

# 3 Modellierung und Simulation eines Roboters

Das Ziel dieser Übung ist die mathematische Modellierung und die Simulation des Roboters KUKA LBR iiwa 14 R820, eines kollaborativen Industrieroboter mit serieller Kinematik. Hierbei werden aus didaktischen Gründen vier der sieben Drehachsen als starr angesehen und der Roboter damit auf einen 3-Achs-Roboter reduziert. 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 2020/21) sowie der VU Modellbildung (SS 2021) vorgestellt und anhand verschiedener Beispiele erläutert. Studieren Sie daher zur Vorbereitung dieser Übung folgende Unterlagen:

- Skriptum zur VU Fachvertiefung: Automatisierungs- und Regelungstechnik (WS 2020/21) [1]
  - Kapitel 1 und 2
- Skriptum zur VU Modellbildung (SS 2021) [2]
  - Kapitel 4
  - Kapitel 5

Beachten Sie insbesondere auch die Musterlösungen zu den Beispielen in den Skripten und 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

- Maderthaler Jakob <[maderthaler@acin.tuwien.ac.at](mailto:maderthaler@acin.tuwien.ac.at)>
- Schwegel Michael <[schwegel@acin.tuwien.ac.at](mailto:schwegel@acin.tuwien.ac.at)>

## 3.1 Mathematisches Modell eines 3-Achs-Roboters

In diesem Abschnitt wird ein 3-Achs-Roboter basierend auf dem KUKA LBR iiwa 14 R820 mathematisch modelliert. Die Freiheitsgrade des Roboters werden mit  $\mathbf{q}^T = [q_1 \ q_2 \ q_3]$  bezeichnet. Der Roboter und die zur Modellierung verwendeten Koordinatensysteme

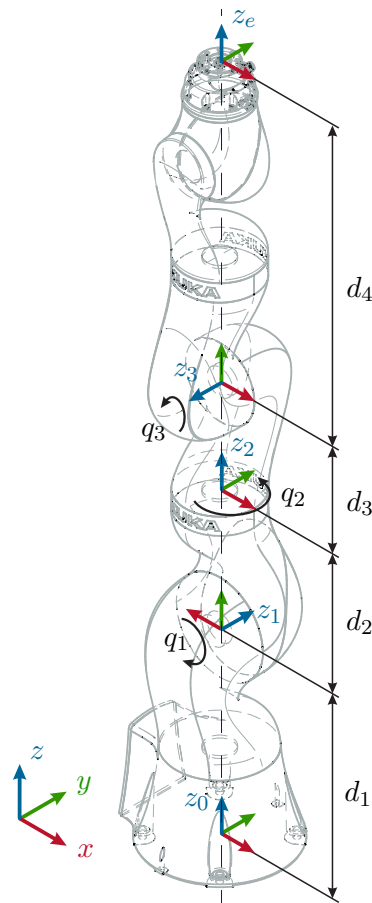


Abbildung 3.1: Koordinatensysteme des Roboters KUKA LBR iiwa 14 R820.

sind in Abbildung 3.1 dargestellt. Die geometrischen Daten sind in Tabelle 3.1 gegeben. Verwenden Sie für die folgenden Aufgaben die bereitgestellte MAPLE-Datei `UE3.mw` und die in den einzelnen Aufgaben beschriebenen MATLAB- und SIMULINK-Dateien.

### 3.2 Vorwärtskinematik

In diesem Kapitel soll die Vorwärtskinematik  $\mathbf{H}_0^e(\mathbf{q})$  gemäß

$$\mathbf{H}_0^e(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^e(\mathbf{q}) & \mathbf{d}_0^e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.1)$$

für den betrachteten 3-Achs-Roboter berechnet und verifiziert werden.

**Aufgabe 3.1.** Bearbeiten Sie für die Vorwärtskinematik folgende Punkte:

1. Bestimmen Sie die homogenen Transformationen  $\mathbf{H}_{i-1}^i(q_i)$ ,  $i = \{1, 2, 3\}$ , sowie  $\mathbf{H}_3^e$ , die die geometrischen Zusammenhänge der Koordinatensysteme des

| Parameter | Wert     |
|-----------|----------|
| $d_1$     | 0.36 m   |
| $d_2$     | 0.2045 m |
| $d_3$     | 0.2155 m |
| $d_4$     | 0.552 m  |

Tabelle 3.1: Geometrische Daten des Roboters KUKA LBR iiwa 14 R820

Roboters beschreiben. Überprüfen Sie Ihr Ergebnis anhand der SIMSCAPE-Visualisierung im MATLAB/SIMULINK-Modell `UE3_Kinematik.slx`. Tragen Sie dazu Ihre berechneten Rotationsmatrizen,  $\mathbf{R}_{i-1}^i(0)$ ,  $i = \{1, 2, 3\}$  und  $\mathbf{R}_3^e(0)$ , und die Verschiebungsvektoren,  $\mathbf{d}_{i-1}^i$ ,  $i = \{1, 2, 3\}$  und  $\mathbf{d}_3^e$ , in der MATLAB-Datei `parameters.m` ein und simulieren Sie das Modell.

**Hinweis:** Beachten Sie, dass Sie für das SIMSCAPE-Modell die Rotationen,  $\mathbf{R}_{i-1}^i(0)$ ,  $i = \{1, 2, 3\}$  und  $\mathbf{R}_3^e(0)$ , benötigen, da die Gelenkwinkel  $\mathbf{q}$  direkt in SIMSCAPE vorgegeben werden.

**Hinweis:** Die analytischen Ergebnisse in MAPLE können mithilfe des Befehls `Matlab(...)` als MATLAB-Code exportiert werden.

**Hinweis:** SIMULINK ruft beim jedem Start der Simulation automatisch die MATLAB-Datei `parameters.m` auf.

- Bestimmen Sie anschließend die homogenen Transformationen  $\mathbf{H}_0^i(\mathbf{q})$ ,  $i = \{1, 2, 3, e\}$ , die die jeweiligen Koordinatensysteme  $(0_i x_i y_i z_i)$  im Bezug auf das Roboterbasis-Koordinatensystem  $(0_0 x_0 y_0 z_0)$  darstellen. Berechnen Sie weiters auch die Rotationsmatrizen  $\mathbf{R}_0^i(\mathbf{q})$  und die Verschiebungsvektoren der Koordinatenursprünge  $\mathbf{d}_0^i(\mathbf{q})$ ,  $i = \{1, 2, 3, e\}$ , welche später für die Dynamik benötigt werden.

**Aufgabe 3.2.** Die Schwerpunkte  $\mathbf{s}_i$ ,  $i = 1, 2, 3$ , der Roboterglieder  $i$  sind in den körperfesten Koordinatensystemen  $(0_i x_i y_i z_i)$  gemäß

$$\mathbf{s}_1 = \begin{bmatrix} 0.3 \text{ mm} \\ 59 \text{ mm} \\ 42 \text{ mm} \end{bmatrix}, \quad \mathbf{s}_2 = \begin{bmatrix} 0 \text{ mm} \\ 30 \text{ mm} \\ 130 \text{ mm} \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} 0 \text{ mm} \\ 230 \text{ mm} \\ 19.5 \text{ mm} \end{bmatrix} \quad (3.2)$$

gegeben. Berechnen Sie die Positionen der Schwerpunkte  $\mathbf{s}_{i,0}$ ,  $i = 1, 2, 3$ , im Roboterbasis-Koordinatensystem  $(0_0 x_0 y_0 z_0)$ .

Nachfolgend wird die berechnete Vorwärtskinematik  $\mathbf{H}_0^e(\mathbf{q})$  auf Plausibilität geprüft. Für

diese Überprüfung werden unterschiedliche Konfigurationen  $\mathbf{q}$  in  $\mathbf{H}_0^e(\mathbf{q})$  eingesetzt, welche ein leicht interpretierbares Ergebnis im Ausgangsraum ergeben. Beispielsweise ist die Rotationsmatrix  $\mathbf{R}_0^e(\mathbf{q})$  der Vorwärtskinematik  $\mathbf{H}_0^e(\mathbf{q})$  eine Einheitsmatrix, wenn das Endeffektor-Koordinatensystem  $(0_e x_e y_e z_e)$  gleich dem Roboterbasis-Koordinatensystem  $(0_0 x_0 y_0 z_0)$  orientiert ist. Weiters führen  $90^\circ$ -Drehungen der Robotergelenke  $q_i$ ,  $i = 1, 2, 3$ , dazu, dass die Einheitsvektoren der Rotationsmatrix  $\mathbf{R}_0^e(\mathbf{q})$  wieder parallel oder antiparallel zu den Achsen des Roboterbasis-Koordinatensystems liegen. Zur Prüfung von Positionen  $\mathbf{d}_0^e(\mathbf{q})$  in der Vorwärtskinematik  $\mathbf{H}_0^e(\mathbf{q})$  sind ausgestreckte Lagen und einfache abgewinkelte Konfigurationen geeignet. In diesem Fall treten Summen bzw. Differenzen der Längen der Glieder  $d_i$ ,  $i = 1, \dots, 4$  in der Vorwärtskinematik auf.

**Aufgabe 3.3.** Prüfen Sie die Positionen und Orientierungen Ihrer Vorwärtskinematik  $\mathbf{H}_0^e(\mathbf{q})$  für die Konfigurationen

$$\mathbf{q}_1 = \begin{bmatrix} \frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix}, \quad \mathbf{q}_3 = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}. \quad (3.3)$$

Überlegen Sie sich zusätzlich drei Konfigurationen,  $\mathbf{q}_4$ ,  $\mathbf{q}_5$  und  $\mathbf{q}_6$ , und die zugehörigen Endeffektorposen und überprüfen Sie auch diese. Begründen Sie die Plausibilität Ihrer Ergebnisse anhand Ihrer Berechnungen.

### 3.3 Differenzielle Kinematik

**Aufgabe 3.4.** Bestimmen Sie die geometrische Manipulator Jacobi-Matrix des Endeffektors für translatorische und rotatorische Bewegungen  $(\mathbf{J}_v)_0^e(\mathbf{q})$  bzw.  $(\mathbf{J}_\omega)_0^e(\mathbf{q})$ .

**Hinweis:** Der rotatorische Teil  $(\mathbf{J}_\omega)_0^e(\mathbf{q})$  kann über die Drehmatrix  $\mathbf{R}_0^e(\mathbf{q})$  und den Vektor der Drehwinkelgeschwindigkeiten  $\omega_0^e(\mathbf{q})$  gefunden werden.

Mit der geometrischen Manipulator Jacobi-Matrix können die Winkelgeschwindigkeiten der Achsen  $\dot{\mathbf{q}}$  mit den kartesischen Geschwindigkeiten des Endeffektors  $\dot{\mathbf{x}}_e$  in Verbindung gebracht werden.

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

$$\mathbf{q}^T = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{4} & -\frac{\pi}{4} \end{bmatrix} \quad (3.4)$$

eine translatorische Geschwindigkeit  $\dot{\mathbf{x}}_e$  von 1 m/s in die Richtung

$$\mathbf{v}^T = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \quad (3.5)$$

erreicht.

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

### 3.4 Singuläre Konfigurationen

In diesem Abschnitt werden die singulären Konfigurationen des 3-Achs-Roboters untersucht. Zunächst soll der Roboter im Folgenden zur reinen Positionierung eingesetzt werden. Dazu wird die kartesische Position  $\mathbf{d}_0^e(\mathbf{q})$  des Ursprungs des Endeffektor-Koordinatensystems  $(0_e x_e y_e z_e)$  relativ zum Roboterbasis-Koordinatensystem  $(0_0 x_0 y_0 z_0)$  als Ausgangsgröße gewählt, siehe [3]. D. h. es gilt

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \mathbf{d}_0^e(\mathbf{q}), \quad (3.6)$$

mit der Vorwärtskinematik  $\mathbf{f}(\mathbf{q})$ . Weiters gilt der Zusammenhang zwischen den Geschwindigkeiten des gewählten Ausgangs und den Achs-Geschwindigkeiten im Konfigurationsraum

$$\dot{\mathbf{x}}_e = (\mathbf{J}_v)_0^e(\mathbf{q})\dot{\mathbf{q}} \quad (3.7)$$

mit dem translatorischen Teil der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$ .

Bei seriellen Robotern können zu jeder singulären Konfiguration zwei ausgezeichnete Vektoren angegeben werden. Der erste Vektor  $\dot{\mathbf{q}}_0$  gibt Winkelgeschwindigkeiten im Konfigurationsraum an, welche zu keiner Bewegung im Arbeitsraum führen, d. h.  $\dot{\mathbf{x}}_e = \mathbf{0}$ . Der zweite Vektor  $\mathbf{e}_0$  gibt eine Richtung im Arbeitsraum an, 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.

*Aufgabe 3.6.* Bestimmen Sie die Menge der singulären Konfigurationen (kinematisch)  $Q_0$  des Roboters für den kartesischen Positionsausgang (3.6) und (3.7). Berechnen Sie die Jacobi-Matrix  $(\mathbf{J}_v)_0^e(\mathbf{q})$  in allen singulären Konfigurationen  $\mathbf{q} \in Q_0$  und berechnen Sie jene Geschwindigkeiten im Konfigurationsraum  $\dot{\mathbf{q}}_0$ , für die keine kartesische Geschwindigkeit des Endeffektors erzeugt werden kann. Zeigen sie weiters, aus welchen Richtungen  $\mathbf{e}_0$  keine äußeren Kräfte durch die Drehachsen aufgenommen werden können (Kräfte, die die aktuelle Pose nicht verändern). Geben Sie diese blockierten Richtungen im Roboterbasis-Koordinatensystem  $(0_0 x_0 y_0 z_0)$  sowie im Endeffektor-Koordinatensystem  $(0_e x_e y_e z_e)$  an. Interpretieren Sie die singulären Konfiguration  $\mathbf{q} \in Q_0$  und die berechneten Richtungen anhand der SIMSCAPE-Visualisierung. Überlegen Sie sich, wie allgemeine äußere Kräfte am Endeffektor auf die Drehachsen wirken.

**Hinweis:** Nutzen Sie den MAPLE-Befehl `NullSpace(...)` für die Berechnung der Richtungen  $\dot{\mathbf{q}}_0$  und  $\mathbf{e}_0$ .

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 3.7.** Geben Sie eine singuläre Konfiguration an und berechnen Sie eine weitere Konfiguration, bei der die Achsen  $5^\circ$  in beliebiger Richtung zu der von Ihnen gewählten singulären Stellung verdreht sind. Bestimmen Sie nun die Geschwindigkeiten im Konfigurationsraum, um den Endeffektor mit  $\|\dot{\mathbf{x}}_e\|_2 = 1$  m/s in die zur Singularität gehörige blockierte Richtung  $\mathbf{e}_0$  zu bewegen. Vergleichen Sie Ihr Ergebnis mit dem Ergebnis aus Aufgabe 3.5.

In der nächsten Aufgabe soll nun die Orientierung des Endeffektors des 3-Achs-Roboters als neuer Ausgang  $\mathbf{x}_e$  untersucht werden, wobei die Position nicht berücksichtigt wird. Es wird die Orientierung des Endeffektors bezüglich des Roboterbasis-Koordinatensystems parametrisiert durch EULER-Winkel als Ausgangsgröße gewählt, d. h. es gilt

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \phi_e = \phi_{zyz}(\mathbf{R}_0^e(\mathbf{q})) . \quad (3.8)$$

**Aufgabe 3.8.** Berechnen Sie die Menge aller Konfigurationen  $Q_0$ , die zu kinematischen Singularitäten der Orientierung führen. Berechnen Sie auch hier jene Geschwindigkeiten im Konfigurationsraum  $\dot{\mathbf{q}}_0$  die zu keiner Drehung des Endeffektors führen. Geben Sie auch hier die Richtung des Moments an, welches nicht durch die Drehachsen aufgenommen werden kann und somit keine Drehungen herbeiführt. Ermitteln Sie weiters die Menge jener Konfigurationen  $Q_{0,R}$  des 3-Achs-Roboters, die zu Repräsentationssingularitäten der gewählten Parametrisierung der Orientierung führen. Interpretieren Sie Ihr Ergebnis.

**Hinweis:** Sie benötigen hier den allgemeinen Zusammenhang zwischen einer Rotationsmatrix und den EULER-Winkeln.

**Hinweis:** Für den vorliegenden 3-Achs-Roboter ist die Menge der Repräsentationssingularitäten  $Q_{0,R}$  eine Teilmenge der kinematischen Singularitäten  $Q_0$ .

### 3.5 Inverse Kinematik

In der Vorlesung zur Fachvertiefung wurde ein numerisches Verfahren vorgestellt, welches einen Beobachter verwendet, um eine Trajektorie im Arbeitsraum  $\mathbf{x}_{e,d}(t)$  in eine Trajektorie im Konfigurationsraum  $\mathbf{q}_d(t)$  zu übersetzen. Als Trajektorie im Arbeitsraum  $\mathbf{x}_{e,d}(t)$  ist ein Kreis in der  $(y_0z_0)$ -Ebene im Roboterbasis-Koordinatensystem  $(0_0x_0y_0z_0)$  gegeben, welcher gemäß

$$\mathbf{x}_{e,d}(t) = \begin{bmatrix} x_K \\ y_K + r \sin(t) \\ z_K + r \cos(t) \end{bmatrix} \quad (3.9)$$

parametrisiert ist. Nachfolgend soll wieder die Position des Endeffektors nach (3.6) als Ausgang betrachtet. Die Orientierung des Endeffektors wird in den folgenden Aufgaben vernachlässigt. Dadurch ergibt sich  $m = \dim(\mathbf{x}_e) = 3$  und  $n = \dim(\mathbf{q}) = 3$ .

**Aufgabe 3.9.** Berechnen Sie die Trajektorie im Konfigurationsraum  $\mathbf{q}_d(t)$  aus der Trajektorie im Arbeitsraum (3.9) mithilfe des vollständigen Beobachters erster Ordnung für nicht-redundante Roboter. Verwenden Sie das MATLAB/SIMULINK-Modell `UE3_Inverse_Kinematik.slx`. Die Trajektorie kann mithilfe des SIMSCAPE-Modells visualisiert werden. Bearbeiten Sie dazu folgende Punkte:

1. Berechnen Sie die analytische Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  in MAPLE und exportieren Sie diese nach MATLAB.
2. Implementieren Sie den vollständigen Beobachter erster Ordnung für kinematisch nicht-redundante Roboter in der dafür vorgesehenen *Matlab Function*.
3. Implementieren Sie die Trajektorie (3.9) und deren Ableitung ebenfalls in der dafür vorgesehenen *Matlab Function*. Berechnen Sie die Trajektorie  $\mathbf{x}_{e,d}(t)$  als Funktion der Simulationszeit  $t$ . Die Kreisparameter können mit  $x_K = 0.6707$  m,  $y_K = 0$  m,  $z_K = 0.9593$  m und  $r_K = 0.2$  m 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 die Bewegung des Roboters, wenn eine andere Startkonfiguration gewählt wird bzw. wenn Sie die Parameter  $z_K$  und  $y_K$  verändern. Bleibt der Beobachter stabil? Begründen Sie Ihre Antwort.
6. Wählen Sie  $x_K = 0.8$  m,  $y_K = 0$  m,  $z_K = 0.4$  m und  $r_K = 0.3$  m und interpretieren Sie die Roboterbewegung. Warum kommt der Endeffektor von der vorgegebene Trajektorie  $\mathbf{x}_{e,d}(t)$  ab?

## 3.6 Dynamik

Die Bewegungsgleichungen mechanischer Systeme lassen sich immer in der Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (3.10)$$

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. Weiters ist die Formulierung von Regelungsstrategien in dieser Form besonders einfach. Die in (3.10) auftretenden Terme sollen im Folgenden bestimmt werden.

**Aufgabe 3.10.** Bestimmen Sie den Vektor der äußeren generalisierten Kräfte und Momente  $\boldsymbol{\tau}$ , die auf die generalisierten Koordinaten wirken. Nehmen Sie an, dass keine externen Kräfte auftreten, d.h.  $\boldsymbol{\tau}_{ext} = \mathbf{0}$  und die dissipativen Kräfte  $\boldsymbol{\tau}_d$  vernachlässigt werden.

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

Mithilfe der kinematischen Zusammenhänge aus den Abschnitten 3.2 und 3.3 können alle Matrizen, die die Dynamik beschreiben, berechnet werden.

**Aufgabe 3.11.** Bestimmen Sie die Massenmatrix  $\mathbf{M}(\mathbf{q})$ , die Matrix der Coriolis- und Zentripetal-Terme  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ , sowie den Vektor der Potenzialterme  $\mathbf{g}(\mathbf{q})$  auf Basis der jeweiligen Definitionen im Kapitel *Euler-Lagrange Gleichungen* des Vorlesungsskriptums zur VU Modellbildung [2]. Überprüfen Sie weiters die Gültigkeit der Zusammenhänge

$$\dot{\mathbf{M}} = \mathbf{C} + \mathbf{C}^T, \quad (3.11)$$

$$\mathbf{M} = \mathbf{M}^T. \quad (3.12)$$

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

**Aufgabe 3.12.** 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. Berechnen Sie die Matrizen  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  und  $\mathbf{g}(\mathbf{q})$  exportieren Sie diese mithilfe der gegebenen MAPLE-Funktion `file_export(...)` nach MATLAB/SIMULINK.
2. Formen Sie Gleichung (3.10) 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 und als Eingang  $\mathbf{u}$  sollen die Motordrehmomente  $\boldsymbol{\tau}_c$  verwendet werden.
3. Implementieren Sie Ihr Modell als *Matlab Function* mit anschließendem Integratorblock im MATLAB/SIMULINK-Modell `UE3_Modellvergleich.slx`.
4. Vergleichen Sie das von Ihnen exportierte Modell mit dem als `p-Code` gegebenen Referenzmodell.

## 3.7 Belastung am Endeffektor

In diesem Abschnitt wird in der Konfiguration

$$\mathbf{q}^T = \left[ \frac{\pi}{2} \quad \frac{\pi}{2} \quad \frac{\pi}{2} \right] \quad (3.13)$$

eine Lastkraft  $\mathbf{F}_{ext}$  von 9.81 N in negative  $z_0$ -Richtung, d. h.

$$\mathbf{F}_{ext}^T = \left[ 0 \quad 0 \quad -9.81 \text{ N} \right], \quad (3.14)$$



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

Tabelle 3.2: Parameter für das dynamische Modell

am Endeffektor aufgebracht, siehe Abbildung 3.2.

**Aufgabe 3.13.** Bestimmen Sie die nötigen Drehmomente  $\tau_c$ , um die Last in dieser Konfiguration zu halten. Prüfen Sie Ihr Ergebnis mit dem gegebenen MATLAB/SIMULINK-Modell `UE3_Belastung.slx` in dem Sie die berechneten Drehmomente  $\tau_c$  aufschalten. Tragen Sie dazu Ihre Berechnungen in die MATLAB-Datei `parameters.m` ein.

**Hinweis:** Betrachten Sie die Lastkraft  $\mathbf{F}_{ext}$  als externe Kraft und nicht als Teil des Robotermodells.

**Hinweis:** Finden Sie geeignete Simulationseinstellungen (Solver).

### 3.8 Identifikation der Gelenksreibung

Zur Erreichung größtmöglicher Genauigkeit sind in Robotern spielfreie Getriebe verbaut. Dadurch entsteht in den Gelenken ein großes Reibungsmoment. Dieses soll im Folgenden anhand von Messungen identifiziert werden.

Bei der Vermessung der Reibung eines Gelenks  $q$  wird eine spezielle Geschwindigkeitstrajektorie  $\dot{q}(t)$  mittels eines Geschwindigkeitsreglers abgefahren und das aufgebrachte Motordrehmoment wird gemessen. Die Trajektorie enthält lange Abschnitte mit einer konstanten Geschwindigkeit. Bei einem reibungsfreien, idealen Gelenk würde in diesen Abschnitten aufgrund des Drehimpulserhaltungssatzes kein Moment  $\tau$  benötigt werden. Aufgrund der Reibung wird aber durch den Geschwindigkeitsregler ein Drehmoment

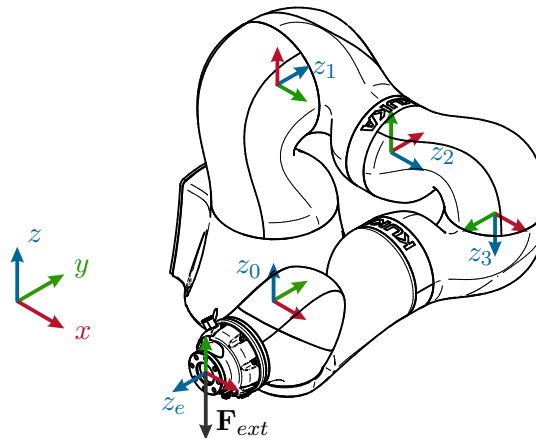


Abbildung 3.2: Lastkraft  $\mathbf{F}_{ext}$  am Endeffektor des KUKA LBR iiwa 14 R820.

aufgeschaltet, welches genau dem dissipativen Anteil  $\tau_d$ , also dem gesuchten momentanen Reibungsmoment, entspricht. Dabei ist darauf zu achten, dass die betrachtete Achse parallel zur Erdbeschleunigung ausgerichtet ist oder die auftretenden Gravitationskräfte kompensiert werden.

**Aufgabe 3.14.** Gegeben ist eine Messreihe des Motormoments von einer Achse des Roboters und den zugehörigen Geschwindigkeiten im Konfigurationsraum  $\dot{q}$  in der MATLAB-Datei `Reibungsmessung.mat`. Stellen Sie das gemessene Reibungsmoment  $\tau_d$  als Funktion der Winkelgeschwindigkeit  $\dot{q}$  dar.

Im Weiteren sollen die Parameter eines Reibungsmodells anhand des gemessenen Reibungsmoment  $\tau_d$  identifiziert werden. Ein einfaches und häufig verwendetes Modell ist das Coulombsche Reibungsgesetz zusammen mit viskoser Reibung, welches durch

$$\tau_d = -\operatorname{sgn}(\dot{q})k_c + k_v\dot{q} \quad (3.15)$$

beschrieben wird. Die Konstanten  $k_c$  und  $k_v$  können aus den Messungen von  $\tau_d$  und  $\dot{q}$  durch das Prinzip der kleinsten Fehlerquadrate bestimmt werden. Dazu werden die  $N$  Messungen des Reibungsmoments  $\tau_{d,k}$  mit  $k = 1, 2, \dots, N$  und der der Winkelgeschwindigkeit  $\dot{q}_k$  in (3.15) eingesetzt. Man erhält ein überbestimmtes Gleichungssystem zur Bestimmung der Parameter in der Form

$$\mathbf{S}\mathbf{p} = \mathbf{y} , \quad (3.16)$$

mit der sogenannten Regressormatrix

$$\mathbf{S} = \begin{bmatrix} -\operatorname{sgn}(\dot{q}_1) & \dot{q}_1 \\ -\operatorname{sgn}(\dot{q}_2) & \dot{q}_2 \\ \vdots & \vdots \\ -\operatorname{sgn}(\dot{q}_N) & \dot{q}_N \end{bmatrix} , \quad (3.17)$$

dem Parametervektor

$$\mathbf{p} = \begin{bmatrix} k_c \\ k_v \end{bmatrix} \quad (3.18)$$

sowie dem Messvektor

$$\mathbf{y} = \begin{bmatrix} \tau_{d,1} \\ \tau_{d,2} \\ \vdots \\ \tau_{d,N} \end{bmatrix}. \quad (3.19)$$

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

$$\mathbf{e} = \mathbf{S}\mathbf{p} - \mathbf{y} \quad (3.20)$$

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

$$(\mathbf{S}\mathbf{p} - \mathbf{y})^T(\mathbf{S}\mathbf{p} - \mathbf{y}) \rightarrow \min. \quad (3.21)$$

Dies führt auf die Bedingung, dass die Ableitung des quadratischen Fehlers nach dem Parameter verschwindet

$$\frac{d}{d\mathbf{p}} \left( (\mathbf{S}\mathbf{p} - \mathbf{y})^T(\mathbf{S}\mathbf{p} - \mathbf{y}) \right) = 2\mathbf{S}^T\mathbf{S}\mathbf{p} - 2\mathbf{S}^T\mathbf{y} = \mathbf{0}. \quad (3.22)$$

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

$$\mathbf{p} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{y}. \quad (3.23)$$

**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 auszugleichen.

**Hinweis:** In der Praxis sollten Gleichungssysteme der Art (3.23) in MATLAB mit dem Backslash-Operator

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

gelöst werden. Hierbei wird automatisch der geeignete Algorithmus ausgewählt.

### 3.9 Literatur

- [1] A. Kugi und W. Kemmetmüller, *Skriptum zur VU Fachvertiefung Automatisierungs- und Regelungstechnik (WS 2020/2021)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2020.
- [2] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2020)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2020.

- [3] A. Kugi, *Skriptum zur VU Automatisierung (WS 2020/2021)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2020.