

Objektorientierte Modellierung Physikalischer Systeme, Teil 17

Martin Otter, DLR Oberpfaffenhofen

13 Modellierung mechanischer Systeme

Im vorliegenden, letzten Teil dieser Serie wird gezeigt wie *drei-dimensionale, mechanische Systeme* mit der objektorientierten Modellierungstechnik behandelt werden können. Hiermit kann z.B. die mechanische Struktur von Robotern, Fahrzeugen und Satelliten modelliert werden. Im Artikel werden Verformungen von Körpern und von Gelenken vernachlässigt. Die erläuterte Vorgehensweise kann aber auch auf verformbare Komponenten erweitert werden. Programme zur Simulation von mechanischen Systemen gibt es schon seit Beginn der 60-Jahre. Heute ist die Zahl der Softwaresysteme kaum überschaubar. Viel verwendet werden z.B. die Mechanik-Simulationsprogramme ADAMS [1], SIMPACK [6] und WorkingModel [7]. Die Theorie hierzu wird z.B. in [3, 4, 5] beschrieben. Die objektorientierte Modellierungstechnik bietet auch hier Vorteile, da Bibliotheken aus verschiedenen Fachgebieten verwendet werden können, um Gesamtsystem-Modelle zu erstellen. In diesem Umfang ist dies mit traditionellen Mechanik-Programmen kaum möglich.

Beispielsweise wird in [2] gezeigt, wie ein detailliertes Gesamtfahrzeugmodell bestehend aus 3D-Mechanik des Chassis, Verbrennungsmotor und Antriebsstrang aufgebaut wurde, wobei das Antriebsstrangmodell die Hydraulik und ein detailliertes Modell des Automatikgetriebes enthält. Weitere aktuelle Beispiele sind unter <http://www.Modelica.org> in den herunterladbaren Beiträgen zum Modelica Workshop'2000 enthalten.

13.1 Koordinatensysteme

Bei mechanischen Systemen müssen viele der beschreibenden Variablen bezüglich eines Koordinatensystems definiert werden. Die benötigten Beziehungen werden in der hier verwendeten Notation kurz zusammengestellt:

In Bild 1 sind zwei Koordinatensysteme a und b dargestellt. Der Ursprung von b wird eindeutig durch den Ortsvektor von a nach b , d.h. mit $\vec{r}^{ab} = {}^a r_1^{ab} \vec{e}_{a1} + {}^a r_2^{ab} \vec{e}_{a2} + {}^a r_3^{ab} \vec{e}_{a3}$, beschrieben, wobei \vec{e}_{ai} die Einheitsvektoren der orthonormalen Basis von a sind. Die Koordinaten von \vec{r}^{ab} bzgl. a werden in der Spaltenmatrix ${}^a \mathbf{r}^{ab} = [{}^a r_1^{ab}, {}^a r_2^{ab}, {}^a r_3^{ab}]^T$ zusammengefasst. Generell wird das Koordinatensystem in dem eine Größe dargestellt wird, durch einen linken oberen Index angegeben. Die Verdrehung von b gegenüber a wird

durch die 3×3 Transformations-Matrix ${}^b \mathbf{T}^a$ beschrieben. Diese gibt an wie die Koordinaten eines im Koordinatensystems a dargestellten Vektors ${}^a \mathbf{h}$ in die Koordinatendarstellung von b umgerechnet wird: ${}^b \mathbf{h} = {}^b \mathbf{T}^a \cdot {}^a \mathbf{h}$.

13.2 Redundante Beschreibungsformen

Jede Transformationsmatrix \mathbf{T} ist orthogonal, d.h. $\mathbf{T}^T \mathbf{T} = \mathbf{E}$ (\mathbf{E} ist die Einheitsmatrix). Die 9 Elemente von \mathbf{T} sind deswegen nicht unabhängig voneinander, sondern erfüllen 6 Zwangsgleichungen. Damit könnte eine Rotation durch $9 - 6 = 3$ voneinander unabhängigen Variablen beschrieben werden. Jede solche Beschreibungsform ist jedoch zumindest an einer Stelle singular, siehe z.B. [3, 4]. Aus diesem Grunde muss entweder während einer Simulation zwischen unterschiedlichen nicht-redundanten Variablensätzen geschaltet werden, bei denen die Singularität an verschiedenen Stellen liegt, oder es muss eine *redundante Beschreibungsform* gewählt werden.

Für eine redundante Beschreibung ist die Transformationsmatrix aus Effizienzgründen nicht geeignet, da 6 nichtlineare Zwangsgleichungen zu berücksichtigen sind. Eine Alternative ist die Darstellung der Rotation mit 4 voneinander abhängigen Variablen ${}^b \mathbf{p}^a = [{}^b p_1^a, {}^b p_2^a, {}^b p_3^a, {}^b p_4^a]^T$, den sogenannten Quaternionen, siehe [3, 4]. Diese sind definiert als:

$${}^b \mathbf{p}^a = \begin{bmatrix} \mathbf{n} \cdot \sin(\varphi/2) \\ \cos(\varphi/2) \end{bmatrix} \quad (13.1)$$

wobei \mathbf{n} ein Einheitsvektor in Richtung der Drehachse ist, um das Koordinatensystem a in das Koordinatensystem b mittels einer ebenen Rotation um den Winkel φ zu überführen. Die 4 Variablen vom Vektor $\mathbf{p} = {}^b \mathbf{p}^a$ sind ebenfalls nicht unabhängig voneinander, sondern müssen die Zwangsgleichung $\mathbf{p}^T \mathbf{p} = 1$ erfüllen. Man könnte jetzt z.B.

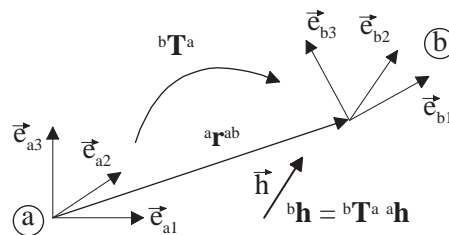


Bild 1: Beziehungen zwischen Koordinatensystemen.

eine nicht-redundante Beschreibungsform mit den 3 Variablen $\mathbf{r} = [p_1, p_2, p_3]^T$ wählen und könnte die 4-te Variable durch Auflösung der Zwangsgleichung berechnen:

$$p_4 = \pm \sqrt{\mathbf{r}^T \mathbf{r} - 1}; \quad \dot{p}_4 = \frac{\pm \dot{\mathbf{r}}}{\sqrt{\mathbf{r}^T \mathbf{r} - 1}} \cdot \dot{\mathbf{r}} \quad (13.2)$$

Es ist leicht zu sehen, dass die Lösung der Zwangsgleichung nicht eindeutig ist. Darüberhinaus liegt eine Singularität vor, wenn $\mathbf{r}^T \mathbf{r} = 1$ ist, da dann die Ableitung von p_4 auf Grund des verschwindenden Nenners "unendlich" wird. Als Fazit kann festgehalten werden, dass eine überall gültige Beschreibung der Rotation mit denselben Variablen nur mit einem redundanten Variablensatz möglich ist.

Diese Eigenschaft führt auf erhebliche Schwierigkeiten bei der mathematischen Behandlung mechanischer Systeme. Um die Zahl der nichtlinearen Zwangsgleichungen zu reduzieren wird versucht, Quaternionen rekursiv zu berechnen. Zum Beispiel kann \mathbf{p}_1 aus der Gleichung $\mathbf{p}_1 := \mathbf{p}_2$ ohne Zuhilfenahme der Zwangsgleichung für \mathbf{p}_1 bestimmt werden, wenn \mathbf{p}_2 an anderer Stelle schon konsistent berechnet wird. Eine solche Formulierung kann jedoch dazu führen, dass das Gesamtsystem überzählige Gleichungen enthält. In Abschnitt 13.5 wird gezeigt, wie diese vor der eigentlichen Transformation auf Zustandsform automatisch entfernt werden können, vorausgesetzt die Gleichungen zur Berechnung redundanter Variablen sind entsprechend gekennzeichnet (hier: durch die Bezeichnung "redundant").

Um die Darstellung eines Vektors \mathbf{h} zwischen zwei verschiedenen Koordinatensystemen umzurechnen, werden die beiden folgenden Hilfsfunktionen verwendet:

$${}^b \mathbf{h} = \text{resolve}^{ba}({}^b \mathbf{p}^a, {}^a \mathbf{h}); \quad {}^a \mathbf{h} = \text{resolve}^{ab}({}^b \mathbf{p}^a, {}^b \mathbf{h})$$

Hierbei werden als erstes Argument die Quaternionen übergeben, die die Rotation von a nach b beschreiben. Zum Beispiel hat die Funktion resolve^{ba} den folgenden Aufbau, siehe z.B. [3] ($\mathbf{p} = {}^b \mathbf{p}^a$, $\mathbf{e} = [p_1, p_2, p_3]^T$):

$${}^b \mathbf{h} = \{(2p_4 p_4 - 1)\mathbf{E} + 2\mathbf{e}\mathbf{e}^T - 2p_4 \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}\} {}^a \mathbf{h}$$

13.3 Schnittstellen für 3-dim. Mechanik

Wie üblich besteht der erste Schritt beim Aufbau einer Komponentenbibliothek in der Festlegung der Komponenten-Schnittstellen. Bei mechanischen Systemen wird eine Komponente freigeschnitten, siehe Bild 2. In der Schnittebene wird ein Koordinatensystem a definiert. Die Position und Orientierung von a bezüglich des Inertialsystems 0 wird durch den Ortsvektor ${}^0 \mathbf{r}^{0a}$ und die (redundanten) Quaternionen ${}^a \mathbf{p}^0$ beschrieben. In der Schnittebene wirken die resultierende Schnittkraft ${}^a \mathbf{f}^a$ und das resultierende Schnittmoment ${}^a \boldsymbol{\tau}^a$. Beide Vektoren sind Flußvariablen (siehe Teil 3 dieser Serie) und werden in a angeschrieben, da die Interpretation von z.B. Lagerreaktionen am anschaulichsten in einem bauteilfesten System erfolgt. Eine Modelica-Definition hat deswegen den folgenden Aufbau:

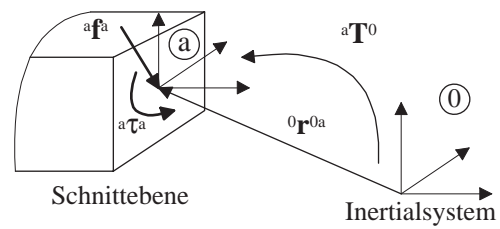


Bild 2: 3-dimensionale Mechanik Schnittstelle.

```
connector Frame "3D-Schnittstelle"
  Real r[3] "Ortsvektor zum Ursprung";
  Real p[4] "Rotation 0->Frame";
  flow Real f[3] "Schnittkraft";
  flow Real t[3] "Schnittmoment";
end Frame;
```

Werden mehrere 3D-Schnittstellen miteinander verbunden, sind die Koordinatensysteme dieser Schnittstellen *identisch*. Damit werden die Vektoren dieser Schnittstellen in demselben Koordinatensystem angegeben. Da die Schnittkräfte und -Momente Flußvariablen sind, wird die Summe der Schnitt-Kräfte und die Summe der Schnitt-Momente zu Null gesetzt.

13.4 Grundelemente

In Tabelle 1 sind die wichtigsten Grundelemente für eine 3-dim. Mechanik Bibliothek aufgeführt. Jede Komponente besitzt ein oder zwei Schnittstellen. Die Gleichungen einer Komponente beschreiben die Beziehungen zwischen den Schnittstellen- und internen Variablen:

Beim *Inertialsystem* werden die kinematischen Größen initialisiert. Die Komponente *Translation* definiert eine starre, masselose Verbindung zwischen zwei Schnittstellen mit einem vorgegebenen Abstand. Die letzten beiden Gleichungen ergeben sich aus dem Kräfte- und Momentengleichgewicht. Die Komponente *Drehgelenk* besitzt noch eine 1D-Schnittstelle (siehe auch Teil 3 und 9 dieser Serie) um das Gelenk anzutreiben. Da die Ableitungen der Relativgrößen ϕ_1 und $\omega_1 = \dot{\phi}_1$ aufgeführt sind, werden diese Variablen potentiell als Zustandsgrößen verwendet. Die restlichen Gleichungen beschreiben die kinematischen Beziehungen zwischen den Schnittstellen, sowie das Kräfte- und Momentengleichgewicht. Die letzte Gleichung, $\tau_1 = \mathbf{n}^T \boldsymbol{\tau}^b$, ergibt sich aus dem Energiesatz: Schnittkräfte leisten keine Arbeit, so dass in Richtung der freien Bewegungsrichtung nur das externe, antreibende Moment τ_1 wirkt. Ein *Schubgelenk* wird ähnlich wie ein Drehgelenk beschrieben; wird aber aus Platzgründen nicht diskutiert. Die Komponente *Kugelgelenk* ist etwas anders aufgebaut als Schub- und Drehgelenk. Hier werden keine Zustände eingeführt und es wird im wesentlichen nur definiert, dass die Ursprünge der beiden Schnittstellen identisch sind und es keine Einschränkung der Rotation zwischen den beiden Schnittstellen gibt. Hierbei ist \mathbf{h} eine Hilfsgröße, die die relative Orientierung zwischen a und b durch relative Quaternionen beschreibt, ähnlich wie beim Drehgelenk.

Mit der Komponente *Starrkörper* werden Masse und Trägheit beschrieben. Als potentielle Zustände werden Lage, Orientierung, Geschwindigkeit und Winkelgeschwindigkeit des Körpers verwendet. Zur Definition der Rotation werden die Quaternionen \mathbf{p}^a des Körpers benutzt, zusammen mit der Zwangsgleichung $\mathbf{p}^{aT} \mathbf{p}^a = 1$, so dass effektiv 3 unabhängige Zustände während der Simulation gewählt werden können. Die letzten beiden Gleichungen in der Komponente enthalten die Newton/Euler-Gleichungen bezüglich der Schnittstelle a . Aus Platzgründen werden Grundelemente, wie Kraftelemente, Gravitationsfeld und Sensoren, nicht aufgeführt. Die Komponenten von Tabelle 1 können benutzt werden, um komplexere Komponenten aufzubauen. Zum Beispiel kann durch eine Verschaltung von einem Drehgelenk mit einem Schubgelenk ein Zylindergelenk, oder von Translationen und einem Starrkörper ein Körper mit mehreren Schnittstellen erstellt werden. Ein einfaches Beispiel für die Benutzung der Bibliothek ist in

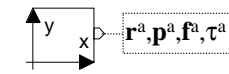
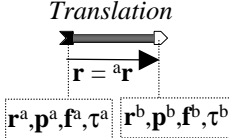
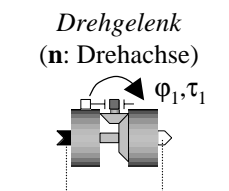
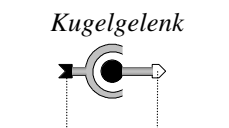
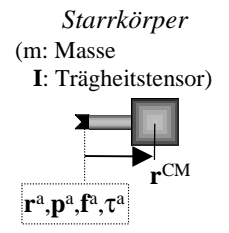
Abkürzungen: $\mathbf{r}^a := {}^0\mathbf{r}^{0a}$ $\mathbf{f}^a := {}^a\mathbf{f}^a$ $\mathbf{r}^b := {}^0\mathbf{r}^{0b}$ $\mathbf{f}^b := {}^b\mathbf{f}^b$ $\mathbf{p}^a := {}^a\mathbf{p}^0$ $\mathbf{t}^a := {}^a\mathbf{t}^a$ $\mathbf{p}^b := {}^b\mathbf{p}^0$ $\mathbf{t}^b := {}^b\mathbf{t}^b$	
	Inertialsystem $\mathbf{r}^a = [0, 0, 0]$ $\mathbf{p}^a = [0, 0, 0, 1]$ (redundant)
	Translation $\mathbf{r}^b = \mathbf{r}^a + \text{resolve}^{0a}(\mathbf{p}^a, \mathbf{r})$ $\mathbf{p}^b = \mathbf{p}^a$ (redundant) $\mathbf{f}^a = \mathbf{f}^b$ $\boldsymbol{\tau}^a = \boldsymbol{\tau}^b + \mathbf{r} \times \mathbf{f}^b$
	Drehgelenk (n: Drehachse) $\omega_1 = \dot{\phi}_1$ (ϕ_1 ist Zustand) $\alpha_1 = \dot{\omega}_1$ (ω_1 ist Zustand) $\mathbf{r}^b = \mathbf{r}^a$ $\mathbf{h} = [\mathbf{n} \cdot \sin(\phi_1/2), \cos(\phi_1/2)]$ $\mathbf{p}^b = \begin{bmatrix} h_4 & -h_3 & h_2 & -h_1 \\ h_3 & h_4 & -h_1 & -h_2 \\ -h_2 & h_1 & h_4 & -h_3 \\ h_1 & h_2 & h_3 & h_4 \end{bmatrix} \mathbf{p}^a$ (redund.) $\mathbf{f}^a = \text{resolve}^{ab}(\mathbf{h}, \mathbf{f}^b)$ $\boldsymbol{\tau}^a = \text{resolve}^{ab}(\mathbf{h}, \boldsymbol{\tau}^b)$ $\tau_1 = \mathbf{n}^T \boldsymbol{\tau}^b$
	Kugelgelenk $\mathbf{r}^b = \mathbf{r}^a$ $\mathbf{h} = \begin{bmatrix} -p_4^a & -p_3^a & p_2^a & p_1^a \\ p_3^a & -p_4^a & -p_1^a & p_2^a \\ -p_2^a & p_1^a & -p_4^a & p_3^a \\ p_1^a & p_2^a & p_3^a & -p_4^a \end{bmatrix} \mathbf{p}^b$ $\mathbf{f}^a = \text{resolve}^{ab}(\mathbf{h}, \mathbf{f}^b)$ $\boldsymbol{\tau}^a = [0, 0, 0]; \quad \boldsymbol{\tau}^b = [0, 0, 0]$
	Starrkörper (m: Masse, I: Trägheitstensor) $1 = \mathbf{p}^{aT} \mathbf{p}^a$ ${}^0\mathbf{v} = \dot{\mathbf{r}}^a$ (\mathbf{r}^a ist Zustand) ${}^0\mathbf{a} = \dot{{}^0\mathbf{v}}$ (${}^0\mathbf{v}$ ist Zustand) ${}^0\boldsymbol{\omega} = 2 \cdot \begin{bmatrix} p_4^a & -p_3^a & p_2^a & -p_1^a \\ p_3^a & p_4^a & -p_1^a & -p_2^a \\ -p_2^a & p_1^a & p_4^a & -p_3^a \end{bmatrix} \cdot \dot{\mathbf{p}}^a$ (\mathbf{p}^a ist Zustand) ${}^0\boldsymbol{\alpha} = \dot{{}^0\boldsymbol{\omega}}$ (${}^0\boldsymbol{\omega}$ ist Zustand) $[\mathbf{a}, \boldsymbol{\omega}, \boldsymbol{\alpha}] = \text{resolve}^{a0}(\mathbf{p}^a, [{}^0\mathbf{a}, {}^0\boldsymbol{\omega}, {}^0\boldsymbol{\alpha}])$ $\mathbf{f}^a = m(\mathbf{a} + \boldsymbol{\alpha} \times \mathbf{r}^{CM} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}^{CM}))$ $\boldsymbol{\tau}^a = \mathbf{I}\boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{r}^{CM} \times \mathbf{f}^a$

Tabelle 1: Bibliothek von 3D-Mechanik-Komponenten.

Bild 4 zu sehen. Es handelt sich um ein einfaches Viergelenk, welches sich aus 3 Drehgelenken, 1 Schubgelenk und 1 Kugelgelenk zusammensetzt.

Bei der Benutzung der vorgestellten Bibliothek gibt es keine wesentlichen Einschränkungen. Insbesondere können kinematische Schleifen auftreten, Körper direkt mit Körper und Gelenke direkt mit Gelenken verschaltet werden. Körper müssen auch nicht über Gelenke mit dem Inertialsystem verbunden sein, da sie eigene Zustände besitzen.

13.5 Symbolische Vorverarbeitung

Durch die Verwendung von redundanten Variablen zur Beschreibung der Rotationen treten Schwierigkeiten auf. Zur Verdeutlichung wird das System im linken Teil von Bild 3 untersucht, welches aus 2 Drehgelenken und 3 Translationen besteht. Dies führt auf die im linken Teil von (13.3) angegebenen Gleichungen für die Rotationen

$$\begin{aligned}
 \mathbf{p}_1 &= [0, 0, 0, 1]^T & [\mathbf{p}_1] &= [0, 0, 0, 1]^T \\
 \mathbf{p}_2 &= \mathbf{p}_1 & [\mathbf{p}_2] &= \mathbf{p}_1 \\
 \mathbf{p}_3 &= \mathbf{f}(\mathbf{p}_2, \phi_1) & [\mathbf{p}_3] &= \mathbf{f}(\mathbf{p}_2, \phi_1) \\
 \mathbf{p}_4 &= \mathbf{p}_3 & [\mathbf{p}_4] &= \mathbf{p}_3 \\
 \mathbf{p}_5 &= \mathbf{f}(\mathbf{p}_4, \phi_2) & [\mathbf{p}_5] &= \mathbf{f}(\mathbf{p}_4, \phi_2) \\
 \mathbf{p}_6 &= \mathbf{p}_5 & [\mathbf{p}_6] &= \mathbf{p}_5 \\
 & & \mathbf{p}_1 &= \mathbf{f}(\mathbf{p}_6, \phi_3)
 \end{aligned} \tag{13.3}$$

Da die Winkel ϕ_1, ϕ_2 Zustände sind, und damit beim Sortieren als bekannt angenommen werden, sind dies $6 \cdot 4 = 24$ Gleichungen zur Bestimmung der 24 Unbekannten $\mathbf{p}_1, \dots, \mathbf{p}_6$, so dass eine Berechnung unproblematisch ist.

Wird jedoch noch ein Drehgelenk hinzugefügt, wie im rechten Teil von Bild 3 gezeigt, so ergibt sich eine Beziehung zwischen \mathbf{p}_1 , das als $[0, 0, 0, 1]^T$ vorgegeben ist und \mathbf{p}_6 , das aus anderen Quaternionen berechnet wird. Wird nun einfach die Gleichung eines Drehgelenks verwendet, wie im rechten Teil von (13.3) gezeigt, werden 4 Gleichungen hinzugefügt, obwohl die Gleichheit zweier Rotationen durch 3 Zwangsgleichungen beschrieben wird, d.h., es ist eine redundante Gleichung enthalten. (Hinweis: durch Hinzufügen von 3 Gleichungen können die Drehwinkel nicht mehr als Zustände verwendet werden, so dass es auch 3 neue Unbekannte gibt; siehe Teil 5+6 dieser Serie). Dieses Problem kann auf die folgende Weise gelöst werden:

In einer Modellierungssprache wie Modelica, muß für redundante Variable \mathbf{v} ein neuer Typ eingeführt werden, ei-

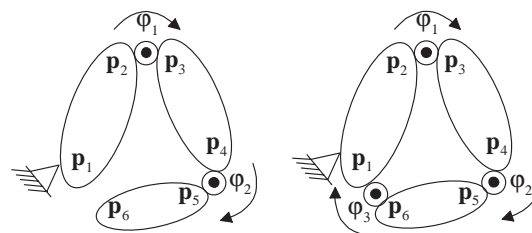


Bild 3: Redundante Gleichungen bei Schleifen.

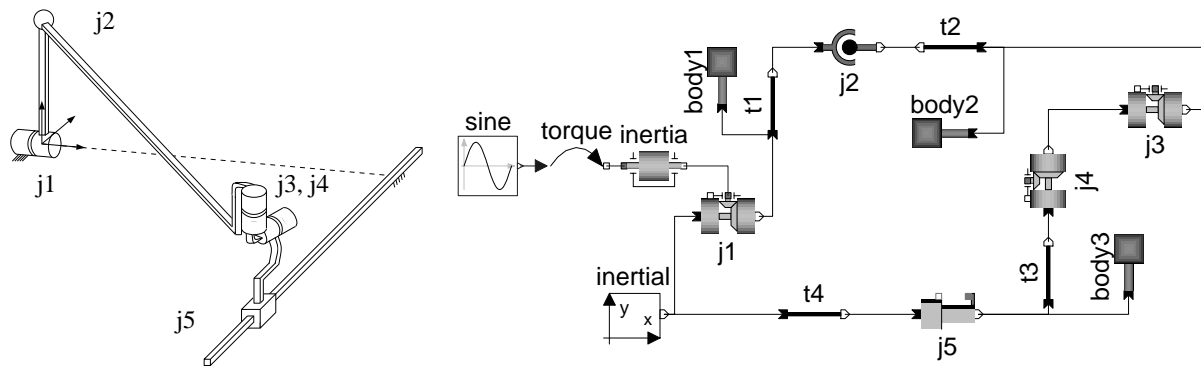


Bild 4: Modell eines Viergelenks (links: 3D-Bild; rechts: Modelica-Modell inklusive 1D-Antrieb von Gelenk j1).

ne Hilfsfunktion $\mathbf{constraint}(\mathbf{v}) = \mathbf{0}$, um die Zwangsgleichungen zwischen diesen Variablen anzugeben, sowie die Hilfsfunktion $\mathbf{equality}(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{0}$ um die nicht-redundante Zahl von Gleichungen zur Beschreibung der Gleichheit zwischen 2 Variablen diesen Typs zu definieren. Hierbei muß gelten: $\mathbf{equality}(\mathbf{v}, \mathbf{v}) \equiv \mathbf{0}$. Zum Beispiel besteht die $\mathbf{equality}(\mathbf{p}^a, \mathbf{p}^b)$ Funktion bei Quaternionen aus den folgenden 3 Gleichungen:

$$\begin{bmatrix} -p_4^a - p_3^a & p_2^a & p_1^a \\ p_3^a - p_4^a & -p_1^a & p_2^a \\ -p_2^a & p_1^a & -p_4^a & p_3^a \end{bmatrix} \mathbf{p}^b = \mathbf{0}$$

In einer Vorverarbeitung werden nun alle als "redundant" gekennzeichneten Gleichungen untersucht, wie im rechten Teil von (13.3) gezeigt. Hierbei wird der Zuordnungsalgorithmus (siehe Teil 4 dieser Serie) benutzt um für alle unbekannt redundanten Variablen eine Zuordnung zu finden (in (13.3) durch [...] charakterisiert). Verbleiben Gleichungen in denen keine Variable eine Zuordnung erhalten hat, wie in der letzten Gleichung vom rechten Teil von (13.3), dann sind dies Zwangsgleichungen zwischen redundanten Variablen. Diese werden durch nicht-redundante Gleichungen ersetzt, indem die Funktion $\mathbf{equality}$ verwendet wird (hier: $\mathbf{equality}(\mathbf{p}_1, \mathbf{f}(\mathbf{p}_6, \phi_3)) = \mathbf{0}$). Danach können die in Teil 4-6 besprochenen Algorithmen verwendet werden, um das System auf Zustandsform zu transformieren. In der Regel werden dabei die kinematischen Gleichungen zweimal differenziert und Zustände werden statisch oder dynamisch während der Simulation ausgewählt bzw. entfernt.

Starrkörper enthalten die Zwangsgleichung $\mathbf{constraint}(\mathbf{p}_1) = \mathbf{p}_1^T \mathbf{p}_1 - 1$. In vielen Fällen wird die $\mathbf{equality}()$ Funktion nun auf die Verbindungsgleichung zu einem Starrkörper angewendet. Dann liegen die Gleichungen $\mathbf{0} = \mathbf{equality}(\mathbf{p}_1, \mathbf{p}_2)$ und $0 = \mathbf{constraint}(\mathbf{p}_1)$ vor. Es kann leicht nachgewiesen werden, dass diese 4 Gleichungen durch die identische Beziehung $\mathbf{p}_1 = \mathbf{p}_2$ ersetzt werden können.

13.6 Schlusswort zu dieser Serie

In Teil 1–8 dieser Serie wurde eine Einführung in die objektorientierte Modellierung physikalischer Systeme gegeben und die verschiedenen Algorithmen skizziert um

diese Technik einsetzen zu können. In Teil 9–17 wurde dann diese neue Modellierungsmethodik auf Antriebsstränge, elektronische Schaltungen, elektrische Energiesysteme, hydraulische Systeme, thermo-fluid Rohrströmungen, Verbrennungsmotoren, und 3-dim. mechanische Systeme angewendet. Viele der besprochenen Modelle sind in frei verfügbaren Modelica Komponenten-Bibliotheken enthalten (siehe <http://www.Modelica.org/library/library.html>) und können damit sofort eingesetzt werden. Mit dem vorliegenden Artikel wird diese Serie jetzt beendet. Ich möchte mich an dieser Stelle ganz herzlich beim AT-Chefredakteur Dr. Reißweber für die Initiierung dieser Serie und seine tatkräftige Unterstützung bedanken.

Danksagung

Für hilfreiche Diskussionen zur objektorientierten Beschreibung von mechanischen Systemen bedanke ich mich bei Hilding Elmqvist, Sven-Erik Mattsson und Hans Olsson.

Literatur

- [1] ADAMS. Homepage: <http://www.adams.com/>
- [2] Bowles, P., Tiller, M., Elmqvist, H., Brück, D., Mattsson, S.-E., Möller, A., Olsson, H. und Otter, M.: Feasibility of Detailed Vehicle Modeling. 2001 SAE World Congress, Detroit, März 2001, (angenommener Beitrag).
- [3] Nikravesh, P.E.: Computer-Aided Analysis of Mechanical Systems. Prentice Hall, 1988.
- [4] Roberson, R.E. und Schwertassek, R.: Dynamics of Multibody Systems. Springer 1988.
- [5] Schiehlen, W.: Multibody Systems Handbook, Springer 1990.
- [6] SIMPACK. Homepage: <http://www.simpack.de/>
- [7] WorkingModel. Homepage: <http://www.workingmodel.com/>

Manuskripteingang: 8. Oktober 2000.



Dr.-Ing. Martin Otter ist Projektleiter beim DLR, Lehrbeauftragter an der TU München und Chairman der Modelica Association. Hauptarbeitsgebiete: Steuerung und Regelung von Industrierobotern, Echtzeitsimulation, objektorientierte Modellierung, Modelica Sprachentwurf.

Adresse: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Robotik und Mechatronik Postfach 1116, D-82230 Weßling, Tel.: 08153/28-2473, Email: Martin.Otter@DLR.de