

Peter Junglas

Wozu braucht der Roboter Lineare Algebra?

Auszug. Anhand zweier einfacher Beispiele wird gezeigt, wie sich die Bewegungsgleichungen komplexer dreidimensionaler Systeme in systematischer Weise aus den Gleichungen der benutzten Bausteine und einfachen Verknüpfungsregeln gewinnen lassen. Dafür sind allerdings gute Rechenfertigkeiten aus der Linearen Algebra essenziell, die sich hier – weit entfernt von ihrem Image einer “abstrakten Theorie” – als hochgradig praktisch und nützlich erweist.

Mehrkörper-Simulation

Das Aufstellen der Bewegungsgleichungen für mechanische Systeme ist für die Studierenden schon bei einfachen Problemen oft ein schwieriges und fehleranfälliges Unterfangen. Betrachtet man dreidimensionale Modelle, werden zusätzlich noch gute Kenntnisse der Linearen Algebra benötigt. Solche Untersuchungen bleiben daher – von einfachen Beispielen abgesehen – in der Regel Spezialvorlesungen zur Mechanik in höheren Semestern vorbehalten. Dies gilt erst recht bei den komplexen Systemen der Mehrkörperdynamik [6], die sich mit der Kinematik und Dynamik gekoppelter starrer Körper befasst, vom einfachen Schubkurbelgetriebe bis zum komplexen Sechssachsen-Roboter.

Mit der Entwicklung von objektorientierten Bibliotheken zur Mehrkörper-Simulation [5] wurde das Problem der Modellierungs-Software übertragen (z. B. [4]). Sie stellt universelle Bausteine zur Verfügung, die den Aufbau und die Simulation komplexer Systeme mit wenigen Grundkenntnissen ermöglichen. Die Bewegungsgleichungen werden dabei nach dem üblichen Modelica-Schema automatisch aus den Gleichungen der Bausteine und der Verknüpfungen abgeleitet [2]. Für die anschließende numerische Simulation sind allerdings noch einige, ebenfalls automatisch erfolgende Vor-Bearbeitungen nötig [3, 1].

Im Folgenden soll das grundlegende Verfahren an zwei einfachen Beispielen (dem physikalischen Pendel und der Verladebrücke) per Hand nachvollzogen werden. Die Studierenden lernen daran nicht nur, wie die Software grundsätzlich funktioniert, sondern können diese Methode anwenden, um selbst die Bewegungsgleichungen einfacher Systeme aufzustellen. Darüberhinaus lernen sie dabei eine praktische Anwendung der Linearen Algebra kennen, die sonst eher als besonders abstrakt verschrien ist.

Benötigte Vorkenntnisse

Zum Verständnis des Verfahrens und der konkreten Rechnungen werden aus der Mechanik Grundlagen der Dynamik und die Eulergleichung vorausgesetzt, aus der Linearen Algebra die Beherrschung der Vektor- und Matrixrechnung im \mathbb{R}^3 und der Umgang mit Drehmatrizen. Darüberhinaus müssen einige spezielle Beziehungen vorgestellt – und bei ausreichender Zeit auch abgeleitet – werden:

- eine Schreibweise des Kreuzprodukts als Matrix (alternativ zum Levi-Civita-Tensor ε)

$$\tilde{\mathbf{a}} := \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \Rightarrow \mathbf{a} \times \mathbf{b} = \tilde{\mathbf{a}} \cdot \mathbf{b}$$

- die Berechnung der Drehmatrix \mathbf{R} bei gegebener Achse \mathbf{n} und Drehwinkel φ

$$\mathbf{R} = \mathbf{n} \cdot \mathbf{n}' + (\mathbf{1} - \mathbf{n} \cdot \mathbf{n}') \cos \varphi - \tilde{\mathbf{n}} \sin \varphi$$

- die Berechnung der Winkelgeschwindigkeit $\boldsymbol{\omega}$ bei zeitabhängiger Drehmatrix $\mathbf{R}(t)$

$$\tilde{\boldsymbol{\omega}} = \mathbf{R} \cdot \dot{\mathbf{R}}'$$

Modelicas MultiBody-Bibliothek

Bei der objektorientierten Modellierung mit Modelica werden Bausteine graphisch miteinander verbunden. Zu diesem Zweck haben sie Anschlusspunkte (“Connectoren”), die die Größen definieren, die miteinander verknüpft werden. In der MultiBody-Bibliothek beschreibt ein Connector ein lokales Koordinatensystem (**frame**) relativ zu einem global definierten Inertialsystem **world**, indem er folgende Größen festlegt: den Verschiebungsvektor \mathbf{r} von **world** zu **frame** (bezogen auf **world**), die Drehmatrix \mathbf{R} , die **world** in **frame** dreht, die am Verbindungspunkt wirkende Schnittkraft \mathbf{F} und das Schnittmoment \mathbf{M} (beide bezogen auf **frame**). Verbindet man mehrere Anschlusspunkte miteinander, werden dadurch Beziehungen zwischen diesen Variablen generiert: Die “Potentialgrößen” \mathbf{r} und \mathbf{R} haben alle denselben Wert, die “Flowgrößen” \mathbf{F} , \mathbf{M} addieren sich zu Null.

Die einzelnen Bausteine definieren weitere Gleichungen zwischen den Anschlussgrößen und ggf. intern definierten zusätzlichen Variablen. Dazu können sie die Werte von Parametern verwenden, die vom Anwender festgelegt

werden. Für die beiden Beispielm Modelle benötigt man fünf verschiedene Komponenten (Abb. 1) und ihre zugehörigen Gleichungen, die sich jeweils direkt aus der Funktion eines Bausteins und grundlegenden Beziehungen der Mechanik ergeben:

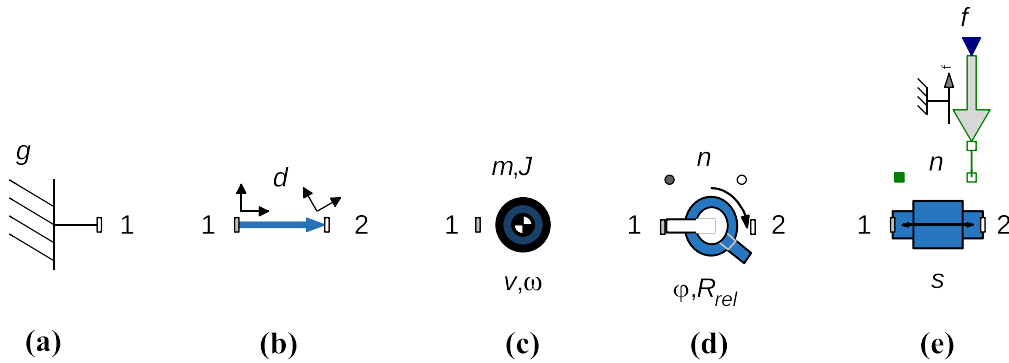


Abbildung 1: Bausteine der MultiBody-Bibliothek

- a.) **World** liefert an seinem Connector das globale Inertialsystem und enthält den Parameter \mathbf{g} für die Schwerebeschleunigung.

$$\begin{aligned}\mathbf{r}_1 &= \mathbf{0} \\ \mathbf{R}_1 &= \mathbf{1}\end{aligned}$$

- b.) **FixedTranslation** beschreibt zwei Frames, die sich nur um eine feste Verschiebung \mathbf{d} unterscheiden.

$$\begin{aligned}\mathbf{r}_2 &= \mathbf{r}_1 + \mathbf{R}'_1 \cdot \mathbf{d} \\ \mathbf{R}_2 &= \mathbf{R}_1 \\ \mathbf{F}_2 &= -\mathbf{F}_1 \\ \mathbf{M}_2 &= -\mathbf{M}_1 + \mathbf{d} \times \mathbf{F}_1\end{aligned}$$

- c.) **Body** definiert einen starren Körper mit Masse m und Trägheitsmoment \mathbf{J} . An internen Variablen enthält er die Geschwindigkeit \mathbf{v} und die Winkelgeschwindigkeit $\boldsymbol{\omega}$.

$$\begin{aligned}\mathbf{v} &= \dot{\mathbf{r}}_1 \\ \tilde{\boldsymbol{\omega}} &= \mathbf{R}_1 \cdot \dot{\mathbf{R}}'_1 \\ \mathbf{F}_1 &= \mathbf{R}_1 \cdot m(\dot{\mathbf{v}} - \mathbf{g}) \\ \mathbf{M}_1 &= \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})\end{aligned}$$

d.) **Revolute** ist ein Gelenk, das eine Drehung um eine feste Drehachse \mathbf{n} erlaubt. Weitere Variablen sind der Drehwinkel φ und die Drehmatrix \mathbf{R}_{rel} zwischen beiden Frames.

$$\begin{aligned}\mathbf{R}_{rel} &= \mathbf{n} \cdot \mathbf{n}' + (\mathbf{1} - \mathbf{n} \cdot \mathbf{n}') \cos \varphi - \tilde{\mathbf{n}} \sin \varphi \\ \mathbf{r}_2 &= \mathbf{r}_1 \\ \mathbf{R}_2 &= \mathbf{R}_{rel} \cdot \mathbf{R}_1 \\ \mathbf{F}_2 &= -\mathbf{R}_{rel} \cdot \mathbf{F}_1 \\ \mathbf{M}_2 &= -\mathbf{R}_{rel} \cdot \mathbf{M}_1 \\ 0 &= \mathbf{M}_2 \cdot \mathbf{n}\end{aligned}$$

e.) **Prismatic** ermöglicht eine lineare Verschiebung der beiden Frames in fester Richtung \mathbf{n} , die interne Größe s beschreibt den Betrag der Verschiebung. Zusätzlich kann durch einen weiteren Anschluss eine äußere Kraft f längs der Achse eingebracht werden.

$$\begin{aligned}\mathbf{r}_2 &= \mathbf{r}_1 + s \mathbf{R}'_1 \cdot \mathbf{n} \\ \mathbf{R}_2 &= \mathbf{R}_1 \\ \mathbf{F}_2 &= -\mathbf{F}_1 \\ \mathbf{M}_2 &= -\mathbf{M}_1 + s \mathbf{n} \times \mathbf{F}_1 \\ f &= -\mathbf{F}_2 \cdot \mathbf{n}\end{aligned}$$

Beispiel Pendel

Für das physikalische Pendel (Abb. 2) lässt sich mit den beschriebenen Komponenten leicht ein Modelica-Modell erstellen (Abb. 3). Die Achsen

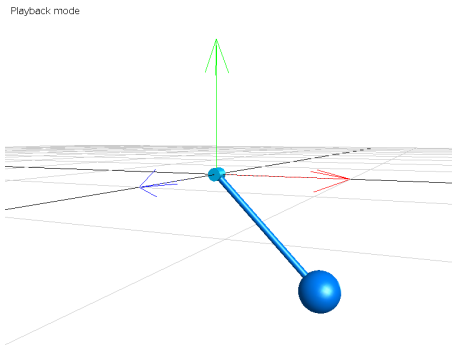


Abbildung 2: Beispiel Pendel

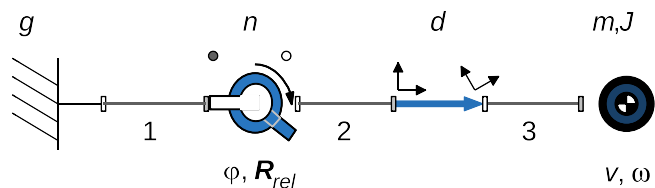


Abbildung 3: Modell Pendel

und sonstigen Parameter werden folgendermaßen festgelegt:

$$\mathbf{g} = \begin{pmatrix} 0 \\ -g \\ 0 \end{pmatrix}, \quad \mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{J} = J \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Die Verbindungen zwischen den Bausteinen reduzieren die Zahl der Frames auf drei, die zugehörigen Schnittkräfte und -momente werden jeweils nach rechts genommen. Durch direktes Zusammenstellen aller Bauteilgleichungen erhält man für das Modell damit das komplette System:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{0} & (1) & & \mathbf{r}_3 &= \mathbf{r}_2 + \mathbf{R}'_2 \cdot \mathbf{d} & (9) \\ \mathbf{R}_1 &= \mathbf{1} & (2) & & \mathbf{R}_3 &= \mathbf{R}_2 & (10) \\ \mathbf{R}_{rel} &= (\mathbf{1} - \mathbf{n} \cdot \mathbf{n}') \cos \varphi & & & \mathbf{F}_3 &= \mathbf{F}_2 & (11) \\ &+ \mathbf{n} \cdot \mathbf{n}' - \tilde{\mathbf{n}} \sin \varphi & (3) & & \mathbf{M}_3 &= \mathbf{M}_2 - \mathbf{d} \times \mathbf{F}_2 & (12) \\ \mathbf{r}_2 &= \mathbf{r}_1 & (4) & & \mathbf{v} &= \dot{\mathbf{r}}_3 & (13) \\ \mathbf{R}_2 &= \mathbf{R}_{rel} \cdot \mathbf{R}_1 & (5) & & \tilde{\boldsymbol{\omega}} &= \mathbf{R}_3 \cdot \dot{\mathbf{R}}'_3 & (14) \\ \mathbf{F}_2 &= \mathbf{R}_{rel} \cdot \mathbf{F}_1 & (6) & & \mathbf{F}_3 &= \mathbf{R}_3 \cdot m(\dot{\mathbf{v}} - \mathbf{g}) & (15) \\ \mathbf{M}_2 &= \mathbf{R}_{rel} \cdot \mathbf{M}_1 & (7) & & \mathbf{M}_3 &= \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) & (16) \\ 0 &= \mathbf{M}_2 \cdot \mathbf{n} & (8) & & & & \end{aligned}$$

Die Zahl der Gleichungen (skalar gezählt) beträgt somit $15 \times 3 + 1 = 46$, die Zahl der Variablen ist $3 \times 4 \times 3 + 3 \times 3 + 1 = 46$. Dabei wurden Drehmatrizen als drei Variable gezählt; intern werden sie als 9 Größen mit 6 Nebenbedingungen dargestellt.

Zum Vereinfachen der Gleichungen werden zunächst die trivialen Gleichungen dazu genutzt, möglichst viele Größen zu eliminieren:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{0} & (1) & & \mathbf{v} &= \dot{\mathbf{R}}'_{rel} \cdot \mathbf{d} & (13) \\ \mathbf{R}_1 &= \mathbf{1} & (2) & & \mathbf{F}_3 &= \mathbf{R}_{rel} \cdot m(\ddot{\mathbf{R}}'_{rel} \cdot \mathbf{d} - \mathbf{g}) & (15) \\ \mathbf{r}_2 &= \mathbf{0} & (4) & & \mathbf{F}_2 &= \mathbf{R}_{rel} \cdot m(\ddot{\mathbf{R}}'_{rel} \cdot \mathbf{d} - \mathbf{g}) & (11) \\ \mathbf{R}_2 &= \mathbf{R}_{rel} & (5) & & \mathbf{F}_1 &= m(\ddot{\mathbf{R}}'_{rel} \cdot \mathbf{d} - \mathbf{g}) & (6) \\ \mathbf{R}_3 &= \mathbf{R}_{rel} & (10) & & \mathbf{M}_1 &= \mathbf{R}'_{rel} \cdot \mathbf{M}_2 & (7) \\ \mathbf{r}_3 &= \mathbf{R}'_{rel} \cdot \mathbf{d} & (9) & & & & \end{aligned}$$

Damit bleiben noch die folgenden Gleichungen für die Variablen \mathbf{R}_{rel} , $\boldsymbol{\omega}$, \mathbf{M}_3 , \mathbf{M}_2 und φ übrig:

$$\begin{aligned} \mathbf{R}_{rel} &= \mathbf{n} \cdot \mathbf{n}' + (\mathbf{1} - \mathbf{n} \cdot \mathbf{n}') \cos \varphi - \tilde{\mathbf{n}} \sin \varphi & (3) \\ \tilde{\boldsymbol{\omega}} &= \mathbf{R}_{rel} \cdot \dot{\mathbf{R}}'_{rel} & (14) \\ \mathbf{M}_3 &= \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) & (16) \\ \mathbf{M}_2 &= \mathbf{M}_3 + \mathbf{d} \times \mathbf{R}_{rel} \cdot m(\ddot{\mathbf{R}}'_{rel} \cdot \mathbf{d} - \mathbf{g}) & (12) \\ 0 &= \mathbf{M}_2 \cdot \mathbf{n} & (8) \end{aligned}$$

Setzt man schließlich die konkreten Parameterwerte ein, vereinfachen sich die Beziehungen deutlich:

$$\mathbf{R}_{rel} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\dot{\varphi} & 0 \\ \dot{\varphi} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow \boldsymbol{\omega} = \dot{\varphi} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\mathbf{M}_3 = J\ddot{\varphi} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{M}_2 = [(J + md^2)\ddot{\varphi} + mgd \cos \varphi] \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$0 = (J + md^2)\ddot{\varphi} + mgd \cos \varphi$$

Der Winkel φ ist der Gelenkwinkel, er ist 0 bei horizontaler Lage. Ersetzt man ihn durch $\theta = \frac{\pi}{2} + \varphi$, erhält die letzte Gleichung die vertraute Form

$$\ddot{\theta} + \frac{mgd}{J + md^2} \sin \theta = 0$$

Beispiel Verladebrücke

Die Bewegungsgleichung für das Pendel ist mit dem Standardverfahren natürlich wesentlich einfacher zu erhalten. Für das Beispiel der Verladebrücke (Abb. 4) gilt dies nicht: Mit dem Schnittverfahren kommen die Studierenden nicht zum Ziel. Mit MapleSim ist ein entsprechendes Modelica-Modell

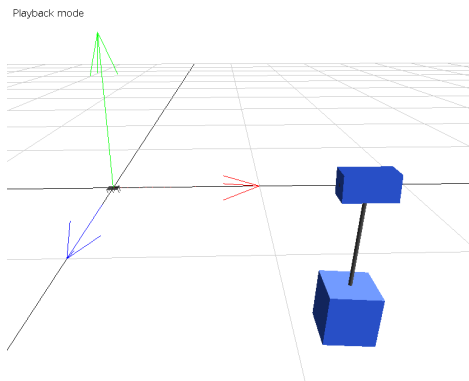


Abbildung 4: Beispiel Verladebrücke

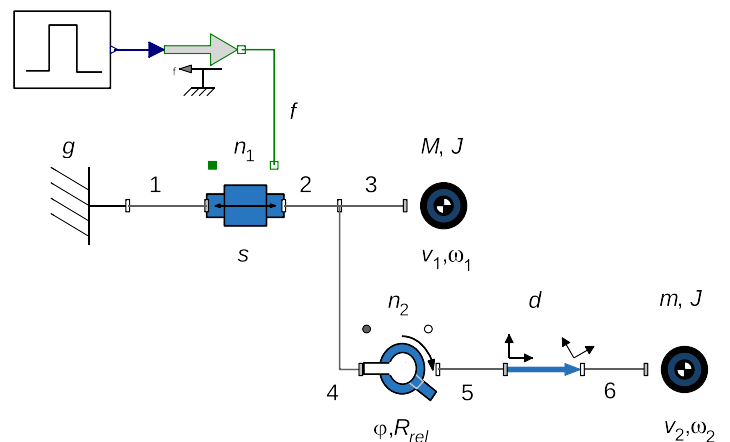


Abbildung 5: Modell Verladebrücke

aber schnell erstellt (Abb. 5), die Parameter haben die Werte

$$\mathbf{g} = \begin{pmatrix} 0 \\ -g \\ 0 \end{pmatrix}, \quad \mathbf{n}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{n}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{J} = \mathbf{0}.$$

Die einfachen Verbindungen reduzieren die Zahl der Frames auf insgesamt sechs, der Dreier-Knoten (2,3,4) liefert die Beziehungen

$$\begin{aligned} \mathbf{r}_2 &= \mathbf{r}_3 = \mathbf{r}_4, & \mathbf{R}_2 &= \mathbf{R}_3 = \mathbf{R}_4 \\ \mathbf{F}_2 &= \mathbf{F}_3 + \mathbf{F}_4, & \mathbf{M}_2 &= \mathbf{M}_3 + \mathbf{M}_4 \end{aligned}$$

Fügt man noch alle Baustein-Gleichungen hinzu, erhält man insgesamt $29 \times 3 + 2 = 89$ Gleichungen für $6 \times 4 \times 3 + 5 \times 3 + 2 = 89$ Variablen.

Völlig analog zum Vorgehen beim Pendel kann man das System in einfacher, wenn auch länglicher Rechnung reduzieren auf

$$\begin{aligned} (m + M) \ddot{s} - md \sin \varphi \ddot{\varphi} &= f + md \cos \varphi \dot{\varphi}^2 \\ - \sin \varphi \ddot{s} + d \ddot{\varphi} &= -g \cos \varphi \end{aligned}$$

Um die Standardform zu erhalten, isoliert man die zweiten Ableitungen, führt wieder den Winkel θ gegen die Senkrechte ein und erhält schließlich

$$\begin{aligned} \ddot{s} &= \frac{f + (g \cos \theta + d\dot{\theta}^2)m \sin \theta}{M + m \sin^2 \theta} \\ \ddot{\theta} &= -\frac{f \cos \theta + Mg \sin \theta + (g + d\dot{\theta}^2 \cos \theta)m \sin \theta}{d(M + m \sin^2 \theta)} \end{aligned}$$

Falls den Studierenden das Euler-Lagrange-Verfahren schon bekannt ist, könnte man hier eine vergleichende Rechnung anführen. Sie ist zwar deutlich kürzer, verlangt dafür aber auch tieferes Verständnis (z. B. der beteiligten Energien).

Schlussfolgerungen

Die den Studierenden aus der Mechanik-Vorlesung vertraute Methode zur Aufstellung von Bewegungsgleichungen über das d'Alembert-Prinzip und die auftretenden Schnittkräfte und -momente ist für die komplexen Gleichungen der Mehrkörper-Dynamik ungeeignet. Außerdem ist sie anfällig für Fehler wie vergessene Kräfte oder falsche Vorzeichen.

Das hier vorgestellte Verfahren durch systematisches Zusammensetzen der Komponentengleichungen führt sehr schnell zu einem hochkomplexen

DAE-System, das sich aber per Hand stark vereinfachen und gelegentlich auch komplett in ein Differenzialgleichungssystem umformen lässt.

Auch wenn die verwendeten Umformungen trivial erscheinen mögen, machten sie den Studierenden anfangs mehr Mühe als erwartet. Neben konzeptionellen Problemen, vor allem der Unterscheidung zwischen einem Vektor und seinen Koordinaten bzgl. einer Basis, musste vor allem das flüssige Rechnen mit Vektoren und Matrizen wiederholt werden – die Veranstaltung zur Linearen Algebra war lange her.

In der Praxis wird das Aufstellen und Umformen der Gleichungen vollständig vom Simulationsprogramm übernommen. Die vorgestellten Überlegungen führen aber nicht nur zu einem besseren Verständnis der Funktionsweise eines solchen Programms, sondern machen auch deutlich, wie Fortschritte in der Mathematik und der Informatik Ingenieuren zur Lösung konkreter Probleme nützlich sind.

Literaturverzeichnis

- [1] **Cellier, F. E.; Kofman, E.:** *Continuous System Simulation*. Springer, New York (2010).
- [2] **Fritzson, P.:** *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley & Son, New York (2004).
- [3] **Junglas, P.:** *Praxis der Simulationstechnik*. Verlag Europa-Lehrmittel, Haan-Gruiten (2014).
- [4] **Maplesoft:** *MapleSim*. <http://www.maplesoft.com/products/maplesim/>
- [5] **Otter, K.; Elmqvist, H.; Mattsson, S. E.:** *The New Modelica MultiBody Library*. In: Proc. 3rd International Modelica Conference, Linköping, Sweden, 311-330 (2003).
- [6] **Wittenburg, J.:** *Dynamics of Multibody Systems*. Springer, Berlin, Heidelberg, 2. Aufl. (2008).

Autor

Prof. Dr. rer. nat. Peter Junglas
 Private Fachhochschule für Wirtschaft und Technik Vechta/Diepholz/Oldenburg
 Schlesierstraße 13a
 D-49356 Diepholz
 E-Mail: peter@peter-junglas.de