

## 3 Trajektorienfolgeregelung

Der Inhalt dieser Übung ist der Entwurf von Trajektorienfolgeregelungen für flache und nichtflache Ausgänge im Rahmen eines theoretischen Beispiels und des Laborversuchs Helikopter.

Studieren Sie als Vorbereitung auf die vorliegende Übung zumindest folgendes Skriptum:

- Skriptum zur VO Nichtlineare dynamische Systeme und Regelung (SS 2024) [1]
  - Kapitel 6.



Die im Folgenden angegebenen MAPLE- und MATLAB-Dateien stehen auf der Homepage der Lehrveranstaltung im zip-Archiv **TrajFR.zip** zur Verfügung.



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

- Micheal Schwegel [<schwegel@acin.tuwien.ac.at>](mailto:schwegel@acin.tuwien.ac.at) oder
- Luca Gasparini [<gasparini@acin.tuwien.ac.at>](mailto:gasparini@acin.tuwien.ac.at).

### 3.1 Riementrieb

Als Beispiel zur Trajektorienfolgeregelung eines nichtflachen Ausgangs wird ein Riementrieb betrachtet. Dieser besteht aus einer von einem permanentenerregten Synchronmotor angetriebenen Antriebsscheibe und einer mit einer Last verbundenen Abtriebsscheibe, siehe Abbildung 3.1. Dabei wird angenommen, dass der Riemen masselos und elastisch ist. Zur Beschreibung des permanentenerregten Synchronmotors (PSM) erfolgt eine Transformation der Ströme in ein orthogonales rotorfestes Koordinatensystem ( $dq$ -Koordinaten). Es gelten dann die folgenden Differentialgleichungen

$$\frac{d}{dt}i_q = \frac{2}{3L_m} \left( -R_m i_q + \frac{3}{2} L_m p \omega_m i_d - \Phi p \omega_m + u_q \right) \quad (3.1a)$$

$$\frac{d}{dt}i_d = \frac{2}{3L_m} \left( -R_m i_d - \frac{3}{2} L_m p \omega_m i_q + u_d \right), \quad (3.1b)$$

wobei  $i_q$  und  $i_d$  die Ströme in der Quadraturachse bzw. direkten Achse bezeichnen.  $u_q$  und  $u_d$  sind die zugehörigen Spannungen,  $L_m$  ist die Induktivität,  $R_m$  ist der elektrische Widerstand,  $p$  beschreibt die Polpaarzahl und  $\Phi$  ist der magnetische Fluss des Permanentmagneten. Das elektrische Moment  $M_m$  des Motors berechnet sich aus

$$M_m = \frac{3}{2} p \Phi i_q. \quad (3.2)$$

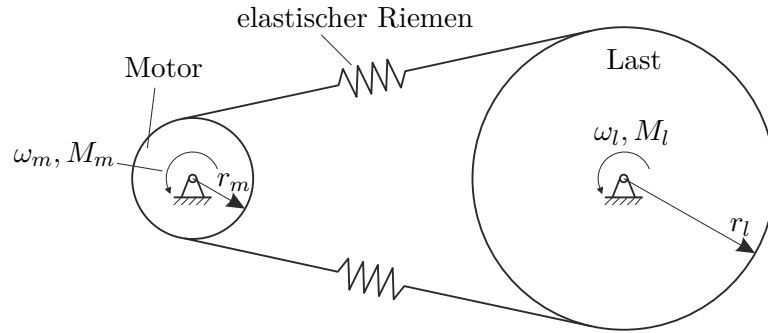


Abbildung 3.1: Prinzipskizze des Riementriebs.

Zur Berechnung der Bewegungsgleichungen der Last und des Motors betrachtet man die durch den Riemen übertragene Kraft  $F_r$  als Funktion der Längung  $\varepsilon = r_l\varphi_l - r_m\varphi_m$  des Riemens, wobei  $\varphi_l$  die Verdrehung der Riemenscheibe der Last und  $\varphi_m$  die Verdrehung der Antriebsscheibe des Motors beschreibt. Es wird angenommen, dass folgender nichtlinearer Zusammenhang zwischen  $\varepsilon$  und  $F_r$  gilt:

$$F_r(\varepsilon) = c_{r,1}\varepsilon + c_{r,3}\varepsilon^3. \quad (3.3)$$

Dabei bezeichnen  $c_{r,1}$  und  $c_{r,3}$  die lineare bzw. die kubische Steifigkeit des Riemens. Die Bewegungsgleichung des Motors kann somit in der Form

$$\frac{d}{dt}\omega_m = \frac{1}{I_m}(2F_r(\varepsilon)r_m - d_m\omega_m + M_m) \quad (3.4)$$

mit dem Massenträgheitsmoment des Motors  $I_m$  angegeben werden. Die Konstante  $d_m$  beschreibt darin die viskose Dämpfung des Motors. Die Bewegungsgleichung der Last ergibt sich mit dem externen Lastmoment  $M_l$  in der Form

$$\frac{d}{dt}\omega_l = \frac{1}{I_l}(-2F_r(\varepsilon)r_l - d_l\omega_l + M_l), \quad (3.5)$$

wobei  $d_l$  die viskose Dämpfung und  $I_l$  das Massenträgheitsmoment der Last beschreiben. Das Modell wird durch die Differentialgleichung für  $\varepsilon$  in der Form

$$\frac{d}{dt}\varepsilon = r_l\omega_l - r_m\omega_m \quad (3.6)$$

vervollständigt.

**Aufgabe 3.1 (Analyse des AI-Systems).** Stellen Sie das System in Form eines Systems mit affinem Eingang (AI-System) dar. Verwenden Sie dazu den Zustand  $\mathbf{x} = [\omega_m \ i_q \ i_d \ \varepsilon \ \omega_l]^T$  sowie den Eingang  $\mathbf{u} = [u_q \ u_d]^T$ . Berechnen Sie den (vektoriellen) relativen Grad für den Ausgang  $\mathbf{y} = [\omega_m \ i_d]^T$ . Bestimmen Sie die Ruhelagen  $\mathbf{x}_R$  und  $\mathbf{u}_R$  des Systems für ein allgemeines  $\mathbf{y}_R \neq 0$  und  $M_{l,R} = 0$ .

**Hinweis:** Im vorliegenden Fall empfiehlt es sich, die Ruhelage der Riemendehnung nicht explizit zu lösen, sondern als implizite Gleichung mitzuführen und gegebenenfalls zur Implementierung numerisch zu evaluieren. Des Weiteren ist es für die betrachtete Anwendung zweckmäßig, die Ruhelagen  $\omega_{m,R}$  und  $i_{d,R}$  vorzugeben und daraus  $u_{q,R}$  sowie  $u_{d,R}$  zu bestimmen.

In der betrachteten Anwendung soll nun eine Trajektorienfolgeregelung für die Motordrehzahl  $\omega_m$  und den Strom  $i_d$  entworfen werden.

**Aufgabe 3.2 (Feedforwardlinearisierung).** Berechnen Sie auf Basis einer exakten Feedforwardlinearisierung eine Steuerung für die Eingangsgrößen  $u_q$  und  $u_d$ , mit welcher den hinreichend oft stetig differenzierbaren Sollgrößen  $\omega_{m,d}(t)$  und  $i_{d,d}(t)$  gefolgt werden kann. Wählen Sie als Zustände der internen Dynamik die Längung des Riemens  $\varepsilon$  sowie die Winkelgeschwindigkeit  $\omega_l$  der Last. Überlegen Sie sich, wie oft die Sollgrößen  $\omega_{m,d}(t)$  und  $i_{d,d}(t)$  stetig differenzierbar sein müssen, damit die Stellgrößen  $u_q$  und  $u_d$  stetig sind.

Zeichnen Sie ein Schaltbild der Vorsteuerung unter Berücksichtigung der internen Dynamik. Benennen Sie dabei die Blöcke und die benötigten Signale. Überlegen Sie in welcher Funktion das Lastmoment  $M_l$  auftritt.

**Hinweis:** Wegen der im Weiteren geforderten Implementierung ist es sinnvoll, die einzelnen Komponenten der Vorsteuerung separat anzugeben.

Damit eine Implementierung der Steuerung möglich ist, muss die Stabilität der Nulldynamik nachgewiesen werden.

**Aufgabe 3.3 (Stabilität der Nulldynamik).** Weisen Sie die asymptotische Stabilität der Nulldynamik mittels der direkten Methode von Lyapunov nach. Verwenden Sie als Lyapunovfunktion die Summe der potentiellen Energie der Riemendehnung und der kinetischen Energie der Abtriebsscheibe.

**Aufgabe 3.4 (Implementierung der internen Dynamik).** Implementieren Sie die interne Dynamik sowohl als zeitkontinuierliches System in Form einer MATLAB `Function` mit einem zusätzlichen Integrator-Block, als auch als zeitdiskretes System in Form einer MATLAB `Function` mit Abtastzeit  $T_a$ . Verwenden und implementieren Sie zur numerischen Integration das explizite *Heun*-Verfahren mit einer Schrittweite von  $T_a = 0.1$  ms. Begründen Sie, weshalb die Akausalität des *Heun*-Verfahrens für die Berechnung der Vorsteuerung kein Problem darstellt. Implementieren Sie die algebraischen Abbildungen der Vorsteuerung ebenfalls in Form von MATLAB `Functions`.

**Hinweis:** Das explizite *Heun*-Verfahren ist ein numerisches Integrationsverfahren für gewöhnliche Differentialgleichungen der Form

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

und berechnet sich zu (mit  $\mathbf{x}(t_k) = \mathbf{x}_k$  und  $\mathbf{u}(t_k) = \mathbf{u}_k$ )

$$\begin{aligned} t_k &= t_0 + kh \\ \mathbf{x}_{k+1}^{[P]} &= \mathbf{x}_k + h\mathbf{f}(t_k, \mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{x}_{k+1} &= \frac{1}{2}\mathbf{x}_k + \frac{1}{2}(\mathbf{x}_{k+1}^{[P]} + h\mathbf{f}(t_{k+1}, \mathbf{x}_{k+1}^{[P]}, \mathbf{u}_{k+1})). \end{aligned}$$

**Aufgabe 3.5 (Validierung der Steuerung).** Testen Sie das Führungs- und Störverhalten unter Verwendung der vorgegebenen Implementierung des Systems (SIMULINK-Datei `Sim_Riementrieb.slx`) und vergleichen Sie die Ergebnisse der zeitkontinuierlichen und zeitdiskreten Implementierung der internen Dynamik.

Durch reine Steuerung kann nicht auf externe Störungen wie hier  $M_l$  oder auf Modellungenauigkeiten reagiert werden. Daher ist es sinnvoll, die Steuerung um eine Regelung im Sinn der Zwei-Freiheitsgrad-Regelkreisstruktur zu erweitern. Durch den wohldefinierten relativen Grad wäre eine solche durch die Eingangs-Ausgangslinearisierung möglich. Da hierbei jedoch nichtlineare Terme in der Rückführung auftreten, ist diese Strategie nicht robust gegen Parameterschwankungen. Wie im Skriptum [1] angeführt, kann es schwierig sein, die Stabilität der Dynamik des Fehlers  $\mathbf{e}_y = \mathbf{y} - \mathbf{y}_d$  nachzuweisen, weil diese oft nichtlinear und zeitvariant ist. Eine besonders einfache Lösung des Problems besteht nun darin, die Dynamik des Fehlersystems um eine Ruhelage zu linearisieren und den Reglerentwurf am linearisierten System durchzuführen.

**Aufgabe 3.6 (Fehlerdynamik).** Bestimmen Sie die Dynamik des Fehlers  $\mathbf{e} = \mathbf{x} - \mathbf{x}_d$ . Linearisieren Sie die erhaltene Fehlerdynamik in weiterer Folge um  $\mathbf{e} = \mathbf{0}$ . Setzen Sie für  $\mathbf{x}_d$  einen allgemeinen stationären Arbeitspunkt (Ruhelage) des Systems ein. Gehen Sie dazu von beliebig vorgebbaren stationären Arbeitspunkten für  $\omega_{m,d}$  und  $i_{d,d}$  aus und ermitteln Sie aus diesen die stationären Arbeitspunkte für  $i_{q,d}$ ,  $\omega_{l,d}$  und  $\varepsilon_d$ .

**Hinweis:** Behalten Sie auch hier  $\varepsilon_d$  in impliziter Form bei. Die Systemmatrix des linearisierten Fehlersystems soll sich durch die Vorgabe der stationären Arbeitspunkte für  $\omega_{m,d}$  und  $i_{d,d}$  eindeutig bestimmen lassen.

**Aufgabe 3.7 (Trajektorienfolgeregler).** Berechnen Sie in MATLAB das zur linearisierten Fehlerdynamik gehörende Abtastsystem. Wählen Sie für die Abtastzeit  $T_a = 0.1 \text{ ms}$  und für den stationären Arbeitspunkt  $\omega_{m,d} = 0$  und  $i_{d,d} = 0$ . Bestimmen Sie die zugehörige  $z$ -Übertragungsmatrix und entwerfen Sie eine geeignete Regelung mittels zeitdiskreter PI-Regler. Implementieren Sie die Regler in Form

von MATLAB Functions und testen Sie das Führungs- und das Störverhalten unter Verwendung der vorgegebenen Implementierung des Systems (SIMULINK-Datei `Sim_Riementrieb.slx`).

**Hinweis:** Die  $z$ -Übertragungsmatrix ist für den gewählten Arbeitspunkt eine Diagonalmatrix.

Zur Lösung der obigen Aufgaben steht auf der Homepage der Lehrveranstaltung ein zip-Archiv [TrajFR.zip](#) zur Verfügung. Dessen Unterverzeichnis **Riementrieb** enthält ein MATLAB/SIMULINK-Modell zur Simulation des Riementriebs.

### 3.1.1 Parameter des Riementriebs

Größe	Wert	
$c_{r,1}$	10	N/m
$c_{r,3}$	$1 \cdot 10^6$	N/m <sup>3</sup>
$I_l$	$4 \cdot 10^{-5}$	kg m <sup>2</sup>
$d_l$	$1 \cdot 10^{-3}$	N m s
$r_l$	0.05	m
$I_m$	$5 \cdot 10^{-6}$	kg m <sup>2</sup>
$d_m$	0.01	N m s
$r_m$	0.025	m
$R_m$	1.388	$\Omega$
$L_m$	$1.475 \cdot 10^{-3}$	H
$p$	2	-
$\Phi$	$2.715 \cdot 10^{-2}$	Wb

Tabelle 3.1: Parameter des Riementriebs.

## 3.2 Helikopter

### 3.2.1 Einleitung und Modellbildung

Als weiteres Anwendungsbeispiel für die Trajektorienfolgeregelung wird der Laborversuch Helikopter nach Abbildung 3.2 untersucht. Für diesen soll im Rahmen dieses Praktikums eine flachheitsbasierte Mehrgrößenregelung zur Trajektorienfolge entwickelt und implementiert werden.

Der Helikopter besteht aus dem Mast, welcher sich um den Winkel  $q_1$  frei drehen kann, dem Arm mit der Drehung um den Winkel  $q_2$  und der Aufhängung mit der Drehung um den Winkel  $q_3$ . An beiden Enden dieser Aufhängung sind Luftschrauben (Rotoren) angebracht, welche durch Gleichstrommotoren angetrieben werden. Über Anlegen einer elektrischen

Spannung an die Motoren ergeben sich durch die Drehungen der Rotoren entsprechende Rotorkräfte. Im Rahmen der Modellbildung werden die Summe bzw. die Differenz der Rotorkräfte als Stellgrößen des Systems dienen. Mit Hilfe dieser beiden Stellgrößen sollen nun die drei Freiheitsgrade  $q_1$ ,  $q_2$  und  $q_3$  geregelt werden. Wie im Skriptum [1] erwähnt,

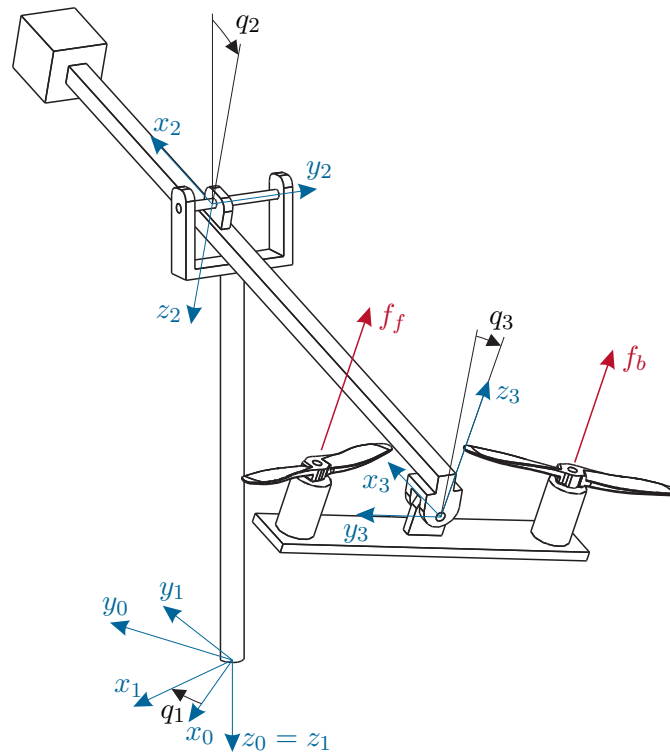


Abbildung 3.2: Der Laborversuch Helikopter.

handelt es sich bei dem Laborversuch um ein mechanisch unteraktuiertes Starrkörpersystem. Im Allgemeinen kann das mathematische Modell zum Beispiel mit dem Lagrange-Formalismus hergeleitet werden, siehe Skriptum zur Modellbildung [2], was in diesem Fall auf ein implizites Modell der Form

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

mit der Massenmatrix  $\mathbf{M}(\mathbf{q})$ , der Matrix  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  der Coriolis- und Zentrifugalterme, dem Vektor der Potentialkräfte  $\mathbf{g}(\mathbf{q})$  und dem Vektor der verallgemeinerten Kräfte  $\boldsymbol{\tau}$  führt.

Da mit diesem Modell die Untersuchungen und der Reglerentwurf nur schwer durchzuführen sind, sucht man durch gezielte Vernachlässigungen ein einfacheres mathematisches Modell, das aber dennoch die Realität hinreichend gut beschreibt. Einige der vorgenommenen Vereinfachungen sind im Folgenden angeführt:

1. Vernachlässigung von kleinen Anteilen in der kinetischen Energie
2. Vernachlässigung der Dynamik der elektrischen Teilsysteme der Gleichstrommotoren

3. Vernachlässigung der Dynamik der mechanischen Teilsysteme der Gleichstrommotoren mit angeschlossenen Rotoren
4. Vernachlässigung der Reibungseffekte

Durch die Vernachlässigungen 2. und 3. erhält man einen statischen Zusammenhang zwischen den Rotorkräften  $f_f$  und  $f_b$  und den Spannungen  $U_f$  und  $U_b$  in der Form

$$f_i = \begin{cases} k_+ U_i^2, & U_i \geq 0 \\ k_- U_i^2, & U_i < 0 \end{cases}, \quad i \in \{f, b\} \quad (3.8)$$

mit

$$\begin{aligned} k_+ &= 4.855 \cdot 10^{-3} \text{ N/V}^2 \\ k_- &= -1.503 \cdot 10^{-3} \text{ N/V}^2. \end{aligned} \quad (3.9)$$

Durch die weitere Annahme  $\sin(q_2) \approx 0$  bei der Berechnung der verallgemeinerten Kräfte zufolge der eingprägten Rotorkräfte erhält man die Bewegungsgleichungen in der vereinfachten Form

$$\ddot{q}_1 = b_1 \cos(q_2) \sin(q_3) v_1 \quad (3.10a)$$

$$\ddot{q}_2 = a_1 \sin(q_2) + a_2 \cos(q_2) + b_2 \cos(q_3) v_1 \quad (3.10b)$$

$$\ddot{q}_3 = a_3 \cos(q_2) \sin(q_3) + b_3 v_2. \quad (3.10c)$$

Die Stelleingänge  $v_1$  und  $v_2$  stellen die Summe bzw. die Differenz der Rotorkräfte

$$v_1 = f_f + f_b \quad (3.11a)$$

$$v_2 = f_f - f_b \quad (3.11b)$$

dar. Die aus den physikalischen Größen folgenden numerischen Parameter des mathematischen Modells (3.10) sind in Tabelle 3.2 zu finden. Zur Lösung der folgenden Aufgaben

Parameter	Wert	
$a_1$	-1.1713	rad/s <sup>2</sup>
$a_2$	0.3946	rad/s <sup>2</sup>
$a_3$	-0.5326	rad/s <sup>2</sup>
$b_1$	-0.6354	rad/(kg.m)
$b_2$	-0.6523	rad/(kg.m)
$b_3$	4.6276	rad/(kg.m)

Tabelle 3.2: Parameter des vereinfachten mathematischen Modells.

steht auf der Homepage der Lehrveranstaltung ein zip-Archiv [TrajFR.zip](#) zur Verfügung. Dessen Unterverzeichnis **Helikopter** enthält ein MATLAB/SIMULINK-Modell zur Simulation des vollständigen Systems (3.7), bei welchem im Rahmen der Modellbildung keine

Vernachlässigungen getroffen sowie Haftreibungseffekte berücksichtigt wurden. Dieses Modell bildet die tatsächliche Strecke relativ genau ab und soll zur Überprüfung der entwickelten Steuerungs- und Regelungsstrategien vor der Implementierung am realen System genutzt werden. Als Eingangsgrößen für dieses Modell dienen wiederum die Summen- und Differenzkraft. Die Ausgänge sind die 6 mechanischen Zustandsgrößen

$$\mathbf{y}^T = [q_1 \quad \dot{q}_1 \quad q_2 \quad \dot{q}_2 \quad q_3 \quad \dot{q}_3], \quad (3.12)$$

welche auch am realen Versuchsaufbau durch Messungen bzw. näherungsweise Differenziation zur Verfügung stehen.

*Aufgabe 3.8 (Berechnung der Ruhelagen, Implementierung und Verifikation des Modells).*

1. Bestimmen Sie die Ruhelagen des Modells (3.10) für  $v_{1,R} = v_{2,R} = 0$ .
2. Implementieren Sie das Modell (3.10) mit den Eingängen  $v_1$  und  $v_2$  als **MATLAB Function** mit einem zusätzlichen Integrator und testen Sie die grundsätzliche Funktionalität.
3. Überprüfen Sie das im vorigen Unterpunkt implementierte Modell anhand des vollständigen Modells.

### 3.2.2 Flachheitsbasierte Vorsteuerung

Im Folgenden soll die Flachheit des Laborversuchs Helikopter nachgewiesen und zur Herleitung geeigneter Steuerungs- und Regelungsalgorithmen genutzt werden.

*Aufgabe 3.9 (Differenzielle Flachheit und flachheitsbasierte Vorsteuerung).*

1. Weisen Sie nach, dass das Paar  $q_1$  und  $q_2$  einen flachen Ausgang des Systems (3.10) darstellt. Berechnen Sie explizit die Parametrierungen der Stellgrößen

$$\begin{aligned} v_1 &= \phi_1(q_1, q_2, \dot{q}_1, \dot{q}_2, \dots) \\ v_2 &= \phi_2(q_1, q_2, \dot{q}_1, \dot{q}_2, \dots) \end{aligned}$$

in Abhängigkeit des flachen Ausgangs sowie dessen zeitlicher Ableitungen.

2. Basierend auf den Parametrierungen der Stellgrößen  $v_1$  und  $v_2$  kann eine flachheitsbasierte Vorsteuerung durchgeführt werden. Überlegen Sie, bis zu welcher Ordnung die Solltrajektorie stetig differenzierbar sein muss, um stetige Stellgrößen zu erhalten. Welche zusätzlichen Anforderungen an die Solltrajektorie müssen für einen Arbeitspunktwechsel erfüllt sein? Unter einem Arbeitspunktwechsel ist der Übergang von einer Ruhelage in eine andere zu verstehen.

Die flachheitsbasierte Vorsteuerung benötigt eine geeignete Wahl der Solltrajektorien für  $q_1$  und  $q_2$ . Verwenden Sie eine polynomiale Darstellung dieser Solltrajektorien.



**Aufgabe 3.10 (Implementierung der flachheitsbasierten Vorsteuerung).**

1. Implementieren Sie die flachheitsbasierte Vorsteuerung für den Laborversuch Helikopter basierend auf den Ergebnissen von Aufgabe 3.9 in MATLAB/SIMULINK.
2. Passen Sie den Trajektoriengenerator so an, dass der Helikopter zunächst innerhalb von 5 s aus der Ruhelage (gemäß Aufgabe 3.8) auf  $q_2 = 0$  aufsteigt und anschließend innerhalb von 10 s eine volle Umdrehung ( $360^\circ$ ) um die  $q_1$ -Achse durchführt.
3. Testen Sie die Vorsteuerung in der Simulation mit dem vollständigen und dem vereinfachten Modell.

**3.2.3 Trajektorienfolgeregelung**

Im Folgenden soll eine Trajektorienfolgeregelung für die Freiheitsgrade  $q_1$  und  $q_2$  des Helikopters entwickelt werden. Dazu gibt es verschiedene Möglichkeiten, von denen zwei in Folge erläutert werden sollen.

Die erste Möglichkeit besteht darin, die entwickelte Vorsteuerung aus Aufgabe 3.10 zu verwenden und im Sinne der Zwei-Freiheitsgrad-Regelkreisstruktur einen zusätzlichen Regelungsanteil aufzuschalten. Wie in [1] erwähnt, entsteht dadurch im Allgemeinen ein zeitvariantes, nichtlineares Fehlersystem, dessen Stabilität schwierig zu beurteilen ist.

Bei der zweiten Möglichkeit kann man sich eine spezielle Eigenschaft flacher Systeme zu Nutze machen. Diese soll nachfolgend erläutert und auch für den Entwurf des Trajektorienfolgereglers herangezogen werden. Man betrachte ein flaches System

$$E_i(\bar{\mathbf{w}}, \dot{\bar{\mathbf{w}}}, \dots, \bar{\mathbf{w}}^{(\rho)}) = 0, \quad i = 1, \dots, n. \quad (3.13)$$

Diesem System seien Zustandsgrößen  $\mathbf{x} \in \mathbb{R}^n$ , Eingangsgrößen  $\mathbf{u} \in \mathbb{R}^m$  und Ausgangsgrößen  $\mathbf{y} \in \mathbb{R}^m$  zugeordnet.

Ziel ist es, eine (Zustands-)Rückführung so zu finden, dass das unter dieser Rückführung entstehende System exakt linear von einem neuen Eingang  $\mathbf{w} \in \mathbb{R}^m$  zum Ausgang  $\mathbf{y}$  ist. Sofern dies gelingt, können alle Methoden der Trajektorienfolgeregelung für lineare Systeme angewandt werden und das Problem ist gelöst.

Bei flachen Systemen gilt nun, dass man immer eine so genannte quasi-statische Zustandsrückführung finden kann, die das System (3.13) exakt linearisiert, d.h. in die Form

$$y_i^{(\lambda_i)} = w_i, \quad i = 1, \dots, m \quad (3.14)$$

bringt. Eine quasi-statische Zustandsrückführung ist eine Rückführung der Form

$$u_i = \varphi_i(\mathbf{x}, \mathbf{w}, \dot{\mathbf{w}}, \dots, \mathbf{w}^{(\beta_i)}), \quad i = 1, \dots, m. \quad (3.15)$$

Die Zustandsrückführung (3.15) hängt im Allgemeinen auch von den Zeitableitungen der neuen Eingänge  $\mathbf{w}$  ab, weshalb man von einer *quasi*-statischen Zustandsrückführung spricht. Das erhaltene System (3.14) besteht aus  $m$  entkoppelten linearen Differenzialgleichungen.

Das Modell des Helikopters gemäß (3.10) ist flach in den Ausgängen  $q_1$  und  $q_2$  und daher kann die vorgestellte Methode direkt angewandt werden. Die Berechnung der exakt linearisierenden quasi-statischen Zustandsrückführung sowie die anschließende Stabilisierung des Fehlersystems werden im Folgenden durchgeführt.

### Berechnung der exakt linearisierenden quasi-statischen Zustandsrückführung

Ausgangspunkt ist das flache Modell (3.10) des Helikopters, welches in die Standardform eines nichtlinearen Systems mit affinem Eingang

$$\begin{bmatrix} \dot{q}_1 \\ \dot{\omega}_1 \\ \dot{q}_2 \\ \dot{\omega}_2 \\ \dot{q}_3 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ 0 \\ \omega_2 \\ f_2(q_2) \\ \omega_3 \\ f_3(q_2, q_3) \end{bmatrix} + \begin{bmatrix} 0 \\ g_1(q_2, q_3) \\ 0 \\ g_2(q_3) \\ 0 \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g_3 \end{bmatrix} v_2 \quad (3.16)$$

mit

$$g_1(q_2, q_3) = b_1 \cos(q_2) \sin(q_3) \quad (3.17a)$$

$$g_2(q_3) = b_2 \cos(q_3) \quad (3.17b)$$

$$g_3 = b_3 \quad (3.17c)$$

$$f_2(q_2) = a_1 \sin(q_2) + a_2 \cos(q_2) \quad (3.17d)$$

$$f_3(q_2, q_3) = a_3 \cos(q_2) \sin(q_3) \quad (3.17e)$$

gebracht wird. Die flachen Ausgänge lauten  $y_1 = q_1$  und  $y_2 = q_2$ .

Das System (3.16) kann gedanklich in 3 Teilsysteme bestehend aus jeweils 2 Zuständen  $(q_1, \omega_1)$ ,  $(q_2, \omega_2)$  und  $(q_3, \omega_3)$  unterteilt werden. Um nun eine quasi-statische exakt linearisierende Zustandsrückführung zu finden, kann man das zweite Teilsystem

$$\dot{q}_2 = \omega_2 \quad (3.18a)$$

$$\dot{\omega}_2 = f_2(q_2) + g_2(q_3)v_1 \quad (3.18b)$$

betrachten und  $q_3$  vorerst als gegebene Größe auffassen. Dann kann sofort eine exakte Eingangs-Ausgangslinearisierung für den Ausgang  $y_2 = q_2$  durchgeführt werden. Dazu differenziert man  $y_2$  so lange, bis der Eingang  $v_1$  erscheint. Dies ergibt

$$y_2 = q_2 \quad (3.19a)$$

$$\dot{y}_2 = \dot{q}_2 = \omega_2 \quad (3.19b)$$

$$\ddot{y}_2 = \ddot{q}_2 = \dot{\omega}_2 = f_2(q_2) + g_2(q_3)v_1 \quad (3.19c)$$

und man erkennt, dass die Wahl

$$v_1 = \frac{1}{g_2(q_3)}(-f_2(q_2) + \ddot{y}_2) \quad (3.20)$$

mit dem neuen Eingang  $w_2$  ein lineares Eingangs-Ausgangsverhalten

$$\ddot{y}_2 = w_2 \quad (3.21)$$

zur Folge hat. Man beachte, dass für das Teilsystem 1 ein analoges Vorgehen nicht möglich ist, weil  $g_1(q_2, q_3)$  im interessierenden Arbeitsbereich (bei  $q_3 = 0$ ) null ist. Da  $q_3$  keine zu regelnde Ausgangsgröße der Trajektorienfolgeregelung ist, muss dieser Prozess nicht auf das dritte Teilsystem angewandt werden.

Setzt man (3.20) in (3.16) ein, erhält man die neuen Systemgleichungen

$$\begin{bmatrix} \dot{q}_1 \\ \dot{\omega}_1 \\ \dot{q}_2 \\ \dot{\omega}_2 \\ \dot{q}_3 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ f_1(q_2, q_3) \\ \omega_2 \\ 0 \\ \omega_3 \\ f_3(q_2, q_3) \end{bmatrix} + \begin{bmatrix} 0 \\ g_4(q_2, q_3) \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} w_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g_3 \end{bmatrix} v_2 \quad (3.22)$$

mit

$$f_1(q_2, q_3) = -\frac{g_1(q_2, q_3)}{g_2(q_3)} f_2(q_2) = -\frac{b_1 \cos(q_2) \tan(q_3)}{b_2} (a_1 \sin(q_2) + a_2 \cos(q_2)) \quad (3.23a)$$

$$g_4(q_2, q_3) = \frac{g_1(q_2, q_3)}{g_2(q_3)} = \frac{b_1 \cos(q_2) \tan(q_3)}{b_2}. \quad (3.23b)$$

Für das System (3.22) verbleibt nun ein freier Eingang  $v_2$ , um ein lineares Eingangs-Ausgangsverhalten in  $y_1$  zu bewirken. Analog zu vorhin kann man im Sinne der Eingangs-Ausgangslinearisierung den Ausgang  $y_1 = q_1$  ableiten, bis der Eingang  $v_2$  erscheint. Eine explizite Auswertung ergibt

$$\dot{y}_1 = \omega_1 \quad (3.24a)$$

$$\ddot{y}_1 = f_1(q_2, q_3) + g_4(q_2, q_3) w_2 \quad (3.24b)$$

$$\begin{aligned} y_1^{(3)} &= \underbrace{\frac{\partial f_1(q_2, q_3)}{\partial q_2}}_{\bar{f}_1(q_2, q_3)} \dot{q}_2 + \underbrace{\frac{\partial f_1(q_2, q_3)}{\partial q_3}}_{\bar{\bar{f}}_1(q_2, q_3)} \dot{q}_3 + \underbrace{\frac{\partial g_4(q_2, q_3)}{\partial q_2}}_{\bar{g}_4(q_2, q_3)} \dot{q}_2 w_2 \\ &\quad + \underbrace{\frac{\partial g_4(q_2, q_3)}{\partial q_3}}_{\bar{\bar{g}}_4(q_2, q_3)} \dot{q}_3 w_2 + g_4(q_2, q_3) \dot{w}_2 \\ &= \bar{f}_1(q_2, q_3) \omega_2 + \bar{\bar{f}}_1(q_2, q_3) \omega_3 + \bar{g}_4(q_2, q_3) \omega_2 w_2 + \bar{\bar{g}}_4(q_2, q_3) \omega_3 w_2 + g_4(q_2, q_3) \dot{w}_2 \end{aligned} \quad (3.24c)$$

$$\begin{aligned}
y_1^{(4)} = & \frac{\partial \bar{f}_1(q_2, q_3)}{\partial q_2} \omega_2^2 + \frac{\partial \bar{f}_1(q_2, q_3)}{\partial q_3} \omega_2 \omega_3 + \bar{f}_1(q_2, q_3) \dot{\omega}_2 + \frac{\partial \bar{f}_1(q_2, q_3)}{\partial q_2} \omega_2 \omega_3 \\
& + \frac{\partial \bar{f}_1(q_2, q_3)}{\partial q_3} \omega_3^2 + \bar{f}_1(q_2, q_3) \dot{\omega}_3 + \frac{\partial \bar{g}_4(q_2, q_3)}{\partial q_2} \omega_2^2 w_2 + \frac{\partial \bar{g}_4(q_2, q_3)}{\partial q_3} \omega_2 \omega_3 w_2 \\
& + \bar{g}_4(q_2, q_3) \dot{\omega}_2 w_2 + \bar{g}_4(q_2, q_3) \omega_2 \dot{w}_2 + \frac{\partial \bar{g}_4(q_2, q_3)}{\partial q_2} \omega_2 \omega_3 w_2 + \frac{\partial \bar{g}_4(q_2, q_3)}{\partial q_3} \omega_3^2 w_2 \quad (3.24d) \\
& + \bar{g}_4(q_2, q_3) \dot{\omega}_3 w_2 + \bar{g}_4(q_2, q_3) \omega_3 \dot{w}_2 + \frac{\partial g_4(q_2, q_3)}{\partial q_2} \omega_2 \dot{w}_2 \\
& + \frac{\partial g_4(q_2, q_3)}{\partial q_3} \omega_3 \dot{w}_2 + g_4(q_2, q_3) \ddot{w}_2
\end{aligned}$$

wobei in  $y_1^{(4)}$  über  $\dot{\omega}_3 = f_3(q_2, q_3) + g_3 v_2$  erstmals der Eingang  $v_2$  erscheint. Weiters kann gemäß (3.22)  $\dot{\omega}_2 = w_2$  eingesetzt werden. Mit dem neuen Eingang  $w_1$  und einer geeigneten Wahl

$$v_2 = \varphi_2(w_1, q_2, q_3, \omega_2, \omega_3, w_2, \dot{w}_2, \ddot{w}_2), \quad (3.25)$$

welche aus einem Rechenschritt analog zu (3.19c) und (3.20) folgt, kann auch hier ein lineares Eingangs-Ausgangsverhalten

$$y_1^{(4)} = w_1 \quad (3.26)$$

erreicht werden. In Gl. (3.25) erkennt man den quasi-statischen Charakter der Zustandsrückführung. Es tritt nicht nur  $w_2$ , sondern auch  $\dot{w}_2$  und  $\ddot{w}_2$  auf.

### Aufgabe 3.11 (Quasi-statische Zustandsrückführung).

1. Berechnen Sie die quasi-statischen Zustandsrückführungen (3.20) und (3.25) explizit.
2. Untersuchen Sie, ob die Berechnung der Stellgröße  $v_2$  gemäß (3.25) immer möglich ist, oder ob Singularitäten auftreten können.
3. Testen Sie die quasi-statische Zustandsrückführung am vereinfachten Modell.

**Hinweis:** Es empfiehlt sich, jeden Teilschritt auf seine Plausibilität zu prüfen!

## Stabilisierung des Folgefehlersystems

Ausgangspunkt ist das lineare System (3.26) und (3.21)

$$y_1^{(4)} = w_1 \quad (3.27a)$$

$$\ddot{y}_2 = w_2. \quad (3.27b)$$

Die Fehlerdynamik soll zu

$$e_1^{(4)} = -k_{13}e_1^{(3)} - k_{12}\ddot{e}_1 - k_{11}\dot{e}_1 - k_{10}e_1 - k_{1I}e_{1I} \quad (3.28a)$$

$$\ddot{e}_2 = -k_{21}\dot{e}_2 - k_{20}e_2 - k_{2I}e_{2I} \quad (3.28b)$$

mit den positiven Parametern  $k_{10}$ ,  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{1I}$ ,  $k_{20}$ ,  $k_{21}$  und  $k_{2I}$  vorgegeben werden, wobei  $e_1 = q_1 - q_{1,d}$ ,  $e_2 = q_2 - q_{2,d}$  und

$$e_{1I} = \int e_1 dt \quad (3.29a)$$

$$e_{2I} = \int e_2 dt. \quad (3.29b)$$

### Aufgabe 3.12 (Berechnung des Regelgesetzes).

1. Berechnen Sie die fiktiven Eingangsgrößen  $w_1$  und  $w_2$ , um die Fehlerdynamik gemäß Gl. (3.28) zu erhalten.
2. Setzen Sie die erhaltenen expliziten Ausdrücke für  $w_1$  und  $w_2$  in die Gl. (3.20) und (3.25) bzw. das Ergebnis von Aufgabe 3.11 ein, um die Stellgesetze für die tatsächlichen Eingänge  $v_1$  und  $v_2$  zu erhalten.
3. Bestimmen Sie die Reglerparameter  $k_{10}$ ,  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{1I}$ ,  $k_{20}$ ,  $k_{21}$  und  $k_{2I}$  so, dass die Fehlerdynamik (3.28) asymptotisch stabil ist.

### Aufgabe 3.13 (Implementierung des Regelgesetzes).

1. Implementieren Sie die Regelung gemäß Aufgabe 3.12 als MATLAB Function `FLATreg` im MATLAB/SIMULINK-Modell aus dem zip-Archiv [TrajFR.zip](#). Behalten Sie die vorgegebene Definition der Ein- und Ausgänge bei.
2. Testen Sie die Regelung in der Simulation mit dem vereinfachten Modell. Der Test der Regelung in der Simulation mit dem vollständigen Modell ist rechenzeitintensiv, kann aber natürlich auf freiwilliger Basis ebenfalls durchgeführt werden.

### 3.3 Literatur

- [1] A. Kugi, *Skriptum zur VO Nichtlineare dynamische Systeme und Regelung (SS 2024)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2024. Adresse: <https://www.acin.tuwien.ac.at/master/nichtlineare-dynamische-systeme-und-regelung/>.
- [2] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2024)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2024. Adresse: <https://www.acin.tuwien.ac.at/bachelor/modellbildung/>.
- [3] B. Siciliano und O. Khatib, *Springer Handbook of Robotics* (Springer Handbook of Robotics). Springer Berlin Heidelberg, 2008, ISBN: 9783540239574.