

Modelica-Modellbibliothek zur Simulation der Dynamik von Schaltvorgängen bei Automatikgetrieben

Modelica model library for shift dynamics simulation of automatic gearboxes

C. Schweiger, Prof. Dr. M. Otter

DLR, Institut für Robotik und Mechatronik, Oberpfaffenhofen

Kurzfassung

In diesem Beitrag wird gezeigt, wie die Modelica-Komponentenbibliothek PowerTrain zur Modellierung von Automatikgetrieben verwendet werden kann. In der Modelica-Simulationsumgebung Dymola läßt sich aus den Modellen effizienter Simulationscode erzeugen, der selbst für Echtzeitanwendungen geeignet ist. Gekoppelte Reibelemente (Kupplungen, Bremsen, Verzahnungsreibung) werden in der Simulation zuverlässig behandelt.

Abstract

It is shown, how the Modelica PowerTrain component library is utilized to model automatic gearboxes. Such models can be processed in the Modelica simulation environment Dymola to generate efficient simulation code suitable even for realtime purposes. In simulations, coupled friction elements (clutches, brakes, mesh friction) are handled reliably.

1. Einleitung

Eine Schwierigkeit in der Simulation der Dynamik von Schaltvorgängen in Automatikgetrieben besteht im Auftreten von gekoppelten Reibelementen. Zu deren Modellierung und Simulation hat sich die von der objektorientierten Modellierungssprache Modelica unterstützte hybrid diskret-kontinuierliche Modellierungstechnik in Kombination mit der Modelica-Simulationsumgebung Dymola bewährt. Eine weitere Stärke von Modelica liegt in der integrierten Betrachtung verschiedenster Ingenieursdisziplinen, was beispielsweise bei mechatronischen Produkten unumgänglich ist. Die Modelica-Sprachspezifikation ist ebenso wie viele Modellbibliotheken frei verfügbar.

Speziell zu Modellierung von Antriebssträngen mit Automatikgetrieben etablierte sich die Modelica-PowerTrain-Bibliothek. Der Schwerpunkt liegt auf den in Getrieben vorkommenden Reibkomponenten wie Kupplungen, Bremsen, Lagerreibung und Verzahnungsreibung. Wei-

tere Komponenten stehen zu Verfügung, um den Aufbau von Gesamtfahrzeugmodellen zu ermöglichen, mit denen Untersuchungen der Dynamik der Schaltvorgänge oder zum Kraftstoffverbrauch vorgenommen werden können. Im weiteren werden exemplarisch Ergebnisse der Simulation eines Sechsgang-Automatikgetriebes gezeigt.

Nach einer Einführung in die objektorientierte Modellierung mit Modelica wird gezeigt, wie Automatikgetriebe unter Verwendung der PowerTrain-Bibliothek modelliert werden können. Auf die Modellierung von last- und geschwindigkeitsabhängiger Reibung und die Modellierung 3D-mechanischer Effekte von 1D-Antriebssträngen wird detaillierter eingegangen. Am Beispiel der automatischen Applikation von Steuergeräten wird ein mögliches Einsatzszenario der Bibliothek beschrieben.

2. Objektorientierte Modellierung und Simulation

Es wird eine Übersicht über die objektorientierte Modellierungssprache Modelica und die Modelica-Simulationsumgebung Dymola gegeben. Modelica eignet sich besonders zur Modellierung mechatronischer Systeme, die sich aus elektrischen, mechanischen, hydraulischen, pneumatischen, thermischen, fluidmechanischen und regelungstechnischen Teilkomponenten zusammensetzen. Durch die parametrische Darstellung komplexer Systeme sind Modelica-Modelle ein sehr guter Ausgangspunkt für die nichtlineare Parameteridentifikation und den optimierungsbasierten Systementwurf.

Zunächst wird, basierend auf [18], eine Übersicht über die Modellierungssprache Modelica gegeben, mit der multidisziplinäre Systeme komfortabel modelliert werden können. Mit Hilfe eines geeigneten Softwareprogramms (z. B. Dymola) können aus solchen Modellen effizienter Simulationscode erzeugt, Simulationen durchgeführt und Ergebnisse ausgewertet werden. Wahlweise wird der erzeugte Simulationscode in Simulink [11] eingebunden.

Modelica ist eine frei verfügbare Sprache die zusammen mit der ebenfalls frei verfügbaren Modelica-Standard-Bibliothek seit 1996 von der gemeinnützigen Modelica Association kontinuierlich entwickelt wird. In dieser internationalen Vereinigung engagieren sich Partner aus Hochschulen, Forschungseinrichtungen und Industrie. Modelica basiert wesentlich auf der von Hilding Elmqvist entwickelten objektorientierten Modellierungsmethodik [5]. Die ersten Anwendungen mit Modelica im Zusammenspiel mit der kommerziellen Modelica-Simulationsumgebung Dymola [4] gibt es seit Ende 1999. Seitdem steigt die Anzahl der industriellen Anwendungen stark an. Detaillierte Informationen zu Modelica sind im Internet erhältlich [14]. Zur Einführung in Modelica bieten sich insbesondere die Bücher von Tiller [23] und Fritzson [7] an.

Aus Benutzersicht werden in Modelica Modelle durch Objektdiagramme beschrieben. In Bild 1 ist eine Übersicht solcher Modelle aus unterschiedlichen Anwendungsgebieten beispielhaft zusammengestellt.

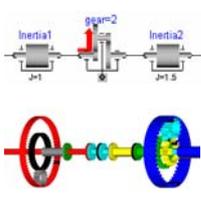
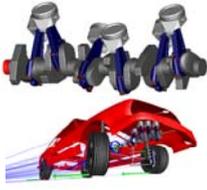
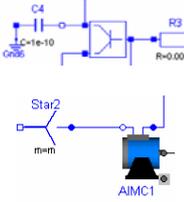
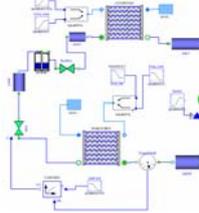
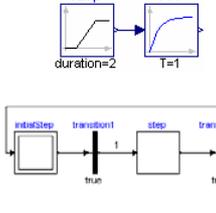
				
Antriebsstränge	3D-Mechanik	Elektr. Komp. und Maschinen	Thermofluid-Rohrströmungen	Signalblöcke und Zustandsmaschinen

Bild 1: Modelica-Objektdiagramme und -Animationsansichten aus unterschiedlichen Fachgebieten

Ein Objektdiagramm, siehe Bild 2, ist dabei eine Verallgemeinerung eines hierarchischen Blockschaltbildes, wobei die Schnittstellen von Komponenten nicht nur gerichtete Signale (typisch bei signalflussorientierter Darstellung), sondern auch physikalische Schnittstellen enthalten können (wie mechanischer Flansch, elektrische Klemme). Auf unterster Ebene werden Komponenten durch Differentialgleichungen, algebraische und diskrete Gleichungen beschrieben, ohne dass manuell nach einer bestimmten Variable aufgelöst werden muss. Einem Modelica-Werkzeug werden genügend Informationen bereitgestellt, um dies automatisch zu entscheiden. Unter Verwendung spezieller Algorithmen zur effizienten Behandlung großer Modelle können mehr als hunderttausend Gleichungen bewältigt werden. Hervorzuheben ist, dass die Verbindungen zwischen den Komponenten keine Signalflüsse, sondern physikalische Verbindungen von *Konnektoren* darstellen. Ein Konnektor enthält alle notwendigen physikalischen Größen, um die Interaktion mit anderen Komponenten zu beschreiben.

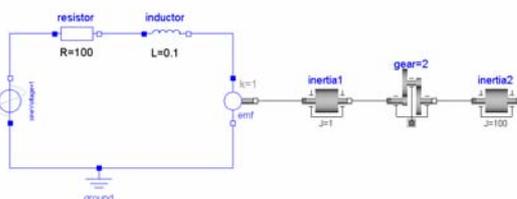


Bild 2: Typisches Modelica-Modell in Objektdiagramm-Darstellung (Elektrischer Antrieb)

Neben Basis-Sprachelementen unterstützt Modelica ein- und mehrdimensionale Felder mit einer Matlab-ähnlichen Syntax. Jede Instanz einer Klasse kann als Feldelement verwendet werden. Auf diese Weise können einfach diskretisierbare partielle Differentialgleichungen komfortabel definiert werden. Modelica hat eine ausgefeilte Unterstützung von un stetigen und strukturvariablen Komponenten, wie elektrische Schalter, Lagerreibung, Kupplungen, Abtastsysteme etc. Hierbei werden die notwendigen Zeit- und Zustandsereignisse automatisch erzeugt.

2.1. Modellierung mit Modelica-Komponentenbibliotheken

Ein bewährtes Vorgehen zur Modellierung komplexer Systeme besteht in der Verwendung von hierarchischen Komponentenbibliotheken. Komponenten auf unterster Hierarchieebene sind gleichungsorientiert modelliert. Die Bibliotheken entstehen durch Einstellen dieser Basis-komponenten. Für den Aufbau eines konkreten Modells werden in einem graphischen Modelleditor Komponenten aus Bibliotheken entnommen, miteinander verschaltet und parametrisiert. Teilmodelle, welche auf diese Weise generiert wurden (Beispiel siehe Bild 2), können ebenfalls in die Bibliotheken aufgenommen werden und stehen somit für eine Wiederverwendung in hierarchischen Modellen zur Verfügung.

Für die Austauschbarkeit und Weitergabe von Modellen ist es wichtig, Bibliotheken verfügbar zu haben, die die am häufigsten verwendeten Komponenten enthalten. Neben der Modelica-Sprache entwickelt die Modelica Association auch frei verfügbare Modelica-Bibliotheken. Alle ausgereiften Bibliotheken werden in der Modelica-Standardbibliothek *Modelica* zusammengefasst und gepflegt. Darüber hinaus gibt es auch aus anderen Quellen freie und kommerzielle Bibliotheken wie die unten vorgestellte PowerTrain-Bibliothek. Detailliertere Informationen zu allen Bibliotheken gibt es unter <http://www.modelica.org/library>. Weitere Bibliotheken sind zur Zeit in Entwicklung. Weiter hat Modelica ein leistungsfähiges Bibliothekskonzept mit Versionsverwaltung, um viele und große Komponentenbibliotheken verwalten zu können.

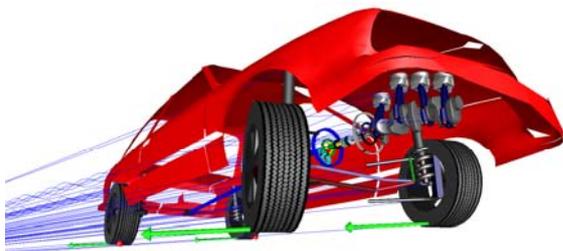


Bild 3: Komplexes Fahrzeugsimulationsmodell

In Bild 3 ist eine komplexere Anwendung zu sehen: Mit der VehicleDynamics-Bibliothek [1] wird die Fahrdynamik unter Verwendung der Modelica.Mechanics.MultiBody-Bibliothek [19] beschrieben, und mit der PowerTrain-Bibliothek wird der Antriebsstrang definiert, insbesondere ein detailliertes Modell des Automatikgetriebes.

2.2 Simulation von Modelica-Modellen

Zur Vorbereitung der Simulation muss aus dem Modelica-Modell simulationsfähiger Code erzeugt werden. Dazu wird in einem ersten Schritt eine Menge von Differential-, algebraischen und diskreten Gleichungen erzeugt, indem die aus allen Komponenten und ihrer Verschaltung resultierenden Gleichungen zu einem Gesamtgleichungssystem zusammengefasst werden. Dieses Gleichungssystem wird, wie im Folgenden beschrieben, weiterverarbeitet und schließlich zu ausführbarem Simulationscode übersetzt.

Eine direkte numerische Lösung des erwähnten Gesamtgleichungssystems ist in den meisten Fällen *ineffizient* und *unzuverlässig*. Daher werden symbolische Transformationsalgorithmen angewandt, um das Gleichungssystem dergestalt zu vereinfachen, dass es mit Standard-Integrationsverfahren besser gelöst werden kann. Die wesentlichen Schritte bei einer solchen Transformation sind Umsortieren und Aufteilen in explizit lösbare und implizit zu lösende Teilsysteme. Bei letzteren wird mittels BLT-(block-lower-triangular)-Transformation [5] versucht, sie in möglichst kleine, voneinander unabhängige algebraische Schleifen aufzuteilen. Eine weitere Reduzierung erfolgt über eine „intelligente Variablensubstitution“ mit dem Tearing-Verfahren [16].

Beim Zusammenschalten von Modelica-Komponenten kommt es häufig vor, dass in unterschiedlichen Komponenten definierte Systemzustände miteinander verkoppelt und nicht mehr voneinander unabhängig sind. Eine direkte numerische Lösung daraus abgeleiteter Systeme ist problematisch. Aus diesem Grund wird in Dymola eine geeignete Variablentransformation bzw. die „Dummy-Derivative“-Methode [12, 13, 20] angewendet, um ein zuverlässiges Lösen linearer bzw. nichtlinearer Differentialgleichungen und algebraischer Gleichungen (DAE) zu ermöglichen.

Aus Effizienzgründen und Praxiserwägungen werden bei der Modellierung physikalischer Effekte häufig Vereinfachungen vorgenommen, die strukturvariable Systeme zur Folge haben. Beispiele sind die ideale Diodenkennlinie, Schalter und Reibung. Hierfür hat sich die hybrid diskret-kontinuierliche Modellierungstechnik etabliert, die in Modelica und Dymola besonderen Eingang gefunden hat. In den resultierenden strukturvariablen Gleichungssystemen treten neben reellen Größen auch Boolesche bzw. Integer-Variablen als Unbekannte auf. In Modelica lösen alle Wertänderungen von logischen Ausdrücken standardmäßig ein

Ereignis aus, da hierbei potentiell eine Unstetigkeit eingeführt wird. Nach Detektion eines Ereignisses wird die Integration angehalten, die Änderung des logischen Ausdrucks durchgeführt und die Integration neu gestartet. Bei dieser Vorgehensweise sind Boolesche Variablen während der Integration bekannt (= Wert vom letzten Ereignis), nicht jedoch an einem Ereignispunkt. Dort liegt ein gekoppeltes System von reellen und Booleschen Unbekannten vor, das nicht mehr mit Standardalgorithmen gelöst werden kann. In [17] sind benötigte Erweiterungen dieser Algorithmen skizziert.

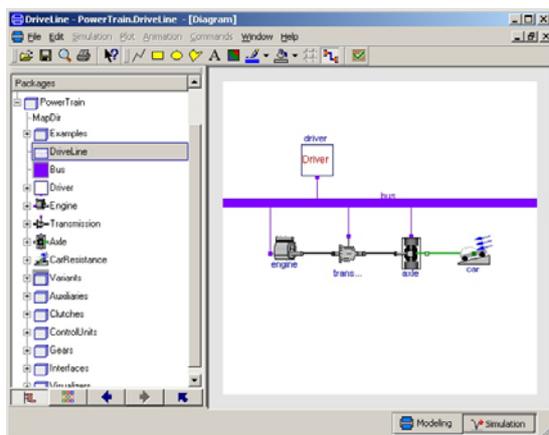


Bild 4: Bildschirmabzug von Dymola mit Modelldefinition

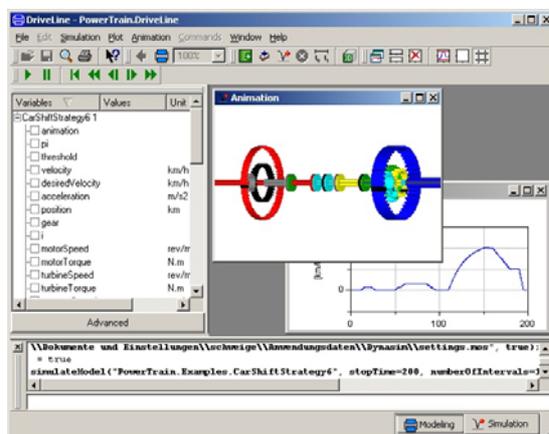


Bild 5: Bildschirmabzug von Dymola mit Simulation/Plot

Die beschriebenen Transformationsalgorithmen werden von der kommerziellen Modellierungs- und Simulationsumgebung Dymola (siehe Bild 4 und Bild 5) bereitgestellt, welche nahezu die gesamte Modelica-Sprache unterstützt. Zusätzlich enthält Dymola eine ganze Reihe weiterer Algorithmen, um die Effizienz und Robustheit zu steigern. Zum Beispiel werden die Gleichungen so umsortiert, dass alle *konstanten Gleichungen* vor Beginn der Simu-

lation nur einmal und alle *Ausgangsgleichungen* nur an Kommunikationszeitpunkten ausgewertet werden. Dymola kann sehr große Systeme (bis zu 100000 Gleichungen) behandeln (siehe z. B. [2]).

Abschließend wird das resultierende Gleichungssystem von Dymola als ANSI-C-Code ausgegeben, der mit einem Compiler in Maschinencode übersetzt wird. Dies kann wahlweise ein eigenständiges ausführbares Programm oder eine SIMULINK-S-Function im C-Mex-Format sein, welche in SIMULINK als Ein-/Ausgangsblock des Modells verwendet werden kann.

2.3 Echtzeitsimulation von Modelica-Modellen

Modelica und die Modelica-Simulationsumgebung Dymola werden aufgrund der Effizienz des erzeugten Codes insbesondere auch für Echtzeitanwendungen eingesetzt. Unter Ausnutzung symbolischer Gleichungsvorverarbeitung, bei der das Diskretisierungsverfahren des Integrators zusammen mit den Modellgleichungen gelöst wird (sogenannte Inline-Integration), ist die Echtzeitsimulation selbst komplexer Modelle möglich [6]. Eingesetzt wird sie zum Beispiel bei verschiedenen Automobilherstellern und -zulieferern zur Hardware-in-the-Loop-Simulation zum Test und zur automatisierten Applikation von Steuergeräten. Auch inverse Echtzeit-Dynamikmodelle für „embedded control“ bei Robotersteuerungen und bei neuartigen, im Flugversuch schon erfolgreich getesteten Autopilot-Reglern zum automatischen Landen von Verkehrsflugzeugen [10] können generiert werden.

3. Modellierung und Simulation von Antriebssträngen

Im folgenden wird beschrieben, wie Kraftfahrzeug-Antriebsstränge mit Modelica modelliert werden können. Der wesentliche, in Antriebssträngen zu modellierende Effekt ist Reibung. Sie ist zum Beispiel in einem Automatikgetriebe sowohl in den Kupplungen und Bremsen, als auch in der Verzahnung und den Lagern von Bedeutung.

Zahlreiche auf die Verwendung in Antriebssträngen zugeschnittene Komponenten werden in der vom DLR entwickelten PowerTrain-Bibliothek bereitgestellt. Neben 45 Einzelkomponenten sind zehn zum Teil sehr detaillierte Beispielmodelle enthalten, die die Verwendung demonstrieren und dem Benutzer als Hilfestellung bei der Entwicklung eigener Gesamtmodelle dienen. Eine Besonderheit der Bibliothek ist die effiziente und robuste Behandlung von geschwindigkeits- und lastabhängiger Reibung, welche unten detaillierter beschrieben wird. Bei der Entwicklung der PowerTrain-Bibliothek wurde unter anderem Wert auf Benutzerfreundlichkeit gelegt. So wurde ein vom Benutzer anpaßbarer Signalbus eingeführt, der die ansonsten entstehenden komplexen Signalverbindungen zwischen den Einzelkomponenten vermeidet. Bild 6 zeigt ein Gesamtmodell eines Antriebsstranges, welches als zentrale Kom-

ponente den Signalbus beinhaltet. Die weiteren Komponenten weisen eine einzige Verbindung mit dem Signalbus auf. Sie ist ausreichend, um alle benötigten Signale auf den Bus zu senden oder von ihm zu empfangen.

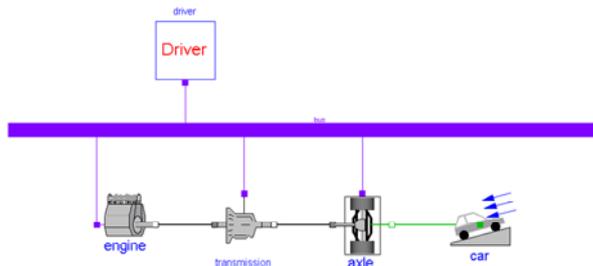


Bild 6: Modell DriveLine

Um ein zu starkes Anwachsen von Modellvarianten zu vermeiden, wurde es ermöglicht, bei den Antriebsstrang-Komponenten verschiedene Ausprägungen auszuwählen. Es gibt somit nur eine Getriebekomponente. Nach dem diese in das Modell eingefügt wurde, ermöglicht das in Bild 7 gezeigte Auswahlfenster die Angabe des tatsächlichen Getriebetyps.

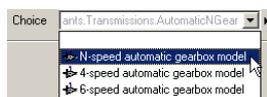


Bild 7: Auswahlfenster zur Festlegung des Getriebetyps

Für Plausibilitätsprüfungen und Demonstrationszwecke wurden alle Getriebekomponenten standardmäßig mit Animationsinformation und einer auf Antriebsstränge zugeschnittenen Parametrierung versehen. Die resultierende Ansicht ist in Bild 8 gezeigt. Um für Echtzeitanwendungen keine gesonderten Modelle anfertigen zu müssen, ist die Animation per Parameter abschaltbar mit der Folge, daß alle lediglich für Animation benötigten Gleichungen aus dem Modell entfernt werden.

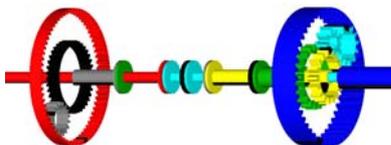


Bild 8: Animationsansicht eines Lepelletier-Radsatzes

Zur Veranschaulichung werden ein typisches, mit der PowerTrain-Bibliothek erstelltes Antriebsstrangmodell vorgestellt und exemplarische Ergebnisse gezeigt. Ausgehend von der in Bild 6 gezeigten Komponente *transmission* zeigt Bild 9 deren Inhalt auf der darunterliegenden Hierarchieebene. Neben dem eigentlichen Getriebe sind als Submodelle ein hydrodynamischer Drehmomentwandler, Steuergeräte für Gangwahl und Ansteuerung von Kupplungen und Bremsen des Getriebes und der Wandlerüberbrückungskupplung erkennbar.

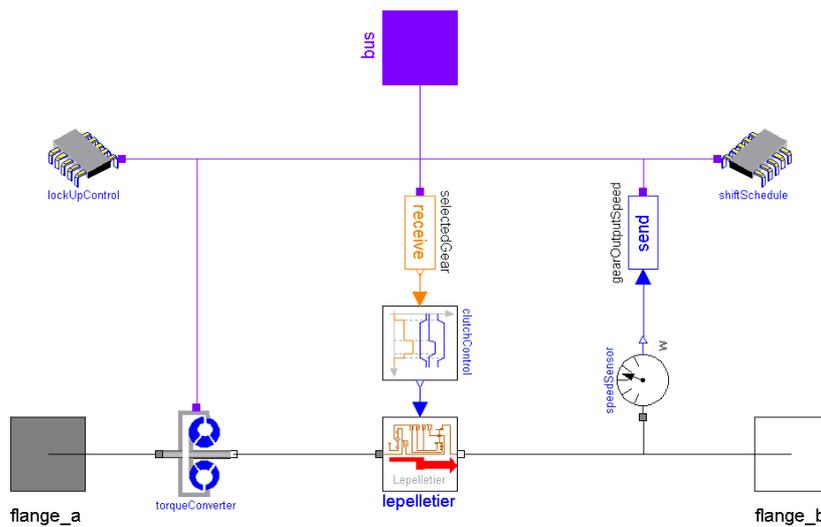


Bild 9: Modellstruktur für Sechsgang-Automatikgetriebe

Die nächsttiefere Hierarchieebene zeigt Bild 10, in dem der Inhalt der Komponente *lepelletier* dargestellt ist.

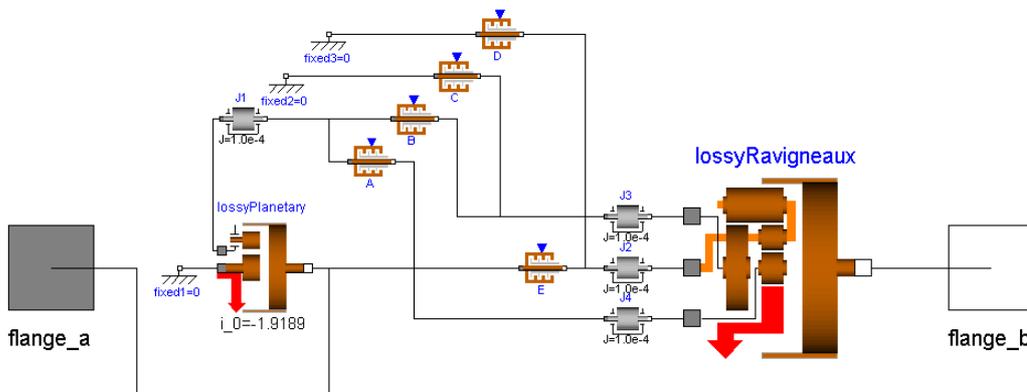


Bild 10: Lepelletier-Radsatz

Der Lepelletier-Radsatz [3] besteht aus einem Planetengetriebe und einem Ravigneaux-Radsatz. Mit dem Präfix *lossy* wird angedeutet, daß das in Abschnitt 4 skizzierte Reibmodell beinhaltet ist. Hervorzuheben ist, daß auch in den Kupplungen A, B, E und den Bremsen C, D detaillierte Reibmodelle verwendet werden. Die Simulation derart verschalteter Reibelemente ist kein triviales Problem, erweist sich bei Verwendung von Modelica/Dymola jedoch als besonders effizient und robust. Wie Bild 11 demonstriert, können damit beispielsweise Aussagen über die in den Kupplungen auftretenden Differenzdrehzahlen getroffen werden.

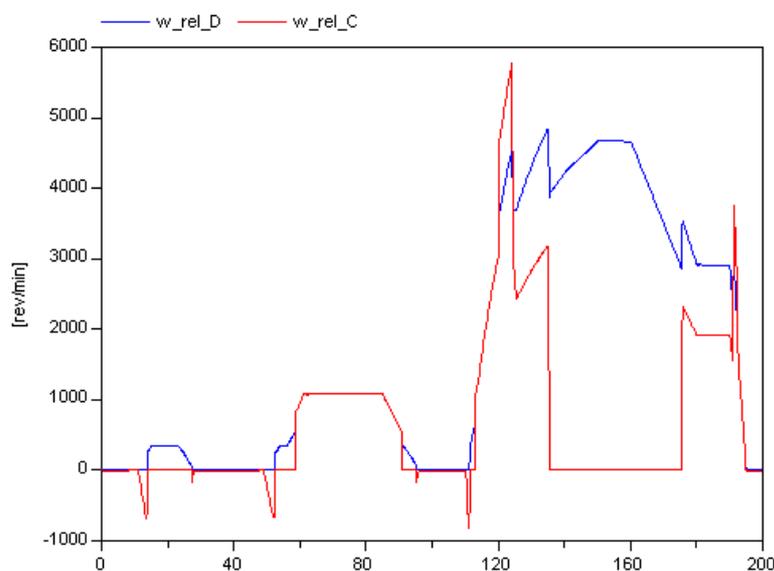


Bild 11: Zeitverläufe der Differenzdrehzahlen in den Kupplungen C und D des Sechsgang-Automatikgetriebes

4. Modellierung von Last- und geschwindigkeitsabhängiger Reibung

Im folgenden wird die eingangs erwähnte robuste und effiziente Behandlung geschwindigkeits- und momentenabhängiger Reibung am Beispiel eines einfachen Getriebes beschrieben. In [21] wird dieses Reibungsmodell als

$$\tau_B = i(-\tau_A + \Delta\tau) \quad \text{mit} \quad \Delta\tau = \begin{cases} (1 - \hat{\eta}_{mf})\tau_A + \hat{\tau}_{bf} & : \omega_A \neq 0 \\ \text{so, daß } \dot{\omega}_A = 0 & : \omega_A = 0 \end{cases} \quad (1)$$

hergeleitet, wobei τ_A und τ_B Flanshmomente, i die Getriebeübersetzung und $\Delta\tau$ das Reibmoment darstellen. Die Winkelgeschwindigkeit von Flansch A ist ω_A . Das Reibmoment wird für den Gleitzustand ($\omega_A \neq 0$) und den Haftzustand ($\omega_A = 0$) unterschiedlich berechnet. Während es im Gleitzustand in Abhängigkeit von den Parametern Verzahnungswirkungsgrad

$\hat{\eta}_{mf}$ und additiver Lagerreibung $\hat{\tau}_{bf}$ berechnet wird, wird es im Haftzustand implizit so bestimmt, daß die Winkelbeschleunigung Null ist und der Haftzustand erhalten bleibt. Eine Umschaltung zwischen beiden Zuständen erfolgt folgendermaßen: Vom Haft- in den Gleitzustand wird umgeschaltet, wenn die Differenz der äußeren Momente das aus $\hat{\eta}_{mf}$ und $\hat{\tau}_{bf}$ bestimmte Haftreibungsmoment überschreitet. Umgekehrt wird vom Gleit- in den Haftzustand umgeschaltet, wenn $\omega_A = 0$ erfüllt ist.

Die Größen $\hat{\eta}_{mf}$ und $\hat{\tau}_{bf}$ werden in Abhängigkeit von der Leistungsflußrichtung ($\text{sgn } \tau_A \omega_A = \pm 1$) und der Winkelgeschwindigkeit ω_A bestimmt, was zusammengefaßt in Tabelle 1 wiedergegeben ist. Dabei stellen die Indizes $i = 1, 2$ bei η_{mfi} und τ_{bfi} die Zuordnung zur entsprechenden Leistungsflußrichtung her.

Tabelle 1: Berechnung des Reibmomentes im Gleitzustand

	$\tau_A \geq 0$	$\tau_A < 0$
$\omega_A > 0$	$\Delta\tau = [1 - \eta_{mf1}(\omega_A)]\tau_A + \tau_{bf1} $	$\Delta\tau = [1 - 1/\eta_{mf2}(\omega_A)]\tau_A + \tau_{bf2} $
$\omega_A < 0$	$\Delta\tau = [1 - 1/\eta_{mf2}(\omega_A)]\tau_A - \tau_{bf2} $	$\Delta\tau = [1 - \eta_{mf1}(\omega_A)]\tau_A - \tau_{bf1} $

Graphisch läßt sich die Abhängigkeit des Reibmomentes vom Lastmoment und der Winkelgeschwindigkeit wie in Bild 12 gezeigt darstellen.

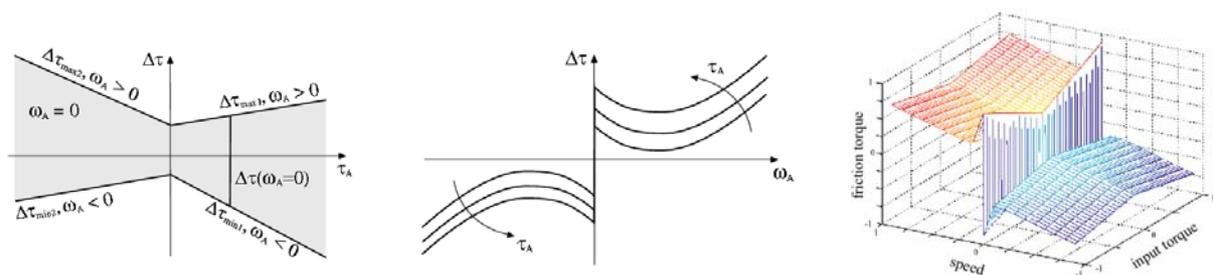


Bild 12: Reibmoment im Gleit- und Haftzustand

Die verschiedenen Bereiche zur Berechnung von $\Delta\tau$ sind im linken Teilbild ersichtlich. Im Gleitzustand nimmt $\Delta\tau$ abhängig vom Vorzeichen von ω_A entweder einen Wert auf der oberen oder unteren Begrenzungslinie an. Mit der Maßgabe $\dot{\omega}_A = 0$ stellt sich im Haftzustand ein Wert zwischen den beiden Begrenzungslinien ein. Das mittlere Teilbild zeigt $\Delta\tau$

mit ω_A als Abszisse und τ_A als Lageparameter. Deutlich ist die Verwandtschaft zu einem rein geschwindigkeitsabhängigen Reibmodell erkennbar. Der einzige Unterschied besteht in der zusätzlichen Abhängigkeit des Reibmomentes vom Lastmoment τ_A . Eine kombinierte Darstellung zeigt das rechte Teilbild.

Dieses Reibmodell wurde in Modelica implementiert und wird in vielen Komponenten der PowerTrain-Bibliothek verwendet. Die Abhängigkeit der Größen $\eta_{mf1,2}$ und $\tau_{bf1,2}$ von ω_A wird mittels Interpolation in einer vom Benutzer bestimmbaren Tabelle realisiert. Diese Darstellung ermöglicht die anschauliche Übernahme von Meßdaten. Die Robustheit des Reibmodells ergibt sich aus zwei Umständen. Zunächst werden durch die besondere Formulierung nichtlineare Gleichungssysteme vermieden, für welche nur unzureichende Löser existieren würden. Weiter wird der Haftzustand explizit berücksichtigt, was bei vergleichbaren Getriebewirkungsgradmodellen nicht der Fall ist. Bei Verwendung von Lösern mit variabler Schrittweite zur Simulation solcher Modelle ist dann häufig eine Reduzierung der Schrittweite gegen Null beobachtbar, was in der Regel zum Abbruch führt.

5. Modellierung 3D-mechanischer Effekte von 1D-Antriebssträngen

Aus Gründen der Effizienz und der leichteren Modellbildung werden Antriebsstränge in Modelica eindimensional betrachtet. Eine Kopplung solcher Modelle mit zum Beispiel 3D-Fahrdynamikmodellen wäre nicht korrekt. So würden die Abstütz- und Kreiselmomente nicht berücksichtigt werden. Um dennoch eine Kopplung zu ermöglichen, stehen für diesen Anwendungsfall spezielle Komponenten zur Verfügung [22].

Die Grundidee der Kopplung von 1D-Antriebsstrangmodellen mit 3D-Fahrdynamikmodellen sieht vor, daß alle statischen Eigenschaften durch einen 3D-Körper, im folgenden als Träger bezeichnet, abgebildet werden, der der Mehrkörper-Bibliothek Modelica.Mechanics.MultiBody entnommen wird. Er repräsentiert einen Antriebsstrang unter der Annahme feststehender Wellen mittels der gemeinsamen Masse, des gemeinsamen Schwerpunkts und des gemeinsamen Trägheitstensors aller Antriebsstrangbestandteile. Voraussetzung für das beschriebene Vorgehen ist, daß alle rotierenden Körper des Antriebsstrangs rotationssymmetrisch sind. Der Träger wird am 3D-Mehrkörpersystem befestigt, an ihm wiederum werden die 1D-Antriebsstrangkomponenten angeschlossen.

In den nächsten Abschnitten wird an zwei Beispielen gezeigt, wie die für eine korrekte Kombination von 1D- und 3D-Modellen nötigen Komponenten hergeleitet werden.

5.1 Rotatorische Massenträgheit

Die zusätzlich für die Berücksichtigung der 3D-Effekte erforderlichen Gleichungen werden zunächst für die 1D-Massenträgheit hergeleitet, mit der Wellen des Antriebsstrangs modelliert werden. Wellen in einem räumlich bewegten Antriebsstrang üben auf den Träger gyroscopische Momente aus. Dieser Effekt wird am Beispiel des in Bild 13 gezeigten, sogenannten Gyrostats erläutert.

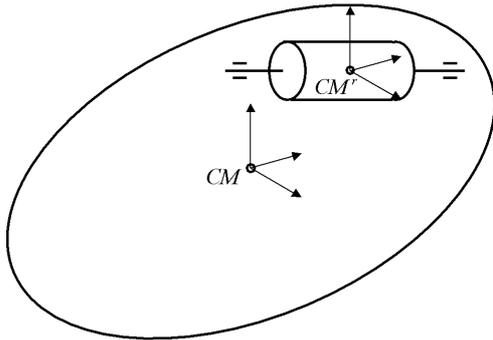


Bild 13: Gyrostat bestehend aus Träger und symmetrischem Rotor

Es werden zwei trägerfeste Koordinatensysteme gleicher Orientierung eingeführt. Das Trägersystem hat seinen Ursprung im Schwerpunkt CM des Gesamtsystems, das Rotorsystem im Schwerpunkt CM^r des Rotors.

Gemäß beispielsweise [15, 24] sind die Bewegungsgleichungen dieses kombinierten Systems mit

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \mathbf{J}^r \dot{\boldsymbol{\omega}}_{rel} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega} + \mathbf{J}^r \boldsymbol{\omega}_{rel}) = \boldsymbol{\tau} \quad (2)$$

gegeben. Dabei ist $\boldsymbol{\omega}$ die absolute Winkelgeschwindigkeit des Trägers, $\boldsymbol{\omega}_{rel}$ die Winkelgeschwindigkeit des Rotors relativ zum Träger, \mathbf{J} der Trägheitstensor des Gesamtsystems bezüglich des Schwerpunkts CM , \mathbf{J}^r der Trägheitstensor des Rotors bezüglich des Schwerpunkts CM^r und $\boldsymbol{\tau}$ ein externes Moment. Alle Vektoren und Tensoren werden im Trägerkoordinatensystem betrachtet. Umsortieren liefert

$$\underbrace{\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}}_{\boldsymbol{\tau}_{body}} + \underbrace{\mathbf{J}^r \dot{\boldsymbol{\omega}}_{rel} + \boldsymbol{\omega} \times \mathbf{J}^r \boldsymbol{\omega}_{rel}}_{\boldsymbol{\tau}_{gyro}} = \boldsymbol{\tau} \quad (3)$$

Vergleicht man dies mit den Bewegungsgleichungen für einen Starrkörper, läßt sich erkennen, daß der Term $\boldsymbol{\tau}_{body}$ die Einwirkung des Gesamtsystems mit feststehendem Rotor und $\boldsymbol{\tau}_{gyro}$ die zusätzliche Einwirkung aufgrund der Relativbewegung des Rotors zum Gesamtsystem

stem repräsentiert. Wie eingangs beschrieben werden die Eigenschaften eines Antriebsstrangs mit feststehenden Wellen vom Trägerkörper modelliert. Entsprechend muß in einer erweiterten 1D-Massenträgheit lediglich der Term $\boldsymbol{\tau}_{gyro}$ berücksichtigt werden.

Nachdem die Rotordrehachse mit einer Hauptträgheitsachse des Rotors zusammenfällt, ergibt sich die Vereinfachung

$$\mathbf{n}J^r\dot{\omega}_{rel} + \boldsymbol{\omega} \times \mathbf{n}J^r\omega_{rel} = \boldsymbol{\tau}_{gyro} , \quad (4)$$

wobei \mathbf{n} ein Einheitsvektor in Richtung der Rotordrehachse , J^r das Rotorträgheitsmoment um \mathbf{n} und ω_{rel} der Absolutwert der Relativwinkelgeschwindigkeit des Rotors bezüglich des Trägers sind.

Zusätzlich zu den in (4) lediglich betrachteten Rotationsfreiheitsgraden des Gesamtsystems existiert ein weiterer Rotationsfreiheitsgrad für den Rotor selbst, nachdem ein externes Moment $\tau^r(t)$ auf ihn ausgeübt wird. In [15, 24] werden die dazugehörigen Bewegungsgleichungen als

$$\mathbf{n}^T (\mathbf{J}^r \dot{\boldsymbol{\omega}} + \mathbf{J}^r \dot{\boldsymbol{\omega}}_{rel}) = \mathbf{n}^T \boldsymbol{\tau}^r \quad (5)$$

hergeleitet. Ähnlich wie oben kann die Gleichung vereinfacht werden zu

$$J^r \mathbf{n} \dot{\boldsymbol{\omega}} + J^r \dot{\omega}_{rel} = \tau^r(t) . \quad (6)$$

Die Gleichungen von (4) und (6) wurden in einer neuen Komponente Rotor1D implementiert, die im Vergleich zur 1D-Massenträgheit zusätzlich noch einen 3D-Frame-Konnektor erhält.

Zur Vermeidung einer algebraischen Schleife kann optional der Term $J^r \mathbf{n} \dot{\boldsymbol{\omega}}$ in (6) vernachlässigt werden, wenn der Rotor viel schneller beschleunigt wird als der Träger. Dies ist beispielsweise in Kfz-Antriebssträngen der Fall. Als Folge gibt es auf Beschleunigungsebene nur eine Einwirkung vom Rotor auf den Träger und nicht umgekehrt.

5.2 Kegelradgetriebe

Abstützmomente von Kegelradgetrieben, allgemeiner Getrieben mit nichtparallelen Achsen, siehe Bild 14, können in 1D-modellierten Antriebssträngen nicht korrekt berechnet werden, nachdem im allgemeinen die Rotationsachsen des An- und Abtriebsflansches und die Richtung des Abstützmomentenvektors auseinanderfallen. Daher wird auch für diesen Fall eine spezielle Komponente benötigt, die darüberhinaus ebenfalls einen 3D-Frame-Konnektor haben muß.



Bild 14: Kegelaradgetriebe als Vertreter eines Getriebes mit nichtparallelen Achsen

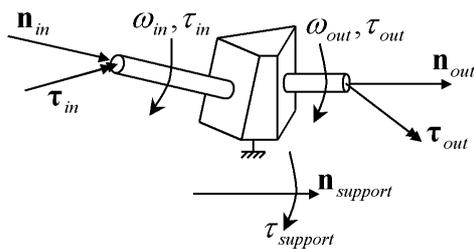


Bild 15: Größen an einem Getriebe mit nichtparallelen Achsen

Mit der Übersetzung i , den Wellen-Winkelgeschwindigkeiten ω und den Wellenmomenten τ ist ein Kegelaradgetriebe in Anlehnung an Bild 15 charakterisiert durch, siehe [9],

$$i = \frac{\omega_{in}}{\omega_{out}} = -\frac{\tau_{out}}{\tau_{in}} . \quad (7)$$

Die Indices beziehen sich gemäß Bild 15 auf die Antriebs- und die Abtriebswelle. Mit \mathbf{n}_{in} und \mathbf{n}_{out} als Vektoren in Richtung der Antriebs- beziehungsweise der Abtriebswelle ergibt sich die 3D-Momentenbilanz zu

$$\mathbf{0} = \tau_{in} \frac{\mathbf{n}_{in}}{|\mathbf{n}_{in}|} + \tau_{out} \frac{\mathbf{n}_{out}}{|\mathbf{n}_{out}|} + \tau_{support} . \quad (8)$$

Eine Implementierung dieser Gleichungen in einer Modelica-Komponente ist in der Mehrkörper-Bibliothek Modelica.Mechanics.MultiBody enthalten.

6. Anwendung: Automatische Applikation von Getriebesteuergeräten

Im folgenden wird eine mögliche Anwendung der PowerTrain-Bibliothek beschrieben. Während in der Praxis Steuergeräte von einem Applikationsingenieur häufig noch manuell parametrisiert werden, wird unter *automatischer Applikation* das optimierungsbasierte Einstellen von Steuergeräteparametern in einer Hardware-in-the-Loop-Umgebung verstanden, siehe Bild 16. Dabei werden die Parameter solange gezielt verändert, bis bestimmte quantifizierbare Kriterien erfüllt sind.

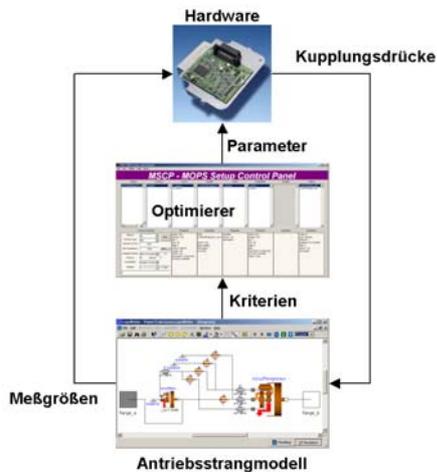


Bild 16: Aufbau einer Entwurfsumgebung für Automatische Applikation

Ein zweckmäßiges Vorgehen zur Echtzeitsimulation von Modelica-Modellen ist die Einbindung in Simulink, von wo aus mittels Real-Time Workshop Code für verschiedene Echtzeitplattformen erzeugt werden kann.

Notwendig für automatische Applikation ist die Anbindung einer Optimierung an das Echtzeitsystem. Am DLR wurde die Optimierungsumgebung Multi-Objective Parameter Synthesis (MOPS) [8] entwickelt, welche die Umsetzung von Entwurfsproblemen zu wohlformulierten, mehrzieligen Optimierungsaufgaben in MATLAB [11] unterstützt. Neben verschiedenen leistungsfähigen Optimierungsalgorithmen sind vordefinierte regelungstechnische Kriterien ebenso wie generische Skripte zur Ablaufsteuerung enthalten. Zudem stehen Visualisierungswerkzeuge zur Überwachung des Entwurfsprozesses zur Verfügung. Weitere Funktionen helfen bei der Bewältigung großer Parameter- und Kriterienmengen, ermöglichen bei zeitaufwändigen Berechnungen verteiltes Rechnen und erlauben die Verwendung externer Simulations- und Analyseserver. Parameterschätzung für Identifikationsprobleme wird ebenso unterstützt wie optimierungsbasierte Robustheitsanalyse.

Zur Demonstration des Zusammenwirkens der Werkzeugkette wurde das in Abschnitt 3 vorgestellte Antriebsstrangmodell mit Sechsganggetriebe auf dem Echtzeitsystem xPC Target [11] simuliert. Für einen vorgegebenen Fahrzyklus (siehe Bild 17) wurden die Schaltkennlinien (siehe Bild 18) und die Kupplungsdruckverläufe (siehe Bild 19) derart optimiert, dass sich gutes Folgeverhalten und ein geringer Kraftstoffverbrauch ergeben.

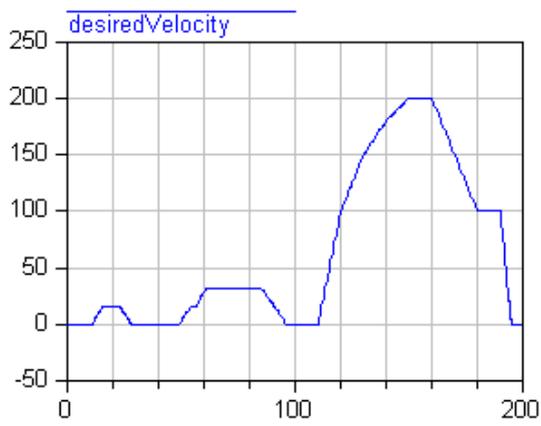


Bild 17: Fahrzyklus

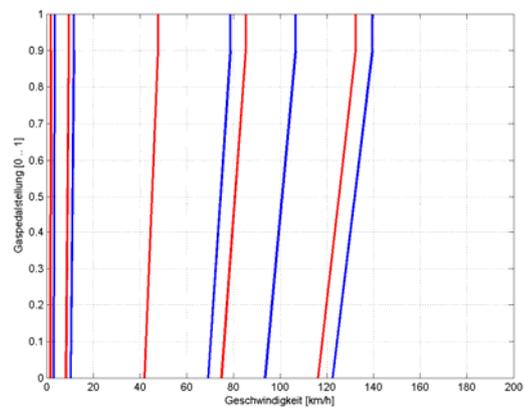
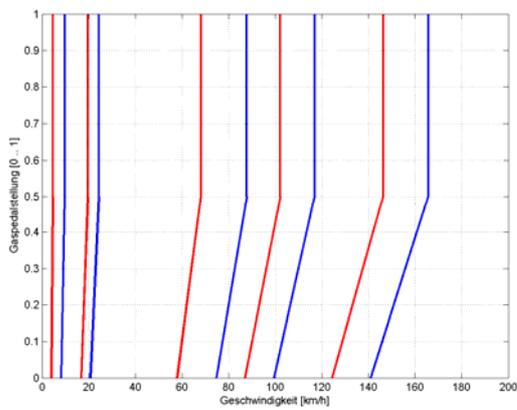


Bild 18: Schaltkennlinien vor (links) und nach (rechts) Optimierung

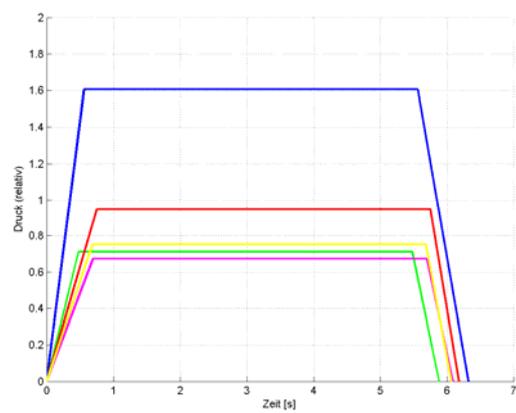
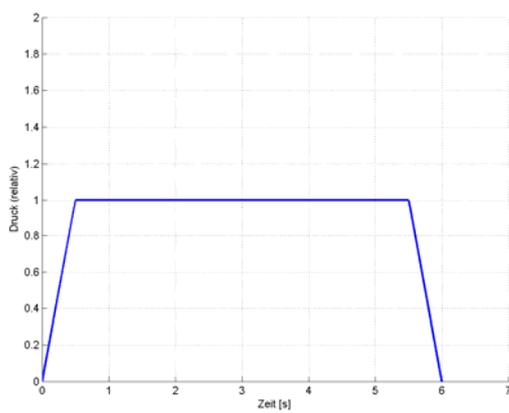


Bild 19: Kupplungsdruckverläufe vor (links) und nach (rechts) Optimierung

Bild 20 und Bild 21 zeigen verschiedene Zeitverläufe vor und nach der Optimierung.

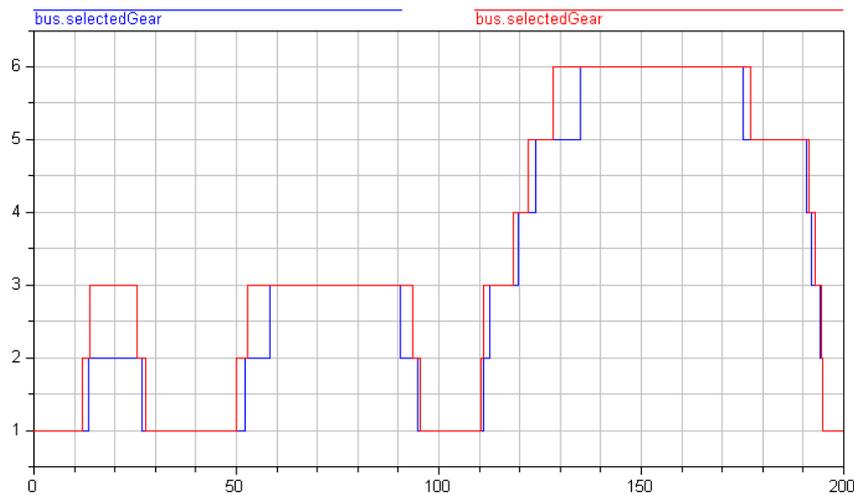


Bild 20: Simulierter Zeitverlauf des den eingelegten Gang repräsentierenden Signals vor (blau) und nach Optimierung (rot)



Bild 21: Simulierter Zeitverlauf der Motordrehzahl vor (blau) und nach Optimierung (rot)

7. Zusammenfassung und Ausblick

Mit der Modelica-Komponentenbibliothek PowerTrain steht ein leistungsfähiges Werkzeug zur Modellierung und Simulation der Dynamik von Schaltvorgängen zur Verfügung, wobei sich in der Modelica-Simulationsumgebung Dymola auch die Simulation von gekoppelten Reibelementen als robust und effizient erweist. Entsprechend ist die PowerTrain-Bibliothek bei verschiedenen Fahrzeugherstellern und -zulieferern im Einsatz. Ende 2005 wird eine neue Version der PowerTrain-Bibliothek erscheinen. Vorgesehene Erweiterungen sind Modelle etwa für Allrad-Antriebsstränge, Hybridfahrzeuge und Getriebeelastizitäten.

Literatur

- [1] Andreasson, J.: VehicleDynamics library. 3rd International Modelica Conference. 2003, S. 11-18
- [2] Bowles, P., Tiller, M., Elmqvist, H., Brück, D., Mattsson, S. E., Möller, A., Olsson, H., Otter M.: Feasibility of Detailed Vehicle Modeling. SAE World Congress 2001.
- [3] Dach, H., Gruhle, W.-D., Köpf, P.: Pkw-Automatgetriebe. Landsberg/Lech: Verlag Moderne Industrie 2001
- [4] Dynasim. <http://www.dynasim.se/>
- [5] Elmqvist, H.: A Structured Model Language for Large Continuous Systems. Lund Institute of Technology, Diss. 1978
- [6] Elmqvist, H., Mattsson, S. E., Olsson, H., Andreasson, J., Otter, M., Schweiger, C., Brück, D.: Realtime Simulation of Detailed Vehicle and Powertrain Dynamics. SAE 2004 World Congress. 2004
- [7] Fritzson, P.: Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE Press 2004
- [8] Joos, H.-D., Bals, J., Looye, G., Schnepper, K., Varga, A.: A multi-objective optimisation-based software environment for control systems design. CACSD '02. 2002, S. 7-14
- [9] Loomann, J.: Zahnradgetriebe. Springer 1996
- [10] Looye, G., Joos, H.-D., Willemsen, D.: Application of an Optimization-based Design Process for Robust Autoland Control Laws. AIAA Guidance, Navigation, and Control Conference and Exhibit. 2001, S. 1-11
- [11] The MathWorks. <http://www.mathworks.com/>
- [12] Mattsson, S. E., Söderlind, G.: Index reduction in differential-algebraic equations using dummy derivatives. SIAM Journal of Scientific and Statistical Computing 14 (1993) S. 677-692
- [13] Mattsson, S. E., Olsson, H., Elmqvist, H.: Dynamic Selection of States in Dymola. 1st International Modelica Workshop. 2000, S. 61-67
- [14] Modelica. <http://www.modelica.org/>
- [15] Otter, M.: Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter. Ruhr-Universität Bochum Diss. 1994
- [16] Otter, M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 4, Transformationsalgorithmen. at Automatisierungstechnik 47 (1999) 4 S. A13-A16
- [17] Otter, M., Elmqvist, H., Mattsson, S. E.: Hybrid Modeling in Modelica based on the Synchronous Data Flow Principle. CACSD '99. 1999

- [18] Otter, M., Elmqvist, H.: Modelica – Language, Libraries, Tools, Workshop and EU Project RealSim. Simulation News Europe (2000), S. 3-8
- [19] Otter, M., Elmqvist, H., Mattsson, S. E.: The New Modelica MultiBody Library. 3rd International Modelica Conference. 2003, S. 311-330
- [20] Pantelides, C.: The Consistent Initialization of Differential-Algebraic Systems. SIAM Journal of Scientific and Statistical Computing 9 (1988) S. 213-231
- [21] Pelchen, C., Schweiger, C., Otter, M.: Modeling and Simulating the Efficiency of Gearboxes and of Planetary Gearboxes. 2nd International Modelica Conference. 2002, S. 257-266
- [22] Schweiger, C., Otter, M.: Modelling 3D Mechanical Effects of 1D Powertrains. 3rd International Modelica Conference. 2003, S. 149-158
- [23] Tiller, M.: Introduction to Physical Modeling with Modelica. Kluwer Academic Publishers 2001
- [24] Wittenburg, J.: Dynamics of Systems of Rigid Bodies. B. G. Teubner Verlag 1977