

Steuerung und Regelung eines Doppelpendels

Projektseminar von Tobias Gebhard und Frederik Tesar

Tag der Einreichung: 29. Oktober 2020

1. Gutachten: Prof. Dr.-Ing. Ulrich Konigorski

2. Gutachten: Dr.-Ing. Eric Lenz

Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

REGELUNGSTECHNIK
UND MECHATRONIK

rtm

Fachbereich Elektrotechnik
und Informationstechnik

Steuerung und Regelung eines Doppelpendels

Projektseminar von Tobias Gebhard und Frederik Tesar

1. Gutachten: Prof. Dr.-Ing. Ulrich Konigorski

2. Gutachten: Dr.-Ing. Eric Lenz

Tag der Einreichung: 29. Oktober 2020

Darmstadt

Inhaltsverzeichnis

1	Einleitung	5
2	Modellierung	6
2.1	Modell des Schlitten-Pendel-Systems	6
2.1.1	Koordinaten	6
2.1.2	Herleitung der Bewegungsgleichungen	7
2.1.3	Zustandsraummodell	8
2.1.4	Coulomb-Reibung	9
2.1.5	Ruhelagen	10
2.1.6	Beschränkungen	11
2.2	Motor-Modell	11
2.2.1	Spannungs-Strom-Wandler	11
2.2.2	Gleichstrommotor	11
2.2.3	Getriebe	14
2.3	Modellparameter	14
2.3.1	Begründeter Stand der Modellparameter	15
2.3.2	Parameter des Motor-Modells	16
2.3.3	Parameter des Schlittendoppelpendels	17
2.4	Aufbau des Simulationsmodells	18
2.4.1	Aufbau in SIMULINK	19
2.4.2	SIMULINK Module und Modelle	20
2.4.3	Parameter	22
2.4.4	Initialisierung	22
2.4.5	Auswertung	22
3	Arbeitspunkt-Regelung	24
3.1	System-Linearisierung und Analyse	25
3.1.1	Linearisierung	25
3.1.2	Eigenwerte	26
3.1.3	Steuerbarkeit und Beobachtbarkeit	27
3.2	Aufbau der Regelung	27
3.2.1	Zustandsregler	27
3.2.2	Zustandsermittlung	28
3.2.3	Vorsteuerung F/a und a/v-Regler	29
3.2.4	Motor-Vorsteuerung	31
3.3	Simulationsmodell in SIMULINK	32
3.3.1	SIMULINK Module und Modelle	32
3.3.2	Initialisierung	34

3.3.3	Güteparameter (Q,R)	34
3.3.4	Weitere MATLAB-Funktionen	35
3.4	Anfangswert-Tests	35
3.4.1	Auswertung eines Arbeitspunkt-Tests	36
3.4.2	Anfangswerttest	37
3.4.3	Kritische Anfangswerttests	37
3.4.4	Vergleiche von Anfangswerttests	39
3.5	QR Parameter Tests	41
3.5.1	Vorgehen bei der Optimierung	41
3.5.2	Anwendung in MATLAB	41
3.5.3	Diagramme	42
3.5.4	Weitere Ergebnisse	44
3.6	System Parameter Tests	44
3.6.1	Vorgehen	45
3.6.2	Anwendung in MATLAB	45
3.6.3	Diagramme	46
3.6.4	Schlussfolgerungen zu Parameteränderungen	51
4	Trajektorien	52
4.1	Trajektorienberechnung	52
4.1.1	searchTrajectories	53
4.1.2	calculateTrajectory	55
4.2	Trajektorienfolgeregelung	58
4.3	Stabilisierung der berechneten Trajektorien	60
4.3.1	Vorgehen	60
4.3.2	Vergleichstrajektorie	61
4.3.3	Ohne Gegeninduktion	62
4.3.4	Mit Gegeninduktion	66
4.4	Untersuchung des Einflusses der Modellparameter	68
4.4.1	Vorgehen	68
4.4.2	Durchführung und Ergebnisse	70
5	Fazit und Ausblick	75
5.1	Zusammenfassung und Fazit	75
5.2	Ausblick	76
A	AP-Regelung Systemparameter tests	78
	Literatur	81

1 Einleitung

Das Doppelpendel ist ein nichtlineares System, an dem interessante steuerungs- und regelungstechnische Methoden untersucht werden können. Am Fachgebiet Regelungstechnik und Mechatronik (rtm) der TU Darmstadt existiert dazu ein Versuchsstand, bei dem ein Doppelpendel an einem Schlitten befestigt ist, welcher über einen Elektromotor gesteuert werden kann. Daran sind Untersuchungen und Demonstrationen am (inversen) Einfach- und Doppelpendel möglich.

Eine Übersicht über die in der Vergangenheit durchgeführten Arbeiten zu diesem Versuchsstand kann [17] und [16] entnommen werden. Im Zuge einer Neukonstruktion des Doppelpendels durch Chang [17] im Jahr 2019 haben sich die Systemparameter geändert. Erste Erfahrungen mit dem neuen System zeigen im Vergleich zu dem vorherigen jedoch ein ungünstigeres Verhalten in Bezug auf die regelungstechnischen Eigenschaften. Vermutet wird dabei, dass dies mit den Systemparametern und der erhöhten trockenen Reibung des ersten Pendelgelenks, das mit einem Schleifringkontakt konstruiert wurde, zusammenhängt.

Ziel dieser Arbeit ist daher die systematische Untersuchung des Einflusses der Systemparameter auf Steuerung und Regelung des Schlitten-Pendel-Systems. Das System wird im Rahmen dieser Arbeit rein simulativ betrachtet. Aus diesem Grund soll in einem ersten Schritt ein detailliertes Simulationsmodell des Versuchsstands aufgebaut werden. Da in der Vergangenheit mehrfach verschiedene Systemparameter angenommen worden waren, soll nun ein begründeter Stand der Parameter unter Berücksichtigung der vorherigen Arbeiten recherchiert werden.

Anschließend ist eine Regelung in den vier Arbeitspunkten des Doppelpendels auf Basis der Vorgängerarbeiten auszulegen und die Stabilität und Regelgüte mit verschiedenen Reglerparametern zu untersuchen. Der Einfluss von Entwurfparametern auf die „Regelbarkeit“ des Systems wird analysiert, sodass diese optimiert werden können. Außerdem wird der Einfluss des Beobachters getestet. Darauf aufbauend soll der Einfluss der Systemparameter auf die Regelbarkeit untersucht werden.

Außerdem sollen Verfahren zur Ermittlung von Trajektorien für Arbeitspunktwechsel implementiert und erweitert werden. Der Einfluss der Systemparameter wird zudem in Bezug auf die Berechnung der Trajektorien untersucht. Hierzu ist die bestehende Modellprädiktive Regelung für die Parameteruntersuchungen sowie den zukünftigen Einsatz als Instrument für die Trajektoriensuche weiterzuentwickeln. Um an die Arbeit von Fauvé [8] anzuknüpfen, soll eine Trajektorienfolgeregelung entworfen werden, mit der zu zeigen ist, dass die gefundenen Trajektorien stabilisiert werden können.

2 Modellierung

In diesem Kapitel wird die Modellierung des Gesamtsystems erläutert und auf dessen Implementierung in MATLAB und SIMULINK eingegangen.

2.1 Modell des Schlitten-Pendel-Systems

Die Modellierung des Schlitten-Pendel-Systems orientiert sich zunächst an den Modellen der vergangenen Arbeiten. Diese bezogen sich auf die Herleitung von [14]. Dabei gibt es die Variante *Kraftsystem*, das als Eingang die Kraft annimmt, welche am Schlitten wirkt, sowie das vereinfachte *Beschleunigungssystem*, das direkt die Beschleunigung des Schlittens als Eingang erhält.

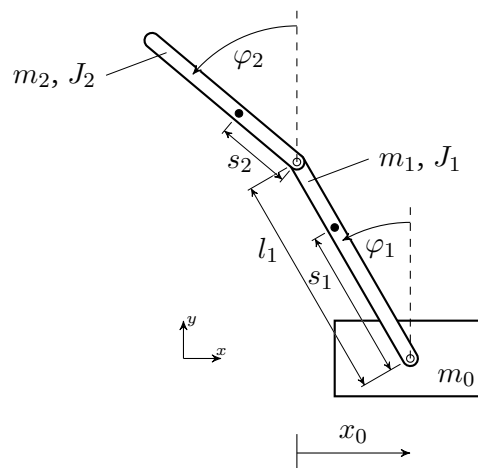


Abbildung 2.1: Doppelpendel

2.1.1 Koordinaten

Das System aus Schlitten und Doppelpendel hat 3 Freiheitsgrade, die mit den *Minimal-Koordinaten*

$$q_0 = x_0$$

$$q_1 = \varphi_1$$

$$q_2 = \varphi_2$$

beschrieben werden. Die Koordinaten sind nach Abbildung 2.1 definiert.

Die Schwerpunktskoordinaten der Pendelstäbe ergeben sich zu

$$\begin{aligned}x_1 &= x_0 - s_1 \sin \varphi_1 \\y_1 &= s_1 \cos \varphi_1 \\x_2 &= x_0 - l_1 \sin \varphi_1 - s_2 \sin \varphi_2 \\y_2 &= l_1 \cos \varphi_1 + s_2 \cos \varphi_2 .\end{aligned}$$

2.1.2 Herleitung der Bewegungsgleichungen

Um auf die Bewegungsgleichungen des Systems zu gelangen, wird in [14] der *Lagrange*-Formalismus verwendet. Dazu werden zunächst die kinetische und potentielle Energie des Gesamtsystems bestimmt sowie die nicht-konservativen Kräfte/Momente.

Die kinetische Gesamtenergie ergibt sich zu

$$T = \frac{1}{2}m_0 \dot{x}_0^2 + \frac{1}{2}m_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}m_2(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}J_1 \dot{\varphi}_1^2 + \frac{1}{2}J_2 \dot{\varphi}_2^2 \quad (2.1)$$

und die potentielle Energie beträgt

$$U = m_1 g y_1 + m_2 g y_2 . \quad (2.2)$$

Die nicht-konservativen Kräfte/Momente setzen sich aus der Antriebskraft F am Schlitten (in x_0 -Richtung) und den Reibungs-Kräften/Momenten zusammen und lauten

$$Q_0^* = F + F_d + F_c \quad (2.3a)$$

$$Q_1^* = M_{d1} + M_{c1} - M_{d2} - M_{c2} \quad (2.3b)$$

$$Q_2^* = M_{d2} + M_{c2} \quad (2.3c)$$

mit den viskosen Dämpfungen

$$F_d = -d_0 \dot{x}_0 \quad (2.4a)$$

$$M_{d1} = -d_1 \dot{\varphi}_1 \quad (2.4b)$$

$$M_{d2} = -d_2 (\dot{\varphi}_2 - \dot{\varphi}_1) \quad (2.4c)$$

sowie den Coulomb-Reibungen F_c , M_{c1} und M_{c2} , welche in Abschnitt 2.1.4 genauer betrachtet werden.

Mit dem Formalismus ergeben sich die 3 *gekoppelten* Bewegungsgleichungen für die Minimal-Koordinaten. Um nach den zweiten Ableitungen aufzulösen, muss allerdings noch das Gleichungssystem gelöst werden.

Damit ergeben sich die Bewegungsgleichungen

$$\ddot{x}_0 = f_{x_0}(\dots)$$

$$\ddot{\varphi}_1 = f_{\varphi_1}(\dots)$$

$$\ddot{\varphi}_2 = f_{\varphi_2}(\dots)$$

als Funktion der Minimal-Koordinaten, der Systemparameter und des Eingangs.

Die Berechnung der Ableitungen und das Lösen des Gleichungssystems geschah bisher händisch, was im Allgemeinen fehleranfällig ist. In dieser Arbeit werden die Bewegungsgleichungen mithilfe der *symbolischen Toolbox* von MATLAB gelöst. Die für den *Lagrange*-Formalismus benötigten Ableitungen werden symbolisch berechnet und das Gleichungssystem mit dem symbolischen Solver gelöst und vereinfacht. Diese Vorgehensweise ist nicht nur weniger fehleranfällig, dadurch lässt sich das System auch sehr flexibel modifizieren.

2.1.3 Zustandsraummodell

Um das Schlitten-Pendel-System als Zustandsraummodell (mit ausschließlich ersten Ableitungen) darzustellen, wird folgender Zustandsvektor definiert:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \varphi_1 \\ \dot{\varphi}_1 \\ \varphi_2 \\ \dot{\varphi}_2 \end{bmatrix} \quad (2.5)$$

Mit den Bewegungsgleichungen ergibt sich für das Kraftsystem (Eingangsgröße $u = F$) das folgende nichtlineare, eingangsaffine Zustandsraummodell:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (2.6)$$

$$= \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) \cdot F \quad (2.7)$$

$$\frac{d}{dt} \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \varphi_1 \\ \dot{\varphi}_1 \\ \varphi_2 \\ \dot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}, u) \\ f_2(\mathbf{x}, u) \\ f_3(\mathbf{x}, u) \\ f_4(\mathbf{x}, u) \\ f_5(\mathbf{x}, u) \\ f_6(\mathbf{x}, u) \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ a_2(\mathbf{x}) \\ \dot{\varphi}_1 \\ a_4(\mathbf{x}) \\ \dot{\varphi}_2 \\ a_6(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ b_2(\mathbf{x}) \\ 0 \\ b_4(\mathbf{x}) \\ 0 \\ b_6(\mathbf{x}) \end{bmatrix} \cdot F \quad (2.8)$$

Beim Beschleunigungssystem gilt für den Eingang $u = \ddot{x}_0 = a$, womit die Schlittenbeschleunigung direkt vorgegeben und die Dynamik des Schlittens umgangen wird. Damit wird die erste der drei gekoppelten Bewegungsgleichungen ersetzt. Das geänderte Gleichungssystem muss wieder gelöst werden und für das Zustandsraummodell folgt:

$$\frac{d}{dt} \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \varphi_1 \\ \dot{\varphi}_1 \\ \varphi_2 \\ \dot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ 0 \\ \dot{\varphi}_1 \\ a_4(\mathbf{x}) \\ \dot{\varphi}_2 \\ a_6(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ b_4(\mathbf{x}) \\ 0 \\ b_6(\mathbf{x}) \end{bmatrix} \cdot a \quad (2.9)$$

Tabelle 2.1 gibt Aufschluss über die Abhängigkeiten von Zuständen und Systemparametern. Grau hinterlegte Variablen sind nur im Kraftmodell vorhanden. Der Zustand x_0 hat keinen Einfluss auf die Systemdynamik.

Tabelle 2.1: Abhängigkeiten von Zuständen und Parametern

Zustände	$\dot{x}_0, \varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2$
Trägheitsparameter	m_0, m_1, m_2, J_1, J_2
Geometrieparameter	l_1, s_1, s_2
Reibungsparameter	$d_0, d_1, d_2, F_{c0}, M_{c10}, M_{c20}, \dot{x}_{0,c76}, \dot{\varphi}_{1,c76}, \dot{\varphi}_{2,c76}$
Umgebungskonstanten	g

Am Versuchsstand können die Schlittenposition und beide Pendelwinkel über Sensoren gemessen werden. Daher ergibt sich für den Ausgang beider Systeme:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \mathbf{C}\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_0 \\ \varphi_1 \\ \varphi_2 \end{bmatrix}. \quad (2.10)$$

Es handelt sich demnach um ein *SIMO-System* (single input multiple output).

2.1.4 Coulomb-Reibung

Neben der einfach zu beschreibenden viskosen Dämpfung (2.4) gibt es in der Realität komplexere Arten von Reibung, die die Systembeschreibung erschweren. Eine Übersicht zu verschiedenen Reibungsmodellen ist in [16] gegeben.

Da die Coulomb-Reibung sowohl des Schlittens als auch der Pendelstäbe einen wesentlichen Einfluss zu haben scheint, darf diese nicht vernachlässigt werden. In den bisherigen Modellierungen wurde höchstens die Coulomb-Reibung des Schlittens berücksichtigt. Bei der Neukonstruktion ist jedoch in den Gelenken eine höhere Coulomb-Reibung vorhanden (siehe Abschnitt 2.3.3), weswegen diese ebenfalls modelliert wird. Die eigentlich vorhandene Haftreibung wird nicht modelliert.

Die Formel der Gleitreibung lautet eigentlich

$$F_c = F_{c0} \cdot \text{sign}(\dot{x}_0),$$

allerdings führt diese Implementierung aufgrund der signum-Funktion (siehe Abbildung 2.2a) zu Komplikationen in der Simulation. In der Nähe des Vorzeichenwechsels ist die Funktion unendlich steil, was bei Nulldurchgängen in der Simulation problematisch ist, weil die Schrittweite dadurch sehr klein werden muss. Zwar kann man in SIMULINK die Option „Zero Crossing Detection“ abschalten, es kommt dann allerdings meistens zu einem „Rattern“. [12]

In [17] und [11] wurde der Verlauf der signum-Funktion bei sehr niedrigen Geschwindigkeiten mit der arctan Funktion angenähert (siehe Abbildung 2.2b).

$$F_c = F_{c0} \cdot \frac{2}{\pi} \arctan\left(\frac{\dot{x}_0}{150}\right)$$

Die Funktion nähert sich jedoch nur sehr langsam dem Endwert an. In dieser Arbeit wird daher die \tanh Funktion verwendet, welche deutlich früher den Endwert annimmt (siehe Abbildung 2.2c).

$$F_c = -F_{c0} \cdot \tanh\left(\frac{\dot{x}_0}{\dot{x}_{0,c76}}\right) \quad (2.11a)$$

$$M_{c1} = -M_{c10} \cdot \tanh\left(\frac{\dot{\varphi}_1}{\dot{\varphi}_{1,c76}}\right) \quad (2.11b)$$

$$M_{c2} = -M_{c20} \cdot \tanh\left(\frac{\dot{\varphi}_2 - \dot{\varphi}_1}{\dot{\varphi}_{2,c76}}\right) \quad (2.11c)$$

Dabei ist $\dot{x}_{0,c76}$ gerade die Geschwindigkeit, bei der $\tanh(1) = 0.7616$ von F_{c0} erreicht ist.

2.1.5 Ruhelagen

Die Ruhelagen bzw. Arbeitspunkte eines nichtlinearen Systems ergeben sich aus

$$\dot{\mathbf{x}}_R = \mathbf{f}(\mathbf{x}_R, u_R) = \mathbf{0}.$$

Für das Schlitten-Pendel-System gibt es aufgrund der Periodizität von Sinus und Kosinus theoretisch unendlich viele Ruhelagen. Es werden daher die folgenden 4 prinzipiell unterschiedlichen betrachtet:

Tabelle 2.2: Die 4 Ruhelagen

Arbeitspunkt	u	x_0	\dot{x}_0	φ_1	$\dot{\varphi}_1$	φ_2	$\dot{\varphi}_2$	Stabilität
1	0	x_{0R}	0	π	0	π	0	stabil
2	0	x_{0R}	0	π	0	0	0	instabil
3	0	x_{0R}	0	0	0	π	0	instabil
4	0	x_{0R}	0	0	0	0	0	instabil

Die Schlittenposition x_0 ist dabei beliebig, da der Schlitten an jeder Position eine Ruhelage annehmen kann. Die Reihenfolge der Arbeitspunkte entspricht den Pendelpositionen „von unten nach oben“.

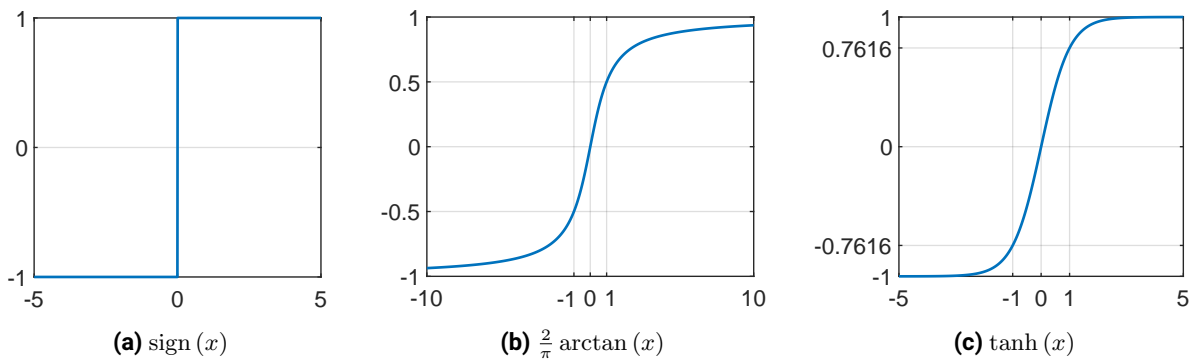


Abbildung 2.2: Annäherung signum-Funktion – Vergleich

2.1.6 Beschränkungen

Die Länge der Schlittenführung ist am realen Versuchsstand endlich, weswegen die Schlittenposition begrenzt ist. Es gilt für die Position des Schlittenmittelpunktes:

$$-0,8 \text{ m} \leq x_0 \leq 0,8 \text{ m}$$

2.2 Motor-Modell

Das Motormodell besteht aus den drei Baugruppen Spannungs-Strom-Wandler, Gleichstrommotor und Getriebe. Die Modellierung der Baugruppen basiert auf den in Franke [9] erstmalig aufgestellten Gleichungen, die auch in den nachfolgenden Arbeiten zum Versuchsstand zur Anwendung kamen. Das Modell des Gleichstrommotors wird im Rahmen dieser Arbeit nun zusätzlich um die Berücksichtigung der Gegeninduktion des Motors erweitert.

2.2.1 Spannungs-Strom-Wandler

Bei dem am Versuchsstand eingesetzten Spannungs-Strom-Wandler handelt es sich um einen Servoverstärker, der ursprünglich zur Drehzahlregelung von Gleichstrommotoren vorgesehen war. Entsprechend folgt der Verstärker dem Prinzip einer übergeordneten Drehzahlregelung mit unterlagerter Stromregelung. Für die Anwendung am Versuchsstand ist der übergeordnete Drehzahlregelkreis jedoch aufgetrennt und in einen Spannungs-Strom-Wandler umfunktioniert worden. Dieser dient zur Vorgabe eines konstanten Ankerstroms mittels Pulsweitenmodulation (PWM) der Zwischenkreisspannung des Wandlers proportional zur eingehenden Steuerspannung. Das stationäre Verhalten kann daher durch einen Proportionalitätsfaktor K_{UI} beschrieben werden.

$$I_a = K_{UI} \cdot U_{\text{Steuer}} \quad (2.12)$$

Gemäß Franke [9] lässt sich die Dynamik des Wandlers durch ein PT_1 -Glied modellieren, sodass sich für den Wandler im Laplace-Bereich die Gleichung

$$I_a(s) = \frac{K_{UI}}{1 + T_{UI} \cdot s} \cdot U_{\text{Steuer}}(s) \quad (2.13)$$

ergibt.

2.2.2 Gleichstrommotor

Bei dem verwendeten Motor handelt es sich um eine fremderregte Gleichstrommaschine, wobei die magnetische Erregung durch einen Permanentmagneten erzeugt wird [9]. Das elektromagnetische Drehmoment des Gleichstrommotors ist näherungsweise proportional zum Ankerstrom [4]. Hierbei wird vorausgesetzt, dass keine magnetische Sättigung vorliegt.

$$M_e = K_I \cdot I_a \quad (2.14)$$

Neben dem elektromagnetischen Drehmoment sind außerdem parasitäre Reibmomente zu berücksichtigen. Die Modellierung der Reibung von Motor und Getriebe wird zusammen mit der Schlittenreibung in Abschnitt 2.1 behandelt. Ein rückwirkendes Moment durch die Federkraft des Riemens wird auf Grund der Annahme unendlicher Riemensteifigkeit vernachlässigt.

Weiterhin wird das Drehmoment durch die Gegeninduktion geschwächt. Dieser Effekt ist in den Motormodellen von Franke [9] und den Nachfolgearbeiten bisher nicht berücksichtigt worden. Da die Erfahrung am realen Versuchsstand jedoch gezeigt hat, dass eine Modellierung der Gegeninduktion sinnvoll erscheint, wird diese im Rahmen dieser Arbeit in das Motormodell integriert.

Zum besseren Verständnis des Effekts wird zunächst das physikalische Prinzip der Gegeninduktion betrachtet. Fließt ein Strom durch den ruhenden Anker, der sich im Magnetfeld der Permanentmagneten befindet, so wirkt senkrecht zu den Richtungen des Stroms und des Magnetfelds die Lorentzkraft auf die in der Ankerwicklung befindlichen Ladungsträger (Drei-Finger-Regel bzw. Rechte-Hand-Regel in technischer Stromrichtung). Durch das entstehende Drehmoment beginnt der Anker zu rotieren. Auf Grund der Rotation bewegen sich die Ladungsträger nun zusätzlich zur eigentlichen Stromrichtung auch in Rotationsrichtung des Ankers. Auf diese Bewegungskomponente kann nun erneut das Prinzip der Lorentzkraft angewendet werden. Die resultierende zusätzliche Kraftkomponente, die in Abhängigkeit der Rotationsgeschwindigkeit auf die Ladungsträger wirkt, zeigt nun gegen die Stromrichtung (Lenz'sche Regel). Der resultierenden Ladungsbeschleunigung kann ein Spannungsabfall über der Ankerwicklung zugeordnet werden, die sogenannte induzierte Gegenspannung oder Gegen-EMK (Gegen-Elektromagnetische-Kraft). Der in Folge sinkende Ankerstrom reduziert das Drehmoment des Motors. [4]

Am Versuchsstand wird dieser Effekt bei geringen Winkelgeschwindigkeiten durch den Stromregler des Spannungs-Strom-Wandlers kompensiert. Ab einer bestimmten Winkelgeschwindigkeit bei konstantem Sollstrom wird die induzierte Gegenspannung größer als die maximale Zwischenkreisspannung des Spannungs-Strom-Wandlers, sodass nicht mehr ausgegletet werden kann. Nun nimmt der Ankerstrom und damit das Drehmoment mit steigender Winkelgeschwindigkeit ab bis die Leerlaufdrehzahl erreicht ist.

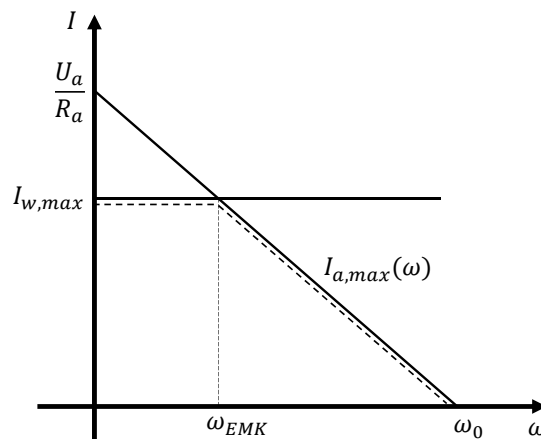


Abbildung 2.3: Stromkennlinie

Auf Grund der integrierten Strombegrenzung zum Schutz von Motor und Verstärker wird der vom Wandler bereitstellbare Strom zusätzlich begrenzt. Der dadurch parametrisierte Maximalstrom des Wandlers wird durch den Stromregler zunächst konstant gehalten bis die Strombegrenzung der Gegen-EMK ab

einer bestimmten Winkelgeschwindigkeit ω_{EMK} die des Wandler unterschreitet (siehe Abbildung 2.3). An dieser Stelle tritt ein „Knick“ im Stromverlauf auf. Für den Stromregler wird dabei vereinfachend angenommen, über ausreichend hohe Dynamik und Stellenergie zu verfügen, um den Strom bis zur EMK-Grenze zu jedem Zeitpunkt konstant halten zu können.

Es werden zunächst nur positive Winkelgeschwindigkeiten betrachtet. Der Abschnitt des Verlaufs in Abbildung 2.3 mit konstantem maximalen Strom wird durch

$$I_{a,\max} = I_{w,\max} = \text{const.}, \quad 0 \leq \omega \leq \omega_{\text{EMK}} \quad (2.15)$$

beschrieben, wobei $I_{w,\max}$ der Begrenzungsstrom des Wandlers ist.

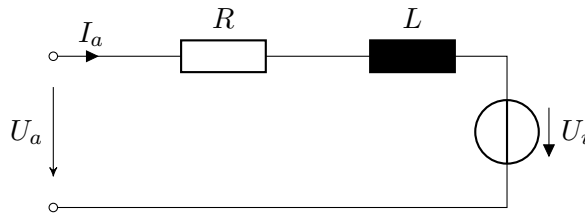


Abbildung 2.4: Ersatzschaltbild eines Gleichstrommotors

Der Verlauf des Bereichs mit zeitlich abnehmender maximaler Stromstärke wird aus dem Ersatzschaltbild des Gleichstrommotors in Abbildung 2.4 hergeleitet. Aus dem zweiten Kirchhoff'schen Gesetz (Maschenregel) ergibt sich

$$U_a = RI_a + L \frac{dI_a(t)}{dt} + U_i. \quad (2.16)$$

Für die Kennlinie des maximalen Stroms liegt der Tastgrad (engl.: *duty cycle*) der pulsweitenmodulierten Ankerspannung bei 100%, sodass

$$U_{a,\max} = \text{const.}$$

gilt. Die elektrische Zeitkonstante des Motors, die mit den Angaben des Datenblatts aus Franke [9] zu

$$\tau_{\text{el}} = \frac{L}{R} \approx 0,0023 \text{ s} \ll 1 \text{ s},$$

berechnet werden kann, ist ausreichend klein, dass näherungsweise von stationärem Betrieb ausgegangen werden kann und für die selbstinduzierte Spannung

$$L \frac{dI_a(t)}{dt} \approx 0$$

gilt. Die induzierte Gegenspannung wird gemäß [4] durch Proportionalität zur Winkelgeschwindigkeit

$$U_i = K_I \cdot \omega$$

modelliert. Damit kann Gleichung (2.16) zur Geraden

$$I_{a,\max}(\omega) = \frac{U_{a,\max}}{R} - \frac{K_I}{R} \omega, \quad \omega > \omega_{\text{EMK}} \quad (2.17)$$

umgeformt werden.

Zusammenfassend ergibt sich aus den Gleichungen (2.15) und (2.17) der Verlauf des maximal verfügbaren Stroms für $\omega > 0$

$$I_{a,\max}(\omega) = \begin{cases} I_{w,\max} & 0 \leq \omega \leq \omega_{\text{EMK}} \\ \frac{U_{a,\max}}{R} - \frac{K_I}{R}\omega, & \omega > \omega_{\text{EMK}} \end{cases},$$

wobei ω_{EMK} durch

$$\omega_{\text{EMK}} = \frac{U_{a,\max} - I_{w,\max}R}{K_I} \quad (2.18)$$

berechnet wird.

Die Beschreibung des Bereichs negativer Winkelgeschwindigkeiten kann aus den Gleichungen (2.15) und (2.17) durch eine Punktspiegelung des in Abbildung 2.3 dargestellten Verlaufs in den dritten Quadranten gewonnen werden. Im zweiten und vierten Quadranten (Bremsbetrieb) wird der Strom durch den Spannungs-Strom-Wandler begrenzt.

2.2.3 Getriebe

Die Rotorbewegung des Motors wird über das Getriebe und das Antriebszahnrad auf den Zahnriemen weitergegeben. Auf Grund der hohen Steifigkeit des mit eingebetteten Stahlseilen unterstützten Riemens wird dieser wie in Apprich [3] als unendlich starr angenommen, sodass die Kinematik zwischen der Winkelgeschwindigkeit des Motors und der Schlittengeschwindigkeit ohne Federkopplung durch

$$\dot{x}_0 = K_G \cdot r_{32} \cdot \omega \quad (2.19)$$

modelliert werden kann. Hierbei wird die Getriebeübersetzung K_G über das Zahnverhältnis

$$K_G = \frac{Z_{16}}{Z_{60}}$$

berechnet, während der Antriebsradius r_{32} als Abstand zwischen der neutralen Faser des Riemens und der Drehachse definiert ist. Für die Beziehung zwischen Antriebsmoment und Schlittenkraft ergibt sich der Zusammenhang

$$F = M \cdot \frac{1}{K_G} \cdot \frac{1}{r_{32}}.$$

2.3 Modellparameter

Bevor die in dieser Arbeit verwendeten Parametersätze für Motor- und Schlitten-Pendel-Modell vorgestellt werden, soll im folgenden Abschnitt zunächst ein Überblick über die Entwicklung der in den vergangenen Arbeiten verwendeten Modellparameter gegeben werden.

2.3.1 Begründeter Stand der Modellparameter

Die für die Modellierung erforderlichen Systemparameter des Versuchsstands wurden erstmalig 1997 von Franke [9] durch Messungen identifiziert. Die Antriebseinheit aus Spannungs-Strom-Wandler, Motor, Getriebe und Riemen ist seitdem nicht verändert worden. Daher repräsentieren die von Franke [9] identifizierten Modellparameter in Bezug auf die Antriebseinheit weiterhin den aktuellen Stand (siehe Abschnitt 2.3.2)

Während zuvor noch ein Einfachpendel verwendet worden war, konstruierte Apprich [3] 2009 erstmalig ein Doppelpendel für den Versuchsstand. Von den Änderungen betroffen war neben den Pendeln auch die Schlittenmasse, da auch der obere Teil des Schlittens neu konstruiert wurde. Die Modellparameter für die neuen Pendelstäbe wurden, anders als bei Franke [9], nicht gemessen, sondern aus dem CAD-Modell abgeleitet. Dies betrifft die Massen, Trägheitsmomente, Längen und Schwerpunkte der beiden Stäbe. Die Masse des Schlittens wurde von Franke [9] übernommen. Durch Messungen wurden lediglich die viskose und trockene Reibung des Schlittens gegenüber den Schienen erneut identifiziert.

Im selben Jahr wurde von Kämmerer [10] die viskose Lagerreibung d_1 zwischen Stab 1 und dem Schlitten als fehlender Modellparameter durch Messungen ergänzt. Die viskose Lagerreibung d_2 zwischen Stab 1 und Stab 2 wurde rechnerisch bestimmt, da das Lager gegen Ende der Arbeit getauscht werden musste. Die viskose Dämpfung des Schlittens, die von Apprich [3] zuvor gemessen worden war, wurde durch einen deutlich höheren Schätzwert ersetzt. Außerdem wurde erstmalig die Masse von Schlitten und Antrieb zu einer schlittenseitig wirkenden Gesamtmasse zusammengefasst und ebenfalls als Schätzwert ausgewiesen.

Die Reibwerte d_1 und d_2 wurden 2011 durch Kisner [11] erneut bestimmt. Durch die *Prediction-Error Minimization Method* aus der *System Identification Toolbox* von MATLAB wurden die Parameter d_1 und d_2 so variiert, dass die quadratische Fehlersumme minimal wird. Als Fehler ist hierbei die Differenz zwischen den gemessenen und den vom Modell vorhergesagten zeitlichen Winkelverläufen $\varphi_1(t)$ und $\varphi_2(t)$ bei ruhendem Schlitten und frei gewählter Anfangsauslenkung zu verstehen. Für die Optimierung wurden zudem nicht näher beschriebene Anfangsschätzwerte für d_1 und d_2 gewählt. Es wird davon ausgegangen, dass es sich um Erfahrungswerte handelt, da sie einerseits nicht mit den zuletzt von Kämmerer [10] bestimmten Werten übereinstimmen, jedoch andererseits zu guten Ergebnissen am Versuchsstand führten. Statt der optimierten Werte wurden in den Nachfolgearbeiten die Anfangsschätzwerte weiterverwendet. Der Reibwert d_0 der viskosen Schlittenreibung, der nicht Gegenstand der Optimierung war, wurde mit einem deutlich höheren Wert als bei Kämmerer [10] angegeben. Da keine explizite Begründung vorliegt, wird von einem Erfahrungswert ausgegangen. Die zuvor von Kämmerer [10] geschätzte effektive Gesamtmasse von Schlitten und Antrieb wurde nach unten korrigiert, wobei die Hintergründe für diesen Schritt ebenfalls nicht bekannt sind. Der neue Wert ist jedoch plausibel und wird daher ebenfalls als erfahrungsbasierter Schätzwert verstanden. Der ist in den weiteren Arbeiten nicht mehr verändert worden, sodass er als aktueller Stand zu betrachten ist. Bei Kisner [11] wurde erstmalig auch eine Begrenzung der Stellkraft von $F_{\max} = 400 \text{ N}$ bezüglich des Schlittens angegeben, die als gegeben dokumentiert ist.

2011 wurde außerdem von Noupa [15] sowohl die viskose als auch die trockene Schlittenreibung gemessen, wobei besonders für die viskose Reibung eine hohe Richtungsabhängigkeit beobachtet wurde. Die gemessenen Werte wurden in den weiteren Arbeiten jedoch nicht weiter beachtet.

Auf Grund eines Austauschs des Lagers von Stab 2 wurde 2014 von Brehl [5] eine erneute Identifikation der viskosen Dämpfungskonstanten d_2 messungsbasiert durchgeführt. Dabei wurden auch Länge und

Masse von Stab 2 gemessen, womit ebenfalls das Massenträgheitsmoment neu berechnet wurde. In den Nachfolgearbeiten wurden jedoch nur die Dämpfungskonstante weiterverwendet, während für Länge, Masse und Trägheitsmoment weiterhin die CAD-Werte von Apprich [3] verwendet wurden.

Chang [17] konstruierte im Sommersemester 2019 ein neues Doppelpendel, wobei Schlitten und Antrieb nicht verändert worden sind. Die neuen Modellparameter wurden wieder aus dem CAD-Modell abgeleitet. Die Angabe des Schwerpunkts s_2 von Stab 2 scheint in der Ausarbeitung jedoch zu fehlen. Die Dämpfungskonstanten d_1 und d_2 wurden von den Vorgängern übernommen, da bei der Konstruktion die gleichen Rillenkugellager gewählt wurden wie bei Apprich [3]. Für d_1 wurde der Schätzwert von Kisner [11] und für d_2 der Messwert von Brehl [5] übernommen, wobei in der Ausarbeitung die Werte von Apprich [3] genannt werden. Die viskose und die trockene Reibung des Schlittens wurden durch Messung selbst bestimmt. Wie bei Noupa [15] wurde bei der viskosen Reibung eine auffällige Richtungsabhängigkeit festgestellt. Die Berücksichtigung der Richtungsabhängigkeit mit einer Vorsteuerung führte jedoch zu einer Verschlechterung des Systemverhaltens. Daher wurde schließlich die linksseitige Dämpfungskonstante für beide Seiten übernommen. Da das neu konstruierte Pendel noch nicht für die weiteren Bestandteile der Arbeit, wie die Auslegung der Regelung und deren Erprobung am Versuchsstand, zur Verfügung stand, wurde weiterhin auf die CAD-Werte von Apprich [3] zurückgegriffen. Der Betrag der maximalen Stellkraft F_{\max} wurde zudem von dem von Kisner [11] zuletzt genannten Wert von 400 N auf 421 N erhöht. Der neue Wert lässt sich rechnerisch nachvollziehen, wie Abschnitt 2.3.2 entnommen werden kann.

Im Rahmen einer Verifikation der von Chang [17] angegebenen Modellparameter für das neue Doppelpendel, wurden im Wintersemester 2019/2020 durch Ribeiro [16] die Parameter aus dem CAD-Modell erneut abgeleitet. Da hierbei nicht genauer auf den Anlass der Neubestimmung eingegangen wurde, sich die neu bestimmten Parameter jedoch deutlich von den Vorherigen unterscheiden, wird angenommen, dass sich die von Chang [17] angegebenen Werte im Rahmen der Verifikation als fehlerbehaftet herausstellten. Darüber hinaus wurde durch Messung die Reibung der Pendelgelenke identifiziert. Hierbei wurde neben der viskosen Reibung erstmalig auch die trockene Reibung in den Gelenken ermittelt. Die Parameter von Ribeiro [16] zu den Pendelstäben sind somit aktueller Stand. Übernommen wurde weiterhin die von Kisner [11] geschätzte Gesamtmasse mit Schlitten und Antrieb. Für die maximale Stellkraft wurde im Gegensatz zur Vorgängerarbeit Chang [17] statt 421 N wieder 400 N angenommen. Es wird vermutet, dass dadurch eine Stellkraftreserve für die Regelung vorgehalten werden sollte.

2.3.2 Parameter des Motor-Modells

Gemäß Abschnitt 2.3.1 werden die Modellparameter für das Motormodell Franke [9] entnommen.

Die Steuerspannung U_{Steuer} am Eingang des Spannungs-Strom-Wandlers darf maximal ± 10 V betragen. Ein Betrag größer als 10 V sollte vermieden werden, da sonst der Impulsstrom über den zulässigen Wert ansteigen und der Stromregler zerstört werden kann.

Die Verstärkung K_{UI} des Spannungs-Strom-Wandlers kann durch ein Potentiometer bis zu einem Wert von etwa 2 A/V eingestellt werden, sodass mit der maximalen Steuerspannung der maximal zulässige Impulsstrom von 20 A erreicht wird. Die zuletzt dokumentierte Einstellung liegt bei

$$K_{UI} = 1,87 \frac{\text{A}}{\text{V}} .$$

Der Wert beinhaltet bereits eine Idealisierung, da Messungen von Franke [9] gezeigt haben, dass die reale Verstärkung am Versuchsstand eine leichte Richtungsabhängigkeit bezüglich des Vorzeichens aufweist.

Die Zeitkonstante T_{UI} wird als ausreichend klein angesehen, sodass die Dynamik des Wandlers vernachlässigt werden kann.

Alle weiteren verwendeten Parameter des Motormodells sind in Tabelle 2.3 aufgeführt.

Tabelle 2.3: Parameter – Motorsystem

Bezeichnung	Symbol	Einheit	Franke97
Maximale Steuerspannung	$U_{\text{Steuer,max}}$	V	10
Wanderverstärkung	K_{UI}	$\frac{\text{V}}{\text{A}}$	1,87
Wandlerzeitkonstante	T_{UI}	s	0,00075
Maximale Ankerspannung	$U_{a,\text{max}}$	V	65
Ankerwiderstand	R	Ω	0,9
Drehmomentkonstante	K_I	$\frac{\text{Nm}}{\text{A}}$	0,153
Getriebeübersetzung	K_G	-	$\frac{16}{60}$
Antriebsradius	r_{32}	m	0,0255

Mit der statischen Verstärkung zwischen Eingang U_{Steuer} und Ausgang F

$$K_{\text{MotorGain}} = K_{UI} \cdot K_I \cdot \frac{1}{K_G} \cdot \frac{1}{r_{32}} = 42,075 \frac{\text{N}}{\text{V}} \quad (2.20)$$

lässt sich die maximale Stellkraft

$$F_{\text{max}} = U_{\text{Steuer,max}} \cdot K_{\text{MotorGain}} = 420,75 \text{ N} \quad (2.21)$$

berechnen, wobei $K_{UI} = 1,87 \text{ A/V}$ ist. Für $K_{UI,\text{max}} = 2 \text{ A/V}$ läge die maximale Stellkraft bei $F_{\text{max}} = 450 \text{ N}$.

Mit $K_{UI} = 1,87 \text{ A/V}$ kann zudem

$$I_{w,\text{max}} = U_{\text{Steuer,max}} \cdot K_{UI} = 18,7 \text{ A}$$

berechnet werden, sodass sich mit Hilfe von Gl. (2.18) für die Drehzahl, ab der die Gegeninduktion als Strombegrenzung wirksam wird,

$$\omega_{\text{EMK}} = 314,84 \frac{\text{rad}}{\text{s}}$$

ergibt. Dies entspricht nach Gl. (2.19) einer Schlittengeschwindigkeit von $\dot{x}_0 = 2,14 \frac{\text{m}}{\text{s}}$

2.3.3 Parameter des Schlittendoppelpendels

Ausgehend von Abschnitt 2.3.1 werden in dieser Arbeit drei Parametersätze angelegt (siehe Tabelle 2.4). Dies ermöglicht einen Vergleich der unterschiedlichen Systeme hinsichtlich des Systemverhaltens und der Stabilisierbarkeit.

Aufgrund der Neukonstruktion des Doppelpendels und dabei aufgetretener Schwierigkeiten in der Regelung ist ein Vergleich zum vorigen System von Interesse. Die Parameter der alten Konstruktion werden mit „Apprich09“ bezeichnet, obwohl hier auch neu bestimmte Werte von Kisner [11] und Brehl [5] (Reibung und Schlittenträgheit) enthalten sind. Der zweite Parametersatz „Chang19“ stellt die Daten der Ausarbeitung von [17] dar, in welcher die Neukonstruktion stattfand. Da diese erst in der nächsten Arbeit (Ribeiro [16]) in Betrieb genommen wurde und dort sowohl die CAD-Parameter erneut bestimmt als auch die Reibungsparameter identifiziert wurden, werden die Werte „Ribeiro20“ als korrekte Parameter des neukonstruierten Doppelpendels betrachtet.

Tabelle 2.4: Parameter – Schlitten-Pendel-System

Bezeichnung	Symbol	Einheit	Apprich09	Chang19	Ribeiro20
Masse des Schlittens	m_0	kg	16,5	16,5	16,5
Masse des ersten Pendels	m_1	kg	0,615	0,329	0,8534
Masse des zweiten Pendels	m_2	kg	0,347	0,3075	0,3957
Trägheitsmoment des ersten Pendels	J_1	kg m^2	0,00647	0,01457	0,01128
Trägheitsmoment des zweiten Pendels	J_2	kg m^2	0,00407	0,00334	0,003343
Länge des ersten Pendels	l_1	m	0,2905	0,325	0,282
Länge des zweiten Pendels	l_2	m	0,3388	0,305	0,280
Schwerpunkt des ersten Pendels	s_1	m	0,0775	0,1425	0,09373
Schwerpunkt des zweiten Pendels	s_2	m	0,146	0,114254	0,114254
Viskose Dämpfung des Schlittens	d_0	$\frac{\text{Ns}}{\text{m}}$	17	17,6	17
Viskose Dämpfung des ersten Pendels	d_1	$\frac{\text{Nm s}}{\text{rad}}$	0,0091	0,005	0,00768
Viskose Dämpfung des zweiten Pendels	d_2	$\frac{\text{Nm s}}{\text{rad}}$	0,0006905	0,005	0,000285
Coulomb-Reibung des Schlittens	F_{c0}	N	16,232	17,5	13,43
Coulomb-Reibung des ersten Pendels	M_{c10}	Nm	0	0,0538	0,0538
Coulomb-Reibung des zweiten Pendels	M_{c20}	Nm	0	0,0000912	0,0000912
Erdbeschleunigung	g	$\frac{\text{m}}{\text{s}^2}$	9,81	9,81	9,81

Es wird somit im Folgenden zwischen den „alten Parametern“ (Apprich09) und den „neuen Parametern“ (Ribeiro20) unterschieden. Der größte Unterschied zwischen beiden Systemen ist die Coulomb-Reibung in den Pendelgelenken, die zusätzlich zur viskosen Reibung erstmals bei Ribeiro [16] bestimmt wurde. Da bei der Neukonstruktion die Messsignalübertragung zur Vermeidung einer Kabelaufwicklung über einen Schleifring realisiert wurde, besteht die Vermutung, dass dieser für die erhöhte Coulomb-Reibung verantwortlich ist. Dies macht das System bereits um einen sehr kleinen Arbeitsbereich nichtlinear, was die Regelung erschwert.

Für die Skalierungsparameter $\dot{x}_{0,c76}$, $\dot{\varphi}_{1,c76}$ und $\dot{\varphi}_{2,c76}$ der Coulomb-Reibung (siehe Abschnitt 2.1.4) wird jeweils 0,01 angenommen.

2.4 Aufbau des Simulationsmodells

In dieser Arbeit wird das Doppelpendel-System rein simulativ betrachtet. Das Simulationsmodell wird im Vergleich zu Vorgängerarbeiten systematischer und detaillierter aufgebaut. Dadurch sind umfangreiche und automatisierte Tests möglich, mit denen das System umfassend untersucht werden kann.

Besonders wird in dieser Arbeit Wert auf einen strukturierten, modularen Aufbau gelegt. Dies soll flexible Änderungen am System, wie z. B. Systemparameter oder Reglerparameter ermöglichen. Von der Berechnung der Bewegungsgleichungen über die Parametrisierung des Systems bis zur Reglerberechnung kann das Modell vollständig und konsistent initialisiert werden. Außerdem soll dadurch die Nutzbarkeit und Wiederverwendbarkeit in zukünftigen Arbeiten gewährleistet sein.

Das Simulationsmodell voriger Arbeiten wie [17] ist nur wenig strukturiert aufgebaut, enthält Redundanzen und viele fest-kodierte Parameter. Es wird daher lediglich zur Orientierung verwendet. Hart-kodierte Parameter werden in dieser Arbeit vermieden, sodass jeder Parameter bei der Initialisierung geändert werden kann (ohne das SIMULINK-Modell ändern zu müssen). Auch Gleichungen können variabel ausgelegt werden, was durch die symbolische Schreibweise von MATLAB ermöglicht wird. Code-Dopplungen werden vermieden. Die gesamte Initialisierung ist stark funktionalisiert, da dies für die Variationstests notwendig ist. Dadurch wird auch sichergestellt, dass Änderungen nur an einer Stelle vorgenommen werden müssen und automatisch direkt an allen entsprechenden Stellen angepasst werden.

Die SIMULINK-Modelle und MATLAB-Funktionen zur Modellierung des Systems und Initialisierung des Simulationsmodells befinden sich im Ordner `Model1`.

2.4.1 Aufbau in SIMULINK

Subsysteme

In SIMULINK lassen sich sogenannte *Subsysteme* erstellen, um ein Simulationsmodell übersichtlicher zu gestalten. Diese Subsysteme (im Folgenden auch Module genannt) lassen sich zudem als Datei externalisieren (*Referenced Subsystem*), wodurch sie einerseits separat bearbeitet werden können und andererseits an mehreren Stellen modular wiederverwendet werden können. Somit wird lediglich in der obersten Schicht (die Testebene) ein *SIMULINK-Model* verwendet.

In dieser Arbeit werden immer Module erstellt, sofern es sinnvoll erscheint. Dadurch wird das Gesamtsystem übersichtlich gehalten und kann sehr flexibel modifiziert werden. Die einzelnen Module (wie z. B. Motor, Gesamtmodell, Zustandsregler) stellen zudem wiederverwendbare Teile dar und können so an verschiedenen Stellen im Projekt eingebaut werden, aber möglicherweise auch am SIMULINK-Modell des Versuchsstandes zum Einsatz kommen. So könnte relativ einfach derselbe Regler an dem echten und dem simulierten System eingesetzt und verglichen werden.

Maske

Bei Subsystemen kann eine sogenannte *Maske* eingerichtet werden, wodurch Parameter übergeben werden können. In deren Abhängigkeit können in einem Subsystem auch initialisierende Berechnungen durchgeführt werden.

Grundsätzlich können in SIMULINK alle Variablen aus dem globalen MATLAB-Workspace verwendet werden, was allerdings fehleranfällig ist. Geschickter ist es, dem Simulationsmodell eine einzige Variable vom Typ `struct` zu übergeben, in der alle Parameter vorhanden sind. So werden jedem Modul über die jeweilige Maske nur genau die Parameter übergeben, von denen es abhängt. Dadurch bleibt der Gesamtaufbau modularer und strukturierter.

Scopes und ToWorkspace

Um die Signalverläufe direkt zu untersuchen, werden an allen wichtigen Stellen in den Simulationsmodellen *Scopes* installiert. So lässt sich das Systemverhalten oder Probleme in der Regelung schnell und einfach analysieren. Oft werden *Scopes* mit mehreren Eingängen verwendet, um Signale besser vergleichen zu können.

Für alle Variablen, die später in MATLAB für weitere Auswertungen zur Verfügung stehen sollen, werden *ToWorkspace*-Blöcke verwendet. Diese geben die Daten an die Ausgabevariable *out*.

Für die Screenshots werden diese Blöcke meist aus Übersichtsgründen entfernt.

2.4.2 SIMULINK Module und Modelle

Motor.slx

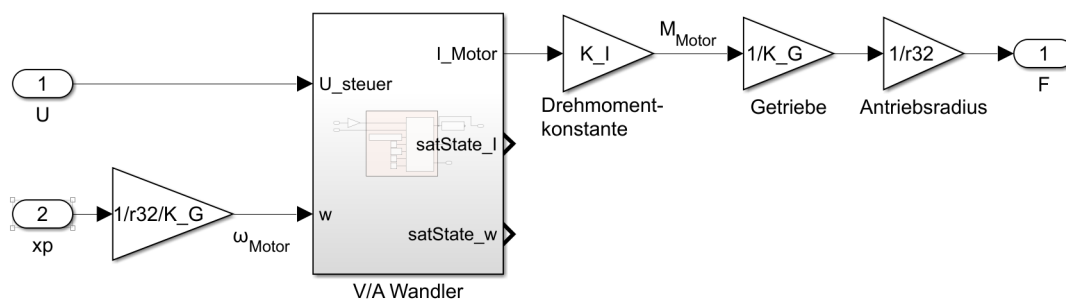


Abbildung 2.5: Motormodell in SIMULINK

Das Motormodell wird Abschnitt 2.2 entsprechend als Modul implementiert (siehe Abbildung 2.5). Eingangsgröße ist die Steuerspannung sowie die Schlittengeschwindigkeit, da sich daraus die Induktionsspannung ergibt. Ausgangsgröße ist die Kraft, die am Schlitten wirkt. Der V/A Wandler wird als Subsystem implementiert und gibt den Motorstrom aus (siehe Abbildung 2.6). Die Berechnung der Strombegrenzung (siehe Abschnitt 2.2.2) wird als *matlabFunction* realisiert. Dabei werden die Signale *satState_w* (falls die Sättigungsdrehzahl erreicht ist) und *satState_I* (falls der Sollstrom nicht erreicht werden kann) gesetzt und können später analysiert werden. Die Tiefpass-Charakteristik der Induktivität wird mit dem PT_1 -Glieder (2.13) dargestellt.

SchlittenPendel.slx

Das Zustandsraummodell des Schlitten-Pendel-Systems wird mit einer *S-Function* dargestellt. Die zugehörige MATLAB-Datei *SchlittenPendelFunc.m* wird automatisch bei der Initialisierung erstellt (siehe Abschnitt 2.4.4). Lediglich die Anfangswerte werden über die Simulationsparameter übergeben.

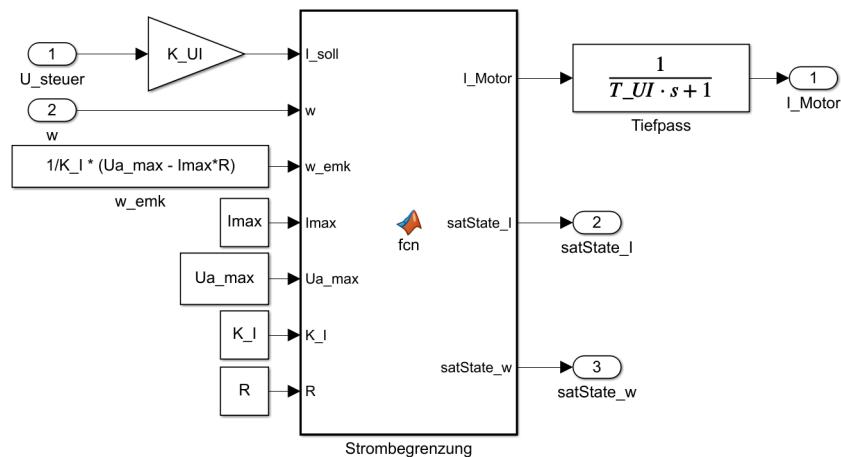


Abbildung 2.6: V/A Wandler in SIMULINK

Gesamtmodell.slx

Dieses Modul stellt das gesamte Modell dar und fasst die beiden vorigen Module zusammen (siehe Abbildung 2.7). Die Eingangsspannung wird an den Motor gegeben und dessen Ausgang F ist die Schnittstelle zum Schlitten-Doppelpendel. Die Schlittengeschwindigkeit wird zum Motor zurückgeführt, da durch diese die Motordrehzahl bestimmt wird.

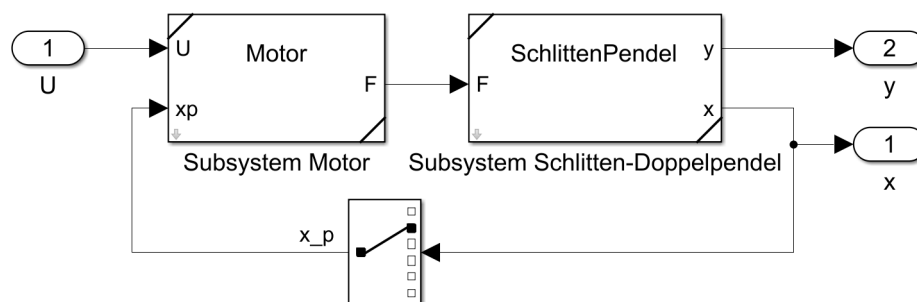


Abbildung 2.7: Gesamtmodell in SIMULINK

Gesamtmodell_Test.slx

Dies ist ein SIMULINK-Modell und bindet das Gesamtmodell-Modul ein. Auf dieses können testweise verschiedene Eingangsverläufe gegeben werden, um das Systemverhalten auf Plausibilität zu überprüfen, beispielsweise:

- Keine Spannung, aber Anregung durch Anfangsauslenkung
- Konstante Spannung
- Sinusförmige Spannung

Da die durchgeführten Tests am Modell plausible Ergebnisse zeigen, wird im Folgenden von der Korrektheit des Simulationsmodells ausgegangen.

2.4.3 Parameter

Größere zusammenhängende Parameterdaten werden als Datei bzw. Funktion ausgelagert, um eine einfache Austauschbarkeit zu erreichen. Die drei Parametersätze des Schlitten-Pendels (siehe Tabelle 2.4) sowie die Motorparameter befinden sich im Ordner `Modellparameter`.

2.4.4 Initialisierung

Zum Initialisieren des Modells und der Simulation führt man das Skript `Init.m` aus. Dadurch werden folgende 4 Schritte durchgeführt:

1. **InitSymEq:** Berechnet symbolisch die Bewegungsgleichungen nach Abschnitt 2.1.2 für das Kraftsystem (`SchlittenPendelSymF`) und das Beschleunigungssystem (`SchlittenPendelSymA`) und speichert sie als globale Variablen. Auch die Ruhelagen werden initialisiert (definiert in `SchlittenPendelRuhelagen`).
2. **InitParams:** Initialisiert die Parameter des Motors und des Schlittens (standardmäßig `Ribeiro20`) als globale Variablen. Hier können die Parameter ausgetauscht oder verändert werden (z. B. Coulomb-Reibung abschalten).
3. **InitSystem(SchlittenPendelParams):** Diese Funktion erstellt die Zustandsraummodelle des Schlitten-Pendel-Systems nach Abschnitt 2.1.3. Dafür wird die Funktion `SchlittenPendelNLZSR.m` genutzt, welche das Zustandsraummodell in Abhängigkeit der Bewegungsgleichungen und der Parameter erstellt. Aus dem Kraftsystem wird mittels `sys2sfct` eine *S-Function* generiert und in `SchlittenPendelFunc.m` gespeichert.
4. **InitSim:** Bereitet die globale Variable `simparams` vor, in welcher alle Parameter für das SIMULINK-Modell übergeben werden. Dazu gehören beim Modell die Motorparameter und die Startwerte des Schlitten-Pendel-Systems.

Nach der Initialisierung kann `Gesamtmodell_Test.slx` ausgeführt werden.

2.4.5 Auswertung

Nach einer Simulation ist es nützlich, die wesentlichen Ergebnisse direkt ablesen zu können (z. B. ob der Schlitten außerhalb der Begrenzung war oder ob der Motor in Sättigung war). Außerdem wird eine Plot-Funktion implementiert, die die wichtigsten Verläufe darstellt. Um das Verhalten des Doppelpendels besser interpretieren zu können, wird zudem eine Animationsfunktion erstellt, die direkt das Pendelverhalten visualisiert.

Die entsprechenden MATLAB-Funktionen sind im Ordner `Auswertung` vorhanden. Weitere Informationen und Kommentare finden sich auch meist in den MATLAB-Funktionen selbst.

plot_outputs

Erstellt ein Diagramm, das die drei Verläufe der Ausgangsgrößen (inklusive deren Schätzwerte) und die Stellgröße (Soll- und Ist-Wert) darstellt. Der Funktion können einige Parameter übergeben werden, sodass das Diagramm auch abgespeichert werden kann. Für weitere Informationen sei auf die Dokumentation in der Datei `plot_outputs.m` verwiesen.

animate_outputs

Stellt die Bewegungen des Schlittens und des Doppelpendels zeitlich dar. Parameter für die Bildwiederholrate (FPS) und ein Zeitfaktor können angegeben werden. Außerdem kann das Video gespeichert werden. Für mehr Informationen siehe `animate_outputs.m`.

plotanimate

Führt die beiden obigen Funktionen aus, sodass Ergebnisse mit einem Funktionsaufruf gemeinsam gespeichert werden können. Beispiel:

```
plotanimate(out, 'u=0,AP4,phi2=0.01', 'Modell Tests', 60, 1/4 )
```

3 Arbeitspunkt-Regelung

Nachdem im letzten Kapitel die Modellierung des Gesamtsystems erläutert wurde, stellt sich nun die Frage, wie sich das System regeln lässt. Dabei geht es einerseits um die Regelung an den 4 Arbeitspunkten (siehe Abschnitt 2.1.5), d. h. das Stabilisieren und Halten dieser, sowie um Trajektorien, die das Doppelpendel von einem Arbeitspunkt in einen anderen überführen (siehe folgendes Kapitel 4).

Alle Arbeitspunkt-Regelungen vergangener Arbeiten basieren auf einer Linearisierung und der anschließenden Auslegung eines *Riccati*-Reglers. Die Güteparameter wurden dabei jedes Mal anders und meist heuristisch bestimmt. Da am Versuchsstand nicht alle Zustände gemessen werden können, wurden meistens Beobachter mit Polplatzierung ausgelegt. Kämmerer [10] verglich diesen mit einem Zustandsvariablenfilter, mit welchem allerdings keine zufriedenstellende Ergebnisse gelangen. Apprich [3] und Kämmerer [10] setzten Störgrößenbeobachter ein, die in den folgenden Arbeiten nicht mehr verwendet wurden. Alle Arbeiten setzten Vorsteuerungsmethoden ein, da die Coulomb-Reibung am Schlitten bei der Regelung stets problematisch war. Chang [17] erstellte ein Simulationsmodell zur Arbeitspunkt-Regelung, das als Orientierung dient.

Beim Reglerentwurf wird zunächst dem Vorgehen vergangener Arbeiten gefolgt. Einzelne Aspekte in der Steuerung/Regelung werden anschließend verbessert. Die Reglerparameter der vorigen Arbeiten werden überprüft und optimiert. Außerdem wird der Einfluss des Beobachter analysiert.

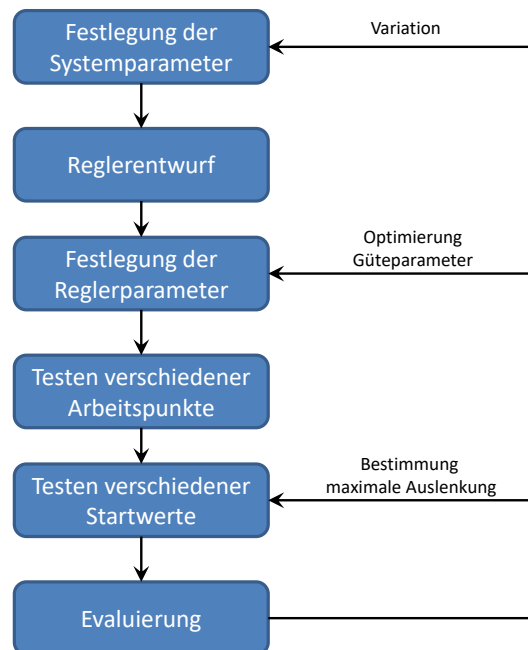


Abbildung 3.1: Vorgehen bei der AP-Regelung

Abbildung 3.1 stellt das prinzipielle Vorgehen bei der Untersuchung und Optimierung der Arbeitspunkt-Regelung dar. Es werden zunächst Auswertungsmethoden implementiert, die einzelne Simulationstests evaluieren. Mit den Anfangswerttests wird das Ermitteln der maximalen Winkelabweichung, ab welcher das System nicht mehr stabilisiert werden kann, automatisiert. Dadurch können im nächsten Schritt die Reglerparameter systematisch optimiert werden. Kann dadurch keine gute Regelung erzielt werden, müssen beim Reglerentwurf möglicherweise andere Verfahren eingesetzt werden. Abschließend wird die Regelbarkeit in Abhängigkeit der Systemparameter untersucht.

Regler können nach verschiedenen Gesichtspunkten wie Dynamik, Überspringen, Energieverbrauch optimiert werden. In dieser Arbeit wird bei der Untersuchung und Optimierung des Doppelpendels stets die Stabilität priorisiert. Die Optimierung des Einschwingvorgangs steht nicht im Vordergrund. Daher werden vor allem die instabilen Arbeitspunkte 2, 3 und 4 untersucht. Außerdem werden hauptsächlich die maximalen *Winkelabweichungen* betrachtet, da diese wesentlich für die Stabilität sind. Die Regelung von Positionsabweichungen wird nachrangig behandelt.

3.1 System-Linearisierung und Analyse

Das nicht-lineare System wird zunächst linearisiert, dessen Eigenwerte analysiert, sowie auf Steuer- und Beobachtbarkeit überprüft.

3.1.1 Linearisierung

Da das Zustandsraummodell des Schlitten-Pendel-Systems (2.7) nicht-linear ist, muss es um jeden Arbeitspunkt linearisiert werden, bevor lineare Regelungsmethoden angewandt werden können.

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, u_R)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_R} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial a_2(\mathbf{x})}{\partial \dot{x}_0} & \frac{\partial a_2(\mathbf{x})}{\partial \dot{\varphi}_1} & \frac{\partial a_2(\mathbf{x})}{\partial \dot{\varphi}_2} & \frac{\partial a_2(\mathbf{x})}{\partial \ddot{\varphi}_1} & \frac{\partial a_2(\mathbf{x})}{\partial \ddot{\varphi}_2} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{\partial a_4(\mathbf{x})}{\partial \dot{x}_0} & \frac{\partial a_4(\mathbf{x})}{\partial \dot{\varphi}_1} & \frac{\partial a_4(\mathbf{x})}{\partial \dot{\varphi}_2} & \frac{\partial a_4(\mathbf{x})}{\partial \ddot{\varphi}_1} & \frac{\partial a_4(\mathbf{x})}{\partial \ddot{\varphi}_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\partial a_6(\mathbf{x})}{\partial \dot{x}_0} & \frac{\partial a_6(\mathbf{x})}{\partial \dot{\varphi}_1} & \frac{\partial a_6(\mathbf{x})}{\partial \dot{\varphi}_2} & \frac{\partial a_6(\mathbf{x})}{\partial \ddot{\varphi}_1} & \frac{\partial a_6(\mathbf{x})}{\partial \ddot{\varphi}_2} \end{bmatrix}_{\mathbf{x}=\mathbf{x}_R} \quad (3.1)$$

$$\mathbf{B} = \left. \frac{\partial \mathbf{f}(\mathbf{x}_R, u)}{\partial u} \right|_{u=u_R} = \mathbf{b}(\mathbf{x}_R) = \begin{bmatrix} 0 \\ b_2(\mathbf{x}_R) \\ 0 \\ b_4(\mathbf{x}_R) \\ 0 \\ b_6(\mathbf{x}_R) \end{bmatrix} \quad (3.2)$$

Somit ergeben sich für jeden Arbeitspunkt unterschiedliche Systemmatrizen \mathbf{A} und Eingangsmatrizen \mathbf{B} . Beim Beschleunigungssystem fällt zusätzlich die zweite Zeile und die zweite Spalte (bis auf die 1) weg, da dort $a_2 = 0$ ist und keine Abhängigkeit von \dot{x}_0 besteht (siehe Tabelle 2.1), außerdem ist $b_2 = 1$. Die Ausgangsmatrix \mathbf{C} ist immer gleich, da die Ausgangsgleichung (2.10) linear ist.

Für den tatsächlichen Zustand wird der Arbeitspunkt-Zustand des linearisierten Systems Δx zum Arbeitspunkt addiert:

$$x(t) = x_R + \Delta x(t) \quad (3.3a)$$

$$u(t) = u_R + \Delta u(t) \quad (3.3b)$$

3.1.2 Eigenwerte

Anhand der Eigenwerte können Aussagen über die Dynamik eines Systems getroffen werden. Das Beschleunigungssystem hat aufgrund der zweifachen Integration des Eingangs \ddot{x}_0 stets zwei Eigenwerte in 0. Mit den Apprich-Parametern ergeben sich folgende Eigenwerte:

Tabelle 3.1: Eigenwerte des Beschleunigungssystems (Parameter: Apprich)

AP1	AP2	AP3	AP4
0	0	0	0
0	0	0	0
$-0.2972 + 11.4177j$	7.8028	6.8596	11.1308
$-0.2972 - 11.4177j$	-7.9367	-7.2314	-11.7251
$-0.0418 + 4.8515j$	$-0.1858 + 7.0392j$	$-0.0669 + 7.8676j$	4.8089
$-0.0418 - 4.8515j$	$-0.1858 - 7.0392j$	$-0.0669 - 7.8676j$	-4.8926

Im Arbeitspunkt 1 ergeben sich zwei konjugiert-komplexe Polpaare, die sich physikalisch mit der Schwingung der beiden Pendel begründen lassen. Sie befinden sich aufgrund der Dämpfung in der linken s-Halbebene, es ist der einzige stabile Arbeitspunkt. Arbeitspunkt 2 und 3 besitzen jeweils ein konjugiert-komplexes Polpaar, da dort das erste bzw. zweite Pendel nach unten zeigt und damit „stabil“ ist. Der Realteil ist bei Arbeitspunkt 3 deutlich kleiner, was auf die geringere Dämpfung des zweiten Pendelgelenks zurückzuführen ist (siehe Tabelle 2.4). Aufgrund der Instabilität des anderen Pendels ergibt sich jeweils ein positiv reeller Eigenwert. In Arbeitspunkt 4 stehen beide Pendel oben, weswegen es dort zwei Eigenwerte in der rechten s-Halbebene gibt.

Tabelle 3.2: Eigenwerte des Beschleunigungssystems (Parameter: Ribeiro)

AP1	AP2	AP3	AP4
0	0	0	0
0	0	0	0
-174.3727	-172.9211	-173.2473	-175.4362
-0.3493	-0.3494	0.3471	0.3471
$-0.6385 + 7.1578j$	6.7715	$-0.6443 + 7.2040j$	6.7469
$-0.6385 - 7.1578j$	-7.6900	$-0.6443 - 7.2040j$	-7.6570

Mit den neuen Parametern (Ribeiro) ergibt sich ein anderes Bild (siehe Tabelle 3.2). Bei Arbeitspunkt 1 und 2 verschwindet ein Polpaar, welches dem ersten Pendel zugeordnet wurde. Ein sehr schneller Eigenwert kommt hinzu, außerdem ein langsamer reeller, der bei Arbeitspunkt 3 und 4 positiv ist.

Die Ursache für diese Änderung liegt vermutlich in der Modellierung der Coulomb-Reibung (siehe Abschnitt 2.1.4). Bei der Linearisierung um die Ruhelage wird die Annäherungsfunktion der signum-Funktion um 0 linearisiert, wo sie sehr steil ist. Dies resultiert in einer sehr hohen Dämpfung. Daher ist es sinnvoll, die Coulomb-Reibung der Pendelstäbe für den AP-Reglerentwurf zu vernachlässigen. Wird die Coulomb-Reibung zu 0 gesetzt, ergeben sich prinzipiell ähnliche Werte wie in Tabelle 3.1.

3.1.3 Steuerbarkeit und Beobachtbarkeit

Voraussetzung für eine Regelung ist die Steuerbarkeit des Systems. Diese kann für lineare Systeme oder linearisierte Systeme an einem Arbeitspunkt mit dem Steuerbarkeitskriterium nach KALMAN überprüft werden [1]. Außerdem sollte ein System beobachtbar sein, falls nicht alle Zustände bekannt sind und daher über einen Beobachter ermittelt werden.

Das Schlitten-Pendel-System (Beschleunigungssystem) ist an allen 4 Arbeitspunkten vollständig steuer- und beobachtbar.

3.2 Aufbau der Regelung

Die Aufgabe eines Reglers bzw. einer (Vor-)Steuerung ist es nun, anhand der Messgrößen $\mathbf{y} = (x_0 \ \varphi_1 \ \varphi_2)^T$ (2.10) eine geeignete Steuerspannung U_{Steuer} an den Motor vorzugeben. Dabei teilt sich die Steuerung/Regelung in mehrere Schritte auf.

Die eigentliche Regelung bildet ein Zustandsregler, welcher entweder am Kraftsystem oder am Beschleunigungssystem ausgelegt ist. Dessen Stellgröße kann im ersten Fall (F_{Soll}) direkt an die Motorvorsteuerung gegeben werden. Beim Beschleunigungssystem muss die Ausgangsstellgröße a_{Soll} in einem weiteren Block zu einer Sollkraft bestimmt werden. Diese Aufteilung in einen Arbeitspunktregler am einfacheren Beschleunigungssystem und eine unterlagerte Geschwindigkeitsregelung, um der störenden Reibung am Schlitten zu begegnen, hat sich in vergangenen Arbeiten als sinnvoll herausgestellt.

3.2.1 Zustandsregler

Die Arbeitspunkt-Regelung wird mit einem Zustandsregler realisiert ($u = -\mathbf{K}\hat{\mathbf{x}}$). Dieser regelt ein System grundsätzlich in den Ursprung, es muss also noch eine Zustandstransformation durchgeführt werden. Für jeden der Arbeitspunkte (2.1.5) wird der Zustand, der für den Zustandsregler die Endlage darstellt, vorher von den Messgrößen abgezogen. Nach Berechnung der Stellgröße und Schätzung des Zustands wird der Arbeitspunkt entsprechend (3.3) wieder dazu addiert.

Die Verstärkung \mathbf{K} wird durch Lösen des LQ-Problems bestimmt [1]. Parameter für den Entwurf sind somit die positiv-definite Matrix $\mathbf{Q} = \text{diag} (Q_1 \ Q_2 \ Q_3 \ Q_4 \ Q_5 \ Q_6)$ zur Bewertung der Zustände und R für die Stellgröße. Da es sich jeweils um ein *linear-zeitinvariantes System* (LZI) handelt, ist die Matrix \mathbf{P} aus der *Riccati*-Gleichung, und damit auch \mathbf{K} , konstant.

Für jeden Arbeitspunkt muss ein eigener Regler entworfen werden, da jedem Arbeitspunkt ein anderes lineares System zugrunde liegt (siehe Abschnitt 3.1.1). Der AP-Regler ist somit nur in der Umgebung der Ruhelage gültig und kann zu große Abweichungen aufgrund der Nichtlinearität möglicherweise

nicht ausregeln. Außerdem sind für jeden Arbeitspunkt andere Güteparameter \mathbf{Q} und R optimal. Die systematische simulative Optimierung erfolgt in Abschnitt 3.5.

Das Lösen der *Riccati*-Gleichung und die Berechnung von \mathbf{K} für alle Arbeitspunkte erfolgt automatisiert in MATLAB.

3.2.2 Zustandsermittlung

Ein Zustandsregler benötigt zur Regelung zu jedem Zeitpunkt die Information aller Zustände \mathbf{x} (2.5) des Systems. Beim Schlitten-Pendel-System werden die drei Zustände x_0 , φ_1 und φ_2 gemessen. Deren Ableitungen, die Zustände \dot{x}_0 , $\dot{\varphi}_1$ und $\dot{\varphi}_2$ müssen noch ermittelt werden. Dabei wird von drei Möglichkeiten ausgegangen, die in der Simulation miteinander verglichen werden können:

1. Zustandsmessung
2. Beobachter
3. Differenzieren

Am Versuchsstand erfolgt die Zustandsermittlung durch einen Beobachter.

Zustandsmessung

Bei der Zustandsmessung wird von einer idealen Messung aller Zustände ausgegangen, d. h. $\hat{\mathbf{x}} = \mathbf{x}$. Diese Variante wird in der Simulation meist zuerst eingesetzt, um die Regelung des Systems besser beurteilen zu können und spezielle Probleme durch die Zustandsermittlung im Nachhinein analysieren zu können. Wenn nicht anders angegeben, wird von dieser Variante ausgegangen. Am realen Versuchsstand kann diese Methode nicht eingesetzt werden.

Beobachter

Der Schätzzustand $\hat{\mathbf{x}}$ wird durch einen *Luenberger-Beobachter* ermittelt. Dieser „simuliert“ ein lineares Schätzsystem (\mathbf{A} , \mathbf{B} , \mathbf{C}) und regelt Unterschiede in den Ausgangsgrößen \mathbf{y} und $\hat{\mathbf{y}}$ mit der Beobachter-Matrix \mathbf{L} aus [1]. Wie beim Zustandsregler muss auch hier die Auslegung für jeden Arbeitspunkt separat erfolgen. Damit gilt allerdings auch, dass der Beobachter nur in der Nähe dieser Ruhelage gültig ist, bei zu großen Abweichungen kann er aufgrund der Linearisierung die Zustände nicht mehr richtig schätzen.

\mathbf{L} wird in dieser Arbeit mittels Polplatzierung ausgelegt. Einflussparameter sind somit die Pole des Beobachters \mathbf{p}_b . Diese sollten üblicherweise weiter links liegen als die Pole des geregelten Systems \mathbf{p}_r . Sie können aber nicht beliebig weit links liegen, da sonst Messrauschen verstärkt wird.

In [5] werden sie beispielsweise hintereinander auf der negativen reellen Achse verteilt:

$$\mathbf{p}_b = [-40 \quad -41 \quad -42 \quad -43 \quad -44 \quad -45]$$

Im Allgemeinen ist es aus oben genannten Gründen sinnvoll, die Beobachter-Pole in Abhängigkeit der Pole des geschlossenen Regelkreises zu definieren. Bei [17] werden sie nach

$$\mathbf{p}_b = \mathbf{p}_r - 25$$

bestimmt, wodurch allerdings komplexe Beobachter-Pole möglich sind. In dieser Arbeit werden sie daher meist folgendermaßen berechnet:

$$p_b = \Re\{p_r \cdot 5\}$$

Neben dem Messvektor y ist auch die Stellgröße u Eingang des Beobachters. Dafür könnte man direkt die geforderte Stellgröße des Zustandsregler u_{reg} verwenden. Allerdings wird letztere bei bestimmten Regelabweichungen häufig deutlich über der maximalen Stellgröße liegen, sodass die Ausgangsstellgröße spätestens beim Motor begrenzt wird. Wenn der Beobachter dies nicht mitbekommt und daher sein System mit der unbegrenzten Regler-Stellgröße simuliert, gibt es zwangsläufig Abweichungen zwischen Beobachter-System und realem System. Der Zustand wird nicht mehr richtig geschätzt, folglich ergibt sich ein schlechteres Regelverhalten, bis hin zur Instabilität.

Aus diesen Gründen ist es sinnvoll, eine Vorsteuerung zu implementieren (siehe Abschnitt 3.2.4), die die Stellgröße schon vor der Ausgabe an den Motor begrenzt und die tatsächliche Stellgröße zurück an den Beobachter gibt, sodass dieser das reale System besser nachbilden kann. Somit weicht die Stellgröße nur noch bei Stellgrößeneinbrüchen anderer Art (siehe Abschnitt 2.2.2) ab.

Differenzieren

Die übrigen Zustände werden durch Differentiation der Messgrößen und evtl. Tiefpass-Filterung gebildet:

$$\hat{x} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{d}{dt} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{d}{dt} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{d}{dt} \end{bmatrix} y = \begin{bmatrix} x_{0m} \\ \frac{d}{dt} x_{0m} \\ \varphi_{1m} \\ \frac{d}{dt} \varphi_{1m} \\ \varphi_{2m} \\ \frac{d}{dt} \varphi_{2m} \end{bmatrix} \quad (3.4)$$

In der Simulation kam es hierbei zu Problemen, da für eine möglichst glatte und fehlerfreie Differentiation eine kleine Schrittweite notwendig ist, außerdem hat sich die Simulation in manchen Fällen aufgehängt. Daher wird diese Variante in dieser Arbeit nicht weiter verfolgt.

Trotzdem sei angemerkt, dass diese Möglichkeit am realen Versuchsstand in Kombination mit einer sinnvollen Filterung der Messgrößen und den Ableitungen möglicherweise Vorteile gegenüber einem Beobachter darstellt. Es ist kein Einschwingen der Startzustände nötig, es gibt keine Probleme bei Stellgrößenstörungen und die Differentiation muss nicht für jeden Arbeitspunkt einzeln ausgelegt werden.

3.2.3 Vorsteuerung F/a und a/v-Regler

Ist der AP-Regler nach dem Beschleunigungssystem ausgelegt, gibt dieser als Stellgröße eine Sollbeschleunigung a_{soll} für den Schlitten vor. Aus dieser muss die geforderte Motorkraft bestimmt werden. Sie kann zum Großteil mit einer Vorsteuerung bestimmt werden, die Differenz wird mit einem Geschwindigkeitsvergleich ausgeglichen.

a/v-Regler

Die Sollgeschwindigkeit v_{soll} wird durch Integration von a_{soll} berechnet. Dadurch kann ein Soll/Ist-Vergleich mit der realen bzw. geschätzten Schlittengeschwindigkeit \dot{x}_0 durchgeführt werden. Die Regelung wird durch einen P-Regler umgesetzt:

$$F_{av} = K_{pv}(v_{\text{soll}} - \dot{x}_0)$$

Brehl [5] ging für K_{pv} beispielsweise von 500 aus, während im SIMULINK-Modell von [17] 150 verwendet wird. Der Parameter scheint am realen Modell einen starken Einfluss auf die Stabilität und ein mögliches „Ratterverhalten“ zu haben. In dieser Arbeit wird ein Wert von 150 angenommen und der Parameter nicht weiter untersucht, da es in der Simulation meist nur geringe Abweichungen von der Sollgeschwindigkeit gibt.

Als problematischer stellte sich hingegen ein möglicher *Windup-Effekt* heraus: Ist die Sollbeschleunigung größer als wegen der Stellgrößenbegrenzung tatsächlich erreicht werden kann, wird v_{soll} immer weiter aufintegriert und damit auch F_{av} , obwohl die Motorkraft bereits maximal ist. Wenn das System wieder aus der Stellgrößenbegrenzung ist und die Vorsteuerungskraft eigentlich das Vorzeichen wechselt, ist der I-Anteil noch vorhanden und muss erst abgebaut werden. In dieser Zeit übersteigt F_{av} daher F_{vorst} und die Stellgröße geht somit praktisch in die falsche Richtung, was die Regelung sehr schnell instabil werden lassen kann. Aus diesem Grund wird a_{soll} vor der Integration mit einem Sättigungsglied auf

$$a_{\text{soll,max}} = \frac{F_{\text{max}}}{m}$$

begrenzt.

Vorsteuerung „MCD“

Der a/v-Regler alleine könnte die Schlittenkraft zwar regeln, allerdings kann der Hauptteil der Kraft mit einer Vorsteuerung bestimmt werden, wodurch die Dynamik besser ist. Die Vorsteuerung besteht im Wesentlichen aus der Trägheitskraft, da bei reiner Betrachtung des Schlittens $F_{\text{soll}} = m_0 a_{\text{soll}}$ gilt. Wenn davon ausgegangen wird, dass die Reibungsparameter hinreichend genau bestimmt wurden, kann auch die Reibung des Schlittens kompensiert werden. Während [5] nur die Trägheitskraft vorsteuerte, wird in [17] die Vorsteuerung folgendermaßen berechnet:

$$F_{\text{vorst}} = m a_{\text{soll}} + F_{c0} \text{sign}(v_{\text{soll}}) + d_0 \dot{x}_0$$

Vorsteuerung mit Systemgleichung

Auch die obige Gleichung bestimmt die benötigte Kraft nicht exakt, da es je nach Winkellage der Pendel rückwirkende Kräfte auf den Schlitten gibt. Man kann daher einen Schritt weiter gehen und die benötigte Kraft zur Beschleunigung des Schlittens direkt aus der entsprechenden Bewegungsgleichung herleiten. Auch wenn am realen Versuchsstand die Beschleunigung aufgrund von Ungenauigkeiten nicht exakt mit a_{soll} übereinstimmt, wird dem a/v-Regler dadurch weitere „Arbeit abgenommen“.

Die Beziehung zwischen der Schlittenbeschleunigung und der Kraft am Schlitten wird durch die zweite Zustandsgleichung (2.8) beschrieben:

$$a = \ddot{x}_0 = f_2(\mathbf{x}, F) \quad (3.5)$$

Diese Gleichung kann leicht nach der Kraft umgestellt werden:

$$F_{\text{soll}} = \frac{a_{\text{soll}} - a_2(\mathbf{x})}{b_2(\mathbf{x})} \quad (3.6)$$

Damit diese Berechnung für beliebige Systemparameter und somit unterschiedliche Funktionen a_2 und b_2 erfolgen kann, wird die Bestimmung der Vorsteuerkraft in dieser Arbeit symbolisch in MATLAB automatisiert.

Ermittlung reale Beschleunigung

In Abschnitt 3.2.2 wurde erläutert, dass es sinnvoll ist, die reale Stellgröße an den Beobachter zu geben. Im folgenden Vorsteuerungsblock (Abschnitt 3.2.4) wird die vor dem Motor begrenzte Kraft F_{real} zurückgegeben, da die Stellgrößenbegrenzung F_{max} bekannt ist. Ist der Beobachter allerdings am Beschleunigungssystem ausgelegt, geht er bei der Stellgröße von der Beschleunigung aus. Somit muss aus F_{real} zunächst a_{real} bestimmt werden.

Dies kann ähnlich wie bei der Kraftvorsteuerung durch die Trägheit geschehen, was allerdings ungenau ist und der Beobachter demzufolge von einer falschen Eingangsgröße ausgeht. Daher wird die Ermittlung der tatsächlichen Beschleunigung wie im vorigen Abschnitt über die Systemgleichung vorgenommen. Es gilt (3.5).

3.2.4 Motor-Vorsteuerung

Bisher wurde für die Regelung die Ermittlung der Kraft, die am Schlitten wirken soll, betrachtet. Die Schnittstelle zum Motor besteht durch Vorgabe der Steuerspannung. Die ausgegebene Spannung stellt daher letztendlich die eigentliche Stellgröße dar und muss in Abhängigkeit der Sollkraft bestimmt werden.

Auch wenn das Motorsystem für die Simulation genauer modelliert wird (siehe Abschnitt 2.2), handelt es sich (abgesehen von einer sehr schnellen Tiefpass-Dynamik) im Wesentlichen um ein reines Verstärkungsglied, das mit der Konstante $K_{\text{MotorGain}}$ (2.20) beschrieben wird. Die Vorsteuerung rechnet daher mit der Konstante die Sollkraft in die Sollspannung um.

$$U_{\text{soll}} = \frac{F_{\text{soll}}}{K_{\text{MotorGain}}}$$

In der Motor Vorsteuerung wird auch die Stellgrößenbegrenzung berücksichtigt. Wie schon in Abschnitt 3.2.2 erläutert wurde, wird die Motorspannung vor der Ausgabe begrenzt. Die tatsächliche Stellgröße kann dadurch an den Beobachter gegeben werden.

3.3 Simulationsmodell in SIMULINK

Die SIMULINK-Modelle und MATLAB-Funktionen zur Arbeitspunkt-Regelung befinden sich im Ordner Regelung.

3.3.1 SIMULINK Module und Modelle

Die Arbeitspunkt-Regelung wird wie in Abschnitt 3.2 beschrieben in `AP_Regelung_Test.slx` als SIMULINK-Modell implementiert (siehe Abbildung 3.2). Das Gesamtmodell-Modul (siehe Abschnitt 2.4.2) wird hier eingebunden und stellt die Regelstrecke dar. Davor befinden sich drei Blöcke der Regelung/Vorsteuerung. Alternativ kann auch zum Arbeitspunkt-Regler am Kraftsystem umgeschaltet werden.

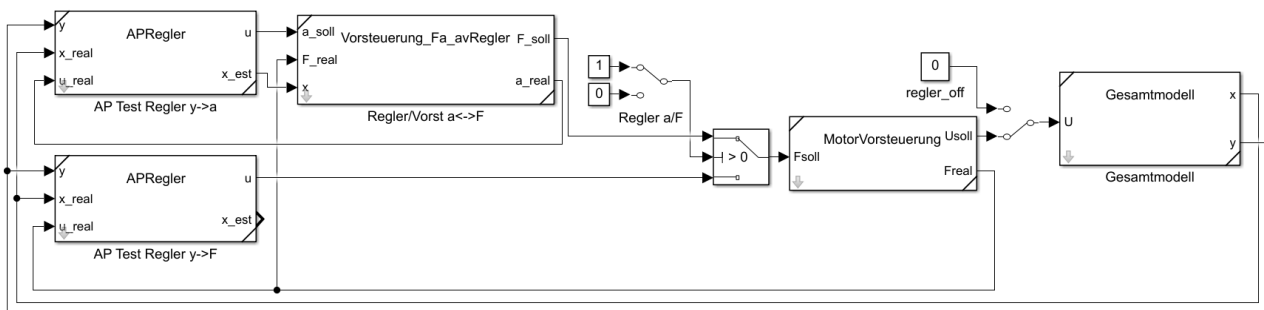


Abbildung 3.2: Aufbau der Arbeitspunkt-Regelung

APRegler.slx

Dieses Modul ist eine Art „Wrapper“ für den eigentlich Zustandsregler, um die Arbeitspunkt-Transformation durchzuführen. Erwartet als Parameter die Daten zum Arbeitspunkt.

Zustandsregler.slx

Implementiert einen gewöhnlichen Zustandsregler, bei dem alle Arten der Zustandsermittlung (siehe Abschnitt 3.2.2) möglich sind (siehe Abbildung 3.3). Die Auswahl wird durch ein *Variant Subsystem* realisiert, das durch die Variable `simparams.Zustandsermittlung` angesteuert wird. Die Stellgröße für den Beobachter kann zwischen intern und extern umgeschaltet werden. Benötigt als Parameter die Verstärkung K sowie eventuelle Daten für den Beobachter.

Beobachter.slx

Implementiert einen Standard-Beobachter. Eingänge sind u und y , Ausgang sind die geschätzten Zustände x_{est} . Die Matrizen des linearen Systems, sowie L und die Beobachterstartwerte werden über die Maske übergeben.

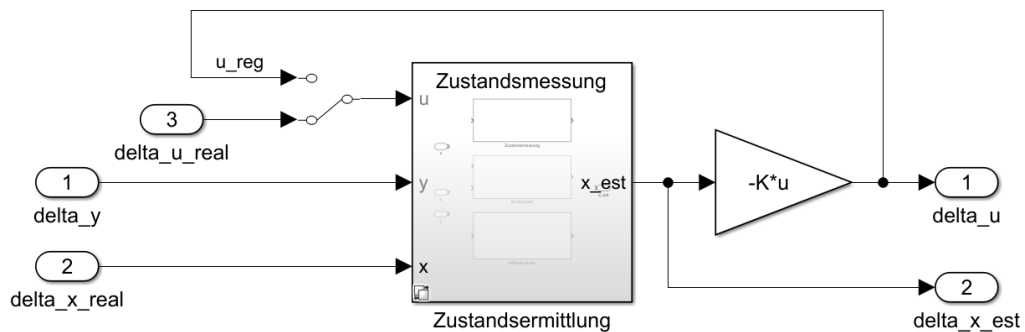


Abbildung 3.3: Zustandsregler in SIMULINK

Vorsteuerung_Fa_avRegler.slx

Abbildung 3.4 zeigt den vereinfachten Aufbau des Blocks, der die Verbindung zwischen Beschleunigung und Kraft herstellt (siehe Abschnitt 3.2.3). Aus der Sollbeschleunigung wird mithilfe von `SchlittenGleichungKraft` die Vorsteuerkraft bestimmt. Im Modell kann zudem zwischen dieser und der Vorsteuerungsvariante „MCD“ (nicht dargestellt) umgeschaltet werden. Der a/v-Regler wird entsprechend Abschnitt 3.2.3 implementiert und dessen Ausgangsgröße zur Vorsteuerkraft addiert. Oben wird mithilfe von `SchlittenGleichungBeschleunigung` aus der nachträglich bestimmten realen Stellkraft die reale Beschleunigung berechnet.

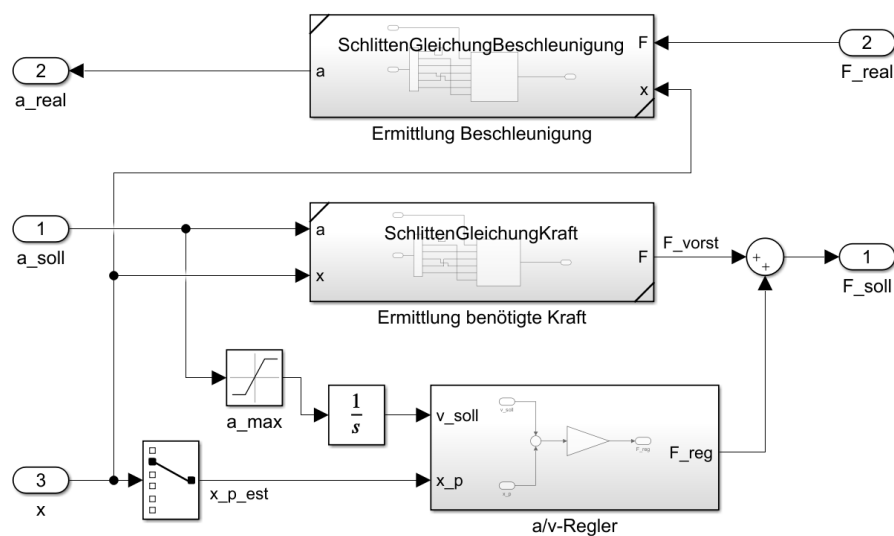


Abbildung 3.4: Aufbau der Vorsteuerung

SchlittenGleichungKraft.slx / SchlittenGleichungBeschleunigung.slx

Setzt die Variante der Vorsteuerung mit den Systemgleichungen (siehe Abschnitt 3.2.3) bzw. die inverse Gleichung zur Ermittlung der Beschleunigung um. Da diese Gleichungen symbolisch vorliegen, müssen sie mithilfe von `matlabFunctionBlock` bei der Initialisierung in den Block geschrieben werden. Dadurch werden diese Dateien bei jeder Initialisierung oder Änderung der Systemparameter überschrieben.

SchlittenVorsteuerung.slx

Enthält die „alte“ Variante der Vorsteuerung (siehe Abschnitt 3.2.3). Dabei werden die Schlittenparameter (m_0 , F_{c0} und d_0) verwendet.

MotorVorsteuerung.slx

Enthält die in Abschnitt 3.2.4 beschriebene Vorsteuerung zum Motor. Berechnet aus der Sollkraft die Sollspannung für den Motor und gibt die möglicherweise begrenzte Kraft zurück an den vorigen Block.

3.3.2 Initialisierung

Zur Initialisierung der Arbeitspunkt-Regelung wird (nach der Initialisierung des Modells) das Skript `InitReg.m` ausgeführt. Dabei werden folgende 4 Schritte durchgeführt:

1. **InitSystemReg()**: Erstellt das Zustandsraummodell, an dem der Regler ausgelegt wird. Dabei wird standardmäßig die beim realen System vorhandene Coulomb-Reibung in den Pendelgelenken für das System des Reglers zu 0 gesetzt, um deren unverhältnismäßigen Einfluss bei der Linearisierung zu vermeiden (siehe Abschnitt 3.1.2). Prinzipiell wäre es hier auch möglich, die Parameter des Reglers zu verändern, um Parameterunsicherheiten zu modellieren.
2. **InitVorstBeob_Fa(sysF)**: Berechnet die Vorsteuerkraft über die Systemgleichung wie in Abschnitt 3.2.3 beschrieben. Diese symbolischen Gleichungen wird mittels `matlabFunctionBlock` in die beiden entsprechenden Module geschrieben und diese werden gespeichert.
3. **InitAPRegData(riccdata)**: Diese Funktion berechnet die Daten für den AP-Regler anhand der gegebenen Güteparameter. Dabei werden alle Arbeitspunkte linearisiert, die Reglerverstärkung mit der MATLAB-Funktion `lqr` und die Beobachtermatrix berechnet.
4. **InitSimReg(Zustandsermittlung, kpv)**: Initialisiert das Simulationsmodell für die AP-Regelung. Werte für K_{pv} und die Art der Zustandsermittlung (Standard: Messung) kann angegeben werden. Die Parameter für die Module zur Regelung werden in die *struct*-Variable `simparams.regler` geschrieben. Die Systemparameter von Motor und Schlitten-Pendel (von denen der Regler ausgeht) werden hier übergeben, da manche von ihnen für die Regelung/Vorsteuerung benötigt werden. Durch die Trennung zu den „realen“ Werten `simparams.gesamtmodell` können die Parameter des Reglers grundsätzlich auch anders sein als die, mit denen das Modell simuliert wird.

Anschließend können mit `SimAP` Arbeitspunkttests durchgeführt werden.

3.3.3 Güteparameter (Q,R)

Ähnlich wie bei den Parametern des Schlitten-Pendels werden auch die Parametersätze für die *Riccati*-Güteparameter ausgelagert. Diese sind für jeden Arbeitspunkt separat definiert, da jeweils andere Werte optimal sein können. So können verschiedene Parameter einfach verglichen werden. Vollständige Güteparameterwerte für alle Arbeitspunkte fanden sich bei den Arbeiten, die sich mit der Arbeitspunkt-Regelung beschäftigt haben, nur in den letzten zwei. Im Ordner `Regelung/Reglerparameter` befinden sich folgenden Datensätze:

-
- AP_QR_Chang19: Werte der Ausarbeitung von [17]
 - AP_QR_Ribeiro20: Werte der Ausarbeitung von [16]
 - AP_QR_20_neu: Neu bestimmte Werte, die in dieser Arbeit optimiert wurden
 - AP_QR_20_neu_Beob: Neu bestimmte Werte, die in dieser Arbeit für den Einsatz mit Beobachter optimiert wurden

3.3.4 Weitere MATLAB-Funktionen

SimAP

Mit dieser Funktion wird eine Simulation mit gegebenem Arbeitspunkt und Startwerten (Arbeitspunkt-abweichung) durchgeführt. Die Simulationszeit T_{sim} ist optional (Standard: 10 s).

```
SimAP(testAP, delta_x0, Tsim)
```

Dabei werden die Initialisierungen des Arbeitspunktes, der Startwerte und der Beobachterstartwerte vorgenommen. Außerdem werden Auswertungen der Simulation vorgenommen (siehe Abschnitt 3.4.1) und die Ergebnisse zurückgegeben. Die Funktion kann auch mit `SimAP_run.m` aufgerufen werden. Dabei werden weitere Auswertungen angezeigt, z. B. die Maximalabweichungen vom Arbeitspunkt, ob die Stellgrößenbegrenzung aktiv war oder ob der Schlitten außerhalb der Begrenzung war. Zusätzlich wird direkt ein Plot und eine Animation (siehe Abschnitt 2.4.5) angezeigt.

SchlittenPendelAnalyse

Mit der Funktion `SchlittenPendelAnalyse(sys)` werden für ein gegebenes Zustandsraummodell Eigenwerte für alle Ruhelagen berechnet und dargestellt, sowie Steuer- und Beobachtbarkeit geprüft.

x0_Tests

Die Funktionen zu den Anfangswerttests (siehe folgender Abschnitt) und den darauf aufbauenden Tests befinden sich im Ordner `Regelung/x0_Tests`.

3.4 Anfangswert-Tests

Ziel dieser Arbeit ist die „Regelbarkeit“ des Systems zu untersuchen. Dafür müssen zunächst Kriterien definiert werden, inwiefern das System unter festgelegten System- und Entwurfsparametern regelbar bzw. stabilisierbar ist.

Dazu wird wieder Abbildung 3.1 herangezogen. Ist ein System fest ausgelegt, können die verschiedenen Arbeitspunkte mit bestimmten Startwerten getestet werden. Dabei gibt es jedoch unzählige Kombinationsmöglichkeiten, die nicht alle getestet werden können. Es muss daher eine Auswahl an Startwerten getroffen werden. Oft wird immer nur für einen Zustand eine Abweichung gesetzt, alle anderen sind am

Arbeitspunkt. Selbst dann ist es mit manuellem Testen und Ausprobieren aufwändig, ein umfassendes Bild vom Regelverhalten zu erlangen. Die Optimierung der „äußeren Schleifen“ wird somit langwierig und ist auf diese Weise nicht zielführend.

Daher wird die „untere Schleife“ automatisiert. Wird ein Arbeitspunkt-Test durchgeführt, können die Simulationsdaten ausgewertet werden und gewisse Gütemaße und Performance-Indikatoren bestimmt werden, um die Regelgüte zu beschreiben. Im Vordergrund steht aber meist die Stabilität und damit die Frage, bei welchen Werten diese nicht mehr vorhanden ist.

3.4.1 Auswertung eines Arbeitspunkt-Tests

Wenn SimAP aufgerufen wird, werden neben den Simulationsdaten auch die Ergebnisse von *Auswertung/AP/APAuswertung* zurückgegeben. Dort werden u. a. folgende Gütemaße, die an das übliche quadratische *Riccati*-Gütemaß angelehnt sind, berechnet:

$$J_x = \int_0^{t_{\text{end}}} \Delta \mathbf{x}^T(t) \mathbf{Q} \Delta \mathbf{x}(t) dt = \int_0^{t_{\text{end}}} \Delta x_{\text{norm}}(t) dt \quad (3.7a)$$

$$J_{x,\text{est}} = \int_0^{t_{\text{end}}} (\Delta \mathbf{x}(t) - \Delta \mathbf{x}_{\text{est}}(t))^T \mathbf{Q} (\Delta \mathbf{x}(t) - \Delta \mathbf{x}_{\text{est}}(t)) dt \quad (3.7b)$$

$$J_u = \int_0^{t_{\text{end}}} R \cdot \Delta u(t)^2 dt \quad (3.7c)$$

$$J_F = \int_0^{t_{\text{end}}} R \cdot F(t)^2 dt \quad (3.7d)$$

mit

$$\mathbf{Q} = \text{diag}(25, 1, 50, 1, 50, 1)$$

$$R = 1$$

Dabei beschreibt J_x den kumulierten Verlauf aller Zustände als skalare Größe. $J_{x,\text{est}}$ berechnet ein Maß zur Bewertung des Beobachterfehlers. Die anderen beiden Maße beschreiben die kumulierte Stellgröße/-energie, hier ist J_F aussagekräftiger, da es die reale Motorkraft verwendet. Für die Berechnung werden die transformierten Arbeitspunkt-Zustände $\Delta \mathbf{x}(t)$ (3.3) verwendet, da diese die Abweichung von der Ruhelage darstellen. In MATLAB werden die Maße als Summe berechnet (der Einfachheit halber konstante Schrittweite vorausgesetzt).

Für die Beurteilung der Stabilität, also ob der Zustand am Ende der Simulationszeit noch in der Nähe des Arbeitspunktes ist, wird $\Delta x_{\text{norm}}(t_{\text{end}})$ verwendet. Ist dieser skalare Wert unter einer festgelegten Schranke, wird von einer Stabilisierung des Arbeitspunktes ausgegangen.

Außerdem wird für jeden Zustand die Maximalabweichung vom Arbeitspunkt bestimmt, da diese ein wichtiges Gütekriterium ist, insbesondere die Schlittenposition, da diese Beschränkungen unterliegt (siehe Abschnitt 2.1.6).

Des Weiteren kann anhand von $\Delta x_{\text{norm}}(t)$ eine Einschwingzeit bestimmt werden. Diese ist dort, wo der Norm-Verlauf eine gewisse Schranke nicht mehr überschreitet.

3.4.2 Anfangswertest

Die Funktion `x0_Test.m` stellt die Basis für einen Anfangswertest dar. Für den angegebenen Arbeitspunkt ermittelt sie mit einzelnen Simulationen und steigenden Anfangsauslenkungen vom Arbeitspunkt schrittweise die maximalen Startwerte. Dabei wird mit obigen Methoden ermittelt, ob die Stabilisierung erfolgreich war. Falls nicht, wird abgebrochen und der Maximalwert sowie einige der Gütemaße für allen Einzelergebnisse zurückgegeben.

Die Schrittweite $y_{st} = [x_{st} \phi_{1st} \phi_{2st}]$ gibt die Schrittweite an, mit der die Tests ausgeführt werden. Eine kleinere Schrittweite löst zwar die Ergebnisse besser auf, führt aber zu einer höheren Rechenzeit.

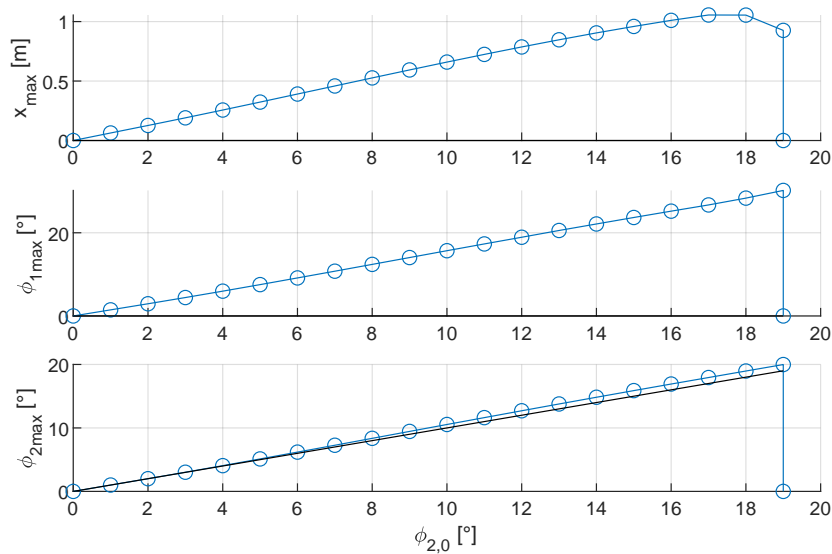


Abbildung 3.5: Anfangswertest AP4, Auslenkung φ_2

Die Ergebnisse eines Anfangswerttests an AP4 sind beispielhaft in Abbildung 3.5 dargestellt. Auf der Abszisse wird der Startwert der Auslenkung (hier φ_2) aufgetragen. In Abhängigkeit von dieser werden die sich in der Folge ergebenden maximalen Abweichungen der drei Ausgänge vom Arbeitspunkt dargestellt. In diesem Fall kann das System bis zu einem Startwert von $\varphi_{2,0} = 19^\circ$ stabilisiert werden. Die Startauslenkung wird auch als schwarze Linie eingezeichnet. Die Maximalabweichung ist stets oberhalb oder auf dieser.

3.4.3 Kritische Anfangswerttests

Wenn immer nur ein Zustand ausgelenkt wird, ergeben sich pro Arbeitspunkt 3 Tests, also insgesamt 9 Tests (unter Vernachlässigung des stabilen AP1). Es hat sich dabei gezeigt, dass Abweichungen von x_0 kein Problem bezüglich der Stabilität darstellen. Außerdem sind bei Arbeitspunkt 2 und 3 Auslenkungen des jeweils „stabilen“ Pendels unkritisch.

Aus diesem Grund sind für die Untersuchung der Stabilität nur die folgenden vier (kritischen) Anfangswert-Tests relevant:

- AP2: Arbeitspunkt 2, Auslenkung φ_2
- AP3: Arbeitspunkt 3, Auslenkung φ_1
- AP41: Arbeitspunkt 4, Auslenkung φ_1
- AP42: Arbeitspunkt 4, Auslenkung φ_2

In `x0_Test_APs.m` werden diese vier Tests durchgeführt und die Ergebnisse zur weiteren Darstellung gespeichert. Maßgeblicher Indikator für die Stabilität ist die jeweilige maximale Auslenkung φ_1 oder φ_2 , bis zur welcher das System noch stabilisiert werden kann.

Diese Tests können nun mit verschiedenen Systemen durchgeführt werden und miteinander verglichen werden. Dabei können Regelparameter oder auch Systemparameter variiert werden. Der Vorteil dieser Darstellung ist, dass die Gütekriterien wie Maximalabweichungen, die obigen Gütemaße und die Einschwingzeit direkt im selben Diagramm für alle Startwerte verglichen werden können. Dadurch lässt sich die Güte der Regelung bzw. die Regelbarkeit einer Konfiguration auf einen Blick beurteilen.

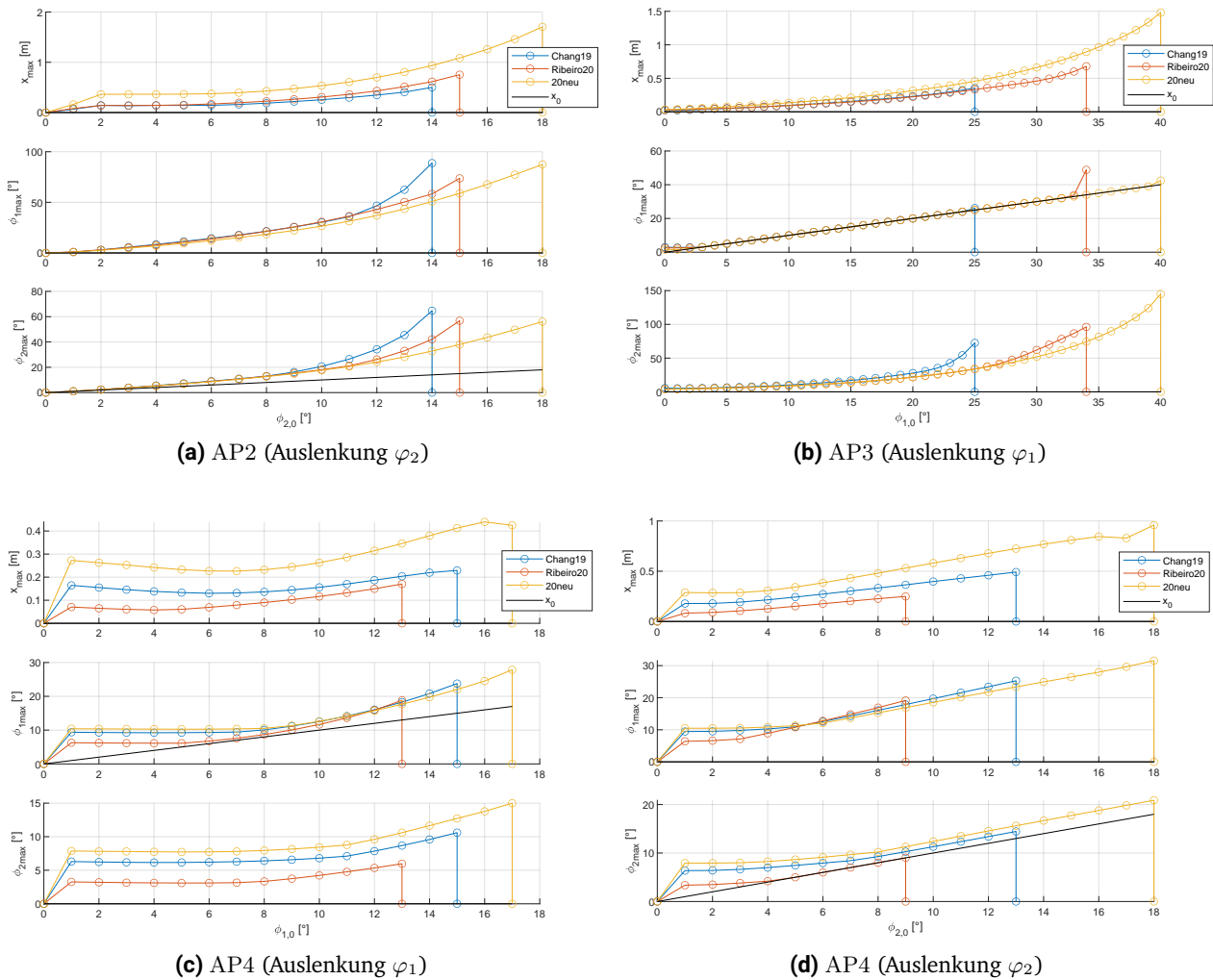


Abbildung 3.6: Maximalabweichungen – Vergleich QR-Parameter (System Ribeiro)

3.4.4 Vergleiche von Anfangswerttests

Beispielsweise kann man die Ergebnisse mit verschiedenen *Riccati*-Parametern Q und R vergleichen. In Abbildung 3.6 sind die Maximalabweichungen mit den verschiedenen QR-Parametern (siehe Abschnitt 3.3.3) am neuen System dargestellt. Die Wahl der QR-Parameter hat offenbar einen wichtigen Einfluss auf Stabilisierbarkeit und Regelgüte. Insgesamt lässt sich feststellen, dass AP3 die größten Startabweichungen ausregeln kann. Die in dieser Arbeit optimierten Parameter können das Doppelpendel bei allen Arbeitspunkten in einem etwas größeren Bereich stabilisieren. Allerdings sieht man auch z. B. an AP41, dass das meist auf Kosten höherer Abweichungen bei den niedrigeren Startwerten geschieht. Bei den Reglerparametern ist somit immer ein gewisser Kompromiss notwendig.

Bei der Regelung mit den neuen Pendelparametern Ribeiro20 ist im Vergleich zu den alten Parametern Apprigh09 aufgefallen, dass der Endwert nicht mehr asymptotisch erreicht wird, sondern sich ein Grenzzyklus einstellt. Die Ursache dafür liegt sehr wahrscheinlich in der Coulomb-Reibung der Pendelgelenke (siehe Tabelle 2.4), die beim alten System nicht vorhanden ist.

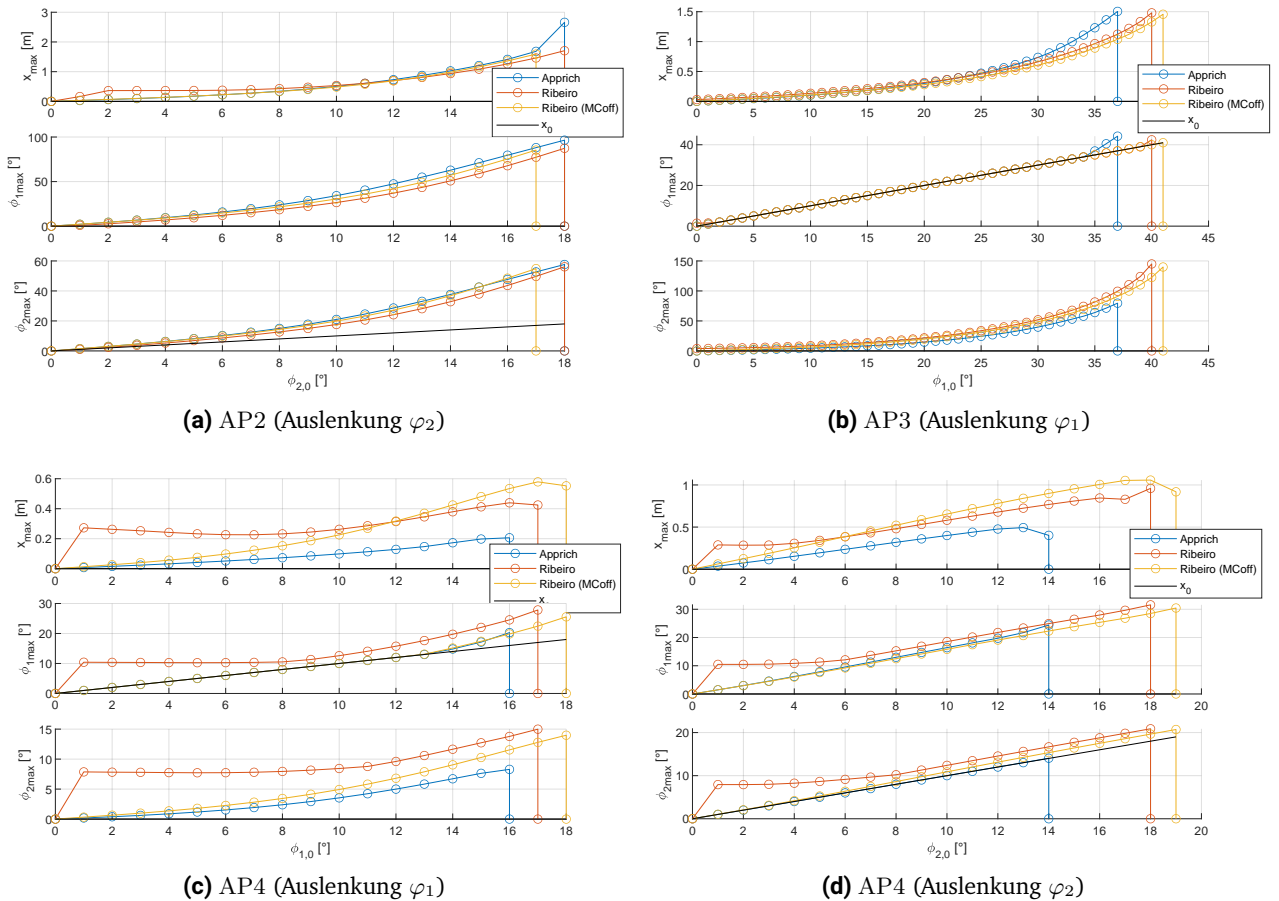


Abbildung 3.7: Maximalabweichungen – Vergleich alter und neuer Pendelparameter

Abbildung 3.7 vergleicht die beiden Systeme, sowie eine Variante, bei der die Coulomb-Reibungsmomente zu 0 gesetzt werden (MCoff). Bei allen Arbeitspunkten fällt auf, dass das System mit den Coulomb-Reibungsmomenten bei sehr niedrigen Startwerten höhere Maximalabweichungen zeigt, die durch

die Dauerschwingung verursacht werden. Zum Beispiel ergibt sich bei AP2 ab einem Startwert von $\varphi_{1,0} = 2^\circ$ ein Positionsausschlag von 0,3 m. Bei AP3 und AP4 ergibt sich sogar bei einem Startwert von 0 eine Dauerschwingung (was bei AP3 auf die numerische Ungenauigkeit von π zurückzuführen ist, bei AP4 erst mit einer minimalen Anregung). Da die Maximalabweichungen insbesondere bei AP4 über einen weiten Bereich konstant sind ($\varphi_{1m} = 10^\circ$, $\varphi_{2m} = 8^\circ$), besteht die Vermutung, dass der Grenzzyklus fast unabhängig von den Startwerten ist. Die alten Parameter sowie die neuen Parameter mit ausgeschalteter Coulomb-Reibung in den Gelenken zeigen dieses Verhalten nicht und beginnen in 0 mit einem etwa linearen Verlauf. Aus diesem Grund wird die nichtlineare Coulomb-Reibung als Ursache dafür gesehen.

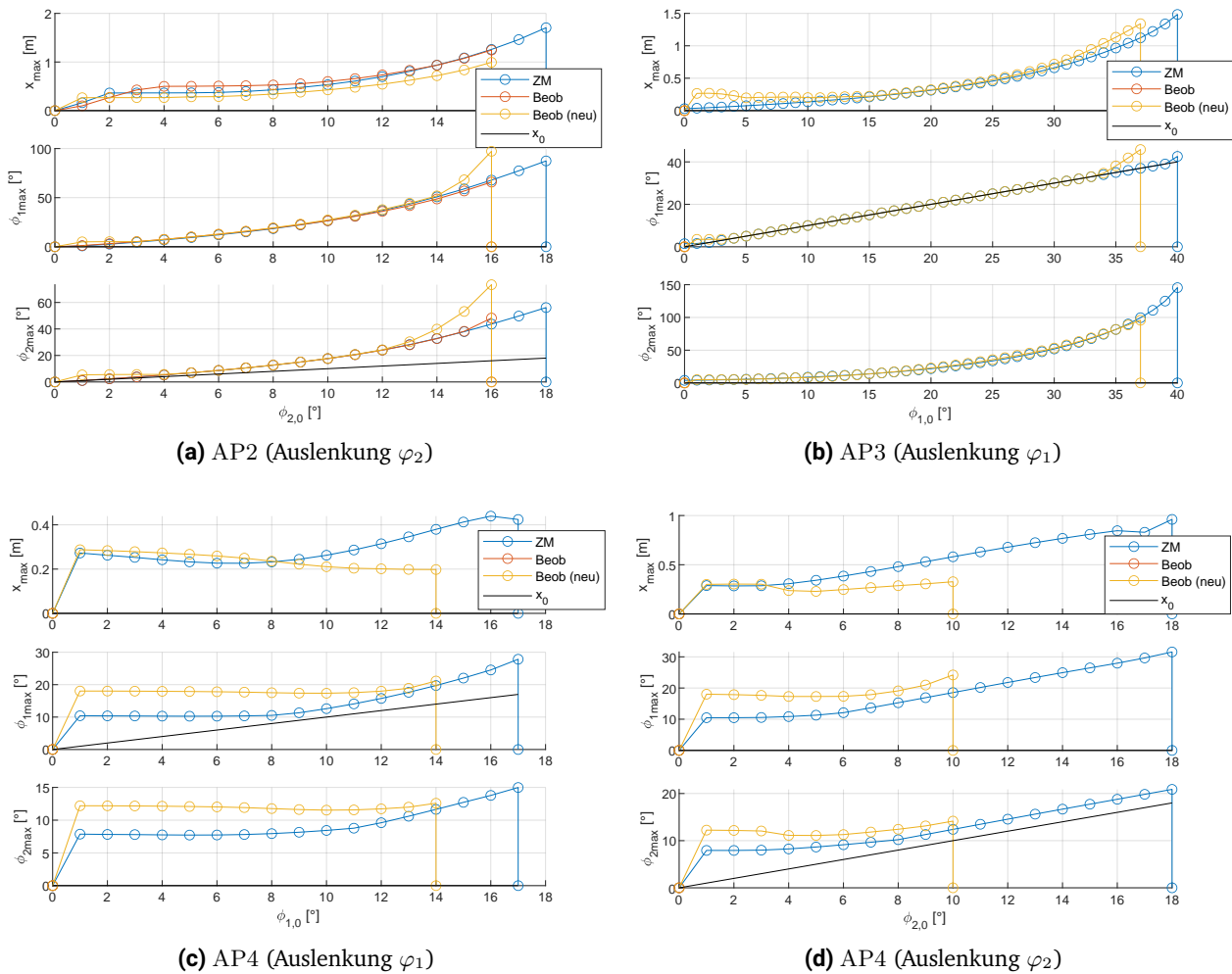


Abbildung 3.8: Maximalabweichungen – Vergleich Zustandsmessung/Beobachter (System Ribeiro)

Auch der Einfluss des Beobachter lässt sich mit den kritischen Anfangswerttests gut darstellen. Beim Test des Beobachters an der Neukonstruktion ist zunächst aufgefallen, dass die QR-Parameter, die vorher für die Zustandsmessung optimiert wurden, nur bei AP2 brauchbare Ergebnisse zeigten. Bei AP4 versagten zudem die beiden Vergleichsparameter der letzten Arbeiten. Aus diesem Grund wurde ein neuer QR-Datensatz erstellt (siehe Abschnitt 3.3.3), der speziell für die Regelung mit Beobachter optimiert ist.

Abbildung 3.8 vergleicht daher die Arbeitspunkt-Regelung mit Zustandsmessung (ZM) und Beobachter (Beob), einerseits mit den QR-Parametern AP_QR_20_neu, sowie mit den angepassten QR-Parametern AP_QR_20_neu_Beob (Beob (neu)). Während bei AP2 und AP3 (nur die Beobachter-QR-Werte) die Regler-Performance vergleichbar ist, zeigt sich bei AP4, dass der Beobachter das System nicht so weit stabilisieren kann wie mit der Zustandsmessung. Die maximalen Startwerte sind mit $\varphi_{1,0\max} = 14^\circ$ und $\varphi_{2,0\max} = 10^\circ$ niedriger, außerdem sind die Winkelausschläge höher.

3.5 QR Parameter Tests

Wie schon festgestellt wurde, haben Reglerparameter nicht nur einen wesentlichen Einfluss auf die Dynamik, sondern auch auf die Stabilität und damit Regelbarkeit des Systems. Daher ist es sinnvoll, deren Einfluss in einem nächsten Schritt zu untersuchen und diese zu optimieren (siehe Abbildung 3.1). Dabei werden in dieser Arbeit hauptsächlich die *Riccati*-Güteparameter $\mathbf{Q} = \text{diag} (Q_1 \ Q_2 \ Q_3 \ Q_4 \ Q_5 \ Q_6)$ und R betrachtet, da diese einen deutlichen Einfluss auf die Arbeitspunkt-Regelung haben. Andere Reglerparameter wie K_{pv} oder die Beobachterpole könnten ebenfalls optimiert werden, jedoch haben diese auf den ersten Blick keinen entscheidenden Einfluss.

3.5.1 Vorgehen bei der Optimierung

Die Gütematrizen wurden in den meisten vergangenen Arbeiten und anfangs auch in dieser heuristisch bestimmt und iterativ manuell angepasst. Dieses Vorgehen ist auf Dauer sehr aufwändig und garantiert aufgrund der vielen Variationsmöglichkeiten keinesfalls optimale Werte. Auch muss beachtet werden, dass jede Änderung an Systemparametern oder das Umschalten zum Beobachter sehr wahrscheinlich zu anderen optimalen Werten führt. Bei [17] wurde eine Zufallssuche zum Finden von geeigneten Reglerparametern durchgeführt. Dort wurde aber lediglich eine feste Anfangsauslenkung von φ_2 getestet, bei der Auswertung wurde eine feste obere Schranke für die maximale Schlittenposition und Endwert der Winkel verwendet.

Durch die Anfangswerttests (siehe Abschnitt 3.4.2) kann schnell ein Bild des Regelverhaltens erlangt werden. Bei den Vergleichen in Abschnitt 3.4.4 wurden lediglich feste QR-Parametersätze verglichen. Zur gezielten Optimierung liegt es nahe, einzelne Parameter zu variieren und deren Auswirkungen in den Diagrammen der Anfangswerttest zu untersuchen. So können die einzelnen Parameter schrittweise optimiert werden, wobei selbstverständlich zu beachten ist, dass es sich eigentlich um eine multi-variable Optimierung handelt (mit 6 Freiheitsgraden) und sich daher die Ergebnisse eines Parameters nach Änderung der anderen Parameter wieder ändern können. Durch iteratives Vorgehen können die Werte trotzdem sehr gut optimiert und das Regelverhalten deutlich verbessert werden.

3.5.2 Anwendung in MATLAB

Da die Güteparameter für jeden Arbeitspunkt festgelegt werden, findet die Variation der Parameter für einen festen Arbeitspunkt statt. Des Weiteren wird der Funktion `QR_Variation` ein QR-Datensatz und der Parameter gegeben, der zu variieren ist (R oder Q_i). Außerdem müssen die Werte angegeben werden, welche der Parameter annehmen soll. Um die Parameter gleichmäßig über einen großen Bereich

zu testen, eignet sich ein logarithmischer Bereich, wofür die MATLAB-Funktion `logspace` genutzt wird. Ein beispielhafter Aufruf lautet:

```
QR_Variation(2, AP_QR_20_neu(), 'Q1', logspace(-2,2,7), 2:3 )
```

In der Funktion sind mehr Informationen angegeben und in `QR_Variation_run.m` weitere Anwendungsbeispiele zum Verwenden der Parametervariation vorhanden.

Welche der Zustände für den Test mithilfe der Anfangswerttests ausgelenkt werden sollen, kann angegeben werden. So kann man nur die jeweils „kritischen“ Tests ausführen (siehe Abschnitt 3.4.3), aber auch die Auswirkungen bei Startwerten der anderen Zustände untersuchen und optimieren. Auch hier ist meistens ein Kompromiss notwendig, der für einen Arbeitspunkt verschiedene Anfangswertkombinationen gut ausregelt.

3.5.3 Diagramme

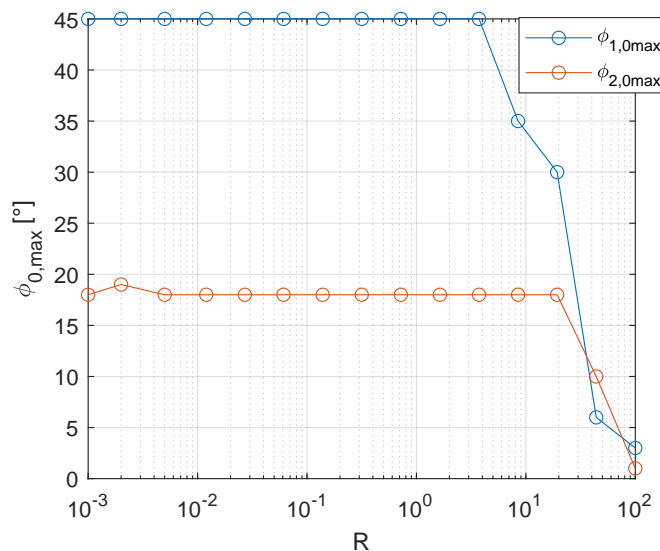


Abbildung 3.9: Maximale Startwerte – Variation von R bei AP2

In Abbildung 3.9 werden die maximalen Startwerte von φ_1 und φ_2 bei AP2 in Abhängigkeit von R dargestellt. Beim ersten Pendel, welches bei AP2 stabil ist, gelingt die Stabilisierung über dem größten Bereich (ab 45° wird der Test abgebrochen). Die maximale Startauslenkung von φ_2 beträgt meistens 18° und bricht ab $R = 20$ ebenfalls ein. Als Schlussfolgerung für den Parameter R gilt hier daher: je kleiner desto besser.

Allerdings muss auch die Dynamik, insbesondere die resultierenden Maximalausschläge der Zustände, beachtet werden. Abbildung 3.10 stellt daher für dieselbe Konfiguration die Maximalabweichungen dar, bei Auslenkung von φ_1 . Hier zeigt sich ebenfalls ein besseres Verhalten bei kleineren Werten von R .

Bei einer Variation von Q_1 an AP4 ergibt sich für die maximalen Startwinkel der in Abbildung 3.11 dargestellte Verlauf. Kleinere Werte von Q_1 führen tendenziell zu höheren möglichen Startwerten. Allerdings sieht man an Abbildung 3.12, dass niedrigere Werte auch zu höheren Maximalabweichungen führen. Daher wurde hier beispielsweise ein Kompromiss von $Q_1 = 1$ getroffen.

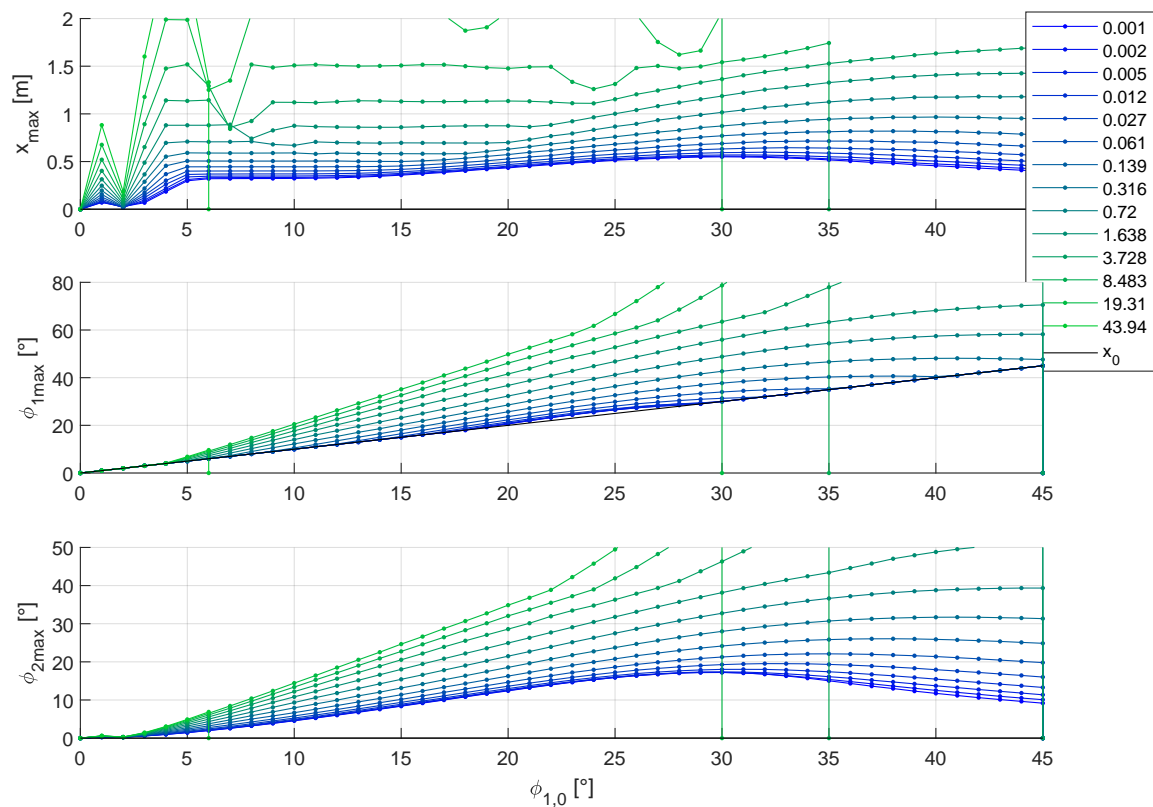


Abbildung 3.10: Maximalabweichungen – Variation von R bei AP2

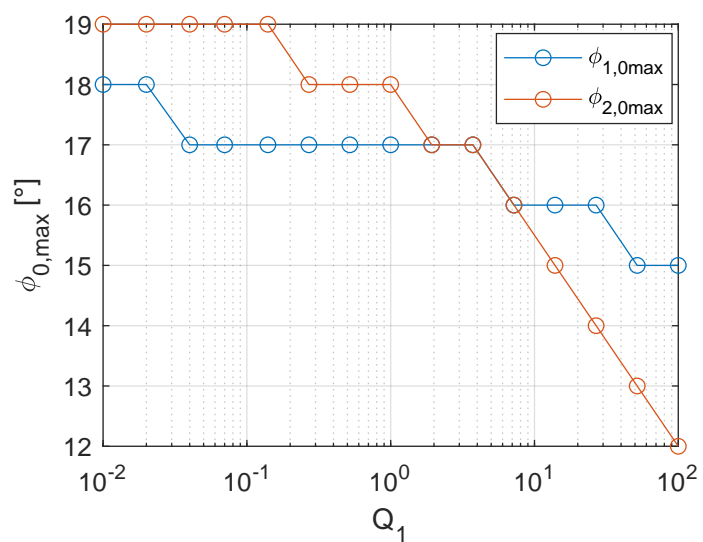


Abbildung 3.11: Maximale Startwerte – Variation von Q_1 bei AP4

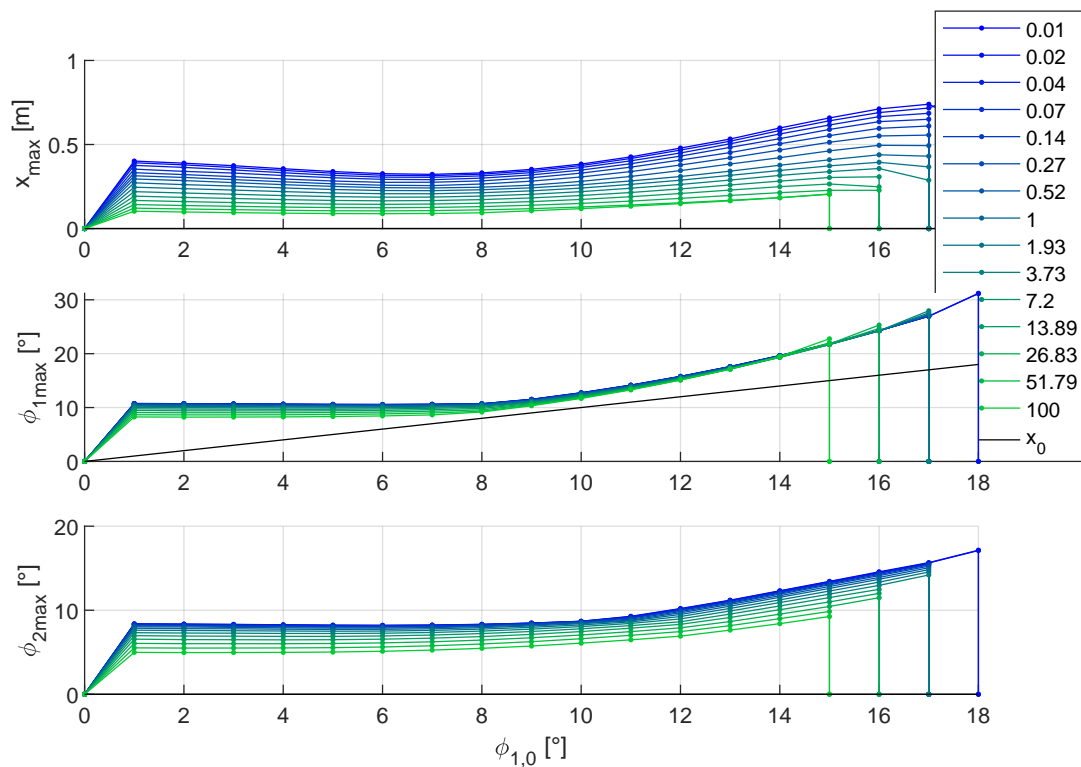


Abbildung 3.12: Maximalabweichungen – Variation von Q_1 bei AP4

3.5.4 Weitere Ergebnisse

Insgesamt zeigte sich für R eine bessere Dynamik bei kleinen Werten, da dadurch die Stellgröße weniger stark gewichtet wird und somit höher sein darf. Da geeignete Begrenzungs- und Anti-Windup-Maßnahmen vorhanden sind, stellt eine höhere Stellgröße grundsätzlich kein Problem dar. R wurde für alle Arbeitspunkte auf 0,01 gesetzt.

Die Werte Q_1 bis Q_6 wurden durch Auswerten aller Tests für alle Arbeitspunkte in AP_QR_20_neu angepasst und für das System Ribeiro20 optimiert (siehe Abbildung 3.6). Die größten Auswirkungen zeigten sich bei Q_1 . Meistens war der Einfluss der anderen Werte gering und brachte bei starker Erhöhung eine schlechtere Dynamik mit sich. Bei AP4 zeigte Q_2 und Q_4 einen deutlichen Einfluss.

Die Werte AP_QR_20_neu_Beob wurden für den Einsatz mit Beobachter optimiert. Hierbei stellte sich heraus, dass für Q_1 generell größere Werte besser waren. Besonders bei AP4 mussten die Werte angepasst werden, damit das System einigermaßen stabilisiert werden kann.

3.6 System Parameter Tests

Durch Verbesserung der Regelung und die systematische Optimierung der Reglerparameter konnte die Regelbarkeit des Doppelpendels deutlich verbessert werden und über einen größeren Bereich stabilisiert werden. Trotzdem gibt es hierbei gewisse Grenzen, da die Eigenschaft der „Regelbarkeit“ nicht nur vom Reglerentwurf, sondern auch vom System selbst abhängt. Auch ist unklar, ob die durch

die Simulation errungenen Ergebnisse direkt auf den realen Versuchsstand übertragbar sind. Versuche mit dem Beobachter haben gezeigt, dass die Zustandsermittlung einen erheblichen Einfluss auf das Regelverhalten und die Stabilisierung hat. Da am realen System weitere, nicht modellierte Effekte auftreten können, ist es von Vorteil, eine gewisse „Reserve“ bezüglich der Regelbarkeit zu haben.

3.6.1 Vorgehen

Aus diesen Gründen wird in diesem Abschnitt die Regelbarkeit des Systems in Abhängigkeit der Systemparameter untersucht. Somit wird die äußerste Schleife aus Abbildung 3.1 betrachtet. Dabei ist zu beachten, dass jede Änderung am Streckenmodell bzw. deren Parametern Auswirkungen auf alle darunterliegenden Schleifen haben. Der Regler muss am geänderten System erneut ausgelegt und berechnet werden.

Zudem muss theoretisch vor der Bewertung der Stabilisierbarkeit zunächst eine erneute Optimierung der Güteparameter stattfinden. Da dieser Schritt jedoch schwer zu automatisieren ist und den Aufwand nochmals deutlich steigern würde, wird darauf verzichtet und es werden immer die im letzten Abschnitt optimierten QR-Parameter verwendet. Somit muss im Hintergrund behalten werden, dass die Ergebnisse mit geänderten Systemparametern möglicherweise durch eine QR-Optimierung besser ausfallen können.

In diesem Abschnitt wird bei der Einflussanalyse der Systemparameter ausschließlich das neukonstruierte Doppelpendel mit den Parametern `Ribeiro20` betrachtet (siehe Tabelle 2.4). Erste Tests mit dem System am Versuchsstand haben eine deutlich schlechtere Regelbarkeit im Vergleich zum vorigen Aufbau gezeigt. Wie bereits in Abschnitt 2.3.3 erwähnt, wird die erhöhte Coulomb-Reibung dafür verantwortlich gemacht, weswegen diese besonders in den Fokus genommen wird.

Es werden die Parameter des Schlitten-Pendel-System (siehe Tabelle 2.1) variiert, um deren Einfluss auf die Stabilisierbarkeit zu analysieren und die Regelbarkeit zu erhöhen. Dabei wird ein Parameter schrittweise geändert, alle anderen behalten ihren eigentlich Wert. Demzufolge ist bei der Interpretation auch darauf zu achten, dass manche Parameter wie Masse, Trägheitsmoment und Schwerpunkt eigentlich voneinander abhängen. Da außer den reinen Zahlenwerten der ermittelten Parameter keine weiteren Informationen über deren Zusammensetzung oder Modellierung vorliegen, werden die Parameter nur einzeln variiert.

3.6.2 Anwendung in MATLAB

Ähnlich wie bei der QR-Variation wird eine Funktion `SysParameter_Variation` angelegt, die für einen gegebenen Parametersatz den spezifizierten Parameter in dem angegebenen Bereich (hier meist linear) variiert. Der Parameter wird so angegeben wie in `SchlittenPendelParams` definiert. Für jeden Wert des Parameters wird das System und der Regler entsprechend initialisiert, anschließend werden die kritischen Anfangswerttests durchgeführt. Ein Aufruf erfolgt beispielsweise mit:

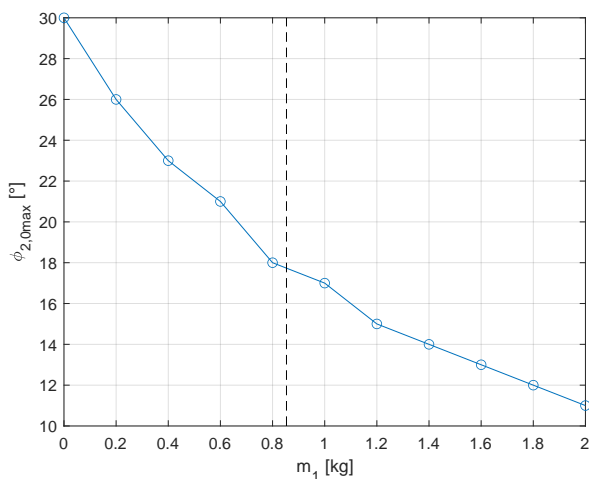
```
SysParameter_Variation(SchlittenPendelParams_Ribeiro20(), 'm1', 0:0.2:2)
```

Weitere Information sind in der Funktion dokumentiert. Zusätzlich befinden sich einige Aufrufbeispiele in `SysParameter_Variation_run.m`.

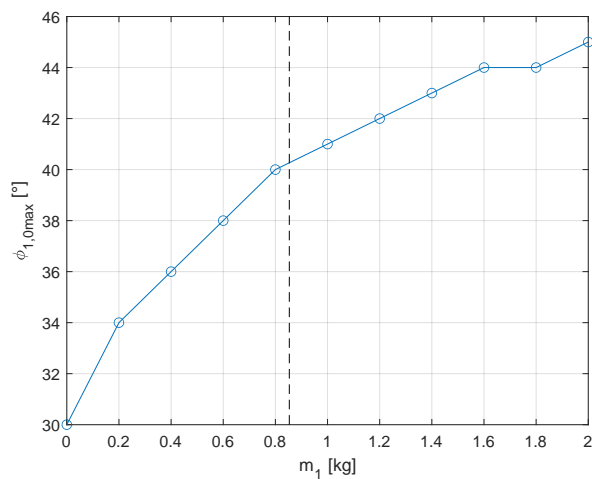
3.6.3 Diagramme

Trägheits- und Geometrieparameter

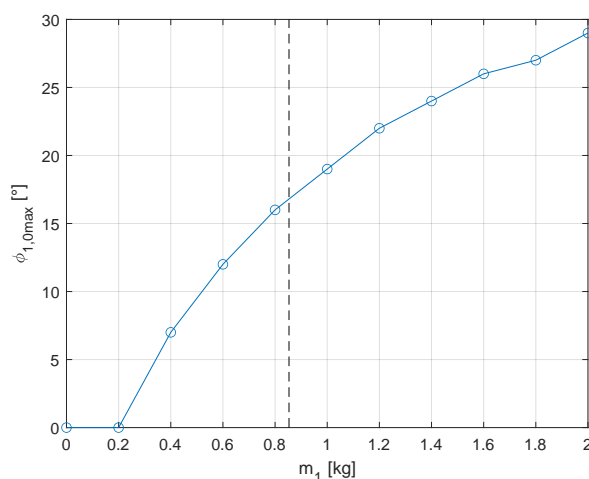
Abbildung 3.13 stellt die maximalen Startwerte der kritischen Anfangswerttests für die Variation der Pendelmasse m_1 im Bereich 0 bis 2 kg dar. Die gestrichelte Linie markiert den tatsächlichen Wert des Parameters (Ribeiro20). Wie man sehen kann, hat der Parameter einen starken Einfluss auf das Stabilitätsgebiet, wobei die Tendenz Arbeitspunkt-abhängig ist. Bei AP2 führt eine *geringere* Masse zu einem besseren Verhalten. Vermutlich kann dadurch das erste Pendel schneller beschleunigt werden, wodurch auch größere Winkelabweichungen ausgeregelt werden können. Im Gegensatz dazu führt bei AP3 und AP4 eine *höhere* Masse zu einer besseren Regelbarkeit. Bei diesen steht das erste Pendel, dessen Gelenk den nicht unerheblichen Coulomb-Reibungskoeffizienten aufweist, oben. Ein Erklärungsversuch besteht daher in der Coulomb-Reibung: Durch die erhöhte Masse wird sie relativ gesehen kleiner und damit das System „linearer“, was die bessere Regelbarkeit erklären könnte.



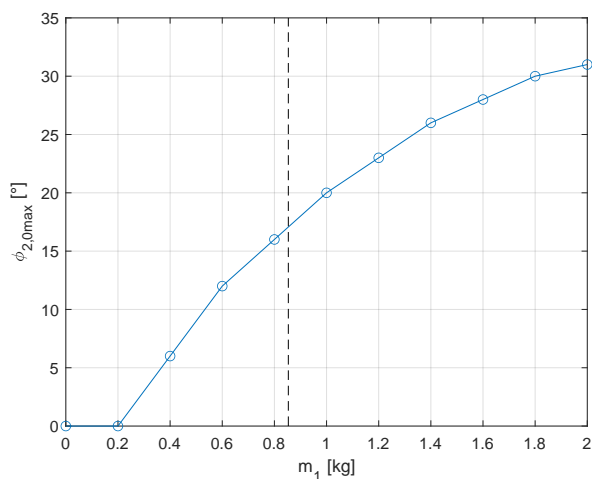
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)

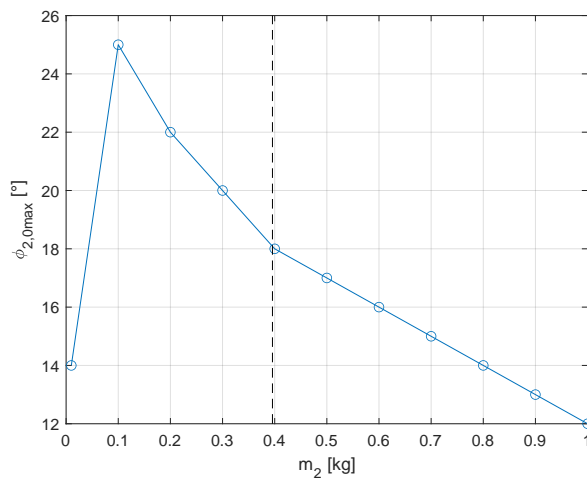


(c) AP4 (Auslenkung φ_1)

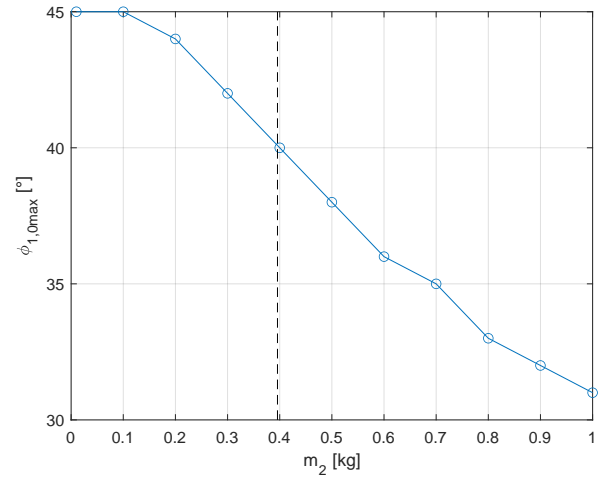


(d) AP4 (Auslenkung φ_2)

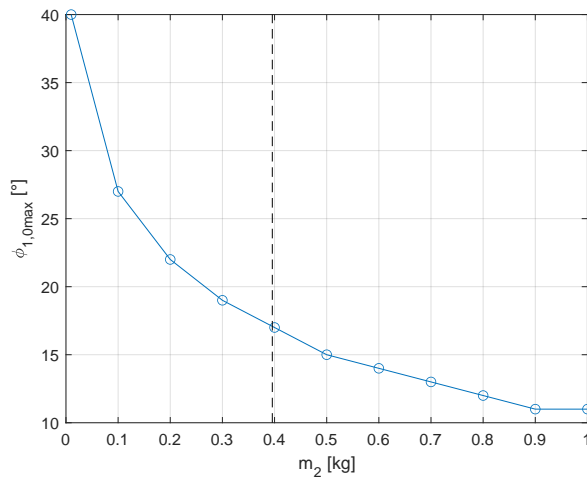
Abbildung 3.13: Maximale Startwerte – Variation m_1



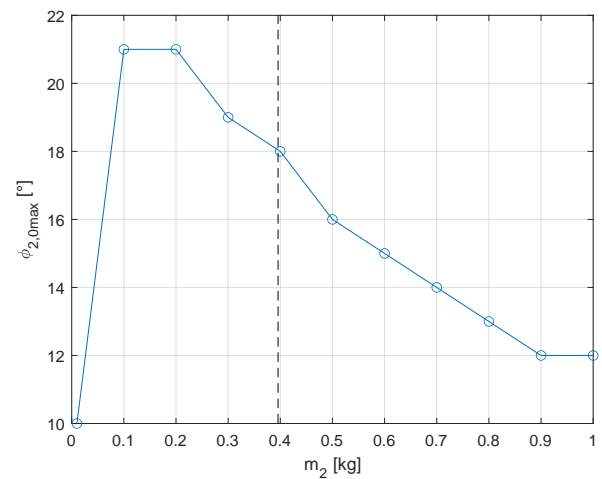
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)



(c) AP4 (Auslenkung φ_1)



(d) AP4 (Auslenkung φ_2)

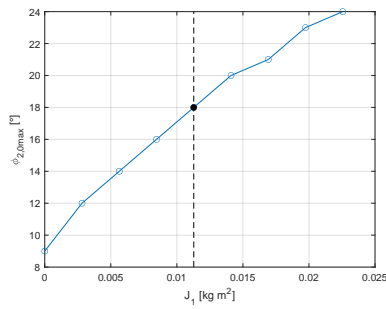
Abbildung 3.14: Maximale Startwerte – Variation m_2

Die Variation von m_2 im Bereich 0 bis 1 kg ist in Abbildung 3.14 dargestellt. Hier zeigt sich für alle Arbeitspunkte dieselbe eindeutige Tendenz: Je niedriger die Masse, desto besser ist die Stabilisierbarkeit.

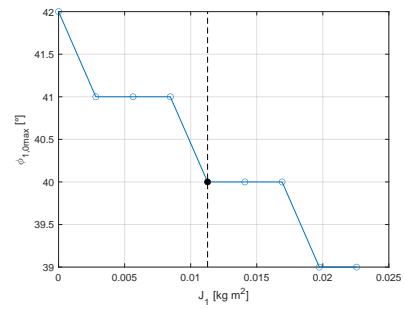
Für die Schlittenmasse m_0 zeigt sich der erwartete Trend: Je größer die Masse, desto weniger lassen sich die Arbeitspunkte stabilisieren (siehe Abbildung A.1). Dies lässt sich hauptsächlich auf die Stellgrößenbegrenzung zurückführen.

Das Trägheitsmoment J_1 wird im Bereich von 0 bis zum doppelten Originalwert variiert (siehe Abbildung 3.15). Es zeigt sich exakt der gegenteilige Trend im Vergleich zu m_1 : Bei AP2 verbessert ein *höheres* Trägheitsmoment das Regelverhalten, bei den anderen ein *niedrigeres*. Die „schnellere Drehbarkeit“ sorgt bei den Arbeitspunkten mit dem ersten Pendel oben offenbar für bessere Stabilisierbarkeit. Auch bei J_2 stellt sich der gegenteilige Trend zu m_2 ein: Ein höheres Trägheitsmoment *erhöht* die maximale Startabweichung.

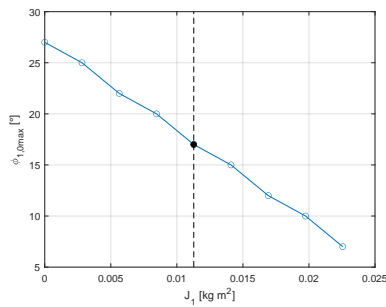
Bei s_1 zeigt sich eine ähnliche Tendenz wie bei m_1 , wobei die Änderungen ab Werten größer des



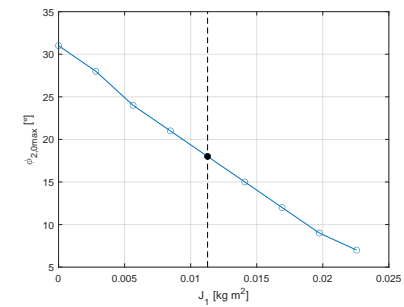
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)

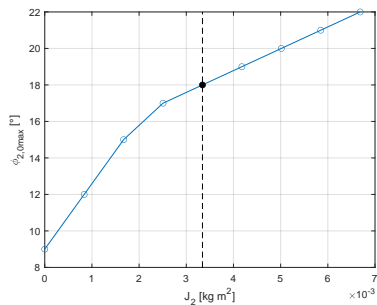


(c) AP4 (Auslenkung φ_1)

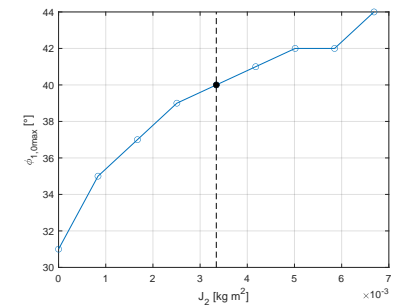


(d) AP4 (Auslenkung φ_2)

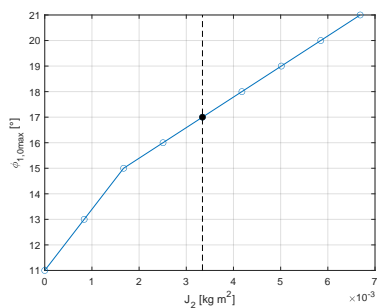
Abbildung 3.15: Maximale Startwerte – Variation J_1



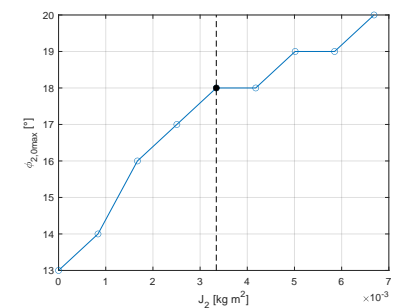
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)



(c) AP4 (Auslenkung φ_1)



(d) AP4 (Auslenkung φ_2)

Abbildung 3.16: Maximale Startwerte – Variation J_2

originalen gering sind (siehe Abbildung A.2).

Der Einfluss von s_2 ist nur bei AP3 deutlich und verbessert das Regelverhalten bei geringeren Werten. Auch bei den anderen Arbeitspunkten wird das Verhalten für geringere Werte etwas besser (siehe Abbildung A.3).

Für l_1 zeigt sich bei AP2 und AP3 derselbe Trend: Je größer, desto schlechter die Stabilisierung (siehe Abbildung A.4). Bei AP2 kann die Stabilität mit geringeren Werten deutlich verbessert werden. Bei AP4 hingegen verschlechtern kleine Werte das Verhalten deutlich, während größer Werte als die originalen nicht zu einer Verbesserung führen.

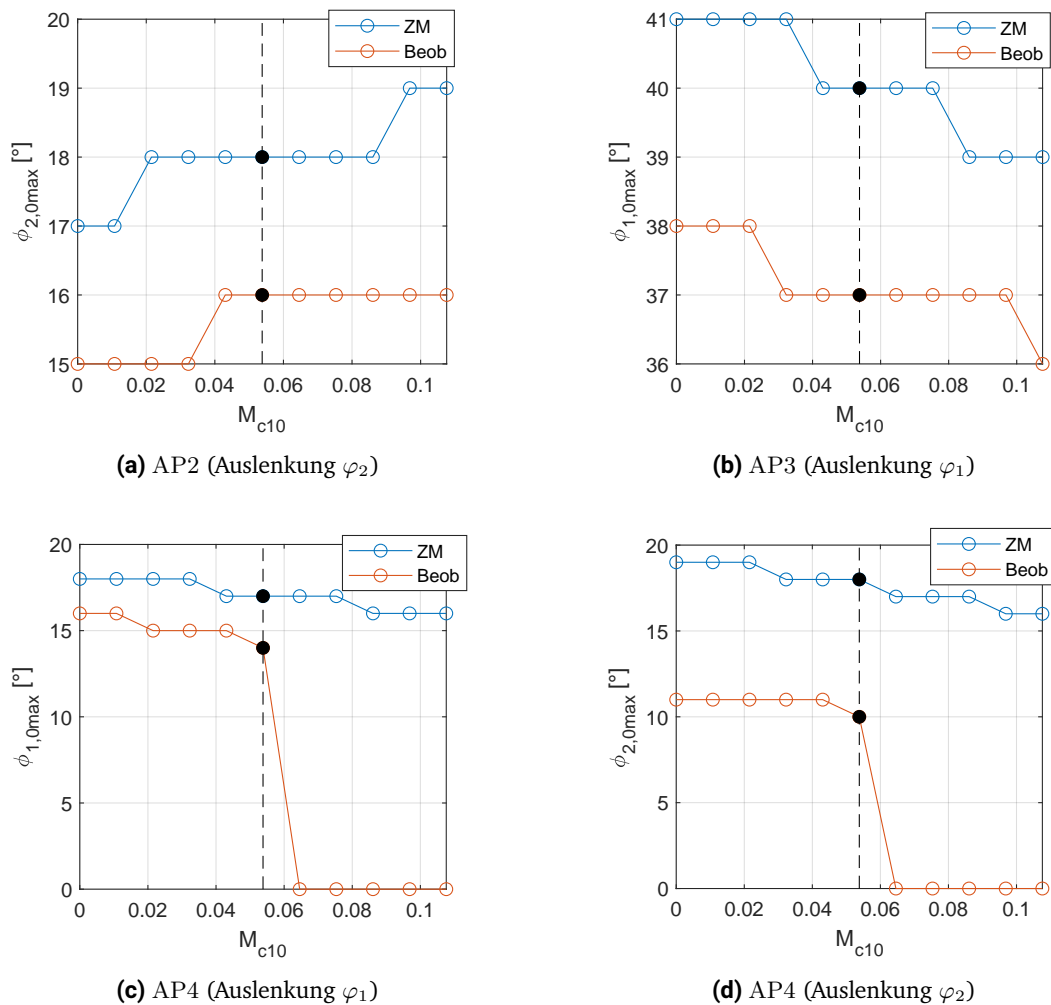


Abbildung 3.17: Maximale Startwerte – Variation M_{c10}

Reibungsparameter

Die Parameter der viskosen Dämpfung zeigten im Bereich 0 bis doppelter Originalwert keinen oder nur einen geringen Einfluss. Auch der Coulomb-Reibungskoeffizient des Schlittens F_{c0} zeigte keinen Einfluss, vermutlich weil die Schlittenreibung durch die Vorsteuerung sehr gut umgangen wird.

Anhand von Abbildung 3.17 sieht man, dass M_{c10} einen leicht negativen Einfluss hat, außer bei AP2, dort ist der Einfluss sogar positiv. Wie schon in Abbildung 3.7 festgestellt wurde, scheint es bei AP2 weniger Probleme bezüglich der Gelenk-Coulomb-Reibung zu geben und auch weniger Probleme mit Beobachter (siehe Abbildung 3.8). Da außerdem ein deutlicher Einfluss des Beobachters gerade am neuen System vorhanden ist (siehe Abbildung 3.8), wird für M_{c10} der Test ebenfalls mit Beobachter (und den dafür optimierten QR-Parametern) durchgeführt. Bei AP4 stellt sich heraus, dass schon geringfügig höherer Werte der Reibung das System überhaupt nicht mehr stabilisieren lassen. Offensichtlich ist gerade die Kombination von erhöhter Coulomb-Reibung, die das System auch in der Nähe des Arbeitspunktes nichtlinear macht, und dem Einsatz des Beobachters, welcher am linearen System ausgelegt ist und durch die Coulomb-Reibung Schätzfehler entstehen, äußerst kritisch.

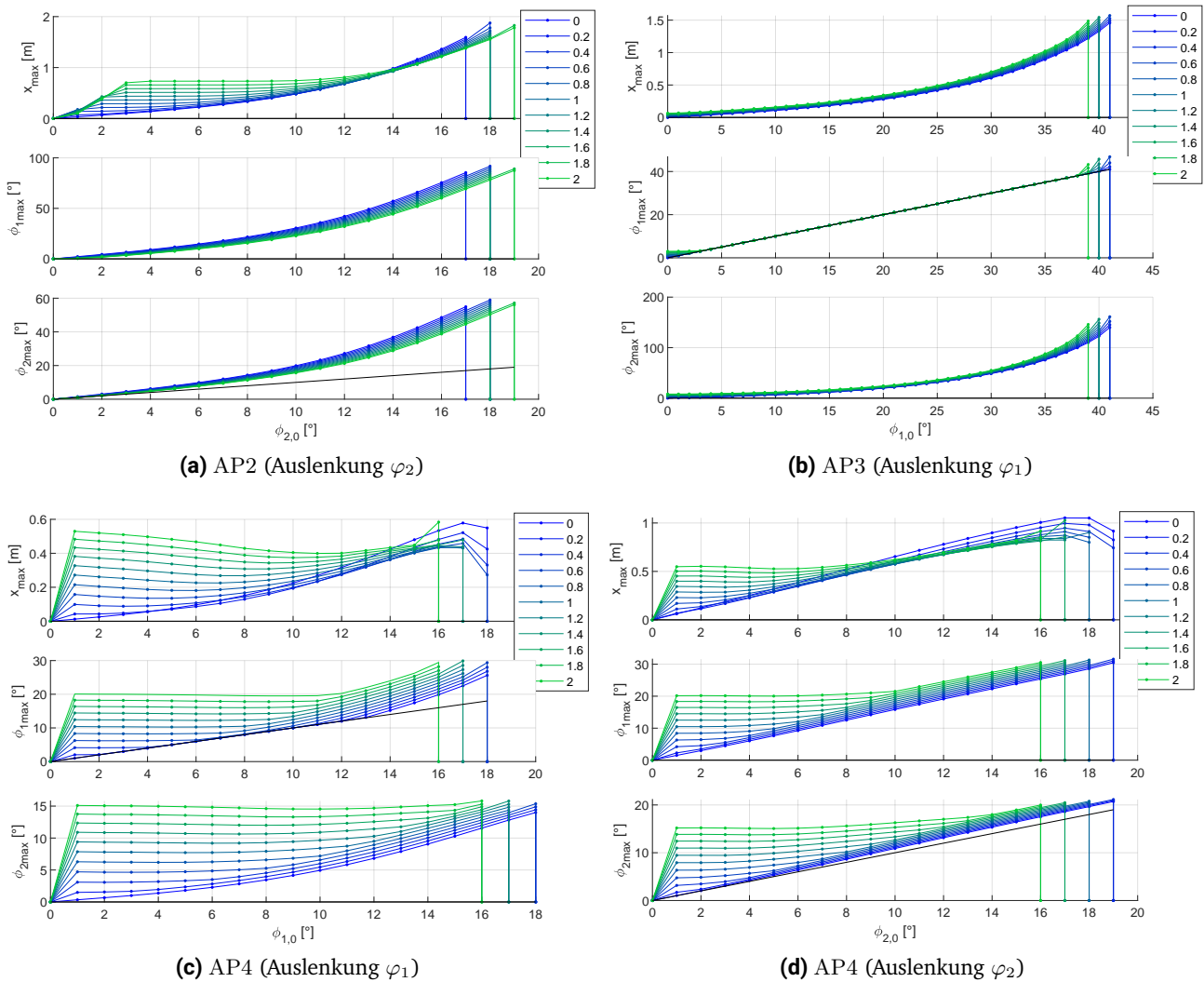


Abbildung 3.18: Maximalabweichungen – Variation M_{c10} in Abh. des Originalwerts (0.0538)

Abbildung 3.18 zeigt zudem die Arbeitspunktabweichungen bei der Variation von M_{c10} (Zustandsmessung). Hier lässt sich sehr gut der Einfluss auf den sich einstellenden Grenzzyklus insbesondere bei niedrigen Startwerten feststellen, wie er auch schon in Abbildung 3.7 beobachtet wurde. Bei $M_{c10} = 0$ beginnen die Kurven in 0 und verhalten sich anfangs linear. Mit steigender Reibung wird jedoch die

Dauerschwingung größer. Besonders bei AP4 sieht man, dass das System schon bei sehr kleinen Anfangswerten aus der Ruhe in den Grenzyklus gebracht wird. Bei AP2 startet die Dauerschwingung zwar nicht bei 0, ist aber im Bereich von 2-3 bis etwa 10 ° konstant. Bei AP3 ist der Einfluss geringer.

Bei M_{c20} zeigte sich bei einer Änderung bis zum 4-fachen Originalwert keinen Einfluss.

3.6.4 Schlussfolgerungen zu Parameteränderungen

Es stellt sich nun die Frage inwieweit die Pendelparameter verändert werden können, um das Regelverhalten zu verbessern. Insbesondere sind die Parameter von Interesse, die ohne großen konstruktiven Aufwand angepasst werden können.

Aus den Verläufen der Trägheits- und Geometrieparameter kann noch keine direkte Schlussfolgerung zu Änderungen am System gezogen werden, da diese alle voneinander abhängen. Wird beispielsweise eine zusätzliche Masse angebracht, ändert sich auch das Trägheitsmoment sowie der Schwerpunkt. Daher müssen zuvor alle Parameter entsprechend angepasst und weitere Tests durchgeführt werden, bevor eine eindeutige Aussage getroffen werden kann.

Bei der Neukonstruktion [17] wurden an den Pendelstäben Gewindebohrungen realisiert, sodass zusätzliche Gewichte angebracht werden können, die für ein anderes Systemverhalten sorgen. Erste Überlegungen sind daher, am ersten Pendel eine Masse anzubringen, dies laut Diagramm für AP3 und AP4 zunächst vorteilhaft ist. Andererseits hat das dadurch höhere Trägheitsmoment einen negativen Einfluss, was für eine Anbringung nahe der Drehachse spricht. Die Ergebnisse der Schwerpunktseinflusses sprechen bei AP3 und AP4 hingegen für eine Erhöhung des Schwerpunkts, also eine Anbringung näher am zweiten Gelenk. Aufgrund der gegensinnigen Tendenzen sind weitere Tests für bestimmte Ausführungen nötig.

Auch am zweiten Pendel könnten Massen angebracht werden. Während das Diagramm der reinen Masse zunächst dagegen spricht, verbessert ein höheres Drehmoment die Stabilisierbarkeit. Demzufolge könnte der Einfluss einer Masse an der Pendelspitze getestet werden. Außerdem könnte die Masse durch Verkürzen des Pendels verringert werden.

Außerdem hat sich bei Variation der Fallbeschleunigung g gezeigt, dass niedrigere Werte die Stabilisierbarkeit verbessern. Daher könnte man überprüfen, ob sich der Versuchsstand zu einem Standort auf dem Mond verlegen lässt.

4 Trajektorien

Da es bei der Trajektorienfolgeregelung aufgrund der Beschränkung einiger Systemzustände in der Vergangenheit immer wieder zu Schwierigkeiten bei der Berechnung und Umsetzung von Trajektorien kam, wurde durch Fauvé [8] der Ansatz der Modellprädiktiven Regelung (engl.: *Model Predictive Control*, MPC), die im Gegensatz zu den klassischen Regelungsverfahren eine Berücksichtigung der Systembeschränkungen ermöglicht, für das bestehende Pendelsystem untersucht. Obwohl sich der Ansatz aufgrund ungenügender Echtzeitfähigkeit nicht für die Regelung eignete, erwies er sich in Bezug auf die Berechnung von Trajektorien für Arbeitspunktwechsel als vielversprechend. Zur Trajektorienberechnung wurden die nichtlinearen Systemgleichungen verwendet, sodass hierbei eine nichtlineare MPC, kurz NMPC, vorliegt. Erste Versuche, die mit der NMPC berechneten Trajektorien in der Simulation mit Hilfe einer Trajektorienfolgeregelung zu stabilisieren, gelangen jedoch nicht. Daher bleibt zu zeigen, dass die gefundenen Trajektorien am System stabilisiert werden können. Zudem wird vermutet, dass neben der bereits betrachteten Regelbarkeit auch das Ergebnis der Trajektorienberechnung durch die Systemparameter beeinflusst wird.

Zunächst wird daher überprüft, inwiefern die berechneten Trajektorien durch eine Trajektorienfolgeregelung stabilisiert werden können. Außerdem wird eine geeignete Konfiguration für die NMPC ermittelt, die anschließend zur Durchführung einer Einflussuntersuchung der Systemparameter auf die Trajektorienberechnung verwendet wird.

Um das geplante Untersuchungsvorhaben durchführen zu können, werden im ersten Schritt eine Trajektorienberechnung und eine Trajektorienfolgeregelung sowie die benötigten Auswertungsfunktionen implementiert.

4.1 Trajektorienberechnung

Die Implementierung der Trajektorienberechnung basiert auf den Erkenntnissen der Vorgängerarbeit Fauvé [8] und erfolgt in MATLAB mit Hilfe des *CasADi*-Frameworks [2]. Es wird einerseits auf eine Erhöhung der Modularisierung im Vergleich zur Vorgängerarbeit geachtet, andererseits werden weitere Funktionalitäten ergänzt, um die Trajektorienberechnung für die Anwendungen im Rahmen dieser Arbeit und die zukünftige Weiterverwendung zu optimieren.

Die Berechnung der Trajektorien wird durch fünf Matlab-Funktionen modularisiert:

- `searchTrajectories`
- `calculateTrajectory`
- `getODE`

- `getInitDev`
- `determineAPinit`

Vom Anwender aufgerufen wird lediglich `searchTrajectories`. Die weiteren vier Funktionen werden auch für andere Anwendungen dieser Arbeit eingesetzt, wodurch das Kopieren von Code vermieden wird. Änderungen, die mehrere Funktionen und Skripte betreffen, können daher effizient an einer Stelle im Programmcode durchgeführt werden. Die Argumente der Funktionen werden so gewählt, dass die in dieser Arbeit zu untersuchenden Parameter leicht durch den Funktionsaufruf in einer Schleife variiert werden können. Weitere Parameter die im Rahmen dieser Arbeit innerhalb der Funktionen konstant festgelegt sind, können zu einem späteren Zeitpunkt jedoch leicht zu den Funktionsargumenten ergänzt werden.

Die implementierten Werkzeuge zur Trajektoriensuche eignen sich für eine weitere Verwendung in zukünftigen Arbeiten. Obwohl im Rahmen dieser Arbeit in erster Linie eine definierte Vergleichstrajektorie verwendet wird, ist eine systematische Suche für alle zwölf Trajektorien im Sinne zukünftiger Anwendungen vorgesehen. Diese wurde im Rahmen dieser Arbeit erfolgreich durch eine Reproduktion der Ergebnisse von Fauvé [8] getestet.

Im Folgenden werden die wichtigsten Funktionen `searchTrajectories` und `calculateTrajectory` genauer beschrieben, wobei auch auf die Verwendung der darüber hinaus genannten Funktionen eingegangen wird.

4.1.1 `searchTrajectories`

Durch die Funktion `searchTrajectories` wird vom Anwender eine Trajektoriensuche in Auftrag gegeben. Als Beispiel für die Anwendung kann der folgende Programmcode eines MATLAB-Skriptes betrachtet werden:

```

1  %% Demo - Aufruf von searchTrajectories
   mode = 2;           % Suchprogramm
   N = 500;            % Prädiktionshorizont
   T = 0.005;         % Schrittweite
   sol = 'RK4';        % Integrator für Kontinuitätsbedingung
6  params= SchlittenPendelParams_Apprich09(); % Parametersatz
   u_max = 410;        % Maximale Stellkraft
   coulMc = false;     % Gelenkreibung nicht berücksichtigen
   coulFc = true;      % Schlittenreibung berücksichtigen

11 % Starte Suche nach Trajektorien
   searchTrajectories(mode, N, T, sol, params, u_max, coulMc, coulFc)

```

Zunächst wird ein Suchprogramm ausgewählt. Es kann zwischen den drei Programmen

1. Suche nach der im Rahmen dieser Arbeit definierten Vergleichstrajektorie
2. Suche nach der Aufschwingtrajektorie mit der Variationsstrategie
3. Suche nach allen 12 Trajektorien mit der Variationsstrategie

ausgewählt werden.

Dem Ansatz der Programmauswahl liegen die Erkenntnisse von Fauvé [8] bezüglich einer Variationsstrategie zu Grunde. Da auf Grund der nichtlinearen MPC nur lokal optimiert wird, kann je nach Anfangswert ein anderes Optimum erreicht werden. Eine Variation des Anfangswertes ermöglicht somit eine effektive Suche nach Trajektorien für einen bestimmten Arbeitspunktwechsel. Im Code-Beispiel ist Suchprogramm 2 ausgewählt, der gesuchte Arbeitspunktwechsel ist allgemein der Aufschwung von AP1 nach AP4. Hierbei können innerhalb einer Periode vier mögliche Winkelanfangsauslenkungen gegenüber AP4 variiert werden, die als Startwert AP1 definieren:

$$\begin{bmatrix} 0 \\ 0 \\ \pi \\ 0 \\ \pi \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -\pi \\ 0 \\ \pi \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \pi \\ 0 \\ -\pi \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -\pi \\ 0 \\ -\pi \\ 0 \end{bmatrix}.$$

Eine weitere Variation durch Ausnutzung der 2π -Periodizität bringt gemäß Fauvé [8] hingegen keine Vorteile, sondern verlängert die Berechnungszeit bei gleichzeitig sinkender Wahrscheinlichkeit, dass der Algorithmus zu einer gültigen Lösung konvergiert. Die möglichen Anfangsauslenkungen werden über die Funktion `getInitDev` aufgerufen. Neben den Anfangsauslenkungen der Pendel hat auch die Schlittenposition Einfluss auf die Lösungsfindung. Daher wird diese mit den folgenden auf Fauvé [8] basierenden Auslenkungen symmetrisch zur Bahnmitte variiert (siehe Tabelle 4.1).

Tabelle 4.1: Variation der Schlittenposition

$x_{0,\text{init}}$	$x_{0,\text{end}}$
-0,5	0,5
-0,3	0,3
-0,1	0,1
0	0

Als letzte Komponente der Variationsstrategie wird noch die Positionsbeschränkung variiert. Obwohl der Versuchsstand eine Positionsbeschränkung von $\pm 0,8$ m vorgibt, ist es sinnvoll, auch diese zu variieren, da der Algorithmus sich sensitiv gegenüber den Nebenbedingungen verhält. Durch Verschärfung oder Lockerung der Positionsbeschränkung verändert sich die Optimierungslandschaft und damit die Chance, eine Trajektorie zu finden. Bei Lockerungen der Beschränkung muss später immer geprüft werden, ob bei den gefundenen Trajektorien die Positionsbeschränkungen eingehalten werden. Es werden folgende von Fauvé [8] entnommene Variationen implementiert.

$$-\underline{g}_{x_0} = \bar{g}_{x_0} \in \{0,6; 0,8; 1; 1,2; 1,4\}$$

Insgesamt werden bei Suchprogramm 2 somit $4 \cdot 4 \cdot 5 = 80$ Trajektorienberechnungen durchlaufen. Bei Suchprogramm 3 werden die beschriebenen Variationen auf alle zwölf Trajektorien ausgeweitet, indem der Endwert durch alle vier Arbeitspunkte iteriert und die Variation der Winkelanfangsauslenkungen um die vier zusätzlich entstehenden Kombinationen durch die Arbeitspunkte 2 und 3 erweitert wird. Insgesamt werden $8 \cdot 4 \cdot 5 \cdot 4 = 640$ Trajektorienberechnungen durchgeführt. Im Rahmen dieser Arbeit wird eine Vergleichstrajektorie für die in Kapitel 4.4 behandelten Parameteruntersuchungen verwendet. Diese wird mit Suchprogramm 1 berechnet.

Nachdem das Suchprogramm für die Trajektorienberechnung ausgewählt ist, werden Prädiktionshorizont, Schrittweite und Integrationsverfahren für die NMPC konfiguriert. Als Integratoren stehen das Eulerverfahren „Euler“ und das Runge-Kutta-Verfahren „RK4“ zur Verfügung. Außerdem wird einer der in Kapitel 2.3.3 besprochenen Parametersätze übergeben sowie eine maximale Stellkraft definiert. Diese stellt eine weitere Variationsmöglichkeit dar, die von Fauvé [8] nicht untersucht worden ist. Daher wird sie zur manuellen Variation in den Argumenten der `searchTrajectories` zur Verfügung gestellt. Als Letztes wird noch bestimmt, ob für die Trajektorienberechnung die Coulomb-Reibung im Schlitten bzw. in den Gelenken berücksichtigt werden soll.

Die berechneten Trajektorien werden automatisch gespeichert. Es wird eine Namenskonvention zur Identifizierbarkeit der gespeicherten Ergebnisse eingeführt, die an folgendem Trajektoriennamen beispielhaft gezeigt wird: `Traj14_dev0_-3.14_-3.14_x0max0.8_Fmax410`.

Am Anfang steht die verallgemeinerte Trajektorienbezeichnung mit AP_{init} und AP_{end} , wobei AP_{init} erst noch aus den Anfangsauslenkungen abgeleitet werden muss. Dies geschieht mit Hilfe der Funktion `determineAPinit`. Es folgen der Variationswert der Schlittenposition, die Winkelanfangsauslenkung von φ_1 und φ_2 , die Positionsbeschränkung und die maximale Stellkraft. Wenn nicht anders angegeben, werden die Ergebnisse im Ordner `searchResults` in einer definierten Ordnerstruktur gespeichert, deren Benennung sich nach der Konfiguration der NMPC richtet. Im obigen Beispiel werden die Ergebnisse in folgendem Unterordner gespeichert, wobei `Fc` anzeigt, dass die Coulomb-Reibung des Schlittens in der Berechnung berücksichtigt wurde: `Results_app09_Fc_T0.005N500_RK4`.

Optional kann der Funktion `searchTrajectories` auch ein Pfad für den Speicherordner übergeben werden. Außerdem können zusätzliche Erweiterungen für den Trajektoriennamen übergeben werden, um später bei der Variation von Modellparametern die Variationswerte im Dateinamen zu berücksichtigen. Als weitere Option kann noch bestimmt werden, ob alle berechneten Trajektorien oder nur die mit gültiger Lösung gespeichert werden sollen.

4.1.2 calculateTrajectory

Die Funktion `calculateTrajectory` wird innerhalb von `searchTrajectories` zur eigentlichen Berechnung einer bestimmten Trajektorie aufgerufen. Sie implementiert die NMPC mit Hilfe des *CasADi*-Frameworks auf Grundlage der Erkenntnisse der Vorgängerarbeit. Für detaillierte Hintergrundinformationen wird daher auf die Ausarbeitung von Fauvé [8] verwiesen. Als Argument erwartet die Funktion eine Struktur mit verschiedenen Konfigurationsparametern, die im Quelltext ausführlich kommentiert sind. Da die Funktion `calculateTrajectory` eigens für die Trajektorienberechnung vorgesehen ist, wird, anders als bei Fauvé [8], auf die Implementierung einer Iterationsschleife für die Anwendung als Regler verzichtet.

Zielfunktion

Zunächst muss ein Optimalsteuerungsproblem (OCP) mit Zielfunktion und Nebenbedingungen definiert werden. Als Zielfunktion wird

$$J = \sum_{i=1}^N ((x_i - x_{\text{end}})^T \mathbf{Q} (x_i - x_{\text{end}}) + (u_i - u_{\text{end}})^T R (u_i - u_{\text{end}})) + \sum_{i=2}^N (u_i - u_{i-1})^T S (u_i - u_{i-1})$$

implementiert. Sie besteht aus dem bekannten *Lagrange'schen* Güteintegral der LQ-Regelung und einem Bestrafungsterm für Änderungen in der Stellgröße, durch den hochfrequente Stellgrößenverläufe reduziert werden. Die Wahl der Gewichtungsmatrizen \mathbf{Q} , R und S basiert auf den Erkenntnissen von Fauvé [8]. Tests bestätigen diese Wahl, wobei S etwas nach unten korrigiert wird. Die Matrizen werden anschließend nicht mehr variiert. Sie sind der Funktion `calculateTrajectory` als Konfigurationsparameter zwar zu übergeben, werden in `searchTrajectories` jedoch für die weiteren Anwendungen im Rahmen der Arbeit hartkodiert.

$$\mathbf{Q} = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,1 \end{bmatrix}, \quad R = 5 \cdot 10^{-7}, \quad S = 1,5 \cdot 10^{-8}$$

Nebenbedingungen

Neben der Zielfunktion sind darüber hinaus folgende Nebenbedingungen zu implementieren:

- Kontinuitätsbedingungen
- Anfangsbedingung
- Endwertbedingung
- Positionsbegrenzung des Schlittens
- Stellkraftbegrenzung

Dabei ist zu beachten, dass die Endwertbedingung, anders als die restlichen Nebenbedingungen, nicht als harte Grenze formuliert, sondern implizit durch die Zielfunktion angenähert wird.

Um eine möglichst hohe Vergleichbarkeit mit der Vorgängerarbeit Fauvé [8] zu erzielen, wird für die Systemgleichungen das Kraftmodell (2.8) gewählt. Die nichtlinearen Gleichungen werden mit Hilfe der Funktion `getODE` in *CasADi-Symbolik* zur Verfügung gestellt.

Die Stellstrombegrenzung in Abhängigkeit der Winkelgeschwindigkeit des Motors wird in Abschnitt 2.2.2 hergeleitet. Diese muss für die Implementierung noch in eine Stellkraftbegrenzung in Abhängigkeit der Schlittengeschwindigkeit umgerechnet werden. Mit

$$\omega = \frac{1}{r_{32} K_G} \dot{x}_0$$

und

$$I_a = \frac{r_{32} K_G}{K_I} \cdot F$$

kann Gleichung (2.17) in die Form

$$F_{\max} = \frac{K_I}{r_{32} K_G} \cdot \frac{U_{a,\max}}{R} - \left(\frac{K_I}{r_{32} K_G} \right)^2 \cdot \frac{\dot{x}_0}{R}$$

gebracht werden. Gleiches gilt für (2.15). Die Fallunterscheidung nach Vorzeichen der Bewegungsrichtung wird wie in Abschnitt 2.2.2 berücksichtigt.

In *CasADi* sind zwei Arten von Beschränkungen zu unterscheiden: Pfadbeschränkungen in Form von Gleichungen und Ungleichungen (z.B. Systemgleichungen) sowie Ober- und Untergrenzen für die Wertebereiche der Optimierungsvariablen (z. B. $u_{\min} \leq u \leq u_{\max}$). Die konstante Stellbegrenzung durch den Spannungs-Strom-Wandler wird durch eine Begrenzung des Wertebereichs der Eingangsgröße F realisiert. Die variable Stellbegrenzung durch die Gegeninduktion wird hingegen als Pfadbeschränkung durch die Ungleichungen

$$0 \leq \frac{K_I}{r_{32} K_G} \cdot \frac{U_{a,\max}}{R} - \left(\frac{K_I}{r_{32} K_G} \right)^2 \cdot \frac{\dot{x}_0}{R} - F$$

und

$$0 \geq -\frac{K_I}{r_{32} K_G} \cdot \frac{U_{a,\max}}{R} - \left(\frac{K_I}{r_{32} K_G} \right)^2 \cdot \frac{\dot{x}_0}{R} - F$$

implementiert.

Multiple Shooting

Das zu lösende OCP wird durch Diskretisierung in ein endlich-dimensionales nichtlineares Programm (engl.: *Nonlinear Programming*, NLP) überführt, was auch als direkter Ansatz bzw. „first discretize, then optimize“ bezeichnet wird. Dies geschieht durch Einteilung des OCP in N Intervalle der Schrittweite T , wobei N als Zeit- oder Prädiktionshorizont bezeichnet wird. Als numerische Lösungsmethode wird das in Fauvé [8] empfohlene *Direct-Multiple-Shooting*-Verfahren implementiert. Neben dem Eingangsgrößenverlauf zählt bei diesem Verfahren auch der Zustandsgrößenverlauf zu den Optimierungsvariablen. Anders als beim *Single Shooting* werden anstelle des gesamten Prädiktionshorizonts für jedes Intervall die Systemgleichungen gelöst. Dadurch wird das Randwertproblem zunächst in N Anfangswertprobleme zerlegt, die durch die Systemgleichungen und einen jeweils zu optimierenden Startwert definiert werden. Um sicherzustellen, dass die abschnittsweise gelösten Systemgleichungen einen stetigen Verlauf ergeben, werden N Kontinuitätsbedingungen implementiert. Der Lösungsverlauf eines Intervalls soll zum nächsten Zeitschritt dem Startwert des nächsten Intervalls entsprechen. Vorteil des Verfahrens gegen über dem *Single Shooting* ist, dass die Schrittfehler des Integrationsverfahrens sich lediglich über ein Intervall fortpflanzen anstelle des gesamten Prädiktionshorizonts. [7]

Integrations- und Optimierungsverfahren

Zur Lösung der Systemgleichungen werden das Euler-Verfahren und das Runge-Kutta-Verfahren 4. Ordnung implementiert. Als Lösungsverfahren für das NLP wird nach dem Beispiel von Fauvé [8] das von *CasADi* bereitgestellte IPOPT-Verfahren (Innere-Punkte-Verfahren, engl.: *Interior Point Optimization*) eingesetzt [6]. Die maximale Anzahl an Optimierungsiterationen wird großzügig mit 20000 Iterationen begrenzt, da bei zugeschalteter Reibung in den Systemgleichungen die durchschnittliche Iterationszahl deutlich ansteigen kann.

Initialisierung

Vor Beginn der Optimierung müssen alle Optimierungsvariablen mit Startwerten initialisiert werden. Auf Grund des Multiple-Shooting-Ansatzes sind Startwerte über den gesamten Prädiktionshorizont sowohl für die Eingangsgröße als auch für den Zustandsvektor zu schätzen. Für den diskretisierten Eingang u werden die Startwerte mit dem Nullvektor $u_0 = [0 \ 0 \ \dots \ 0]^T \in \mathbb{R}^{1 \times N}$ initialisiert. Für die Zustände x wird ein S-förmiger Verlauf von x_{init} nach x_{end} geschätzt, da hiermit bei Fauvé [8] gute Ergebnisse erzielt wurden. Der Verlauf wird durch Interpolation mit Hilfe der von MATLAB bereitgestellten Funktion `chip` über 4 Stützstellen gewonnen. Die zugrunde liegende Erwartung ist, dass das System erst langsam anfährt, dann stark beschleunigt und sich am Ende in der Nähe des Zielwerts nur noch langsam ändert.

4.2 Trajektorienfolgeregelung

Zur Überprüfung der Stabilisierbarkeit der berechneten Trajektorien wird eine Trajektorienfolgeregelung (TFR) als zeitvarianter Zustandsregler nach [13] realisiert.

Dazu wird das nichtlineare System zunächst wie in Abschnitt 3.1.1 linearisiert. Die Linearisierung um die Trajektorie führt im Gegensatz zum zeitinvarianten Arbeitspunkt auf ein *linear-zeitvariantes System* (LZV):

$$\Delta \dot{x}(t) = \mathbf{A}(t)\Delta x(t) + \mathbf{B}(t)\Delta F(t)$$

mit

$$\begin{aligned} x(t) &= x_{\text{Traj}}(t) + \Delta x(t) \\ F(t) &= F_{\text{Traj}}(t) + \Delta F(t) . \end{aligned}$$

Hierfür lässt sich der linear-zeitinvariante LQ-Reglerentwurf aus Abschnitt 3.2.1 übertragen, indem das zeitvariante Gütemaß

$$J = \int_0^\infty \Delta x^T(t) \mathbf{Q} \Delta x(t) + R \cdot (\Delta F(t))^2 dt$$

durch Lösen der zeitvarianten *Riccati*-Differentialgleichung

$$\dot{\mathbf{P}}(t) = \mathbf{P}(t)\mathbf{B}(t)R^{-1}\mathbf{B}^T(t)\mathbf{P}(t) - \mathbf{P}(t)\mathbf{A}(t) - \mathbf{A}^T(t)\mathbf{P}(t) - \mathbf{Q}$$

minimiert wird. Die Gleichungen werden als Endwertproblem rückwärts in der Zeit gelöst, wobei das System der Endlage $t_{\text{end}} = (N + 1) \cdot T$ als Randwert vorgegeben wird.

Damit lässt sich die Reglerverstärkung

$$\mathbf{K}(t) = R^{-1}\mathbf{B}^T(t)\mathbf{P}(t)$$

für das lineare, zeitvariante Regelgesetz

$$\Delta F(t) = -\mathbf{K}(t)\Delta x(t)$$

berechnen.

Für die Implementierung der Linearisierung und die Berechnung der Reglerverstärkung werden die vom Fachgebiet RTM zur Verfügung gestellten „common“-Funktionen `linSys` und `getTrajFBController_LQR` verwendet. Um die Regelung zu initialisieren wird das MATLAB-Skript `InitTrajReg` erstellt. Die Initialisierung beinhaltet das Laden eines Modellparametersatzes und der zu simulierenden Trajektorie, die Berechnung des trajektorienspezifischen Zustandsreglers sowie die Bereitstellung der Reglerdaten für die Simulation.

Das Simulationsmodell ist in Abbildung 4.1 dargestellt.

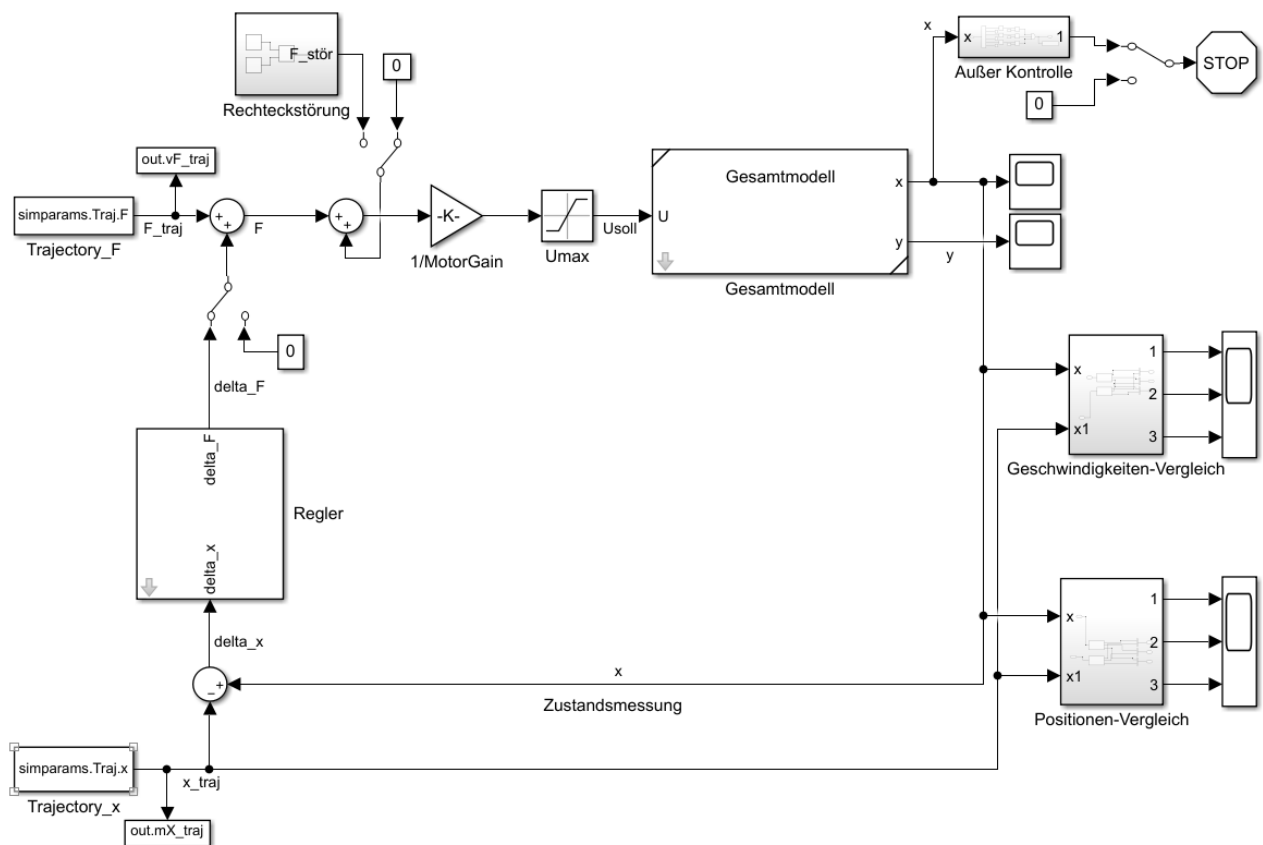


Abbildung 4.1: Trajektorienfolge Regelung in Simulink

Die Trajektorien werden der Simulation als `timeseries` mithilfe von `FromWorkspace` Blöcken zugeführt. Für die anschließende Auswertung in MATLAB hingegen können benötigte Signale über `ToWorkspace` an MATLAB gereicht werden. Um instabiles Verhalten zu erkennen und vorzeitig zu beenden, wird das Subsystem `AußerKontrolle` implementiert. Optional kann die Regelung zu Vergleichszwecken abgeschaltet werden. Ebenso kann kurzzeitig eine Rechteckstörung aufgeschaltet werden. Die genannten Optionen lassen sich über `ManualSwitches` ein- und ausschalten, wobei deren Zustand aus MATLAB heraus über die Funktion `set_param` angesteuert werden kann.

4.3 Stabilisierung der berechneten Trajektorien

Für die mittels NMPC berechneten Trajektorien soll gezeigt werden, dass eine Stabilisierung in der Simulation mit Hilfe eines Trajektorienfolgereglers möglich ist. Im Rahmen der Vorgängerarbeit Fauvé [8] war es nicht gelungen, die berechneten Trajektorien am störungsfreien System durch einen zeitvarianten Regler zu stabilisieren. Um das System zu stabilisieren, waren die Gewichtungsmatrizen Q und R für die Berechnung des Reglers variiert worden.

Aus diesem Grund wird ein anderer Ansatz gewählt, um eine Stabilisierung der mittels NMPC berechneten Trajektorien zu erreichen. Aus dem Skript [12] zur Vorlesung *Modellbildung und Simulation* ist bekannt, dass die Stabilität der Simulation maßgeblich von der Schrittweite und dem Integrationsverfahren abhängt. Bezüglich der Trajektorien können diese an zwei Stellen variiert werden: Einerseits innerhalb der NMPC zur Berechnung der Trajektorie und andererseits in der Simulation in SIMULINK. Lässt sich eine Trajektorie mit verschiedenen Integrationsverfahren stabil simulieren, wird davon ausgegangen, dass das (ideale) System stabil ist. Über die Integration innerhalb der NMPC soll hingegen die Güte der berechneten Trajektorie variiert werden.

4.3.1 Vorgehen

Im Gegensatz zu den linearen Systemen können für nichtlineare Systeme nicht ohne Weiteres Stabilitätsgebiete für die verschiedenen Simulationsverfahren angegeben werden. Daher sind prinzipiell nur heuristische Ansätze anwendbar, um eine geeignete Schrittweite zu finden. Allgemein ist bei einer kleineren Schrittweite ein geringerer Schrittfehler und somit eine höhere Güte der Trajektorie zu erwarten. Andererseits steigen Rundungsfehler und Rechenzeit an. Bezüglich der Trajektorienberechnung wirkt sich besonders die Berechnungsdauer dominant aus. Bei einer kleineren Schrittweite muss darauf geachtet werden, dass der Prädiktionshorizont, der in Abtastpunkten angegeben wird, groß genug gewählt wird. Anderenfalls wird keine gültige Lösung mehr gefunden. Bei Fauvé [8] wurde für die Trajektorienberechnung eine Schrittweite von $T = 0,01$ (in Sekunden) bei einem Prädiktionshorizont von $N = 350$ verwendet. Versuche zur Findung einer geeigneten Variation zu den von Fauvé [8] verwendeten Parametern ergeben, dass $T = 0.005$ und $N = 500$ einen geeigneten Kompromiss aus möglichst kleiner Schrittweite und akzeptabler Rechenzeit liefern.

Als Integrationsverfahren zur Trajektorienberechnung wurde von Fauvé [8] das Euler-Verfahren eingesetzt. Als Alternative wird dem Euler-Verfahren nun das Runge-Kutta-Verfahren 4. Ordnung (RK4) gegenübergestellt.

Als weiterer Einflussfaktor auf die Güte der Trajektorien und somit auch auf ihre Stabilisierbarkeit werden die Systemparameter vermutet. Ihr Einfluss auf die Trajektorienberechnung wird in Abschnitt 4.4 näher untersucht. Hierbei soll die Variation einzelner Systemparameter von einem Anfangs-Parametersatz ausgehen, für den sich die definierte Vergleichstrajektorie finden und stabilisieren lässt. Daher wird neben Schrittweite und Integrationsverfahren auch zwischen den in Abschnitt 2.3.3 definierten Parametersätzen *Apprich* und *Ribeiro* unterschieden.

Die Versuche zur Stabilisierbarkeit werden störungsfrei und unter Vernachlässigung der Coulomb-Reibung durchgeführt. Auf das Verhalten unter Berücksichtigung der Reibwerte wird in Abschnitt 4.4 eingegangen.

Um eine Vergleichbarkeit zur Vorgängerarbeit herzustellen, werden die Versuche sowohl mit als auch ohne Berücksichtigung der Gegeninduktion des Motormodells durchgeführt.

Die Skripte `TFRSim_SchlittenPendel_run` und `TFRSim_Gesamtmodell_run` werden zur Durchführung der Simulationen implementiert. Beide Funktionen führen automatisiert drei Simulationen mit den Solvern `ode1` (Euler), `ode4` (RK4) und `ode45` (RK5(4) von Dormand-Prince) für eine mit `InitTrajReg` geladene Trajektorie durch. Darüber hinaus werden Plots und Animationen der Simulationsergebnisse erstellt. Bei den verwendeten Simulationsverfahren handelt es sich um explizite Einschrittverfahren. Der `ode4`-Solver (4. Ordnung) unterscheidet sich von `ode1` (1. Ordnung) im Wesentlichen durch die höhere Verfahrensordnung, wobei beide Verfahren mit konstanter Schrittweite operieren. Mit `ode45` wird zusätzlich ein Verfahren mit Schrittweitensteuerung eingesetzt. Hierbei handelt es sich wieder um ein Runge-Kutta-Verfahren, das jedoch gegenüber dem reinen RK4-Verfahren den Schrittfehler durch die Auswertung einer zusätzlichen Stützstelle schätzt und daraus für jeden Schritt eine eigene Schrittweite bestimmt.

Die Trajektorien werden für die Dauer des Prädiktionshorizonts und mit der gleichen Schrittweite, wie bei ihrer Berechnung verwendet wurde, simuliert. Ausnahme stellt `ode45` aufgrund der variablen Schrittweite da.

Die Funktion `TFRSim_SchlittenPendel_run` ruft das Simulinkmodell `TFR_SchlittenPendel_test` auf, das die Trajektorie direkt am Schlittenpendel-System simuliert, während `TFRSim_Gesamtmodell_run` das Simulinkmodell `TFR_Gesamtmodell_test` aufruft, das die Trajektorie am Gesamtsystem einschließlich des modellierten Gegeninduktionseffekts simuliert. Zum Vergleich werden die Trajektorien jeweils auch ohne Regler simuliert. Da die Simulationen störungsfrei sind, wird zunächst erwartet, dass auch eine Steuerung bereits gute Ergebnisse liefert.

Für die Berechnung des Reglers wurden im Voraus verschiedene QR-Matrizen als Ausgangskonfiguration getestet und sich schließlich für

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0,1$$

entschieden, die der Arbeit von Chang [17] entnommen werden.

4.3.2 Vergleichstrajektorie

Für die Versuche zur Stabilisierbarkeit und zum Einfluss der Systemparameter auf die Trajektorienberechnung (Abschnitt 4.4) wird eine Vergleichstrajektorie definiert.

Allgemein wird hierfür die klassische Aufschwingtrajektorie von AP1 nach AP4 gewählt. Es sind jedoch die in Abschnitt 4.1.2 vorgestellten Variationen zu beachten. Daher wird im Speziellen die Trajektorie

Traj14_dev0_-3.14_-3.14_x0max0.8 als Vergleichstrajektorie definiert. Anfangs- und Endzustand sind somit definiert als

$$\mathbf{x}_{\text{init}} = \begin{bmatrix} 0 \\ 0 \\ -\pi \\ 0 \\ -\pi \\ 0 \end{bmatrix}, \quad \mathbf{x}_{\text{end}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

wobei die Positionsbeschränkung $-0,8 \leq x_0 \leq 0,8$ als Nebenbedingung fest vorgegeben wird.

Eine maximale Stellkraft wird in Abhängigkeit der Versuche zur Vergleichstrajektorie ergänzt.

4.3.3 Ohne Gegeninduktion

Das System wird zunächst ohne die Gegeninduktion des Motormodells betrachtet, um im ersten Schritt an den Stand von Fauvé [8] anzuknüpfen. Entsprechend wird auch die Stellkraftbegrenzung auf die in Fauvé [8] verwendete Maximalkraft $F_{\text{max}} = 400 \text{ N}$ eingestellt.

Die Ergebnisse für $T = 0.01$ und $N = 350$ sind in Tabelle 4.2 zusammengefasst.

Tabelle 4.2: $T = 0.01$, $N = 350$

Simulation		Apprich		Ribeiro	
Solver	TFR	Euler	RK4	Euler	RK4
ode1	ohne mit	„leicht instabil“ stabil	instabil stabil	instabil instabil	instabil „leicht instabil“
ode4	ohne mit	instabil instabil	instabil instabil	instabil instabil	instabil stabil
ode45	ohne mit	instabil instabil	instabil instabil	instabil instabil	instabil „leicht instabil“

Es wird zwischen

- *stabil* - Aufschwung gelingt, AP4 kann bis zum Ende der Simulation gehalten werden
- „leicht instabil“ - Aufschwung gelingt weitgehend, AP4 wird nicht gehalten
- *instabil* - Aufschwung gelingt nicht

unterschieden. Zum besseren Verständnis soll die Zuordnung der dargestellten Ergebnisse an Hand repräsentativer Beispiele erläutert werden.

Zunächst wird der Eintrag links oben in Tabelle 4.2 betrachtet. Hierbei wird die Vergleichstrajektorie mit dem Apprich-Parametersatz und dem Euler-Verfahren berechnet und anschließend auch wieder mit dem Euler-Verfahren am Schlittendoppelpendel simuliert. Die in Abbildung 4.2 dargestellten Ergebnisse zeigen die Verläufe der Ausgänge und der Stellkraft für das zunächst unregelte System. Es ist zu

erkennen, dass der Aufschwung ohne Regelung gelingt, AP4 jedoch nicht gehalten wird. Das Ergebnis wird daher als *leicht instabil* beurteilt. In der anschließenden Simulation am geregelten System kann das Pendel schließlich bis zum Ende der Simulationszeit stabilisiert werden.

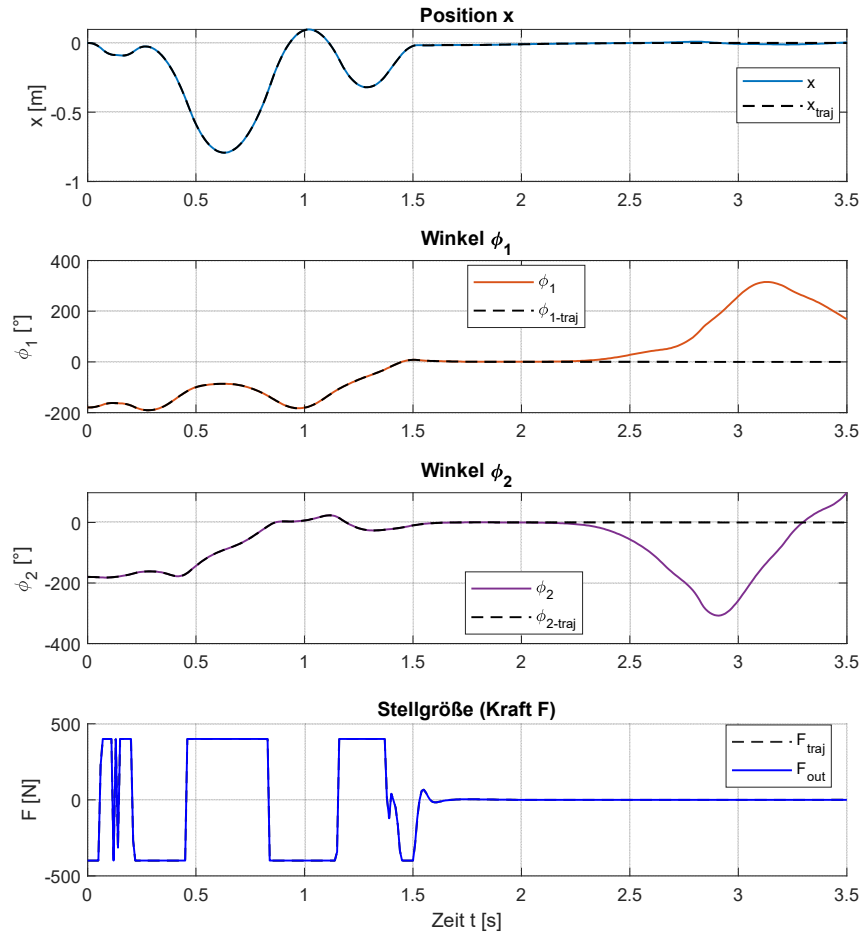


Abbildung 4.2: $T = 0,01$, $N = 350$, Apprigh-Parameter, Euler-MPC, ode1-Sim, ohne Regelung

Mit den Verfahren ode4 und ode45 am ungeregelten System zeigen die Verläufe demgegenüber hohe Abweichungen von der Trajektorie. Erwartungsgemäß lassen sie sich anschließend am geregelten System nicht stabilisieren. Die Verläufe werden daher als *instabil* bewertet.

Die mit dem Eulerverfahren berechnete Trajektorie lässt sich zwar stabilisieren, jedoch nur, wenn die bei der Trajektorienberechnung gemachten Schrittfehler durch das gleiche Verfahren in der Simulation näherungsweise reproduziert werden. Dass die Steuerung alleine nicht ausreicht, obwohl mit gleicher numerischer Fehlerfortpflanzung simuliert wird, lässt sich einerseits damit begründen, dass die Nebenbedingungen des Optimierungsverfahrens mit bestmöglicher, jedoch endlicher Genauigkeit eingehalten werden. Die beobachteten Genauigkeiten für die Nebenbedingungen liegen zwischen 10^{-13} und 10^{-34} für (lokal) optimale Lösungen. Andererseits ist insbesondere zu beachten, dass der zu erreichende Endwert durch die Optimierung lediglich angenähert wird. Die im Rahmen der Arbeit beobachteten absoluten Fehler einzelner Zustandsgrößen im letzten Prädiktionsschritt lagen bei den vom Optimierer ausgegebenen „optimalen“ Lösungen im Bereich von 10^{-5} und $2 \cdot 10^1$.

Als weiteres Beispiel wird die für den Ribeiro-Parametersatz und das RK4-Verfahren berechnete Trajektorie in der Simulation mit `ode4` betrachtet. Die Verläufe sind für das unregelte System in Abbildung 4.3 zu sehen.

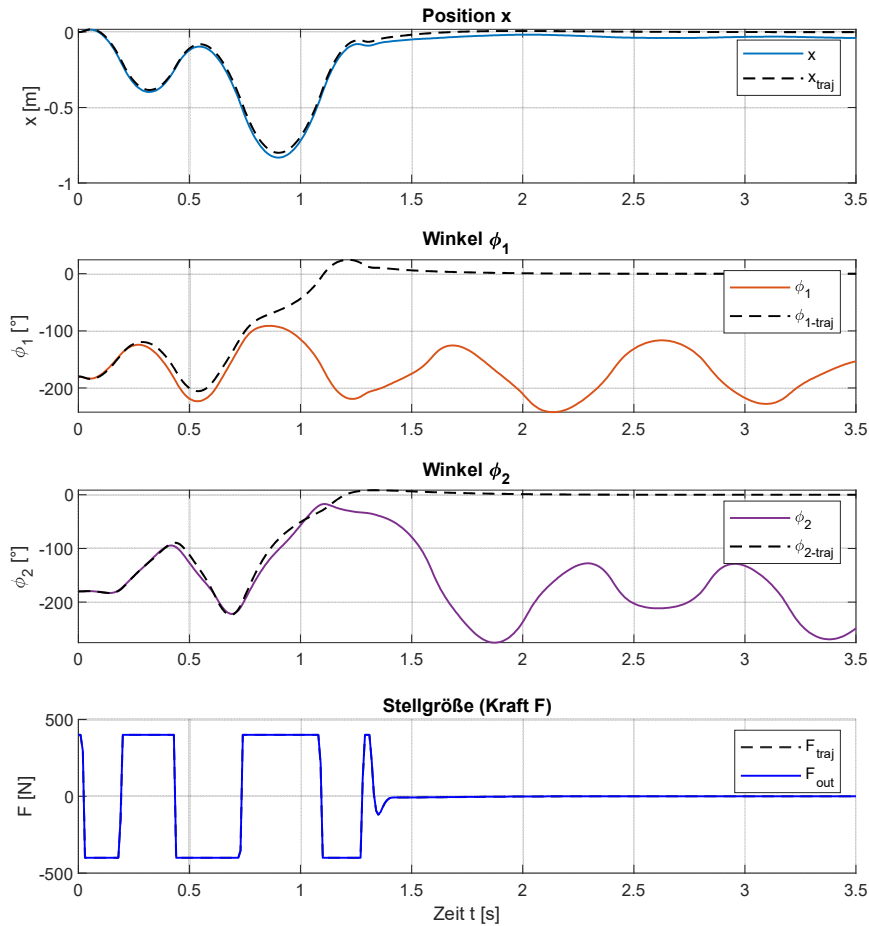


Abbildung 4.3: $T = 0,01$, $N = 350$, Ribeiro-Parameter, RK4-MPC, ode4-Sim, ohne Regelung

Es ist deutlich ersichtlich, dass der Aufschwung nicht gelingt. Das erste Pendel fällt bereits gleich zu Anfang des Aufschwungs zurück, sodass in der Folge beide Pendel unkontrolliert zu schwingen beginnen. Die Verläufe werden daher als *instabil* bewertet. In Abbildung 4.4 sind demgegenüber die Verläufe mit Trajektorienfolgeregelung zu sehen.

Das Doppelpendel kann demnach stabilisiert werden, jedoch ist in der Schlittenposition ein „Weglaufen“ zu beobachten. Dieses sorgt dafür, dass die vorgegebene Positionsbegrenzung weit überschritten wird. Es liegt daher nahe, dass durch eine höhere Bestrafung der Position mit Hilfe der QR-Parameter des Reglers eine Verringerung der maximalen Auslenkung zu erwarten ist. Eine Variation von Q_1 ergibt, dass mit einer Erhöhung von $Q_1 = 1$ auf wahlweise $Q_1 = 1,05$ oder $Q_1 = 4,5$ sich die maximale Positionsauslenkung von etwa $-3,2$ m auf $-2,5$ m senken lässt. Damit liegt die erreichte Auslenkung weiterhin weit außerhalb der zulässigen Begrenzung von $\pm 0,8$ m. Es zeigt sich außerdem, dass die Stabilisierung der betrachteten Trajektorie empfindlich auf Veränderung von Q_1 reagiert. Bereits bei $Q_1 = 1,07$ bzw. $Q_1 = 4,6$ wird das System instabil. Da allgemein jedoch eine Stabilisierung erreicht

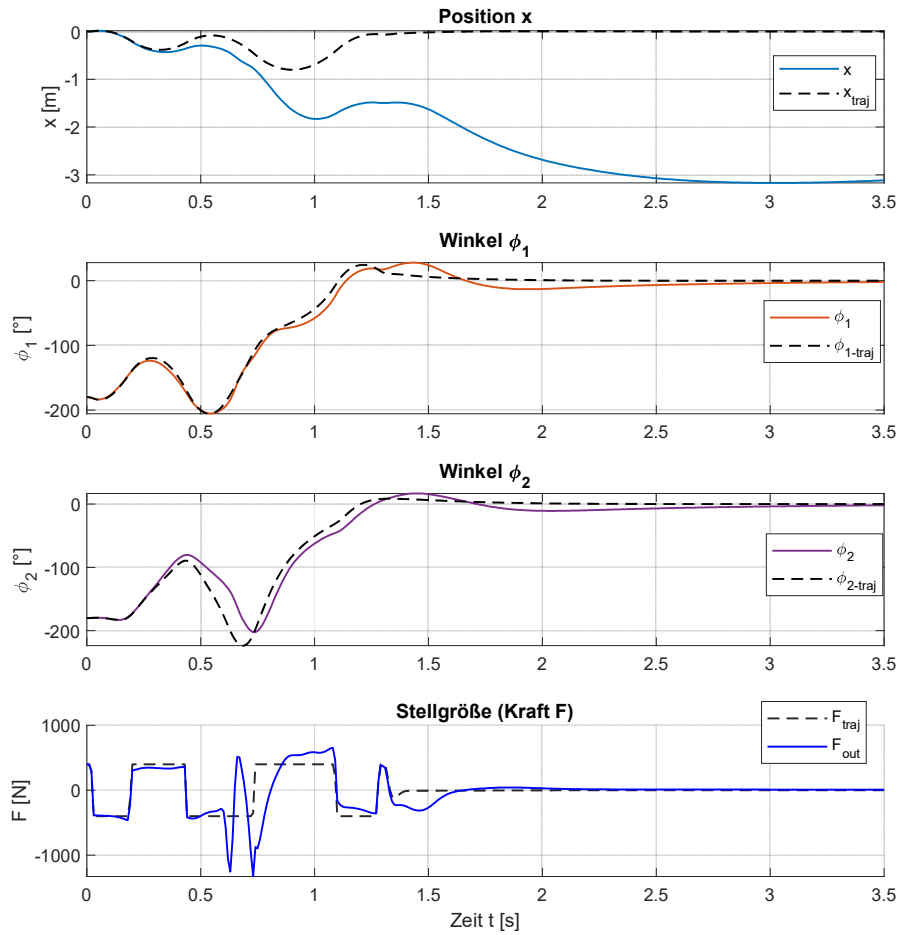


Abbildung 4.4: $T = 0,01$, $N = 350$, Ribeiro-Parameter, RK4-MPC, ode4-Sim, mit Regelung

wird, ist die Trajektorie für die Simulation ode4 am geregelten System als *stabil* zu bewerten.

Nach dem beschriebenen Vorgehen werden auch die weiteren Simulationsversuche ausgewertet. Die Ergebnisse in Tabelle 4.2 zeigen, dass sich drei der vier berechneten Trajektorien in zumindest einer der drei betrachteten Simulationen stabilisieren lässt. Die mit dem Euler-Verfahren und den Ribeiro-Parametern berechnete Trajektorie fällt dabei auf, da eine Stabilisierung generell in keiner der betrachteten Simulationen erreicht wird. Als Nächstes wird daher eine höhere Güte der Trajektorien gefordert, indem die Schrittweite für die Trajektorienberechnung halbiert wird bei entsprechender Erhöhung des Prädiktionshorizonts. Die Ergebnisse für $T = 0.005$ und $N = 500$ sind in Tabelle 4.3 dargestellt.

Tabelle 4.3: $T = 0.005$, $N = 500$

Simulation		Apprich		Ribeiro	
Solver	TFR	Euler	RK4	Euler	RK4
ode1	ohne mit	„leicht instabil“ stabil	„leicht instabil“ stabil	instabil instabil	Trajektorie
ode4	ohne mit	instabil instabil	„leicht instabil“ stabil	instabil instabil	nicht
ode45	ohne mit	instabil instabil	„leicht instabil“ stabil	instabil instabil	gefunden

Hierbei fällt auf, dass die Trajektorie, die mit Hilfe des Apprich-Parametersatzes und des RK4-Verfahrens berechnet wird, mit Hilfe des Reglers in allen drei Simulationen stabilisiert werden kann. Auch die Verläufe am ungeregelten System liefern bereits gute Ergebnisse. Die Stabilisierung durch den Regler erfolgt zudem innerhalb der vorgegebenen Positionsbegrenzung bei gleichzeitig geringer Abweichung von der Trajektorie.

4.3.4 Mit Gegeninduktion

Wird die Gegeninduktion des Motors berücksichtigt und am Gesamtmodell simuliert, fällt zunächst auf, dass mit $F_{\max} = 400 \text{ N}$ für den konstanten Bereich der Strombegrenzung keine der vier untersuchten Trajektorien gefunden wird. Durch eine Variation der Maximalkraftbegrenzung ergibt sich $F_{\max} = 410 \text{ N}$ als neue Grenze. Damit ist weiterhin eine Stellgrößenreserve von 11 N gegenüber der zuletzt am Versuchsstand eingestellten Maximalkraft von $F_{\max} = 421 \text{ N}$ gewährleistet.

Für $T = 0.01$ und $N = 350$ kann dennoch keine der Trajektorien gefunden werden. Die Ergebnisse für $T = 0.005$ und $N = 500$ sind in Tabelle 4.4 zusammengetragen. Die Trajektorie auf Basis der Apprich-Parameter und des RK4-Verfahrens bestätigt die Ergebnisse aus Abschnitt 4.3.3. Sie lässt sich unabhängig vom betrachteten Simulationsverfahren stabilisieren.

Tabelle 4.4: $T = 0.005$, $N = 500$

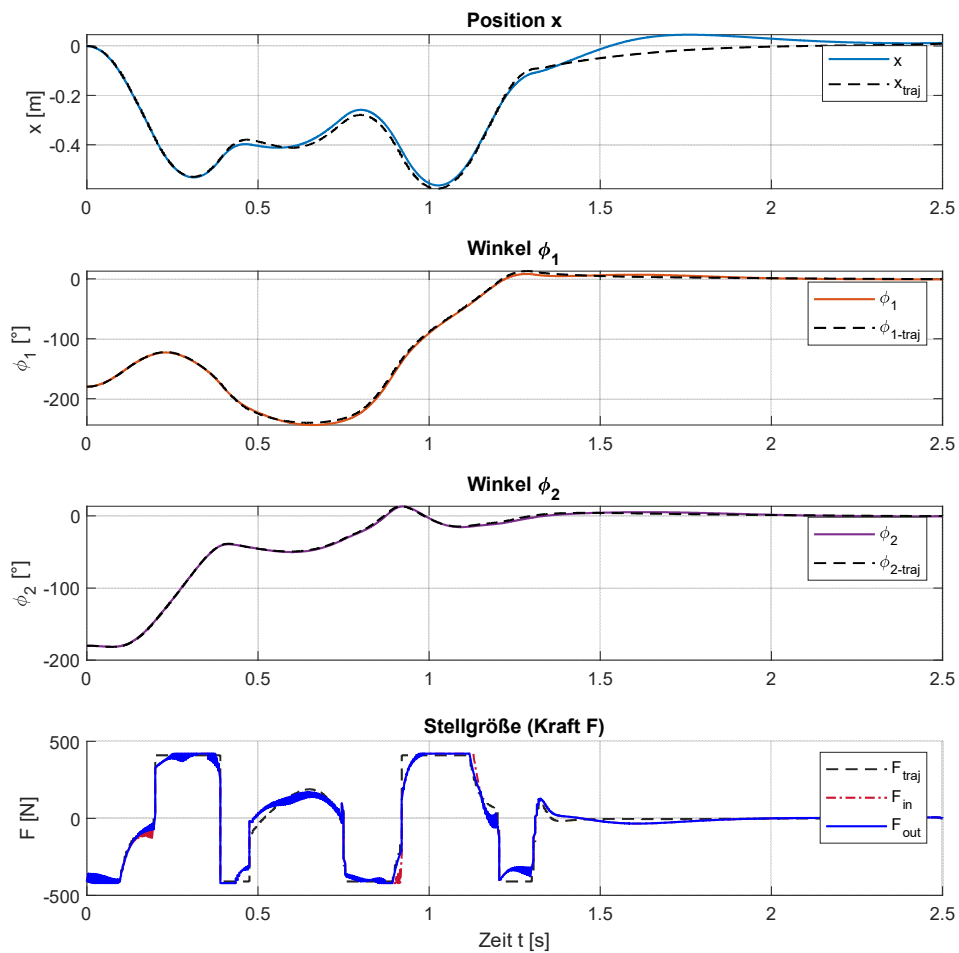
Simulation		Apprich		Ribeiro	
Solver	TFR	Euler	RK4	Euler	RK4
ode1	ohne mit	Trajektorie	instabil stabil	leicht instabil stabil	Trajektorie
ode4	ohne mit	nicht	instabil stabil	instabil instabil	nicht
ode45	ohne mit	gefunden	instabil stabil	instabil instabil	gefunden

Durch Variation der QR-Parameter können die Simulationsergebnisse zudem noch optimiert werden. Sehr gute Ergebnisse werden für die QR-Matrizen in Tabelle 4.5 erzielt.

Tabelle 4.5: Neue QR-Matrizen für Apprigh-RK4-Trajektorie

ode1	$\mathbf{Q} = \text{diag}(1, 1, 100, 1, 100, 1)$	$R = 0,01$
ode4, ode45	$\mathbf{Q} = \text{diag}(1000, 0.01, 100, 0.1, 100, 0.1)$	$R = 0,001$

Die Ausgangsverläufe für die Simulation mit ode45 und den neuen QR-Parametern sind in Abbildung 4.5 dargestellt. Im Vergleich zu den Plots in Abschnitt 4.3.3 wird im Stellgrößenverlauf zusätzlich zwischen F_{in} und F_{out} unterschieden. F_{in} bezeichnet den durch Regler und Trajektorie festgelegten Stellwert, der auf das Gesamtsystem gegeben wird. Auf Grund der Strombegrenzungskennlinie (vgl. Abbildung 2.3) kann am Ausgang des Motors jedoch ein abweichender Kraftverlauf entstehen, der mit F_{out} dargestellt wird. Da die Stromkennlinie in den Nebenbedingungen der Trajektorienberechnung berücksichtigt wird, entstehen Abweichungen zwischen F_{in} und F_{out} in erster Linie auf Grund des Reglers.

**Abbildung 4.5:** $T = 0,005$, $N = 500$, Apprigh-Parameter, RK4-MPC, ode45-Sim, mit Regelung

Fazit: Es lässt sich zeigen, dass die mit dem Verfahren der NMPC berechneten Trajektorien in der Simulation stabilisierbar sind. Zudem bestätigt sich die Vermutung, dass die Schrittweite und die Wahl des Integrationsverfahrens zur Berechnung der Trajektorien einen Einfluss auf die numerische Stabilität haben. Eine geringere Schrittweite und ein Integrationsverfahren mit geringerem Schrittfehler haben in

den betrachteten Versuchsdurchführungen zu einer höheren Stabilität geführt. Darüber hinaus fällt auf, dass die Apprigh-Parameter gegenüber den Ribeiro-Parametern ein günstigeres Verhalten aufweisen. Der Einfluss der Modellparameter wird im folgenden Kapitel daher näher untersucht.

4.4 Untersuchung des Einflusses der Modellparameter

Im Folgenden soll der Einfluss der Modellparameter auf die Trajektorienberechnung untersucht werden.

4.4.1 Vorgehen

Gegenstand der Untersuchung sind die Parameter Masse, Massenträgheitsmoment und Schwerpunkt jedes Pendelstabs. Durch Variation der Parameter sollen günstige Wertebereiche identifiziert und formuliert werden, in denen gültige Trajektorien gefunden werden können. Darüber hinaus werden die Variationsergebnisse in Bezug auf Häufung und Güte der gültigen Trajektorien analysiert.

Die Parameterwerte für die Parameteruntersuchung orientieren sich an den Werten der zum Versuchsstand aufgestellten Parametersätze aus Abschnitt 2.3.3. Die Schrittweiten der Parametervariationen werden möglichst gering gewählt, um die Aussagekraft der Ergebnisse durch eine hohe Auflösung sicherzustellen. Auf Grund der hohen Berechnungszeiten für die Trajektorien, sind beliebig kleine Schrittweiten jedoch nicht möglich, sodass ein sinnvoller Kompromiss zwischen Auflösung und Berechnungszeit gefunden werden muss. Das geplante Variationsvorhaben ist in Tabelle 4.6 aufgeführt. Hierbei wird die konstruktive Realisierbarkeit der Parameterwerte nicht berücksichtigt, da vorwiegend der Einfluss der Modellparameter auf die Berechnungsergebnisse untersucht wird.

Tabelle 4.6: Parametervariationen

Parameter	Startwert	Endwert	Schrittweite	Anzahl Trajektorien
m_1 [kg]	0	2	0,01	201
m_2 [kg]	0	2	0,01	201
J_1 [kgm ²]	0	0,02	0,0001	201
J_2 [kgm ²]	0	0,02	0,0001	201
s_1 [m]	0	0,29	0,001	291
s_2 [m]	0	0,338	0,001	339

Die zu untersuchenden Modellparameter werden einzeln gegenüber einem definierten Ausgangsparametersatz variiert, wobei alle weiteren Parameter konstant gehalten werden. Zur Durchführung der Versuche wird die Funktion `examParameters` implementiert. Ihr kann eine Versuchsreihe bestehend aus der Bezeichnung des zu untersuchenden Parameters und einem Vektor mit den Variationswerten übergeben werden. Über den Aufruf der in Abschnitt 4.1.1 beschriebenen Funktion `searchTrajectories` werden damit die benötigten Trajektorienberechnungen ausgeführt. Die Trajektorien werden anschließend im Ordner `ParameterExams` gespeichert. Um die Ergebnisse später sinnvoll identifizieren zu können, wird der in Abschnitt 4.1.1 definierten Namenskonvention eine Endung bestehend aus der Bezeichnung des untersuchten Parameters und dessen Wert angefügt.

Zur Bewertung der betrachteten Parametervariationen wird zunächst unterschieden, ob für die jeweilige Variation eine Trajektorie gefunden werden kann oder nicht. Darüber hinaus wird das Gütemaß J_{dev} eingeführt, um die Güte der gefundenen Lösungen zu quantifizieren. Der Ansatz für das Gütemaß basiert auf der Definition von Trajektorien durch ihre Randbedingungen. Da diese bei der Optimierung mit endlicher Genauigkeit erfüllt werden (vgl. Abschnitt 4.3.3), ist der resultierende Fehler ein Kriterium für die Güte der Trajektorien.

Das Gütemaß wird mit

$$J_{\text{dev}} = \|\Delta \mathbf{x}_{\text{init}}\| + \|\Delta \mathbf{x}_{\text{end}}\|$$

definiert. $\|\Delta \mathbf{x}_{\text{init}}\|$ und $\|\Delta \mathbf{x}_{\text{end}}\|$ sind hierbei als die euklidischen Normen der mit den Fehlerdifferenzen gefüllten Zustandsvektoren in Anfangs- und Endlage der Trajektorie zu verstehen.

$$\|\Delta \mathbf{x}_{\text{init}}\| = \sqrt{\sum_{i=1}^6 (x_{\text{Traj},i}(t_{\text{init}}) - x_{\text{init},i})^2}$$

mit $t_{\text{init}} = 0$ und

$$\|\Delta \mathbf{x}_{\text{end}}\| = \sqrt{\sum_{i=1}^6 (x_{\text{Traj},i}(t_{\text{end}}) - x_{\text{end},i})^2}$$

mit $t_{\text{end}} = (N + 1) \cdot T$

Wie in Abschnitt 4.3 erläutert, ist im Falle einer konvergierten Lösung nur der Endwertfehler relevant, da dieser indirekt durch die Zielfunktion und nicht durch die Nebenbedingungen für die Optimierung vorgegeben wird. Unter der Voraussetzung einer konvergierten Lösung gilt daher

$$J_{\text{dev}} = \|\Delta \mathbf{x}_{\text{end}}\| .$$

Basierend auf den Erkenntnissen des vorherigen Abschnitts wird weiterhin folgende Ausgangskonfiguration für die Berechnung der Trajektorien gewählt, wobei die Gegeninduktion des Motors berücksichtigt und die Coulomb-Reibung vernachlässigt wird:

- $T = 0,005$
- $N = 500$
- RK4-Integration
- Aprich-Parameter

Als Trajektorie wird die in Abschnitt 4.3.2 definierte Vergleichstrajektorie verwendet. Die Gewichtungsmatrizen für die Zielfunktion sind Abschnitt 4.1.2 zu entnehmen.

4.4.2 Durchführung und Ergebnisse

Die Auswertung erfolgt mit Hilfe der Funktion `plot_Jdev_params`. Mit dieser wird das Gütemaß berechnet und über dem untersuchten Parameter in einem Koordinatensystem aufgetragen. Hierbei werden auch die berechneten Werte des Apprich- und des Ribeiro-Parametersatzes nach Abschnitt 2.3.3 als Vergleichswerte durch die Funktion hinzugefügt. Es ist zu beachten, dass der eingefügte Ribeiro-Vergleichswert sich nur auf den untersuchten Parameter bezieht. Die übrigen Parameter werden weiterhin dem Apprich-Parametersatz entnommen. Liegen die Werte von Apprich und Ribeiro nicht auf einem durch Tabelle 4.6 definierten Variationswert, ist eine Erhöhung der Anzahl an berechneten Trajektorien zu berücksichtigen. Trajektorien ohne lokale Konvergenz zu einem Optimum werden als ungültig gewertet.

Werden zunächst die relativen Häufigkeiten gültiger Trajektorien h_g miteinander verglichen, zeigt sich, dass neben dem Massenträgheitsmoment von Stab 1 besonders die Variation der beiden Stabmassen einen vergleichsweise geringen Anteil an Trajektorien verzeichnet, bei denen der Optimierer lokal konvergiert bzw. eine gültige Lösung findet (Tabelle 4.7). Der Einfluss von m_1 und m_2 sowie J_1 auf die Trajektorienberechnung ist daher als hoch anzusehen.

Tabelle 4.7: Statistik gültiger Trajektorien

Parameter	Ungültig	Gültig	Gesamt	Gültigkeitsanteil h_g
m_1 [kg]	177	26	203	13%
m_2 [kg]	189	14	203	7%
J_1 [kgm ²]	175	28	203	14%
J_2 [kgm ²]	79	124	203	61%
s_1 [m]	231	62	293	21%
s_2 [m]	190	150	340	44%

Besonders auffällig ist in diesem Zusammenhang m_2 mit einem Anteil gültiger Trajektorien von nur $h_g = 7\%$. Abbildung 4.6 visualisiert die Ergebnisse der m_2 -Variation. Die Datenpunkte für Apprich und Ribeiro werden allgemein durch Quadrate markiert, wobei Dunkelblau für Apprich und Dunkelorange für Ribeiro steht. Da für die Ribeiro-Masse in diesem Fall keine gültige Trajektorie gefunden wird, erscheint der Wert als „x“-Markierung auf der Abszisse. Die übrigen ungültigen Trajektorien sind hingegen im Sinne der Übersichtlichkeit nicht explizit dargestellt.

In der Abbildung zu erkennen ist, dass 12 von 14 gefundenen Trajektorien im Bereich $0,09 \text{ kg} \leq m_2 \leq 0,39 \text{ kg}$ liegen. Zwei weitere Trajektorien werden gefunden, wenn der Wertebereich bis $1,43 \text{ kg}$ deutlich vergrößert wird. Eine Erweiterung bis 2 kg führt schließlich zu keiner weiteren Trajektorie mehr. Wird nur der Bereich $0,09 \text{ kg} \leq m_2 \leq 0,39 \text{ kg}$ betrachtet, steigt die Häufigkeit gültiger Trajektorien von $h_g = 7\%$ auf $h_g = 38\%$. Es treten jedoch weiterhin auch ungültige Lösungen auf. Zwischen den diskret untersuchten Parametern (vgl. Tabelle 4.6) existieren im Kontinuierlichen unendlich viele weitere, potentielle Parameterwerte, für die eine Trajektorie berechnet werden kann. Wird die relative Häufigkeit gültiger Trajektorien h_g als Konvergenz-Wahrscheinlichkeit $h_g \approx P_{\text{kon}}$ der Optimierung interpretiert, dann kann auch eine Aussage über die weiteren, potentiellen Trajektorien im Kontinuierlichen innerhalb des betrachteten Bereichs getroffen werden. Der Bereich $0,09 \text{ kg} \leq m_2 \leq 0,39 \text{ kg}$ signalisiert somit allgemein eine hohe Konvergenzwahrscheinlichkeit P_{kon} . Der markierte Apprich-Wert für m_2 befindet sich bereits nahe der Grenze dieses Bereichs, der Ribeiro-Wert sogar leicht darüber. Im Sinne einer hohen Konvergenzwahrscheinlichkeit kommt daher nur eine Verringerung von m_2 in Frage.

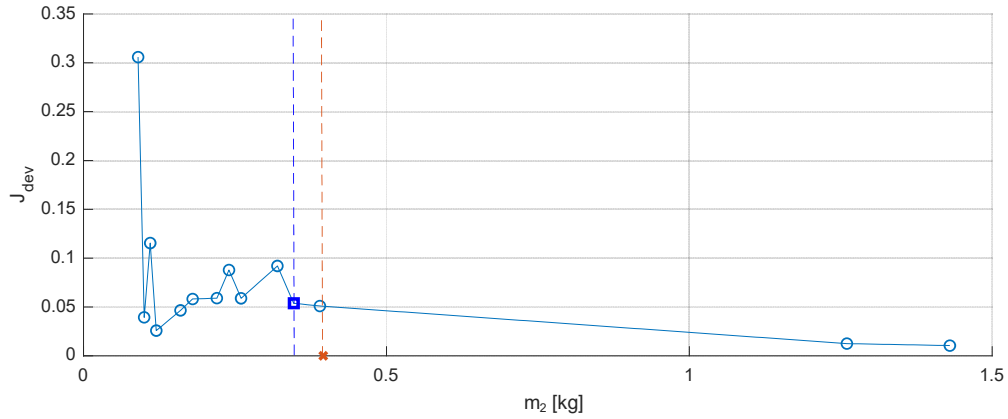


Abbildung 4.6: Variation m_2

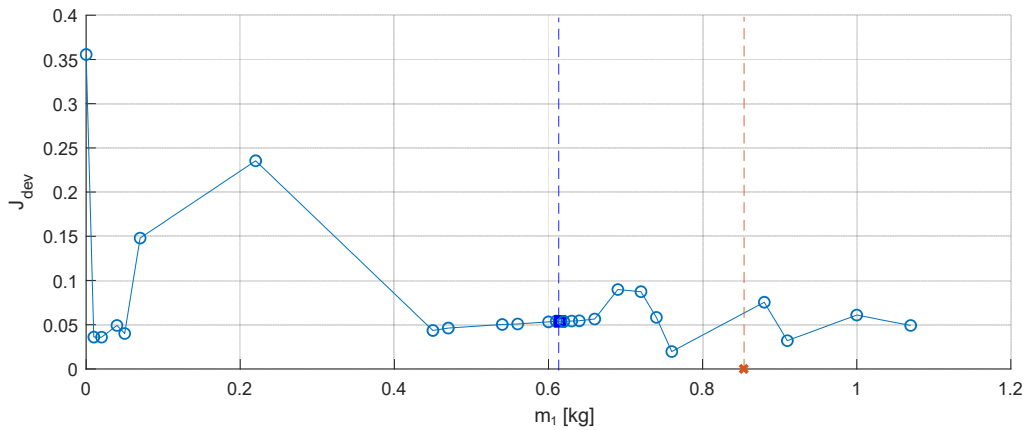


Abbildung 4.7: Variation m_1

Bei Betrachtung der Güte der gefundenen Trajektorien, fällt auf, dass für 12 von 14 Trajektorien $J_{\text{dev}} = \|\Delta \mathbf{x}_{\text{end}}\| < 0,1$ gilt. Besonders gut wird der Endwert für die beiden Trajektorien approximiert, die jedoch außerhalb des Bereichs von $P_{\text{kon}} \approx 38\%$ liegen und daher als „Ausreißer“ bezeichnet werden. Die ungünstigeren Werte mit $J_{\text{dev}} < 0,1$ liegen hingegen im unteren Grenzbereich. Die Erfahrung in der Simulation hat gezeigt, dass für $J_{\text{dev}} \gg 1$ ohne Weiteres keine QR-Matrizen für die Regelung gefunden werden, mit denen die Trajektorien stabilisiert werden können. Diese Trajektorien werden daher in Bezug auf Stabilisierbarkeit als ungünstig bis unbrauchbar eingeschätzt. Damit liegen die gefundenen Trajektorien mit $J_{\text{dev}} < 1$ im akzeptablen Bereich. Ein eindeutiges Muster bezüglich der Trajektoriengüte lässt sich in Abbildung 4.6 jedoch nicht erkennen.

Für m_2 wird schließlich der Bereich $0,09 \text{ kg} \leq m_2 \leq 0,39 \text{ kg}$ mit einer Konvergenzwahrscheinlichkeit von $P_{\text{kon}} \approx h_g = 38\%$ als Wertebereich gültiger Trajektorien formuliert, wobei die zwei identifizierten „Ausreißer“ vernachlässigt werden. Für den aktuelle Ribeiro-Parameter wird zudem empfohlen, diesen im Sinne des formulierten Wertebereichs zu verringern.

Für m_1 lässt sich ein ähnliches Verhalten beobachten. Auch hier zeichnet sich der Bereich gültiger Trajektorien durch eine gut erkennbare Obergrenze ab. Diese liegt mit $m_{1,\text{max}} = 1,07 \text{ kg}$ allerdings deutlich über der von m_2 . Auffällig ist darüber hinaus eine Häufung gültiger Trajektorien mit hoher

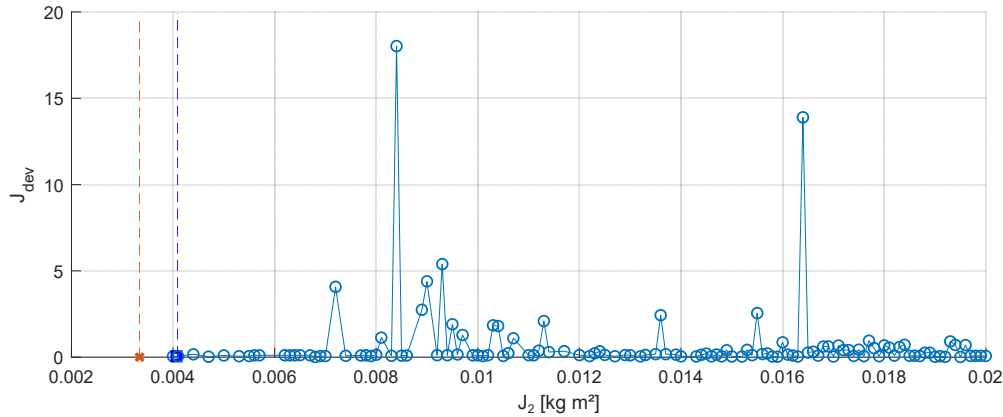


Abbildung 4.8: Variation J_2

Güte im stark eingeschränkten Bereich $0,6 \text{ kg} \leq m_1 \leq 0,66 \text{ kg}$ mit $h_g = 88\%$, wo auch der Apprigh-Wert einzuordnen ist. Bei einer Erweiterung des Bereichs auf $0,45 \text{ kg} \leq m_1 \leq 1,07 \text{ kg}$ wird noch ein Anteil von $h_g = 29\%$ erreicht. Eine ähnliche Häufung tritt im Bereich $0 \text{ kg} \leq m_1 \leq 0,07 \text{ kg}$ mit $h_g = 75\%$ auf, wobei die Bereichsgröße stark eingeschränkt ist. Allgemein liegen wie bei m_2 auch für m_1 die Trajektorien mit einer geringeren Güte von $J_{\text{dev}} < 0,1$ im unteren Bereich. Besonders ungünstige Trajektorien mit $J_{\text{dev}} \gg 1$ treten allerdings erneut nicht auf. Auch wird wieder für den Ribeiro-Parameter keine Lösung gefunden, wobei dieser nun in einem Bereich liegt, in dem prinzipiell Lösungen gefunden werden.

Allgemein wird als Wertebereich $0 \text{ kg} \leq m_1 \leq 1,07 \text{ kg}$ formuliert, wobei $P_{\text{kon}} \approx h_g = 26\%$ gilt.

Anders als bei den Massen unterscheiden sich die Massenträgheitsmomente in ihrem Verhalten recht deutlich. Während für J_1 bezogen auf die Anzahl untersuchter Werte nur ein Anteil gültiger Trajektorien von 14% vorliegt, werden für J_2 mit einem Anteil von 61% überwiegend gültige Trajektorien gefunden. Dies deutet auf einen geringen Einfluss von J_2 auf die Trajektorienberechnung hin. Die in Abbildung 4.8 dargestellten Ergebnisse zeigen, dass beinahe über den gesamten Variationsraum von J_2 Trajektorien gefunden werden. Lediglich im untersten Bereich $0 \text{ kgm}^2 \leq J_2 < 0,004 \text{ kgm}^2$ können keine Trajektorien gefunden werden. In diesem Bereich ist der Ribeiro-Parameter angesiedelt, für den entsprechend keine Trajektorie gefunden wird. Der Apprigh-Parameter befindet sich mit $0,00407 \text{ kgm}^2$ unmittelbar im Grenzbereich. Durch eine Untergrenze der Wertebereichs mit $0,004 \text{ kgm}^2 \leq J_2 \leq 0,02 \text{ kgm}^2$ liegt der Anteil gültiger Trajektorien bei 77%. Wird die Untergrenze auf $J_{2\text{min}} = 0,0152 \text{ kgm}^2$ erhöht, ergeben sich mit $h_g = 100\%$ ausschließlich gültige Trajektorien über den gesamten Bereich. Die Trajektoriengüte unterliegt im Bereich von $h_g = 100\%$ jedoch starken Schwankungen

Anders als bei den Stabmassen treten für J_2 erstmals besonders hohe Werte der Gütefunktion von $J_{\text{dev}} \gg 1$. Die für $J_2 = 0,0084 \text{ kgm}^2$ gefundene Trajektorie erreicht einen Wert von $J_{\text{dev}} = 18,02$ und ist damit als unbrauchbar anzusehen. Im Bereich $0,0072 \text{ kgm}^2 \leq J_2 \leq 0,0113 \text{ kgm}^2$ tritt eine leichte Häufung von Trajektorien mit hohem J_{dev} auf. Da es sich dennoch um lokale Minima und somit aus Sicht des Optimierers um gültige Trajektorien handelt, entsteht der hohe Wert weiterhin alleine aufgrund der hohen Ungenauigkeit der Approximation der Endlage. Für die weitere Verwendung berechneter Trajektorien ist daher dringend die Approximationsgüte zu überprüfen. Auch im beschriebenen Bereich von $h_g = 100\%$ treten vereinzelt gültige Trajektorien sehr geringer Güte auf.

Als Wertebereich gültiger Lösungen wird der Bereich $0,004 \text{ kgm}^2 \leq J_2 \leq 0,02 \text{ kgm}^2$ mit $P_{\text{kon}} \approx h_g = 77\%$ formuliert. Es wird zudem eine Erhöhung des aktuellen Ribeiro-Werts für J_2 empfohlen, sodass sich

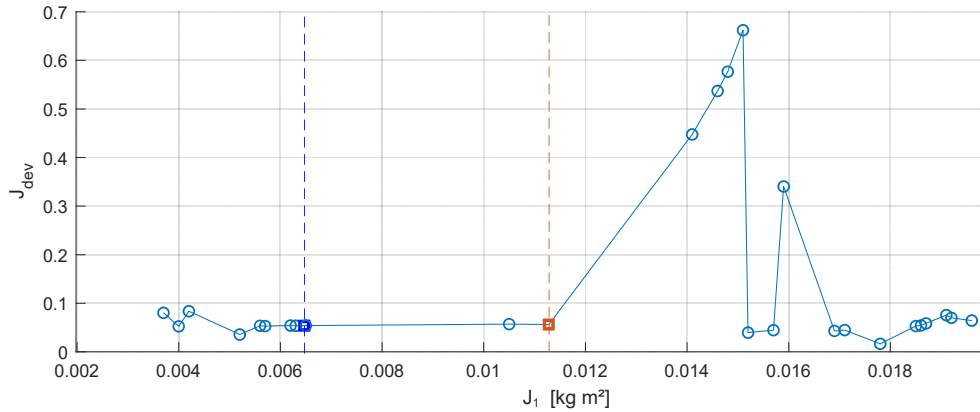


Abbildung 4.9: Variation J_1

dieser zukünftig innerhalb des empfohlenen Wertebereichs befindet.

Für J_1 werden mit insgesamt $h_g = 14\%$ nur wenige gültige Trajektorien gefunden. Diese sind über den größten Teil des untersuchten Wertebereichs verteilt. Im Gegensatz zu den anderen Parametervariation wird hierbei auch eine Trajektorie für den Ribeiro-Wert gefunden. Dies lässt sich jedoch nicht mit einem Häufungsbereich weiterer gültiger Trajektorien begründen, wie Abbildung 4.9 entnommen werden kann. Häufungen gültiger Trajektorien treten in den gegensätzlichen Bereichen $0,0037 \text{ kgm}^2 \leq J_1 \leq 0,0065 \text{ kgm}^2$ mit $h_g = 33\%$ und $0,0141 \text{ kgm}^2 \leq J_1 \leq 0,0196 \text{ kgm}^2$ mit $h_g = 29\%$ auf. Es können sich daher sowohl niedrige als hohe Werte für J_1 positiv auf die Konvergenz der Trajektorienberechnung auswirken. Der mittlere Bereich hingegen ist hingegen als ungünstig anzusehen, wobei dennoch gültige Lösungen gefunden werden können. Für sehr niedrige Werte existieren wie bei J_2 keine gültigen Trajektorien mehr, weshalb wieder eine Untergrenze $J_{2\min} = 0,0037 \text{ kgm}^2$ eingeführt wird. Trajektorien mit $J_{\text{dev}} > 1$ kommen anders als bei J_2 nicht vor. Ein erhöhtes Vorkommen von Trajektorien geringerer Güte mit $J_{\text{dev}} > 0,1$ wird im Bereich $0,0141 \text{ kgm}^2 \leq J_1 \leq 0,0159 \text{ kgm}^2$ beobachtet. Da dieser Bereich wieder einen Grenzbereich einer Häufung gültiger Trajektorien markiert wie bei m_1 und m_2 , könnte dies als Hinweis auf ein Verhaltensmuster der Güte verstanden werden. Die Variationen von J_2 , s_1 und s_2 bestätigen diese Interpretation jedoch nicht.

Mit der definierten Untergrenze wird ein Wertebereich für gültige Trajektorien von $0,0037 \text{ kgm}^2 \leq J_2 \leq 0,0196 \text{ kgm}^2$ mit $P_{\text{kon}} \approx h_g = 17\%$ formuliert.

Aus der Variation der Schwerpunkte s_1 und s_2 wird kein auffälliges Verhalten identifiziert. Für beide Schwerpunkte können näherungsweise über die gesamte jeweilige Stablänge gültige Lösungen gefunden werden, sodass eine Eingrenzung des Wertebereichs nicht notwendig erscheint. Der Anteil gültiger Trajektorien ist für s_1 jedoch deutlich geringer als für s_2 . Die Trajektorienberechnung reagiert damit prinzipiell empfindlicher auf eine Veränderung von s_1 . Eine Häufung gültiger Trajektorien tritt wieder im Bereich des Apprigh-Wertes auf. Obwohl der Ribeiro-Wert beider Schwerpunkte jeweils in einem solchen Häufungsbereich liegt, wird jeweils keine Trajektorie gefunden. Für die scheinbar ungünstigen Ribeiro-Schwerpunkte ist daher in diesem Fall keine explizite Ursache erkennbar.

Die formulierten Wertebereiche gültiger Trajektorien sind in Tabelle 4.8 noch einmal abschließend übersichtlich dargestellt.

Es ist in Bezug auf die Ergebnisse zu berücksichtigen, dass die betrachteten Systemparameter jeweils

Tabelle 4.8: Wertebereiche gültiger Trajektorien

Parameter	Einfluss	min	max	P_{kon}
m_1 [kg]	hoch	0	1,07	26%
m_2 [kg]	hoch	0,09	0,39	38%
J_1 [kgm ²]	hoch	0,0037	0,0196	17%
J_2 [kgm ²]	gering	0,004	0,02	77%
s_1 [m]	mittel	0	0,29	21%
s_2 [m]	gering	0	0,338	44%

für sich alleine variiert wurden. Eine Untersuchung bei Variation der Parameter in Abhängigkeit zu einander kann daher von den vorliegenden Ergebnissen abweichen. Ebenfalls sind die vielfältigen Konfigurationsmöglichkeiten der NMPC in Bezug auf die Ergebnisse zu problematisieren. Es ist entsprechend anzunehmen, dass formulierten Wertebereiche bei einer anderen Konfiguration der NMPC mit beispielsweise Variation von Schrittweite, Prädiktionshorizont, Integrator, Gewichtungsmatrizen der Gütefunktion oder anderem Optimierer variieren. Die formulierten Bereichsempfehlungen sind daher nur für die im Rahmen dieser Arbeit definierten Voraussetzungen gültig und in Bezug auf Allgemeingültigkeit in erster Linie als Orientierungshilfen, jedoch nicht als harte Grenzen zu verstehen.

Die Gültigkeit der ermittelten Bereichsgrenzen muss außerdem für weitere Vergleichstrajektorien überprüft werden und gegebenenfalls trajektorienspezifische Grenzen formuliert werden.

Die starke Begrenzung der Stabmassen lässt als Ursache eine verstärkte Abhängigkeit von der Stellgrößenbegrenzung vermuten. Die Stellgrößenreserve bzw. die der NMPC als Nebenbedingung übergebene Kraftbegrenzung birgt somit ein weiteres Variationspotential, wodurch besonders eine Verschiebung der Bereichsobergrenzen zu erwarten ist.

Die Coulomb-Reibung wurde im Rahmen der Untersuchung vernachlässigt. Erste Tests weisen auf einen erkennbar negativen Einfluss auf die Trajektorienberechnung hin, was auf die hohe Nichtlinearität zurückgeführt wird. Wie in Abschnitt 4.3.4 werden hierzu verschiedene NMPC-Konfigurationen getestet, wobei lediglich für die Apprigh-Parameter inklusive Coulomb-Reibung am Schlitten, exklusive Gelenkreibung, eine akzeptable Trajektorie gefunden wird, ebenfalls wieder für die bekannte NMPC-Konfiguration, die auch für die anderen Parametertests verwendet wurde. Es wird daher für zukünftige Arbeiten empfohlen, die für diese Arbeit gewählte Konfiguration weiterzuverwenden, wenn eine Variation der Reibwerte angestrebt wird.

5 Fazit und Ausblick

5.1 Zusammenfassung und Fazit

Diese Arbeit beschäftigt sich mit der Steuerung und Regelung eines Doppelpendels, das an einem Schlitten angebracht ist. Das System wurde in dieser Arbeit rein simulativ betrachtet, wobei die Anwendung am realen Versuchsstand stets im Hintergrund behalten wurde. Es wurde ein umfassendes Simulationsmodell in MATLAB/SIMULINK aufgebaut, welches in zukünftigen Arbeiten weiterverwendet werden kann.

Im Vergleich zu den bisherigen Arbeiten wurden bei der Modellierung Nichtlinearitäten wie die Coulomb-Reibung und die Gegeninduktion berücksichtigt. Die Herleitung der Bewegungsgleichungen des Schlitten-Pendel-Systems wurde symbolisch umgesetzt, wodurch das System leicht erweitert werden kann. Hierdurch konnten Fehler in den Systemgleichungen, die bislang in den vergangenen Arbeiten verwendet worden waren, aufgedeckt werden. Bei allen im Rahmen dieser Arbeit realisierten Implementierungen wurde auf eine sinnvolle Struktur und einen modularen Aufbau geachtet. Der zugehörige MATLAB-Code wurde funktional aufgebaut, wodurch unterschiedliche Parameter einfach getestet und verglichen werden können.

Die Regelung wurde an den vier Arbeitspunkten betrachtet und jeweils deren Regelbarkeit untersucht. Dazu wurde zunächst ein Regelungsmodell aufgebaut, das einen Arbeitspunkt-abhängigen Zustandsregler und eine Vorsteuerungseinheit enthält. Um Testreihen effektiv auswerten zu können, wurden die Anfangswertest-Funktionen programmiert, welche das Simulationsmodell automatisch ausführen und auf denen weitere Tests aufbauen können.

Nach einzelnen Weiterentwicklungen der Vorsteuerung und des Beobachters wurde eine weitere Erhöhung der Regelgüte durch Variation der Reglerparameter erreicht. Dazu wurden die Güteparameter des *Riccati*-Reglers schrittweise variiert und deren Einfluss mithilfe verschiedener Diagramme analysiert. Es konnten dadurch für alle Arbeitspunkte Parameter ermittelt werden, die das System gegenüber den vorherigen Werten besser stabilisieren können.

In einem weiteren Schritt wurde der Einfluss der Parameter des Doppelpendels auf die Arbeitspunktregelung untersucht. Die Untersuchungen zeigen, dass Trägheits- und Geometrieparameter einen erheblichen Einfluss auf die Stabilisierbarkeit des Systems haben. Teilweise hat sich gezeigt, dass manche Parametervariationen die Regelbarkeit deutlich erhöhen können. Die Ergebnisse sind jedoch unter dem Vorbehalt zu betrachten, dass die Systeme mit geänderten Parametern durch eine zusätzliche Optimierung der Reglerparameter unter Umständen ein besseres Verhalten aufweisen können.

Außerdem zeigt sich, dass die erhöhte Coulomb-Reibung des ersten Pendelgelenks einen erkennbar negativen Einfluss hat. Dies betrifft zum einen die Stabilisierbarkeit, aber auch das stationäre Verhalten,

da sich dadurch ein nicht ausregelbarer Grenzyklus einstellt. Insbesondere in Kombination mit dem Beobachter ist die Regelbarkeit wenig robust.

Weiterhin wurden eine NMPC für die Trajektorienberechnung sowie eine Trajektorienfolgeregelung realisiert. Anknüpfend an die Ergebnisse der Vorgängerarbeit Fauvé [8] konnte eine günstige Konfiguration der NMPC sowie eine geeignete Auslegung der Trajektorienfolgeregelung ermittelt werden, für die sich die betrachteten Aufschwungtrajektorien sowohl am Schlittenpendelmodell als auch am Gesamtmodell unter Berücksichtigung der Gegeninduktion des Motors stabilisieren lassen.

Die Untersuchung des Einflusses der Systemparameter auf die Trajektorienberechnung zeigt, dass besonders die Pendelmassen eine erkennbare Eingrenzung des Bereichs, in dem Trajektorien gefunden werden, aufweisen. Neben der Analyse der Verteilung gültiger und ungültiger Trajektorien sowie der Trajektoriengüte, wurden daher Begrenzungen der Wertebereiche für die untersuchten Parameter formuliert, die zur Orientierung für zukünftige konstruktive Veränderungen am Versuchsstand dienen können. Das ungünstige Verhalten der aktuellen Systemparameter in Bezug auf die Trajektorienberechnung wird durch die Untersuchung bestätigt, da im Vergleich zu den vorherigen Parametern nur wenige Trajektorien gefunden werden. Eine Begründung durch die ermittelten Bereichsgrenzen ist jedoch nur teilweise möglich, da oft auch dann keine Trajektorie gefunden wird, wenn sich der Parameter des aktuellen Versuchsstands im empfohlenen Wertebereich befindet. Einen erkennbaren Veränderungsbedarf signalisieren die Untersuchungen in erster Linie für das Trägheitsmoment von Stab 2. Insgesamt sind die Ergebnisse jedoch aufgrund der vielfältigen Einflussgrößen auf die Optimierungslandschaft der NMPC zunächst nur für die im Rahmen dieser Arbeit festgelegten Voraussetzungen gültig und in Bezug auf Allgemeingültigkeit zu problematisieren.

5.2 Ausblick

Die in dieser Arbeit optimierte Regelung und Steuerung kann am realen Versuchsstand erprobt werden. Möglicherweise führen die extra für den Beobachter, welcher auch am Versuchsstand eingesetzt wird, optimierten Reglerparameter wieder zu einer gut stabilisierenden Arbeitspunkt-Regelung. Dabei ist zu beachten, dass für die Optimierung die Stabilität der Winkelausregelung im Vordergrund stand. Für andere Szenarien wie horizontale Fahrten, müssen die Reglerparameter gegebenenfalls nach anderen Gesichtspunkten optimiert werden.

Um die Arbeitspunktregelung weiter zu verbessern, könnten die bisherigen Regelungskonzepte überdacht und erweitert werden. Aktuell basiert die Arbeitspunkt-Regelung auf einer Linearisierung, sodass ein linearer Zustandsregler und Beobachter verwendet werden kann. Da die angesprochene Coulomb-Reibung das System jedoch stark nichtlinear macht, sollte die Verwendung nichtlinearer Regelungen geprüft werden. In einem ersten Schritt sollte der Beobachter um die nichtlinearen Effekte ergänzt werden, da sich ansonsten große Schätzfehler ergeben, die das Regelverhalten verschlechtern. Alternativ kann die angesprochene Variante *Differenzieren* als Zustandsermittlung möglicherweise bessere Ergebnisse erzielen, da hier keine Arbeitspunktabhängigkeit oder weitere Modellungenauigkeiten beachtet werden müssen. Da selbst bei der Variante *Zustandsmessung* Schwierigkeiten wie der Grenzyklus auftreten, sollten auch beim Zustandsregler nichtlineare Regelungskonzepte in Erwägung gezogen werden. Im Gegensatz zum Schlitten lässt sich die Coulomb-Reibung in den Gelenken nämlich nicht über eine Vorsteuerung kompensieren. Eventuell lässt sich der Grenzyklus aber auch durch eine weitere Optimierung der Reglerparameter verringern, wobei für eine Einflussanalyse auf dem erstellten Code aufgebaut werden kann.

Des Weiteren kann die Regelbarkeit durch Änderungen am System selbst beeinflusst werden. Mithilfe der Systemparametertests kann der Einfluss aller Parameter systematisch untersucht werden. Für eine Entscheidungsfindung und weitere Test- und Analysemöglichkeiten stellen die Ergebnisse dieser Arbeit eine Grundlage dar. Eine direkte Aussage zur Modifizierung des aktuellen Versuchsstands durch Anbringen zusätzlicher Gewichte am Pendel oder durch Neukonstruktion ist problematisch, da die Konstruktionsparameter teilweise direkt voneinander abhängen und erste Parametertests gegensinnige Tendenzen zeigten, die sich zudem je nach Arbeitspunkt unterscheiden. Es muss daher stets ein Kompromiss getroffen werden. Für weiterführende Analysen und Aussagen sind eine genauere Modellierung der Massenverteilung sowie weitere Tests und Auswertungen nötig.

Die Ergebnisse der Parameteruntersuchungen, die für die Trajektorienberechnung mittels NMPC durchgeführt wurden, gelten für die im Rahmen dieser Arbeit definierten Voraussetzungen. Als nächster Schritt steht somit die Überprüfung der Gültigkeit bei einer Änderung dieser Voraussetzungen durch beispielsweise die Variation der Vergleichstrajektorie, der Kraftbeschränkung und den Konfigurationsmöglichkeiten der NMPC aus. Die ermittelten Bereichsverfehlungen der aktuellen Systemparameter sind hierbei besonders zu beobachten und gegebenenfalls nach ausreichender Prüfung am Versuchsstand zu korrigieren. Es ist hierbei auf mögliche Kompromisse bezüglich der in dieser Arbeit untersuchten Einflüsse auf die Arbeitspunktregelung zu achten.

Erste Tests in Bezug auf den Einfluss der Coulomb-Reibung auf die Trajektorienberechnung weisen auf einen erkennbar negativen Einfluss hin. Eine Variation der Reibwerte ist daher für zukünftige Arbeiten denkbar.

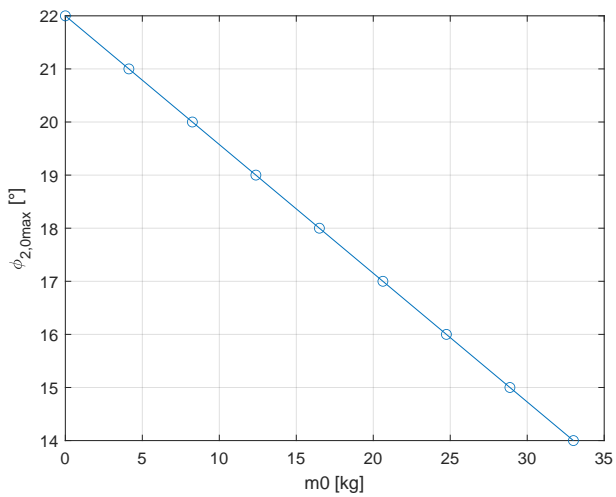
Da sich die im Rahmen dieser Arbeit durchgeführten Untersuchungen zur Stabilisierbarkeit der berechneten Trajektorien auf die Tests zur Wahl einer günstigen Ausgangskonfiguration der NMPC für die anschließenden Parametertests beschränken, bleibt eine detaillierte Untersuchung der Stabilisierbarkeit in Abhängigkeit der Trajektoriengüte offen.

Um die Qualität der Trajektorien zu steigern, wurde ein im Vergleich zur Vorgängerarbeit aufwendigeres Integrationsverfahren mit kleinerem Schrittfehler für die NMPC implementiert. Zukünftig wäre an dieser Stelle eine weitere Untersuchung denkbar, um zu entscheiden, welche Integrationsverfahren sich für die Berechnung von Trajektorien mit Hilfe des NMPC-Verfahrens prinzipiell besonders eignen.

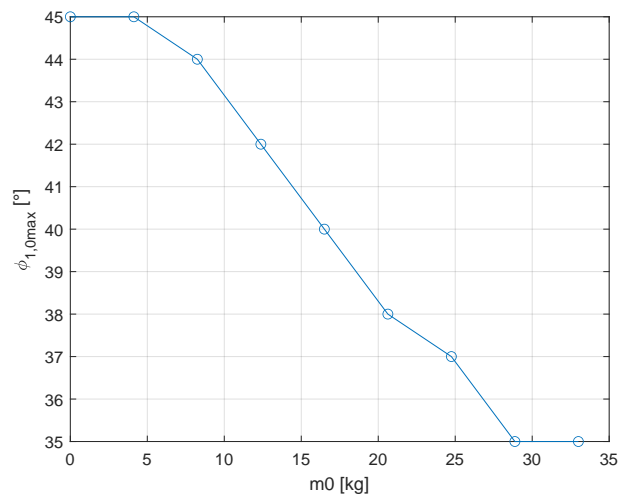
Der Einfluss der Systemparameter des Motors wurde in dieser Arbeit nicht betrachtet. Ebenso könnten die Parametervariationen auch für die Motorparameter durchgeführt werden. Der Code müsste dafür nur geringfügig erweitert werden. Somit könnte der Einfluss des Motors untersucht und unter Umständen bei zukünftigen Änderungen am System berücksichtigt werden.

Außerdem besteht die Möglichkeit, die Robustheit des Systems zu testen, indem Unsicherheiten von Parametern modelliert werden. Dazu könnten die Parameter des Modells, an dem die Reglerauslegung stattfindet, im Vergleich zu den Parametern, mit denen das Modell simuliert wird, mit Abweichungen versehen werden.

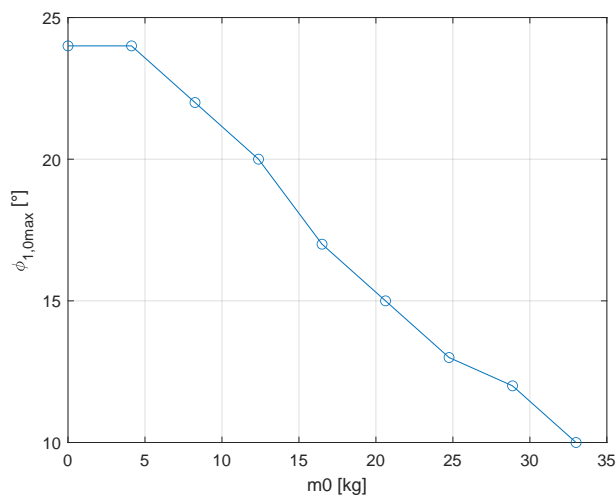
A AP-Regelung Systemparametertertests



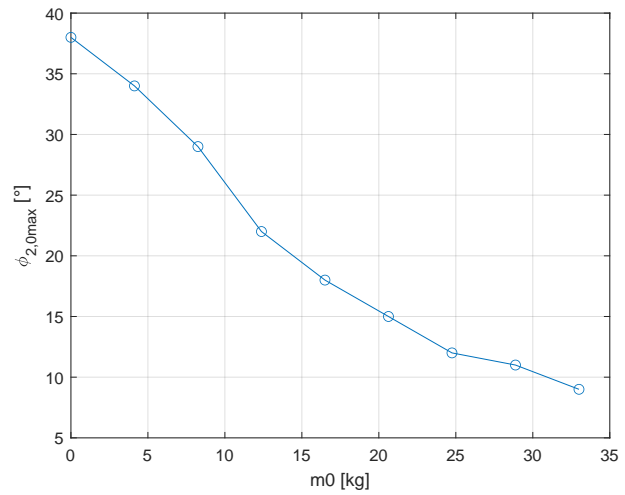
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)

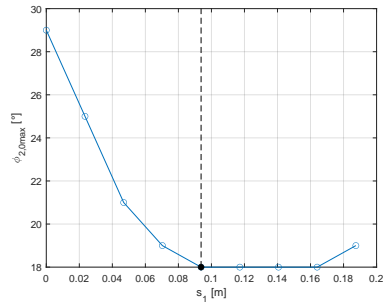


(c) AP4 (Auslenkung φ_1)

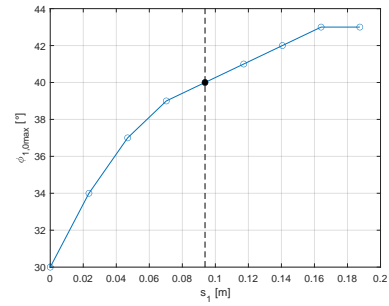


(d) AP4 (Auslenkung φ_2)

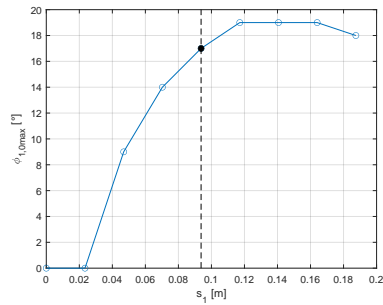
Abbildung A.1: Maximale Startwerte – Variation m_0



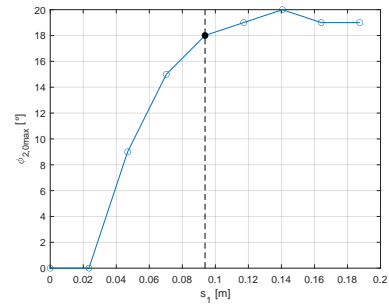
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)

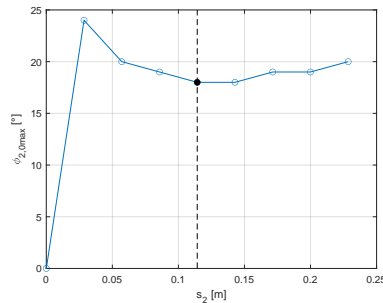


(c) AP4 (Auslenkung φ_1)

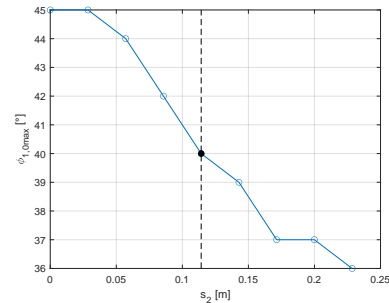


(d) AP4 (Auslenkung φ_2)

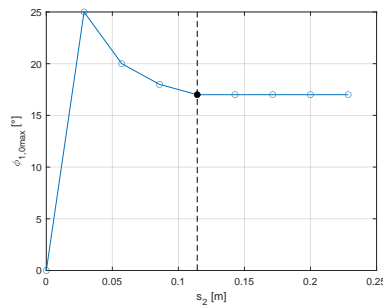
Abbildung A.2: Maximale Startwerte – Variation s_1



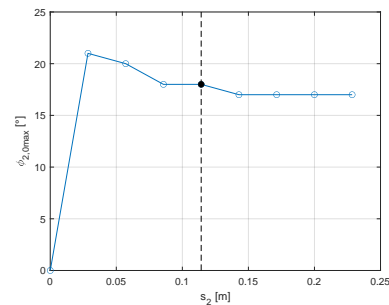
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)

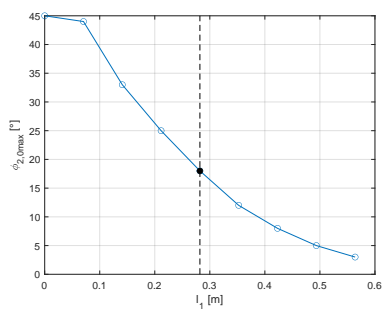


(c) AP4 (Auslenkung φ_1)

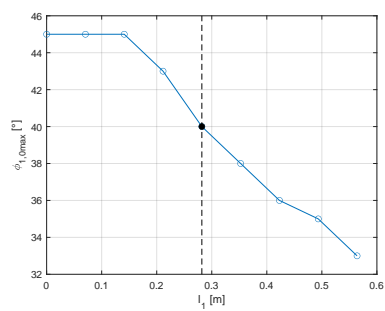


(d) AP4 (Auslenkung φ_2)

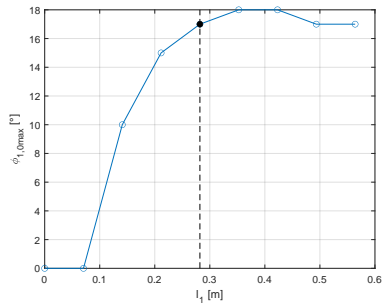
Abbildung A.3: Maximale Startwerte – Variation s_2



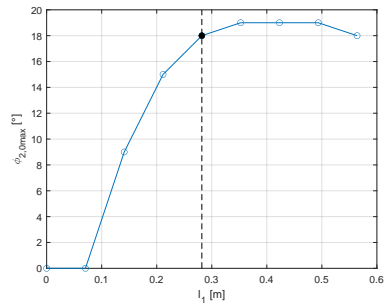
(a) AP2 (Auslenkung φ_2)



(b) AP3 (Auslenkung φ_1)



(c) AP4 (Auslenkung φ_1)



(d) AP4 (Auslenkung φ_2)

Abbildung A.4: Maximale Startwerte – Variation l_1

Literatur

- [1] Jürgen Adamy. *Systemdynamik und Regelungstechnik II*. Shaker, 2019.
- [2] Joel A E Andersson u. a. „CasADi – A software framework for nonlinear optimization and optimal control“. In: *Mathematical Programming Computation* 11.1 (2019), S. 1–36. DOI: 10.1007/s12532-018-0139-4.
- [3] Stefanie Apprich. *Konstruktion und Regelung eines inversen Doppelpendels*. Studienarbeit. Technische Universität Darmstadt, 2009.
- [4] Andreas Binder. *Elektrische Maschinen und Antriebe - Grundlagen, Betriebsverhalten*. Darmstadt: Springer-Verlag Berlin Heidelberg, 2012.
- [5] Daniel Brehl. *Implementierung und Vergleich verschiedener Reglervarianten zur Regelung eines inversen Doppelpendels*. Bachelorarbeit. Technische Universität Darmstadt, 2014.
- [6] *casadi::NlpSolver Class Reference*. URL: http://casadi.sourceforge.net/v2.0.0/api/html/d6/d07/classcasadi_1_1NlpSolver.html#plugin_NlpSolver_ipopt.
- [7] M. Diehl u. a. „Fast Direct Multiple Shooting Algorithms for Optimal Robot Control“. In: *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Hrsg. von Moritz Diehl und Katja Mombaur. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, S. 65–93. ISBN: 978-3-540-36119-0. URL: https://doi.org/10.1007/978-3-540-36119-0_4.
- [8] Nicolas Fauvé. *Nichtlineare MPC-Regelung eines Doppelpendels*. Studienarbeit. Technische Universität Darmstadt, 2020.
- [9] Andreas Franke. *Modellierung und Regelung eines mechatronischen Systems - Eine Realisierung des invertierten Pendels mit momentengeregeltem Antriebsmotor*. Diplomarbeit. Technische Universität Clausthal, 1997.
- [10] Andreas Kämmerer. *Inbetriebnahme und Regelung eines inversen Doppelpendels*. Studienarbeit. Technische Universität Darmstadt, 2009.
- [11] Viktor Kisner. *Trajektorienfolgerung eines inversen Doppelpendels*. Bachelorarbeit. Technische Universität Darmstadt, 2011.
- [12] Ulrich Konigorski. *Modellbildung und Simulation, Skript*. Darmstadt, 2019.
- [13] Ulrich Konigorski. *Praktikum MATLAB/Simulink II*. Darmstadt, 2019.
- [14] Eric Lenz. *Modellierung Pendel*. Technische Universität Darmstadt, Fachgebiet rtm.
- [15] Karl Max Crépin Noupa. *Reibkompensation am Doppelpendel*. Bachelorarbeit. Technische Universität Darmstadt, 2011.
- [16] Igor de Sousa Ribeiro; Thomas Gassen; Vitaliy Omelchuk. *Steuerung und Regelung eines inversen Doppelpendels*. Projektseminar. Technische Universität Darmstadt, 2020.
- [17] Feiyu Chang; Nicolas Fauvé; Antonia Gießler; Charles Studd Tchoutchui. *Konstruktion und Regelung eines inversen Doppelpendels*. Projektseminar. Technische Universität Darmstadt, 2019.