2. Praktikum
Elastisches Getriebe / Inverses Pendel

2.1 Elastisches Getriebe

Bei realen Motor-Getriebe-Armeinheiten darf das Getriebe (aufgrund von Getriebeelastizitäten) nicht als vollkommen starr angenommen werden. Abbildung 2.1 zeigt einen solchen Aufbau schematisch. Der Arm besitzt eine Masse m, einen Schwerpunktsabstand s und ein Trägheitsmoment C_A^S. Das Gesamtträgheitsmoment der Motorgetriebeeinheit C_M^S ist motorseitig angegeben. Die Getriebeelastizität kann als lineare Drehfeder mit Federkonstante c eingebracht werden. Als Minimalkoordinaten sollen der Getriebeausgangswinkel q_M und der Relativwinkel q_A verwendet werden.

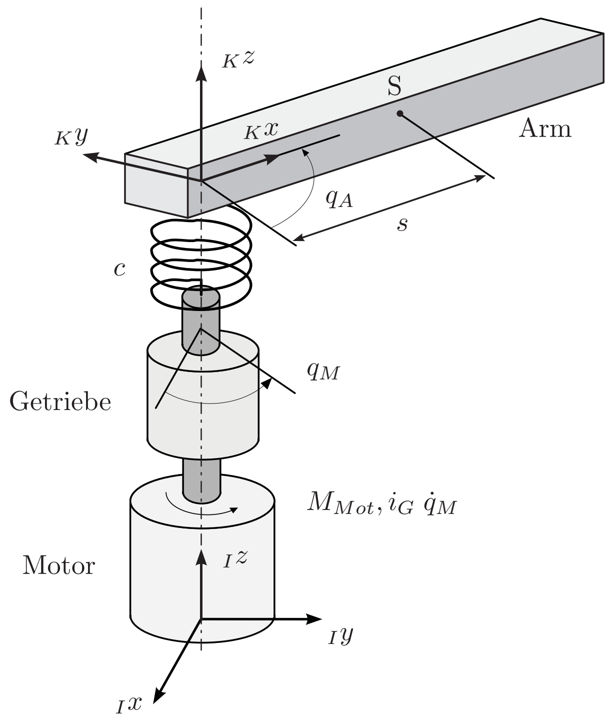


Abbildung 2.1: Motor-Getriebe-Armeinheit

Die Parameter lauten:

Parameter	Wert
Getriebeübersetzung i_G	100
Federkonstante c	1 Nm/rad
Schwerpunktsabstand s	0.208 m
Masse Arm m	0.106 kg
Trägheitsmoment C_A^S	0.0039 kgm^2
Trägheitsmoment C_M^S	$1.05 \cdot 10^{-4}$ kgm^2

Aufgabe 2.1 *Leiten Sie die Bewegungsgleichung der Motor-Getriebe-Armeinheit mit Hilfe der Projektionsgleichung her. Sie erhalten ein lineares System. Stellen Sie die lineare Bewegungsgleichung in der Form*

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u \\ \mathbf{y} &= \mathbf{x} \end{aligned} \tag{2.1}$$

mit dem Zustandsvektor

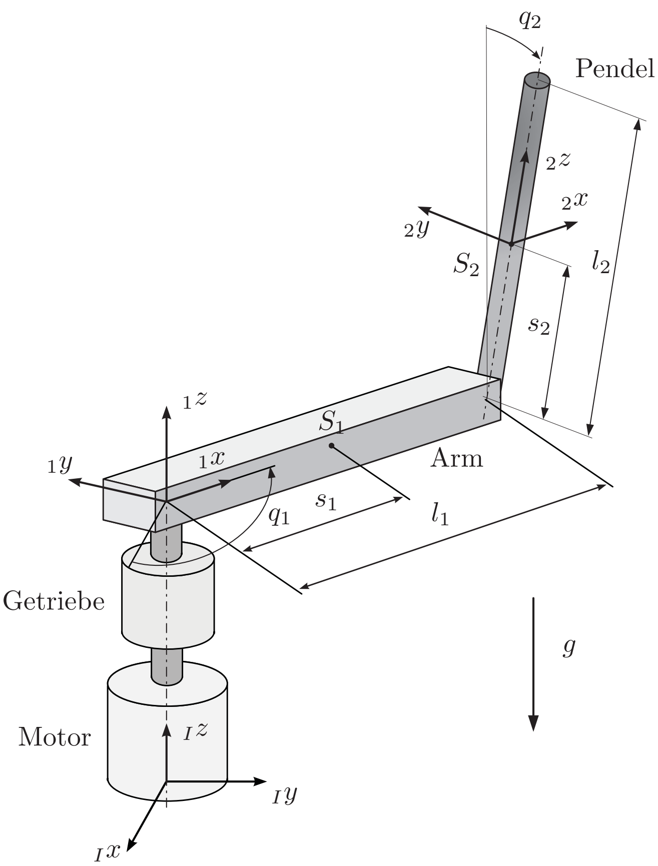
$$\mathbf{x} = [q_M, q_A, \dot{q}_M, \dot{q}_A]^T, \tag{2.2}$$

und dem Eingang $u = M_{Mot}$ in Matlab/Simulink dar. Dazu kann in Simulink z.B. ein State-Space-Block *verwendet werden.*

Aufgabe 2.2 *Machen Sie sich mit dem LQR, auch Riccati-Regler genannt, vertraut. Die dabei zu berechnende Rückführmatrix kann mittels Matlab einfach mit dem Befehl lqr berechnet werden.*

2.2 Inverses Pendel

Abbildung 2.2 zeigt das Modell eines „Inversen Pendels“ (hier ein Furuta Pendel). Dabei lautet das Regelungsziel, beide Winkel q_1 und q_2 auf 0 zu regeln. Es steht aber nur ein Eingriffsmoment dafür zur Verfügung. Man nennt dies ein unteraktuiertes System, da es weniger Stelleingriffe als Regelgrößen gibt. Bis auf die Getriebeelastizität ist der Aufbau des Antriebsstrangs gleich wie beim Modell „Elastisches Getriebe“. Die Parameter für das Getriebe sind daher gleich wie oben, die zusätzlichen oder veränderten Parameter können der Tabelle unten entnommen werden.



Parameter	Wert
Masse m_1	1.2 kg
Masse m_2	0.0725 kg
Länge l_1	0.3 m
Länge l_2	0.25 m
Schwerpunktsabstand s_1	0.225 m
Schwerpunktsabstand s_2	0.125 m
Trägheitsmoment des Armes C_1^S	0.002 kgm^2

Abbildung 2.2: Inverses Pendel

Der Trägheitstensor des Pendels soll mit dem eines homogenen schlanken Stabes der Länge l_2 angenähert werden,

die Trägheit um die Längsachse des Stabs kann dabei vernachlässigt werden ($C^S_{Pendel} = 0$). Das elastische Getriebe wird hier nicht verwendet, daher kann der Motorwinkel direkt mit der Übersetzung aus dem Armwinkel q_1 berechnet werden.

Aufgabe 2.3 *Leiten Sie die (nichtlineare) Bewegungsgleichung des „Inversen Pendels" mit Hilfe von Lagrange II her. Linearisieren Sie die Bewegungsgleichung um den stationären Punkt* $\mathbf{x}_0 = [0\ 0\ 0\ 0]^T$ *und stellen Sie die linearisierten Bewegungsgleichungen in der Form*

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ \mathbf{y} &= \mathbf{x} \end{aligned} \tag{2.3}$$

mit dem Zustandsvektor

$$\mathbf{x} = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T \tag{2.4}$$

und dem Eingang $u = M_{Mot}$ *dar.*

Toolchain

In dieser Lehrveranstaltung wird für die Modellbildung das Computeralgebraprogramm *Maple* und für numerische Berechnungen, Simulationen und grafische Darstellungen *MATLAB* verwendet.
Zur Aufbereitung von symbolischen Modellen für die numerische Simulation ist es möglich, die entsprechenden Ausdrücke aus *Maple* als S-Function für *MATLAB/Simulink* zu implementieren.
Bei einer S-Function handelt es sich um einen Simulink-Block, dessen Funktionalität durch Programmierung in einer Hochsprache wie C, C++ oder *MATLAB* nahezu beliebig vorgegeben werden kann. Die Kommunikation mit *Simulink* erfolgt dabei über einige Schnittstellenfunktionen, die in der S-Function implementiert werden müssen und in verschiedenen Abschnitten der Simulation von *Simulink* mehrfach aufgerufen werden. Dabei werden z.B. die Startwerte für ein dynamisches System initialisiert oder die Zustandsableitungen für die Zeitintegration von Zuständen eines Systems berechnet. Alternativ kann eine solche Simulation aus diskreten *Simulink*-Blöcken (z.B. Zeitintegratoren, Funktionen für Zustandsableitung und Ausgang) aufgebaut werden.
Um das oft mühsame Generieren einer Funktion zur Simulation eines Systems mit bereits hergeleiteter Bewegungsgleichung zu erleichtern, kann das am Institut für Robotik entwickelte *Maple*-Paket *SimCode C* verwendet werden. *SimCode C* generiert aus einer in *Maple* vorliegenden Bewegungsgleichung in Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2.5a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \tag{2.5b}$$

mit dem Zustand **x** und dem Eingang **u** bzw. in Konfigurationsraumdarstellung

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}, \mathbf{u}) \tag{2.6a}$$

mit den Minimalkoordinaten **q** entsprechenden *MATLAB*-Code zur Simulation in *Simulink*.

Aufgabe 2.4 *Machen Sie sich mittels der Datei* `Beispiel.mw` *mit dem* SimCode C*-Package vertraut und erzeugen Sie die Make-Datei* `make_example.m` *und führen Sie letztere aus, um die Simulink-Modelldatei* `SimCodeGen_mfun_model.slx` *zu erzeugen. Bauen Sie das Simulationsmodell entsprechend der Vorlage in* `example_mode` *auf und führen Sie eine Simulation durch. Beachten Sie dabei auch die Abtastzeit sowie das Integrationsverfahren aus* `example_model_demo.slx`*. Zur Simulation werden Startwerte und Parameterwerte benötigt, die Sie durch Ausführen der Initialisierungsdatei* `init_example.m` *erhalten.*

Aufgabe 2.5 *Erstellen Sie eine Simulation des nichtlinearen dynamischen Modells des inversen Pendels mittels* SimCode C. *Bilden Sie in derselben Simulation zusätzlich das linearisierte Modell ab. Vergleichen Sie das Verhalten des nichtlinearen Modells (Startwert* $\mathbf{x}_0$*) mit dem des linearisierten Modells durch Simulation.*

Aufgabe 2.1 *Leiten Sie die Bewegungsgleichung der Motor-Getriebe-Armeinheit mit Hilfe der Projektionsgleichung her. Sie erhalten ein lineares System. Stellen Sie die lineare Bewegungsgleichung in der Form*

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u \\ \mathbf{y} &= \mathbf{x} \end{aligned} \tag{2.1}$$

mit dem Zustandsvektor

$$\mathbf{x} = [q_M, q_A, \dot{q}_M, \dot{q}_A]^T, \tag{2.2}$$

und dem Eingang $u = M_{Mot}$ *in Matlab/Simulink dar. Dazu kann in Simulink z.B. ein* State-Space-Block *verwendet werden.*

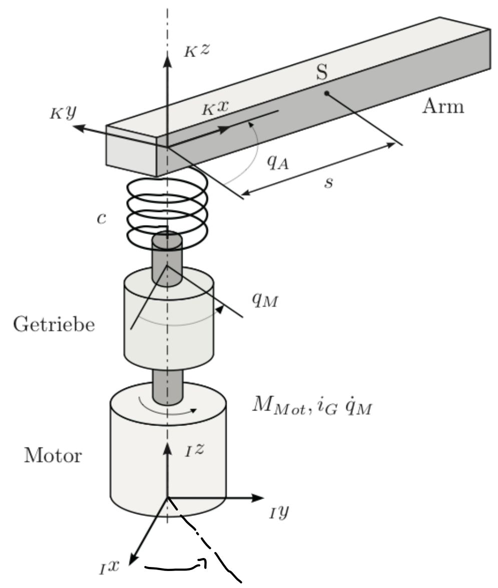


Abbildung 2.1: Motor-Getriebe-Armeinheit

$$R_{KI} = \begin{bmatrix} \cos q_M + q_A & \sin q_M + q_A & 0 \\ -\sin q_M + q_A & \cos q_M + q_A & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$_K r_{SA} = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix} \qquad _I r_{SA} = R_{KI}^T \, _K r_{SA}$$

$$_I v_{SA} = {_I \dot{r}_{SA}}$$

$$_I \omega_{IK} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_M + \dot{q}_A \end{bmatrix} \qquad _I \omega_{IM} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_M i_g \end{bmatrix}$$

$$_I J_M^S = diag(A_M^S, B_M^S, C_M^S) \qquad _K J_A^S = diag(A_A^S, B_A^S, C_A^S)$$

$$_I L_M = {_I J^S_{MI}} \omega_{IM} \qquad _K L_A = {_K J_A^S} \, _K \omega_{IK}$$

$$_K p_A = m_S \, _K v_{SA}$$

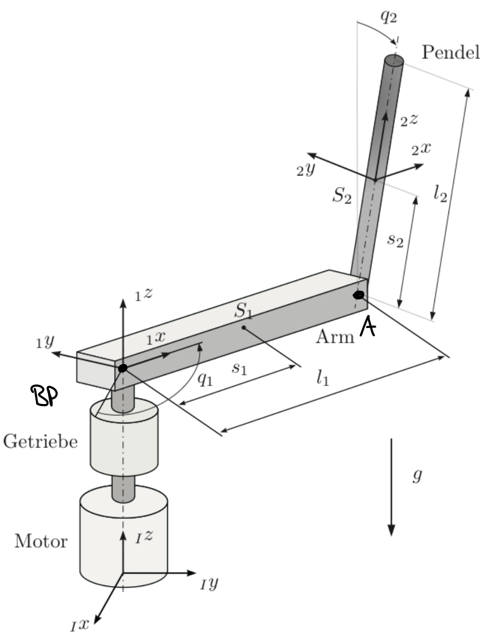
$$f_A^e = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}$$

$$\left.\begin{aligned} M_M &= [0 \;\; 0 \;\; M_{mot}]^T \\ M_{feder,M} &= c \frac{q_A}{i_g} \end{aligned}\right\} M_M^e = M_M + M_{feder,M}$$

$$M_{feder,A} = c\, q_A \quad = M_A^e$$

2.2 Inverses Pendel

Abbildung 2.2 zeigt das Modell eines „Inversen Pendels“ (hier ein Furuta Pendel). Dabei lautet das Regelungsziel, beide Winkel q_1 und q_2 auf 0 zu regeln. Es steht aber nur ein Eingriffsmoment dafür zur Verfügung. Man nennt dies ein unteraktuiertes System, da es weniger Stelleingriffe als Regelgrößen gibt. Bis auf die Getriebeelastizität ist der Aufbau des Antriebsstrangs gleich wie beim Modell „Elastisches Getriebe“. Die Parameter für das Getriebe sind daher gleich wie oben, die zusätzlichen oder veränderten Parameter können der Tabelle unten entnommen werden.



Parameter	Wert
Masse m_1	1.2 kg
Masse m_2	0.0725 kg
Länge l_1	0.3 m
Länge l_2	0.25 m
Schwerpunktsabstand s_1	0.225 m
Schwerpunktsabstand s_2	0.125 m
Trägheitsmoment des Armes C_1^S	0.002 kgm^2

Abbildung 2.2: Inverses Pendel

Der Trägheitstensor des Pendels soll mit dem eines homogenen schlanken Stabes der Länge l_2 angenähert werden, die Trägheit um die Längsachse des Stabs kann dabei vernachlässigt werden ($C^S_{Pendel} = 0$). Das elastische Getriebe wird hier nicht verwendet, daher kann der Motorwinkel direkt mit der Übersetzung aus dem Armwinkel q_1 berechnet werden.

Aufgabe 2.3 *Leiten Sie die (nichtlineare) Bewegungsgleichung des „Inversen Pendels“ mit Hilfe von Lagrange II her. Linearisieren Sie die Bewegungsgleichung um den stationären Punkt $\mathbf{x}_0 = [0\ 0\ 0\ 0]^T$ und stellen Sie die linearisierten Bewegungsgleichungen in der Form*

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ \mathbf{y} &= \mathbf{x}\end{aligned} \tag{2.3}$$

mit dem Zustandsvektor

$$\mathbf{x} = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T \tag{2.4}$$

und dem Eingang $u = M_{Mot}$ dar.

Arm:

$$_1r_{S1} = \begin{bmatrix} s_1 \\ 0 \\ 0 \end{bmatrix} \quad R_{I1} = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 \\ \sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}_I\underbrace{\omega_{I1}}_{S_1} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \quad {}_1J^S = diag(A_1^S, B_1^S, C_1^S)$$

$$_Ir_{S1} = R_{I1}\,{}_1r_{S1} \qquad = {}_1\omega_{S1}$$

$$_Iv_{S1} = {}_I\dot{r}_{S1}$$

$$T_1 = \frac{1}{2}\,{}_Iv_{S1}^T\, m_1\, {}_Iv_{S1} + \frac{1}{2}\,{}_1\omega_{S1}^T\, {}_1J^S\, {}_1\omega_{S1}$$

$$V_1 = -m_1\, {}_I[0 \quad 0 \quad -g]\, {}_Ir_{S1}$$

Pendel:

$$_1r_A = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad {}_2r_{AS2} = \begin{bmatrix} 0 \\ 0 \\ s_2 \end{bmatrix} \quad R_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos q_2 & \sin q_2 \\ 0 & -\sin q_2 & \cos q_2 \end{bmatrix} \quad {}_1r_{S2} = {}_1r_A + R_{21}^T\, {}_2r_{AS2} \quad {}_1\omega_{12} = \begin{bmatrix} \dot{q}_2 \\ 0 \\ 0 \end{bmatrix} \quad {}_I\omega_{12} = R_{I1}\, {}_1\omega_{12}$$

$$_I\omega_{S2} = {}_I\omega_{S1} + {}_I\omega_{12}$$

$${}_2J^s = \mathrm{diag}(A_2^s, B_2^s, 0)$$

$$T_1 = \frac{1}{2}\, {}_I v_{s2}^T\, m_2\, {}_I v_{s2} + \frac{1}{2}\, {}_2\omega_{s2}^T\, {}_2J^s\, {}_2\omega_{s2}$$

$$V_1 = -m_2\, {}_I[0 \quad 0 \quad -g]\, {}_I r_{s2}$$

Motor:

$${}_I\omega_M = \begin{bmatrix} 0 \\ 0 \\ i_g \dot{q}_1 \end{bmatrix} \qquad {}_IJ_M^s = [A_M^s, B_M^s, C_M^s]$$

$$T = \frac{1}{2}\, {}_I\omega_M^T\, {}_IJ_M^s\, {}_I\omega_M$$

Generalisierte Kräfte:

$${}_IM_M^e = \begin{bmatrix} 0 \\ 0 \\ M_{mot} \end{bmatrix} \qquad \left(\frac{\partial\, {}_I\omega_M}{\partial \dot{q}}\right)^T = \begin{bmatrix} 0 & 0 & i_g \\ 0 & 0 & 0 \end{bmatrix} \qquad Q_M = \left(\frac{\partial\, {}_I\omega_M}{\partial \dot{q}}\right)_I^T M_M^e = \begin{bmatrix} i_g M_{mot} \\ 0 \end{bmatrix}$$