

Kombinierte Simulation am Beispiel eines hydrostatischen Fahrtriebes

Die Simulation nimmt im Entwicklungsprozess eines Produktes einen immer höheren Stellenwert ein. Mit steigender Leistungsfähigkeit der Computerhardware bieten sich auch neue Möglichkeiten für die computergestützte Simulation.

Um den Funktionsumfang von Simulationsprogrammen zu erweitern, wurden hauptsächlich zwei Möglichkeiten realisiert. Nachfolgend werden deren Eigenschaften und Nutzen für den Anwender diskutiert. Besonderes Augenmerk liegt dabei auf der Kopplung von Simulationsprogrammen. Die Variante „Co-Simulation“ wird beispielhaft an der Kopplung von ADAMS und MatLab/Simulink erläutert. Als Funktionsbeispiel dient ein hydrostatischer Fahrtrieb einer selbstfahrenden Feldspritze.

Dipl.-Ing. Andreas Schumacher ist wissenschaftlicher Mitarbeiter am Institut für Landmaschinen und Fluidtechnik der TU Braunschweig (Leiter: Prof. Dr.-Ing. Dr. h.c. H.-H. Harms), Langer Kamp 19a, 38106 Braunschweig; e-mail: a.schumacher@tu-bs.de

Schlüsselwörter

Hydrostatischer Fahrtrieb, gekoppelte Simulation, Fahrdynamiksimulation

Keywords

Hydrostatic drive train, joint simulation, driving dynamics simulation

Literatur

Literaturhinweise sind unter LT 05203 über Internet <http://www.landwirtschaftsverlag.com/landtech/lo-cal/fliteratur.htm> abrufbar.

Die Simulation ist ein ideales Hilfsmittel zur Unterstützung der Entwicklung, da sie die realen Zusammenhänge im Modell realitätsnah abbilden kann. Somit können unter anderem die Entwicklungskosten gesenkt und Entwicklungszeiten verkürzt werden.

Im Zuge der fortschreitenden Entwicklung im Bereich der Computer-Hardware konnten immer leistungsfähigere Rechner mit ausführlichen Simulationsprogrammen zur Verfügung gestellt werden.

Im Gleichschritt mit diesen Entwicklungen haben sich auch die Forderungen des Marktes verändert: Erwünscht sind mittlerweile Simulationsprogramme, die nicht nur auf die Abbildung und Simulation einer Baugruppe - etwa die Hydraulikanlage - spezialisiert sind, sondern Programme, die auch die Interaktion dieser Baugruppe mit anderen oder mit der elektronischen Regelung einbeziehen und berechnen können.

Möglichkeiten der Simulationssoftware

Von den Softwareentwicklern sind vor allem zwei Möglichkeiten realisiert worden, diese Forderungen des Marktes zu bedienen: Der eine Weg besteht in der Erweiterung des Simulationsprogramms um weitere Werkzeuge, die einen erweiterten Funktionsumfang ermöglichen. Ein bedeutender Vorteil dieser Variante liegt darin, dass nur ein einziges Programm verwendet wird. Somit sinken die Anforderungen an die Computerhardware. Allerdings hat diese Variante den Nachteil, dass das Programmpaket in der Regel nur für das eigentliche Anwendungsgebiet spezialisiert sein kann. Dies bedeutet oft, dass die Fähigkeiten der zusätzlichen Werkzeuge - im Vergleich zu einem darauf spezialisierten Programm - begrenzt und eventuell ungenauer sind.

Die zweite Möglichkeit, einen erweiterten Funktionsumfang zu realisieren, besteht in der Kopplung von verschiedenen, spezialisierten Simulationsprogrammen. Die Kopplung der Simulationsprogramme miteinander kann unterschiedlich ausfallen, man unterscheidet hier drei verschiedene Kopplungsarten.

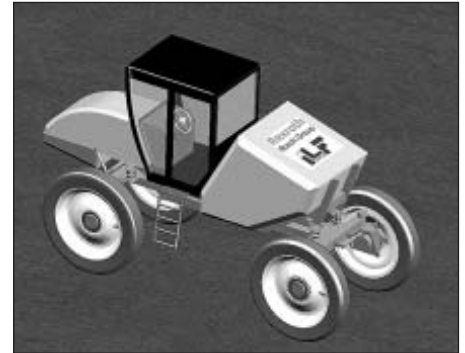


Bild 1: Selbstfahrende Feldspritze

Fig. 1: Self propelled sprayer

Co-Simulation

Die Modelle werden in ihrer jeweiligen Simulationsumgebung aufgebaut und berechnet. Zur Kopplung werden spezifizierte Ein- und Ausgabewerte zwischen den Programmen übertragen. Dabei steuert ein übergeordnetes Programm die anderen gekoppelten und nachgeordneten Algorithmen, die die Gleichungen lösen (Solver). Diese Methode ist rechen- und zeitaufwändig, aber genauer, da die Teilmodelle von ihrem jeweils eignen Solver gelöst werden.

Modellverbund

Hier wird ein Simulationsprogramm übergeordnet, in dem alle Modelle zusammengefügt werden. Dazu werden die zu koppelnden Modelle exportiert. Das Leitprogramm berechnet das Gesamtmodell, allerdings mit dem eigenen Programm, so dass hier eventuell aufgrund der spezialisierten Teilmodelle Probleme auftreten, da es hierfür nicht optimiert ist.

Solververbund

Diese Kopplung funktioniert ähnlich dem Modellverbund. In diesem Fall wird aber der Solver mit exportiert. Aufgrund der allgemeinen Lesbarkeit der Exportformate sind somit auch die Algorithmen der speziellen Programme und damit gleichzeitig auch Firmen-Know-how offengelegt. Daher ist verständlich, dass in der Regel die ersten beiden Kopplungsarten verwendet werden.

Viele der erhältlichen Simulationsprogramme offerieren beide der hier diskutier-

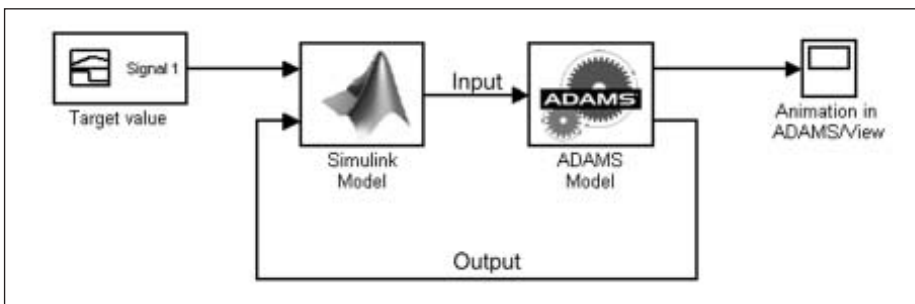


Bild 2: Signalfluss zwischen ADAMS und MatLab

Fig. 2: Information flow between ADAMS and MatLab

ten Möglichkeiten zur Funktionserweiterung. Es erscheint allerdings die Kopplung von Simulationspaketen für den Entwicklungsprozess eines Produktes interessanter, da wesentlich detailliertere Modelle aufgebaut werden können.

Hydrostatischer Fahrtrieb

Am Institut für Landmaschinen und Fluidtechnik (ILF) entstand in Zusammenarbeit mit der Fa. Bosch Rexroth eine Co-Simulation eines Fahrzeugs mit hydrostatischem Fahrtrieb (Bosch-Rexroth). Das Modell bildet eine selbstfahrende Feldspritze ab, die über Allradantrieb, realisiert durch hydrostatischen Einzelradantrieb, verfügt (Bild 1).

An die Abbildung der Fahrphysik wurden hohe Ansprüche gestellt. Zum einen sollte das Modell realitätsnah nur über Radmomente (Antriebs- sowie Bremsmomente) und Lenkwinkel bewegt werden. Zudem sollte eine freie Bewegung des Fahrzeugs in der Modellwelt umgesetzt werden. Die Antriebsmomente sollten über eine Simulation des Hydrostaten ermittelt werden.

Zum anderen war auch die realitätsnahe Wiedergabe der Fahrzeugreaktionen in allen drei translatorischen und rotatorischen Richtungen Bestandteil der Simulation.

Aufbau der Simulation

Aufgrund dieser Anforderungen wurde eine Co-Simulation gewählt: Das Fahrzeugmodell wurde in dem Mehrkörpersimulationsprogramm ADAMS aufgebaut, die Simulation der Hydrostaten erfolgte mit MatLab.

Der Aufbau des Modells in ADAMS hatte mehrere Vorteile: Da die Reaktionen des Fahrzeugs möglichst gut wiedergegeben werden sollten, musste das Modell entsprechend den physikalischen Eigenschaften wie zum Beispiel den Massenträgheitsmomenten realitätsnah aufgebaut werden. ADAMS bietet die Möglichkeit, die physikalischen Eigenschaften zu berechnen. Somit konnte davon ausgegangen werden, dass die für das Gesamtfahrzeug berechneten Trägheiten, Massen, etc. hinreichend genau sind.

Hinzu kommt, dass ADAMS eine gute und komfortable Visualisierung der Ergebnisse liefert, so dass das Modell auch zu Präsentationszwecken eingesetzt werden kann.

Die Verwendung von MatLab ist bei Co-Simulationen hilfreich, da MatLab - mit Simulink - nicht nur als Simulationssoftware für Regelalgorithmen, sondern auch als „Kommunikationsbasis“ dienen kann. Im Hinblick auf die spätere Simulation des hydrostatischen Fahrtriebs mit dem Simulationsprogramm AMESim zeigte sich, dass AMESim bisher eine Schnittstelle zu MatLab bietet, an einer direkten Schnittstelle zu ADAMS wird allerdings noch gearbeitet.

Somit hat MatLab den doppelten Zweck, einerseits regelungstechnische Elemente abzubilden und andererseits zwischen ADAMS und AMESim zu vermitteln [2].

Vor dieser geplanten Realisierung der Hydrauliksimulation mittels AMESim wird die Hydraulik mit MatLab abgebildet.

Kopplung von ADAMS und MatLab

Mit der aktuellen Version von ADAMS ist die Kopplung sehr komfortabel. Durch entsprechende ADAMS-Werkzeuge erhält man eine vorgefertigte Schnittstelle, mit deren Hilfe die notwendigen Modelldaten entsprechend formatiert exportiert werden.

In MatLab gibt es eine ebenso vorgefertigte Schnittstelle, die die exportierten Daten einliest und daraus ein Subsystem für Simulink erstellt (Bild 2).

Bei der Co-Simulation von ADAMS und MatLab ist MatLab ADAMS übergeordnet. Das bedeutet, dass zwar beide Solver verwendet werden, der ADAMS-Solver allerdings von dem MatLab-Solver gesteuert wird. Die Simulationsparameter wie zum Beispiel die Schrittweite werden somit in MatLab konfiguriert.

Davon unabhängig werden die Ein- und Ausgangs-Variablen in ADAMS vor dem Export des ADAMS-Modells bestimmt, denn in diese Variablen werden die Daten geschrieben, die zwischen ADAMS und MatLab ausgetauscht werden. Da jede ADAMS-Rechenvariable als Ein- oder Ausgangs-Variable definiert werden kann, ist der Freiheitsgrad sehr hoch. Es können besonders im regelungstechnischen Bereich mit Simulink sehr viele Untersuchungen und Testläufe gefahren werden. Bild 3 zeigt dazu als Beispiel eine einfache Tempomat-Regelung auf 10 m/s Fahrgeschwindigkeit. Dargestellt sind die Fahrgeschwindigkeit und

das Moment eines vorderen Hydromotors über der Zeit. Die Diagramme verdeutlichen die Interaktion zwischen ADAMS und MatLab: Das Motordrehmoment wird von MatLab als Eingangswert an ADAMS gesendet. Dort wird als Reaktion auf das Motormoment die Fahrgeschwindigkeit ermittelt und an MatLab gesendet. Diese Regelgröße wird dort zur Ermittlung des notwendigen Motormomentes verwendet.

Erfahrungen mit der gekoppelten Simulation

Da die Kopplungsform zwischen ADAMS und MatLab die Co-Simulation ist, bringt diese Kopplung natürlich die angesprochenen Eigenschaften einer Co-Simulation mit sich: Bei einem komplexen Modell wie dem hier verwendeten Fahrzeug mit hydrostatischem Antrieb sind die Anforderungen an die Rechnerleistung hoch, wenn zeitnahe Ergebnisse betrachtet werden sollen. Bei dieser Simulation wurde mit einem aktuellen Prozessor und hinreichend Arbeitsspeicher eine einigermaßen schnelle Rechnung erreicht.

Allerdings ist darauf hinzuweisen, dass die Verteilung der Rechenzeit auf die jeweiligen Programme von der Komplexität des Modells abhängt. Bei diesem Modell benötigte das ADAMS-Programm den Großteil der Rechenzeit, da die komplexen und zum Teil un stetigen fahrdynamischen Zusammenhänge hohe Berechnungsanforderung an das ADAMS-Programm stellten.

Von diesen modellabhängigen Anforderungen an die Rechnerleistung abgesehen funktioniert die Kopplung sehr gut. Die Simulationsgüte hingegen wird von der Schnittstelle nicht beeinflusst, sondern sie basiert auf den verwendeten Modellen.

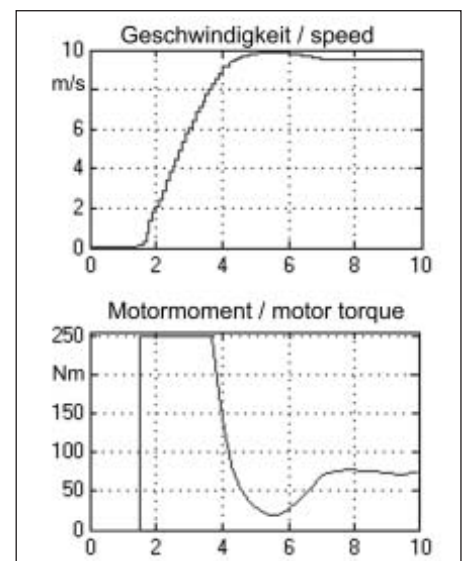


Bild 3: Ein- und Ausgangs-Variable

Fig. 3: Input and output-variables