

3 Modellierung und Simulation eines Roboters

Das Ziel dieser Übung ist die mathematische Modellierung und die Simulation eines Roboters mit serieller Kinematik, des sogenannten KUKA LBR iiwa R820. Hierbei werden aus didaktischen Gründen vier der sieben Drehachsen als starr angesehen und ein 3-Achs-Roboter betrachtet. Dieses Vorgehen erlaubt eine vereinfachte Betrachtung und Interpretation der Ergebnisse.

Die für die Modellierung der Kinematik und der Dynamik von Starrkörpersystemen benötigten theoretischen Grundlagen wurden bereits in der VU Fachvertiefung: Automatisierungs- und Regelungstechnik (WS 2019/20) sowie der VU Modellbildung (SS 2020) vorgestellt und anhand mehrerer Beispiele erläutert. Studieren Sie daher zur Vorbereitung dieser Übung folgende Unterlagen:

- Skriptum zur VU Fachvertiefung: Automatisierungs- und Regelungstechnik (WS 2019/20) [1]
 - Kapitel 2–3, vollständig und Kapitel 4 abschnittsweise
- Skriptum zur VU Modellbildung (SS 2020) [2]
 - Kapitel 4

Beachten Sie zur Lösung der nachfolgenden Aufgabenstellungen die Musterlösungen zu den in diesen Kapiteln angegebenen Beispielen und hier vor allem die auf der Institutshomepage bereitgestellten MATLAB/SIMULINK-Dateien.



Alle Dateien, die zum Bearbeiten dieser Übung benötigt werden, finden Sie in `uebung3.zip` auf der Homepage der Lehrveranstaltung.



Bei Fragen oder Anregungen zu dieser Übung wenden Sie sich bitte an

- Florian Beck <beck@acin.tuwien.ac.at>
- Georg Janisch <janisch@acin.tuwien.ac.at>

3.1 Mathematisches Modell einer seriellen Kinematik

In diesem Abschnitt wird ein 3-Achs-Roboter basierend auf dem KUKA LBR iiwa R820 mathematisch modelliert. Hierzu sind die geometrischen Daten in Tabelle 3.1 gegeben. Des Weiteren ist in Abbildung 3.1 der Roboter und die zur Modellierung verwendeten

Koordinatensysteme abgebildet. Verwenden Sie für die folgenden Aufgaben die bereitgestellte Maple Datei `UE3.mw` und die in den einzelnen Aufgaben beschriebenen Matlab und Simulink Dateien.

Parameter	Wert
d1	0.36 m
d2	0.2045 m
d3	0.2155 m
d4	0.552 m

Tabelle 3.1: Geometrische Daten KUKA LBR iiwa R820

3.2 Vorwärtskinematik

In diesem Kapitel soll die Vorwärtskinematik für den betrachteten 3-Achs-Roboter berechnet und verifiziert werden.

Aufgabe 3.1. Bearbeiten Sie für die Vorwärtskinematik folgende Punkte:

1. Bestimmen Sie die homogene Transformation von allen gegebenen Koordinatensystemen zum jeweils vorhergehenden Koordinatensystem.

Hinweis: Definieren Sie hierzu allgemeine Drehmatrizen und Verschiebungen als homogene Transformationen und verwenden Sie diese unter Zuhilfenahme von Abbildung 3.1.

Überprüfen Sie Ihr Ergebnis anhand der Simscape Visualisierung in der Datei `UE3_Kinematik.slx` in Matlab-Simulink. Beachten Sie, dass Sie für das Simscape-Modell die Transformationen für $\mathbf{q} = \mathbf{0}$ benötigen, da die Gelenkwinkel direkt in Simscape vorgegeben werden. Tragen Sie dazu Ihre berechneten Rotationsmatrizen und Verschiebungsvektoren in der Datei `parameters.m` ein und simulieren Sie das Modell.

Hinweis: Maple Code kann mithilfe des Befehls `Matlab()` exportiert werden.

2. Bestimmen Sie anschließend die homogene Transformation der jeweiligen Koordinatensysteme in das Basiskoordinatensystem. An dieser Stelle ist es sinnvoll auch die Rotationsmatrizen ohne Verschiebungen zu berechnen, da diese später für die Dynamik benötigt werden.

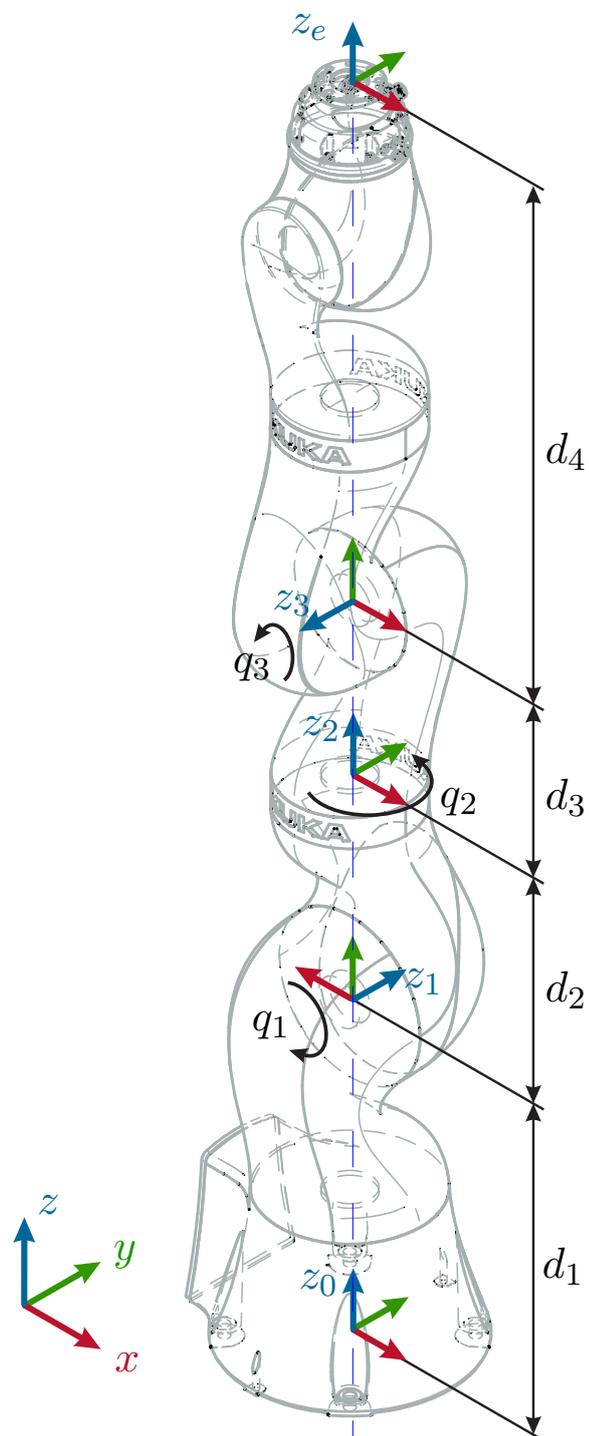


Abbildung 3.1: Koordinatensysteme des KUKA LBR iiwa R820.

Hinweis: Hier besteht die Transformation zwischen zwei Koordinatensystemen aus einer statischen Rotation und Translation und zusätzlich einer Rotation abhängig vom jeweiligen Achswinkel. Diese können durch die Multiplikation (Hintereinanderausführung) homogener Transformationen kombiniert werden.

Durch diese Ausdrücke können die Schwerpunkte der Glieder vom jeweils körperfesten in das Basiskoordinatensystem umgerechnet werden. Die Schwerpunkte in den körperfesten Koordinatensystemen sind wie folgt gegeben:

$$\mathbf{s}_1 = \begin{bmatrix} 0.3 \text{ mm} \\ 59 \text{ mm} \\ 42 \text{ mm} \end{bmatrix} \quad \mathbf{s}_2 = \begin{bmatrix} 0 \text{ mm} \\ 30 \text{ mm} \\ 130 \text{ mm} \end{bmatrix} \quad \mathbf{s}_3 = \begin{bmatrix} 0 \text{ mm} \\ 230 \text{ mm} \\ 19.5 \text{ mm} \end{bmatrix} \quad (3.1)$$

Aufgabe 3.2. Ermitteln Sie die Vektoren vom Ursprung des Basiskoordinatensystems zu den Schwerpunkten der Glieder des Roboters gegeben im Basiskoordinatensystem.

Nachfolgend wird die berechnete Vorwärtskinematik auf ihre Plausibilität geprüft. Dazu werden Konfigurationen verwendet, die ein leicht interpretierbares Ergebnis im Ausgangsraum bedingen. So nimmt die Rotationsmatrix der Vorwärtskinematik die Einheitsmatrix an, wenn das Endeffektor-Koordinatensystem mit dem Basiskoordinatensystem gleich orientiert ist. Auch 90° Drehungen mit den zugehörigen Vertauschungen der Achsrichtungen sind leicht anhand ihrer Drehmatrizen interpretierbar. Zur Prüfung von Positionen bieten sich ausgestreckte Lagen und einfache abgewinkelte Stellungen an. Hier treten Summen beziehungsweise einzelne Verschiebungsparameter in den Homogenen Transformationen auf. Überprüfen Sie demnach die folgenden Konfigurationen.

Aufgabe 3.3. Prüfen Sie Ihre Vorwärtskinematik (Positionen und Orientierungen) für folgende Konfigurationen:

$$\mathbf{q}_1 = \begin{bmatrix} \frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix}, \quad (3.2)$$

$$\mathbf{q}_2 = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix}, \quad (3.3)$$

$$\mathbf{q}_3 = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}. \quad (3.4)$$

Überlegen Sie sich zusätzlich 3 Konfigurationen und die zugehörigen Endeffektorposen und überprüfen Sie auch diese. Begründen Sie die Plausibilität Ihrer Ergebnisse.

3.3 Differenzielle Kinematik

Es stellt sich heraus, dass es sehr nützlich ist zu wissen, wie die Drehachsen gedreht werden müssen, um den Endeffektor im kartesischen Raum in eine vorgegebene Richtung zu bewegen. Dieser Zusammenhang wird als differenzielle Kinematik bezeichnet. Kernelement dieser ist die Jacobi-Matrix des Endeffektors beziehungsweise des Werkzeugs.

Aufgabe 3.4. Bestimmen Sie die translatorische Manipulator Jacobi-Matrix des Endeffektors. Berechnen Sie ferner die rotatorische Manipulator Jacobi-Matrix.

Hinweis: Der rotatorische Teil kann über die Drehmatrizen und den Vektor der Drehwinkelgeschwindigkeiten gefunden werden.

Hiermit können die Achs-Winkelgeschwindigkeiten mit den kartesischen Geschwindigkeiten des Endeffektors in Verbindung gebracht werden.

Aufgabe 3.5. Berechnen Sie die nötigen Geschwindigkeiten der Drehachsen, sodass der Endeffektor in der Konfiguration

$$\mathbf{q} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{4} \\ -\frac{\pi}{4} \end{bmatrix} \quad (3.5)$$

eine Geschwindigkeit von 1 m/s in die Richtung

$$\mathbf{v} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \quad (3.6)$$

erreicht.

Hinweis: Um in Maple Ausdrücke numerisch auszuwerten verwenden Sie den Befehl `evalf()`.

3.4 Dynamik

Mithilfe der bisher berechneten Beziehungen lassen sich die dynamischen Bewegungsgleichungen des Roboters bestimmen. Die Bewegungsgleichungen mechanischer Systeme lassen sich immer in der Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (3.7)$$

darstellen. Die generalisierten Kräfte $\boldsymbol{\tau} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}$ setzen sich hier aus den dissipativen Kräften und Momenten $\boldsymbol{\tau}_d$, die Stelleingänge $\boldsymbol{\tau}_c$ und den externen Kräften und Momenten $\boldsymbol{\tau}_{ext}$ zusammen. Diese Darstellung wird in der Robotik häufig verwendet, da wichtige Eigenschaften mechanischer Systeme besonders deutlich ablesbar sind. Gleichsam ist

die Formulierung von Regelungsstrategien in dieser Form besonders einfach. Die darin auftretenden Terme sollen im Folgenden bestimmt werden.

Aufgabe 3.6. Bestimmen Sie den Vektor der äußeren generalisierten Kräfte $\boldsymbol{\tau}$, die auf die generalisierten Koordinaten wirken. Nehmen Sie an, dass keine externen Kräfte auftreten, d.h. $\boldsymbol{\tau}_{ext} = 0$. Außerdem werden an dieser Stelle die dissipativen Kräfte vernachlässigt.

Hinweis: Hier sei angemerkt, dass auf ein freigeschnittenes Glied zwei Drehmomente wirken. Achten Sie dabei auf die jeweiligen Koordinatensysteme.

Weiters können die, die Dynamik beschreibenden, Matrizen direkt aus den zuvor bestimmten Größen berechnet werden.

Aufgabe 3.7. Bestimmen Sie die Massenmatrix $\mathbf{M}(\mathbf{q})$, die Matrix der Coriolis- und Zentripetal-Terme $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})$, sowie den Vektor der Potenzialterme $\mathbf{g}(\mathbf{q})$ auf Basis der jeweiligen Definitionen im Kapitel Bewegungsgleichungen des Vorlesungsskriptums [1]. Überprüfen Sie die folgenden Zusammenhänge:

$$\dot{\mathbf{M}} = \mathbf{C} + \mathbf{C}^T, \quad (3.8)$$

sowie die Symmetrie Ihrer berechneten Massenmatrix

$$\mathbf{M} = \mathbf{M}^T. \quad (3.9)$$

Hinweis: Verwenden Sie zum Vergleich von Matrizen in Maple den Befehl `simplify()`.

Aufgabe 3.8. Überprüfen Sie Ihr Modell in Matlab-Simulink. Die benötigten Parameter sind in der Matlab Datei `parameters.m` gegeben und in Tabelle 3.2 aufgelistet. Beachten Sie, dass die Trägheitsmatrizen als diagonal angenommen werden. Bearbeiten Sie dazu folgende Punkte:

1. Exportieren Sie die berechneten Matrizen \mathbf{M} , \mathbf{C} und \mathbf{g} mithilfe der gegebenen Export Funktion nach Matlab-Simulink.
2. Formen Sie Gleichung (3.7) so um, dass das Modell in Zustandsraumdarstellung der Form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ vorliegt. Als Zustand soll $\mathbf{x} = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T$ gewählt werden. Als Eingang \mathbf{u} sollen die Motordrehmomente $\boldsymbol{\tau}_c$ verwendet werden.
3. Implementieren Sie Ihr Modell als Matlab function in der Datei `UE3_Modellvergleich.slx` mit anschließendem Integratorblock in Simulink.
4. Vergleichen Sie das von Ihnen exportierte Modell mit dem als `p-function` gegebenen Modell.

Parameter	Wert
m_1	4 kg
m_2	3 kg
m_3	6.5 kg
$I_{1,xx}$	0.05 kgm ²
$I_{1,yy}$	0.018 kgm ²
$I_{1,zz}$	0.044 kgm ²
$I_{2,xx}$	0.08 kgm ²
$I_{2,yy}$	0.075 kgm ²
$I_{2,zz}$	0.01 kgm ²
$I_{3,xx}$	0.2091 kgm ²
$I_{3,yy}$	0.0247 kgm ²
$I_{3,zz}$	0.2030 kgm ²

Tabelle 3.2: Parameter für das dynamische Modell

3.5 Belastung am Endeffektor

Nehmen Sie an, es wird in der Konfiguration

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix} \quad (3.10)$$

eine Lastkraft von 9.81 N in negative z -Richtung am Endeffektor aufgebracht.

Aufgabe 3.9. Bestimmen Sie die nötigen Drehmomente um die Last in dieser Konfiguration zu halten. Prüfen Sie Ihr Ergebnis mit dem gegebenen belasteten Modell UE3_Belastung.slx in dem Sie die berechneten Drehmomente aufschalten.

Hinweis: Betrachten Sie die Kraft als externe Kraft und nicht als Teil des Robotermodells.

3.6 Singuläre Konfigurationen

Der 3-Achs-Roboter soll im Folgenden zur Positionierung eingesetzt werden. Dazu wird die Kartesische Position des Endeffektors \mathbf{x}_e im Basiskoordinatensystem als Ausgangsgröße (vgl. Automatisierung) gewählt.

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \quad (3.11)$$

Mit der Vorwärtskinematik $\mathbf{f}(\mathbf{q})$. Weiters gilt als Zusammenhang zwischen den kartesischen Geschwindigkeiten und den Achs-Winkelgeschwindigkeiten

$$\dot{\mathbf{x}}_e = \mathbf{J}_v(\mathbf{q})\dot{\mathbf{q}} \quad (3.12)$$

mit der translatorischen Manipulator Jacobi-Matrix \mathbf{J}_v .

Singuläre Konfigurationen definieren für serielle Manipulatoren zwei ausgezeichnete Richtungen. Eine Richtung im Geschwindigkeitsraum der generalisierten Koordinaten, die angibt welche Achs-Winkelgeschwindigkeiten zu keinen Ausgangsgeschwindigkeiten führen. Des Weiteren gibt es eine Richtung im Ausgangsraum (hier dem kartesischen Raum) aus der externe Kräfte nicht durch die Drehachsen aufgenommen werden können. Dadurch bleibt der Roboter bei äußeren Belastungen aus dieser Richtung steif und führt die Belastungen über seine Konstruktion in die Verankerung ab. Gleichzeitig kann der Roboter nicht in diese Richtungen bewegt werden, da keiner der Motoren zur Aktuierung dieser Richtung führt. Zunächst soll nur die Position als Ausgangsgröße betrachtet werden.

Aufgabe 3.10. Bestimmen Sie die singulären Konfigurationen des Roboters. Berechnen Sie die Jacobi-Matrix in allen singulären Konfigurationen und berechnen Sie die Achs-Winkelgeschwindigkeiten, für die keine kartesische Geschwindigkeit des Endeffektors auftritt.

Zeigen sie, aus welchen Richtungen keine äußeren Kräfte durch die Drehachsen aufgenommen werden können. Interpretieren Sie die singuläre Konfiguration und die berechneten Richtungen anhand der Visualisierung in Matlab-Simulink.

Hinweis: Überlegen Sie sich hierzu, wie allgemeine äußere Kräfte am Endeffektor auf die Drehachsen wirken.

In der Praxis werden die singulären Konfigurationen eines Roboters durch geeignete Pfadplanung vermieden. Es kommt jedoch durch die schlechte Konditionierung der Jacobi-Matrix auch in der Umgebung von Singularitäten zu unerwünschten Effekten.

Aufgabe 3.11. Berechnen Sie eine Konfiguration, bei der die Achsen 5° in beliebiger Richtung aus einer singulären Stellung verdreht sind. Bestimmen Sie nun die Achs-Winkelgeschwindigkeiten um den Endeffektor mit 1 m/s in die zur Singularität gehörige blockierte Richtung zu bewegen. Vergleichen Sie Ihr Ergebnis mit dem zuvor Errechneten.

Auch die Orientierung des Roboters ist in vielen Anwendungen entscheidend. Zur weiteren Analyse wird die Orientierung des Endeffektors bezüglich des Basiskoordinatensystems als Ausgangsgröße gewählt.

Hinweis: Die Orientierung des Endeffektors muss an dieser Stelle nicht explizit berechnet werden. Daher ist keine Parametrierung der Orientierung (zB. Rotationsmatrix, Roll-Pitch-Yaw, Quaternionen) nötig. Arbeiten Sie also mit dem rotatorischen Teil der geometrischen Jacobi-Matrix.

Exemplarisch soll nun mit den drei Achsen des Roboters nur die Orientierung untersucht werden, wobei die Position des Roboters vernachlässigt wird.

Aufgabe 3.12. Berechnen Sie die Konfigurationen, die zu rotatorischen Singularitäten führen. Berechnen Sie auch hier die Richtung der Drehachse, um die Momente nicht durch die Drehachsen aufgenommen und in die keine Drehungen herbeigeführt werden können. Interpretieren Sie Ihr Ergebnis.

3.7 Inverse Kinematik

Bei seriellen Robotern ist im Allgemeinen die Berechnung der Vorwärtskinematik, also das Bestimmen der Endeffektor Pose $\mathbf{x}_e = \mathbf{f}(\mathbf{q})$ in Abhängigkeit der Gelenkwinkel systematisch lösbar. Die inverse Aufgabe, also das Finden der nötigen Achswinkel zum Erreichen einer gegebenen Endeffektor-Pose $\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_e)$ ist jedoch im Allgemeinen nicht eindeutig lösbar und muss für jeden Mechanismus individuell bestimmt werden. Bei einem 6-Achs-Roboter zum Beispiel können im Allgemeinen bis zu 16 Konfigurationen für eine Pose gefunden werden, die dann durch Achs-Limits und die gewünschte "Ellenbogen"-Stellung ausgewählt werden müssen. Zur Lösung der inversen Kinematik gibt es analytische und numerische Ansätze.

Analytische Methoden machen sich die Geometrie des Roboters zu Nutze um das inverse Kinematikproblem zu lösen. Die daraus resultierenden Lösungen sind oft hoch komplex. Ein häufig zielführendes Vorgehen, das an dieser Stelle erwähnt werden soll ist die Trennung der Orientierung und Position beim Lösen des Problems. Dazu werden serielle Roboter oft mit einer sogenannten *Zentralhand* gebaut, also drei sich schneidende Achsen am Ende der kinematischen Kette (vgl. Handgelenk des Menschen). Dann wird durch das Handgelenk die gewünschte Orientierung eingenommen und durch den Ellenbogen und die verbleibenden Achsen die Position des TCP festgelegt.

Numerische Ansätze suchen eine Lösung des Problems mithilfe von iterativen mathematischen Methoden. Häufig wird dazu ein Optimierungsproblem gelöst, welches eine optimale Konfiguration im Sinne einer Kostenfunktion unter Gelenkwinkelbeschränkungen liefert. Diese Ansätze verwenden größtenteils die Jacobi-Matrix, da diese den Gradienten der Endeffektor-Pose bezüglich der Achswinkel darstellt.

In der Vorlesung zur Fachvertiefung wurde ein numerisches Verfahren mittels vollständigem Beobachterentwurf vorgestellt, welches hier verwendet werden soll um eine Endeffektor-Trajektorie im Arbeitsraum in eine Gelenks-Trajektorie im Konfigurationsraum zu übersetzen. Als Trajektorie im Arbeitsraum ist ein Kreis in der yz -Ebene im Endeffektor Koordinatensystem gegeben, welcher folgendermaßen parametrisiert ist:

$$\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 + r \sin(t) \\ z_0 + r \cos(t) \end{bmatrix}. \quad (3.13)$$

Nachfolgend soll nur die Endeffektor-Position als Ausgang

$$\mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad (3.14)$$

betrachtet werden, die Orientierung wird vernachlässigt. Dadurch ergibt sich $m = \dim(\mathbf{x}_e) = 3$ und $n = \dim(\mathbf{q}) = 3$. Die Gleichungen für den vollständigen Beobachter sind daher

$$\frac{d}{dt}\hat{\mathbf{q}}_d = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\hat{\mathbf{e}}_x) = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)(\dot{\mathbf{x}}_{e,d} + \mathbf{K}(\mathbf{x}_{e,d} - \mathbf{f}(\hat{\mathbf{q}}_d))) \quad (3.15a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0}, \quad (3.15b)$$

mit der gewünschten Endeffektorposition $\mathbf{x}_{e,d}$.

Aufgabe 3.13. Bestimmen Sie die inverse Kinematik für die Arbeitsraumtrajektorie (3.13) mithilfe des vollständigen Beobachters in Gleichung (3.15). Verwenden Sie die Datei `UE3_inv_Kinematik.slx`. Die Trajektorie kann mithilfe des Simscape Modells visualisiert werden. Bearbeiten Sie dazu folgende Punkte:

1. Berechnen Sie die analytische Jacobi-Matrix \mathbf{J}_A in Maple und exportieren Sie diese nach Matlab.
2. Implementieren Sie den vollständigen Beobachter in Gleichung (3.15) mithilfe einer Matlab function.
3. Implementieren Sie die Trajektorie (3.13) und deren Ableitung ebenfalls als Matlab function. Die Berechnung der Trajektorie soll in Abhängigkeit der Simulationszeit erfolgen. Die Kreisparameter können mit $x_0 = 0.6707$, $y_0 = 0$, $z_0 = 0.9593$ und $r = 0.2$ gewählt werden, welche einer Startkonfiguration von $\mathbf{q}_0 = [\pi/8, 0, -\pi/4]^T$ entsprechen.
4. Wählen Sie geeignete Werte für die Matrix \mathbf{K} für ein stabiles Beobachterverhalten.
5. Untersuchen Sie, was passiert wenn eine andere Startkonfiguration gewählt wird bzw. wenn Sie die Parameter z_0 und y_0 verändern. Bleibt der Beobachter stabil? Begründen Sie Ihre Antwort.

In der nächsten Übung werden Methoden präsentiert um so gewonnene Trajektorien für den gesamten Roboter zu regeln.

3.8 Identifikation der Gelenksreibung

Zur Erreichung größtmöglicher Genauigkeit sind in Robotern spielfreie Getriebe verbaut. Dadurch entsteht in den Gelenken ein großes Reibmoment. Dieses soll im Folgenden anhand von Messungen identifiziert werden. Für derartige Reibmessungen wird mittels Geschwindigkeitsregler die Winkelgeschwindigkeit auf einen konstanten Wert geregelt und das anliegende Motordrehmoment gemessen. Aufgrund der konstanten Winkelgeschwindigkeit erwartet man, dass das Motordrehmoment für ein ideales Gelenk verschwindet, da nicht beschleunigt wird. Das trotzdem gemessene Drehmoment entspricht genau dem dissipativen Anteil τ_r , welcher durch Reibung verursacht wird. Dabei ist darauf zu Achten,

dass die Achse senkrecht zur Erdbeschleunigung steht, oder Gravitations-kompensiert ist.

Aufgabe 3.14. Gegeben ist eine Messreihe dieses Drehmoments für die 4. Achse des Roboters und der zugehörigen Gelenksgeschwindigkeiten in der Datei `Reibmessung.mat`. Plotten Sie das gemessene Reibmoment τ_r als Funktion der Winkelgeschwindigkeit \dot{q} .

Im Weiteren soll dieses Reibmoment für eine Achse identifiziert und kompensiert werden. Ein einfaches Modell mit dem dieses Reibmoment durch einen Coulomb- und einen viskosen Reibanteil parametrisiert werden kann ist gegeben durch

$$\tau_r = -\text{sign}(\dot{q})k_c + k_v\dot{q} . \quad (3.16)$$

Die Konstanten k_c und k_v können anhand der Messungen durch das Prinzip der kleinsten Fehlerquadrate bestimmt werden. Dazu werden die N Messungen des Reibmoments $\tau_{r,k}$ mit $k = 1, 2, \dots, N$ und der der Winkelgeschwindigkeit \dot{q}_k in (3.16) eingesetzt. Man erhält ein überbestimmtes Gleichungssystem zur Bestimmung der Parameter in der Form

$$\mathbf{S}\mathbf{p} = \mathbf{y} \quad (3.17)$$

mit der sogenannten Regressormatrix

$$\mathbf{S} = \begin{bmatrix} -\text{sign}(\dot{q}_1) & \dot{q}_1 \\ -\text{sign}(\dot{q}_2) & \dot{q}_2 \\ \vdots & \vdots \\ -\text{sign}(\dot{q}_N) & \dot{q}_N \end{bmatrix} \quad (3.18)$$

dem Parametervektor

$$\mathbf{p} = \begin{bmatrix} k_c \\ k_v \end{bmatrix} \quad (3.19)$$

sowie dem Messvektor

$$\mathbf{y} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_N \end{bmatrix} . \quad (3.20)$$

Im Allgemeinen ist es für $N > 2$ nicht möglich, die Parameter k_c und k_v eindeutig aus diesen N Gleichungen zu bestimmen. Daher führt man den Fehler

$$\mathbf{e} = \mathbf{S}\mathbf{p} - \mathbf{y} \quad (3.21)$$

ein und fordert, dass der Parametervektor \mathbf{p} so bestimmt wird, dass der quadratische Fehler minimal wird, d.h.

$$(\mathbf{S}\mathbf{p} - \mathbf{y})^T(\mathbf{S}\mathbf{p} - \mathbf{y}) \rightarrow \min. \quad (3.22)$$

Dies führt auf die Bedingung, dass die Ableitung des quadratischen Fehlers nach dem Parameter gleich 0 ist

$$\frac{d}{d\mathbf{p}} \left((\mathbf{S}\mathbf{p} - \mathbf{y})^T(\mathbf{S}\mathbf{p} - \mathbf{y}) \right) = 2\mathbf{S}^T\mathbf{S}\mathbf{p} - 2\mathbf{S}^T\mathbf{y} = \mathbf{0}. \quad (3.23)$$

Der optimale Parametervektor \mathbf{p} im Sinne des kleinsten quadratischen Fehlers \mathbf{e} ist damit

$$\mathbf{p} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{y} . \quad (3.24)$$

Aufgabe 3.15. Verwenden Sie die Methode der kleinsten Fehlerquadrate um die Parameter k_c und k_v zu bestimmen. Überlegen Sie sich weiters, wie dieses Modell verwendet werden kann, um die Reibung der Gelenke durch eine geeignete Vorsteuerung auszugleichen.

Hinweis: Es ist wichtig zu erwähnen, dass das beschriebene Vorgehen richtig und lehrreich ist. In der Praxis sollen derartige Gleichungssysteme jedoch mit dem Backslash-Operator in Matlab in der Form

$$\mathbf{p} = \mathbf{S} \backslash \mathbf{y};$$

gelöst werden. Hierbei wird automatisch der effizienteste und genauesten Algorithmus ausgewählt und es kann auch die rechte Seite zur Lösung mittels Substitution herangezogen werden.

3.9 Literatur

- [1] A. Kugi und W. Kemmetmüller, *Skriptum zur VU Fachvertiefung Automatisierungs- und Regelungstechnik (WS 2018/2019)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2018.
- [2] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2019)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2019.