5.4 Zusatzaufgaben

5.4.1 Flexibler Roboterarm

Abbildung 5.3 zeigt die Prinzipskizze eines flexiblen Robotorarms, siehe auch [5.1]. Wählt

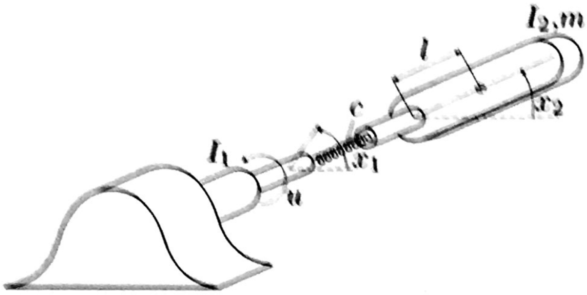

Abbildung 5.3: Einfacher elastisch gekoppelter Roboterarm.

man als Zustandsgrößen die Winkel x_1 und x_2 sowie die Winkelgeschwindigkeiten $\dot{x}_1 = x_3$ und $\dot{x}_2 = x_4$ des Antriebsmotors und des Roboterarmes und als Eingangsgröße das Motormoment u, so erhält man die Bewegungsgleichungen in der Form

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ -\frac{c}{I_1}x_1 + \frac{c}{I_1}x_2 - \frac{d_1}{I_1}x_3 \\ \frac{c}{I_2}x_1 - \frac{c}{I_2}x_2 - \frac{m\,g\,l}{I_2}\cos(x_2) - \frac{d_2}{I_2}x_4 \end{bmatrix}}_{=\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{1}{I_1} \\ 0 \end{bmatrix}}_{=\mathbf{g}(\mathbf{x})} u \,. \tag{5.30}$$

Dabei bezeichnet c die lineare Steifigkeitskonstante der elastischen Kopplung, m die Masse des Roboterarmes, g die Gravitationskonstante, l den Abstand von der Antriebsachse zum Schwerpunkt des Roboterarmes und I_k bzw. d_k, $k \in \{1, 2\}$ beschreiben die Massenträgheitsmomente sowie die viskosen Reibungskonstanten des Antriebsmotors und des Roboterarmes. Die Parameterwerte sind in Tabelle 5.3 gegeben.

Aufgabe 5.14 (Differenzielle Flachheit). Ist das System (5.30) mit dem Ausgang $y = x_2$ differenziell flach? Wenn ja, bestimmen Sie die Parametrierungen der Zustandsgrößen und der Eingangsgröße durch den flachen Ausgang.

Aufgabe 5.15 (Flachheitsbasierte Vorsteuerung). Implementieren Sie eine flachheitsbasierte Vorsteuerung in MATLAB/SIMULINK für den flexiblen Roboterarm.

Parameter	Wert	
c	10.0	N m
d_1	0.5	N m s
d_2	0.5	N m s
m	1.0	kg
g	9.81	m/s^2
l	0.5	m
I_1	1.0	kg m^2
I_2	1.0	kg m^2

Tabelle 5.3: Parameter des elastisch gekoppelten Roboterarms.

Hinweis: Verwenden Sie das zur Verfügung gestellte SIMULINK-Modell `flex_robot_flat.mdl` mit dem Modell der Strecke, welches als `C-code S-Function` realisiert ist (`flex_robot.mexw64`), sowie eine MATLAB m-Datei mit den numerischen Parametern des Systems. Das Modell der Strecke besitzt den Ausgang $\mathbf{y} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$ und startet für $t = 0$ aus der stabilen Ruhelage $x_1 = x_2 = -\pi/2$ und $x_3 = x_4 = 0$. Verwenden Sie den Trajektoriengenerator aus der Aufgabe 5.10 mit dem Startwert $x_{2,S} = -\pi/2$ und dem Endwert $x_{2,E} = \pi/2$, sodass sich ein Arbeitspunktwechsel von der stabilen in die instabile Ruhelage ergibt.

Aufgabe 5.16 (Regelung des Fehlersystems). Erweitern Sie die Vorsteuerung aus Aufgabe 5.15 um einen linearen P-Regler, der das Fehlersystem in der oberen Ruhelage stabilisiert. Schätzen Sie für die gegebenen Systemparameter den Bereich der Reglerverstärkung K ab, damit der geschlossene Kreis stabil ist.

5.4.2 Riementrieb

Damit ein einfacher Entwurf von PI-Reglern zur Trajektorienfolgeregelung wie in Aufgabe 5.7 möglich ist, sind eine Vielzahl an Vereinfachungen nötig. Um diese Vernachlässigungen und die damit verbundenen Probleme zu umgehen, besteht die Möglichkeit einen zeitvarianten LQR-Regler zu entwerfen.

Aufgabe 5.17 (Berechnung des erweiterten Fehlersystems). In dieser Aufgabe wird das in Aufgabe 5.6 betrachtete Fehlersystem für den Fehler $\mathbf{e} = \mathbf{x} - \mathbf{x}_d$ um Integratoren für die Ausgangsgrößen des Systems erweitert. Führen Sie geeignete Zustände $e_{\omega_m,I}$ und $e_{i_d,I}$ ein und bestimmen Sie für eine allgemeine Trajektorie $\mathbf{x}_d$ die Dynamik des erweiterten Fehlersystems mit dem Zustand $\bar{\mathbf{e}}^{\mathrm{T}} = [\mathbf{e}^{\mathrm{T}}, e_{\omega_m,I}, e_{i_d,I}]^{\mathrm{T}}$. Führen Sie anschließend eine Linearisierung um $\bar{\mathbf{e}} = \mathbf{0}$ und die allgemeine Trajektorie $\mathbf{x}_d$ durch.

Für den Entwurf eines zeitdiskreten und zeitvarianten LQR-Reglers wird im Folgenden eine zeitdiskrete Darstellung des erweiterten Fehlersystems benötigt. Berechnen Sie diese in der Form

$$\bar{\mathbf{e}}_{k+1} = \boldsymbol{\Phi}_k \bar{\mathbf{e}}_k + \boldsymbol{\Gamma}_k \mathbf{u}_{e,k} \tag{5.31}$$

mit dem Regelungsanteil $\mathbf{u}_e$ für das Stellgesetz.

Hinweis:

- Eine analytische Berechnung der zeitvarianten Systemmatrizen $\boldsymbol{\Phi}_k$, $\boldsymbol{\Gamma}_k$ ist nicht mehr ohne weiteres möglich. Berechnen Sie diese daher näherungsweise mittels des expliziten Eulerverfahrens in Abhängigkeit der Trajektorie $\mathbf{x}_d$.
- Verwenden Sie für $\mathbf{x}_d$ die anhand der Feedforward-Linearisierung berechneten Verläufe aus Aufgabe 5.4. Mit dem `To Workspace`-Block können Sie die Sollverläufe einfach in den MATLAB Workspace exportieren.

Aufgabe 5.18 (Entwurf eines zeitvarianten LQR-Reglers). Entwerfen Sie einen zeitvarianten LQR-Regler für das zeitvariante zeitdiskrete erweiterte Fehlersystem (5.31) aus der vorigen Aufgabe. Verwenden Sie für die Gewichtung der Stellgrößen und Zustände die konstanten Gewichtungsmatrizen

$$\mathbf{R} = \mathrm{diag}\Big(\begin{bmatrix}1 & 1\end{bmatrix}\Big),\ \mathbf{Q} = \mathrm{diag}\Big(\begin{bmatrix}10 & 1 & 100 & 1 & 1 & 200 & 100\end{bmatrix}\Big) \text{ und } \mathbf{N} = \mathbf{0}. \tag{5.32}$$

Für die Matrix $\mathbf{S}$ der Endzustandsgewichtung $\mathbf{x}_N^T \mathbf{S} \mathbf{x}_N$ empfiehlt es sich, die stationäre Lösung der Riccati-Gleichung für die Systemmatrizen am Ende des betrachteten Horizonts heranzuziehen. Für die Berechnung der Reglerverstärkung $\mathbf{K}_k$ eignet sich somit die folgende Vorgehensweise:

1. Berechnung der zeitvarianten Matrizen $\boldsymbol{\Phi}_k$ und $\boldsymbol{\Gamma}_k$ von (5.31) aus dem zeitlichen Verlauf von $\mathbf{x}_d$.
2. Bestimmung der Matrix $\mathbf{S}$ durch Lösung der stationären Riccati-Gleichung mit den Matrizen $\boldsymbol{\Phi}_N$ und $\boldsymbol{\Gamma}_N$ am Ende des betrachteten Zeithorizonts.
3. Die Berechnung der zeitvarianten Rückführmatrix $\mathbf{K}_k$ erfolgt durch Rückwärts-Iteration der diskreten Riccati-Gleichung für die Matrix $\mathbf{P}_k$. Beachten Sie, dass dieses Vorgehen bis auf die Verwendung der zeitvarianten Systemmatrizen identisch ist zu jener des LQR-Reglers für ein zeitinvariantes System. Speichern Sie die erhaltene zeitvariante LQR-Rückführmatrix $\mathbf{K}_k$ in einer dreidimensionalen „Matrix“ entsprechender Größe ab.

> **Hinweis:** Die Lösung $\mathbf{P}_k$ der Riccati-Gleichung muss in jedem Zeitschritt eine symmetrische Matrix sein. Um trotz numerischer Ungenauigkeiten die Symmetrie zu erhalten, kann als einfachste Lösung die Operation $\mathbf{P}_k \leftarrow \frac{1}{2}\left(\mathbf{P}_k + \mathbf{P}_k^T\right)$ verwendet werden.

Aufgabe 5.19 (Implementierung). Implementieren Sie den entworfenen LQR-Regler in der Simulation. Verwenden Sie dazu eine `MATLAB Function`. Nehmen Sie weiters an, dass alle Zustände des Systems messbar sind.

> **Hinweis:** Verwenden Sie zum Aufruf der zeitvarianten Rückführmatrix $\mathbf{K}_k$ den SIMULINK-Block `Direct Lookup Table (n-D)`.