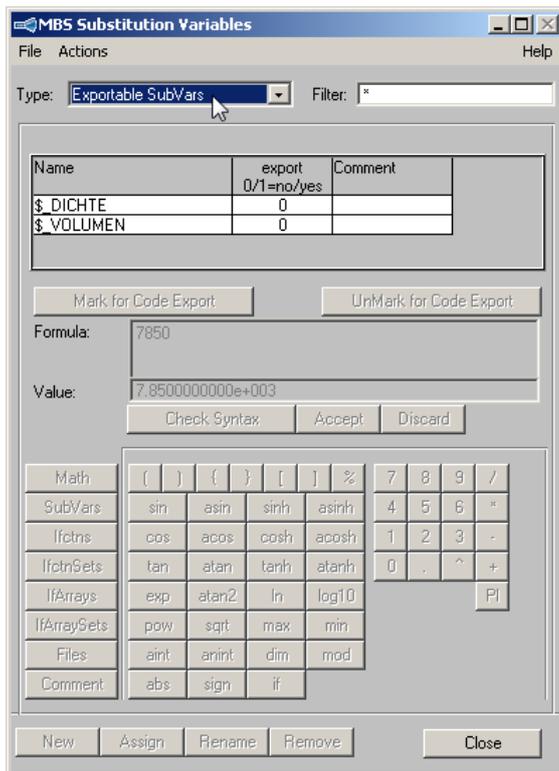


Abbildung 9.30 – Exportierbare Subvars

Wie in der **Abbildung 9.30** erkennbar, ist die SubVar „\$_MASSE“ nicht in der Liste der exportierbaren Parameter aufgeführt. Dies liegt daran, dass dieses SubVar sich aus einer Formel (Masse = Dichte * Volumen) berechnet. Formel-behafteten Parameter können zurzeit nicht exportiert werden. Siehe dazu auch das entsprechende Kapitel unter den „Offenen Punkten“.

In einem nächsten Schritt markiert der SIMPACK Anwender diejenigen SubVars, welche er für den SIMAT_2 Block freigeben möchte. Dazu muss die Taste „Mark for Code Export“ gedrückt werden (siehe **Abbildung 9.31**).

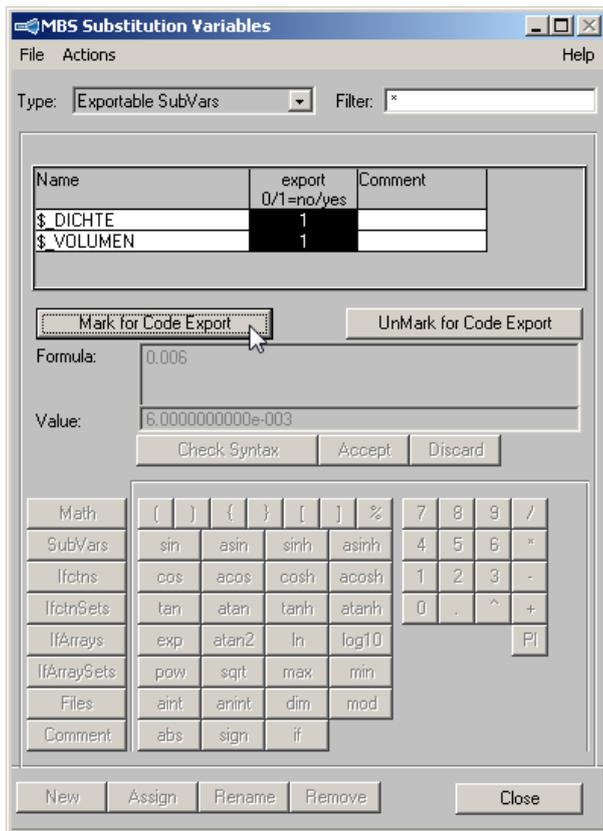


Abbildung 9.31 – Markierte Exportierbare Subvas

Ein manuelles Eintragen einer „1“ in der „export“ Spalte, wird nicht in die SIMPACK Modelldatei (<model_name>.sys) geschrieben, da das Dialogfenster zur Zeit nur über einen „Close“ Knopf und nicht über einen „Apply“ oder „Ok“ Knopf verfügt. In einer späteren Version von SIMAT_2 soll berücksichtigt werden, dass der Wert auch manuell eingetragen werden kann.

Alternativ zum Knopf „Mark for Code Export“, kann das Kontextmenü (mittels rechtem Mausklick) für das Markieren der SubVars verwendet werden. Die folgende Abbildung zeigt das Kontextmenü.

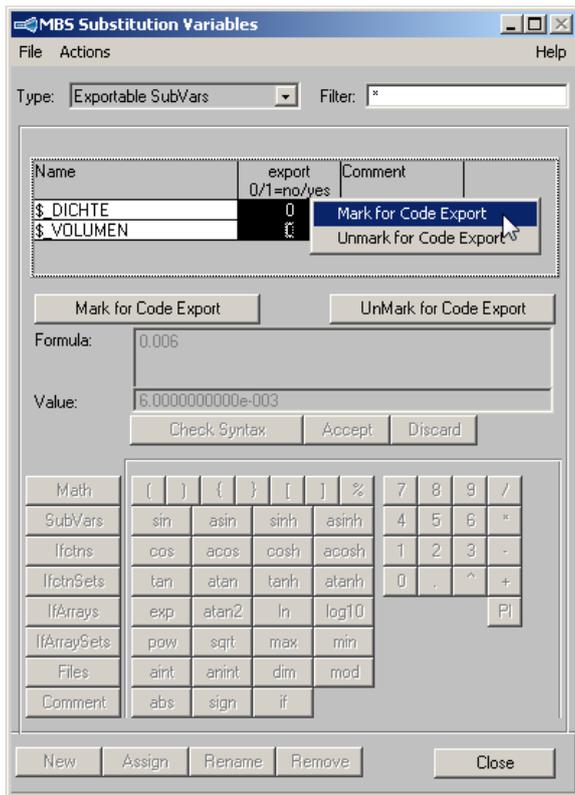


Abbildung 9.32 –Kontextmenü der exportierbaren Subvars

Des Weiteren werden, wie bereits erwähnt, die Kommentare der SubVars mitfreigeschalten. Die nachstehende Abbildung zeigt das Eintragen des Kommentares „Dichte von Stahl“ für die SubVar „\$_DICHTE“.



Abbildung 9.33 - Kommentar für Subvars

Nach Eingabe des Kommentars muss dieser durch Drücken der „Accept...“ Taste bestätigt werden. Der Kommentar wird daraufhin auch bei für „Type:“ gleich „Exportable SubVars“ in der dritten Spalte erscheinen.

Damit eine Co-Simulation mit MATLAB durchgeführt werden kann, muss jetzt der SIMPACK „Server for Matlab“ gestartet werden. Die folgende Abbildung stellt das manuelle Starten des Servers durch den SIMPACK Anwender beispielhaft dar.

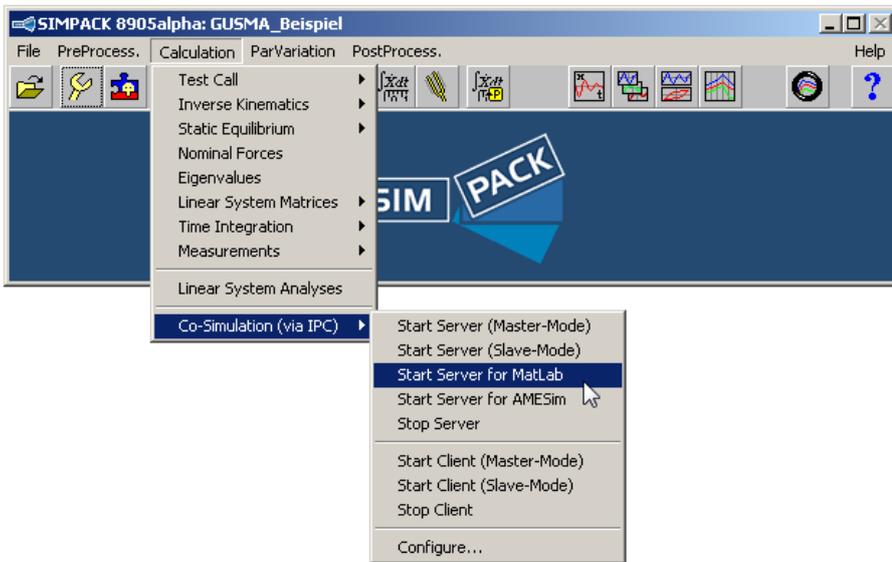


Abbildung 9.34 – Manueller Start eines Servers zu Matlab

Hiermit ist der für die GUSMA-Plattform notwendige Modellierungsbedarf in SIMPACK vollständig vorgestellt.

Anzeigen der Ein- und Ausgangskanäle am SIMAT_2 Block

Nach der Abhandlung der neuen Möglichkeiten für den SIMPACK Anwender, wird im Folgenden auf die Erweiterungen, die den SIMAT_2 Block und somit den MATLAB Anwender bzw. den Nutzer der GUSMA Plattform betreffen, eingegangen.

Wenn im Folgenden vom „MATLAB Anwender“ die Rede ist, ist dies synonym für „Nutzer der GUSMA Plattform“.

Um den SIMAT_2 Block anstelle des standardmäßig verwendeten SIMAT Blocks zu verwenden, muss in einem ersten Schritt der „Pfad“ neu gesetzt werden (MATLAB: File → Set Path...). In der nachstehenden Abbildung steht der Pfad mit dem Ordner „simat2“ an oberster Stelle in der „MATLAB search path:“ Liste.

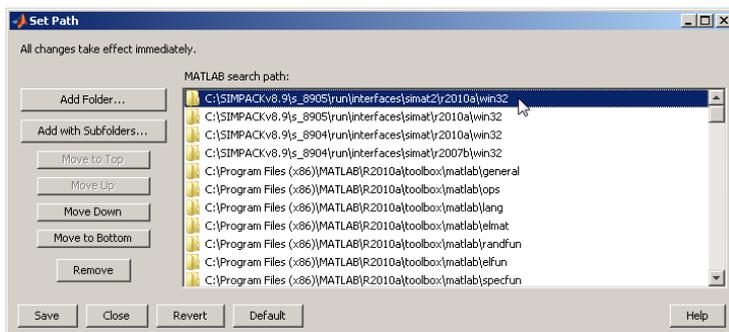


Abbildung 9.35 - Pfadsetzen in Matlab für SIMAT 2 Block

Somit wird, wenn in der MATLAB Kommandozeile der Befehl „simat“ eingegeben wird, automatisch der im Rahmen des GUSMA-Projekts entwickelte SIMAT_2 Block geladen und nicht mehr der ursprüngliche SIMAT Block.

Der neue SIMAT_2 Block zeigt, nach erfolgreichem Initialisieren (durch zweimaliges Öffnen und Schließen des Blocks), die Ein- und Ausgänge mit Namen an, wie in der folgenden Abbildung ersichtlich.

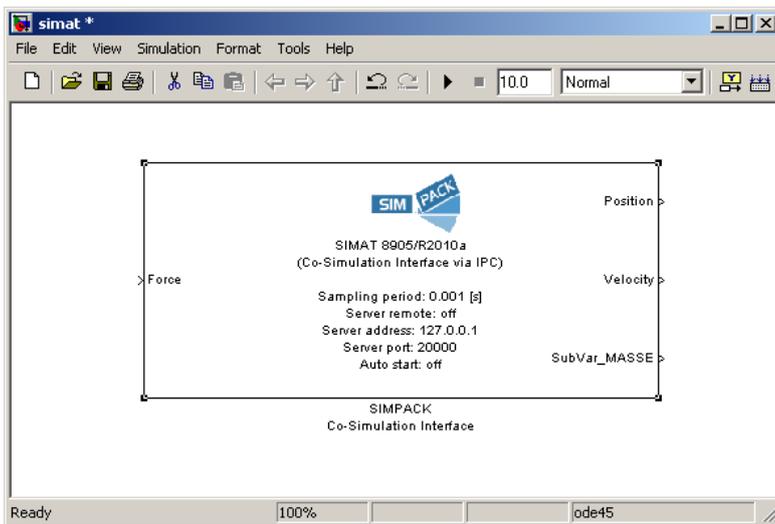


Abbildung 9.36 - SIMAT 2 Block in Simulink

In der Abbildung ersichtlich sind der Eingang „Force“ und die Ausgänge „Position“ und „Velocity“. Auf den dritten Ausgang „SubVar_MASSE“ wird in dem späteren Kapitel mit „Beantworteten Fragen“ bei der Frage zu „SubVars mit Formeln“ näher drauf eingegangen.

Anzeigen des SIMPACK Logos im SIMAT_2 Block

Eine weitere Anforderung der GUSMA-Plattform war das Anzeigen des SIMPACK Logos im SIMPACK Co-Simulationsblock. Die vorherige Abbildung zeigt, dass das SIMPACK Logo im SIMAT_2 Block dargestellt wird.

Das gut sichtbare Logo ermöglicht ein schnelles Zuordnen der Blöcke zu den jeweiligen Spezialprogrammen innerhalb von MATLAB und erleichtert die Arbeit des Anwenders der GUSMA-Plattform.

Anzeigen und Modifizieren der SubVars im SIMAT_2 Block

Durch Doppelklick auf den SIMAT_2 Block erscheinen die Block Eigenschaften, wie in der folgenden Abbildung dargestellt.

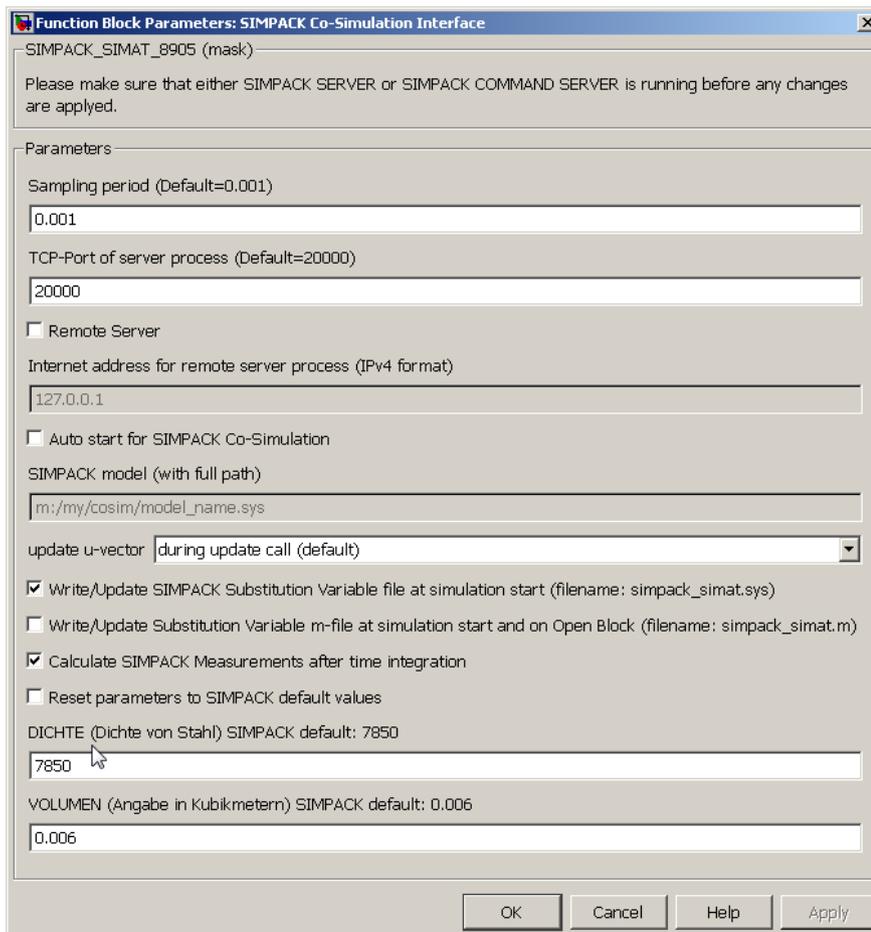


Abbildung 9.37 - Blockeigenschaften des SIMAT_2 Blocks

Im unteren Teil der Abbildung sind folgende zuvor vom SIMPACK Anwender freigegebene SubVars angezeigt:

- DICHTE
- VOLUMEN

Nach dem Namen der SubVar („DICHTE“) steht in Klammern, falls es einen entsprechenden Eintrag in SIMPACK gab, der Kommentar („Dichte von Stahl“). Darauf folgt der ursprüngliche Wert, welcher im SIMPACK Modell für die SubVar verwendet wurde („SIMPACK default: 7850“). Im Anschluss an diese Angaben folgt eine weiß hinterlegte Eingabezeile, in der standardmäßig der Originalwert aus dem ursprünglichen SIMPACK Modell für die SubVar eingetragen ist („7850“).

Der MATLAB Anwender kann nun die Werte, welche für die SubVars verwendet werden sollen so belassen, wie ursprünglich im SIMPACK Modell definiert, oder einen neuen Wert eintragen.

Beim Schließen des Dialogfensters mit „OK“ werden die in den Eingabezeilen eingetragenen Werte für die SubVars in die im MATLAB Workspace befindliche Struktur „SPCK“ geschrieben; diese wird im nächsten Kapitel näher beschrieben.

Schreiben der SubVars in den MATLAB Workspace als Struktur „SPCK“

Wie bereits im vorherigen Kapitel erwähnt, werden die im SIMAT_2 Block eingetragenen Werte für die SubVars an die Struktur „SPCK“ in den MATLAB Workspace übergeben. Im oben beschriebenen Beispiel wird die „SPCK“ Struktur im MATLAB Workspace u.a. folgende Einträge enthalten:

MATLAB Workspace, Struktur „SPCK“:

SPCK.subvar.DICHTE.comment = 'Dichte von Stahl'

SPCK.subvar.DICHTE.value = 7850

SPCK.subvar.VOLUMEN.comment = 'Angabe in Kubikmetern'

SPCK.subvar.VOLUMEN.value = 0.0120

Beim Starten einer Co-Simulation werden dann immer die aktuell in der „SPCK“ Struktur befindlichen Werte für die SubVars verwendet.

Wenn der Anwender in der „SPCK“ Struktur nachträglich den Wert für die SubVar „VOLUMEN“ ändert, wird dieser geänderte Wert verwendet und nicht der zuletzt im SIMAT_2 Dialogfenster eingetragene. In der folgenden Abbildung hat der Anwender den Wert für das Volumen in der „SPCK“ Struktur von „0.012“ auf „0.024“ erhöht.

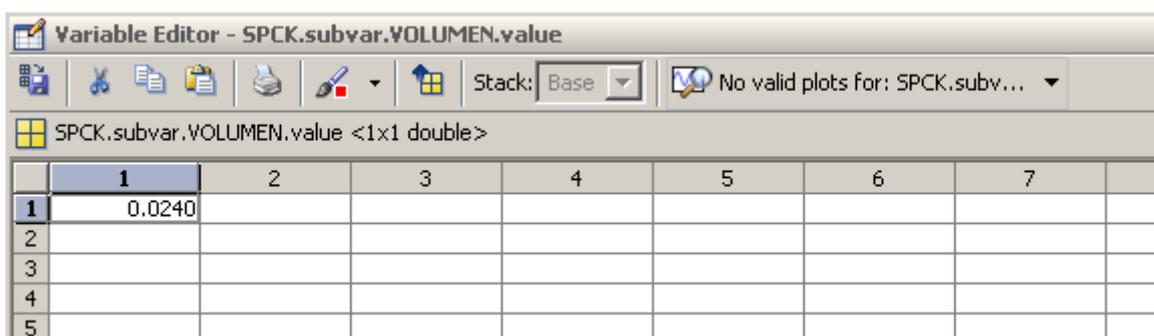


Abbildung 9.38 – Manuelles Verändern der Variablen

Wird jetzt eine Co-Simulation gestartet, wird der Wert „0.024“ für das Volumen im SIMPACK Modell verwendet werden. Ein erneutes Öffnen des SIMAT_2 Blocks durch Doppelklick zeigt diesen Wert nun auch dort an. Wichtig ist es zu wissen, dass für die Co-Simulation immer der zur Zeit des Beginns der Co-Simulation in der im MATLAB Workspace befindlichen „SPCK“ Struktur stehende Wert verwendet wird.

Schreiben der SubVars in das MATLAB Arbeitsverzeichnis als .m Datei

Beim Öffnen des SIMAT_2 Blocks werden die freigegebenen SubVars auf Wunsch des MATLAB Anwenders als MATLAB typische .m Datei mit dem Namen „simpack_simat.m“ in das aktuelle MATLAB Arbeitsverzeichnis geschrieben. Die .m Datei wird ausgeschrieben, wenn die Option „Write/Update Substitution Variable m-file at simulation start and on Open Block (filename: simpack_simat.m)“ aktiviert ist, wie in der folgenden Abbildung dargestellt.

Die „simpack_simat.m“ Datei wird unmittelbar nach Start der Co-Simulation in das aktuelle Arbeitsverzeichnis von MATLAB geschrieben. Die in die .m Datei ausgeschriebenen SubVars haben das von der „SPCK“ Struktur bekannte Format:

MATLAB Arbeitsverzeichnis, .m Datei:

```
SPCK.subvar.DICHTE.value      = 7.850000000000000e+003;
SPCK.subvar.DICHTE.comment    = sprintf('Dichte von Stahl');
SPCK.subvar.VOLUMEN.value     = 2.400000000000000e-002;
SPCK.subvar.VOLUMEN.comment   = sprintf('Angabe in Kubikmetern');
```

Die Werte für die SubVars können also alternativ vom MATLAB Anwender in der .m Datei geändert werden. Allerdings muss die .m Datei vor der Co-Simulation ausgeführt werden, damit sich die Werte der „SPCK“ Struktur im MATLAB Workspace aktualisieren.

Wie bereits geschildert, wird immer mit dem aktuellen Wert der „SPCK“ Struktur aus dem MATLAB Workspace gerechnet. Mithilfe von ‘Callbacks’ (MATLAB: SIMAT_2 Block → File → Model Properties) kann das Aufrufen der .m Datei vor Ausführen der Co-Simulation im SIMAT_2 Block automatisiert werden.

Zurückschreiben der SubVars nach SIMPACK

Per Knopfdruck können die bei der Co-Simulation verwendeten SubVar-Werte in eine externe Datei mit dem Namen „simpack_simat.sys“ ausgeschrieben werden. Diese Datei enthält die SubVars mit ihren Werten in dem bekannten, von SIMPACK lesbaren, Format:

MATLAB Arbeitsverzeichnis, SIMPACK .sys Datei:

```
subvar.str ( $_DICHTE )      = '7.850000000000000e+003'
subvar.txt ( $_DICHTE )      = 'Dichte von Stahl'
subvar.str ( $_VOLUMEN )     = '2.400000000000000e-002'
subvar.txt ( $_VOLUMEN )     = 'Angabe in Kubikmetern'
```

Das Ausschreiben der aktuellen Werte in die .sys Datei erfolgt, wenn die Option „Write/Update SIMPACK Substitution Variable file at simulation start (filename: simpack_simat.sys)“ im SIMAT_2 Block gesetzt ist.

Die .sys Datei wird erst unmittelbar nach dem Start der Co-Simulation rausgeschrieben. Dieses Format bzw. die gesamte .sys Datei kann vom SIMPACK Anwender in seinem SIMPACK Modell verwendet werden. Dadurch kann sichergestellt werden, dass der SIMPACK Anwender auch über die aktualisierten SubVar Werte verfügt und diese in seinem Modell, außerhalb der Co-Simulation, nutzt.

Automatisches Durchführen der „Measurements“ nach der Co-Simulation

Per Knopfdruck können die SIMPACK „Measurements“ automatisch im Anschluss an die Co-Simulation durchgeführt werden. Durch diese Option entfällt das bisher händische Nachschalten der „Measurements“ in SIMPACK durch den SIMPACK Anwender. Das automatische Erstellen der „Measurements“ kann durch Setzen der Option „Calculate SIMPACK Measurements after time integration“ im SIMAT_2 Block eingeschalten werden, wie folgende Abbildung zeigt. Auf diese Weise können beispielsweise die Ergebnisse aus der Co-Simulation im Post-Processing-Tool von SIMPACK am MKS-Modell visualisiert werden.

9.3 Erweiterte Beschreibung der Simulationsmodelle und –ergebnisse

9.3.1 Hydraulischer Federungskreis

Parameter für die Co-Simulation

Für die GUSMA-Plattform sind einige Parameter zur Abstimmung, aber auch zur gezielten Manipulation des Simulations-Systems identifiziert worden.

# Kolbendurchmesser.....	50mm (Auslieferungszustand)
# Stangendurchmesser.....	40mm (Auslieferungszustand)
# Speichernennvolumen an der Kolbenseite 1.....	0,75L (Auslieferungszustand)
# Speichernennvolumen an der Kolbenseite 2.....	1,4L (Auslieferungszustand)
# Speichernennvolumen an der Ringseite.....	0,75L (Auslieferungszustand)
# Speichervorspanndruck der Kolbenseite.....	45bar (Auslieferungszustand)
# Speichervorspanndruck der Stangenseite.....	80bar (Auslieferungszustand)
# Lastdruck an der Stangenseite.....	100bar (Einstellbar)
# Lastdruck an der Kolbenseite.....	116,873bar (Einstellbar)
# Durchmesser Blende 1.....	3mm (Auslieferungszustand)
# Durchmesser Blende 2.....	2mm (Auslieferungszustand)
# Durchmesser Blende Ringseite.....	7mm (Systembedingt)
# Durchmesser Blende Kolbenseite.....	6,5mm (Systembedingt)
# Haftreibung.....	370N (errechnet aus Versuchen)
# Gleitreibung.....	360N (errechnet aus Versuchen)

In **Abbildung 9.39** sind die Positionen der Blenden und Speicher kenntlich gemacht worden. Alle anderen Parameter sind durch ihre Benennung selbsterklärend.

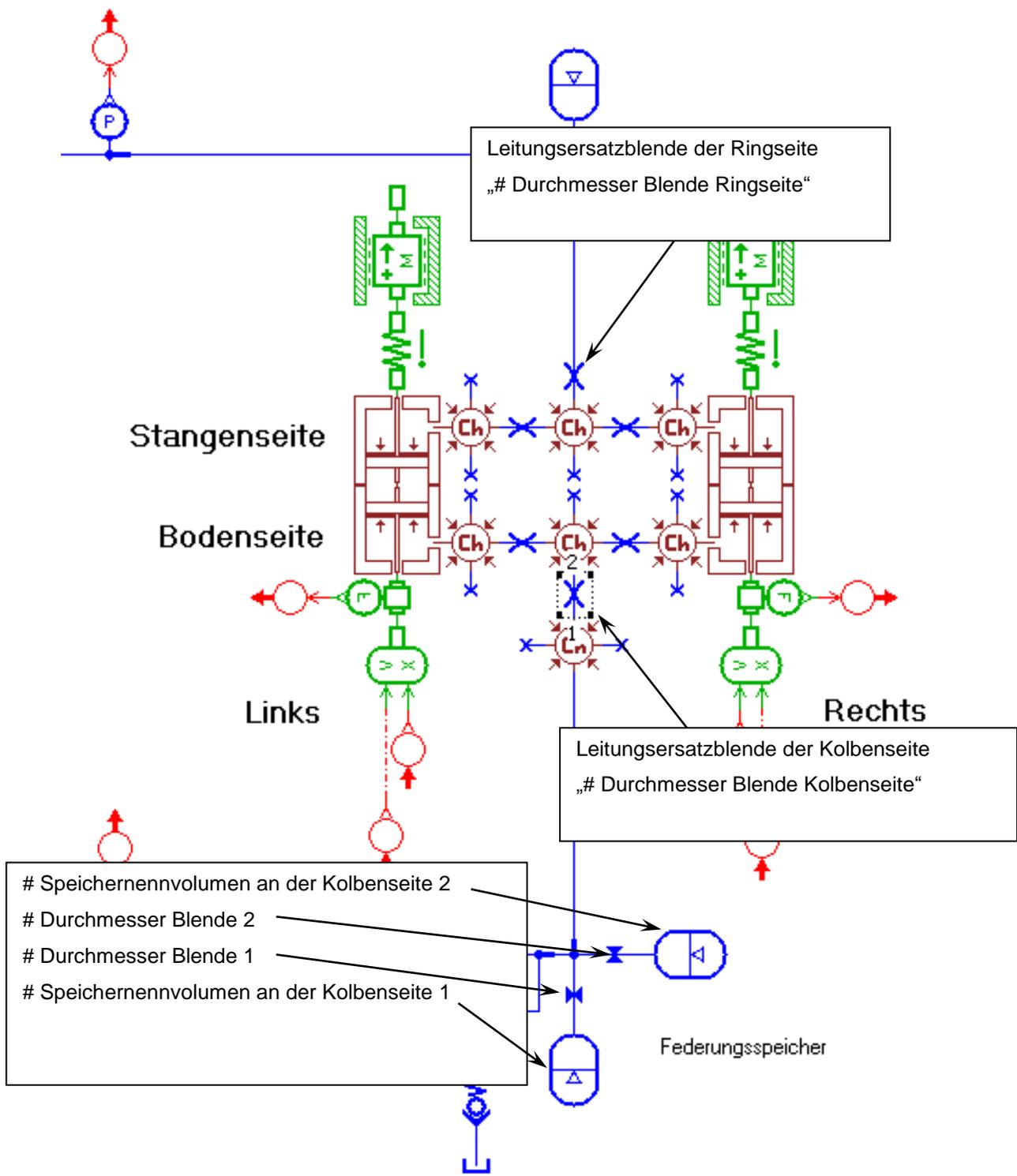


Abbildung 9.39 - Position der Blenden

9.3.2 Hydraulische Druckversorgung

Variablen / Parameter

In der folgenden Tabelle sind die Parameter und Startwerte der Variablen aufgelistet, mit denen das Modell bedatet ist.

Parameter

Volumen p1

E_Oel	14000 bar
Volumen	1 l
AnteilUngelLuft	0,1 %
PolytopenExponent	1,1

SigBegr

Signalmax	100 %
Signalmin	-2 %

Blende1

Volumenstrom	30 l/min
Druckdifferenz	5 bar

PT1_PReg

Kp	10 %/(bar)
T	0,1 s

LS-Offset

Faktor	1
Offset	20 bar

DBVstatisch1

Oeffnungsdruck	210 bar
Druckdifferenz	5 bar
Volumenstrom	30 l/min

VP1

Schluckvolumen	45 cm ³
InterneLeckage	0,001 l/min/bar
ExterneLeckage	0 l/min/bar
ViskoseDaempfung	0,0001 Nms
EthaHM	1
EingangMax	100 %
EingangMin	-100 %

S1

Faktor1	-1
Faktor2	1

Faktor3

1

Variablen

p1.Druck

0 bar

Sig_Pumpe.Wert

0 %

VP1.DrehzahlPT1

0

p_System.Wert

0 bar

p_LS.Wert

0 bar

Q_Pumpe.Wert

0 l/min

SigRegl.Wert

0 %

n_VP1.Wert

0 1/min

9.4 Untersuchung relevanter Verluste in der hydropneumatischen Achsfederung

9.4.1 Untersuchung der Reibung

Die Reibungsmessungen wurden auf einem Hydropuls-Prüfstand durchgeführt und die zugehörigen Randbedingungen wurden so ausgewählt, dass die Zuordnung der Drücke so abgebildet wird, wie es in der Realität der Fall wäre. Somit wurde das Federungssystem auf dem Hydropulser genau so aufgebaut, wie es im Traktor verbaut ist. Aus dieser Anordnung werden die schwankenden Druckpaare realitätsgetreu nachgebildet. Dieses Verhalten ist wichtig, weil die Reibung der Dichtungen nicht nur von der Höhe des beaufschlagenden Druckes, sondern im großen Umfang auch von der vorherrschenden Druckdifferenz abhängt.

Die Anregung erfolgte mittels eines Dreieck-Signals (**Abbildung 9.40**) mit einer Amplitude von $\pm 30\text{mm}$ und einer Periodendauer von 10s. Bei dieser Anregung ist es gegeben, dass die Verfahrgeschwindigkeit konstant bleibt und keine zusätzlichen dynamischen Effekte auftreten.

In der **Abbildung 9.41** ist ein beispielhafter Kräfteverlauf zu sehen. Die blaue Kurve zeigt die gemessene Kraft an der Kraftmessdose zwischen dem Hydropulser und dem Prüfling. In der roten Kurve ist die Kraft dargestellt, die mittels des Speicherdruckes errechnet wurde. Diese Kraft stellt die wahre Federkennlinie des Prüflings-Systems dar. Wird nun die Federkennlinie von der jeweiligen gemessenen Kraft subtrahiert, erhält man die Reibung im System wie in **Abbildung 9.42** und **Abbildung 9.43** dargestellt.

Die in **Abbildung 9.42** und **Abbildung 9.43** gezeigten Kurven sind für beide geprüften Federungszyylinder nahezu identisch, daher wird hier nur ein Satz Werte dargestellt.

Randbedingung der Simulation

Simulationszeit 400 sek.

100 sek. Einschwingen

200 sek. Signal

100 sek. Ausschwingen

Sinus-Sweep 0 bis 2,5 Hz

Amplitude $\pm 25\text{ mm}$

Ergebnisse der Simulation für Reibmodelle:

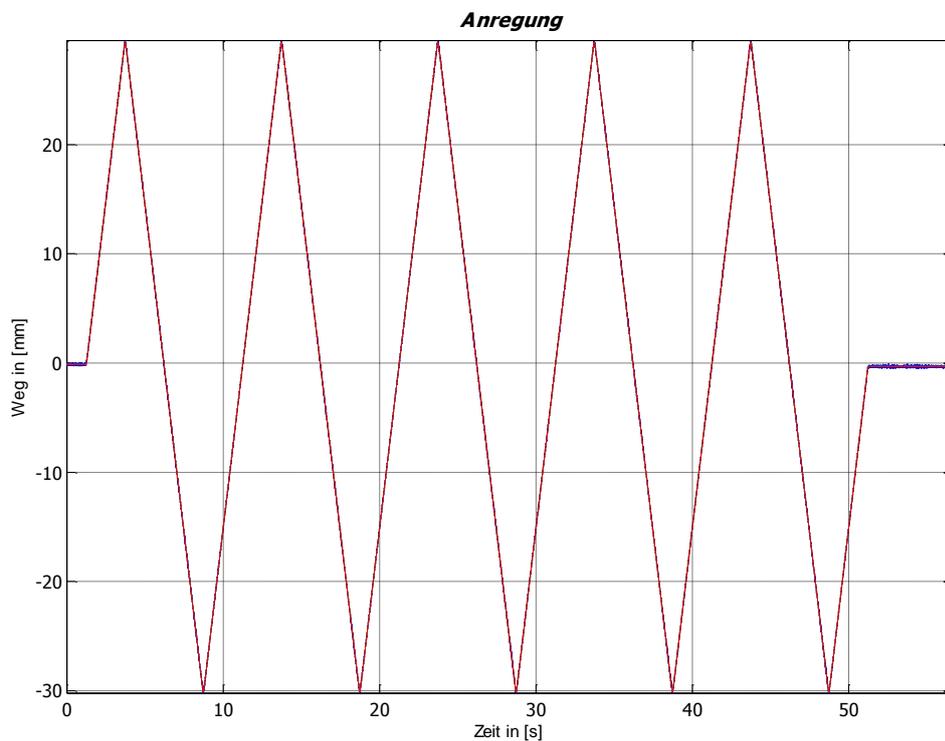


Abbildung 9.40 - Zylinderanregung

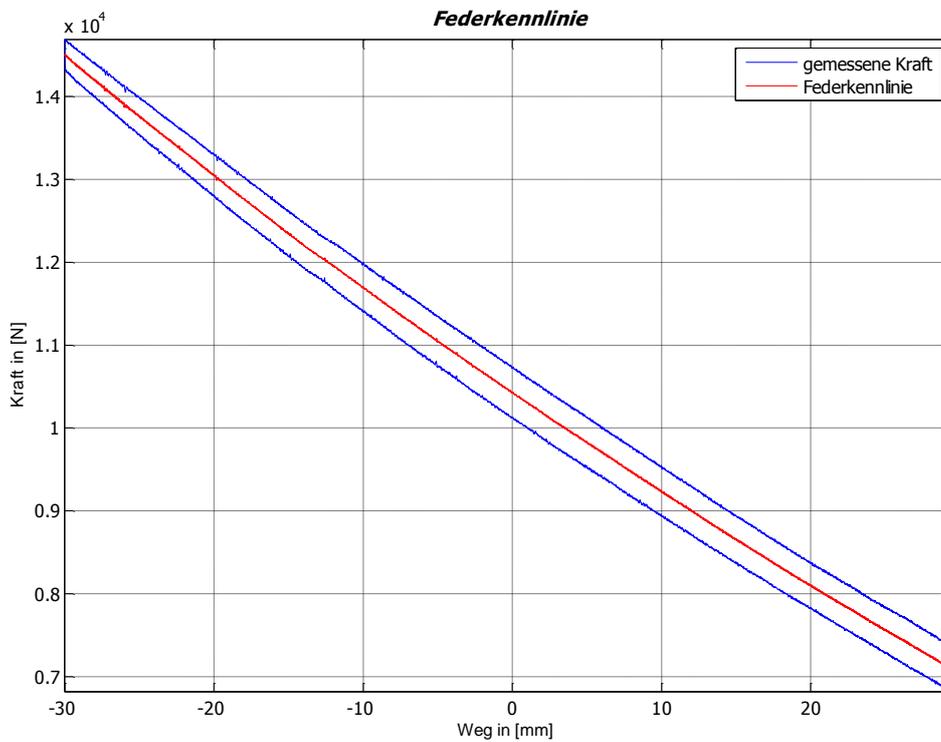


Abbildung 9.41 - Beispielhafter Kräfteverlauf

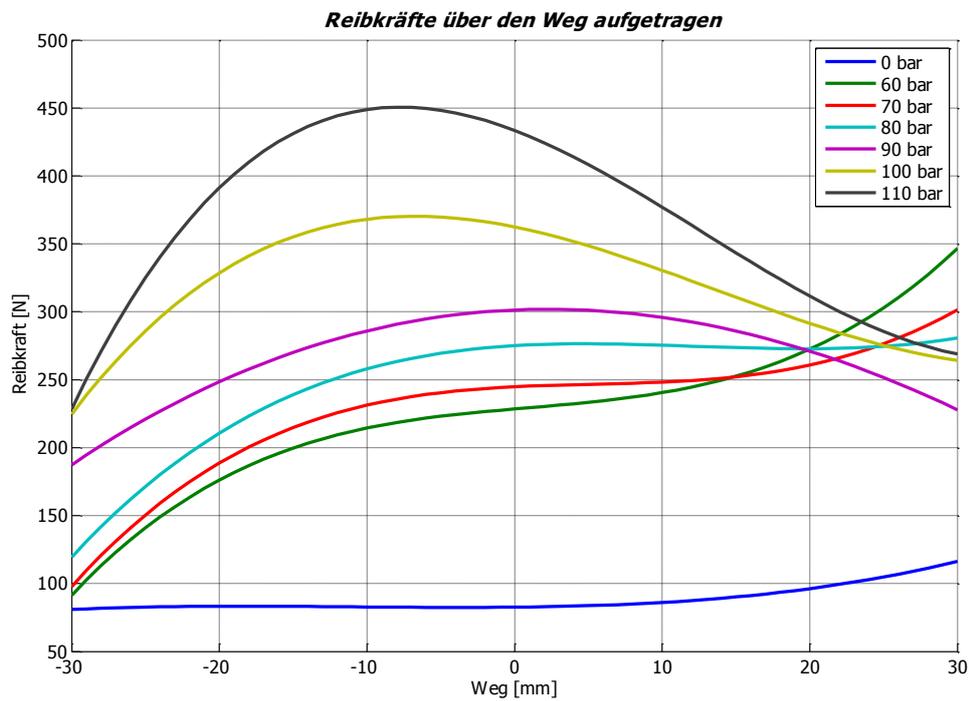


Abbildung 9.42 - Wegbezogene Reibkräfte

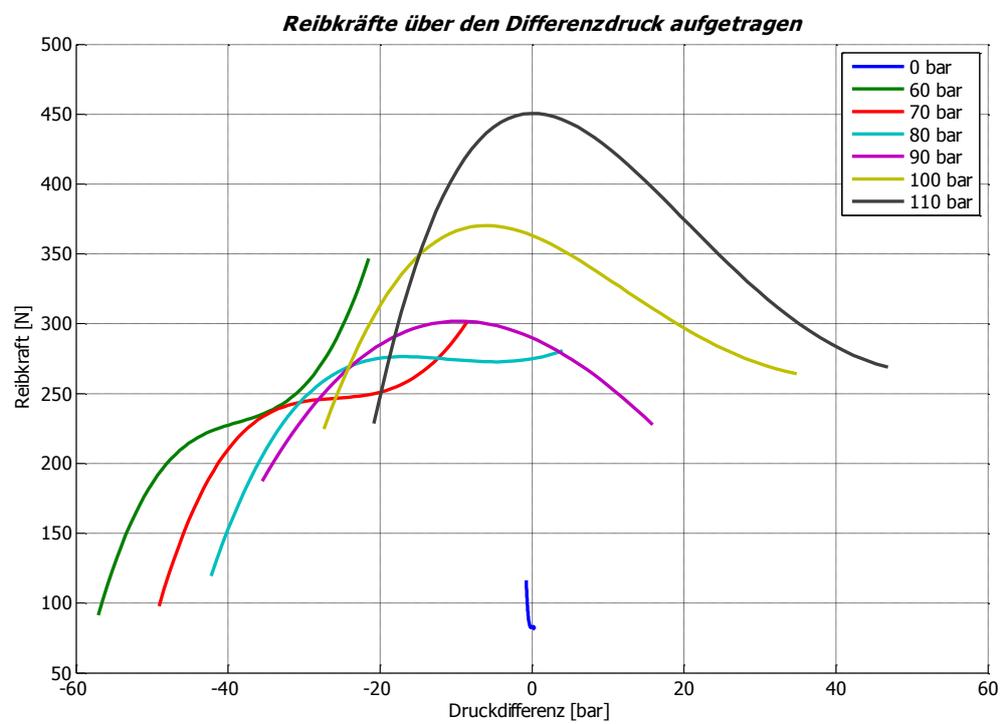


Abbildung 9.43 - Druckdifferenzbezogene Reibkräfte

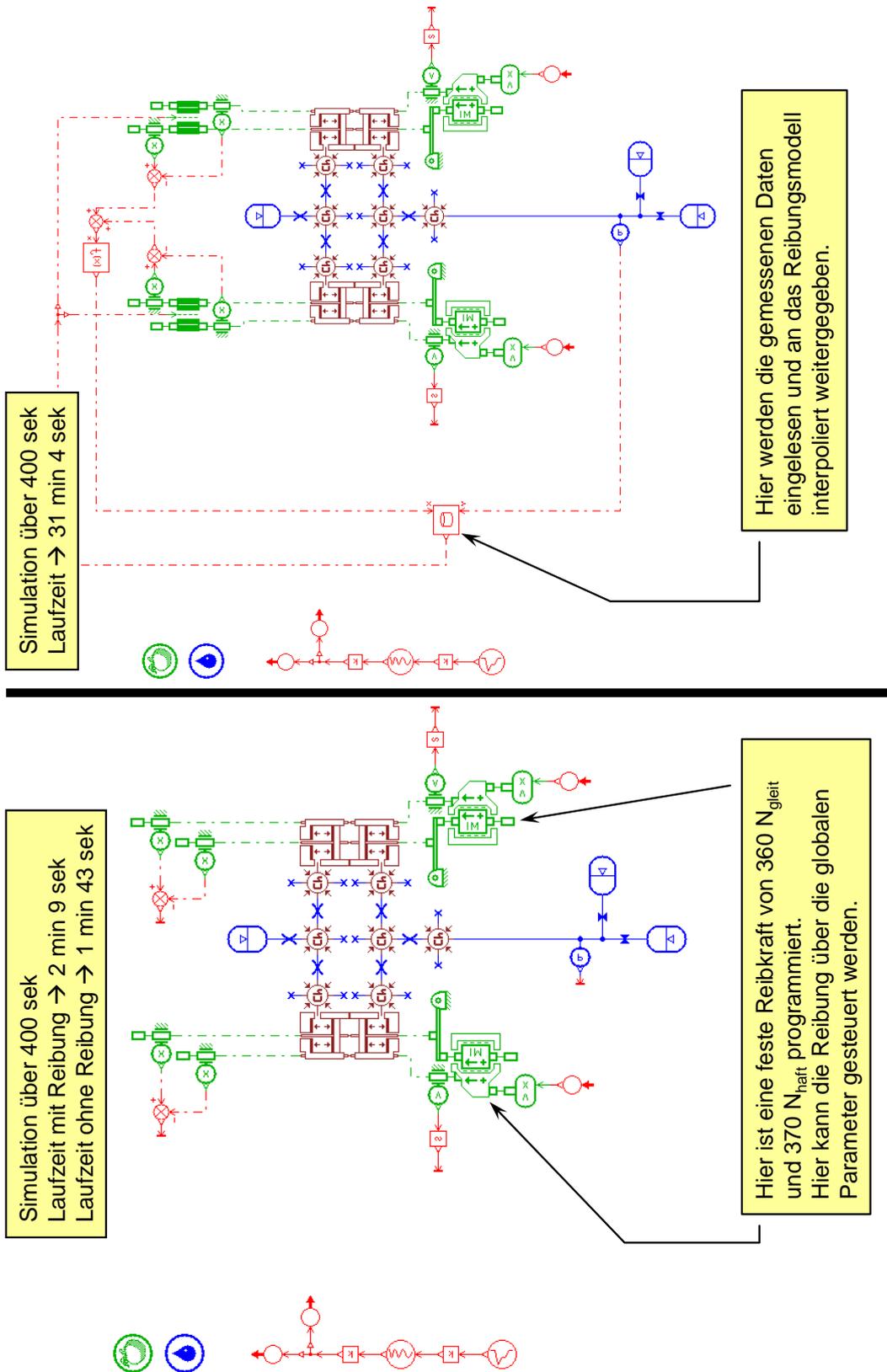


Abbildung 9.44 - Auszug aus den Simulationsergebnissen der Reibmodelle

9.4.2 Untersuchung des Speicher- und Leitungsmodelle

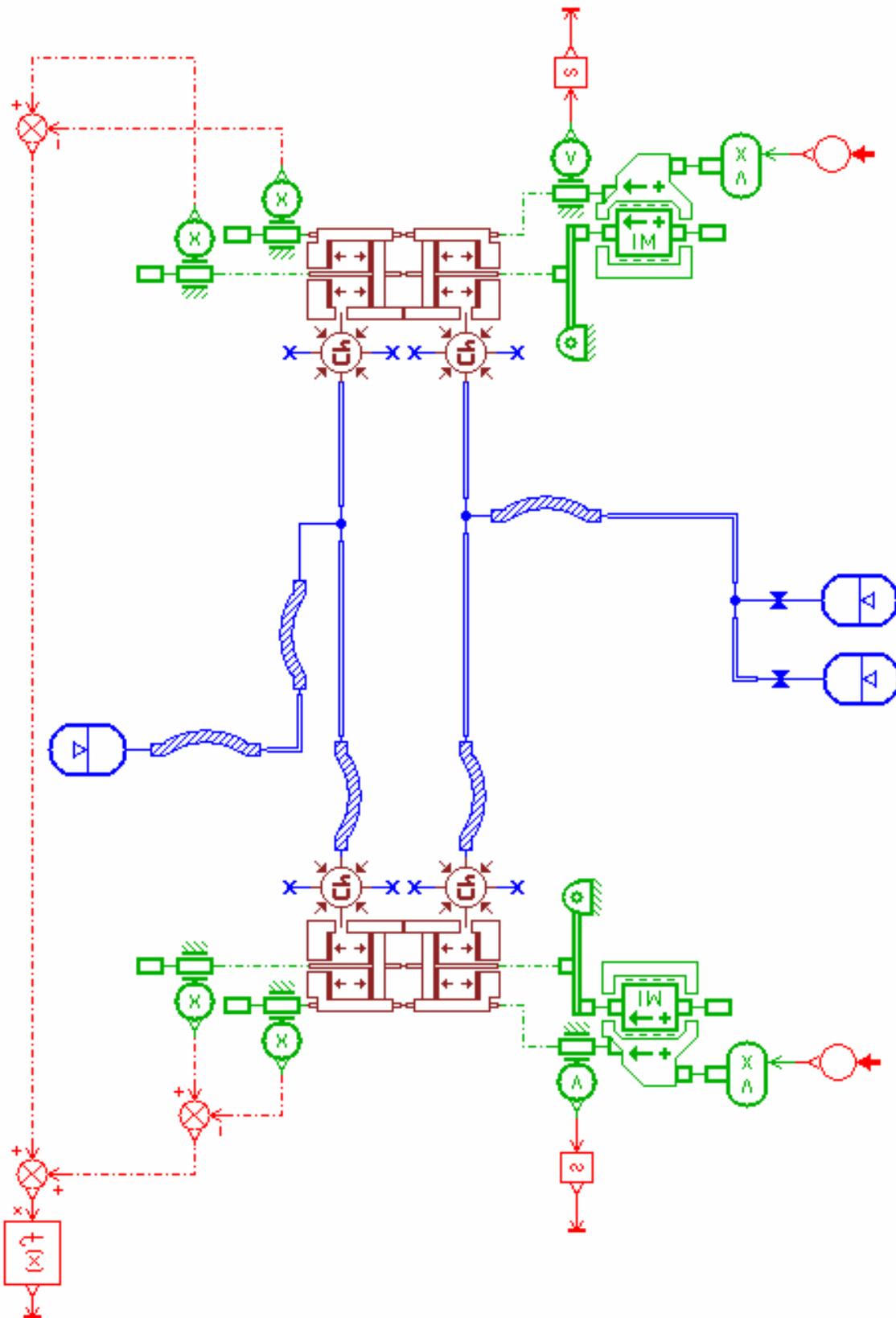


Abbildung 9.45 - Vollständiges Leitungsmodell

9.4.3 Abstimmung des Gesamtsimulationsmodells

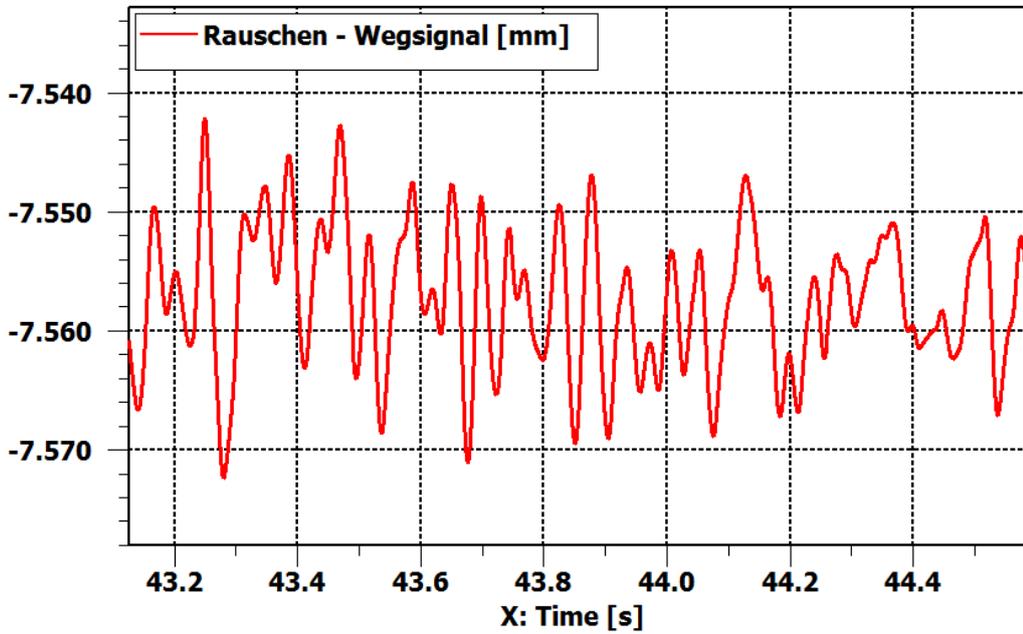


Abbildung 9.46 - Rauschen auf der Weganregung

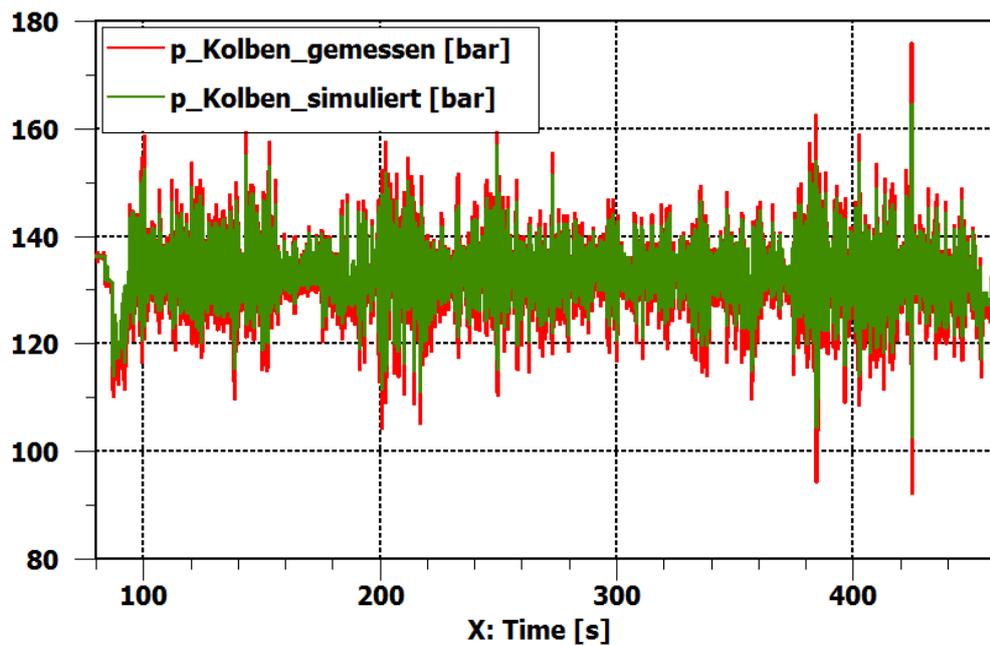


Abbildung 9.47 - Drücke an der Bodenseite der Federungszyylinder

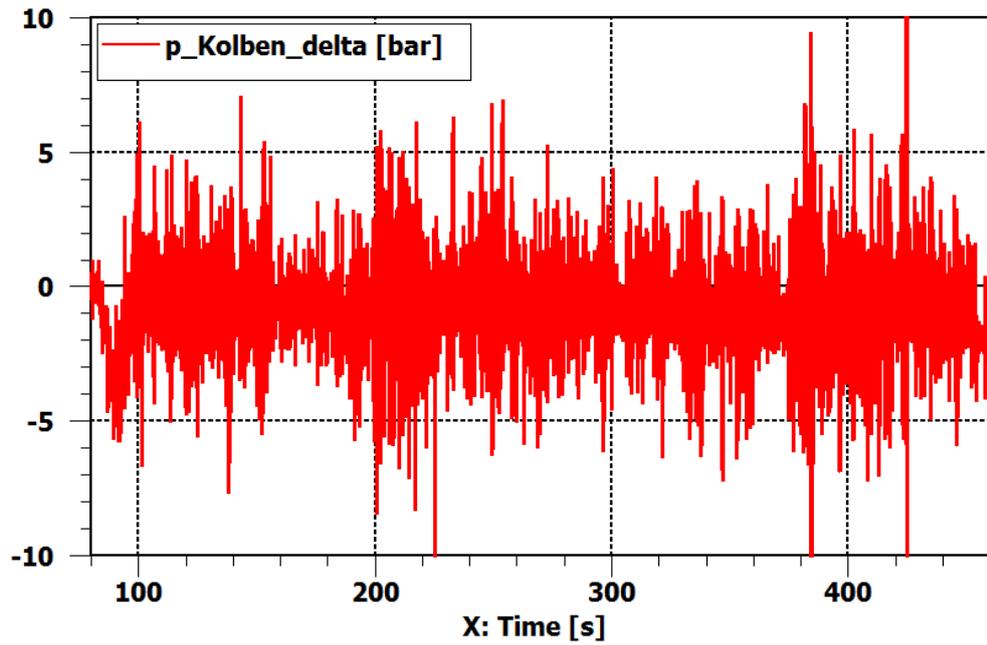


Abbildung 9.48 - Differenz der Drücke aus Abbildung 9.47

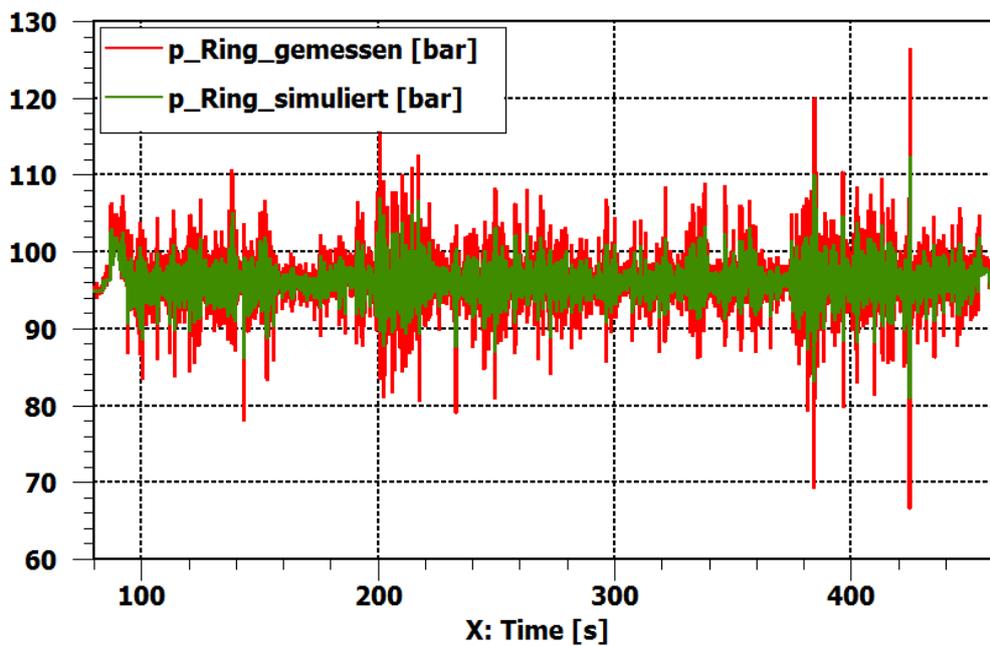


Abbildung 9.49 - Drücke an der Stangenseite der Federungszyylinder

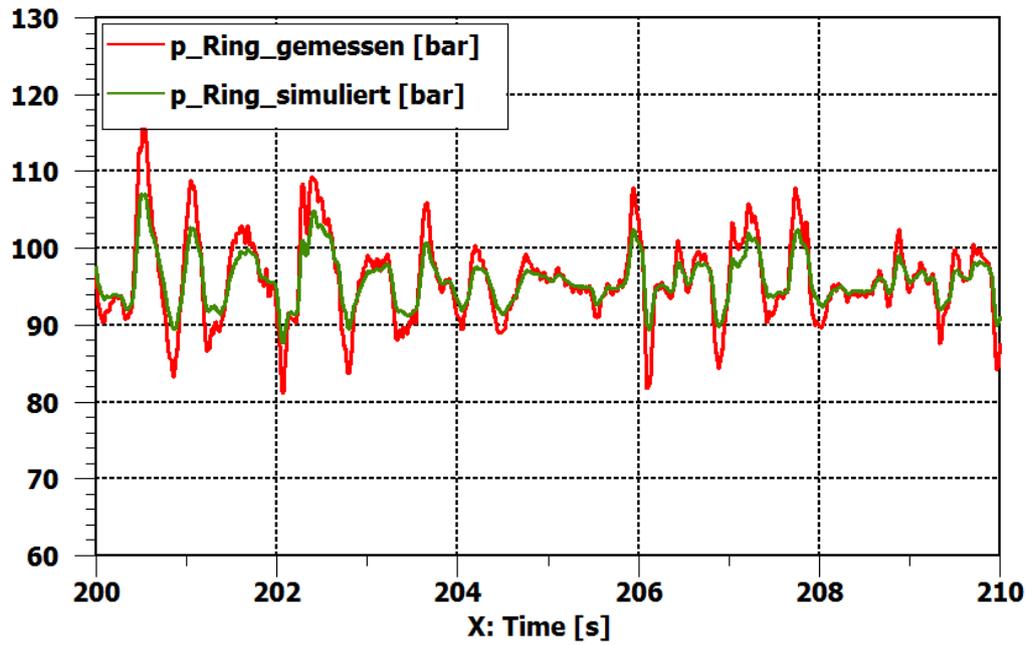


Abbildung 9.50 - Detail aus Abbildung 9.49

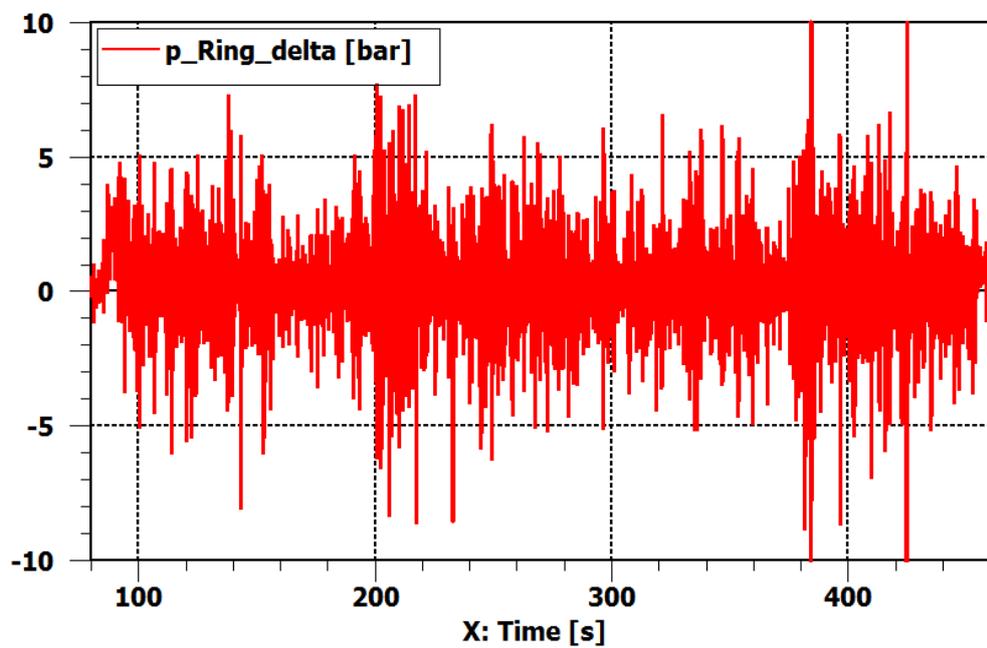


Abbildung 9.51 - Differenz der Drücke aus Abbildung 9.49

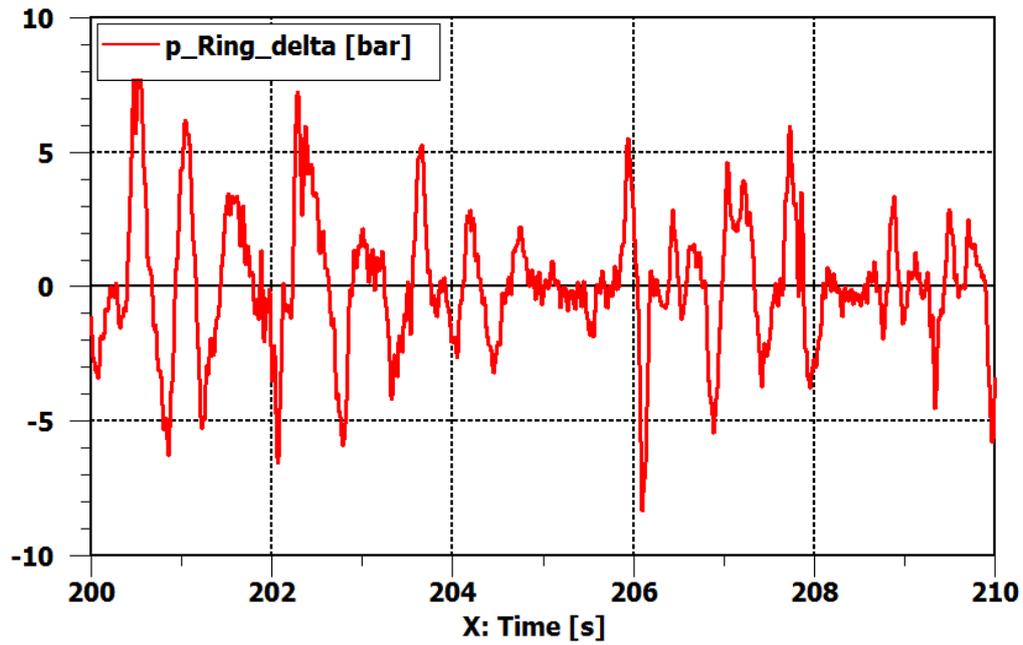


Abbildung 9.52 - Detail aus Abbildung 9.51

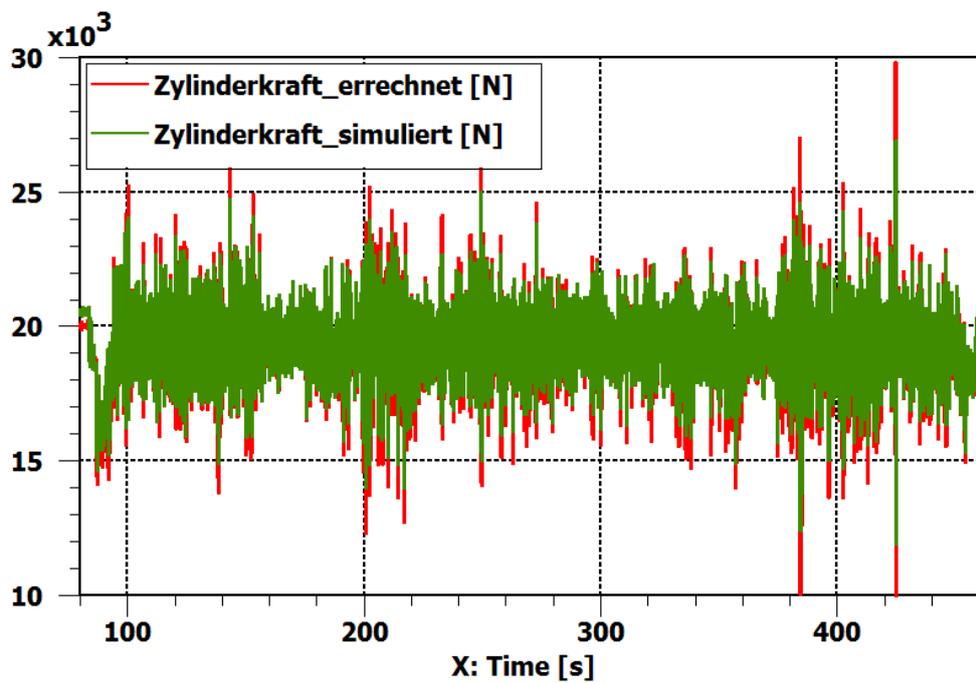


Abbildung 9.53 - Kräfte an einem einzigen Federungszyylinder

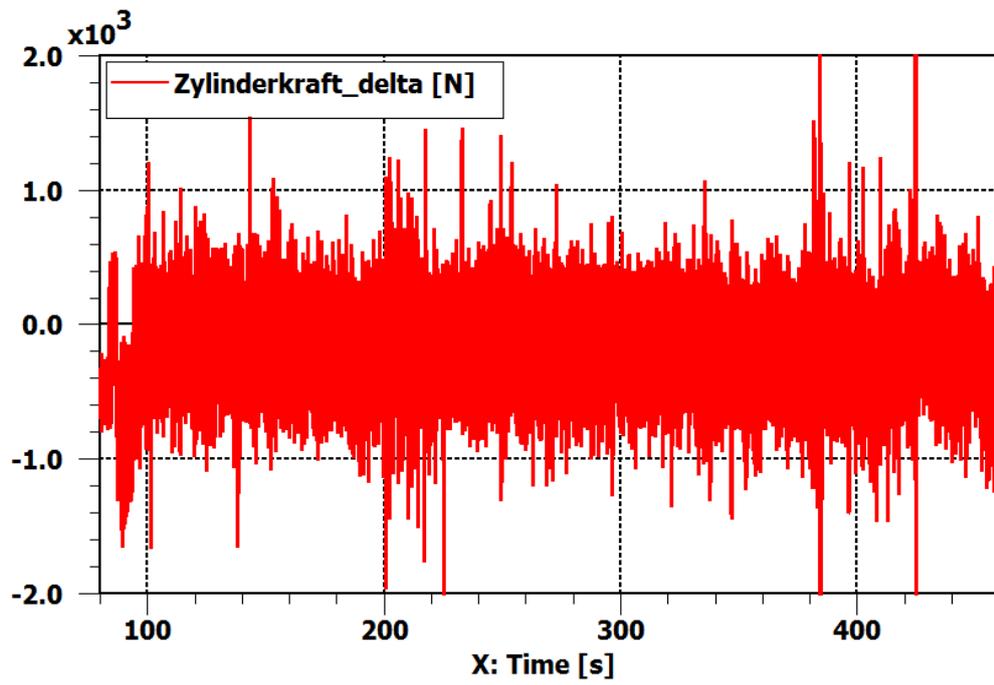


Abbildung 9.54 - Differenz aus Abbildung 9.53

9.5 Erfahrungen aus der Simulation

Abhängig von der Simulationsaufgabe können bereits Modelle an sich überschaubarer Systeme, wie auch das GUSMA-Beispiel der Vorderachsfederung, eine hohe Komplexität aufweisen, sowohl bei ihrer Erstellung selbst als auch bei der Anwendung und Aufbereitung der Resultate.

Aus Sicht eines OEM aus dem Bereich der mobilen Arbeitsmaschinen ist die Zusammenarbeit mit dem Zulieferer eines Teilsystems in der Phase des Modellaufbaus sehr wertvoll. Selbst wenn sich geeignete Modellierungswerkzeuge im Haus mit entsprechendem Bedienungs-Know-how in der Anwendung befinden, wie etwa die Simulationsumgebung LMS.Imagine.Lab bei der AGCO GmbH u. A. für Hydrauliksysteme, ergeben sich zwei wesentliche Vorteile. Zum einen lässt sich die Tiefe der Modellierung deutlich steigern. Ein Modell des Hydrauliksystems einer Vorderachsfederung ließe sich von AGCO zwar mit der Kenntnis von Funktion, Schaltplänen oder geometrischen Verhältnissen selbst erstellen und für prinzipielle Zusammenhänge auch nutzen, qualitativ gute Ergebnisse im Detail/Dynamik lassen sich aber erst mit dem Expertenwissen und der Erfahrung des Zulieferers Hydac zu Ventilblock und Speicher erzielen. Konkret am GUSMA - System hat die Kombination aus Modellierung der Speicher und Parametrierung der Blenden große Auswirkungen auf die Güte der Ergebnisse (siehe Exkurs „Einfluss Blenden- und Speichermodellierung“ im Anschluss). Eine ähnliche Modelltiefe, wie sie das Hydraulikmodell von Hydac besitzt, wäre für AGCO mit dem Einsatz hoher Kapazitäten für Versuch und Modellerstellung verbunden. Dieser reduzierte Aufwand ist bei begrenzten Simulationsressourcen der zweite große Vorteil. Aus der Erfahrung des Projekts GUSMA heraus, ist die vermeintliche Idealvorstellung, der OEM erhält das Modell eines Teilsystems vom Zulieferer, ohne sich mit dessen Verhalten auseinandersetzen zu müssen, allerdings wohl eher nicht der Regelfall. In der Praxis wird die Kommunikation zwischen Ersteller und Nutzer von Modellen, selbst bei guter Definition des Simulationsziels und Dokumentation des Modells, nicht ersetzbar sein. Ansätze zur Diskussion ergeben sich rasch etwa zu Simulationsergebnissen, Einflüssen von Parametervariationen oder Erweiterungen bzw. auch Eingrenzung der Nutzbarkeit, die vom reinen Anwender meist nicht zu beantworten sind. Letzterer trägt allerdings nach wie vor die Verantwortung der Interpretation der Ergebnisse.

Um die Simulationsaufgabe erfüllen zu können muss der Anwender die Teilmodelle verschiedener Disziplinen, Ersteller und Simulationsprogramme miteinander koppeln. Aufgrund der hohen Verbreitung und Verfügbarkeit von Schnittstellen vieler Simulationswerkzeuge bot sich auch bisher schon die Verwendung von MATLAB/Simulink an. Die Überarbeitung der Simulink-Schnittstelle des Simulationsumgebung SIMPACK im Rahmen des GUSMA-Projekts bringt durch die bessere Übersichtlichkeit deutliche Vorteile beim Aufbau der Kommunikation der Übergabewerte. Aus der Erfahrung heraus ist eine sinnvolle Handhabung von (gemeinsamen) Parametern und Initialisierungsvariablen ein besonders wichtiger Aspekt für eine fehlerresistente und komfortable Anwendung von Simulationsmodellen. Hier leistet die GUSMA-Plattform wertvolle Dienste, indem sie den Bediener in strukturierter Abfolge durch die Identifikation mehreren Modellen gemeinsamer Parameter und Initialisierungsvariablen, Variation und Sicherung aller änderbaren Größen leitet. Voraussetzung ist allerdings, dass der Ersteller eines Teilmodells die änderbaren Größen sorgfältig ausgewählt und freigegeben hat. Gerade bei „einfacheren“ Co-Simulationen (z.B. zwei Teilmodelle, vom Anwender selbst erstellt, keine gemeinsamen Parameter) erfordert die Nutzung anfangs

etwas mehr zeitlichen Aufwand und die Bereitschaft, gewohnte Abläufe anzupassen. Mit zunehmender Komplexität des Gesamtmodells steigt dann allerdings der Nutzerkomfort deutlich an.