

T. Schierz, M. Arnold

MODELISAR: Innovative numerische Methoden bei der Kopplung von multidisziplinären Simulationsprogrammen. - In: A. Brenke, editor, *Tagungsband ASIM-Konferenz STS/GMMS 2011, Krefeld, 24.02.-25.02.11.* – Shaker Aachen, 2011.

Zusammenfassung. Die Modellierung technischer Systeme führt häufig zu gekoppelten Differentialgleichungssystemen. Dabei ist in vielen Fällen eine Zerlegung in einzelne Teilsysteme vorteilhaft, welche dann mit unterschiedlichen Zeitintegrationsverfahren in spezialisierten Simulationswerkzeugen unabhängig voneinander gelöst werden (Co-Simulation oder Simulatorkopplung). Jedoch existieren heutzutage oftmals nur spezialisierte Interfaces zwischen zwei Simulationswerkzeugen, was eine allgemeine Anwendbarkeit auf beliebige Systeme einschränkt. Zudem erfolgt der Austausch von Koppelgrößen überwiegend zu fest vorgegebenen Kommunikationszeitpunkten. Das MODELISAR-Projekt hat das Ziel, ein standardisiertes Interface zur Kopplung multidisziplinärer Simulationswerkzeuge zu schaffen. Untersucht werden auch Weiterentwicklungen, wie z. B. eine adaptive Wahl der Kommunikationsschritte. Diesen Ansatz stellen wir im Vortrag vor und gehen insbesondere näher auf die mathematischen Hintergründe und die praktische Umsetzung ein.

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

MODELISAR: Innovative numerische Methoden bei der Kopplung von multidisziplinären Simulationsprogrammen

Tom Schierz, Martin Arnold

Martin-Luther-Universität Halle-Wittenberg

NWF II – Institut für Mathematik

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

Tel.: 0345/5524654, Fax: 0345/5527004

Zusammenfassung

Die Modellierung technischer Systeme führt häufig zu gekoppelten Differentialgleichungssystemen. Dabei ist in vielen Fällen eine Zerlegung in einzelne Teilsysteme vorteilhaft, welche dann mit unterschiedlichen Zeitintegrationsverfahren in spezialisierten Simulationswerkzeugen unabhängig voneinander gelöst werden (*Co-Simulation* oder *Simulatorkopplung*). Jedoch existieren heutzutage oftmals nur spezialisierte Interfaces zwischen zwei Simulationswerkzeugen, was eine allgemeine Anwendbarkeit auf beliebige Systeme einschränkt. Zudem erfolgt der Austausch von Koppelgrößen überwiegend zu fest vorgegebenen *Kommunikationszeitpunkten*. Das MODELISAR-Projekt hat das Ziel, ein standardisiertes Interface zur Kopplung multidisziplinärer Simulationswerkzeuge zu schaffen. Untersucht werden auch Weiterentwicklungen, wie z.B. eine adaptive Wahl der Kommunikationsschritte. Diesen Ansatz stellen wir im Vortrag vor und gehen insbesondere näher auf die mathematischen Hintergründe und die praktische Umsetzung ein.

1 Einleitung

In praktischen Anwendungen hat die Kopplung von etablierten mono-disziplinären Simulationswerkzeugen zum Modellaufbau des gekoppelten Systems und zur numerischen Lösung der gekoppelten Modellgleichungen 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 jedes einzelnen Teilsystems ein eigenes, an die spezielle Struktur dieser Modellgleichungen angepasstes Zeitintegrationsverfahren (*modulare* Zeitintegration gekoppelter Systeme [1]).

Üblicherweise benötigt die Zeitintegration in einer Co-Simulationsanwendung sehr viel mehr Prozessorleistung als die Zeitintegration von hochentwickelten mono-disziplinären Solvern in bestehenden kommerziellen Simulationsprogrammen. Studien in High-End Anwendungen [2, 3] zeigen jedoch, dass Co-Simulation durchaus vom numerischen Standpunkt aus vorteilhaft sein kann, wenn das System stark unterschiedliche Zeitskalen besitzt (Multirate Zeitintegration mit schwacher Kopplung [4]).

Vom mathematischen Standpunkt aus erfolgt bei der Co-Simulation eine numerische Zeitintegration von gekoppelten Systemen, die durch zeitabhängige gewöhnliche Differentialgleichungen (ODEs) oder differential algebraische Gleichungen (DAEs) beschrieben werden. Dabei können unterschiedliche speziell auf die Teilsysteme zugeschnittene Zeitintegrationsverfahren (z. B. explizite und implizite Integratoren) und / oder unterschiedliche Schrittweiten verwendet werden.

Wir betrachten $r \geq 2$ gekoppelte Teilsysteme in nichtlinearer Zustandsform

$$\left. \begin{aligned} \dot{x}_i &= f_i(t, x_i, u_i), \\ y_i &= g_i(t, x_i, u_i), \\ u_i &= \sum_{j=1}^r L_{ij} y_j \end{aligned} \right\} \quad i = 1, \dots, r, \quad t \in [t_{\text{start}}, t_{\text{stop}}] \quad (1)$$

mit Zuständen $x_i \in \mathbb{R}^{n_{x_i}}$, Eingängen $u_i \in \mathbb{R}^{n_{u_i}}$, Ausgängen $y_i \in \mathbb{R}^{n_{y_i}}$, rechter Seite

$$f_i : \mathbb{R} \times \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} \rightarrow \mathbb{R}^{n_{x_i}},$$

und der Ausgangsfunktion

$$g_i : \mathbb{R} \times \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} \rightarrow \mathbb{R}^{n_{y_i}},$$

zugehörigen Anfangswerten $x_{i,0} := x_i(t_{\text{start}})$, $y_{i,0} := y_i(t_{\text{start}})$ und Dimensionen n_{x_i} , n_{y_i} , $n_{u_i} \in \mathbb{N}$. Die Teilsysteme sind über die Eingänge u_i und Ausgänge y_i miteinander gekoppelt. Es gelten die allgemeinen Koppelbedingungen $u_i = \sum_{j=1}^r L_{ij} y_j$ mit Inzidenzmatrizen $L_{ij} \in \mathbb{R}^{n_{u_i} \times n_{y_j}}$, ($i \neq j$), $L_{ii} = 0$ [5]. Zur Vereinfachung wählen wir für $i, j = 1, \dots, r$

$$\begin{aligned} L_{ij} &:= I_{n_{y_i}}, & (i \neq j), \\ L_{ii} &:= 0. \end{aligned}$$

Nach Jackson [1], bezeichnet man die Zeitintegrationsverfahren zur Lösung der $r \geq 2$ Teilsysteme (1) als *modular*, um die modulare Struktur dieses Ansatzes zu unterstreichen. Das Simulationszeitintervall wird durch ein diskretes Gitter von *Kommunikationszeitpunkten* $t_{\text{start}} = T_0 < T_1 < \dots < T_N = t_{\text{stop}}$ zerlegt, wobei ein Datenaustausch zwischen den Teilsystemen auf diese Zeitpunkte beschränkt bleibt, vgl. Abbildung 1. In jedem *Kommunikationsschritt* $T_n \rightarrow T_n + H_n =: T_{n+1}$, ($n = 0, \dots, N - 1$), mit *Kommunikationsschrittweite* H_n , werden die Teilsysteme unabhängig voneinander gelöst.

Da ein Update der Systemeingänge u_i auf die diskreten Kommunikationszeitpunkte

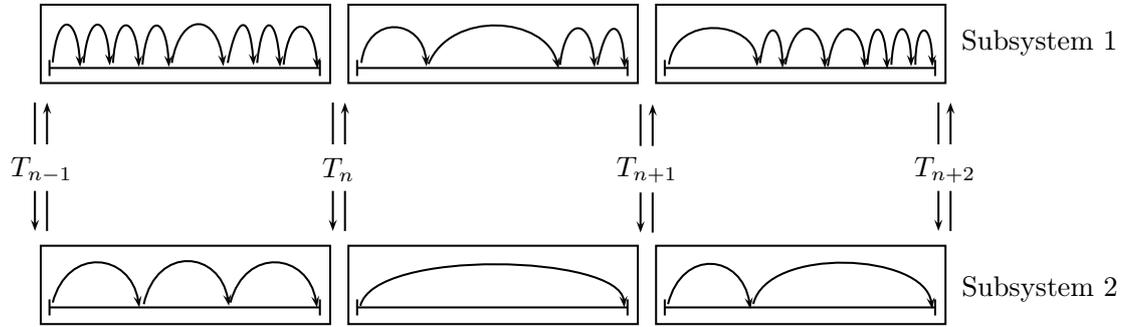


Abbildung 1: Co-Simulation von zwei gekoppelten Teilsystemen mit diskreten Kommunikationszeitpunkten.

beschränkt ist, müssen diese Terme durch Extrapolation im aktuellen Kommunikationsschritt $T_n \rightarrow T_{n+1}$ approximiert werden

$$\bar{u}_i(t) \approx u_i(t), \quad t \in [T_n, T_{n+1}], \quad (i = 1, \dots, r).$$

Die Approximation $\bar{u}_i(t)$ mit Ordnung q basiert auf Informationen $y_j(t)$, für $t \leq T_n$, $j = 1, \dots, r$ und möglicherweise auch $y_j(T_n + c_k H_n)$, ($k = 0, \dots, K$; $j = 1, \dots, r$) mit Gewichten c_k . Dies führt zu dem neuen gekoppelten System

$$\left. \begin{aligned} \dot{\bar{x}}_i(t) &= f_i(t, \bar{x}_i, \bar{u}_i), \\ \dot{\bar{y}}_i(t) &= g_i(t, \bar{x}_i, \bar{u}_i) \end{aligned} \right\} \bar{x}_i(T_n) = x_i(T_n), \quad t \in [T_n, T_n + H_n], \quad i = 1, \dots, r. \quad (2)$$

Die Teilsysteme in (2) werden nun unabhängig voneinander gelöst, wobei wir eine numerische Lösung

$$\bar{x}_i(t) \approx x_i(t), \quad \bar{y}_i(t) \approx y_i(t), \quad t \in [T_n, T_n + H_n], \quad i = 1, \dots, r$$

erhalten.

2 Variable Kommunikationsschrittweite

In vielen industriellen Co-Simulationsanwendungen hat sich bisher eine Wahl von äquidistant verteilten Kommunikationszeitpunkten mit fester Kommunikationsschrittweite bewährt, da nur so die Wechselwirkungen mit den internen Schrittweitensteuerungsalgorithmen der Zeitintegrationsverfahren der einzelnen Teilsysteme zuverlässig beherrscht werden. In einigen Anwendungen kommen auch vordefinierte Zeitgitter zum Einsatz. Diese sind an das entsprechende Systemverhalten angepasst, und die Wahl der Schrittweite erfolgt durch praktische Erfahrungswerte im Vorfeld der Simulation.

Dieser eher unflexibler Ansatz stellt einen der größten Nachteile der Co-Simulation dar, denn nahezu jedes klassische Zeitintegrationsverfahren (z. B. DASSL [6] oder der MATLAB-Integrator `ode15s` [7]) besitzt eine Schrittweitensteuerung, um numerische Probleme so effizient und genau wie möglich zu lösen. Die Implementierung in dem ex-

pliziten Runge-Kutta-Verfahren DOPRI5 (Download: <http://www.unige.ch/~hairer/prog/nonstiff/dopri5.f>) kann als fortgeschrittene Referenzimplementierung für die klassische Zeitintegration von gewöhnlichen Differentialgleichungen angesehen werden, vgl. [8].

Die Effizienz eines Integrators hängt stark von der verwendeten Schrittweite in jedem Integrationsschritt ab. So erlauben Zeitintervalle, in denen sich die Lösung stark ändert, nur sehr kleine Schritte, damit der sogenannte lokale Diskretisierungsfehler der numerischen Lösung unter einer vorgegebenen Toleranz bleibt. Jedoch führen zu kleine Schrittweiten zu einer ineffizienten Integration (Rundungsfehler, viele Integrationsschritte). Im Gegensatz dazu kann der Integrator in Zeitintervallen, in denen sich die Lösung nur langsam ändert, entsprechend größere Schritte ausführen, womit Rechenaufwand eingespart wird. Im Allgemeinen wählt man die Schrittweite so klein wie nötig, um die Genauigkeitsforderung einzuhalten, und so groß wie möglich, um Rechenzeit zu sparen. Das Optimierungsziel ist es, eine Ausgewogenheit zwischen Effizienz und Genauigkeit zu erreichen [8, 9].

Die Steuerung des Verfahrensfehlers und die adaptive, automatische Wahl von Kommunikationsschrittweiten können demzufolge zu zuverlässigeren und sehr viel effizienteren Master-Algorithmen in der Co-Simulation beitragen. Letztendlich hätten Co-Simulationsanwendungen die eine Kommunikationsschrittweitensteuerung unterstützen, einen technologischen Vorsprung im Vergleich zu Konkurrenzprodukten, denn die Forschung in diesem Bereich ist weit weniger fortgeschritten als z. B. die Weiterentwicklung klassischer Integratoren.

Die Vorteile einer Kommunikationsschrittweitensteuerung liegen auf der Hand. Erkennt der Algorithmus zur Regelung der Schrittweiten eine sich nur langsam ändernde Lösung, erlaubt das Systemverhalten folglich größere Kommunikationsschrittweiten unter Einhaltung der Genauigkeitsforderungen, was problemabhängig einen beträchtlichen Effizienz- und Geschwindigkeitsvorteil zur Folge haben kann. Dies ist vor allem bei Testproblemen mit hohem Rechenaufwand der Fall (z. B. Mehrkörpersysteme mit mehreren Tausend Freiheitsgraden). Desweiteren kann durch entsprechende Fehlerschätzer auch auf Zeitintervalle reagiert werden, in denen eine hohe Systemdynamik infolge der Genauigkeitsforderung extrem kleine Integrationsschritte erfordert.

Die Bestimmung der Schrittweiten birgt auch hier einige Risiken und Schwierigkeiten. Wählt man den Kommunikationsschritt zu groß, ergibt sich möglicherweise ein unvorhersehbares Verhalten der internen Schrittweitensteuerung der Integratoren. So könnte z. B. die *Mikroschrittweite* (Schrittweite der Integratoren der einzelnen Teilsysteme) infolge des internen Fehlerschätzers zur Einhaltung der geforderten Genauigkeit stark verkleinert werden. Dies führt zu einem Effizienzverlust oder sogar zum Abbruch der Simulation. Demgegenüber ist bei der Wahl einer zu kleinen Kommunikationsschrittweite der Effizienzverlust offensichtlich. Eine effektive Kommunikationsschrittweitensteuerung mit zugehörigen Fehlerschätzern und Algorithmen zur Wahl der optimalen Schrittweite sollte diese Probleme zuverlässig behandeln können.

Bei der adaptiven Wahl der Kommunikationsschrittweite in Co-Simulationsanwendungen ergeben sich jedoch einige algorithmische Probleme. So erhöht sich der Aufwand durch zusätzliche Berechnungsschritte des Fehlerschätzers und der Berechnung der optimalen Schrittweite. Zudem müsste man für eine Adaption der klassischen Schrittweitensteuerungsalgorithmen auch verworfene Kommunikationsschritte infolge der Nichteinhaltung der Genauigkeitsforderung in Betracht ziehen. Dabei gilt zu beachten, dass heutige Co-Simulationsanwendungen keinen Neustart bzw. Neuinitialisierung des Kommunikationsschrittes mit geänderter Schrittweite unterstützen, denn die strikt sequentielle Bearbeitung der Kommunikationsschritte macht eine Rückkehr zu zurückliegenden Zeitpunkten unmöglich. Diese Probleme und die Tatsache, dass für die Berechnung der Kommunikationsschritte mehr Informationen zur Verfügung gestellt werden müssen, wird in der Entwicklung des MODELISAR FMI for Co-Simulation Interface behandelt.

Bei der Entwicklung einer Kommunikationsschrittweitensteuerung sind effektive Algorithmen zur problemangepassten Wahl der Schrittweite zwingend erforderlich. Diese basiert auf dem Vergleich einer Schätzung des lokalen Diskretisierungsfehlers des aktuellen Zeitschrittes mit einer vom Nutzer vorgegebenen Fehlertoleranz. Hier können wir auf diverse Ansätze zur Verbesserung der Schrittweitensteuerung klassischer Integrationsverfahren zurückgreifen. Ein möglicher Ansatz besteht in der Verwendung von Methoden der Regelungstheorie zur Untersuchung der Wechselwirkungen zwischen den externen Algorithmen der Kommunikationsschrittweitensteuerung und den internen Ordnungs- und (Mikro-) Schrittweitensteuerungsalgorithmen der Zeitintegrationsverfahren der einzelnen Teilsysteme. Eine beispielhafte Anwendung dieser Strategie kann in [10] durch die regelungstechnische Interpretation der klassischen Schrittweitensteuerung nachgelesen werden. Die Standard-Schrittweitensteuerung mit Fehlerschätzer (vgl. z. B. [8] oder [11]) wird als ein I-Regler aufgefasst und entsprechend der Regelungstheorie zu einem PI-Regler weiterentwickelt, der im Allgemeinen die Vorteile des P- und I-Reglers vereint und mit glatterem Schrittweitenverlauf auf nötige Änderungen reagiert.

Das größte Problem bei der Entwicklung einer Kommunikationsschrittweitensteuerung ist es, einen geeigneten Fehlerschätzer zu konstruieren, wobei dieser möglichst einfach gehalten werden sollte [12]. Die klassischen Fehlerschätzer erweisen sich in der Regel als zu konservativ, da sie auch Fehleranteile berücksichtigen, die in aktiv oder passiv geregelten technischen Systemen konstruktionsbedingt keine praktische Rolle spielen und somit vernachlässigt werden können. In diesem Fall sind *gefilterte Fehlerschätzer* vorstellbar, die nur diejenigen Anteile der lokalen Diskretisierungsfehler berücksichtigen, die tatsächlich zum globalen Diskretisierungsfehler beitragen [13, 14], vgl. auch [15].

3 Numerische Ergebnisse

Schrittweitensteuerung in Co-Simulationsanwendungen ist heutzutage in der Praxis noch nicht der Stand der Technik. Es gibt einige Probleme, die in diesem Zusammenhang unerforscht sind, wie zum Beispiel die gegenseitige Beeinflussung der Kommunikationsschritt-

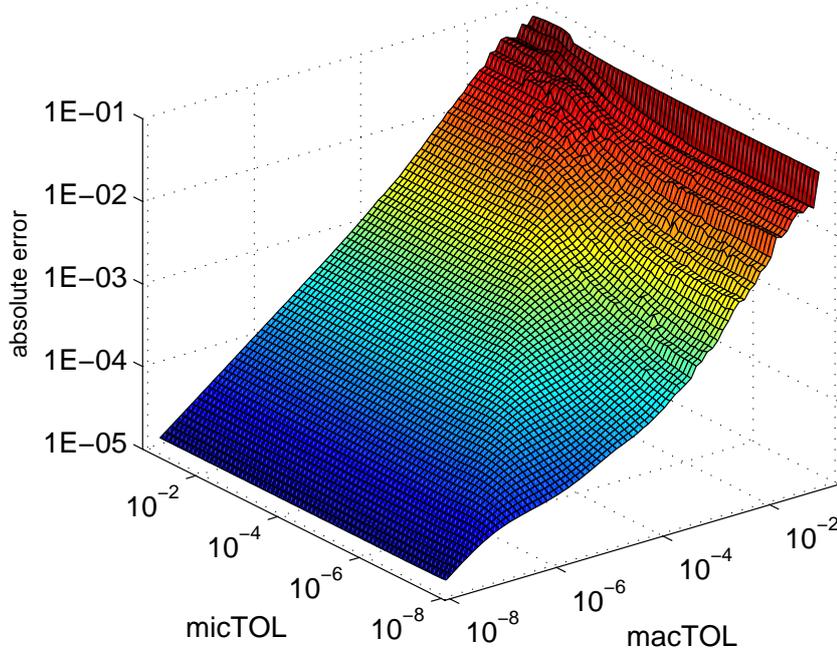


Abbildung 2: Absoluter Fehler der Co-Simulation mit Schrittweitensteuerung bei unterschiedlichen Genauigkeitstoleranzen.

weitensteuerung und der internen Mikroschrittweitensteuerung der Teilsysteme. Es gilt auch daraus resultierende Zusammenhänge in Abhängigkeit von unterschiedlichen Ordnungen der Integratoren und Extrapolation zu untersuchen. In den Abbildungen 2 und 3 kann man beispielhaft die Untersuchung und Visualisierung dieser Zusammenhänge unter unterschiedlichen Genauigkeitsforderungen betrachten. Dabei betrachten wir die Genauigkeit und Effizienz der Co-Simulationsergebnisse an einem einfachen Benchmark-Problem.

Zusammen mit zusätzlichen theoretischen Untersuchungen und Weiterentwicklung bestehender Algorithmen zur Kommunikationsschrittweitensteuerung können Effizienz und Genauigkeit zusätzlich optimiert werden. Das Ziel ist die Steuerung der Kommunikationsschritte in der Co-Simulation so zu verbessern, dass diese genauso robust und leistungsfähig Verwendung findet, wie es bei der klassischen Schrittweitensteuerung etablierter mono-disziplinärer Simulationsanwendungen der Fall ist.

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] S. Dietz. The new powerful linear subsystem solver for flexible bodies in multi-body systems. In J.M. Goicolea, J. Cuadrado, and J.C. García Orden, editors, *Proc. of Multibody Dynamics 2005 (ECCOMAS Thematic Conference)*, Madrid, Spain, 2005.
- [3] G. Hippmann, M. Arnold, and M. Schittenhelm. Efficient simulation of bush and roller chain drives. In J.M. Goicolea, J. Cuadrado, and J.C. García Orden, edi-

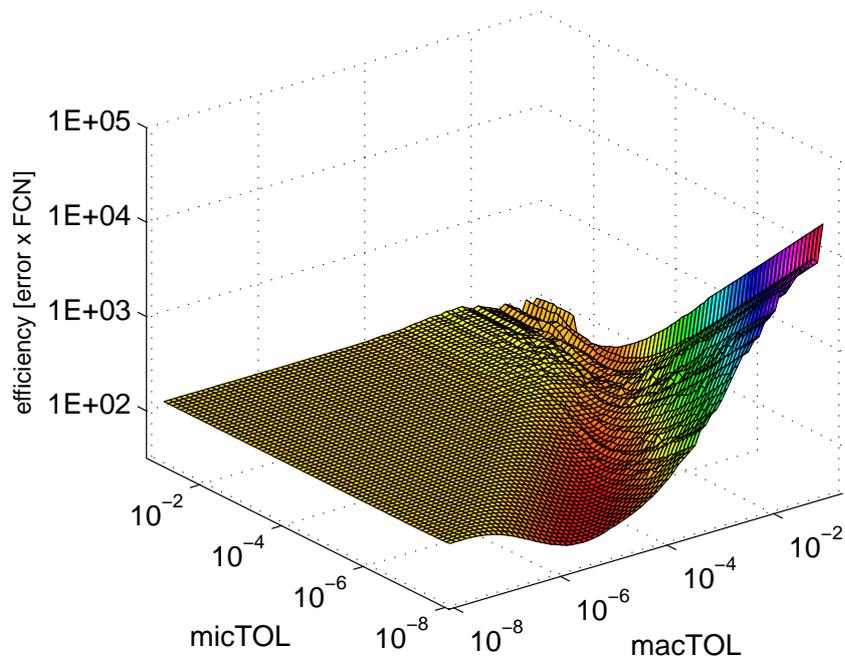


Abbildung 3: Indikator für die Effizienz der Co-Simulation mit Schrittweitensteuerung (Genauigkeit \times Aufwand).

tors, *Proc. of Multibody Dynamics 2005 (ECCOMAS Thematic Conference)*, Madrid, Spain, 2005.

- [4] M. Arnold. Multi-rate time integration for large scale multibody system models. In P. Eberhard, editor, *IUTAM Symposium on Multiscale Problems in Multibody System Contacts*, pages 1–10. Springer, 2007.
- [5] R. Kübler and W. Schiehlen. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, 6:93–113, 2000.
- [6] L. R. Petzold. A description of DASSL: a differential/algebraic system solver. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans. Sci. Comput., I, pages 65–68. IMACS, New Brunswick, NJ, 1983.
- [7] L.F. Shampine and M.W. Reichelt. The MATLAB ODE Suite. *SIAM J. Sci. Comp.*, 18:1–22, 1997.
- [8] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I. Nonstiff Problems*. Springer-Verlag, Berlin Heidelberg New York, 2nd edition, 1993.
- [9] U. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1998.
- [10] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28:270–287, 1988.
- [11] K. Strehmel and R. Weiner. *Numerik gewöhnlicher Differentialgleichungen*. Teubner Studienbücherei Mathematik. Teubner, Stuttgart, 1995.

- [12] L. F. Shampine and A. Witt. A simple step size selection algorithm for ODE codes. *J. Comput. Appl. Math.*, 58(3):345–354, 1995.
- [13] L. F. Shampine and L. S. Baca. Error estimators for stiff differential equations. *J. Comput. Appl. Math.*, 11(2):197–207, 1984.
- [14] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*. Springer–Verlag, Berlin Heidelberg New York, 2nd edition, 1996.
- [15] W. Rulka. *Effiziente Simulation der Dynamik mechatronischer Systeme für industrielle Anwendungen*. PhD thesis, Vienna University of Technology, Department of Mechanical Engineering, 1998.