Gefesselte Masse

Jörg J. Buchholz

23. Mai 2017

1 Einleitung

In Abbildung 1 ist eine Punktmasse m dargestellt, die sich, von einem masselosen starren Stab der Länge l gefesselt, auf einer Kreisbahn bewegt. Dabei wirken auf die Masse eine äußere Antriebskraft **F** und die Stabkraft **S**.

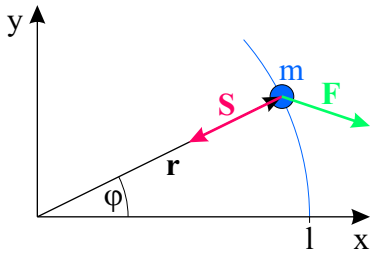

Abbildung 1: Gefesselte Punktmasse auf Kreisbahn

Da die freie Bewegung der Masse (zwei Freiheitsgrade x und y) durch den Stab eingeschränkt wird, ergibt sich bei der Modellierung ein differenzial-algebraisches Gleichungssystem (DAE), das sowohl Differenzialgleichungen als auch algebraische Gleichungen beinhaltet.

Im Folgenden wird die Massenbewegung auf fünf verschiedene Arten simuliert:

1. als klassisches DAE-System mit vier Differenzialgleichungen und einer algebraischen Gleichung unter Matlab
2. als klassisches DAE-System mit vier Differenzialgleichungen und einer algebraischen Gleichung unter Simulink
3. als Differenzialgleichungssystem mit vier Differenzialgleichungen durch analytisches Auflösen der algebraischen Gleichung

4. als Differenzialgleichungssystem mit vier Differenzialgleichungen durch Ersatz des Stabes durch eine Feder
5. als Differenzialgleichungssystem mit zwei Differenzialgleichungen durch Formulierung in Polarkoordinaten

Alle Simulationen produzieren dabei qualitativ vergleichbare Ergebnisse.

2 Gefesselte Masse als DAE

Newton'sches Kräftegleichgewicht:

$$m \cdot \ddot{\mathbf{r}} = \mathbf{F} + \mathbf{S} \tag{1}$$

Die Stabkraft ist antiparallel zum Positionsvektor:

$$\mathbf{S} = -S \cdot \mathbf{r} \tag{2}$$

Gleichung (2) wird in Gleichung (1) eingesetzt und nach der Beschleunigung aufgelöst:

$$\ddot{\mathbf{r}} = \frac{1}{m} \left(\mathbf{F} - S \cdot \mathbf{r} \right) \tag{3}$$

In Komponenten ausgedrückt ergibt sich

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{1}{m} \left(\begin{bmatrix} F_x \\ F_y \end{bmatrix} - S \begin{bmatrix} x \\ y \end{bmatrix} \right) \tag{4}$$

$$\ddot{x} = \frac{1}{m} \left(F_x - S \cdot x \right) \tag{5}$$

$$\ddot{y} = \frac{1}{m} \left(F_y - S \cdot y \right) \tag{6}$$

Um die zwei Differenzialgleichungen zweiter Ordnung in vier Differenzialgleichungen erster Ordnung zu überführen, werden die Position (x und y) und die Geschwindigkeit ($\dot{x}$ und $\dot{y}$) in beiden Richtungen als neue Zustandsgrößen eingeführt:

$$x_1 = x \tag{7}$$

$$x_2 = \dot{x} = \dot{x_1} \quad \Rightarrow \quad \dot{x_1} = x_2 \tag{8}$$

$$\ddot{x} = \dot{x_2} = \frac{1}{m} \left(F_x - S \cdot x \right) = \frac{1}{m} \left(F_x - S \cdot x_1 \right) \quad \Rightarrow \quad \dot{x_2} = \frac{1}{m} \left(F_x - S \cdot x_1 \right) \tag{9}$$

$$x_3 = y \tag{10}$$

$$x_4 = \dot{y} = \dot{x_3} \quad \Rightarrow \quad \dot{x_3} = x_4 \tag{11}$$

$$\ddot{y} = \dot{x_4} = \frac{1}{m} \left(F_y - S \cdot y \right) = \frac{1}{m} \left(F_y - S \cdot x_3 \right) \quad \Rightarrow \quad \dot{x_4} = \frac{1}{m} \left(F_y - S \cdot x_3 \right) \tag{12}$$

Die algebraische Zwangsbedingung schränkt die Bewegung auf eine Kreisbahn ein:

$$x^2 + y^2 = l^2 \quad \Rightarrow \quad 0 = l^2 - x_1^2 - x_3^2 \tag{13}$$

Die Zwangsbedingung Gleichung (13) muss so oft differenziert werden, bis dort die algebraische Variable (Stabkraft S) auftritt:

$$\begin{aligned} 0 &= -2x_1\dot{x_1} - 2x_3\dot{x_3} \\ &= x_1\dot{x_1} + x_3\dot{x_3} \end{aligned} \tag{14}$$

Gleichung (8) und Gleichung (11) in Gleichung (14) einsetzen:

$$0 = x_1 x_2 + x_3 x_4 \tag{15}$$

Auch Gleichung (15) muss differenziert werden, da dort noch keine Stabkraft auftritt:

$$0 = \dot{x_1} x_2 + x_1\dot{x_2} + \dot{x_3} x_4 + x_3\dot{x_4} \tag{16}$$

Gleichung (8), Gleichung (9), Gleichung (11) und Gleichung (12) in Gleichung (16) einsetzen:

$$0 = x_2^2 + x_1\frac{1}{m}\left(F_x - S \cdot x_1\right) + x_4^2 + x_3\frac{1}{m}\left(F_y - S \cdot x_3\right) \tag{17}$$

Der erweiterte Zustandsvektor $\mathbf{z}$ beinhaltet zusätzlich die unbekannte Stabkraft S:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ S \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ S \end{bmatrix} \tag{18}$$

Die erweiterte Vektordifferenzialgleichung beinhaltet auch die algebraische Gleichung als letzte Zeile:

$$\mathbf{M} \cdot \dot{\mathbf{z}} = \mathbf{f}\left(\mathbf{z}, t\right) \quad \text{mit} \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

Einsetzen von Gleichung (8), Gleichung (9), Gleichung (11), Gleichung (12) und Gleichung (18) in Gleichung (19) ergibt

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{S} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m}\left(F_x - S \cdot x_1\right) \\ x_4 \\ \frac{1}{m}\left(F_y - S \cdot x_3\right) \\ x_2^2 + x_1\frac{1}{m}\left(F_x - S \cdot x_1\right) + x_4^2 + x_3\frac{1}{m}\left(F_y - S \cdot x_3\right) \end{bmatrix} \tag{20}$$

2.1 Implementation unter Matlab

Das erweiterte Differenzialgleichungssystem Gleichung (20) kann jetzt mit Matlabs Integrationsverfahren für DAEs (z. B. `ode15s`) integriert werden:

```
M = [ ...
    1 0 0 0 0; ...
    0 1 0 0 0; ...
    0 0 1 0 0; ...
    0 0 0 1 0; ...
    0 0 0 0 0 ...
    ]

x0 = [0; 0; l; 0; 0]

t = linspace (0, 10, 100)

options = odeset ( ...
    'Mass', M, ...
    'RelTol', 1e-6 ...
    )

[t, x] = ode15s (@gefesseltemassedae, t, x0, options);

function out = gefesselte_masse_dae (t, x)

global l m Fx Fy

out = [ ...
    x(2); ...
    1/m*(Fx - x(5)*x(1)); ...
    x(4); ...
    1/m*(Fy - x(5)*x(3)); ...
    x(2)*x(2) + x(1)/m*(Fx - x(5)*x(1)) + ...
        x(4)*x(4) + x(3)/m*(Fy - x(5)*x(3)) ...
    ];
```

Dabei sollten möglichst konsistente Anfangsbedingungen für $\mathbf{z}(t = 0)$ vorgegeben werden, die auch die algebraische Randbedingung Gleichung (17) erfüllen.

Beispielsweise führt eine Wahl von $m = 1, l = 42, \mathbf{F} = \begin{bmatrix}100 & 0\end{bmatrix}^{\mathrm{T}}$ und $\mathbf{z}_0 = \begin{bmatrix}0 & 0 & 42 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ zu der in Abbildung 2 und Abbildung 3 dargestellten Bewegung.

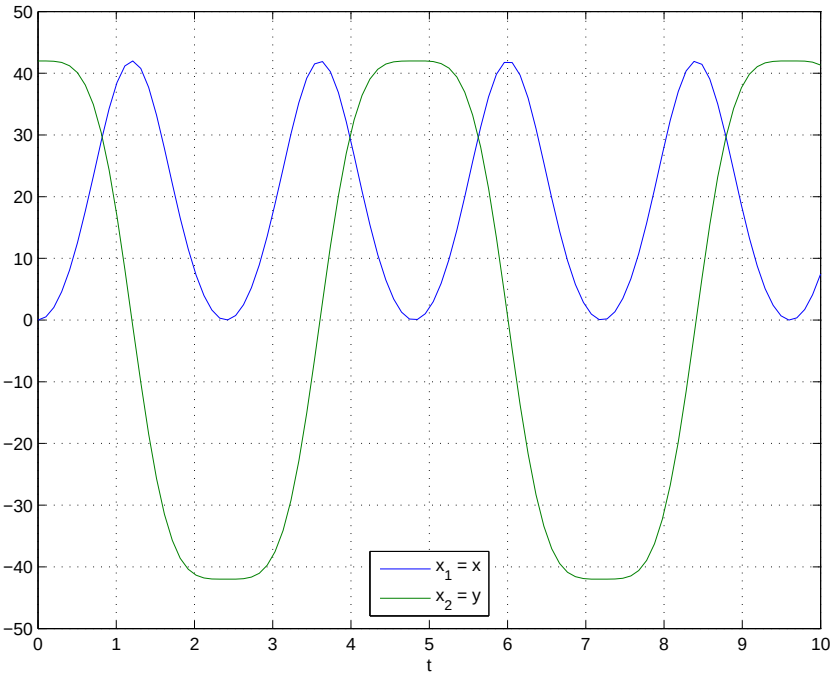

Abbildung 2: Positionskomponenten

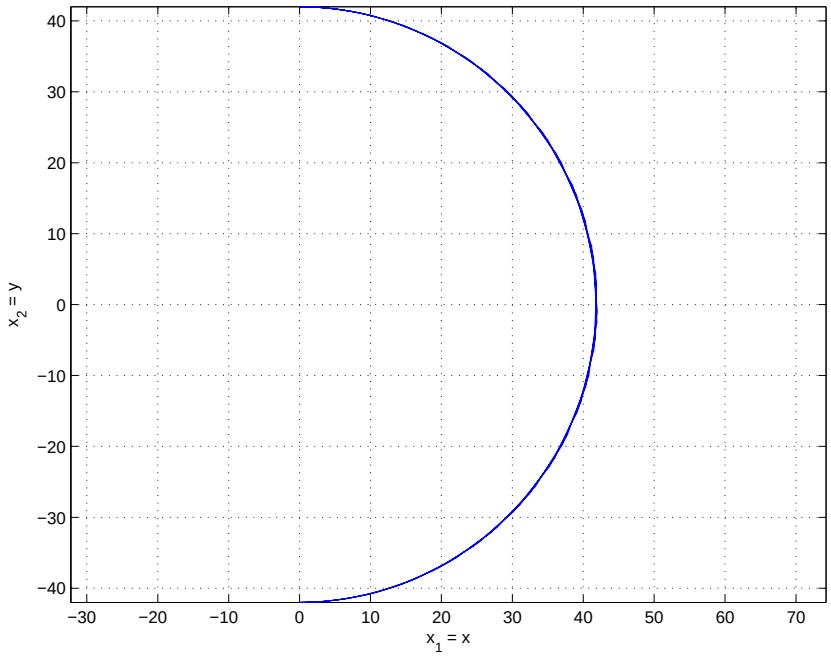

Abbildung 3: Positionsortskurve

2.2 Implementation unter Simulink

Bei der Simulation der DAE unter SIMULINK kann Gleichung (3) direkt mittels zweier Vektorintegratoren für die Vektoren $\dot{\mathbf{r}}$ und $\mathbf{r}$ implementiert werden (Abbildung 4).

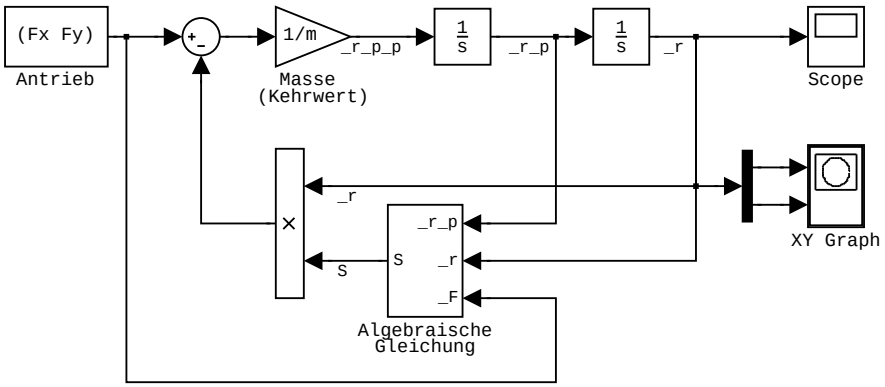

Abbildung 4: DAE-Blockschaltbild

Zusätzlich muss die Stabkraft S im Block „Algebraische Gleichung“ in jedem Zeitschritt mit einem Gleichungslöser (Block „Algebraic Constraint“) aus der algebraischen Gleichung (17) berechnet werden (Abbildung 5).

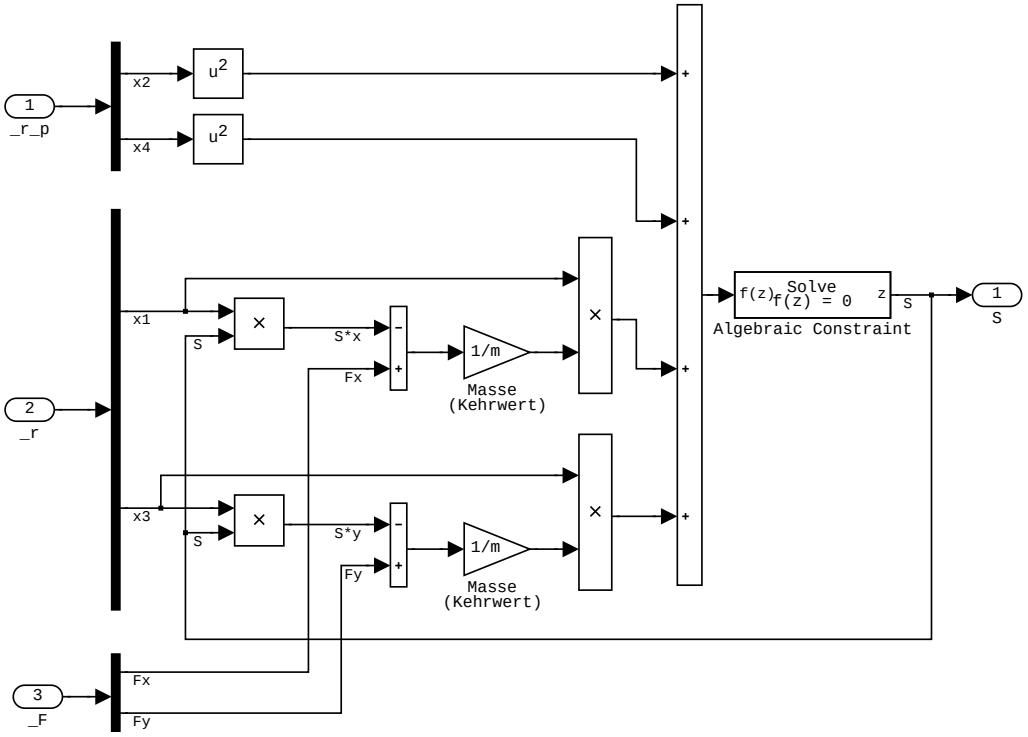

Abbildung 5: Lösung der algebraischen Gleichung

3 Algebraische Gleichung analytisch lösen

In manchen Fällen – wie bei diesem einfachen Beispiel – ist es möglich, die algebraische Randbedingung Gleichung (17) analytisch zu lösen

$$S = \frac{m\left(x_2^2 + x_4^2\right) + F_x x_1 + F_y x_3}{x_1^2 + x_3^2} \tag{21}$$

und einzusetzen, so dass (zusammen mit Gleichung (21)) direkt die vier Differentialgleichungen (8), (9), (11) und (12) modelliert werden können. Die Massenmatrix aus Gleichung (19) ist jetzt leer, da aus dem Algebrodifferenzialgleichungssystem ein gewöhnliches Differentialgleichungssystem geworden ist:

```
options = odeset ( ...
    'Mass', [], ...
    'RelTol', 1e-6 ...
    )

[t, x] = ode15s (@gefesselte_masse_ode, t, x0, options);
```

```
function out = gefesselte_masse_ode (t, x)

global l m Fx Fy

s = (m*(x(2)*x(2) + x(4)*x(4)) + ...
        Fx*x(1) + Fy*x(3)) / (x(1)*x(1) + x(3)*x(3));

out = [ ...
    x(2); ...
    1/m*(Fx - s*x(1)); ...
    x(4); ...
    1/m*(Fy - s*x(3)); ...
    ];
```

Auch unter Simulink vereinfacht sich der in Abbildung 4 unter „Algebraische Gleichung“ zu findende Block (Abbildung 6). Er enthält jetzt keinen rechenzeitaufwendigen Algebraic Constraint-Block mehr.

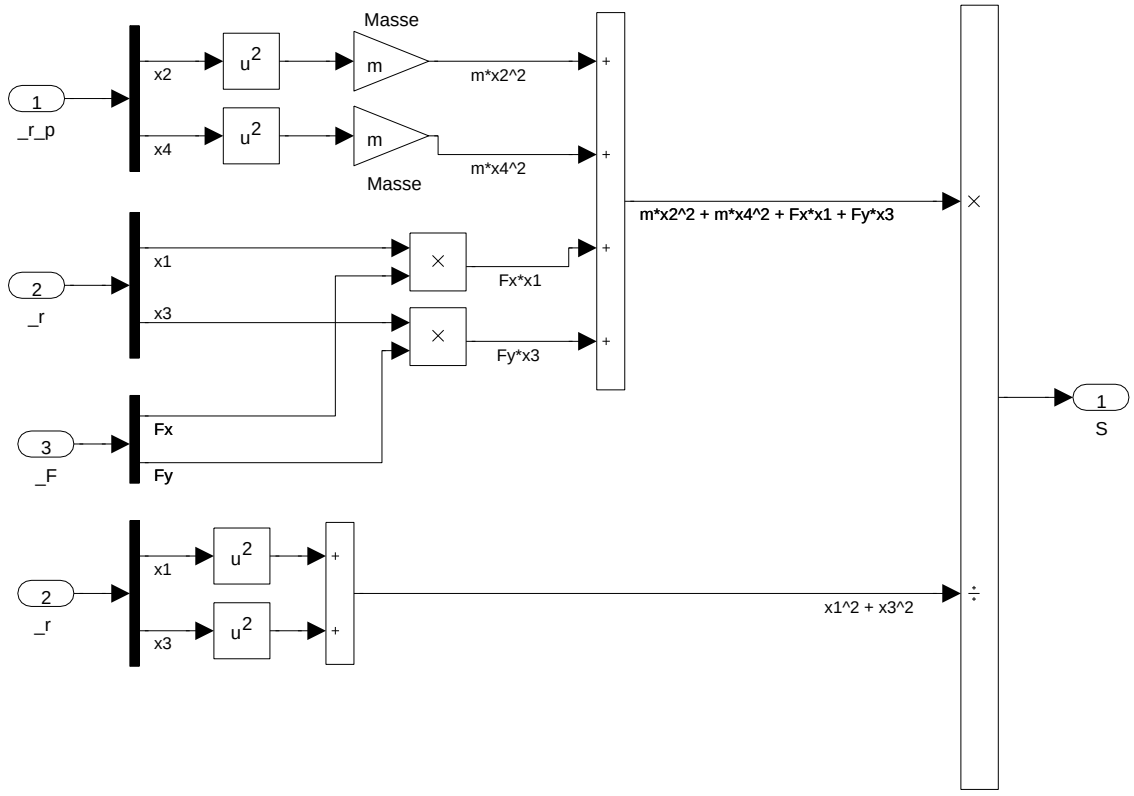

Abbildung 6: Aufgelöste algebraische Gleichung

4 Starren Stab durch Feder ersetzen

Unter der Annahme, dass der Stab nicht völlig starr ist, kann er durch eine sehr steife Feder gleicher Ruhelänge ersetzt werden, so dass die starre algebraische Zwangsbedingung entfällt. Die bisherige Stabkraft S wird jetzt durch die Federkraft S ersetzt, die von der Auslenkung Δl der Feder und von der Federkonstante c abhängt:

$$S = c \cdot \Delta l = c \cdot (|\mathbf{r}| - l) \tag{22}$$

Unter SIMULINK wird dazu der Block ,,Algebraische Gleichung“ in Abbildung 4 durch den wesentlich einfacheren Stabfeder-Block ersetzt, der als Eingangsgröße nur noch die aktuelle Position der Masse benötigt (Abbildung 7).

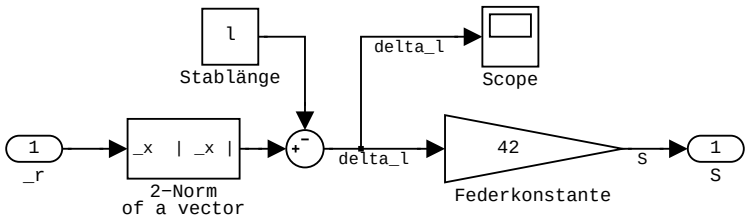

Abbildung 7: Stabfeder-Blockschaltbild

Bedingt durch die Auslenkbarkeit der Feder bewegt sich die Masse jetzt nicht mehr auf einer exakten Kreisbahn; die Abweichung davon (`delta_l` in Abbildung 7) schwingt mit der gleichen Frequenz wie die Horizontalkomponente der Position (Abbildung 8).

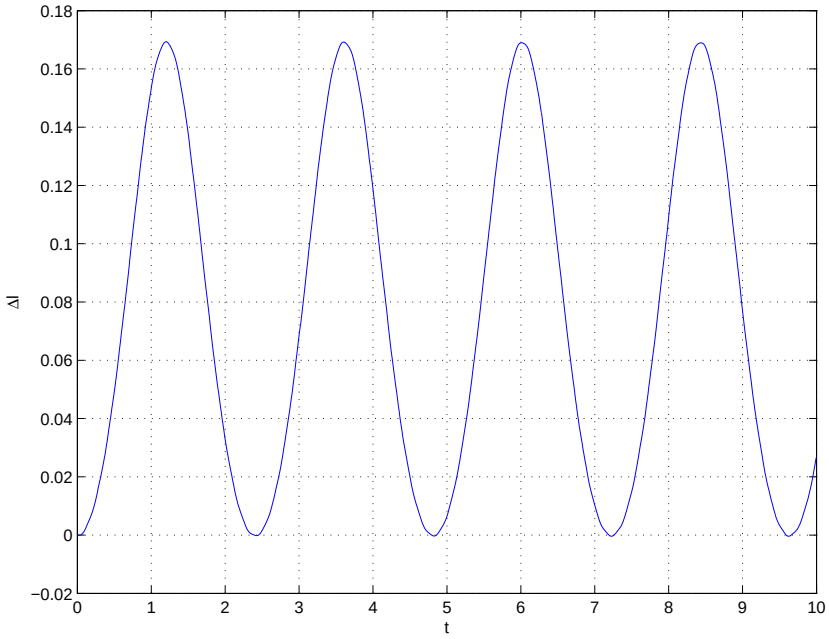

Abbildung 8: Federauslenkung

Eine Vergrößerung der Federsteifigkeit reduziert zwar einerseits wünschenswerterweise die Amplitude der Fehlerschwingung, erhöht aber andererseits („Nomen est Omen“) die Steifigkeit des zu simulierenden Systems, so dass mit hochfrequenten Instabilitäten, angepassten Integrationsverfahren, kleineren Schrittweiten und dadurch längeren Simulationzeiten zu rechnen ist.

5 Polarkoordinaten

Durch das Einführen von Polarkoordinaten lässt sich dieses einfache akademische Beispiel in seine Minimalform (minimale Anzahl von Zustandsgrößen, Differenzialgleichungen, ...) überführen.

Mit dem Momentenvektor $\mathbf{Q}$, dem Trägheitstensor $\mathbf{I}$ und dem Drehbeschleunigungsvektor $\ddot{\mathbf{\Phi}}$ lautet die Momentengleichgewichtsbedingung im Ursprung:

$$\mathbf{I} \cdot \ddot{\mathbf{\Phi}} = \mathbf{Q} = \mathbf{r} \times \mathbf{F} \tag{23}$$

Da es sich um eine zweidimensionale Bewegung handelt, besitzen sowohl der Momenten- als auch der Drehbeschleunigungsvektor nur eine z-Komponente und beim Trägheitstensor

ist nur das z-Trägheitsmoment relevant:

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & ml^2 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} F_x \\ F_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ x \cdot F_y - y \cdot F_x \end{bmatrix} \tag{24}$$

so dass nur eine skalare Differenzialgleichung zweiter Ordnung übrig bleibt:

$$ml^2\ddot{\varphi} = x \cdot F_y - y \cdot F_x \tag{25}$$

die sich, aufgelöst nach der Drehbeschleunigung,

$$\ddot{\varphi} = \frac{1}{ml^2}\left(x \cdot F_y - y \cdot F_x\right) = \frac{1}{ml^2}\left(\mathbf{r} \times \mathbf{F}\right)_z \tag{26}$$

sehr kompakt modellieren lässt (Abbildung 9).

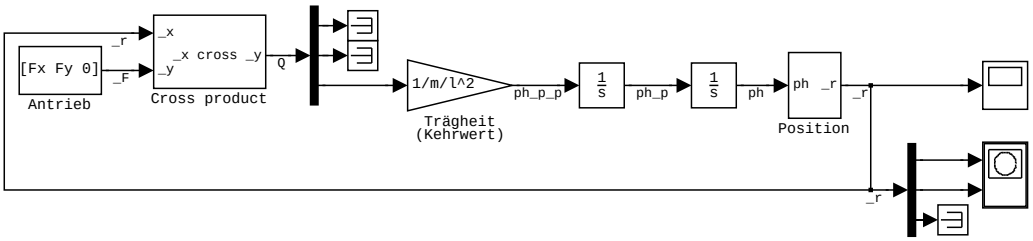

Abbildung 9: Polarkoordinaten-Blockschaltbild

Dabei findet die Umwandlung des Drehwinkels in die kartesischen Koordinaten

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l \cdot \cos\varphi \\ l \cdot \sin\varphi \\ 0 \end{bmatrix} \tag{27}$$

im Block „Position“ statt (Abbildung 10).

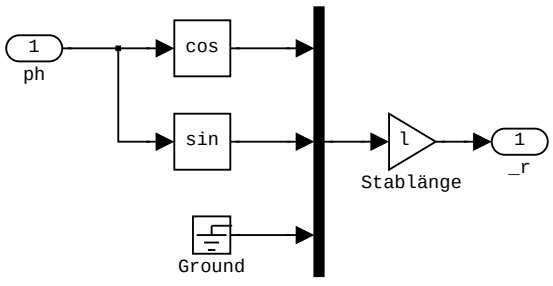

Abbildung 10: Umwandlung des Drehwinkels in kartesische Koordinaten