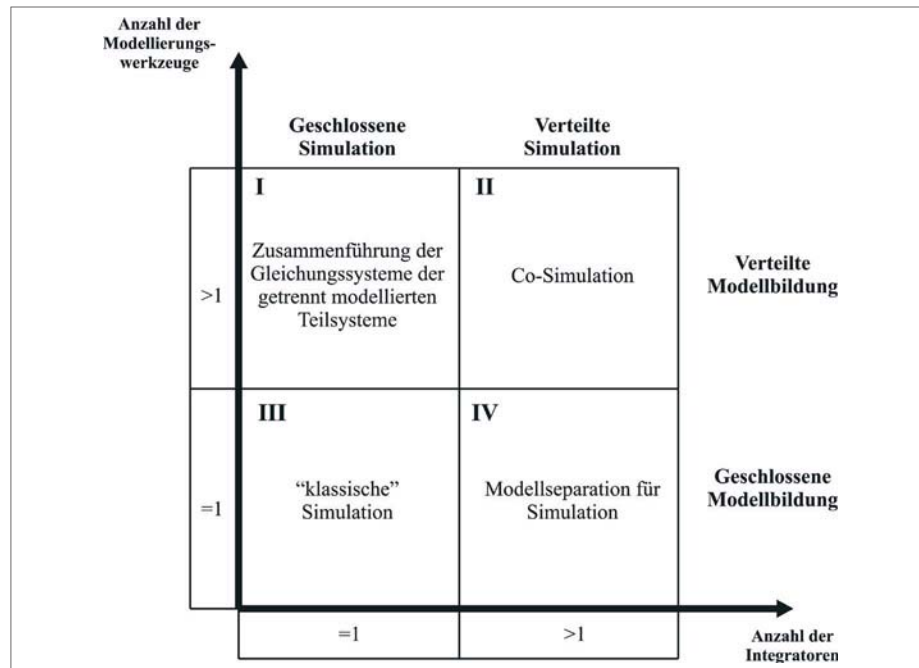


Co-Simulation, gekoppelte Simulation oder Simulatorkopplung?

Ein Versuch der Begriffsvereinheitlichung



1: Modellierungsvarianten (in Anlehnung an [1])

griffsbezeichnung zu finden und an ausgewählten Beispielen die Einordnung vorzunehmen.

Begriffsbestimmung

Wie in **Bild 1** dargestellt, kann die Modellierung multidisziplinärer Systeme grundsätzlich mit einem oder mehreren Simulationswerkzeugen durchgeführt werden. Bei der geschlossenen Modellbildung wird genau ein Modellierungswerkzeug verwendet, in welchem das Gesamtsystem modelliert wird. Dabei kann unterschieden werden, ob ein Integrator (geschlossene Simulation) oder mehrere Integratoren (verteilte Simulation) zum Einsatz kommen. Für die Berechnung mit einer verteilten Simulation muss das zuvor erstellte Modell in Teilsysteme zerlegt werden. Der Grund für dieses Vorgehen liegt in der Spezialisierung verschiedener Integratoren für spezifische Anwendungsfälle. So kann zum Beispiel ein mechanisches System im Vergleich zu einem hydraulischen System träge sein, was sich auf die mathematische Berechnung auswirkt.

Für die verteilte Modellbildung werden mindestens zwei unterschiedliche Modellierungswerkzeuge benötigt. Bei der geschlossenen Simulation werden die getrennt modellierten Teilsysteme oder deren Gleichungssysteme zusammengefasst und mittels eines Integrators berechnet; bei der verteilten Simulation werden die Teilsysteme mit verschiedenen Integratoren berechnet. Im letzteren Fall müssen die verschiedenen Teilsysteme gekoppelt werden, um eine Simulation des Gesamtsystems zu erreichen. Hieraus ergeben sich nach Dronka [1] zwei Ansätze zur Simulation multidisziplinärer Systeme, die sich als vielversprechend und gut durchführbar erwiesen haben. Dies sind zum einen die geschlossene Simulation mit verteilter Modellierung und zum anderen die verteilte Simulation mit verteilter Modellierung.

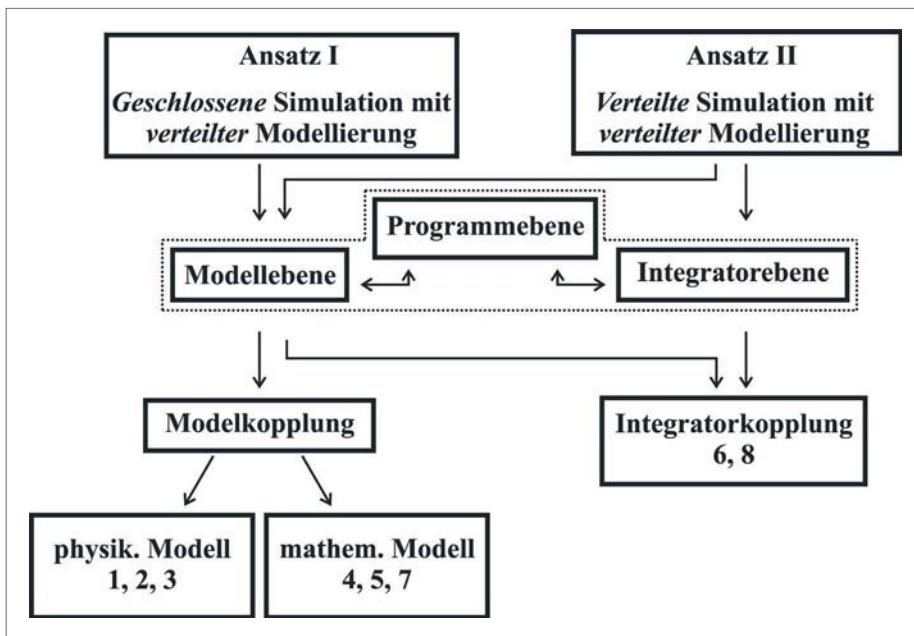
Ausgehend von den beiden genannten Kopplungsansätzen lassen sich zwei unterschiedliche Kopplungsvarianten, Ansatz I und II, ableiten (**Bild 2**). Generell lässt sich die Programmebene eines Simulationswerkzeuges in eine Modellebene und eine

Marcus Geimer, Thomas Krüger,
Peter Linsel

Die Kopplung unterschiedlicher Simulationsprogramme stellt einen Weg dar, ein Gesamtsystem, das aus Teilsystemen unterschiedlicher Disziplinen besteht, zu simulieren. In der Literatur werden hierfür unterschiedliche Wege vorgeschlagen und dabei unterschiedliche Begriffe verwendet. Dieser Beitrag fasst den aktuellen Stand der Technik zusammen und schlägt auf Basis der vorhandenen Literatur die Verwendung einheitlicher Begriffe vor.

Autoren: Prof. Dr.-Ing. Marcus Geimer ist Leiter des Lehrstuhls für Mobile Arbeitsmaschinen der Universität Karlsruhe (TH). Dipl.-Ing. Thomas Krüger und Dipl.-Ing. Peter Linsel waren Diplomanden am selben Lehrstuhl

Die Simulation ist ein etabliertes Werkzeug zur Unterstützung des Entwicklungsprozesses. Insbesondere bei der Entwicklung komplexer Systeme, wie beispielsweise einer ganzen Maschine, kann Zeit und Geld gespart werden, wenn das Verhalten der Maschine zuvor am Rechner simuliert wurde. Die Mechatronik ist ein anderes Beispiel, in dem mechanische, elektrische und informationstechnische Aufgaben zusammenhängend gelöst werden müssen. Der steigende Komplexitätsgrad technischer Systeme, bedingt durch die zunehmende Verflechtung unterschiedlicher Ingenieurdisziplinen, macht es erforderlich, die in domainspezifischen Simulationsprogrammen erstellten Teilmodelle einer gemeinsamen Simulation zugänglich zu machen, um so das Verhalten eines Gesamtsystems analysieren zu können. In der Literatur ist heute eine vielfältige Anzahl von Methoden zu finden, unterschiedliche Simulationsprogramme miteinander zu koppeln. Eine solche Kopplung ist notwendig, da mit heutigen Programmen eine Simulation über alle Fachgebiete nicht zufriedenstellend durchgeführt werden kann. In der Literatur werden dabei unterschiedliche Begriffe für das gleiche Vorgehen oder auch gleiche Begriffe für unterschiedliche Vorgehensweisen verwendet. Es soll daher der Versuch unternommen werden, eine einheitliche Be-



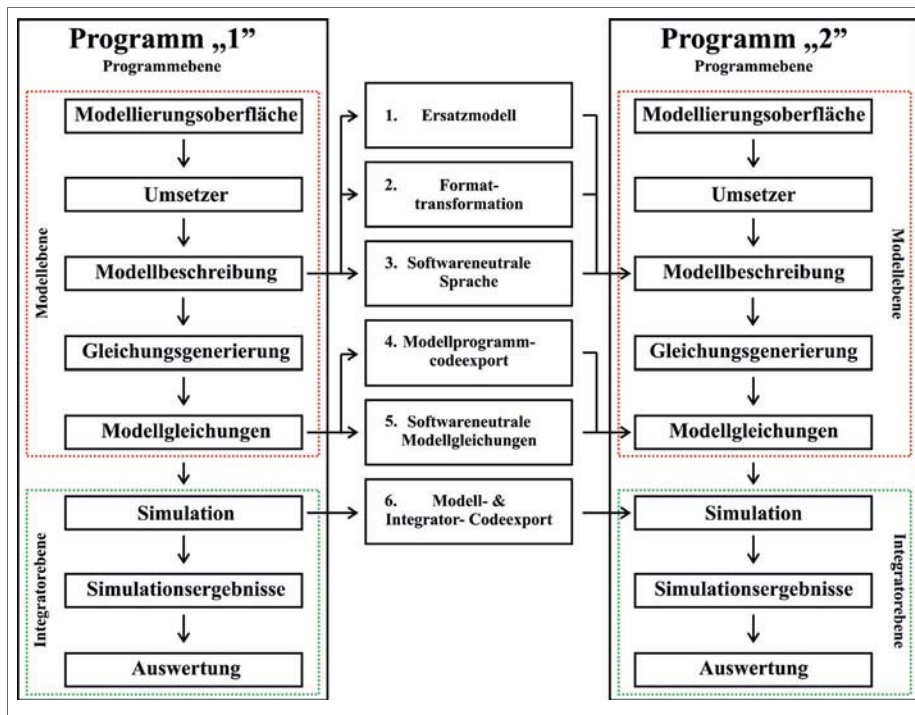
2: Kopplungsansätze und -varianten

Integratorebene unterteilen. Unterschieden wird weiterhin zwischen einer Modell- und einer Integratorkopplung, die entweder auf der Modell- oder Integratorebene des Simulationsprogramms realisiert werden kann. Die Modelkopplung lässt sich in eine physikalische und in eine mathematische Modellebene weiter unterteilen. Modelle, welche eine Ableitung eines realen Systems darstellen, lassen sich in physikalische und in mathematische Modelle unterteilen. Während bei einem physikalischen Modell die „körperliche“ Nachbildung eines Systems von zentraler Bedeutung ist, liegt der Fokus bei einem mathematischen Modell auf einer analytischen Beschreibung durch Gleichungen.

Ansatz I verfolgt ein Konzept, welches davon ausgeht, dass nur ein Integrator (geschlossen) für die Lösung des Gesamtmodells verantwortlich ist; die Modellierung findet jedoch in unterschiedlichen (verteilten) Simulationsprogrammen statt. Für diesen Ansatz eignen sich zwei, im Kern unterschiedliche Kopplungsvarianten. Zum einen ist eine Kopplung durch Modellzusammenschluss möglich, welche auf der Modellebene durchzuführen ist. Die Teilmodelle werden hierfür in den entsprechenden domainspezifischen Simulationswerkzeugen modelliert und im Anschluss daran mittels eines Modellimportes bzw. -exportes in einem Gesamtmodell zusammengeführt. Der Zusammenschluss findet dann innerhalb eines der schon zur Modellierung verwendeten Simulationsprogramme statt. Im Anschluss daran wird das Gesamtmodell vom Integrator des importierenden Simulationsprogramms gelöst. Zum anderen ist eine Kopplung durch Programmzusammenschluss möglich, welche dann auf der Programmebene durchgeführt wird. Hierbei werden die Teilmodelle, welche sich in den entsprechenden Modellebe-

nen befinden, über die jeweiligen Programmebenen zu einem Gesamtmodell miteinander verknüpft. Ein Modellimport bzw. -export ist hierzu nicht erforderlich. Vielmehr übernimmt eines der beiden Simulationsprogramme eine Koordinierungs- bzw. Steuerungsfunktion. Diese besteht in der Organisation des Funktionsaufrufes zur Gleichungsauswertung. In der Regel ist dieses jenes Programm, das auch den gemeinsamen Integrator zur Verfügung stellt. Die beiden beschriebenen Kopplungsvarianten erfüllen trotz ihrer unterschiedlichen Lösungsstrukturen die Forderungen des ersten Ansatzes. Eine geschlossene Simulation, also eine zentrale Berechnung des Gesamtmodells bei verteilter Modellierung, ist mit beiden Varianten durchführbar. Es handelt sich bei diesem Ansatz und den zugehörigen Varianten nicht um eine Co-Simulation, da dieser Ansatz generell von einer geschlossenen Integration ausgeht.

Ansatz II verfolgt dagegen ein Konzept, welches davon ausgeht, dass mehrere Integratoren (verteilt) für die Lösung des Gesamtmodells verantwortlich sind, das Gesamtmodell also durch „Co-Integration“ berechnet wird. Die vorangegangene Modellierung findet dabei ebenfalls in unterschiedlichen (verteilten) Simulationsprogrammen statt. Aus diesem Ansatz gehen wiederum zwei unterschiedliche Kopplungsvarianten hervor, die sich dahingehend voneinander unterscheiden, dass es möglich ist, einen Integratorzusammenschluss sowohl auf der Programm- als auch auf der Modellebene durchzuführen. Soll der Integratorzusammenschluss auf der Modellebene erfolgen, müssen beide Programme zu einem Modellimport bzw. -export inklusive des Solvers in der Lage sein. Die parallele Berechnung in mehreren Programmen ist durch den Austausch von Zustandsgrößen zwischen den Simulatoren



3: Kopplungsmöglichkeiten auf Modellebene (in Anlehnung an [1])

über geeignete Schnittstellen sowie einer Synchronisation der Simulationszyklen gekennzeichnet [2].

Kopplungsmöglichkeiten auf Modellebene

Zur Realisierung einer Kopplung auf Modellebene müssen die in den einzelnen Simulationsprogrammen erstellten Modelle zu einem gemeinsamen Gesamtmodell vereint werden. Geschieht die Modellkopplung innerhalb eines der beteiligten Programme, so ist auch der Integrator des importierenden Simulationsprogramms für die Lösung des Gesamtmodells verantwortlich. Denkbar ist auch ein Modellexport in eine Simulationsplattform mit eigenem Integrator, wobei dann jedoch alle an der Modellierung der Teilmodelle beteiligten Programme zu einem Modellexport in der Lage sein müssen. Der zentrale Integrator müsste dann in der Lage sein, das Gesamtmodell mit der nötigen Leistungsfähigkeit behandeln zu können und die meist sehr unterschiedliche Dynamik der miteinander gekoppelten Modelle in akzeptabler Rechengeschwindigkeit sowie Effektivität durchführen zu können. In der Literatur [1] sind insgesamt fünf Möglichkeiten zu finden, die aufzeigen, auf welche Art und Weise die Kopplung auf Modellebene, mit dem Ziel des Aufbaues eines Gesamtmodells, möglich ist. An erster Stelle werden die Möglichkeiten zur Modellkopplung auf der physikalischen Modellebene vorgestellt, welche im Anschluss daran durch die auf der mathematischen Ebene basierenden Kopplungsmöglichkeiten ergänzt werden. Eine Kopplung auf der physikalischen Modellebene ist nur dann durchführbar,

wenn beide Simulationsprogramme über die gleiche Art der Beschreibung verfügen. Eine sechste Möglichkeit der Kopplung findet im eigentlichen Sinne auf der Integratorebene statt (Bild 3).

1. Ersatzmodell

Mechanische Systeme können durch ihre Affinität zu elektrischen Systemen bezüglich ihrer mathematischen Beschreibung in ihrem Verhalten ersatzweise durch diese beschrieben werden. Diese Möglichkeit wird allerdings nur bei sehr einfachen Modellen angewendet und ist aufgrund dessen nahezu unbedeutend.

2. Formatttransformation

Diese Möglichkeit der Kopplung bezieht sich im Wesentlichen auf einen Modellexport zwischen zwei (identischen) Simulationsprogrammen der gleichen Ingenieurdisziplin. Grundlage der Verknüpfung ist, dass die Art der Beschreibungen gleich und es so möglich ist, die physikalischen Modelle miteinander zu verknüpfen. Entstammen die Modelle jedoch nicht der gleichen Disziplin, erweist sich eine gemeinsame Beschreibung als schwierig. Realisiert wird dies meist durch einfache Import-Export-Filter.

3. Softwareneutrale Sprache

Eine von den Ingenieurdisziplinen losgelöste Beschreibungssprache, wie zum Beispiel Modelica [3] oder VHDL-AMS [4], ermöglicht einen direkten Austausch der Modelle zwischen den Simulationsprogrammen der unterschiedlichen Ingenieurdisziplinen. Es handelt sich dann allerdings nicht mehr um ein rein physikalisches Modell, sondern um eine Art Mischform

zwischen physikalischem und mathematischem Modell.

Die beiden genannten objektorientierten Sprachen führen zwar eine mathematische Beschreibung durch, das System kann jedoch auch im Sinne eines physikalischen Modells abgebildet werden.

Weitere Möglichkeiten der Kopplung auf Modellebene bestehen in der Kopplung der mathematischen Modelle. Dazu werden die Gleichungen der Teilmodelle zu einem Gesamtsystem verbunden und einem gemeinsamen Integrator zugeführt.

4. Modellprogramm-Codeexport

Eine Möglichkeit besteht darin, das mathematische Modell mittels einer Programmiersprache (C, C++ oder FORTRAN) in der Form eines Programmcodes zu exportieren. Das Simulationsprogramm wird für die folgende Simulation nicht benötigt. Voraussetzung für einen erfolgreichen Zusammenschluss ist, dass das importierte Modell im Zielsystem ausführbar (portierbar) ist.

5. Softwareneutrale Modellgleichungen

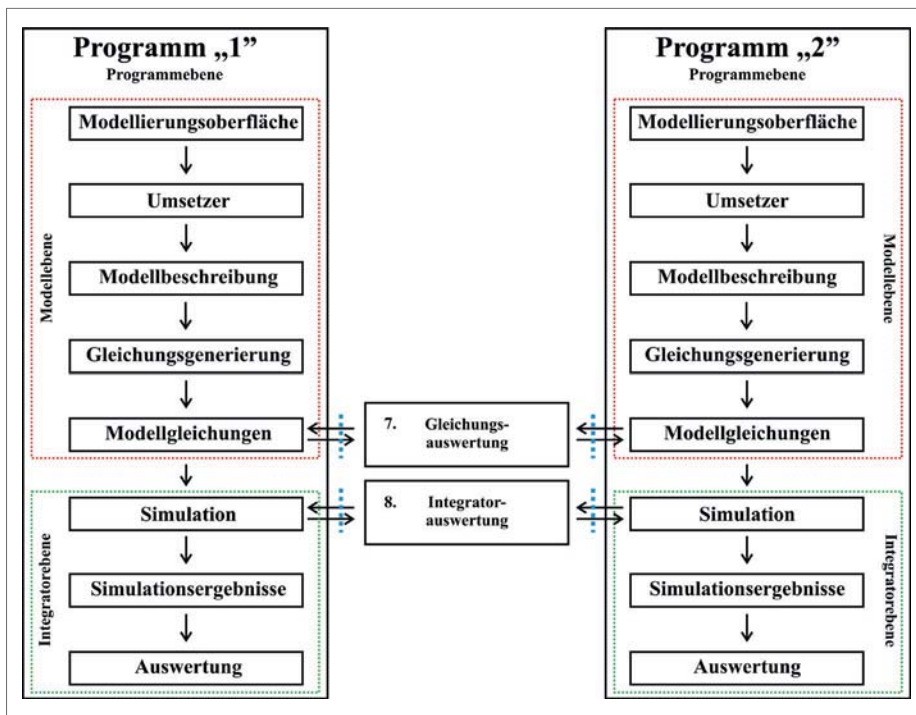
Ein Modellexport erfolgt durch den Austausch neutraler Beschreibungen der Modellgleichungen. Realisiert werden kann dies durch die Verwendung des CDIF- oder XMI-Metamodel-Formates. Diese Formate bilden die datenstrukturellen und algorithmischen Zusammenhänge von Software-Komponenten ab.

6. Modell- & Integrator-Codeexport

Realisiert werden kann ebenfalls ein Export des Modellcodes zusammen mit dem Integrationsalgorithmus. Diese Kopplungsmöglichkeit findet dann im eigentlichen Sinne auf der Integratorebene statt. Da diese Ebene nicht als eigenständig angesehen wird, zählt man diese Kopplungsmöglichkeit zur Modellebene hinzu. Wichtig ist, dass sowohl das Modell als auch der Integrator als Quellcode exportiert werden.

Kopplungsmöglichkeiten auf Programmebene

Eine Programmkopplung findet auf der Programmebene statt. Die Modellierungswerkzeuge werden für die Simulation benötigt. Hierfür greift jedes Simulationsprogramm auf das eigene mathematische Modell sowie den eigenen Integrator zurück. Über eine gemeinsame Schnittstelle kommunizieren die Simulationsprogramme während der Berechnung miteinander und die berechneten Größen werden ausgetauscht. Ein Modellexport muss daher zur Kopplung auf Programmebene nicht durchgeführt werden. Die Steuerungsfunktion übernimmt dabei das Simulationsprogramm mit der geringeren Schrittweite (Bild 4).



4: Kopplungsmöglichkeiten auf Programmebene (in Anlehnung an [1]; Bildnachweis: Verfasser)

7. Gleichungsauswertung

Diese Kopplungsmöglichkeit stellt einen Zusammenschluss der Modellgleichungen auf Programmebene durch Funktionsaufruf dar. Er fordert das untergeordnete Simulationsprogramm zu einer Auswertung seiner Modellgleichungen auf und zu einer anschließenden Übermittlung seiner Ergebnisse. Die anschließende Integration erfolgt dann in dem übergeordneten Programm. Die anfallende Kommunikation kann zum Beispiel mittels des Schnittstellenkonzepts „S-Function“ bei der Software Matlab/Simulink realisiert werden [5].

8. Integratorauswertung

Diese Möglichkeit entspricht einer verteilten Simulation. Das untergeordnete Simulationsprogramm fordert zu bestimmten Zeitpunkten das übergeordnete Simulationsprogramm auf, die Berechnung bis zu einem bestimmten Zeitpunkt durchzuführen. Zu bestimmten Zeitpunkten werden die aktuellen Werte zwischen beiden Programmen ausgetauscht, und die Integration wird fortgesetzt.

Beispiele

Zur Steigerung der Rechengeschwindigkeit wird in [6] eine Möglichkeit gezeigt, ein physikalisches Modell in ein mathematisches Modell umzuwandeln. Hierdurch ist es möglich, das betrachtete Modell von 44 auf 2 Zustandsgrößen zu reduzieren und dadurch die Rechengeschwindigkeit deutlich zu erhöhen. Dies zeigt, dass physikalische Modelle zwar einfach zu verstehen, mathematische Modelle aber üblicherweise Vorteile in Bezug auf die Rechengeschwindigkeit besitzen, ohne dass die Ge-

naugigkeit des Modells maßgeblich eingeschränkt wird. Im Folgenden werden beispielhaft für einige Kopplungsmöglichkeiten die zuvor beschriebenen Definitionen anhand von Beispielen kurz erläutert.

Schyr [7] stellt eine Methode vor, die es ermöglicht, einen gesamten Antriebsstrang mit seinem dynamischen Verhalten zu simulieren. Dafür wählt er basierend auf der Mehrpoltheorie dynamisch konjugierte Variablen, um den Leistungsfluss durch das Gesamtsystem Antriebsstrang zu beschreiben. Die physikalische Beschreibung des in Module (Teilmodelle) zerlegten Antriebsstranges wurde hierfür selbst programmiert. Das Zusammenführen der Teilmodelle wie auch die Simulation des Gesamtmodells erfolgt in Modelica. Dies ermöglicht es, ein softwareneutrales Gesamtmodell des Antriebsstranges auf der physikalischen Modellebene zu realisieren und im Anschluss daran auch zu simulieren. Dies entspricht der Kopplungsmöglichkeit „Softwareneutrale Sprache“ (Bild 3).

In [8] wird allgemein die Simulation in der Fluidtechnik betrachtet. Dabei werden vom Autor die Simulationsdisziplinen sowie die Kopplungsmöglichkeiten von Simulationssystemen im Maschinenbau beschrieben. Es wird eine „Kooperation der Simulationssysteme Adams und ITI-SIM“ ([8], Seite 31) analysiert. Hier erfolgt eine Integratorkopplung auf der Integratorebene. Dies entspricht der „Integratorauswertung“ (Bild 4). Weiterhin wird eine Kopplung zwischen SIMPACK bzw. DSHplus mit Matlab erläutert. Als technische Grundlage zur Simulation werden die elektrohydraulischen Stellemente von Neigetechnikzügen untersucht. Die Erstellung der Modelle für die mechanischen Eigenschaften

des Zuges sowie für die Fahrteinflüsse und Steuersignale der elektrohydraulischen Antriebe erfolgt in SIMPACK, für den elektrohydraulischen Antrieb in DSHplus. Im Anschluss werden die Modelle nach Matlab exportiert und dort berechnet. (Kopplungsmöglichkeit „Modell- & Integrator-Codeexport“, Bild 3).

Kliffken und Behm untersuchen eine Antischlupfregelung für hydrostatische Fahrtriebe [9]. Dabei greifen sie bei der Simulation auf eine Kopplung der Programme AMESim und Adams mit Matlab zurück. AMESim wird hierbei für die hydraulischen und mechatronischen Teilsysteme verwendet, während in Adams die mechanischen Teilsysteme abgebildet werden. In Matlab werden die Teilmodelle miteinander verknüpft. Dabei wird Matlab als sogenannter Master geführt, während AMESim und Adams als sogenannte Slaves mit eigenen Solvern ausgeführt werden. Matlab übernimmt, wie im vorangegangenen Fall, die Steuerungsfunktion. Dieses Beispiel kann der „Integratorauswertung“ (Bild 4) zugeordnet werden.

[10] beschreibt allgemein, unter Berücksichtigung der relevanten Teilsysteme, die Möglichkeiten einer umfassenden Systemsimulation. Dabei wird ein Code-Export bzw. -Import beschrieben. Die Teilmodelle eines Gesamtsystems werden in verschie-

denen Simulationswerkzeugen erstellt, um sie danach als Quellcode in ein Simulationswerkzeug zu transferieren. Dort wird das Gesamtsystem als ein System simuliert (Möglichkeit „Modellprogramm-Codeexport“, Bild 3).

Zusammenfassung

Für den vorliegenden Beitrag wurden die in der Literatur für die verteilte Modellbildung verwendeten Begriffe untersucht und eine einheitliche Verwendung der Begriffe vorgeschlagen. Grundsätzlich kann bei der verteilten Modellbildung zwischen der geschlossenen und der verteilten Simulation unterschieden werden. Insgesamt wurden acht Möglichkeiten aufgezeigt und definiert, je nachdem wie die Programme miteinander gekoppelt werden. An Hand von Beispielen aus der Literatur wurde die Handhabbarkeit der Begriffe verdeutlicht. Es muss sich nun in der Praxis zeigen, ob sich die vorgeschlagene Begriffsdefinition durchsetzen kann.

Literatur:

[1] Dronka, S.: *Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation*, Shaker Verlag, Aachen 2004, von der Fakultät Verkehrswissenschaften „Friedrich List“ genehmigte Dissertation der

Technischen Universität Dresden

[2] Findeisen, D.: *Ölhydraulik (5. Auflage)*; Springer Verlag Berlin, Heidelberg, New York, 2005

[3] Internetseite Modelica: <http://www.modelica.org>, Stand 11.10.2006

[4] Internetseite VHDL-AMS: <http://www.vhdl-ams.org>, Stand 18.10.2006

[5] Simulink® *Simulation and Model-Based Design: Writing S-Functions*, The Mathworks, Inc., Internet: http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/sfunctions.pdf#search=%22Writing%20S-Functions%22, Stand 30.08.2006

[6] Resch, R.: *Fortgeschrittene Modellierung und Simulation im Entwicklungsprozess für Baumaschinen*, Fachtagung Baumaschinentechnik, 5. bis 6. Oktober 2006, Schriftenreihe der Forschungsvereinigung Bau- und Baustoffmaschinen, Heft Nr. 34, S. 223-235

[7] Schyr, C.: *Modellbasierte Methoden für die Validierungsphase im Produktentwicklungsprozess mechatronischer Systeme am Beispiel der Antriebsstrangentwicklung*, von der Fakultät für Maschinenbau der Universität Karlsruhe (TH) genehmigte Dissertation, 2006

[8] Helduser, S.: *Simulation in der Fluidtechnik*, O+P Ölhydraulik und Pneumatik, 46 (2002), Nr. 1, S. 27-36

[9] Kliffken, M. G., Behm, M.: *Zur Schonung von Boden und Reifen – Antischlupfregelung für hydrostatische Fahrtriebe*, O+P Ölhydraulik und Pneumatik, 49 (2005), Nr. 2, S. 88-91

[10] Klein, A., Grätz, U., Schindler, J., Sanders, D.: *Mechatronik als neue Herausforderung für die Hydrauliksimulation*, O+P Ölhydraulik und Pneumatik, 49 (2005), Nr. 3, S. 160-165