

**M. Arnold, T. Schierz**

Effizienz und Robustheit numerischer Kopplungsalgorithmen im MODELISAR Co-Simulation-Interface. In: G. Elst, editor, *Tagungsband ASIM-Treffen STS/GMMS 2009 und DASS 2009, Dresden, 05.03.-06.03.09.* – Fraunhofer-Institut für Integrierte Schaltungen IIS, Institutsteil Entwurfsautomatisierung EAS, Dresden, ISBN 978-3-8167-7981-0, 2009.

**Abstract.** Die Entwicklung von vielen in industriellen Anwendungen genutzten Simulationswerkzeugen geht bis in die 1980'er Jahre zurück. Zur computergestützten Analyse gekoppelter physikalischer Phänomene und gekoppelter technischer Systeme hat heute die *Kopplung* von solchen etablierten mono-disziplinären Simulationswerkzeugen zum Modellaufbau für gekoppelte Systeme und zur numerischen Lösung der zugehörigen gekoppelten mathematischen Modellgleichungen große praktische Bedeutung. Eine solche *Co-Simulation* oder *Simulatorkopplung* nutzt die typische modulare Struktur gekoppelter technischer Systeme nicht nur für den getrennten, modularen Modellaufbau der Teilsysteme, sondern stellt auch die mathematischen Modellgleichungen für die einzelnen Teilsysteme getrennt voneinander auf und verwendet für die Modellgleichungen jedes einzelnen Teilsystems ein eigenes, an die spezielle Struktur dieser Modellgleichungen angepasstes Zeitintegrationsverfahren (*modulare Zeitintegration* [1]).

Im Rahmen des ITEA2-Projektes MODELISAR hat sich ein deutsches Konsortium aus 14 Industrie- und Forschungspartnern gemeinsam mit weiteren 14 Partnern aus 4 europäischen Ländern u. a. zum Ziel gesetzt, eine standardisierte, leistungsfähige, offene Co-Simulationsplattform für Multiphysik-Modelle zu entwickeln (*Functional Mockup Interface*). Im Vortrag werden erste Ergebnisse zur Verbesserung der Effizienz und Robustheit numerischer Kopplungsalgorithmen in modularen Zeitintegrationsverfahren für derartige Co-Simulationsanwendungen vorgestellt.

## References

- [1] K. Jackson. A survey of parallel numerical methods for initial value problems for ordinary differential equations. *IEEE Transactions on Magnetics*, 27:3792–3797, 1991.

**Contact:** martin.arnold@mathematik.uni-halle.de



# Effizienz und Robustheit numerischer Kopplungsalgorithmen im MODELISAR Co-Simulation-Interface

Martin Arnold, Tom Schierz

Martin-Luther-Universität Halle-Wittenberg

NWF III – Institut für Mathematik

{martin.arnold,tom.schierz}@mathematik.uni-halle.de

Tel.: 0345/5524653, Fax: 0345/5527004

## Zusammenfassung

Die Entwicklung von vielen in industriellen Anwendungen genutzten Simulationswerkzeugen geht bis in die 1980'er Jahre zurück. Zur computergestützten Analyse gekoppelter physikalischer Phänomene und gekoppelter technischer Systeme hat heute die *Kopplung* von solchen etablierten mono-disziplinären Simulationswerkzeugen zum Modellaufbau für gekoppelte Systeme und zur numerischen Lösung der zugehörigen gekoppelten mathematischen Modellgleichungen große praktische Bedeutung. Eine solche *Co-Simulation* oder *Simulatorkopplung* nutzt die typische modulare Struktur gekoppelter technischer Systeme nicht nur für den getrennten, modularen Modellaufbau der Teilsysteme, sondern stellt auch die mathematischen Modellgleichungen für die einzelnen Teilsysteme getrennt voneinander auf und verwendet für die Modellgleichungen jedes einzelnen Teilsystems ein eigenes, an die spezielle Struktur dieser Modellgleichungen angepasstes Zeitintegrationsverfahren (*modulare* Zeitintegration [1]).

Im Rahmen des ITEA2-Projektes MODELISAR hat sich ein deutsches Konsortium aus 14 Industrie- und Forschungspartnern gemeinsam mit weiteren 14 Partnern aus 4 europäischen Ländern u. a. zum Ziel gesetzt, eine standardisierte, leistungsfähige, offene Co-Simulationsplattform für Multiphysik-Modelle zu entwickeln (*Functional Mockup Interface*). Im Vortrag werden erste Ergebnisse zur Verbesserung der Effizienz und Robustheit numerischer Kopplungsalgorithmen in modularen Zeitintegrationsverfahren für derartige Co-Simulationsanwendungen vorgestellt.

## 1 Einleitung

Die meisten der heute industriell genutzten Simulationswerkzeuge zur Analyse der Dynamik komplexer Systeme (u. a. Mechanik, Hydraulik, Elektronik, Pneumatik, Regelungstechnik) haben gemeinsame methodische Wurzeln in der Systemdynamik und beruhen auf

einem vom konkreten Anwendungsbereich unabhängigen Netzwerkkonzept [2]. Dennoch haben sie sich letztendlich doch zu hoch spezialisierten mono-disziplinären Werkzeugen entwickelt.

Dieses Konzept stößt an prinzipielle Grenzen bei der Simulation gekoppelter physikalischer Phänomene (z. B. Fluid–Struktur–Wechselwirkung) und gekoppelter technischer Systeme. Klassische Beispiele sind die Wechselwirkung zwischen den mechanischen und den hydraulischen Baugruppen einer Servolenkung und das Zusammenwirken von mechanischen, hydraulischen und elektronischen Komponenten von *mechatronischen* Systemen wie z. B. den Systemen zur Verbesserung der aktiven und passiven Sicherheit von Kraftfahrzeugen [3].

Aus mathematischer Sicht führen diese gekoppelten technischen Systeme im instationären Fall auf gekoppelte Differentialgleichungssysteme

$$\dot{\mathbf{x}}^{(i)}(t) = \mathbf{f}^{(i)}(t, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(r)}), \quad (i = 1, \dots, r) \quad (1)$$

mit  $r$  Teilsystemen, die über die rechten Seiten  $\mathbf{f}^{(i)}$  miteinander verbunden sind. Der heute aus praktischer Sicht weitaus wichtigste Fall sind zwei, höchstens drei gekoppelte Teilsysteme ( $r = 2$  bzw.  $r = 3$ ).

In praktischen Anwendungen hat die *Kopplung* von etablierten mono-disziplinären Simulationswerkzeugen zum Modellaufbau des gekoppelten Systems und zur numerischen Lösung der gekoppelten Modellgleichungen (1) große Bedeutung. Eine solche *Co-Simulation* oder *Simulatorkopplung* nutzt die typische modulare Struktur gekoppelter technischer Systeme nicht nur für den getrennten, modularen Modellaufbau der Teilsysteme, sondern verwendet auch für die Modellgleichungen  $\dot{\mathbf{x}}^{(i)} = \mathbf{f}^{(i)}(\dots)$  jedes einzelnen Teilsystems ein eigenes, an die spezielle Struktur dieser Modellgleichungen angepasstes Zeitintegrationsverfahren (*modulare* Zeitintegration gekoppelter Systeme [1]).

Im Vortrag werden klassische und modulare Zeitintegrationsverfahren für gekoppelte Differentialgleichungssysteme einander gegenübergestellt und Entwicklungspotenziale für die bisher meist in Co-Simulationsumgebungen verwendeten sehr einfachen Kopplungsstrategien für die Kommunikation zwischen den Teilsystemen aufgezeigt. Ziel ist die Verbesserung der Effizienz und Robustheit der numerischen Kopplungsalgorithmen in modularen Zeitintegrationsverfahren.

## 2 Einführendes Beispiel

Abb. 1 zeigt als typisches Beispiel für ein gekoppeltes technisches System die Kopplung zwischen mechanischen und hydraulischen Baugruppen in der Simulation mechatronischer Systeme [3, 4, 5]. Die mechanischen Baugruppen des Fahrzeugs mit servo-hydraulischer Lenkung wurden im 3D-Mehrkörpersystem–Simulationswerkzeug SIMPACK modelliert. Für die mechanischen und hydraulischen Komponenten der servo-hydraulischen Lenkung sowie die zugehörigen Regler stand ein Modelica-Modell zur Verfügung. Eine Zahnstange bildet die physikalische Schnittstelle der beiden Teilsysteme “Fahrzeug” (engl. car) und

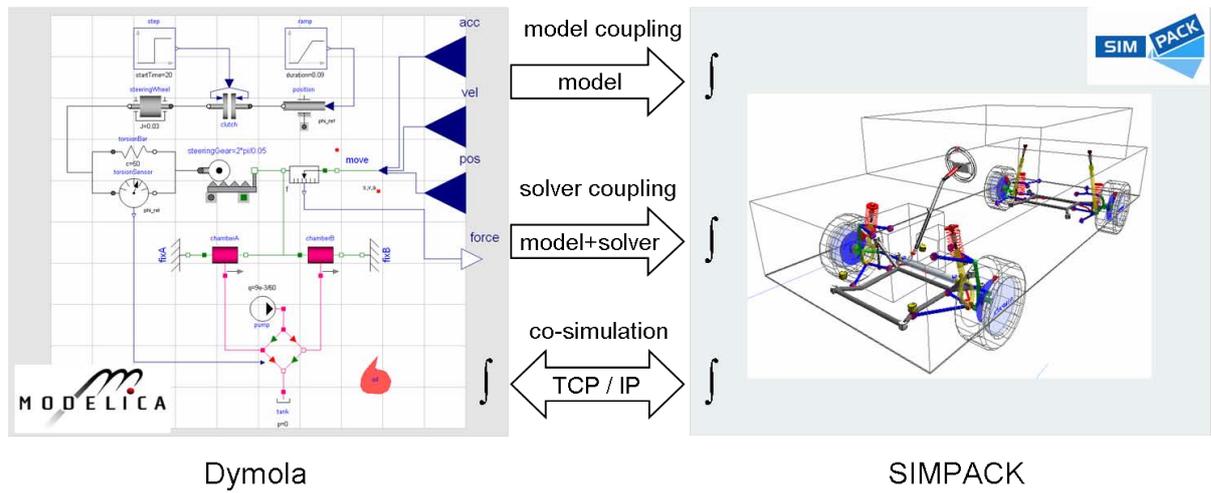


Abbildung 1: Kopplung von Simulationswerkzeugen zur dynamischen Simulation eines Fahrzeugs mit servo-hydraulischer Lenkung [4, 5].

“Lenkung” (engl. steering).

Die Auslenkung des Lenkrads (Lage, Geschwindigkeit, Beschleunigung) ist Eingangsgröße des geregelten hydraulischen Teilsystems. Dieses erzeugt als Systemausgang eine Aktuatorkraft, die auf die mechanischen Komponenten der Lenkung wirkt. Erweitert man das Differentialgleichungsmodell (1) um die Ein- und Ausgänge  $\mathbf{u}(t)$  bzw.  $\mathbf{y}(t)$  der  $r = 2$  Teilsysteme, so haben die gekoppelten Modellgleichungen die Form

$$\left. \begin{aligned} \dot{\mathbf{x}}^{[c]}(t) &= \varphi^{[c]}(\mathbf{x}^{[c]}(t), \mathbf{u}^{[c]}(t)) \\ \mathbf{y}^{[c]}(t) &= \gamma^{[c]}(\mathbf{x}^{[c]}(t)) \end{aligned} \right\}, \quad \left. \begin{aligned} \dot{\mathbf{x}}^{[s]}(t) &= \varphi^{[s]}(\mathbf{x}^{[s]}(t), \mathbf{u}^{[s]}(t)) \\ \mathbf{y}^{[s]}(t) &= \gamma^{[s]}(\mathbf{x}^{[s]}(t)) \end{aligned} \right\} \quad (2a)$$

mit den Kopplungsbedingungen

$$\left. \begin{aligned} \mathbf{u}^{[s]}(t) &= \mathbf{y}^{[c]}(t), \quad (\text{Auslenkung des Lenkrads}), \\ \mathbf{u}^{[c]}(t) &= \mathbf{y}^{[s]}(t), \quad (\text{Aktuatorkraft}). \end{aligned} \right\} \quad (2b)$$

In Abb. 1 werden typische Strategien für die gekoppelte Simulation der mechanischen und hydraulischen Komponenten des mechatronischen Systems veranschaulicht [5]. Für die klassische Modellkopplung exportiert Dymola C-code zur Beschreibung des hydraulischen Teilsystems  $(\cdot)^{[s]}$ , der im Mehrkörpersystem-Simulationswerkzeug SIMPACK als anwenderdefiniertes Kraftelement importiert wird. SIMPACK übernimmt dabei die Rolle des Master-Werkzeugs. Die Zeitintegration des gekoppelten Systems wird vollständig in SIMPACK ausgeführt, wobei der auf dem BDF-code DASSL [6] basierende SIMPACK-Standardintegrator **SODASRT** Verwendung findet.

Alternativ erlaubt Dymola auch den Export von C-code mit eingebettetem Solver [7]. Im Master-Werkzeug SIMPACK wird das Lenkungsmodell mit eingebettetem Solver erneut als anwenderdefiniertes Kraftelement importiert, vgl. Abb. 1. Während der Zeitintegration beschränkt sich der SIMPACK-Integrator **SODASRT** nun jedoch auf das mechanische Teilsystem  $(\cdot)^{[c]}$ , während das hydraulische Kraftelement  $(\cdot)^{[s]}$  mit seinem eigenen

importierten Solver gelöst wird (Solverkopplung). Trotz zahlreicher algorithmischer Probleme [4] ist die Solverkopplung ein aus praktischer Sicht interessantes Lösungskonzept, da man die Zeitschrittweiten  $h_n^{[c]}$  des mechanischen Teils unabhängig von möglichen Stabilitätsbedingten Schranken für die (konstante) Zeitschrittweite  $h^{[s]}$  im hydraulischen Teil wählen kann [4], vgl. auch [8].

Der von Dymola exportierte C-Code ist parametrisiert, so dass Systemparameter des Lenkungsmodells  $(\cdot)^{[s]}$  auch zu einem späteren Zeitpunkt problemlos aktualisiert werden können. Änderungen der Modellstruktur würden hingegen in Dymola einen neuen C-Code-Export erforderlich machen, um das anwenderdefinierte Kraftelement in SIMPACK zu aktualisieren. Diese für Modellkopplung und Solverkopplung typischen stufenweisen Modell-Updates bergen in komplexen Anwendungen die Gefahr inkonsistenter Teilmodelle. Durch Co-Simulation lassen sich diese Probleme vermeiden. Hierzu werden beide Simulationswerkzeuge während der Zeitintegration über eine TCP/IP-Schnittstelle zum Datenaustausch an diskreten Synchronisationszeitpunkten gekoppelt, vgl. Abb. 1.

### 3 Co-Simulation: Grundalgorithmus

Die Co-Simulation von Dymola und SIMPACK wird in *Makroschritten*  $T_j \rightarrow T_{j+1}$  der Länge  $H := T_{j+1} - T_j$  ausgeführt, typische Makroschrittweiten sind in der Größenordnung von  $H = 1.0$  ms, vgl. Abb. 2. An den Synchronisationspunkten  $T_j$ , werden die Systemausgänge  $\mathbf{y}^{[c]}(T_j)$ ,  $\mathbf{y}^{[s]}(T_j)$  zwischen Dymola und SIMPACK ausgetauscht. Approximiert man die physikalischen Kopplungsbedingungen (2b) im Makroschritt  $T_j \rightarrow T_{j+1}$  durch

$$\mathbf{u}^{[s]}(t) := \mathbf{u}^{[s]}(T_j) = \mathbf{y}^{[c]}(T_j), \quad \mathbf{u}^{[c]}(t) := \mathbf{u}^{[c]}(T_j) = \mathbf{y}^{[s]}(T_j), \quad (3)$$

( $t \in [T_j, T_{j+1}]$ ), so ist die Zeitintegration der beiden Teilsysteme “Fahrzeug” (**car**) und “Lenkung” (**steering**) vollständig entkoppelt. Praktisch werden sie unabhängig voneinander mit dem SIMPACK-Standardintegrator **SODASRT** (Teilsystem  $(\cdot)^{[c]}$ ) bzw. mit dem **DASSL**-Integrator in Dymola (Teilsystem  $(\cdot)^{[s]}$ ) gelöst.

Der Synchronisationspunkt  $T_{j+1}$  ist Endpunkt des aktuellen Makroschritts. Vor  $T_{j+1}$  gibt es keine Kommunikation und keinen Datenaustausch zwischen beiden Integratoren. Nach Erreichen von  $T_{j+1}$  wird die Co-Simulation mit dem nachfolgenden Makroschritt  $T_{j+1} \rightarrow T_{j+2}$  fortgesetzt, der bei  $t = T_{j+1}$  mit einem Datenaustausch zwischen beiden Teilsystemen beginnt.

Aus Sicht des Mehrkörpersystem-Simulationswerkzeugs können im Rahmen der Co-Simulation die Eingangsgrößen aus anderen Simulationswerkzeugen wie z. B.  $\mathbf{u}^{[c]}(T_j) = \mathbf{y}^{[s]}(T_j)$  in Abb. 2 als spezielle (zeit-)diskrete Zustandsvariablen betrachtet werden. Die Werte  $\mathbf{u}^{[c]}(T_j)$  der Koppelgrößen werden im Mehrkörpersystem-Simulationswerkzeug für den gesamten Makroschritt  $T_j \rightarrow T_{j+1}$  konstant gehalten. Die Aktualisierung der Koppelgrößen zum Synchronisationszeitpunkt  $t = T_{j+1}$  erfordert im anderen Simulationswerkzeug der Co-Simulationsumgebung (Dymola) die Zeitintegration des Teilsystems  $(\cdot)^{[s]}$  von  $t = T_j$  zum neuen Synchronisationszeitpunkt  $t = T_{j+1}$ , um  $\mathbf{y}^{[s]}(T_{j+1})$  zu berechnen.

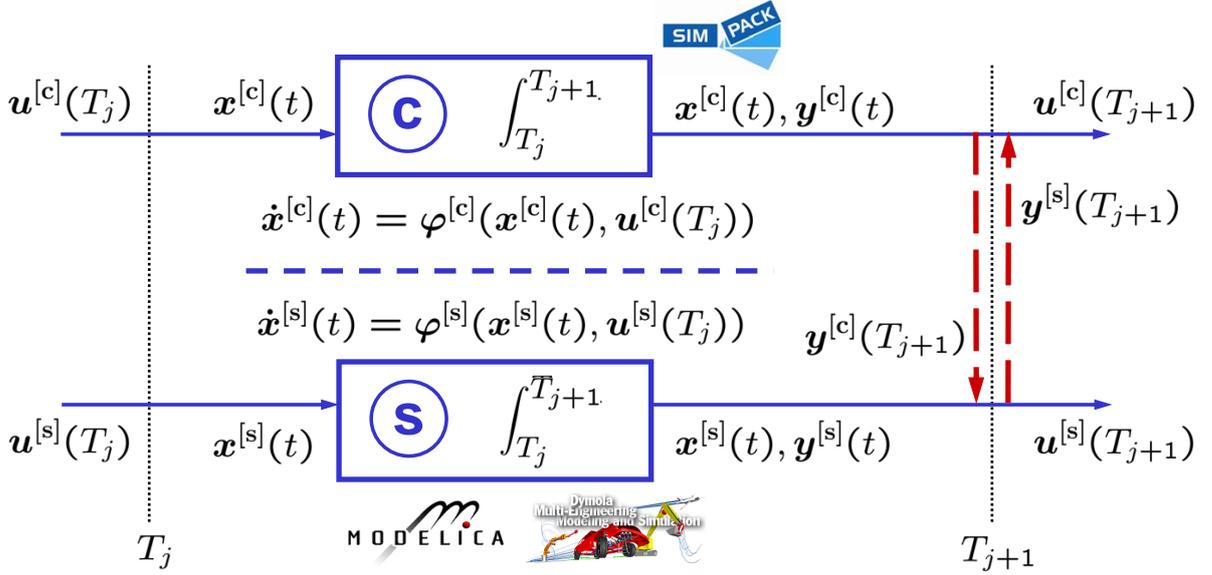


Abbildung 2: Schematische Darstellung eines Makroschritts in der Co-Simulation: Kopplung von SIMPACK für Teilsystem  $(\cdot)^{[c]}$  (car) mit Dymola für Teilsystem  $(\cdot)^{[s]}$  (steering).

## 4 Vergleich mit klassischen Zeitintegrationsverfahren

Für die Analyse von Stabilität und Genauigkeit modularer Zeitintegrationsverfahren erweist sich die abstraktere Darstellung (1) als vorteilhaft. Hierzu ersetzt man in den rechten Seiten von (2a) die Systemeingangsgrößen  $\mathbf{u}^{[c]}$ ,  $\mathbf{u}^{[s]}$  durch  $\mathbf{y}^{[s]}$ ,  $\mathbf{y}^{[c]}$ , vgl. (2b). Die Ausgangsgrößen  $\mathbf{y}^{[s]}$ ,  $\mathbf{y}^{[c]}$  können wegen (2a) als Funktionen von  $\mathbf{x}^{[s]}$ ,  $\mathbf{x}^{[c]}$  dargestellt werden, so dass sich insgesamt ein gekoppeltes System (1) mit  $r = 2$  Teilsystemen ergibt:  $\mathbf{x}^{(1)} = \mathbf{x}^{[s]}$ ,  $\mathbf{x}^{(2)} = \mathbf{x}^{[c]}$ .

Die aus numerischer Sicht entscheidenden Unterschiede zwischen modularen Zeitintegrationsverfahren in Co-Simulationsumgebungen und klassischen Zeitintegrationsverfahren der nichtlinearen Systemdynamik resultieren aus der Beschränkung des Datenaustausches zwischen den Teilsystemen auf die Synchronisationspunkte  $T_j$ . Hierdurch ergeben sich in jedem Makroschritt  $r = 2$  voneinander unabhängige Zeitintegrationsprobleme.

Ein typisches Beispiel ist die in Abschnitt 3 beschriebene Kopplung zweier BDF-Integratoren, wobei sowohl die Verfahrensordnung  $k_n^{(i)}$  als auch die variable (Mikro-) Schrittweite  $h_n^{(i)}$  an die Lösungseigenschaften des jeweiligen Teilsystems angepasst werden können, ( $i = 1, 2$ ):

$$\frac{1}{h_n^{(1)}} \sum_{j=0}^{k_n^{(1)}} \alpha_{n,j}^{(1)} \mathbf{x}_{n+1-j}^{(1)} = \mathbf{f}^{(1)}(\mathbf{x}_{n+1}^{(1)}, \mathbf{x}^{(2)}(T_j), t_{n+1}), \quad (4a)$$

$$\frac{1}{h_n^{(2)}} \sum_{j=0}^{k_n^{(2)}} \alpha_{n,j}^{(2)} \mathbf{x}_{n+1-j}^{(2)} = \mathbf{f}^{(2)}(\mathbf{x}^{(1)}(T_j), \mathbf{x}_{n+1}^{(2)}, t_{n+1}). \quad (4b)$$

Die modulare Zeitintegration bietet insbesondere bei Verwendung unterschiedlicher Zeitschrittweiten  $h_n^{(i)}$  in den Teilsystemen (*Multirate*-Verfahren) ein erhebliches Potenzial für

Effizienzgewinne [1, 8].

Andererseits kann sich die Verwendung konstanter Makroschrittweiten  $H$  ebenso wie das Einfrieren der Koppelgrößen im Makroschritt  $T_j \rightarrow T_{j+1}$ , also die Verwendung von  $\mathbf{x}^{(i)}(T_j)$  statt der “korrekten” Werte  $\mathbf{x}_{n+1}^{(i)}$  in den rechten Seiten von (4), sowohl auf die Genauigkeit als auch auf die Stabilität des modularen Zeitintegrationsverfahrens negativ auswirken [9, 10]. Verbesserte modulare Zeitintegrationsverfahren setzen deshalb auf

- variable, problemangepasste Makroschrittweiten  $H_j$  (Makroschrittweitensteuerung),
- lineare, quadratische oder kubische Approximationen der Koppelgrößen und
- implizite oder linear-implizite Ansätze zur Vermeidung numerischer Instabilität.

Hierzu ist eine Erweiterung der Co-Simulationsschnittstellen erforderlich, um Fehlerschätzer, Zeitableitungen von Koppelgrößen und Jacobimatrizen der in den rechten Seiten  $\mathbf{f}^{(i)}$  enthaltenen Koppelterme zwischen den Simulationswerkzeugen austauschen zu können.

## Literatur

- [1] K. Jackson. A survey of parallel numerical methods for initial value problems for ordinary differential equations. *IEEE Transactions on Magnetics*, 27:3792–3797, 1991.
- [2] M. Hoschek, P. Rentrop, Y. Wagner. Network approach and differential-algebraic systems in technical applications. *Surveys on Math. in Industry*, 9:49–75, 1999.
- [3] M. Arnold und W. Schiehlen, Hrsg.. *Simulation Techniques for Applied Dynamics*, volume 507 of *CISM Courses and Lectures*. Springer, Wien New York, 2008.
- [4] M. Busch. Entwicklung einer SIMPACK–Modelica/Dymola–Schnittstelle. Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, 2007.
- [5] M. Busch, M. Arnold, A. Heckmann, S. Dronka. Interfacing SIMPACK to Modelica / Dymola for multi-domain vehicle system simulations. *SIMPACK News (INTEC GmbH, Wessling, Germany)*, 11(2):01–03, 2007.
- [6] K.E. Brenan, S.L. Campbell, L.R. Petzold. *Numerical solution of initial–value problems in differential–algebraic equations*. SIAM, Philadelphia, 2nd edition, 1996.
- [7] H. Elmqvist, M. Otter, F.E. Cellier. Inline integration: A new mixed symbolic / numeric approach for solving differential–algebraic equation systems. In *Proc. ESM’95*, pages xxiii–xxxiv, 1995.
- [8] G. Hippmann, M. Arnold, M. Schittenhelm. Efficient simulation of bush and roller chain drives. In J.M. Goicolea, J. Cuadrado, J.C. García Orden, editors, *Proc. of Multibody Dynamics 2005*, Madrid, Spain, 2005.
- [9] R. Kübler. *Modulare Modellierung und Simulation mechatronischer Systeme*. Fortschritt-Berichte VDI Reihe 20, Nr. 327. VDI-Verlag GmbH, Düsseldorf, 2000.
- [10] M. Arnold, M. Günther. Preconditioned dynamic iteration for coupled differential-algebraic systems. *BIT Numerical Mathematics*, 41:1–25, 2001.