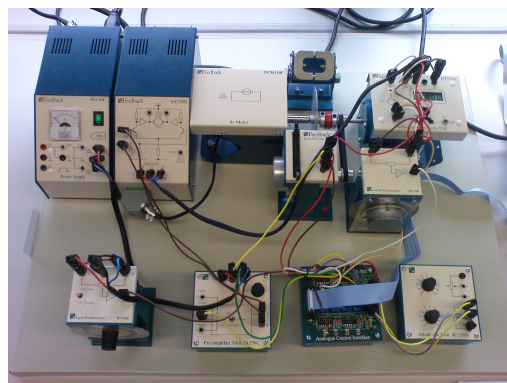
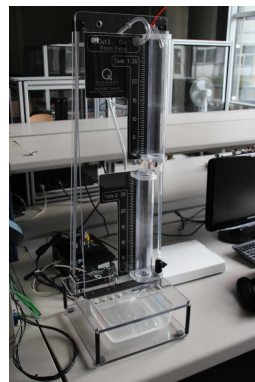




Universität Stuttgart
Institut für Systemtheorie und Regelungstechnik
Prof. Dr.-Ing. F. Allgöwer

— Praktikum zur Vorlesung —

Einführung in die Regelungstechnik



Handbuch

Sommersemester 2019

Inhaltsverzeichnis

1	Einführung	1
2	Systematische Vorgehensweise	2
3	Ablauf des Praktikums	7
4	Versuch: Regelung eines invertierten Pendels	9
5	Versuch: Regelung eines Mehrtank Systems	22
6	Versuch: Identifikation und Regelung eines Gleichstrom-Elektromotors	36
7	Versuch: Identifikation und Regelung eines Torsionsschwingers	58
8	Ansprechpartner	74
	Literaturverzeichnis	75

1

Einführung

Das Praktikum zur Vorlesung *Einführung in die Regelungstechnik* bietet die Möglichkeit, die erlernten Methoden auf praktische Probleme anzuwenden. Durch den Transfer des theoretischen Wissens auf reale Regelungsprobleme wird zum Einen das Verständnis vertieft und zum Anderen eine systematische Vorgehensweise für die Problemlösung erlernt.

Im folgenden Kapitel wird eine allgemeine Vorgehensweise vorgestellt, die es erlaubt, praktische regelungstechnische Probleme systematisch und strukturiert zu lösen. In den Versuchen werden unterschiedliche Themengebiete der Regelungstechnik abgedeckt und jeweils der Bezug zur systematischen Vorgehensweise hergestellt. Die einzelnen Versuche umfassen nicht alle Schritte dieser Vorgehensweise, da die Bearbeitung sonst den zeitlichen Rahmen sprengen würde. Bei jedem Versuch werden exemplarisch ausgewählte Schritte behandelt und vertieft.

Das Praktikum umfasst die folgenden vier Versuche:

- a) Invertiertes Pendel
- b) Mehrtank-System
- c) Identifikation und Regelung eines Gleichstrom-Elektromotors
- d) Torsionsschwinger

2

Ein systematischer Ansatz zur Lösung regelungstechnischer Aufgabenstellungen

Im Folgenden findet sich eine Anleitung zur systematischen Lösung von regelungstechnischen Aufgabenstellungen. Diese Anleitung kann genutzt werden, um bei gegebener Regelungsstrecke eine zugehörige regelungstechnische Aufgabe Schritt für Schritt strukturiert zu lösen.

Das Vorgehen basiert auf [1]. Die folgende Liste fasst das Vorgehen zusammen und gibt einen ersten Überblick über die einzelnen Teilschritte.

- a) Machen Sie sich mit der Strecke vertraut und gewinnen Sie erste Informationen über die regelungstechnischen Ziele.
- b) Modellieren Sie die Strecke und vereinfachen Sie das Modell falls nötig.
- c) Analysieren Sie das Modell und bestimmen Sie wichtige Eigenschaften.
- d) Legen Sie die zu regelnden Größen fest.
- e) Bestimmen Sie die gemessenen Größen und die Stellgrößen; Welche Sensoren und Aktuatoren kommen zum Einsatz und wo an der Strecke werden diese platziert.
- f) Legen Sie eine Regelkreisstruktur fest.
- g) Wählen Sie einen Reglertyp, der zum Einsatz kommen soll.
- h) Legen Sie Spezifikationen für den geschlossenen Kreis hinsichtlich der Gesamtaufgabe fest.
- i) Entwerfen Sie einen Regler.
- j) Analysieren Sie das geregelte System; Modifizieren Sie die Regelkreisstruktur oder den Regler falls die Spezifikationen nicht eingehalten werden.
- k) Simulieren Sie das geregelte System und testen Sie den Regler falls möglich an einer Versuchsanlage.
- l) Iterieren Sie von Schritt 2 an, falls notwendig.
- m) Wählen Sie Hard- und Software um den Regler zu implementieren.
- n) Testen und validieren Sie das geregelte System. Falls erforderlich, nehmen Sie Feinein-

stellungen an den Reglerparametern vor.

Diese einzelnen Schritte sollen nun näher erläutert werden.

- a) **Machen Sie sich mit der Strecke vertraut und gewinnen Sie erste Informationen über die regelungstechnischen Ziele.** In diesem ersten Schritt sollten Sie möglichst viele Informationen über die Regelstrecke und ihre Funktionsweise sammeln. Zu diesem Zweck können gleichermaßen Experten befragt, wie auch Handbücher oder weiterführende Literatur herangezogen werden. In einigen Fällen wird sich auch die Möglichkeit bieten, sich spielerisch mit der Strecke vertraut zu machen und dabei gleichzeitig ein Gefühl für die Strecke zu entwickeln. Schließlich sollten Sie sich, bevor Sie mit dem eigentlichen Reglerentwurfsprozess beginnen, schon einige Gedanken über die Gesamtaufgabe, die damit verbundenen Anforderungen und Einschränkungen, sowie über mögliche Lösungsstrategien machen. Auch wenn solche Punkte zu diesem Zeitpunkt keinesfalls abschließend geklärt werden können, so ist es doch sehr hilfreich, diese Punkte während des gesamten Prozesses im Hinterkopf zu behalten.
- b) **Modellieren Sie die Strecke und vereinfachen Sie das Modell falls nötig.** Grundziel dieses Schrittes ist es, ein mathematisches Modell der zu regelnden Strecke zu erhalten. Üblicherweise wird dabei ein Modell in Form gewöhnlicher Differentialgleichungen angestrebt. Die zu diesem Zweck durchgeführte Modellierung kann einerseits grundlegend und auf physikalischen Grundprinzipien basierend erfolgen. In diesem Fall entsprechen die Parameter des Modells an der Strecke vorkommenden physikalischen Größen. Demzufolge lassen sich diese Parameter häufig direkt messen oder es können andernfalls Verfahren zur Parameteridentifikation angewendet werden. Andererseits kann zur Modellierung auch ein bestimmtes Systemverhalten der Strecke angenommen werden (z.B. PT2). In diesem Fall müssen die Parameter des Modells mit gewissen Identifikationsverfahren aus dem Verhalten der Strecke bestimmt werden.

Innerhalb der gesamten Vorgehensweise ist die Modellierung ein sehr wichtiger Schritt. Alle weiteren Schritte basieren auf dem dabei aufgestellten Modell und falls dieses Modell wichtige Anforderungen nicht erfüllt, ist auch ein für dieses Modell entworfener Regler wertlos. Das Modell sollte einerseits Effekte der realen Strecke in ausreichendem Umfang und mit ausreichender Genauigkeit abbilden. Andererseits sollte die Komplexität des Modells überschaubar bleiben. Daher sollte während der Modellierung immer eine Abwägung zwischen diesen beiden Zielen, nämlich hohe Genauigkeit des Modells auf der einen Seite und geringe Komplexität des Modells auf der anderen Seite, stattfinden. Welcher Grad von Genauigkeit beziehungsweise Komplexität dabei erforderlich beziehungsweise tolerierbar ist, muss im Einzelfall basierend auf den bestehenden Anforderungen entschieden werden.

Darüber hinaus können oder müssen für den Reglerentwurf Effekte wie Totzeiten, schnelle Dynamiken, Nichtlinearitäten, Unstetigkeiten und Beschränkungen oft unberücksichtigt bleiben. Trotzdem können diese Effekte das Systemverhalten entscheidend beeinflussen. Daher ist es erstrebenswert, zu Simulationszwecken über ein Modell zu verfügen, das diese Effekte mit umfasst. Es ist somit in vielen Fällen ratsam, sowohl ein detaillierteres Modell zu Simulations- und Testzwecken, als auch ein vereinfachtes Modell, welches zum Reglerentwurf zum Einsatz kommt, zu bestimmen.

- c) **Analysieren Sie das Modell und bestimmen Sie wichtige Eigenschaften.** In diesem Schritt bestimmen und analysieren Sie relevante Eigenschaften des erstellten Systemmodells. Dies kann entweder geschehen durch Simulation des Systems in verschiedenen aufschlussreichen Szenarien. Oder aber systemtheoretische Eigenschaften können basierend auf dem zuvor hergeleiteten mathematischen Modell systematisch ermittelt und berechnet werden. Typischerweise betrachtete Systemeigenschaften sind zum Beispiel Ruhelagen und dynamische Systemeigenschaften wie Beobachtbarkeit und Steuerbarkeit in Abhängigkeit der gewählten Systemein- und Ausgänge (siehe dazu auch die folgenden Schritte). Schließlich soll für alle so erhaltenen Befunde überprüft werden, ob sie in Übereinstimmung mit der intuitiven Einschätzung und den realen Beobachtungen stehen. Auf diese Weise sollen mögliche Fehler oder zu starke Vereinfachungen im Systemmodell aufgedeckt werden.
- d) **Legen Sie die zu regelnden Größen fest.** Entscheiden Sie in diesem Schritt welche Größen des Systems zu regeln sind. Achten Sie dabei darauf, dass die gesamte Regelungsaufgabe sich leicht mittels dieser Größen formulieren lässt. Außerdem sollten Sie im Blick behalten, dass durch die Wahl von Systemein- und -ausgängen auch die interne Dynamik des Systems festgelegt wird. Die interne Dynamik des Systems sollte stabil falls vorhanden oder aber nicht vorhanden sein.
- e) **Bestimmen Sie die gemessenen Größen und die Stellgrößen; Welche Sensoren und Aktuatoren kommen zum Einsatz und wo werden diese platziert.** Legen Sie fest, wie Ihr System über Ein- und Ausgänge mit seiner Umgebung in Verbindung steht. Für jeden Ausgang, also für jede Größe, die gemessen werden soll, wird ein Sensor benötigt. Jede Größe, die an der Regelstrecke beeinflusst werden soll, erfordert einen Aktuator an dieser. Bei vielen Regelungsaufgaben wird das zu regelnde System schon vollständig feststehen und es bestehen daher keine Freiheiten in der Wahl der Ein- und Ausgänge bzw. der Sensoren und Aktuatoren. In anderen Fällen jedoch können Position, Anzahl und Art der Sensoren und Aktuatoren gewählt und den Anforderungen angepasst werden. Ausschlaggebend sollten dabei nicht nur regelungstechnische Kriterien, sondern auch wirtschaftliche Überlegungen und physikalische Einschränkungen sein.
- Schließlich ist es wichtig, sich darüber klar zu sein, dass in der Praxis nicht zwingend die gemessenen Ausgänge auch gleichzeitig die Regelgrößen sein müssen. Das heißt, es kann sinnvoll sein, Größen zu messen, die nicht direkt für die Regelaufgabe vorgegeben werden sollen, und umgekehrt können im Allgemeinen auch Größen geregelt werden, die nicht direkt gemessen werden können.
- f) **Legen Sie eine Regelkreisstruktur fest.** Um in diesem Schritt eine Regelkreisstruktur zu wählen, müssen Sie entscheiden, über welche Bausteine der komplette Regelkreis verfügen soll (neben Strecke und Regler z.B. Block zur Entkopplung der Eingänge, Störgrößenkompensation, Feedforward Block, Vorfilter, Beobachter, unterlagerte Regelkreise) und Sie müssen festlegen, wie diese einzelnen Bausteine untereinander und mit der Strecke verbunden sind. Gegebenenfalls sollten Sie hier auch festhalten, welche zusätzlichen Schritte notwendig sind, um einen linearen Regler für eine nichtlineare Regelstrecke verwenden zu können. Typischerweise wird das Ergebnis dieser gesamten Überlegungen in Form eines Blockschaltbildes für den kompletten Kreis festgehalten. Dieses Blockschaltbild kann dann beispielsweise direkt in die Struktur eines Simulink-

Modells übersetzt werden und sollte während der folgenden Schritte als Orientierung dienen.

- g) **Wählen Sie einen Reglertyp, der zum Einsatz kommen soll.** Legen Sie zunächst fest, ob der Reglerentwurf im Zeitbereich oder im Frequenzbereich erfolgen soll. Danach wählen Sie den Reglertyp (wie z.B. P-/PI-/PID-Regler oder Zustandsregler mit Beobachter) und eine zugehörige Entwurfsmethode (wie z.B. Loopshaping, Pole Placement, LQR-Entwurf). Je nach Sichtweise überschneidet sich dieser Schritt teilweise mit dem vorhergehenden zur Wahl der Regelkreisstruktur. In jedem Fall sollte aber über den Reglertyp und die Methode zur Bestimmung der zugehörigen Parameter getrennt entschieden werden.
- h) **Legen Sie Spezifikationen für den geschlossenen Kreis hinsichtlich der Gesamtaufgabe fest.** Jetzt sollten Sie die zuvor schon verbal formulierten Anforderungen an den geschlossenen Regelkreis präzisieren und quantitativ festhalten. Über diese Anforderungen an den gesamten Regelkreis werden implizit auch Anforderungen an den Regler gestellt. Behalten Sie daher im Auge, dass Ihre Anforderungen derart gewählt sind, dass Regler existieren, die diese erfüllen können. Außerdem sollten die Anforderungen so formuliert sein, dass sie zum Reglerentwurfsverfahren passen und darin berücksichtigt werden können.
- i) **Entwerfen Sie einen Regler.** Bestimmen Sie in diesem Schritt die Parameter des Reglers so, dass die zuvor definierten Anforderungen eingehalten werden. Um letzteres zu überprüfen, sollte dieser Schritt zusammen mit dem folgenden, der Analyse des resultierenden geschlossenen Kreises, erfolgen.
- j) **Analysieren Sie das geregelte System; Modifizieren Sie die Regelkreisstruktur oder den Regler falls die Spezifikationen nicht eingehalten werden.** Dieser Schritt ist der zentrale Schritt im gesamten Reglerentwurf. Hier entscheiden Sie ob Ihre bisherigen Entscheidungen zielführend waren und das Ergebnis des bisherigen Entwurfsprozesses es erlaubt, die Gesamtanforderungen zu erfüllen. Analysieren Sie dazu das geregelte System mit unterschiedlichen Ihnen zur Verfügung stehenden theoretischen Methoden. Sie können zu diesem Zweck sowohl den offenen als auch den geschlossenen Kreis betrachten und beispielsweise Bode- und Nyquist-Plots ebenso wie Sprungantworten auswerten. Stellt sich in diesem Schritt heraus, dass sich die bestehenden Anforderungen teilweise nicht oder nicht ausreichend erfüllen lassen, so müssen die zuvor getroffenen Entscheidungen überdacht und gegebenenfalls revidiert werden.
- k) **Simulieren Sie das geregelte System und testen Sie den Regler falls möglich an einer Versuchsanlage.** Bevor Sie Ihren Regler an einer echten Strecke implementieren, sollten Sie den Regler in ausführlichen Simulationen testen. In diesen Simulationen können Modelle unterschiedlicher Komplexität zum Einsatz kommen. Beginnen Sie mit Simulationen am Reglerentwurfsmodell, also an dem nominalen Modell, für den der Regler entworfen wurde. Verhält sich dieses geregelte System wie gewünscht, so können Sie im nächsten Schritt realistischere Simulationsmodelle verwenden, um zu testen, ob der Regler auch unter Einbeziehung der im Entwurfsprozess vernachlässigten Effekte das gewünschte Verhalten zeigt. Mit diesen Tests erhöhen Sie die Chancen, dass der Regler auch an der echten Strecke sofort funktioniert und es können möglicherweise folgenschwere Fehler, wie falsche Schnittstellen des Reglers oder Vorzeichenfehler noch

rechtzeitig entdeckt werden.

- l) **Iterieren Sie von Schritt 2 an falls nötig.** Selbst bei streng systematischer Vorgehensweise während des Reglerentwurfsprozesses kann es passieren, dass der resultierende Regler am Ende nicht alle Regelziele erfüllt. In diesem Fall kann es nötig werden, von Schritt 2 an des systematischen Vorgehens zu iterieren. Durch die Erfahrungen und Einsichten, die im ersten Durchgang gewonnen wurden, kann besser abgeschätzt werden, wo mögliche Fehler gemacht wurden und welche Änderungen am entworfenen Regler zielführend sein können.
- m) **Wählen Sie Hard- und Software um den Regler zu implementieren.** Entscheiden Sie, wie der Regler implementiert und realisiert wird. Wichtige Kriterien hierbei sind Speicherbedarf, Rechenleistung und Abtastraten. In einigen Fällen wird die zu verwendende Hard- und Software vorgegeben sein und damit verfügbarer Speicher, Rechenleistung und Abtastraten feststehen. In diesen Fällen sollten diese Anforderungen früh genug im Reglerentwurfsprozess berücksichtigt werden, so dass die entsprechenden Beschränkungen eingehalten werden. Bestehen allerdings Freiheiten in der Wahl von Hard- und Software, so sollten diese nicht nur genutzt werden um systemtheoretischen Anforderungen zu genügen, sondern auch ökonomische Gesichtspunkte sollten berücksichtigt werden. Derartige Überlegungen sollten in jedem Fall schon rechtzeitig in den Entwurfsprozess einfließen. Schließlich sollte auch berücksichtigt werden, dass die meisten Regler zeitdiskret implementiert werden, wenngleich der Entwurf meistens zeitkontinuierlich stattfindet. Da dieser Schritt auch Fehlerpotential birgt, sollte er ebenso rechtzeitig bedacht werden.
- n) **Testen und validieren Sie das geregelte System. Falls erforderlich, nehmen Sie Feineinstellungen an den Reglerparametern vor.** In diesem letzten Schritt testen Sie den Regler am realen System. Beginnen Sie dazu mit moderat eingestellten Reglern und einfachen Manövern der geregelten Strecke. Wenn diese Tests erfolgreich verlaufen, können Sie zu aggressiveren Einstellungen der Reglerparameter und zu komplexeren und schnelleren Manövern übergehen. In jedem Fall sollten Sie alle Regler mit zugehörigen Parametern wiederum zuerst in Simulationen testen, bevor Sie Tests an der realen Strecke durchführen.

3

Ablauf des Praktikums

3.1 Organisation

- Die Gruppen bestehen aus zwei bis vier Studenten
- Das Praktikum wird in zwei Zyklen durchgeführt, wobei jeder Student nur an einem Block teilnimmt
- Jeder Block besteht aus vier Nachmittagen, beginnend jeweils um **14:00h** im **Raum 0.255**.
- Alle nötigen Informationen werden über das ILIAS-System bereitgestellt (<https://ilias3.uni-stuttgart.de/>). Bitte einen Account einrichten und regelmäßig die E-Mails abrufen, oder eine Weiterleitung einrichten.

3.2 Wichtige Termine

1. April bis 21. April	Anmeldung über ILIAS
6. Mai bis 7. Juni	Erster Block
17. Juni bis 19. Juli	Zweiter Block

3.3 Hinweise zur Anmeldung über ILIAS

- Stellen Sie sicher, dass Sie regelmäßig Ihren Account auf neue E-Mails überprüfen, da wichtige, das Praktikum betreffende, Mitteilungen an diesen gesendet werden. Sie können auch E-Mails, die an Ihren ILIAS-Account geschickt werden, an die angegebene E-Mail Adresse weiterleiten lassen. Dies können Sie unter “Persönliches Profil → Mail Einstellungen” einstellen

Informationen für das Praktikum Echtzeitdatenverarbeitung gibt es auf <https://www.isys.uni-stuttgart.de/lehre/praktika/ezdvprak/index.html>.

3 Ablauf des Praktikums

April		Mai		Juni		Juli	
Mo	1 <u>S Anmeldung</u>	Mi	1	Sa	1	Mo	1
Di	2	Do	2	So	2	Di	2
Mi	3	Fr	3	Mo	3	Mi	3
Do	4	Sa	4	Di	4	Do	4
Fr	5	So	5	Mi	5	Fr	5
Sa	6	Mo	6 <u>Start 1. Block</u>	Do	6	Sa	6
So	7	Di	7	Fr	7 <u>Ende 1. Block</u>	So	7
Mo	8 <u>Start Vorl.</u>	Mi	8	Sa	8	Mo	8
Di	9	Do	9	So	9	Di	9
Mi	10	Fr	10	Mo	10 <u>Pfingstferien</u>	Mi	10
Do	11	Sa	11	Di	11	Do	11
Fr	12	So	12	Mi	12	Fr	12
Sa	13	Mo	13	Do	13	Sa	13
So	14	Di	14	Fr	14	So	14
Mo	15	Mi	15	Sa	15	Mo	15
Di	16	Do	16	So	16	Di	16
Mi	17	Fr	17	Mo	17 <u>Start 2. Block</u>	Mi	17
Do	18	Sa	18	Di	18	Do	18
Fr	19	So	19	Mi	19	Fr	19 <u>Ende 2. Block</u>
Sa	20	Mo	20	Do	20	Sa	20 <u>Ende Vorl.</u>
So	21 <u>E Anmeldung</u>	Di	21	Fr	21	So	21
Mo	22	Mi	22	Sa	22	Mo	22
Di	23	Do	23	So	23	Di	23
Mi	24	Fr	24	Mo	24	Mi	24
Do	25	Sa	25	Di	25	Do	25
Fr	26	So	26	Mi	26	Fr	26
Sa	27	Mo	27	Do	27	Sa	27
So	28	Di	28	Fr	28	So	28
Mo	29	Mi	29	Sa	29	Mo	29
Di	30 <u>Infoveranst.</u>	Do	30	So	30	Di	30
		Fr	31			Mi	31

Abbildung 3.1: Kalender

3.4 Praktikumstag

Das Praktikum besteht aus vier Nachmittagen. An jedem Nachmittag wird ein Versuch bearbeitet, der gründlich vorbereitet werden sollte. Dazu gehört:

- Das Kapitel des jeweiligen Versuchs vorher durcharbeiten.
- Die mit einem Stern “★” versehenen Aufgaben **müssen** vorab bearbeitet werden.
- Gegebenenfalls sollten die theoretischen Hintergründe zu den jeweiligen Gebieten wiederholt werden.

Zu Beginn eines jeden Nachmittags wird die Anwesenheit geprüft¹. Vor jedem Versuch muss ein **Kurztest**, der sowohl versuchsspezifisches wie auch regelungstechnisches Wissen abprüft, bestanden werden. Der Test gilt beim Erreichen von mindestens **6 Punkten** als bestanden. Ist dies nicht der Fall, muss der Versuch an einem anderen Tag wiederholt werden. Bei einem zweiten Fehlversuch, wird das gesamte Praktikum als **nicht bestanden** gewertet. Nach dem erfolgreichen Bestehen des Kurztests, kann der Versuch entsprechend der Unterlagen bearbeitet werden. Fragen können an die Hiwis sowie an die Mitarbeiter gerichtet werden. Im Anschluss an jeden Versuch wird ein kurzes Abschlussgespräch geführt.

¹Im Falle von Krankheit ist dem Verantwortlichen des Praktikums rechtzeitig Bescheid zu geben.

4

Versuch: Regelung eines invertierten Pendels

4.1 Einleitung

Anhand dieses Praktikumversuchs sollen die in der Vorlesung *Einführung in die Regelungstechnik* gelernten Grundlagen – Modellbildung, Analyse von Regelkreisen, Regler- und Beobachterentwurf – an einer realen Regelstrecke vertieft werden. Als Regelstrecke wird ein invertiertes Pendel verwendet. Das invertierte Pendel ist ein beliebtes Beispiel in der Regelungstechnik, weil die Regelstrecke instabil und nichtlinear ist und die Anforderungen an den Reglerentwurf somit hoch sind.

Nach der Beschreibung des Praktikumversuchsaufbaus in Kapitel 4.2 und der Herleitung des mathematischen Modells der Regelstrecke in Kapitel 4.3 wird zunächst in Kapitel 4.4 die Zustandsraumdarstellung der Regelstrecke aufgestellt und deren Dynamik hinsichtlich Stabilität, Steuerbarkeit und Beobachtbarkeit analysiert. Anschließend wird in Kapitel 4.5 zur Stabilisierung des invertierten Pendels um die obere, instabile Ruhelage eine Zustandsrückführung entworfen. An der realen Anlage sind nur der Pendelwinkel und die Schlittenposition, nicht aber die Pendelwinkelgeschwindigkeit und die Schlittengeschwindigkeit messbar. Da eine Zustandsrückführung jedoch für die technische Realisierung alle Zustände der Regelstrecke benötigt, müssen die nichtmessbaren Zustände für den Reglerentwurf rekonstruiert werden. Dies soll in Kapitel 4.6 mittels eines Luenberger-Beobachters geschehen. Nach dem Beobachterentwurf wird der Regler bestehend aus einer Zustandsrückführung und einem Luenberger-Beobachter in Matlab/Simulink getestet und anschließend an der Anlage implementiert. In Kapitel 4.7 wird der Regler aus Kapitel 4.6 schließlich um ein zusätzliches Vorfilter erweitert, mit dem der Schlitten einer sprungförmigen Änderung des Sollwerts in der Schlittenposition ohne bleibende Regelabweichung folgen kann.

Lernziele:

- Modellbildung für ein mechanisches System sowie Vereinfachung und Linearisierung des resultierenden nichtlinearen Systemmodells.
- Systemanalyse sowie Entwurf einer beobachterbasierten Zustandsrückführung auf Grundlage des linearisierten Modells.

Insbesondere werden die Punkte a)–c), f) und j)–l) der strukturierten Vorgehensweise aus Kapitel 2 behandelt.

Notwendige Vorkenntnisse:

- Linearisierung und Systemanalyse
- Zustandsrückführung und Luenberger-Beobachter

Notwendige Vorleistung:

- Lösen der mit ★ gekennzeichneten Aufgaben in den Kapiteln 4.4, 4.5, 4.6, und 4.7

4.2 Versuchsaufbau

Der Versuchsaufbau besteht aus einem Computer mit dSPACE-Karte, einem Frequenzumrichter mit integrierter Regelung des Synchronmotors und der in Abbildung 4.1 gezeigten Regelstrecke. Die Regelstrecke besteht aus einem Schlitten, der sich auf einer zwei Meter langen Schiene bewegen kann. Auf diesem Schlitten ist das Pendel mit Hilfe eines drehbaren Lagers montiert. Diese Anordnung wird über einen Riemen durch einen Synchronmotor angetrieben. Das Stellglied des Synchronmotors ist ein Frequenzumrichter mit einer integrierten Stromregelung und Drehzahlregelung. Die Stellgröße der Regelstrecke ist somit die Drehzahl des Synchronmotors bzw. die Geschwindigkeit des Schlittens. Diese Stellgröße wird von der dSPACE-Karte mittels eines D/A-Wandler bereit gestellt. Die Messgrößen der Regelstrecke sind der Winkel des Pendels und die Position des Schlittens auf der Schiene. Der Winkel wird über einen Inkrementalgeber erfasst. Die Schlittenposition kann aus der Ansteuerung des Motors ausgelesen werden, die die Drehung des Motors ebenfalls über einen Inkrementalgeber misst. Die beiden Inkrementalgeber sind über eine digitale I/O-Karte mit der dSPACE-Karte verbunden.



Abbildung 4.1: Invertiertes Pendel mit Schlitten.

4.3 Modell der Regelstrecke

Im Folgendem wird die Modellierung des invertierten Pendels vorgenommen.

4.3.1 Bewegungsgleichungen

Die Bewegungsgleichungen der in Abbildung 4.2 schematisch gezeigten Regelstrecke können mittels der Lagrangeschen Gleichungen 2. Art hergeleitet werden (siehe Anhang). Die Bewegungsgleichungen der Regelstrecke werden durch folgende nichtlineare Differentialgleichungen beschrieben:

$$2l\ddot{\varphi} - 3\cos\varphi \cdot \ddot{w} - 3g\sin\varphi = 0 \quad (4.1)$$

$$(m_S + m_P)\ddot{w} - \frac{1}{2}m_P l \cos\varphi \cdot \ddot{\varphi} + \frac{1}{2}m_P l \sin\varphi \cdot \dot{\varphi}^2 = \frac{M}{R}, \quad (4.2)$$

wobei φ der Winkel des Pendels, w die Position des Schlittens, R der Radius des Motors, M das Motormoment, $l = 0.988 \text{ m}$ die Länge des Pendels, m_S die Masse des Schlittens, m_P die Masse des Pendels, und $g = 9.81 \frac{\text{m}}{\text{s}^2}$ die Gravitationsbeschleunigung ist.

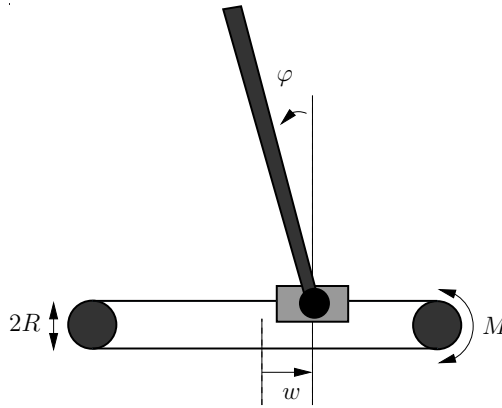


Abbildung 4.2: Invertiertes Pendel mit Schlitten.

4.3.2 Vereinfachte Bewegungsgleichungen

Da der Synchronmotor mit einer unterlagerten Drehzahlregelung betrieben wird, kann der Einfluss des Pendels auf den Schlitten vernachlässigt werden. Es ist daher möglich, bei hinreichender Genauigkeit mit der vereinfachten Regelstrecke

$$2l\ddot{\varphi} - 3\cos\varphi \cdot \ddot{w} - 3g\sin\varphi = 0 \quad (4.3)$$

$$\dot{w} = \zeta, \quad (4.4)$$

fortzufahren, wobei ζ der neue Eingang ist. Die direkte Vorgabe der Schlittengeschwindigkeit \dot{w} führt zu dem Problem, dass die Schlittenbeschleunigung \ddot{w} , die in Gleichung (4.3) erscheint, nicht mehr explizit im System auftaucht. Dieses Problem kann behoben werden, indem ein

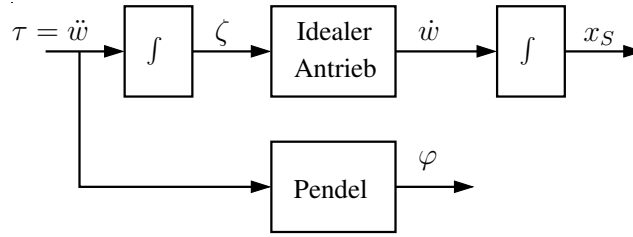


Abbildung 4.3: Invertiertes Pendel mit Schlitten.

Integrator vor das Stellglied eingefügt wird. Somit hat das System die künstliche Stellgröße $\tau = \ddot{w}$ (siehe Abbildung 4.3).

Führt man die Zustände $z = [z_1, z_2]^T = [\varphi, \dot{\varphi}]^T$ ein, so kann die nichtlineare Bewegungsgleichung (4.3) des invertierten Pendels mit dem messbaren Zustand φ in folgender Zustandsraumdarstellung geschrieben werden:

$$\dot{z} = \begin{bmatrix} z_2 \\ \frac{3g}{2l} \sin z_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{3}{2l} \cos z_1 \end{bmatrix} \tau \quad (4.5)$$

$$y_1 = z_1. \quad (4.6)$$

Die vereinfachte Bewegungsgleichung (4.4) für den Schlitten ergibt sich mit den Zuständen $x_S = [x_S, \dot{x}_S]^T = [w, \dot{w}]^T$ und dem messbaren Zustand w zu folgender Zustandsraumdarstellung:

$$\dot{x}_s = A_S x_S + b_S \tau \quad (4.7)$$

$$y_2 = c_S^T x_S, \quad (4.8)$$

wobei

$$A_S = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad b_S = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c_S^T = [1 \quad 0].$$

4.4 Analyse der Dynamik des Pendels

In diesem Teil des Praktikums soll die lineare Zustandsraumdarstellung der Regelstrecke mit den Systemgleichungen (4.5), (4.6), (4.7), (4.8) aufgestellt und analysiert werden.

★ Aufgabe 4.1: Linearisierung und Analyse

- a) Linearisieren Sie die Systemgleichungen (4.5) und (4.6) um die obere Ruhelage $z_R = [0, 0]^T$, $\tau_R = 0$. Geben Sie die linearisierte Regelstrecke in folgender Form mit $x_P = z - z_R$ und $u = \tau - \tau_R$ an:

$$\dot{x}_P = A_P x_P + b_P u, \quad (4.9)$$

$$y_1 = c_P^T x_P. \quad (4.10)$$

- b) Für den späteren Entwurf der Zustandsrückführung und des Beobachters wird die Zustandsraumdarstellung des Gesamtsystems benötigt. Fassen Sie die Gleichungen (4.7), (4.8), (4.9) und (4.10) zu einer Zustandsraumdarstellung in Abhängigkeit von A_P , b_P , c_P^T , A_S , b_S , c_S^T in der Form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx,\end{aligned}\tag{4.11}$$

zusammen, wobei $x = [x_P, x_S]^T$ und $y = [y_1, y_2]^T$ ist.

- c) Bestimmen Sie die Eigenwerte der Regelstrecke (4.11).
d) Ist die Regelstrecke (4.11) asymptotisch stabil?
e) Überprüfen Sie die Regelstrecke (4.11) auf Steuerbarkeit.
f) Überprüfen Sie die Regelstrecke (4.11) auf Beobachtbarkeit.

4.5 Entwurf einer Zustandsrückführung

In diesem Teil des Praktikums soll die Regelstrecke mit den Systemgleichungen (4.3) und (4.4) mittels der in Abbildung 4.4 gezeigten Zustandsrückführung

$$u = -k^T x + r$$

stabilisiert werden. Zunächst wird davon ausgegangen, dass alle Zustände der Regelstrecke messtechnisch erfassbar sind und für die Berechnung der Stellgröße zur Verfügung stehen.

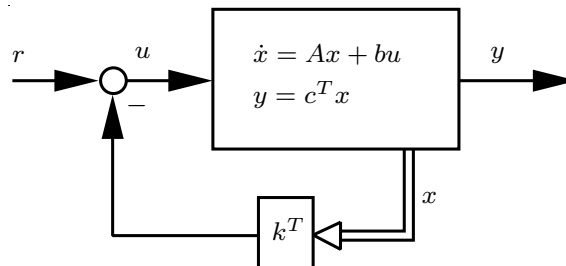


Abbildung 4.4: Zustandsrückführung.

Aufgabe 4.2: Entwurf und Simulation mit linearem Modell

Erstellen Sie zunächst ein Modell der linearen Regelstrecke in Matlab/Simulink. Entwerfen Sie anschließend eine Zustandsrückführung für die Regelstrecke (4.11) und testen Sie die entworfene Zustandsrückführung mit dem linearen Modell. Es stehen Ihnen folgende Matlab/Simulink-Files zur Verfügung: ReglerLin.mdl und ParameterReglerLin.m.

- a) Erstellen Sie ein Modell der Regelstrecke (4.11) im Matlab/Simulink-File ReglerLin.mdl.
b) Ergänzen Sie das Matlab-File ParameterReglerLin.m um den Entwurf einer Zustandsrückführung (Benutzen Sie den Matlab-Befehl `place`). Bei dem Entwurf der Zustandsrückführung sollen folgende Forderungen berücksichtigt werden:

- Stabilität des geschlossenen Regelkreises: Pole links
 - Schnelles Einschwingen: Pole weit nach links
 - Kleine Stellgröße: Pole nicht zu weit nach links
- c) Erweitern Sie das von Ihnen erstellte Matlab/Simulink-File ReglerLin.mdl um die von Ihnen entworfene Zustandsrückführung.
- d) Testen Sie das dynamische Verhalten des geschlossenen Regelkreises für die Anfangsbedingung $x(0) = [0.1 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}, 0 \text{ m}, 0 \frac{\text{m}}{\text{s}}]^T$. Ändern Sie die Parameter der Zustandsrückführung so lange, bis der geschlossene Regelkreis eine zufriedenstellende Performance hat und die Stellgrößenbeschränkung $|\zeta| \leq 3.5 \frac{\text{m}}{\text{s}}$ mit der Anfangsbedingung $x(0) = [0.1 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}, 0 \text{ m}, 0 \frac{\text{m}}{\text{s}}]^T$ eingehalten wird.
- e) Testen Sie nun das dynamische Verhalten des geschlossenen Regelkreises für die Anfangsbedingungen $x(0) = [1.0 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}, 0 \text{ m}, 0 \frac{\text{m}}{\text{s}}]^T$. Wie verhält sich der geschlossene Regelkreis?

Aufgabe 4.3: Simulation mit nichtlinearem Modell

Erstellen Sie nun ein Modell der nichtlinearen Regelstrecke in Matlab/Simulink. Testen Sie anschließend die in Aufgabe 4.2 entworfene Zustandsrückführung an dem nichtlinearen Modell. Es stehen Ihnen folgende Matlab/Simulink-Files zur Verfügung: ReglerNlin.mdl und ParameterReglerNlin.m.

- a) Erstellen Sie ein Modell der nichtlinearen Regelstrecke im Matlab/Simulink-File ReglerNlin.mdl gemäß den Gleichungen (4.5), (4.6), (4.7) und (4.8).
- b) Ergänzen Sie das Matlab-File ParameterReglerNlin.m um den Entwurf einer Zustandsrückführung. Verwenden Sie dazu die in Aufgabe 4.2 ermittelten Werte.
- c) Erweitern Sie das von Ihnen erstellte Matlab/Simulink-File ReglerNlin.mdl um die Zustandsrückführung.
- d) Testen Sie nun das dynamische Verhalten des geschlossenen Regelkreises. Beginnen Sie zunächst wieder mit der Anfangsbedingung $x(0) = [0.1 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}, 0 \text{ m}, 0 \frac{\text{m}}{\text{s}}]^T$ und erhöhen Sie dann die Anfangsbedingung für den Pendelwinkel schrittweise bis $x(0) = [1.0 \text{ rad}, 0 \frac{\text{rad}}{\text{s}}, 0 \text{ m}, 0 \frac{\text{m}}{\text{s}}]^T$. Was hat sich gegenüber den Simulationsergebnissen am linearen Modell von Aufgabe 4.2e) geändert?

4.6 Zustandsrückführung mit Beobachter

Im vorangegangenen Teil des Praktikums wurde eine Zustandsrückführung für die Regelstrecke entworfen. Die Eigenwerte konnten dabei beliebig platziert werden, da die Strecke steuerbar ist. Eine Voraussetzung für die Realisierung einer Zustandsrückführung ist, dass der gesamte Zustand messbar ist – wovon Sie bei der Simulation in Kapitel 4.5 bisher ausgegangen sind. Bei der realen Regelstrecke sind aber nur der Winkel des Pendels und die Position des Schlittens messbar, nicht aber die Winkelgeschwindigkeit des Pendels und die Geschwindigkeit des Schlittens. Somit müssen die nichtmessbaren Zustände mittels eines Beobachters rekonstruiert werden. Dies soll mittels eines Luenberger-Beobachters geschehen. Hierzu muss der in der Vorlesung eingeführte Beobachterentwurf auf Mehrgrößensysteme erweitert werden, da die vorliegende Regelstrecke zwei messbare Ausgänge hat. Die Erweiterung des Beobachterentwurfs von SISO- (Single-Input-Single-Output) auf MIMO- (Multi-Input-Multi-Output) Systeme ist jedoch relativ einfach, wenn für die Berechnung der Beobachterkorrekturmatrix auf Matlab zurückgegriffen wird. Die in diesem Kapitel zu realisierende Regelung bestehend aus einer Zustandsrückführung und einem Beobachter ist in Abbildung 4.5 zu sehen.

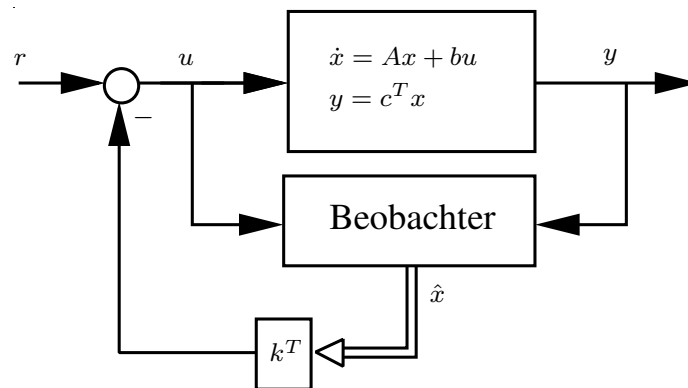


Abbildung 4.5: Zustandsrückführung mit Beobachter.

★ Aufgabe 4.4: Luenberger-Beobachter

- Abbildung 4.6 zeigt das Modell eines Luenberger-Beobachters. Stellen Sie die Differentialgleichung für den Zustand \hat{x} auf.
- Geben Sie die Differentialgleichung für die Beobachterfehlerdynamik $\dot{e} = \dot{x} - \dot{\hat{x}}$ an, wobei x der Zustand der Regelstrecke und \hat{x} der Zustand des Luenberger-Beobachters ist.
- Welche systemtheoretische Eigenschaft der Regelstrecke muss vorausgesetzt werden, damit sich die Eigenwerte der Beobachterfehlerdynamik beliebig platzieren lassen?
- Der Luenberger-Beobachter ist ein Beobachter für lineare Systeme. Die Dynamik der Regelstrecke des invertierten Pendels mit Schlitten ist jedoch nichtlinear. Kann der

Luenberger-Beobachter aus Abbildung 4.6 für alle Anfangszustände $x(0)$ der nichtlinearen Regelstrecke und für alle Anfangszustände $\hat{x}(0)$ des Beobachters den Zustand x richtig rekonstruieren bzw. schätzen?

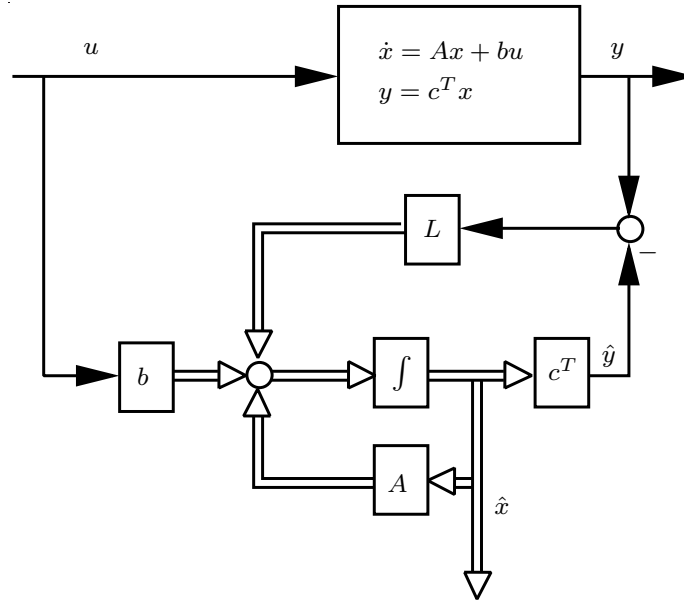


Abbildung 4.6: Luenberger-Beobachter.

★ Aufgabe 4.5: Separationsprinzip

In diesem Teil des Praktikums soll die bereits entworfene Zustandsrückführung um einen Luenberger-Beobachter ergänzt werden. Es stellt sich nun die Frage, ob und wie das Stabilitätsverhalten des geschlossenen Regelkreises beeinflusst wird, wenn der Beobachter wie in Abbildung 4.5 dargestellt im Rückkopplungszweig gemeinsam mit der Zustandsrückführung eingesetzt wird. Stellen Sie dazu zunächst einige Untersuchungen am geschlossenen Regelkreis an.

- Geben Sie eine Zustandsraumdarstellung des geschlossenen Regelkreises mit dem Zustandsvektor $[x, e]$ an, wobei $e = x - \hat{x}$ ist. Was kann man bezüglich der Stabilität des geschlossenen Regelkreises sagen?

Aufgabe 4.6: Entwurf eines Beobachters und Simulation

In diesem Teil des Praktikums soll die in Abbildung 4.5 gezeigte Regelung bestehend aus einer Zustandsrückführung und einem Luenberger-Beobachter in Matlab/Simulink getestet werden. Für die folgende Aufgabe stehen Ihnen die Matlab/Simulink-Files zur Verfügung: Ausgangsregler.mdl und ParameterAusgangsregler.m.

- Erstellen Sie im Matlab/Simulink-File Ausgangsregler.mdl die Regelung bestehend aus Zustandsrückführung und Beobachter entsprechend den Abbildungen 4.5 und 4.6. Das

File Ausgangsregler.mdl enthält bereits das nichtlineare Modell der Regelstrecke.

- b) Berechnen Sie die Beobachtermatrix L im Matlab-File ParameterAusgangsregler.m. Benutzen Sie hierzu den Matlab-Befehl `place`. Die Werte für die Zustandsrückführung können aus den vorangegangenen Aufgaben übernommen werden. Beachten Sie bei dem Entwurf des Beobachters folgende Forderungen:
- Stabilität des Beobachters : Pole links
 - Schnelles Abklingen des Beobachterfehlers: Pole weit nach links
 - Starkes Messrauschen: Pole nicht zu weit nach links
- c) Testen Sie das dynamische Verhalten Ihres Modell für verschiedene Eigenwertplatzierungen der Beobachterdynamikmatrix und verschiedene Anfangsbedingungen.

Aufgabe 4.7: Implementierung der Regelung

Der von Ihnen entworfene Regler soll nun an der Anlage implementiert werden.

Nehmen Sie die Anlage nur in Anwesenheit eines Betreuers in Betrieb!

- a) Ergänzen Sie das Matlab-File ParameterReglerdSPACE.m um die Reglerparameter und rufen das File auf.
- b) Ergänzen Sie das Matlab/Simulink-File ReglerdSPACE.mdl um den Regler aus Aufgabe 4.6. Kopieren Sie dazu den Regler aus Aufgabe 4.6 in das Matlab/Simulink-File ReglerdSPACE.mdl. Erweitern Sie alle Integratoren um einen externen Reset. Kompilieren Sie dann das Matlab/Simulink-File ReglerdSPACE.mdl.
- c) Starten Sie den dSPACE ControlDesk (Version 3.7) und öffnen Sie das Experiment Pendel.cdx. Starten Sie anschließend über den Menüpunkt Instrumentation den Animation Mode.
- d) Schieben Sie den Schlitten in die Mitte der Strecke und lassen Sie das Pendel ruhig nach unten hängen. Schalten Sie die Anlage an und drücken Sie den grünen Knopf.
- e) Kalibrieren Sie das Messsystem. Klicken Sie in der Working Area im Feld “Kalibrierung” auf “Start” und anschließend auf “Stop”.
- f) Klicken Sie in der Working Area im Feld “Regler” auf “Start”. Stellen Sie das Pendel auf und drücken Sie am Pendel einmal auf die schwarze Taste. Achten Sie auf Ihre Hand!

Hinweis: Sollten Probleme auftreten, klicken Sie in der Working Area im Feld “Regler” sofort auf “Stop”!

- g) Stören Sie das Pendel durch Berühren. Vergleichen Sie den Zeitverlauf mit den Simulationsergebnissen aus Aufgabe 4.6.
- h) Untersuchen Sie das dynamische Verhalten der Schlittenposition für einen Sollwertsprung von $r = 0.2$ m. Warum fährt der Schlitten zuerst in die falsche Richtung? Wie groß ist die bleibende Regelabweichung?
- i) Schließen Sie alle Matlab/Simulink-Files und beenden Sie dSPACE.

4.7 Erweiterung des Reglers um einen Vorfilter

In den vorangegangenen Aufgaben wurde das Pendel um die obere, instabile Ruhelage stabilisiert. Ein Verfahren des Schlittens aus der Ruhelage war bisher jedoch nur mit einer bleibenden Regelabweichung möglich (siehe Aufgabe 4.7). Um das Führungsverhalten des Reglers bezüglich der Schlittenposition zu verbessern und die bleibende Regelabweichung zu beseitigen, wird der in Kapitel 4.6 entworfene Regler um ein Vorfilter erweitert. Wie aus Abbildung 4.7 zu entnehmen ist, wird statt der bisherigen Stellgröße $u = -k^T x + r$ nun die Stellgröße

$$u = -k^T x + vr$$

verwendet. Der Effekt dieser Maßnahme wird ersichtlich, wenn man das stationäre Verhalten des Regelkreises und den bleibenden Regelfehler betrachtet. Dies soll in der folgenden Aufgabe 4.8 näher untersucht werden.

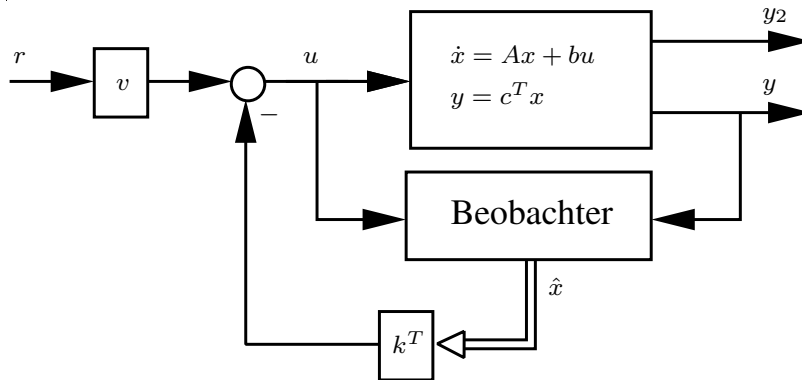


Abbildung 4.7: Zustandsrückführung mit Vorfilter.

★ Aufgabe 4.8: Stationäres Verhalten des Regelkreises

Entwerfen Sie einen Vorfilter v so, dass der Schlitten einer sprungförmigen Änderung des Sollwerts in der Schlittenposition ohne bleibende Regelabweichung folgen kann. Entwerfen Sie das Vorfilter mit dem System

$$\begin{aligned} \dot{x} &= (A - Bk^T)x + Bvr \\ y_2 &= c^T x, \quad (c^T = [0, 0, c_S^T]). \end{aligned}$$

- Berechnen Sie das Vorfilter v , so dass die bleibende Regelabweichung $\lim_{t \rightarrow \infty} e(t) = r(t) - y_2(t)$ für eine sprungförmige Änderung in der Führungsgröße r der Schlittenposition verschwindet.
- Warum muss der Luenberger-Beobachter bei dem Entwurf des Vorfilters nicht berücksichtigt werden?

Aufgabe 4.9: Zustandsrückführung mit Vorfilter

Erweitern Sie nun das Matlab-File ParameterAusgangsregler.m und das Matlab/ Simulink-File Ausgangsregler.mdl aus Aufgabe 4.6 um das Vorfilter.

- a) Testen Sie die Regelung mit einem Sollwertsprung $r = 0.2$ m.

Aufgabe 4.10: Implementierung des Vorfilters

In dieser Aufgabe soll die Zustandsrückführung mit Beobachter und Vorfilter an der Anlage implementiert werden. Es stehen Ihnen die Matlab/Simulink-Files aus Aufgabe 4.7 zur Verfügung: ParameterReglerdSPACE.m und ReglerdSPACE.mdl.

- a) Ergänzen Sie das Matlab-File ParameterReglerdSPACE.m aus Aufgabe 4.7 um den Entwurf des Vorfilters.
- b) Ergänzen Sie das Matlab/Simulink-File ReglerdSPACE.mdl aus Aufgabe 4.7 um das Vorfilter. Kompilieren Sie das Matlab/Simulink-File ReglerdSPACE.mdl.
- c) Gehen Sie für die Inbetriebnahme der Anlage wie in Aufgabe 4.7 vor.
- d) Untersuchen Sie das dynamische Verhalten der Schlittenposition für einen Sollwertsprung von $r = 0.2$ m. Wie groß ist die bleibende Regelabweichung? Hat sich das Führungsverhalten im Vergleich zu Aufgabe 4.7 verbessert?
- e) Schließen Sie alle Matlab/Simulink-Files und den dSPACE ControlDesk.

4.8 Anhang: Modell der Regelstrecke

Die Lagrangeschen Gleichungen 2. Art lauten allgemein:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i,$$

wobei L die Lagrange-Funktion, q_i die verallgemeinerten Koordinaten und Q_i die nichtkonservativen Kräfte, z.B. Reibung, darstellen. Die verallgemeinerten Koordinaten des invertierten Pendels sind

$$q_1 = \varphi, \quad q_2 = w,$$

wobei φ der Pendelwinkel und w die Schlittenposition darstellt. Das invertierte Pendel wird ohne Reibung modelliert. Als nichtkonservative Kraft ergibt sich somit nur die durch den Motor erzeugte Kraft

$$Q_2 = \frac{M}{R},$$

wobei M das Motormoment und R der Radius des zylinderförmigen Motors ist. Die Lagrange-Funktion L berechnet sich aus der Differenz der kinetischen Energie T und der potentiellen Energie U des Gesamtsystems:

$$L = T_{ges} - U_{ges} = T_{Pendel} + T_{Schlitten} - U_{Pendel}.$$

Die potentielle und kinetische Energie ist:

$$\begin{aligned} U_{Pendel} &= m_P g \frac{l}{2} \cos \varphi, \\ T_{Pendel} &= \frac{1}{2} m_P (\dot{w}^2 - l \dot{w} \dot{\varphi} \cos \varphi + \frac{1}{3} l^2 \dot{\varphi}^2), \\ T_{Schlitten} &= \frac{1}{2} m_S \dot{w}^2. \end{aligned}$$

Einsetzen ergibt die Lagrange-Funktion:

$$L = \frac{1}{2} [(m_S + m_P) \dot{w}^2 - m_P l \dot{w} \dot{\varphi} \cos \varphi + \frac{1}{3} m_P l^2 \dot{\varphi}^2 - m_P g l \cos \varphi].$$

Damit ergibt sich für die Teilsysteme:

Pendel:

$$\begin{aligned} \frac{\partial L}{\partial \dot{\varphi}} &= \frac{1}{3} m_P l^2 \dot{\varphi} - \frac{1}{2} m_P l \dot{w} \cos \varphi \\ \frac{\partial L}{\partial \varphi} &= \frac{1}{2} m_P l \sin \varphi (\dot{w} \dot{\varphi} + g). \end{aligned}$$

Wagen:

$$\begin{aligned} \frac{\partial L}{\partial \dot{w}} &= (m_S + m_P) \dot{w} - \frac{1}{2} m_P l \dot{\varphi} \cos \varphi \\ \frac{\partial L}{\partial w} &= 0. \end{aligned}$$

Setzt man nun die ermittelten Differentialgleichungen für die verallgemeinerten Koordinaten $q_1 = \varphi$ und $q_2 = w$ in die Lagrangeschen Gleichungen 2. Art ein, so folgen hieraus die nichtlinearen Bewegungsgleichungen für das Gesamtsystem:

$$\begin{aligned} 2l\ddot{\varphi} - 3\cos\varphi \cdot \ddot{w} - 3g\sin\varphi &= 0 \\ (m_S + m_P)\ddot{w} - \frac{1}{2}m_Pl\cos\varphi \cdot \ddot{\varphi} + \frac{1}{2}m_Pl\sin\varphi \cdot \dot{\varphi}^2 &= \frac{M}{R}. \end{aligned}$$

5

Versuch: Regelung eines Mehrtank Systems

Lernziele

- Linearisierung des nichtlinearen Modells
- Entwurf einer Ausgangsregelung (PI-Regler und Kaskadenregelung)
- Entwurf einer Zustandsregelung (mit und ohne I-Anteil)
- Unterschiede zwischen verschiedenen Reglern kennenlernen in der Simulation und am realen Versuchsaufbau

Der Schwerpunkt des Praktikumsversuchs liegt bei der Bildung des linearen Modells und der Regelung mit verschiedenen Reglern. Insgesamt werden die Punkte c), g), i) und k) der systematischen Vorgehensweise aus Kapitel 2 behandelt.

Notwendige Vorkenntnisse:

- Linearisierung
- Loopshaping
- Kaskadenregelung
- Zustandsrückführung

Notwendige Vorleistungen:

- Die mit ★ gekennzeichneten Aufgaben sind vor dem Praktikum zu bearbeiten und am Tag der Versuchsdurchführung mitzubringen.

5.1 Versuchsbeschreibung

5.1.1 Versuchsaufbau

Der Versuchsaufbau besteht aus mehreren Einheiten (siehe Abbildung 5.1).

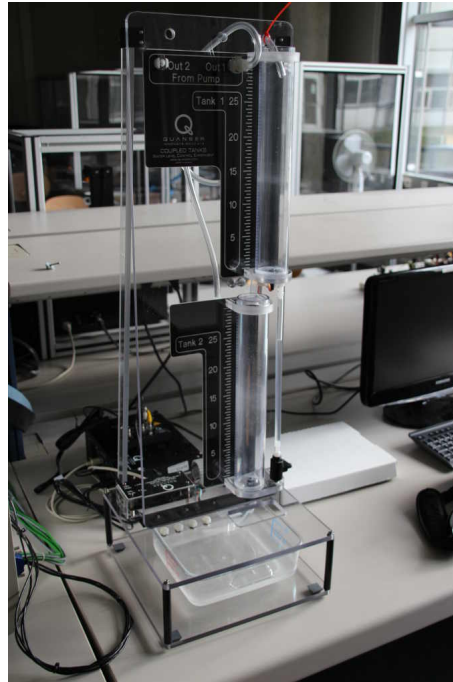


Abbildung 5.1: Versuchsaufbau des 2-Tank Systems.

In der Mitte befindet sich der mechanische Aufbau des 2-Tank Systems. Die Füllstände in den einzelnen Behältern werden mit Drucksensoren gemessen. Hinter dem Aufbau sind ein Verstärker, sowie eine USB Datenschnittstelle von QUANSER zu sehen. Über diese USB Schnittstelle findet die Kommunikation mit dem Versuchsrechner statt, auf dem Matlab/Simulink, sowie die Quanser Real Time Control Software QUARC läuft.

5.1.2 Systembeschreibung

Der schematische Aufbau des 2-Tank Systems ist in Abbildung 5.2 dargestellt.

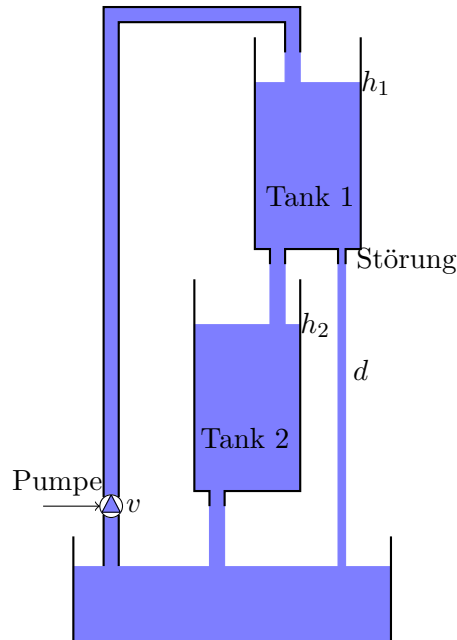


Abbildung 5.2: 2-Tank System.

Ziel der Regelung ist es, den Füllstand h_2 des unteren Behälters zu regeln. Dazu wird als Stellgröße die Fördermenge v der Pumpe, bzw. später die Spannung u , die diese erzeugt, verwendet. Der Abfluss d aus Tank 1 wirkt dabei als Störung auf das System.

5.1.3 Modellierung

1-Tank System:

Zur Herleitung des mathematischen Modells betrachte man zuerst ein 1-Tank System wie in Abbildung 5.3 dargestellt,

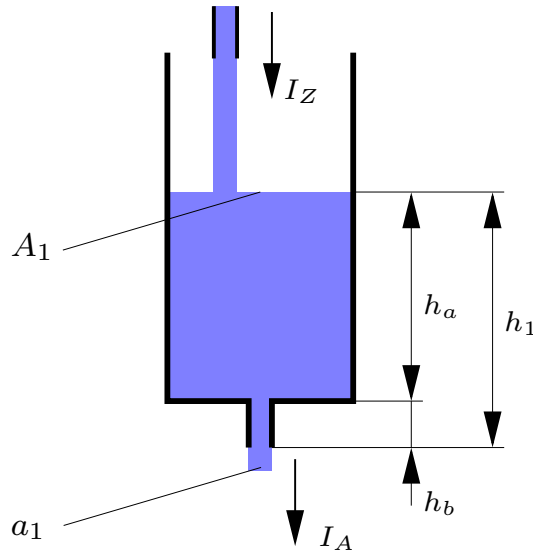


Abbildung 5.3: 1-Tank System.

wobei A_1 den Behälterquerschnitt, a_1 den Ausflussquerschnitt und I_Z und I_A den Zufluss bzw. Abfluss bezeichnen. Weiter bezeichnet h_a die Füllhöhe und h_b den Abstand zwischen Behälterboden und Unterkante des Ausflussventils. Aus der Bernoulli-Gleichung folgt

$$I_A = a_1 v_a = a_1 \mu \sqrt{2gh_1},$$

wobei v_a die Abflussgeschwindigkeit bezeichnet. Die Konstante μ ist ein Maß für den hydrodynamischen Widerstand des Ausflussventils. Die zeitliche Ableitung des Wasservolumens V im Behälter ergibt sich aus der Bilanzgleichung

$$\dot{V} = A_1 \frac{dh_1}{dt} = I_Z - I_A.$$

Somit erhält man

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1} \mu \sqrt{2gh_1} + \frac{1}{A_1} I_Z.$$

2-Tank System:

Aus dem 1-Tank System kann direkt das 2-Tank Modell (5.1) gewonnen werden mit den Zuflüssen v in Behälter 1 und $a_1\sqrt{2gh_1}$ in Behälter 2. Im Folgenden sei $\mu = 1$.

Aus den Bilanzgleichungen und der Bernoulli-Gleichung folgt für das Modell ohne Berücksichtigung der Störgröße d

$$\begin{aligned}\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{1}{A_1}v \\ \frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_1}{A_2}\sqrt{2gh_1}\end{aligned}\tag{5.1}$$

wobei mit h_i , A_i , a_i der Füllstand, der Behälterquerschnitt und der Ausflussquerschnitt des Behälters i bezeichnet wird. g ist die Erdbeschleunigung.

5.2 Matlab/Simulink/QUARC Interface

Folgende Matlab-Skripte und Simulink-Kopplungspläne stehen zum Initialisieren, Simulieren und zum Experimentieren unter `C:\Users\student\Documents\MATLAB\Mehrtankversuch\student` zur Verfügung:

- `parameter.m`: Mit diesem Matlab-Skript werden die Parameter für das 2-Tank System (Durchmesser usw.) festgelegt. Hier sind keine Änderungen vorzunehmen.
- `entwurf.m`: Mit diesem Matlab-Skript wird der Arbeitspunkt festgelegt. Es ist die stationäre Fördermenge v_0 festzulegen. Das lineare Modell wird hier definiert. Zusätzlich werden die Regler in diesem Skript ausgelegt.
- `sim_pi.slx`: Mit diesem Simulink-Kopplungsplan wird der PI-Ausgangsregler implementiert und simuliert.
- `sim_kask.slx`: Mit diesem Simulink-Kopplungsplan wird der Kaskadenregler implementiert und simuliert.
- `sim_zust.slx`: Mit diesem Simulink-Kopplungsplan wird der Zustandsregler implementiert und simuliert.
- `sim_zust_erw.slx`: Mit diesem Simulink-Kopplungsplan wird der erweiterte Zustandsregler implementiert und simuliert.
- `sim_vergleich.slx`: Mit diesem Simulink-Kopplungsplan können die einzelnen Regler verglichen werden.
- `mod_{kask,pi,zust,zust_erw}.slx`: Mit diesen Simulink-Kopplungsplänen werden die einzelnen Regelungsstrategien am realen System erprobt.

Eine detaillierte Beschreibung findet man in den jeweiligen Dateien.

Typisches Vorgehen:

Zu Beginn des Praktikums ist das Skript `parameter.m` auszuführen. Danach sind die Lücken im Teil I des Matlab-Files `entwurf.m` mit den Ergebnissen aus der Vorbereitung auszufüllen.

Zur Implementierung der unterschiedlichen Regler müssen jeweils die entsprechenden Reglerparameter eingegeben werden und ggf. Simulink Kopplungspläne ergänzt werden.

Danach vergleichen Sie alle entworfenen Regler.

Schlussendlich können dann die einzelnen Regler über die jeweiligen Modelle `mod_{kask,pi,zust,zust_erw}.slx` mit Hilfe von QUARC am realen System erprobt werden.

Bitte beachten Sie die Hinweise in den einzelnen Matlab-Skripten und Simulink-Kopplungsplänen. Änderungen sind nur an den gekennzeichneten Stellen vorzunehmen. Bitte verändern Sie auch keine Variablennamen. Die Werte der Variablen (z.B. Reglerparameter) können im Matlab-Workspace festgelegt werden.

5.3 Vorbereitungsaufgaben

★ Aufgabe 5.1: Systemanalyse

Die folgenden Aufgaben sollen in Abhängigkeit von den Systemparametern a_i , A_i , g , v^0 und h_i^0 gelöst werden.

a) **Stationärer Arbeitspunkt:**

Berechnen Sie den notwendigen stationären Zufluss v^0 in Abhängigkeit der Systemparameter, so dass bei stationärem Verhalten (Ruhelage) der Füllstand im untersten Behälter h_2^0 beträgt.

b) **Stationäre Höhen:**

Berechnen Sie die stationäre Höhe h_1^0 in Abhängigkeit von v^0 .

c) **Stationäre Spannung:**

Das Übertragungsverhalten des Motors vom Eingang Spannung u_{pumpe} auf den Ausgang Pumpenstrom v kann mittels der Pumpenkonstante k über die Beziehung $v = k \cdot u_{pumpe}$ modelliert werden. Berechnen Sie die notwendige stationäre Spannung u_{pumpe}^0 in Abhängigkeit des berechneten stationären Pumpenstroms v_0 , so dass bei stationärem Verhalten der Füllstand im untersten Behälter h_2^0 beträgt.

d) **Lineares Modell:**

Linearisieren Sie das nichtlineare Modell (5.1):

$$\begin{aligned}\dot{x} &= Ax + bu \\ y &= c^T x.\end{aligned}\tag{5.2}$$

Verwenden Sie dabei folgende Variablen:

$$x = \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix}, \quad u, \quad y,$$

mit $u = u_{pumpe} - u_{pumpe}^0$, $\Delta h_i = h_i - h_i^0 = x_i$ und $y = h_2 - h_2^0$.

e) **Übertragungsfunktion:**

Berechnen Sie die Übertragungsfunktionen

$$\begin{aligned}G_1(s) &: u \rightarrow x_1, \\ G_2(s) &: u \rightarrow x_2, \\ G(s) &: u \rightarrow y,\end{aligned}$$

wobei $y = x_2$ ist. (Hinweis: Die Lösung kann auch in Abhängigkeit der einzelnen Einträge in den Systemmatrizen A , b angegeben werden.)

★ Aufgabe 5.2: Zustandsregelung mit I-Anteil - Theorie

In einem Standardregelkreis kann die bleibende Regelabweichung bei sprungförmigen Führungs- bzw. Störgrößenänderungen durch einen integrierenden Anteil im offenen Kreis besei-

tigt werden. Der Einsatz einer klassischen Zustandsrückführung führt zwar zu einem asymptotisch stabilen Regelkreis, das stationäre Verhalten wird jedoch aufgrund der vorhandenen Störungen im Allgemeinen unbefriedigend sein, da eine klassische Zustandsrückführung einen P-Regler darstellt. Die in Abbildung 5.4 dargestellte Regelkreisstruktur löst dieses Problem, indem ein I-Anteil in den offenen Kreis integriert wird [2].

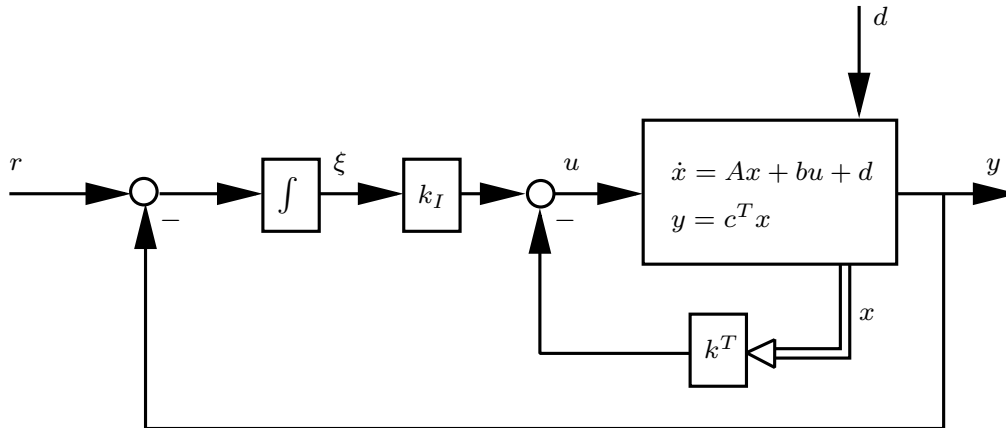


Abbildung 5.4: Zustandsrückführung mit I-Anteil.

- a) Stellen Sie das Zustandsraummodell $(A_{cl}, b_{cl}, g_{cl}, c_{cl})$ der gesamten Regelkreisstruktur in Abbildung 5.4 auf:

$$\begin{aligned} \dot{x}_e &= A_{cl}x_e + b_{cl}r + g_{cl}d \\ y &= c_{cl}^T x_e, \end{aligned} \quad (5.3)$$

mit dem $(n + 1)$ -dimensionalen Zustandsvektor $x_e = [x \ \xi]^T$.

- b) Zeigen Sie, dass die Systemmatrix A_{cl} der gesamten Regelkreisstruktur in folgender Form dargestellt werden kann:

$$A_{cl} = A_e - b_e k_e, \quad (5.4)$$

mit $A_e = \begin{bmatrix} A & 0 \\ -c^T & 0 \end{bmatrix}$, $b_e^T = [b \ 0]$, $k_e = [k^T \ -k_I]$.

5.4 Praktikumsaufgaben

Füllen Sie die Lücken im Teil I der Datei `entwurf.m` aus.

5.4.1 Ausgangsregelung mit PI-Regler

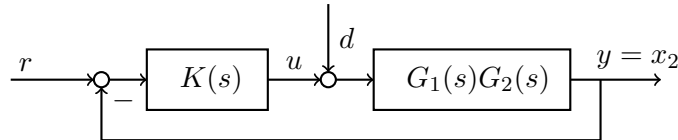


Abbildung 5.5: Ausgangsregelung (Standardregelkreis).

Aufgabe 5.3: Ausgangsregelung

- a) Entwerfen Sie eine PI-Ausgangsregelung $K(s)$ (Abbildung 5.5) mittels Loop-shaping, so dass die Phasenreserve ca. $\phi_R = 40^\circ$ beträgt und die Durchtrittsfrequenz bei $\omega_D = 0.134 \text{ rad s}^{-1}$ liegt.

Vorgehen:

In der Datei `entwurf.m` können im Teil II verschiedene Bodediagramme angezeigt werden. Je nachdem, welches Bodediagramm benötigt wird, kann die Anzeige des Diagramms über die Variable `bode(x)=1` ein- oder ausgeschaltet werden.

- b) Vervollständigen Sie den Regler im Simulinkplan `sim_pi.slx` und verknüpfen Sie die richtigen Werte mit der Anzeige (Scope). (Hinweis: Es muss die Linearisierung berücksichtigt werden, vgl. dazu Handout Regelkreisstrukturen, Appendix A)
Untersuchen Sie das Führungsverhalten (Sprungantwort) und die Störempfindlichkeit (konstante Störung d) des Regelkreises, indem Sie den Simulinkplan ausführen.
- c) Zeichnen Sie einen Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d (Abbildung 5.6).

5.4.2 Kaskadenregelung

Theorie: Entwurf einer Kaskadenregelung [3]

- (1) Entwurf des inneren Reglers $K_1(s)$. Ziel des Entwurfs ist es, die Störung d so weit auszuregeln, dass sie nur noch einen kleinen Einfluss auf den äußeren Kreis hat.
- (2) Zusammenfassung des inneren Kreises zu einem Block mit der Eingangsgröße w , dem Störeingang d sowie der Hilfsausgangsgröße x_1 . Dieser Block bildet zusammen mit $G_2(s)$ die Regelstrecke für den äußeren Regler $K_2(s)$.
- (3) Entwurf des äußeren Reglers. Der äußere Kreis wird so entworfen, dass die Regelgröße y ein gutes Führungsverhalten bezüglich der Führungsgröße r besitzt.

In der Datei `entwurf.m` können im Teil II Bodediagramme für die innere und für die äußere

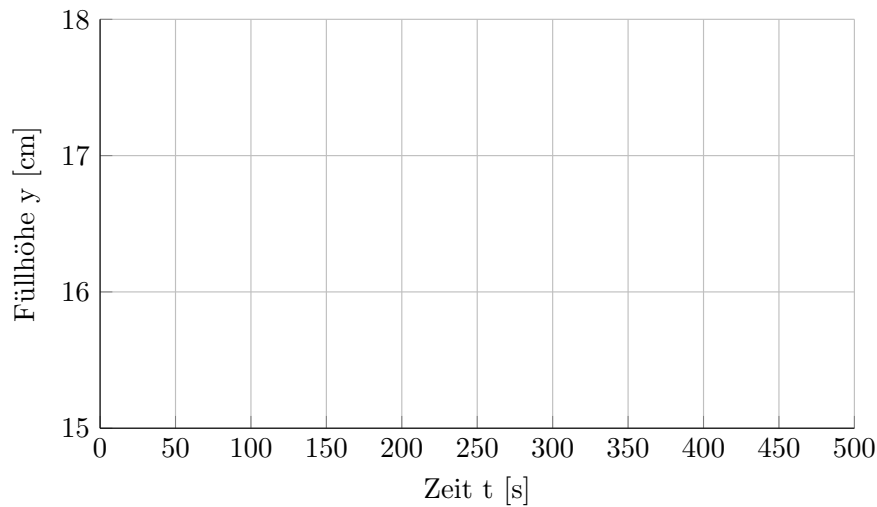


Abbildung 5.6: Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d .

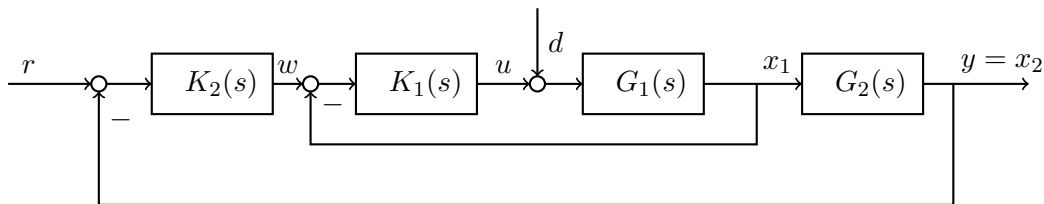


Abbildung 5.7: Kaskadenregelung.

re Strecke angezeigt werden. Je nachdem, welches Bodediagramm benötigt wird, kann die Anzeige des Diagramms über die Variable `bode(x)=1` ein- oder ausgeschaltet werden.

Aufgabe 5.4: Entwurf einer Kaskadenregelung

- Entwerfen Sie mittels Loopshaping eine Kaskadenregelung $K_1(s), K_2(s)$ (Abbildung 5.7), so dass eine deutliche Verbesserung bzgl. Führungsverhalten und Störempfindlichkeit gegenüber dem Ausgangsregler $K(s)$ zu erkennen ist.
(Hinweis: Überlegen Sie, wo die Werte der Durchtrittsfrequenzen der entsprechenden Kreise liegen sollten. Der innere Kreis soll nur die Störung auf einen Tank ausregeln, der Äußere soll die ganze Strecke stabilisieren. (Wie müssten dementsprechend die Zeitkonstanten sein?) Probieren Sie hier einfach verschiedene Werte aus.)
- Vervollständigen Sie den Regler im Simulinkplan `sim_kaskade.slx` und verknüpfen Sie die richtigen Werte mit der Anzeige (Scope). (Hinweis: Auch hier muss die Linearisierung berücksichtigt werden).
Nun untersuchen Sie das Führungsverhalten (Sprungantwort) und die Störempfindlichkeit (konstante Störung d) des Regelkreises.
- Zeichnen Sie einen Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d (Abbildung 5.8).

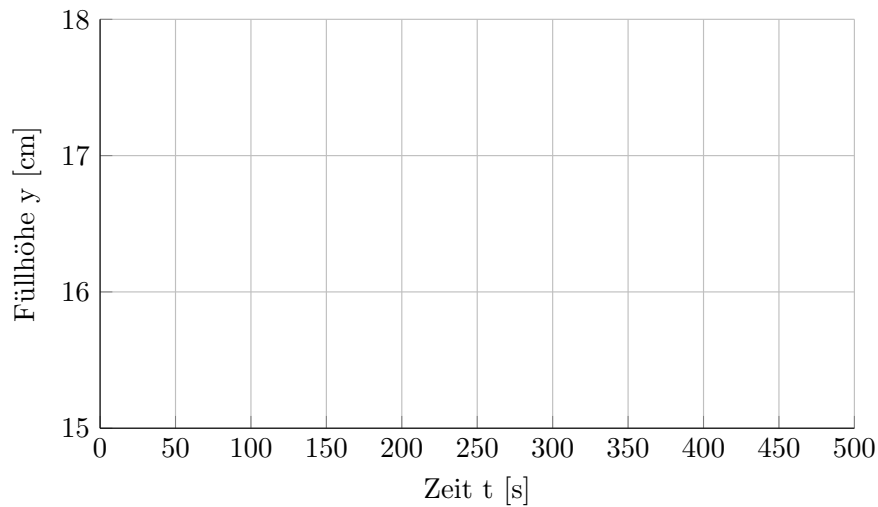


Abbildung 5.8: Zeitverlauf der Füllhöhe x_2 bei einer sprunghaftigen Änderung des Eingangs r bzw. der Störung d .

5.4.3 Zustandsregelung

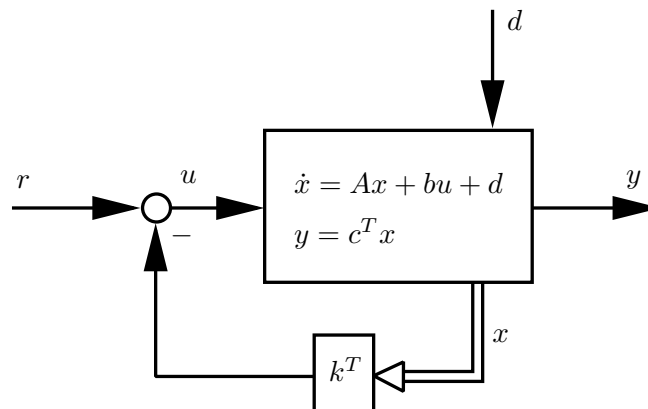


Abbildung 5.9: Zustandsrückführung.

Aufgabe 5.5: Zustandsregelung

- a) Entwerfen Sie eine Zustandsrückführung $u = -k^T x + r$ (Abbildung 5.9). Die Pole des geschlossenen Kreises sollen dabei bei $p_1 = -0.2 \frac{\text{rad}}{\text{s}}$, $p_2 = -0.22 \frac{\text{rad}}{\text{s}}$ liegen.

Vorgehen:

Geben Sie in der Datei `entwurf.m` die fehlenden Pole ein und berechnen Sie die Parameter mit dem `place`-Befehl.

- b) Untersuchen Sie das Führungsverhalten (Sprungantwort) und die Stör-empfindlichkeit (konstante Störung d) des Regelkreises mit dem Simulinkplan `sim_zust.slx`. Berücksichtigen Sie auch hier die Linearisierung.

- c) Zeichnen Sie einen Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d (Abbildung 5.10).
- d) Welche Maßnahme kennen Sie um das Führungsverhalten zu verbessern?

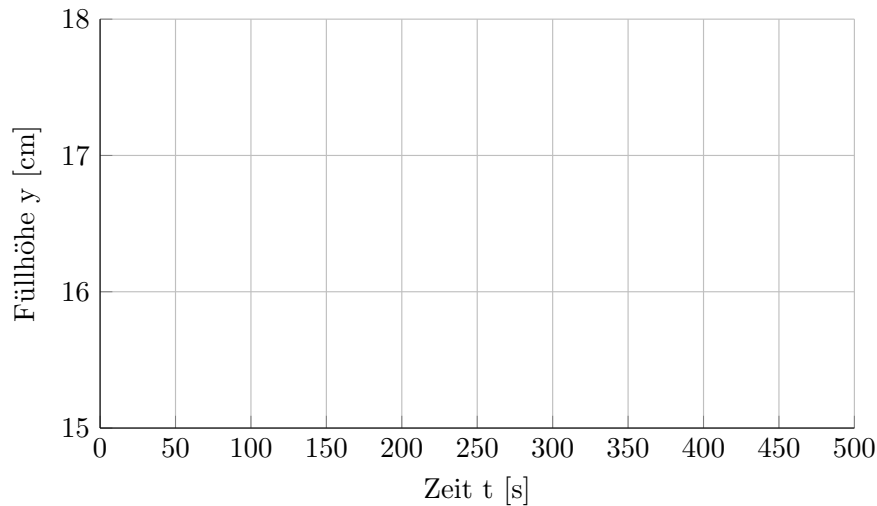


Abbildung 5.10: Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d .

5.4.4 Zustandsregelung mit I-Anteil

Aus Gleichung (5.4) der Vorbereitungsaufgabe 2 (b) ist ersichtlich, dass der Entwurf der Zustandsrückführung mit I-Anteil zurückgeführt wird auf das ursprüngliche Problem der klassischen Zustandsrückführung. In anderen Worten, der Entwurf der Zustandsrückführung mit I-Anteil erfolgt durch Polvorgabe auf das erweiterte Zustandsraummodell (A_e, b_e) .

Aufgabe 5.6: Zustandsregelung mit I-Anteil

- a) Entwerfen Sie eine Zustandsrückführung mit I-Anteil $u = -k_e^T x_e$ (Abbildung 5.4). Die Pole des erweiterten geschlossenen Kreises sollen dabei wie folgt gewählt werden:
 $p_1 = -0.2 \frac{\text{rad}}{\text{s}}, p_2 = -0.22 \frac{\text{rad}}{\text{s}}, p_3 = -0.14 \frac{\text{rad}}{\text{s}}$.

Vorgehen:

Geben Sie in der Datei `entwurf.m` die erweiterten Systemmatrizen und Pole ein und berechnen Sie die Parameter mit dem `place`-Befehl.

- b) Untersuchen Sie das Führungsverhalten (Sprungantwort) und die Störunterdrückung (konstante Störung d) des Regelkreises mit dem Simulinkplan `sim_zust_erw.slx`. Ergänzen Sie hierzu den Regler unter Berücksichtigung der Linearisierung.
- c) Zeichnen Sie einen Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d (Abbildung 5.11).

Aufgabe 5.7: Vergleich der entworfenen Regler

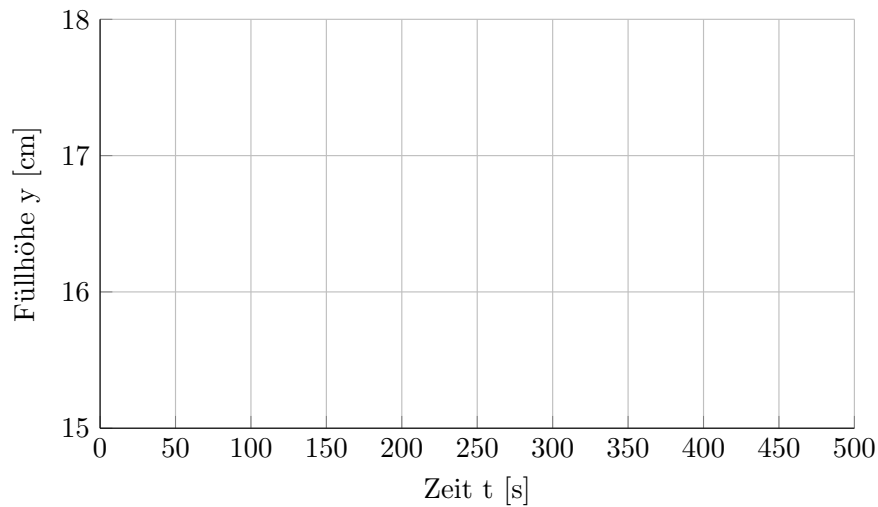


Abbildung 5.11: Zeitverlauf der Füllhöhe x_2 bei einer sprungförmigen Änderung des Eingangs r bzw. der Störung d .

Vergleichen Sie die Ergebnisse der verschiedenen Regler miteinander. Verwenden Sie hierzu die Datei `sim_vergleich.slx`.

Aufgabe 5.8: Erprobung am realen System

Die in den vorhergehenden Aufgaben entworfenen Regler sollen nun am realen System erprobt werden. Dies geschieht mit den bereitgestellten Modellen und dem QUARC Interface.

Öffnen Sie zunächst die gewünschte Simu-link-Datei, z.B. `mod_pi.slx` in Matlab.

Nun muss die Datei `mod_pi.slx` kompiliert werden. Dies geschieht mit der Tastenkombination `strg + b`, wenn das Simulink-Fenster geöffnet ist.

Danach kann nun im Simulink Modell **Connect Target** ausgewählt werden und das Experiment gestartet werden.

Erproben Sie nun die von Ihnen entworfenen Regler. Notieren Sie Ihre Erkenntnisse und Beobachtungen und interpretieren Sie die Resultate.

5.5 Appendix

5.5.1 Parameter/Arbeitspunkte

Die Parameter für das Modell (5.1) lauten:

A_1	$[cm^2]$	15.518
A_2	$[cm^2]$	15.518
a_1	$[cm^2]$	0.178
a_2	$[cm^2]$	0.178
h_2^0	$[cm]$	15
g	$[cm/s^2]$	981
k	$[cm^3/Vs]$	4.3

6

Versuch: Identifikation und Regelung eines Gleichstrom-Elektromotors

Lernziele

- Validation von physikalischen Modellen durch Experimente
- Identifikation von Modellen rein aus Experimenten mithilfe der System Identification Toolbox in MATLAB
- Verwendung eines PI-Reglers mit Anti-Windup Struktur
- Entwurf einer Zustandsrückführung mit Beobachter
- Erstellung von Referenztrajektorien für Zustandsrückführungen

Der Schwerpunkt des Praktikumsversuchs liegt bei der Modellidentifikation. Bezüglich der systematischen Vorgehensweise aus Kapitel 2 sind folgende Punkte relevant:

- Punkt b): Aufgabe 6.2 bis 6.20
- Punkt c): Aufgabe 6.1, 6.21 und 6.25
- Punkte i) bis l): Aufgaben 6.22 bis 6.29

Notwendige Vorkenntnisse:

- Zustandsrückführung
- Beobachterentwurf
- PI-Regelung

Notwendige Vorleistungen:

- Die mit ★ gekennzeichneten Aufgaben sind vor dem Praktikum zu bearbeiten und am Tag der Versuchsdurchführung mitzubringen.

6.1 Einleitung

Elektromotoren finden heutzutage Anwendung in vielen verschiedenen Bereichen. Sie werden in nahezu allen Größen eingesetzt und ihre praktische Bedeutung wird in der näheren Zukunft noch zunehmen. Vor allem im Zuge der Ölknappheit gewinnen hybride Antriebe, in denen Elektromotoren sowohl als Antrieb als auch zur Energierückgewinnung eingesetzt werden, an Bedeutung. Deshalb spielt die Regelung von Gleichstrommotoren eine wichtige Rolle. Desweiteren sind Elektromotoren seit jeher typische Beispiele in der angewandten Regelungstechnik und somit fest in der Lehre und Ausbildung von Studenten der Ingenieurwissenschaften verankert und zugleich immer noch Gegenstand aktueller Forschung.

Ziel dieses Praktikumversuches ist es die Vorgehensweise bei der mathematischen Modellierung und Identifikation eines Prozesses zu vermitteln. Beispielhaft wird dies für einen Elektromotor durchgeführt. Nachdem ein Modell des Motors identifiziert wurde, werden mithilfe dieses Modells verschiedene Regler entworfen, untersucht und getestet. Zur Validierung am tatsächlichen System steht in diesem Versuch als Elektromotor der „Precision Modular Servo“ (PMS) der Firma „Feedback Instruments Ltd.“ zur Verfügung.

Im ersten Schritt werden verschiedene Methoden untersucht, mit denen ein Modell für den PMS ermittelt werden kann. Als erstes wird die Modellierung unter Berücksichtigung physikalischer Gesetzmäßigkeiten und eine nachfolgende Parameterschätzung verwendet. Nachfolgend wird mit den in MATLAB implementierten Funktionen zur Systemidentifikation (System Identification Toolbox) ein Modell identifiziert. Die Ergebnisse die beide Methoden liefern werden dann verglichen.

In den darauffolgenden Abschnitten werden unter Verwendung der ermittelten Modelle verschiedene Regler (PI- und Zustandsregler) entworfen und untersucht. In diesem Zusammenhang werden verschiedene in der Vorlesung kennengelernte Effekte, wie Windup und inkonsistente Referenzsignale, aufgezeigt und verdeutlicht.

6.1.1 Sicherheitshinweise

Lesen Sie diese Sicherheitshinweise bitte aufmerksam durch und beachten Sie diese während und nach jedem Experiment!

- Der Motor und vor allem das Schwungrad müssen immer frei sein und es dürfen sich keine Teile darin verfangen können.
- Bevor Sie den Motor einschalten und nachdem ein Experiment beendet wurde, drehen Sie bitte immer den unteren Drehknopf an der „Attenuator Unit“ auf 0.
- Sollte sich nach Einschalten der Hauptstromversorgung aller Teile der Motor trotzdem bewegen, schalten Sie die Hauptstromversorgung bitte wieder aus und wenden Sie sich an einen Betreuer.
- Stellen Sie sicher, dass der Magnet nicht über das Schwungrad geklappt ist.
- Sollte der Motor eigenartige Geräusche machen, beenden Sie bitte sofort das Experiment durch Drücken des „Stop-Buttons“ im Simulink-Modell oder schalten Sie die

Hauptstromversorgung aus und überprüfen Sie ihre Parameter. Wenden Sie sich hierfür ggf. an einen Betreuer.

6.2 Systembeschreibung

6.2.1 Versuchsaufbau

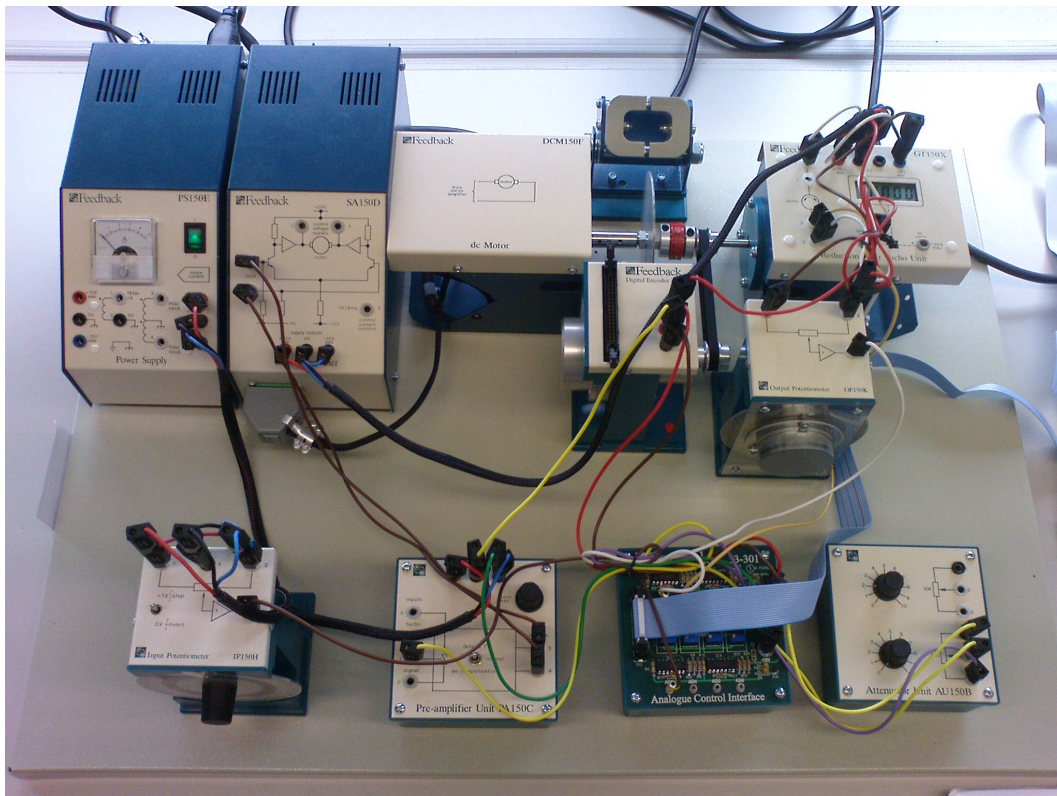


Abbildung 6.1: Precision Modular Servo.

In Abbildung 6.1 ist der PMS zu sehen, mit dem im Verlaufe des Praktikums gearbeitet werden soll. Das Ziel ist es die Geschwindigkeit bzw. die Position des Schwungrades zu regeln.

Der PMS ist mit einer „PC Lab Card“ an den PC angeschlossen. Die Stellsignale werden von dem DA-Wandler in analoge Signale gewandelt und durch den Vorverstärker und den Servo Verstärker entsprechend verstärkt. Da der Motor in einem Bereich von +24 V bis –24 V betrieben werden kann und die Verstärker das Signal insgesamt mit dem Faktor 9,6 verstärken, ist die maximal zulässige Steuerspannung in den Simulink Modellen $\pm 2,5$ V. Der Motor ist mechanisch an das Tachometer, das Ausgangs-Potentiometer und den digitalen Positionssensor angeschlossen. Das Tachometer sendet ein analoges geschwindigkeitsproportionales Signal und das Ausgangs-Potentiometer ein analoges positionsproportionales Signal. Beide Signale werden durch den AD-Wandler in digitale Signale umgewandelt. Der digitale Positionssensor liefert ein inkrementelles Signal das jeweils einer kleinen Winkeländerung entspricht. Es ist

somit nicht möglich mit dem Sensor eine absolute Position zu bestimmen. Aus diesem Grund sollte der Motor vor jedem Versuch, durch drehen am Schwungrad, manuell auf 0 rad gefahren werden. Als Störgröße kann der Magnet über das Schwungrad geklappt werden. Dieser induziert ein elektromagnetisches Feld und bremst durch das erzeugte Moment das Schwungrad des Motors ab.

6.2.2 Lineares Modell des PMS

Da unter Verwendung eines Modells das Verhalten der Regelstrecke einfach und ohne Risiko untersucht werden kann soll zuerst ein Modell des PMS hergeleitet werden. Dies erlaubt im weiteren Verlauf auch die Verwendung modellbasierter Reglerentwurfsverfahren. Im folgenden werden verschiedene Verfahren zur Bestimmung eines Modells für PMS vorgestellt, implementiert und getestet.

Zuerst soll mit Hilfe physikalischer Gesetzmäßigkeiten ein physikalisches Modell hergeleitet werden. Der PMS kann als gekoppelter elektrischer und mechanischer Kreis betrachtet werden, siehe Abbildung 6.2.

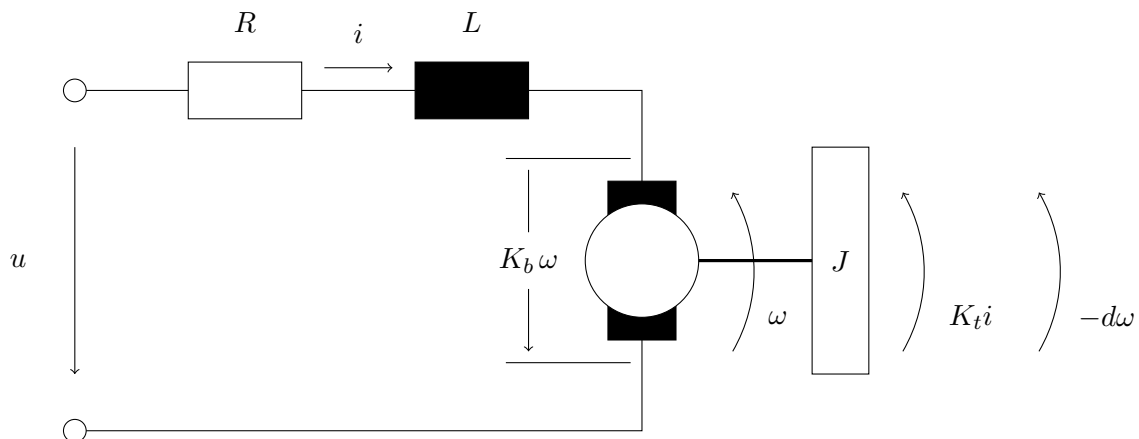


Abbildung 6.2: Modell des PMS.

Hierbei ist u die Spannung, i der Motorstrom, R der Widerstand und L die Induktivität des Motorstromkreises. Desweiteren sind ω die Drehgeschwindigkeit des Motors, K_b die elektromotorische Kraftkonstante, J das Trägheitsmoment des Motors bzgl. seiner Drehachse, K_t eine Maschinenkonstante und d der Reibkoeffizient der angenommenen drehzahlabhängigen viskosen Reibung.

Mithilfe der Kirchhoff'schen Gesetze erhält man für den Stromkreis:

$$u = L \frac{di}{dt} + Ri + K_b \omega$$

und mittels der Newton'schen Gesetze ergibt sich:

$$J\dot{\omega} = -d\omega + K_t i.$$

Desweiteren hängt die Position direkt von der Drehzahl ab,

$$\dot{\theta} = \omega.$$

Damit erhält man das Zustandsraummodell dritter Ordnung,

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d}{J} & \frac{K_t}{J} \\ 0 & -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 9.6 \cdot \frac{1}{L} \end{bmatrix} u, \quad \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x} \\ \text{mit } \mathbf{x} &= [\theta \quad \omega \quad i]^T.\end{aligned}\tag{6.1}$$

Der Faktor 9.6 entspricht hierbei der oben erwähnten Verstärkung durch die Leistungsverstärker.

Aus diesem Zustandsraummodell lassen sich nun die zwei Übertragungsfunktionen, von der Spannung u zur Drehgeschwindigkeit ω ,

$$G_\omega(s) = \frac{\omega(s)}{U(s)} = \frac{9.6 \cdot \frac{K_t}{JL}}{s^2 + \left(\frac{d}{J} + \frac{R}{L}\right) \cdot s + \left(\frac{dR}{JL} + \frac{K_b K_t}{JL}\right)},\tag{6.2}$$

und von der Spannung u zur Position θ ,

$$G_\theta(s) = \frac{\theta(s)}{U(s)} = \frac{9.6 \cdot \frac{K_t}{JL}}{s^3 + \left(\frac{d}{J} + \frac{R}{L}\right) \cdot s^2 + \left(\frac{dR}{JL} + \frac{K_b K_t}{JL}\right) \cdot s},\tag{6.3}$$

bestimmen. Mithilfe der Vorzeichenbedingung können beide Übertragungsfunktionen auf Stabilität geprüft werden. Für die Übertragungsfunktion $G_\omega(s)$, welche die Geschwindigkeitsdynamik beschreibt, folgt damit asymptotische Stabilität und für die Übertragungsfunktion auf die Position $G_\theta(s)$ folgt Grenzstabilität. Letzteres ergibt sich, da die asymptotisch stabile Übertragungsfunktion $G_\omega(s)$ nur mit einem Pol bei 0 bzw. einem Integrator erweitert wurde. Die Pole sind für die Übertragungsfunktion aus Gleichung (6.2) somit

$$-\frac{\frac{d}{J} + \frac{R}{L}}{2} \pm \sqrt{\frac{d^2}{4J^2} + \frac{R^2}{4L^2} - \frac{dR}{2JL} - \frac{K_b K_t}{JL}}$$

beziehungsweise für Gleichung (6.3):

$$-\frac{\frac{d}{J} + \frac{R}{L}}{2} \pm \sqrt{\frac{d^2}{4J^2} + \frac{R^2}{4L^2} - \frac{dR}{2JL} - \frac{K_b K_t}{JL}} \quad \text{und} \quad 0.$$

Die vom Hersteller gegebenen physikalischen Parameter sind in Tabelle 6.1 angegeben.

Mit diesen Parametern lassen sich die Pole der Übertragungsfunktionen bestimmen.

★ Aufgabe 6.1:

Bestimmen Sie die Pole der Übertragungsfunktionen mit diesen Parametern.

Nun ist es an der Zeit, das erhaltene Modell zu validieren. Hierzu wird das genaue Zeitverhalten des PMS untersucht. Wir werden uns im Zuge der Modellierung und Identifikation auf die Geschwindigkeitsübertragungsfunktion konzentrieren, da hieraus leicht die Positionsübertragungsfunktion bestimmt werden kann.

Tabelle 6.1: Parameter des Herstellers.

J – Trägheitsmoment	$140 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$
K_t – Maschinenkonstante	0.052 Nm/A
K_b – elektromotorische Kraftkonstante	0.057 Vs/rad
d – Approximation der viskosen Reibung	$1 \cdot 10^{-6} \text{ Nms/rad}$
L – Induktivität	2.5 mH
R – Widerstand	$2.5 \text{ } \Omega$

6.3 Schätzung der physikalischen Modellparameter

6.3.1 Bewertung des Modells mit den gegebenen Parametern

Um das Modell mit den Parametern des Herstellers zu validieren muss ein Versuch an der realen Strecke durchgeführt werden. Danach wird das Modell des PMS mit demselben Eingang beaufschlagt wie der PMS und die gemessene und simulierte Geschwindigkeit werden verglichen. Als Eingangssignal wird eine Summe aus vier Sinussignalen verwendet.

Öffnen Sie hierzu als erstes das Simulink-Modell „VelocityModel_Ident.mdl“ im Verzeichnis „Identifikation“.

Aufgabe 6.2:

Erstellen Sie an der dafür vorgesehenen Stelle das Eingangssignal. Dieses soll aus 4 Sinussignalen bestehen. Die Sinussignale mit den Frequenzen 1 1/s und 10 1/s haben jeweils eine Amplitude von 0.2. Die Amplituden der Sinussignale mit den Frequenzen 2 1/s und 6.7 1/s sind jeweils 0.4. Zusätzlich sollten Sie noch ein Rauschen mit Mittelwert 0 und Varianz 0.02 auf das Signal addieren. Verwenden Sie hierzu den Block „RandomNumber“ und stellen sie eine Abtastzeit (sample-time) von 0.1 s ein.

Nun wird ein Experiment mit einer Gesamtlänge von 20 Sekunden durchgeführt.

Aufgabe 6.3:

Durch einen Klick auf „Tools - Real-Time Workshop - Build Model“ bzw. „STRG+B“ wird das Modell erstellt. Verbinden Sie bitte das Modell mit dem Motor durch einen Klick auf den „connect to target“-Button neben dem Simulink-Start Button. Stellen Sie sicher, dass der Drehknopf der „Attenuator Unit“ auf 0 gedreht ist und schalten Sie dann den Motor ein. Er sollte sich nicht drehen. Drehen Sie nun den Drehknopf der „Attenuator Unit“ auf 2. Der Motor sollte sich immer noch nicht bewegen. Klicken Sie in Simulink auf den Start Button. Der Computer bzw. der Motor wird nun das Experiment durchführen. Schalten Sie den Motor aus, sobald MATLAB das Experiment beendet hat und drehen Sie den Drehknopf der „Attenuator Unit“ auf 0.

Nachdem das Experiment beendet wurde sollte sich im MATLAB Workspace eine neue Variable namens „velocity_measured“ befinden. Hier sind das Eingangssignal und das gemessene

Ausgangssignal abgelegt. Eine Messung der Signale wurde alle 0.001 s durchgeführt (Abtastfrequenz = $1/(0.001s)$).

Aufgabe 6.4:

Speichern Sie bitte die gemessenen Daten in dem aktuellen Verzeichnis in eine Datei mit Namen „simout_sin“. Lassen Sie sich Ein- und Ausgang plotten.

Aufgabe 6.5:

Öffnen Sie nun das MATLAB Skript „A.1.Validation_des_Modells.m“. Vervollständigen Sie notwendige Teile und führen Sie es aus.

Ihnen wird nun der Fit im MATLAB Befehlsfenster angezeigt. Der Fit wird durch $\text{fit} = 100\% \cdot \left(1 - \frac{|y_h - y|}{|y_h - \text{mean}(y)|}\right)$ bestimmt, wobei y_h der durch das Modell bestimmte Ausgang, y der gemessene Ausgang und $\text{mean}(y)$ der Mittelwert des gemessenen Ausgangs ist. Hierbei würde 100% einer exakten Abbildung der Signale bedeuten. In dem Plot-Fenster werden die Zeitsignale angezeigt.

Aufgabe 6.6:

Was können Sie für eine Aussage über das Verhalten des Modells treffen?

6.3.2 Optimierung der gegebenen Parameter

Bisher wurden die vom Hersteller angegebenen Parameter verwendet. In diesem Abschnitt wird nun gezeigt, wie man diese Parameter verbessern kann um eine bessere Übereinstimmung von Messung und Vorhersage zu erreichen.

Hierzu wird die MATLAB Optimization Toolbox bzw. der `lsqcurvefit` Algorithmus verwendet. Dieser Algorithmus löst ein nichtlineares Optimierungsproblem im Sinne des kleinsten quadratischen Fehlers. Ziel des Algorithmus ist es den quadratischen Fehler zwischen dem Ausgangssignal des Modells und einem am Motor gemessenen Ausgangssignal zu minimieren. Hierzu werden dem Algorithmus neben dem Modell und einem Startwert für die Parameter auch ein gemessenes Signal, bestehend aus Ein- und Ausgang, übergeben.

Der Algorithmus bestimmt als erstes den momentanen quadratischen Fehler in Abhängigkeit der Parameter. Dann wird unter Verwendung der Finiten-Differenzen Methode die Richtung im Paramterraum ermittelt in welche der quadratische Fehler am stärksten abnimmt. Basierend auf dieser Richtungsinformation und des Betrags des Gradienten wird ein Parameterupdate vorgeschlagen. Diese Prozedur wird so oft wiederholt, bis eine untere Schranke in der Verbesserung des quadratischen Fehlers erreicht wurde. Weitere Informationen finden Sie in der MATLAB-Hilfe.

Aufgabe 6.7:

Öffnen Sie nun das MATLAB Skript „A.2.Optimierung_der_Parameter.m“ und führen Sie es aus. Im MATLAB Hauptfenster können die einzelnen Iterationsschritte verfolgt werden. Sobald die Optimierung abgeschlossen ist, können Sie das Ergebnis sehen. Das Skript speichert die Parameter in einer Datei „optimized_parameters.mat“ im Ordner oberhalb des

Ordners „Identifikation“ ab.

Das Modell mit den neuen Parametern erreicht einen wesentlich höheren Fit als das Modell mit den gegebenen Parametern. Desweiteren verhält sich das Modell nun auch qualitativ wie der PMS.

Aufgabe 6.8:

Notieren Sie sich bitte die neuen Parameter, die durch die Optimierung erhalten wurden, in Tabelle 6.2. Beachten Sie insbesondere die Zehner-Potenzen.

Tabelle 6.2: Optimierte Parameter.

J – Trägheitsmoment	$\text{kg}\cdot\text{m}^2$
K_t – Maschinenkonstante	Nm/A
K_b – elektromotorische Kraftkonstante	Vs/rad
d – Approximation der viskosen Reibung	Nms/rad
L – Induktivität	H
R – Widerstand	Ω

Notieren Sie sich hier bitte die Übertragungsfunktion des Modells mit den optimierten Parametern aus Tabelle 6.2.

$$G_{opt}(s) =$$

Aufgabe 6.9:

Wo liegen die Pole der Übertragungsfunktion? Was fällt ihnen auf?

6.4 Identifikation

Nachdem nun ein physikalisches Modell unter Verwendung von physikalischen Gesetzmäßigkeiten hergeleitet wurde, soll nun ein Modell mithilfe von MATLAB identifiziert werden. Bei der Systemidentifikation von linearen Systemen wird keine Modellstruktur sondern nur eine Modellordnung vorgegeben. Desweiteren wird kein Vorwissen (physikalische Zusammenhänge) verwendet.

6.4.1 Grundlagen der Systemidentifikation

Die Systemidentifikation ist ein wichtiger Bereich der Systemtheorie. In diesem Praktikumsdruck kann nur kurz auf die wichtigsten Grundlagen eingegangen werden und ein grober

Eindruck vermittelt werden, wie die Identifikationsalgorithmen arbeiten. Ausführliche Erklärungen finden Sie z.B. in [4].

Zeitdiskrete Darstellung von dynamischen Systemen

Die meisten Identifikationsverfahren arbeiten mit zeitdiskreten Signalen. Es wird also angenommen, dass das Eingangssignal $u(t)$ und das Ausgangssignal $y(t)$ mit einer konstanten Abtastperiode T_s abgetastet werden, d.h. sie stehen nur zu den Zeitpunkten $t = 0 \cdot T_s, 1 \cdot T_s, 2 \cdot T_s, \dots$ zur Verfügung. Nun führt man eine neue Zeitvariable $n \in \mathbb{N}$ ein und erhält mit $t = n \cdot T_s$ die zeitdiskreten Signale

$$u(n) = u(nT_s) \quad \text{und} \quad y(n) = y(nT_s), \quad n \in \mathbb{N}.$$

In der zeitkontinuierlichen Schreibweise können Systeme durch *Differenzialgleichungen* beschrieben werden. Entsprechend dazu gibt es in der zeitdiskreten Darstellung *Differenzengleichungen*. Damit kann das Ausgangssignal wie folgt bestimmt werden:

$$\begin{aligned} y(n) = & -a_1 y(n-1) - \dots - a_{n_a} y(n-n_a) \\ & + b_0 u(n-n_k) + \dots + b_{n_b-1} u(n-n_k-n_b+1), \quad n \in \mathbb{N} \end{aligned} \quad (6.4)$$

Das heißt der aktuelle Ausgang $y(n)$ hängt von den letzten n_a Ausgangswerten und von den n_k bis $n_k + n_b$ Zeitschritte zurückliegenden Eingangswerten $u(n)$ ab. Mit $n_k \geq 0$ kann also angegeben werden, wie groß die Verzögerung vom Eingang zum Ausgang des Systems mindestens sein soll.

Nun wird der Operator q^{-1} eingeführt, der ein zeitdiskretes Signal um einen Abtastschritt verschiebt:

$$\begin{aligned} q^{-1}y(n) &= y(n-1), \\ q^{-k}y(n) &= y(n-k). \end{aligned}$$

Damit lässt sich (6.4) wie folgt darstellen:

$$\begin{aligned} y(n) = & -a_1 q^{-1}y(n) - \dots - a_{n_a} q^{-n_a}y(n) \\ & + b_0 q^{-n_k}u(n) + \dots + b_{n_b-1} q^{-n_k-n_b+1}u(n), \end{aligned}$$

oder kurz

$$A(q) \cdot y(n) = B(q) \cdot u(n)$$

mit den Polynomen in q^{-1}

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\ B(q) &= b_0 q^{-n_k} + \dots + b_{n_b-1} q^{-n_k-n_b+1}. \end{aligned} \quad (6.5)$$

Nachdem wir das System durch eine Differenzengleichung im Zeitbereich beschrieben haben, möchten wir es nun im *Frequenzbereich* mithilfe einer *Übertragungsfunktion* darstellen. Die

Übertragungsfunktion $G_k(s)$ (k für kontinuierlich) von linearen zeitinvarianten Systemen in der zeitkontinuierlichen Betrachtung ist

$$G_k(s) = \frac{Y(s)}{U(s)},$$

wobei $U(s)$ und $Y(s)$ die Laplace-Transformierten der zeitkontinuierlichen Ein- und Ausgangssignale sind. Im zeitdiskreten Fall gilt ganz analog für die Übertragungsfunktion $G_d(z)$ (d für diskret)

$$G_d(z) = \frac{Y(z)}{U(z)}. \quad (6.6)$$

Hierin ist $U(z)$ die z -Transformierte des zeitdiskreten Eingangs $u(n)$ und $Y(z)$ ist die z -Transformierte des zeitdiskreten Ausgangs $y(n)$. Die z -Transformation ist das Pendant zur Laplace-Transformation für zeitdiskrete Signale. $G_d(z)$ entspricht in diesem Fall genau

$$G_d(z) = G_d(q) = \frac{B(q)}{A(q)}$$

mit $A(q)$ und $B(q)$ aus Gleichung (6.5). Das heißt der Verschiebeoperator q entspricht dem komplexen z der z -Transformation. Allerdings sind die zeitdiskrete und die zeitkontinuierliche Übertragungsfunktion nicht identisch ($G_d(z) \neq G_k(z)$).

Die Stabilität von linearen zeitinvarianten Systemen kann durch die Lage der Pole der Übertragungsfunktion untersucht werden. Während Pole und Nullstellen bei kontinuierlichen Systemen in der komplexen s -Ebene dargestellt werden, trägt man sie bei zeitdiskreten Systemen in der komplexen z -Ebene auf. Der stabile Bereich ist hier das Innere des Einheitskreises.

Der Vollständigkeit halber sei ergänzt, dass aufgrund der Abtastung ein Teil des Frequenzbereiches der ursprünglichen Signale verloren geht. Im Zeitdiskreten können nur noch Frequenzen unterhalb der Nyquist-Frequenz $f_{Nyquist} = \frac{1}{2T_s}$ dargestellt werden.

Im Folgenden verwenden wir den Verschiebeoperator q im Zeitbereich, d.h.

$$G_d(q) = \frac{y(n)}{u(n)}$$

und z für den Frequenzbereich (vergleiche Gleichung (6.6)).

An dieser Stelle wird nicht weiter auf den q -Operator und die z -Transformation eingegangen. Eine Einführung bietet jedes Buch über zeitdiskrete Regelung, wie z.B. [5].

Modelle für die Systemidentifikation

Bei der Systemidentifikation versucht man ein geeignetes Modell für einen gegebenen Datensatz $\{u(n), y(n)\}$, $0 \leq n < N$ zu finden. In der Regel lässt sich auf Grund von Störungen kein exakt passendes Modell der Form $G_d(q)$ finden. Daher wird ein Störeingang $e(n)$ als weißes Rauschen mitmodelliert. Eine sehr allgemeine Beschreibung für ein lineares zeitinvariantes System ist:

$$A(q)y(n) = \frac{B(q)}{F(q)}u(n) + \frac{C(q)}{D(q)}e(n) \quad (6.7)$$

Tabelle 6.3: Übersicht zeitdiskreter Frequenzbereichsmodelle. Die nicht benutzten Polynome sind gleich 1 zu setzen.

Modell	Kurz	Freie Polynome
Auto regressive	ARX	$A(q), B(q)$
Auto regressive moving average	ARMAX	$A(q), B(q), C(q)$
Output-error	OE	$B(q), F(q)$
Box-Jenkins	BJ	$B(q), F(q), C(q), D(q)$

mit

$$\begin{aligned}
 C(q) &= 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \\
 D(q) &= 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d} \\
 F(q) &= 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}
 \end{aligned}$$

Man unterscheidet nun die verschiedenen Frequenzbereichsmodelle dadurch, welche Polynome bestimmt werden (siehe Tabelle 6.3). Beispielsweise sehen ARX- und OE-Modelle wie folgt aus:

$$\begin{aligned}
 \text{ARX :} \quad & A(q) \cdot y(n) = B(q) \cdot u(n) + e(n) \\
 \text{OE :} \quad & y(n) = \frac{B(q)}{F(q)} \cdot u(n) + e(n).
 \end{aligned}$$

Der wesentliche Unterschied liegt also darin, wie das Rauschen auf den Eingang wirkt.

Es können auch Modelle im Zustandsraum identifiziert werden. Ein Beispiel hierfür ist die `n4sid`-Struktur:

$$\begin{aligned}
 x(n+1) &= Ax(n) + Bu(n) + Ke(n) \\
 y(n) &= Cx(n) + Du(n) + e(n) ; \quad n \in \mathbb{N}
 \end{aligned}$$

Die freien Parameter des Modells sind in diesem Fall die Matrizen A , B , C , D und K . Das weiße Rauschen wird wie zuvor mit $e(n)$ bezeichnet.

Die MATLAB Identifikationstoolbox stellt verschiedene Modellstrukturen zur Verfügung. Wir konzentrieren uns in diesem Versuch auf parametrische Modelle. Dazu zählen Zustandsraummodelle (`n4sid`) oder Modelle im Frequenzbereich (gemäß Gleichung (6.7)). Eine ausführliche Beschreibung der verschiedenen Modelle finden Sie in [4].

Identifikation

Bei der Systemidentifikation werden die Koeffizienten der oben beschriebenen Modelle so bestimmt, dass das gemessene Ein-/Ausgangsverhalten bestmöglich wiedergegeben wird, d.h. der Ausgang des Modells entspricht möglichst genau dem gemessenen Ausgang.

Da der gemessene Ausgang im allgemeinen mit Messrauschen überlagert ist kann der wahre Parameterwert jedoch meist nicht exakt bestimmt werden. Stattdessen weist jeder Modellparameter Unsicherheiten auf. Je nachdem wie gut die Schätzer sind (was mit der zu modellierenden Strecke und der Zahl N der Messwerte zusammen hängt) haben die Zufallsvariablen

für die Modellparameter eine mehr oder weniger große Varianz. Der *99%-Vertrauensbereich* (englisch *99% Confidence Interval*) gibt den Bereich an, in dem die Parameter mit einer Wahrscheinlichkeit von 99% liegen. Auch für verschiedene Modelldarstellungen wie Pol-Nullstellen-Diagramm oder Bodediagramm kann man einen 99%-Vertrauensbereich angeben. Das Diagramm des realen Systems, also die Pole und Nullstellen bzw. die Übertragungsfunktion, liegt dann mit einer Wahrscheinlichkeit von 99% in den angegebenen Grenzen. Je kleiner der 99%-Vertrauensbereich ist, desto genauer ist natürlich die Schätzung. Eine detaillierte Erläuterung hierzu finden Sie in [4].

Die Wahl der Modellstruktur und der Modellordnung, d.h. der Freiheitsgrade, sind die wesentlichen Punkte bei der Durchführung einer Identifikation. Im allgemeinen gilt: Je höher die Ordnung des Modells gewählt wird, desto genauer ist die Schätzung für den vorgegebenen Datensatz $\{u(n), y(n)\}$. Allerdings hängt das Ergebnis dann auch immer stärker vom Eingangssignal $u(n)$ ab und das Modell kann bei einer zu großen Modellordnung seine Allgemeingültigkeit verlieren, d.h. das Eingangssignal wird „mitmodelliert“.

Bei zu hoher Ordnung enthält das identifizierte Modell Pole und Nullstellen an Punkten, wo das „optimale Modell“ des realen Systems keine hatte. Beim Betrachten des Pol-Nullstellen-Diagramms eines Modells kann man solche „künstlich generierten“ Pol-Nullstellen-Paare erkennen, denn diese liegen sehr nah beisammen.

Für Zustandsraummodelle gibt es darüber hinaus eine weitere Möglichkeit um festzustellen, ob zu viele Zustände modelliert wurden. Dazu untersucht man, wie viel jeder einzelne Zustand zum Ein-/Ausgangsverhalten des Systems beiträgt. Wenn man zum Beispiel acht Zustände modelliert hat und weiß, dass ein weiterer, neunter Zustand das Modellverhalten nur unwesentlich verändert, verzichtet man auf diesen. Den Beitrag der Zustände zum Modellverhalten berechnet Matlab mithilfe der Singulärwerte bestimmter Kovarianzmatrizen. Genauereres hierzu finden Sie in [4].

Anregungssignale für die Identifikation

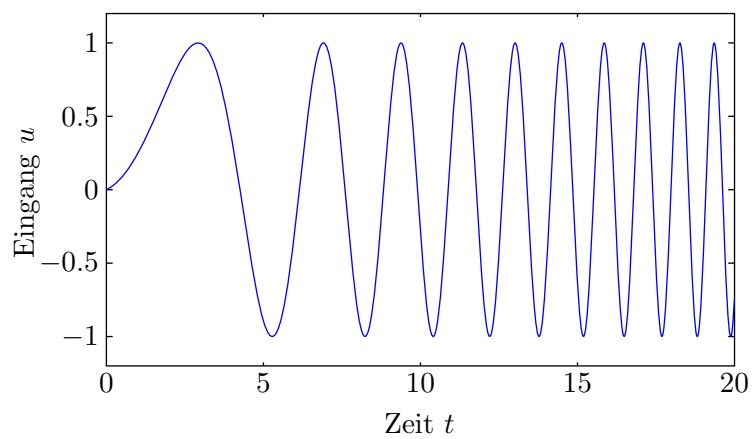
Ein wichtiger Aspekt der Systemidentifikation ist die Wahl des Anregungssignals. Es ist dabei wichtig ein möglichst breites Frequenzspektrum abzudecken. Dies kann mithilfe verschiedener Eingangssignale geschehen (siehe auch Abbildung 6.3), z.B.

Impuls, Sprung, eher für theoretische Betrachtungen hilfreich,

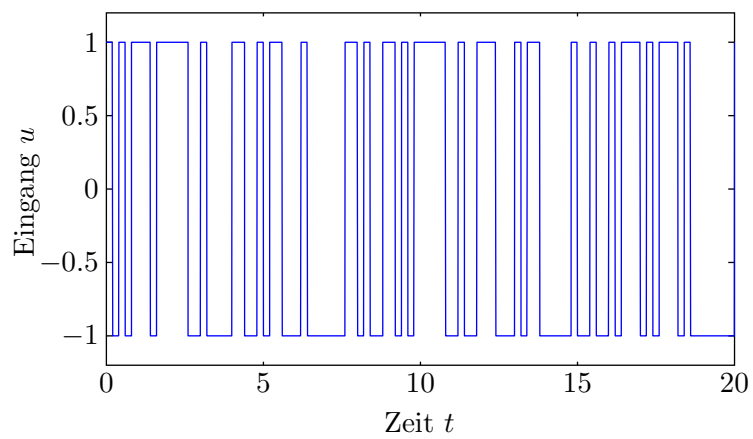
Gleitsinussignal (Sinus-Sweep), Sinusschwingung mit ständig ansteigender Frequenz, wird insbesondere zur Analyse mechanischer Systeme verwendet,

Rauschen, im Zeitdiskreten entweder als regelmäßige Sprünge verschiedener Amplitude oder als Sprung zwischen zwei Werten mit zufälliger Umschaltzeit

Mehr Informationen zur Systemanregung finden Sie in [4].



(a) Sinus-Sweep



(b) binäres Zufallssignal (RBS)

Abbildung 6.3: Verschiedene Testsignale für die Modellidentifikation.

6.4.2 Messen des Ein-/Ausgangsverhaltens

In Abschnitt 6.3.1 und Abschnitt 6.3.2 wurde der PMS bereits mit einem Signal angeregt und das Ein-/Ausgangsverhalten gemessen. Hier soll nun der PMS mit weiteren Testsignalen angeregt und die dazugehörige Geschwindigkeit gemessen werden. Das Ziel ist die Identifikation des Übertragungsverhaltens vom Eingang u auf die Geschwindigkeit ω , d.h. $\omega(s) = G_\omega(s)u(s)$.

Bevor Sie nun ein weiteres Experiment durchführen speichern Sie bitte die Variable `velocity_measured` unter dem Namen `velocity_measured_sin` ab. Dies ist wichtig um sicherzustellen, dass die Daten nicht verloren gehen und damit die einzelnen Experimente später unterscheidbar sind.

Aufgabe 6.10:

Öffnen Sie nun wieder das Modell „VelocityModelIdent.mdl“ und implementieren Sie als Eingang ein Gleitsinussignal mit einer Anfangsfrequenz von 1 Hz und einer Endfrequenz von 10 Hz. Erstellen Sie wieder das Modell mit „STRG+B“ und verbinden Sie es mit dem PMS. Bevor Sie nun den Motor einschalten stellen Sie wieder sicher, dass der Drehknopf der „Attenuator Unit“ auf 0 gedreht ist. Schalten Sie den Motor ein, er sollte sich nicht drehen. Drehen Sie nun den Drehknopf der „Attenuator Unit“ auf 2. Der Motor sollte sich immer noch nicht bewegen. Klicken Sie um das Experiment zu starten auf den Start Button in Simulink. Nachdem das Experiment beendet wurde, speichern Sie die im Workspace vorhandene Variable `velocity_measured` unter der Variable `velocity_measured_sweep` ab, damit die Daten bei der Durchführung des nächsten Experiments nicht verloren gehen.

Wiederholen Sie die Prozedur mit einem weiteren Gleitsinus von 1 Hz bis 9 Hz und dem Random Binary Signal als Eingang und speichern Sie erneut die Variable `velocity_measured` unter der Variable `velocity_measured_sweep2`, bzw. `velocity_measured_rbs` ab. Speichern Sie die Variablen im Workspace unter Verwendung des Befehls `save` in einer Datei namens `io_data.mat` ab. Schalten Sie nach den Experimenten wieder den Motor aus.

Nun sollten sich im Workspace vier Variablen mit den Namen

- `velocity_measured_sin`,
- `velocity_measured_sweep`, `velocity_measured_sweep2` und
- `velocity_measured_rbs`

befinden.

6.4.3 Durchführung der Systemidentifikation

Die graphische Benutzeroberfläche der System Identification Toolbox kann mit dem Befehl `ident` gestartet werden. Hier müssen Sie nun die verschiedenen Messdatensätze importieren. Trennen Sie jeweils die aufgezeichneten Daten in einen „Estimation“ Teil und einen „Validation“ Teil. Der erste Teil dient der Schätzung der Parameter der Modelle und der zweite Datensatz der Validierung. Es empfiehlt sich für den „Estimation“-Datensatz die ersten 15 Sekunden zu benutzen, und den Rest als „Validation“-Teil. Für den Gleitsinus verwenden Sie den ersten Datensatz als „Estimation“-Teil und den zweiten als „Validation“-Teil.

Aufgabe 6.11:

Starten Sie die graphische Benutzeroberfläche der System Identification Toolbox mit dem Befehl `ident`. Importieren Sie die erstellten Daten, die dem Identifikationsverfahren zugrunde gelegt werden sollen („Import Data“ - „Time-Domain Signals“). Geben Sie im „Import Data“ Fenster die Namen der Vektoren ein, die Ein- und Ausgangsdaten ohne Zeitstempel enthalten, sowie die Zeitschrittweite („sampling interval“) und die Anfangszeit. Denken Sie an eine sinnvolle Bezeichnung der Datensätze und teilen Sie die einzelnen Datensätze in einen Datensatz zum Fitten der Modelle und einen zum Validieren der Modelle auf.

In der linken Fensterhälfte werden die importierten Datensätze angezeigt. Die Zeitverläufe können Sie durch Auswahl der Option „Time plot“ betrachten.

Aufgabe 6.12:

Erstellen Sie nun mit den vorhandenen Estimation Datensätzen mehrere ARX Modelle verschiedener Ordnungen (darunter Modelle 1. und 2. Ordnung, aber auch höhere Ordnung (5.-10. Ordnung)). Dafür steht Ihnen die Funktion „Linear parametric models“ unter „Estimate“ – > „Polynomial Models“ zur Verfügung. Beachten Sie, dass das Modell für den als „Working Data“ gekennzeichneten Datensatz berechnet wird. Erstellen Sie für die verschiedenen Estimation-Datensätze einige Modelle (v.a. für das Sinussignal aus Abschnitt 6.3).

6.4.4 Bewertung der Modelle

Wie bereits erwähnt, hängt die Genauigkeit Ihres Modells von der Zahl der Freiheitsgrade des Modells, und damit der zu bestimmenden Parameter, ab. Ist diese Zahl zu hoch, wird allerdings auch das Eingangssignal mitmodelliert. Es gibt kein exaktes Kriterium, welches Modell am besten geeignet ist. Stattdessen müssen die verschiedenen Informationen gegeneinander abgewogen werden. Im Folgenden lernen Sie verschiedene Möglichkeiten kennen, die Güte eines Modells abzuschätzen.

Die Identification Toolbox bietet sechs verschiedene Möglichkeiten, um die Qualität der geschätzten Modelle zu prüfen. Durch Auswahl der verschiedenen „Model Views“ werden diese in separaten Fenstern geöffnet. Mit jedem können Sie bestimmte Aussagen über die Güte der erstellten Modelle machen. Im Folgenden werden beispielhaft drei dieser Möglichkeiten näher betrachtet.

Das Ausgangsverhalten

Das Fenster „Model Output“ vergleicht den tatsächlichen Systemausgang mit dem von den erstellten Modellen vorhergesagten und berechnet die Abweichung zwischen diesen. Hierzu sollte ein Validierungsdatensatz zu Grunde gelegt werden. Würde man zum Validieren denselben Datensatz wie zur Berechnung des Modells verwenden, werden Modelle höherer Ordnung immer die höhere Genauigkeit erzielen, da bei der Modellerstellung mehr Parameter bestimmt werden können und die Berechnung von Modell und Modellgenauigkeit dem gleichen quadratischen Fehlerkriterium unterliegt. Da aber das zu identifizierende System jedoch eine stochastische Komponente enthält - das gemessene Ausgangssignal ist stark rauschbehaftet.

tet - ergibt die Anwendung der errechneten Modelle auf einen Validierungsdatensatz bessere Ergebnisse.

Aufgabe 6.13:

Wählen Sie als „Validation Data“ einen Validation Datensatz. Ziehen Sie dazu dessen Symbol vom linken Fensterteil zum „Validation Data“ Symbol mittig unten in der GUI. Betrachten Sie nun den prädizierten „Model Output“. Der Fit entspricht der in Abschnitt 6.3.1 eingeführten Definition.

Der Frequenzgang

Nun sollen die verschiedenen Modelle im Frequenzbereich untersucht werden.

Aufgabe 6.14:

Betrachten Sie zunächst unter „Frequency Response“ die Bodediagramme der Übertragungsfunktionen. Was fällt Ihnen bei den Modellen höherer Ordnung auf? Woher kommt dieses Verhalten?

Die Korrektheit der Darstellung darf jedoch angezweifelt werden, wie Sie gleich sehen werden.

Aufgabe 6.15:

Lassen Sie sich das „99% Confidence Interval“ (unter „Options“ im „Frequency Response“-Fenster) der erstellten Modelle anzeigen. Was lässt sich also über die Gültigkeit der Bodediagramme gerade bei hohen Frequenzen sagen?

Das Pol-Nullstellen-Diagramm

Ein weiteres Bewertungskriterium für die identifizierten Modelle ist die Lage der Pole und Nullstellen in der komplexen Ebene.

Aufgabe 6.16:

Lassen Sie sich die Pole und Nullstellen Ihrer Modelle in der komplexen z -Ebene anzeigen. Welche Pole liegen sehr dicht an Nullstellen?

Aufgabe 6.17:

Berücksichtigen Sie jetzt auch die Vertrauensbereiche der Pole und Nullstellen (unter „Options“ im „Zeros and Poles“-Fenster). Bei welchen Modellen legt diese Untersuchung nahe, dass zu viele Pole und Nullstellen modelliert wurden?

6.4.5 Identifikation komplexerer Modellstrukturen

In diesem Teil des Praktikums werden andere Modellstrukturen für die Identifikation des PMS verwendet, und zwar ARMAX-, OE- und State Space-Modelle.

Aufgabe 6.18:

Identifizieren Sie für den Sinus-Datensatz verschiedene Ordnungen für die drei Modellstrukturen.

Aufgabe 6.19:

Vergleichen Sie nun die Modelle, deren Parameter Sie anhand des Sinus-Datensatzes identifiziert haben, mit den Validierungsdatensätzen der anderen Experimente (RBS und Sinus Sweep). Was fällt Ihnen auf? Vor allem wenn man nun Modelle mit hoher Ordnung und Modelle mit (zu) niedriger Ordnung vergleicht?

6.4.6 Vergleich mit dem physikalischen Modell

In diesem Teil soll das in Abschnitt 6.3 gewonnene Modell $G_{opt}(s)$ mit den identifizierten Modellen verglichen werden.

Aufgabe 6.20:

Erstellen Sie im Workspace die Übertragungsfunktion `G_opt` mit den in Abschnitt 6.3 optimierten Parametern. Importieren Sie dieses Modell im `ident` GUI und vergleichen Sie es mit den identifizierten Modellen anhand verschiedener Validierungs-Datensätze.

6.4.7 Fazit

Die Identifikation ist ein entscheidender Bestandteil der Regelungstechnik in praktischen Anwendungen. Das Ergebnis der Systemidentifikation hängt entscheidend ab von der Wahl der Ein-/Ausgangsdaten, der Modellstruktur, sowie des numerischen Verfahrens zur Bestimmung der Modellparameter. Daher ist eine sorgfältige Wahl dieser Bestandteile entscheidend und muss im Hinblick auf den späteren Verwendungszweck des Modells durchgeführt werden. Wie Sie in diesem Abschnitt gesehen haben, kommt der Modellvalidierung eine große Bedeutung zu. In der Praxis ist die Modellidentifikation ein iterativer Prozess, an dessen Ende ein verlässliches Modell der Regelstrecke stehen sollte, welches die Basis für den modellbasierten Reglerentwurf darstellt.

Dem Reglerentwurf im nächsten Abschnitt soll das im Abschnitt 6.3 optimierte Modell zu Grunde gelegt werden. Der Grund dafür ist zum Einen, dass so keine Umrechnungen zwischen verschiedenen Modellstrukturen notwendig sind, und zum Anderen, dass die physikalische Bedeutung der Zustände im Modell (6.1) offensichtlich ist. Das erleichtert insbesondere die Trajektorien-Folgeregelung in Abschnitt 6.5.2. Durch geeignete Zustandstransformation wäre dies dennoch auch für die identifizierten Modelle möglich.

6.5 Positionsregelung

Nachdem in den vorigen Abschnitten ein Modell für die Regelstrecke hergeleitet wurde, sollen in diesem Abschnitt verschiedene Ausgangsregler für die Positionierung des Motors entworfen werden. Da die verwendeten Reglerentwurfsverfahren modellbasiert sind, muss hierzu die

Geschwindigkeitsübertragungsfunktion, welche das physikalische Modell beschreibt, um einen Integrator erweitert werden. Man erhält die Übertragungsfunktion für die Position in der Form $G_\theta(s) = G_\omega(s) \frac{1}{s} = \frac{a}{s^3 + bs^2 + cs}$. Diese Übertragungsfunktion hat einen Pol bei 0 und zwei Pole in der linken Halbebene und das System ist daher grenzstabil.

6.5.1 PI-Regelung mit Anti-Windup

Hier soll nun ein PI-Regler untersucht werden. Es werden Probleme deutlich werden, die mit der Anti-Windup Struktur behoben werden können.

In den meisten realen Systemen sind es Stellgrößenbeschränkungen zu beachten, die entweder natürlich gegeben sind oder aus sicherheitsrelevanten Aspekten entstehen. Diese Stellgrößenbeschränkungen können jedoch zu Problemen bei der Regelung führen. Ein Beispiel hierfür ist das sog. Windup. Dieses kann auftreten, wenn ein Prozess mit einem Regler mit integralem Anteil (z.B. einem PI-Regler) geregelt wird. Fordert der Regler eine Stellgröße, die vom Prozess durch die Stellgrößenbeschränkung nicht realisierbar ist, kommt es zu einer Differenz von tatsächlicher und geforderter Stellgröße. Der Regler denkt, er könne mit immer größeren Stellgrößen den Regelfehler überwinden, bzw. der integrale Teil des Reglers „lädt sich auf“, da er davon ausgeht mit einer größeren Stellgröße eine bessere Wirkung zu erzielen. Wenn es nun wieder zu einem negativen Regelfehler kommt, sprich die geforderte Position wieder unter die aktuelle Position gefallen ist, sollte sich der Motor eigentlich in die andere Richtung drehen. Jedoch muss der integrale Teil sich zuerst entladen, weshalb sich der Motor vorerst noch in die falsche Richtung weiter dreht. Dies setzt sich solange fort, bis sich der integrale Teil wieder entladen hat. Es kann vorkommen, dass der Regler überhaupt nicht mehr in der Lage ist, die geforderte Position auch nur annähernd zu erreichen.

Um diesen Effekt zu zeigen wurde bereits ein PI-Regler entworfen. Der PI-Regler wurde als $K(s) = 0.357(1 + \frac{1}{0.2s})$ bestimmt.

Aufgabe 6.21:

Überprüfen Sie als erstes, ob Ihr System mit diesem Regler stabilisiert werden kann. Nutzen Sie hierfür MATLAB.

Aufgabe 6.22:

Öffnen Sie die Datei „A_2_AntiWindup.mdl“ im Verzeichnis „Regelung“. Der Regler wurde schon implementiert. Stellen Sie sicher, dass das AWgain auf „0“ steht. Führen Sie das Experiment durch. Was fällt Ihnen auf? Ist der Regler in der Lage, ein gutes Trajektorien-Folge-Verhalten zu erzeugen? Beachten Sie auch die Stellgröße.

Aufgabe 6.23:

Stellen Sie jetzt bitte den „phase shift“ der Referenztrajektorie (zu erreichen durch einen Doppelklick auf den „Sine Wave“-Block) auf 1. Hierdurch wird ein Anfangsfehler erzeugt, so dass zu Beginn gewünschte und tatsächliche Position nicht übereinstimmen. Stellen Sie sicher, dass das AWgain auf „0“ steht. Führen Sie das Experiment durch. Was fällt Ihnen auf? Ist der Regler immernoch in der Lage, ein gutes Trajektorien-Folge-Verhalten zu erzeugen? Beachten Sie auch die Stellgröße.

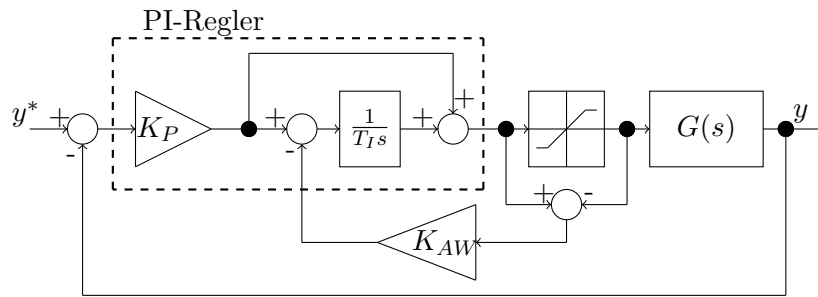


Abbildung 6.4: Anti-Windup Struktur.

Um dieses Problem zu beheben, wurde die Anti-Windup Struktur implementiert, wie in Abbildung 6.4 dargestellt.

Durch eine Rückführung der Differenz zwischen gefordertem und beschränktem Stellsignal und gleichzeitige negative Aufschaltung vor dem integralen Anteil des PI-Reglers kann ein Aufladen des Integrators verhindert werden. Hierbei spielt die Verstärkung der Rückführung eine eher geringe Rolle, jedoch je größer dieser Verstärkungsfaktor ist, desto schneller entlädt sich der Integrator wieder.

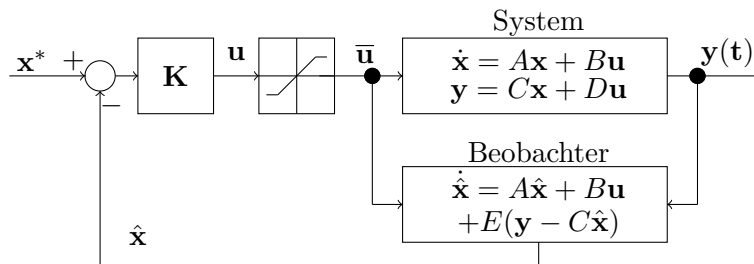


Abbildung 6.5: Zustandsrückführung mit Luenberger Beobachter.

Aufgabe 6.24:

Erhöhen Sie nun bitte den Verstärkungsfaktor, im Simulink Modell „AWgain“ genannt, auf z.B. 100 und führen Sie ein neues Experiment durch. Was fällt Ihnen auf? Ist der Regler wieder in der Lage, ein gutes Trajektorien-Folge-Verhalten zu erzeugen? Beachten Sie auch die Stellgröße. Versuchen Sie auch andere Verstärkungsfaktoren. Anmerkung: Wenn nur „gains“ verändert werden, müssen Sie das Modell nicht neu erstellen (Build Model) sondern können diesen Schritt überspringen und direkt auf „connect to target“ klicken.

6.5.2 Zustandsrückführung mit Luenberger Beobachter

Als letztes Regelungskonzept soll eine Zustandsrückführung entworfen werden. Da bei einer Zustandsrückführung alle Zustände benötigt werden, in unserem Fall aber nur einer bzw. zwei Zustände meßbar sind, müssen die restlichen Zustände mit einem Beobachter geschätzt werden. Im Folgenden gehen wir davon aus, dass nur die Position des Motors gemessen werden kann. Das Prinzip der Zustandsrückführung mit Beobachter ist in Abbildung 6.5 dargestellt.

Wir werden die Untersuchungen an dem physikalisch motivierten Zustandsraummodell (6.1) durchführen und die optimierten Parameter nutzen.

Zuerst soll nun ein Beobachter entworfen werden. Ein Beobachter schätzt aus den Ein- und Ausgängen des Systems und der Kenntnis des dynamischen Verhaltens des Systems den aktuellen Systemzustand. Durch eine geeignete Wahl der Pole der Beobachtdynamik (i.d.R. drei bis vier Mal schneller als die Pole des geschlossenen Regelkreises) kann man garantieren, dass die geschätzten Zustände hinreichend schnell gegen die tatsächlichen Zuständen konvergieren. Bevor jedoch ein Beobachter entworfen werden kann, muss zuerst die Beobachtbarkeit geprüft werden.

★ Aufgabe 6.25:

Überprüfen Sie das System aus Gleichung (6.1) auf Beobachtbarkeit, falls nur die Position gemessen wird, sprich die Ausgangsmatrix ist hierfür $C = [1 \ 0 \ 0]$.

Im Folgenden soll ein Luenberger Beobachter der Form

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} + E(\mathbf{y} - C\hat{\mathbf{x}})$$

entworfen werden. Hierbei sind A und B die Matrizen der Zustandsraumdarstellung des Systems (6.1), $C = [1 \ 0 \ 0]$ die Ausgangsmatrix und E die Beobacherverstärkung.

★ **Aufgabe 6.26:**

Leiten sie die Dynamik des Beobachterfehlers $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ her.

Aufgabe 6.27:

Bestimmen Sie nun die Reglerverstärkung K und die Beobacherverstärkung E . Verwenden Sie dazu den Befehl `place` in MATLAB. Beachten Sie, dass der Beobachterfehler hinreichend schnell konvergieren muss damit das tatsächliche Verhalten des geschlossenen Regelkreises von den vorgegebenen Polen bestimmt wird. Wählen sie daher die Pole der Beobachterfehlerdynamik etwa drei- bis viermal so weit in der linken Halbebene wie die Pole des geschlossenen Regelkreises. Das Skript „A_4_ZRF_script.m“ kann Ihnen dabei hilfreich sein.

$$\begin{aligned} E &= \\ K &= \end{aligned}$$

Neben dem Entwurf des Zustandsreglers ist auch die Berücksichtigung der Referenztrajektorie wichtig. Für den Fall einer konstanten Sollwertvorgabe kann, wie aus der Vorlesung bekannt, das Eingangssignal zu

$$\mathbf{u} = -K\mathbf{x} + V\mathbf{r}$$

gewählt werden, wobei \mathbf{r} einer Vorgabe für den Ausgang \mathbf{y} entspricht und V eine geeignete Matrix ist. In diesem Versuch werden jedoch zeitlich veränderliche Solltrajektorien betrachtet. Die Struktur des Regelkreises wird so gewählt wie in Abbildung 6.5. Hier gibt es eine Vorgabe \mathbf{x}^* für den ganzen Zustandsvektor. Im folgenden soll gezeigt werden, dass für eine bestimmte Wahl dieser Vorgabe der Regelfehler $\tilde{\mathbf{x}} = \mathbf{x}^* - \mathbf{x}$ zu Null konvergiert. Es wird dabei angenommen, dass keine Stellgrößenbeschränkung vorliegt und der beobachtete Zustand $\hat{\mathbf{x}}$ genau dem tatsächlichen entspricht. Damit ergibt sich

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}}^* - \dot{\hat{\mathbf{x}}} \\ &= \dot{\mathbf{x}}^* - (A\mathbf{x} + BK\tilde{\mathbf{x}}) \\ &= \dot{\mathbf{x}}^* - (A\mathbf{x}^* - A\tilde{\mathbf{x}} + BK\tilde{\mathbf{x}}) \\ &= \dot{\mathbf{x}}^* - A\mathbf{x}^* + \underbrace{(A - BK)}_{\text{Hurwitz}} \tilde{\mathbf{x}}. \end{aligned}$$

Somit konvergiert der Regelfehler zu null, wenn der Ausdruck $\dot{\mathbf{x}}^* - A\mathbf{x}^*$ identisch null ist. Dies entspricht der Bedingung, dass

$$\dot{\mathbf{x}}^* = A\mathbf{x}^*$$

gilt. In anderen Worten, wenn die Solltrajektorie $\mathbf{x}^*(t)$ die Differentialgleichung des *autonomous Systems* erfüllt, konvergiert der Regelfehler zu null.

★ **Aufgabe 6.28:**

Bestimmen Sie für eine gegebenes Positionsreferenzsignal $\theta^*(t)$ sinnvolle Referenztrajektorien für die Geschwindigkeit und den Motorstrom. Nutzen Sie dazu das Modell aus Gleichung (6.1).

$$\begin{aligned} \omega^*(t) &= \\ i^*(t) &= \end{aligned}$$

Aufgabe 6.29:

Der Regler soll nun mit Hilfe der Datei „A_4_ZRF.mdl“ implementiert werden. Tragen Sie dort bitte Ihren Beobachter und Ihren Regler K ein. Ebenso muss ein Referenzsignalgenerator implementiert werden. Nutzen Sie hierfür bitte die Parameter aus der Optimierung. Als gewünschte Position soll ein Sinussignal mit Amplitude 50, Frequenz von 0,5 rad/s und einer Phasenverschiebung von 1 gewählt werden, d.h. $\theta^* = 50 \cdot \sin(\pi t + 1)$. Führen Sie zwei Experimente durch, einmal mit der Referenztrajektorie aus Aufgabe 6.28 und einmal mit $\omega^*(t) = i^*(t) = 0$. Bewerten Sie das Verhalten im Vergleich zu den anderen Experimenten, die Sie gesehen haben. Was fällt Ihnen insbesondere an der Stellgröße auf?

7

Versuch: Identifikation und Regelung eines Torsionsschwingers

7.1 Einleitung

Im vorliegenden Versuch sollen verschiedene Methoden, die in der Vorlesung “Einführung in die Regelungstechnik” erlernt wurden, an einem realen Gerät angewandt werden. Einzelne Aspekte der systematischen Vorgehensweise zur Lösung regelungstechnischer Problemstellungen aus Kapitel 2 dieses Handbuches werden dabei vertieft betrachtet. Darüber hinaus werden einige weiterführende regelungstechnische Problemstellungen sowie praxisrelevante Aspekte angesprochen.

Dieser Praktikumsversuch befasst sich mit den Themen

- Modellierung,
- Parameteridentifikation,
- Simulation,
- Regelung und
- Trajektorienplanung.

Im Einzelnen werden in diesem Versuch folgende Punkte der systematischen Vorgehensweise aus Kapitel 2 behandelt:

- Punkt b) in Kapitel 7.5: Erstellung eines Modells und Identifikation der unbekannten Parameter.
- Punkt c) in Kapitel 7.6: Ermittlung von Eigenschaften des Modells und Test auf Übereinstimmung mit der Realität.
- Punkte f), g) und i) in Kapitel 7.7 und 7.8: Beschäftigung mit Regelkreisstruktur, Reglertyp und Bestimmung der Reglerparameter.

Das betrachtete Gerät ist ein sogenannter Torsionsschwinger, ein elektromechanisches Ersatzmodell für rotierende Wellen (wie z.B. in Antriebssträngen) oder elastisch gekoppelte Massen.

Notwendige Vorkenntnisse:

- Bodediagramm
- PD-Regler
- Polvorgabe und Beobachter

Obligatorische Vorleistungen *vor* dem Praktikumstermin:

- Durcharbeiten dieser Anleitung
- Lösen der mit * gekennzeichneten Vorbereitungsaufgaben.

Während der Praktikumsdurchführung sollen Sie Messwerte und erhaltene Ergebnisse in die dafür vorgesehenen Felder bzw. Tabellen eintragen sowie die Antworten zu den Fragen notieren.

7.2 Versuchsaufbau



Abbildung 7.1: Der Torsionsschwinger und die Komponenten des Versuchsaufbaus.

Das im vorliegenden Versuch betrachtete Gerät ist ein sogenannter Torsionsschwinger, siehe Abb. 7.1 links. Drei scheibenförmige Drehmassen sind vertikal übereinander angebracht. Die Scheiben sind durch Torsionsfedern (dünne elastische Schäfte) miteinander gekoppelt. An den Enden ist der Aufbau drehbar gelagert, am unteren Ende befindet sich ein Antrieb mittels eines bürstenlosen Gleichstrommotors und eines Zahnriemens. Die Drehwinkel der drei Scheiben werden jeweils über einen Zahnriemen und einen inkrementellen Winkencoder gemessen. Die drei gemessenen Winkelsignale werden direkt von den A/D-Wandlern der dSpace-Signalverarbeitungskarte (im PC) gelesen und entsprechend ihrer Codierung in Winkelmesswerte umgesetzt. Über die Software “Control Desk” kann auf dem PC mit der dSpace-Karte kommuniziert werden. Die gemessenen Signale können angezeigt und Reglerparameter, Sollwerte usw. können gesetzt werden. Steuer- bzw. Reglerkommandos werden von den D/A-Wandlern der dSpace-Karte an einen Leistungsverstärker und von dort an den Antriebsmotor gegeben. Ebenfalls in Abb. 7.1 sind die Komponenten des Versuchsaufbaus zu sehen, nämlich (von links nach rechts) der Torsionsschwinger, PC-Bedienkonsole, Verstärker, Kabel-Verbindungsbox zur dSpace-Karte. Der PC mit dSpace-Karte steht unter dem Tisch.

Zusammengefasst: Die Messgrößen des Torsionsschwingers sind die Winkelpositionen der drei Drehscheiben, die Stellgröße ist der am Motor anliegende Strom und damit indirekt das Drehmoment des Motors auf die untere Scheibe.

7.3 Sicherheitshinweise

Am Verstärker befindet sich der Haupt-Netzschalter der Apparatur. “Rot” bedeutet “Ein”. Zur Auslesung der Winkelsignale ist keine Energieversorgung über den Verstärker notwendig, wohl aber zur Ansteuerung des Motors.

Versichern Sie sich *immer*, dass alle Schrauben am Versuchsaufbau fest angezogen sind. Grundsätzlich sollten Haare, Kleidungsstücke, Hände und sonstige Körperteile vom Versuchsgesamt ferngehalten werden, wenn der Hauptschalter auf “Ein” gesetzt ist. Bei bestimmten Versuchsaufgaben können die Scheiben aber von Hand ausgelenkt werden, wenn sich die Steuerung bzw. Regelung in einem stabilen Zustand befindet.

Bei allen Versuchen befinden sich im Bedienfeld von Control Desk zwei Schalter mit der Beschriftung “An” und “Aus”. Standardmäßig ist beim Laden eines Experiments der Zustand “An” aktiv. Drücken der Schaltfläche “Aus” sollte alle Stellsignale und die Energiezufuhr zum Motor abschalten. Sollten “unkontrollierte” Stellsignale an den Motor gesandt werden (d.h. liegt eine offensichtliche Instabilität vor), und ist gleichzeitig eine Abschaltung über Control Desk nicht mehr möglich, muss der Hauptschalter am Verstärker sofort auf “Aus” gestellt werden.

7.4 Modell der Regelstrecke

Im Folgenden wird eine physikalische Modellierung des Torsionsschwingers vorgenommen. Eine schematische Darstellung der Elemente des Schwingers mit herrschenden Drehmomenten ist in Abb. 7.2 gezeigt. Die verwendeten Symbole sind in Tab. 7.1 erläutert. Die auftretenden

Momente sind:

- Antriebsmoment T ,
- viskose Reibung abhängig von der Drehgeschwindigkeit $d_{v_i}\dot{\varphi}_i$,
- Coulombsche Reibung abhängig von der Bewegungsrichtung $d_{c_i}\text{sign}(\dot{\varphi}_i)$,
- Motor-Cogging $k_c \sin(n\varphi_1 + \phi)$, und
- Rückstellmomente $k_i\varphi_j$.

Mit Hilfe des Drallsatzes können die Bewegungsgleichungen

$$\begin{aligned} J_1\ddot{\varphi}_1 &= T - k_1(\varphi_1 - \varphi_2) - d_{v1}\dot{\varphi}_1 - d_{c1}\text{sign}(\dot{\varphi}_1) - k_c \sin(n\varphi_1 + \phi) \\ J_2\ddot{\varphi}_2 &= -(k_1 + k_2)\varphi_2 + k_1\varphi_1 + k_2\varphi_3 - d_{v2}\dot{\varphi}_2 - d_{c2}\text{sign}(\dot{\varphi}_2) \\ J_3\ddot{\varphi}_3 &= k_2(\varphi_2 - \varphi_3) - d_{v3}\dot{\varphi}_3 - d_{c3}\text{sign}(\dot{\varphi}_3) \end{aligned}$$

für die allgemeinste Gerätekonfiguration, so wie in Abb. 7.2 dargestellt, angegeben werden. Einfache lineare Gleichungen ergeben sich mit den Vereinfachungen $k_c = 0$ (kein Motor-Cogging) und $d_{c_i} = 0$ (keine Coulombsche Reibung) sowie der Notationsänderung $d_i = d_{v_i}$. Der Motor und die Antriebseinheit werden nicht dynamisch modelliert, sondern vereinfacht mittels der Gerätekonstanten k erfasst, d.h. das Antriebsmoment lautet $T = k \cdot u$. Es ergibt sich

$$\begin{aligned} J_1\ddot{\varphi}_1 &= k_1(\varphi_2 - \varphi_1) - d_1\dot{\varphi}_1 + ku \\ J_2\ddot{\varphi}_2 &= k_1\varphi_1 - (k_1 + k_2)\varphi_2 + k_2\varphi_3 - d_2\dot{\varphi}_2 \\ J_3\ddot{\varphi}_3 &= k_2(\varphi_2 - \varphi_3) - d_3\dot{\varphi}_3. \end{aligned} \tag{7.1}$$

7.5 Identifikation

Im Folgenden sollen die Parameter des Systemmodells, die nur schwer oder gar nicht gemessen werden können, identifiziert werden.

★ Aufgabe 7.1:

In dieser Vorbereitungsaufgabe werden Formeln hergeleitet, mittels derer in der anschließenden Praktikumsaufgabe die gesuchten Parameter aus Messungen am System bestimmt werden können.

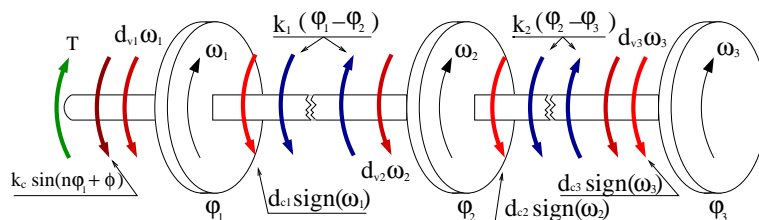


Abbildung 7.2: Freigeschnittener Torsionsschwinger liegend.

Symbol	Bedeutung	Einheit
\mathbf{A}	Dynamikmatrix	-
\mathbf{B}	Eingangsmatrix	-
\mathbf{C}	Ausgangsmatrix	-
\mathbf{x}	Zustandsvektor	-
\mathbf{y}	Ausgangsvektor	rad
u	Streckeneingang, Stellsignal	Nm
d_{vi}	Dämpfungskonstante viskose Reibung zur i -ten Scheibe	Nms/rad
d_{ci}	Dämpfungskonstante Coulombsche Reibung zur i -ten Scheibe	Nm
J_i	Massenträgheitsmoment der i -ten Scheibe	kg m ²
k_c	Motor-Cogging-Konstante	Nm
k_i	Federkonstante der i -ten Feder	Nm/rad
n	Übersetzungsverhältnis Antrieb (=2) mal Anzahl Polpaare (=3), damit $n = 6$	-
r	Radius eines Zusatzgewichtes (2,5 cm)	m
R	Abstand des Massenmittelpunktes einer Zusatzmasse zum Scheibenmittelpunkt (9cm)	m
m	Masse Zusatzgewicht (500 g)	kg
s	Laplacevariable	s ⁻¹
T	Drehmoment des Antriebsmotors	Nm
ϕ	Motor-Cogging-Phasenverschiebung	rad
$\varphi_i(t)$	zeitlicher Verlauf der Auslenkung der i -ten Scheibe	rad
$\omega_i(t)$	Winkelgeschwindigkeit der i -ten Scheibe ($= \dot{\varphi}_i(t)$)	rad/s
k	Gerätekonstante für Motor und Antrieb	-

Tabelle 7.1: Verwendete Symbole für den Torsionsschwinger.

- a) Zeigen Sie, dass eine Zustandsraumdarstellung der Gleichungen (7.1) folgende Form haben kann:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_1}{J_1} & -\frac{d_1}{J_1} & \frac{k_1}{J_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{J_2} & 0 & -\frac{k_1+k_2}{J_2} & -\frac{d_2}{J_2} & \frac{k_2}{J_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{J_3} & 0 & -\frac{k_2}{J_3} & -\frac{d_3}{J_3} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{k}{J_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}, \end{aligned} \quad (7.2)$$

wobei $\mathbf{x} := [\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \varphi_3, \dot{\varphi}_3]^T$ und $\mathbf{y} := [\varphi_1, \varphi_2, \varphi_3]^T$ definiert sind.

- b) In diesem Kapitel wird ein Versuchsaufbau wie in Abb. 7.3 benutzt: Der Motor ist nicht angetrieben ($u=0$), die untere und die obere Scheibe sind frei und zunächst ohne Zusatzmassen, die mittlere Scheibe ist festgeklemt.

Zeigen Sie, entweder durch Vereinfachung des Modells (7.1) oder durch einfache Anwendung des Drallsatzes, dass die Bewegungsgleichungen der beiden beweglichen Scheiben die Form

$$\begin{aligned} J_1 \ddot{\varphi}_1 + d_1 \dot{\varphi}_1 + k_1 \varphi_1 &= 0 \\ J_3 \ddot{\varphi}_3 + d_3 \dot{\varphi}_3 + k_2 \varphi_3 &= 0 \end{aligned}$$

haben.

Die Lösung einer Differentialgleichung der Form

$$\ddot{\varphi} + \underbrace{\frac{d}{J}}_{=:2\delta} \dot{\varphi} + \underbrace{\frac{k}{J}}_{=: \nu^2} \varphi = 0 \quad (7.3)$$

mit den Anfangsbedingungen $\varphi(0) = \varphi_0$, $\dot{\varphi}(0) = 0$ ist (für $\nu > \delta$)

$$\varphi(t) = \varphi_0 e^{-\delta t} \left(\cos \underbrace{\sqrt{\nu^2 - \delta^2} t}_{=: \omega} + \frac{\delta}{\sqrt{\nu^2 - \delta^2}} \sin \sqrt{\nu^2 - \delta^2} t \right) \quad (7.4)$$

mit den Abkürzungen wie zuvor definiert. Die Lösung ist also eine harmonische Schwingung, überlagert mit einer abklingenden Exponentialfunktion, ähnlich zu Abb. 7.4.

- c) Es sollen die Parameter k_1 , k_2 , d_1 und d_3 identifiziert, d.h. anhand von Messungen ermittelt werden. Dazu werden jeweils zwei Hochpunkte (t_1, ϕ_1) und (t_2, ϕ_2) sowie die Periodendauer T der abklingenden harmonischen Schwingungen experimentell bestimmt.

Daraus soll eine Formel zur Bestimmung der d_i , $i = 1, 3$ ermittelt werden. Zeigen Sie mittels der Dgl.-Lösung (7.4), dass eine Division der Messwerte $\phi_1 = \varphi_i(t_1)$ und $\phi_2 = \varphi_i(t_2)$ bei gleicher Anfangsauslenkung auf die Formel

$$d_i = 2J_i \frac{1}{t_2 - t_1} \ln \frac{\phi_1}{\phi_2}, \quad i = 1, 3 \quad (7.5)$$

führt.

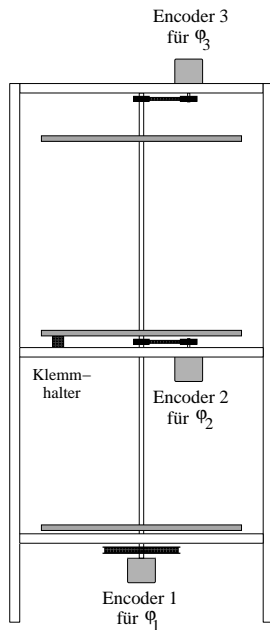


Abbildung 7.3: Versuchsaufbau
Aufg. 1.

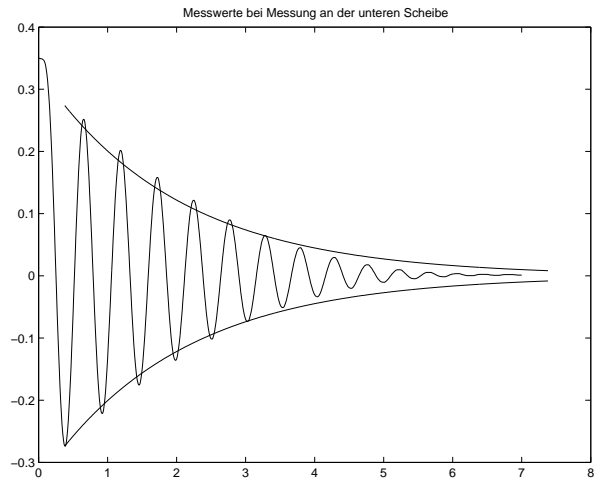


Abbildung 7.4: Abklingende Schwin-
gung.

- d) Zeigen Sie, dass dann die Parameter k_j , $j = 1, 2$ mittels

$$k_j = J_i \left(\left(\frac{2\pi}{T_i} \right)^2 + \left(\frac{d_i}{2J_i} \right)^2 \right), \quad (j, i) \in \{(1, 1), (2, 3)\} \quad (7.6)$$

bestimmt werden können.

Aufgabe 7.2:

Mittels Messungen am echten System und den zuvor bestimmten Formeln wird in dieser Aufgabe auf die gesuchten Parameter des Systems geschlossen.

- Der Verstärker sollte ausgeschaltet sein.
- Starten Sie MATLAB. Wechseln Sie ins Verzeichnis
C:\Documents and Settings\student\Desktop\Torsionsschwinger\
Durchfuehrung\1.Identifikation
- Konfigurieren Sie den Torsionsschwinger, wie er in Abb. 7.3 dargestellt ist, d.h. die untere und die obere Scheibe sind frei und zunächst ohne Zusatzmassen, die mittlere Scheibe ist festgeklemmt.
- Starten Sie den Windows Explorer und öffnen Sie die Datei Aufgabe1.cdx im Ordner
C:\Documents and Settings\student\Desktop\Torsionsschwinger\
Durchfuehrung\1.Identifikation\Dspace durch doppelklicken. Dadurch startet "Control Desk" und Experiment 1 wird geladen und gestartet.
- Aktivieren Sie den **Animation Mode**.

Scheibe	t_1 [s]	t_2 [s]	φ_1 [rad]	φ_2 [rad]	Winkel- Offset [rad]	Anzahl Perioden
1 (unten)						
3 (oben)						

Tabelle 7.2: Messwerte zur Bestimmung der Modellparameter.

- f) Lenken Sie die untere Scheibe um 20° aus, lassen Sie sie los und nehmen Sie die Schwingung auf. (Schwarz = untere Scheibe, blau = obere Scheibe). Sie können den Messschrieb mit dem **Start/Stop**-Bedienfeld neu starten.

Bestimmen Sie aus dem Messschrieb zwei Hochpunkte, den bleibenden Winkeloffset (im Allgemeinen wird der Winkel nicht zu Null zurückgehen), sowie die Anzahl der Schwingungsperioden zwischen den gewählten Hochpunkten. Sie können zum einfacheren Ablesen die Fadenkreuze verschieben und die numerischen Werte ablesen. Eventuell müssen Sie die Aufzeichnungszeit anpassen (Feld "Length" im "Capture Settings Window"). Tragen Sie Ihre Messwerte in Tab. 7.2 ein. Speichern Sie die aufgenommene Kurve mit dem **Save**-Knopf unter einem Namen wie z.B. **mess1.mat** ab.

Wiederholen Sie die Prozedur mit der oberen Scheibe.

Hinweis: Bei einer zu großen Verdrehung der Scheiben tritt eine Sicherheitsabschaltung in Kraft. Dann muss die Scheibe losgelassen und der Status wieder auf **An** gesetzt werden.

- g) Berechnen Sie die Parameter k_1 , k_2 , d_1 und d_3 anhand der in den Vorbereitungsaufgaben hergeleiteten Formeln. Es gelte $J_1 \approx 0,0025 \text{ kgm}^2$ und $J_3 \approx 0,0019 \text{ kgm}^2$. Vervollständigen Sie dazu das MATLAB-Skript **delta_plot.m**. Verstehen Sie, was in dem MATLAB-Skript passiert!

$$\begin{aligned} k_1 &= & , d_1 &= \\ k_2 &= & , d_3 &= \end{aligned}$$

- h) Warum ist $J_1 > J_3$, obwohl die Scheiben praktisch identisch sind? Warum ist $d_1 > d_3$?
Warum können $J_2 = J_3$ und $d_2 = d_3$ angenommen werden?
- i) Speichern Sie die Parameter k_i , $i = 1 \dots 2$, d_i , $i = 1 \dots 3$ und J_i , $i = 1 \dots 3$ zur späteren Verwendung. Vervollständigen Sie dazu das MATLAB-Skript **savaparameters.m**.
- j) Verlassen Sie den **Animation Mode** und beenden Sie das Programm auf der dSpace-Karte (roter quadratischer Knopf in der Control Desk Symbolleiste). Schließen Sie Control Desk. Führen Sie das MATLAB-Skript **weiter.m** aus um ins Verzeichnis **2_Validierung** zu wechseln und MATLAB neu zu initialisieren.

7.6 Modellvalidierung und Amplitudengang

In diesem Kapitel werden die Eigenfrequenzen des Aufbaus, wie er in Abb. 7.5 dargestellt ist, untersucht und Amplitudengänge des Systems werden aufgenommen.

Einerseits vermitteln diese Messungen ein Gefühl für das Konzept des Amplitudengangs im

Allgemeinen und für den Amplitudengang des Torsionsschwingers im Speziellen. Andererseits werden im Versuch die gemessenen Amplitudengänge mit den theoretisch aus dem erstellten Modell ermittelten Verläufen verglichen und auf diese Weise das erstellte Modell verifiziert. So könnten starke Abweichungen zwischen gemessenem und theoretisch ermitteltem Frequenzgang als Indiz für ein fehlerhaftes oder ungenaues Modell gewertet werden.

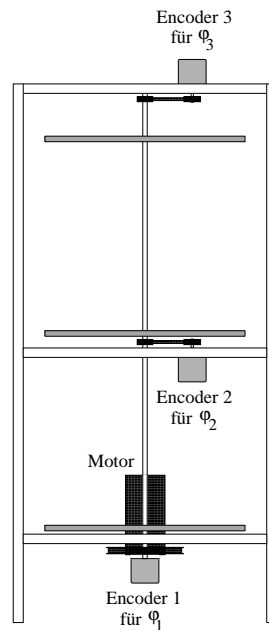


Abbildung 7.5: Versuchsaufbau Aufg. 2.

Aufgabe 7.3:

- Entfernen Sie alle Gewichte und Halter. Stellen Sie den Zustand, wie er in Abb. 7.5 dargestellt ist, her, d.h. alle Scheiben sind frei beweglich. Achten Sie darauf, dass der Verstärker ausgeschaltet ist!
- Öffnen Sie die Matlabdatei `parameter.m` und ergänzen Sie die fehlenden Angaben. Führen Sie das M-File `parameter.m` aus. Welche Resonanzfrequenzen sind zu erwarten?
- Schalten Sie den Verstärker ein.
- Öffnen Sie die Datei `Aufgabe2.cdx` im Verzeichnis `Dspace` durch Doppelklicken im Windows Explorer.
- Im Folgenden soll der Amplitudengang der Übertragungsfunktion vom Stellsignal zum Winkel der oberen Scheibe aufgenommen werden. Dazu soll dem System eine reine Sinus-Anregung mit definierter Amplitude aufgeprägt werden.
Die Frequenzen sind Tab. 7.3 zu entnehmen und in Control Desk in das Feld "Frequenz" einzutragen. Als "Amplitude" eignet sich der Wert 1 Nm. Durch Betätigen und Halten der Schaltfläche "Drück mich" prägen Sie ein sinusförmiges Signal auf den Eingang des Antriebmotors ein. Warten Sie, bis die Systemantwort eingeschwungen ist, und drücken Sie dann **Stop** im Messschrieb-Bedienfeld. Lesen Sie mit Hilfe der Fadenkreuze die stationäre Amplitude der Schwingung der oberen Scheibe ab (Schwarz = untere Scheibe,

ω [rad/s]	5	10	20	30	32	34	35	36	40
φ_3 (blau)									
ω [rad/s]	50	55	60	62	63	64	70	80	
φ_3 (blau)									

Tabelle 7.3: Messwerte zur Bestimmung des Amplitudengangs.

rot = mittlere Scheibe, blau = obere Scheibe, türkis = Sinus-Anregung). Unterscheiden Sie Schwingungsamplitude und Spitze-Spitze-Ausschlag! Tragen Sie Ihre Ergebnisse in Tab. 7.3 ein.

- f) Ein kleines Bonbon, das der Intuition widerspricht: Versuchen Sie eine Frequenz von 60,7 rad/s anzuregen. Die untere Scheibe (mit der ja der Antrieb verkoppelt ist) sollte sich nicht oder kaum bewegen, während die beiden oberen Scheiben hin- und herschwingen.
- g) Welche Resonanzfrequenzen treten im Experiment auf? Stimmen diese mit den aus dem Modell theoretisch berechneten überein?
- h) Öffnen Sie die Matlabdatei `messreihe.m`. Übertragen Sie die Werte aus Tab. 7.3 in die Matrix `messung`. Führen Sie die Funktion `messreihe.m` aus und vergleichen Sie den theoretischen und den gemessenen Amplitudengang. Wenn sich ein vertikaler Offset ergibt, versuchen Sie diesen durch Einführung eines Korrekturfaktors k ("Gerätekonstante") auszugleichen. Die Gerätekonstante k kann der Funktion `messreihe.m` hierbei als Parameter übergeben werden.
 $k =$
- i) Bestimmen Sie aus dem vorhandenen Modell `tsys` den zu erwartenden Abfall des Amplitudenganges für große Frequenzen. Wie lässt sich dieser Wert erklären?
- j) Schalten Sie den Verstärker aus. Verlassen Sie den **Animation Mode** und beenden Sie das Programm auf der dSpace-Karte. Schließen Sie Control Desk. Führen Sie das MATLAB-Skript `weiter.m` aus um ins Verzeichnis `3_PD_Regler` zu wechseln und MATLAB neu zu initialisieren.

Damit sind alle Parameter der Bewegungsgleichungen (7.1) identifiziert und gleichzeitig die relativ gute Übereinstimmung des Versuchsgörätes mit dem mathematischen Modell überprüft worden.

7.7 Reglerentwurf

In diesem Aufgabenteil soll eine einfache Positionsregelung der unteren Drehscheibe entworfen werden. Ziel wird es sein, dass Sie einen Winkelwert für die untere Scheibe vorgeben, und dieser dann ohne bleibende Regelabweichung erreicht wird.

★ Aufgabe 7.4:

Zeigen Sie, dass die Bewegungsgleichungen des Versuchsaufbaus, wie er in Abb. 7.6 darge-

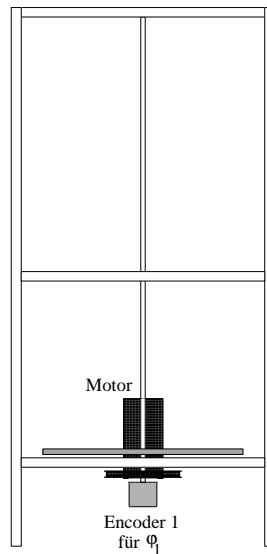


Abbildung 7.6: Versuchsaufbau Aufg. 3.

stellt ist, wie folgt lauten:

$$J_1 \ddot{\phi}_1(t) + d_1 \dot{\phi}_1(t) = ku(t)$$

Berechnen Sie die Übertragungsfunktion $G(s) = \Phi_1(s)/U(s)$. Um welche Klasse von Übertragungsglied handelt es sich?

Diese Art von Strecke kommt in der Praxis sehr häufig vor. Alle bewegten Massen mit Reibung und ohne Rückstellmechanismus können auf dieses einfache Modell reduziert werden, z.B. ein Auto usw.

Aufgabe 7.5:

- Konfigurieren Sie den Torsionsschwinger wie in Abb. 7.6, d.h. nur die untere Scheibe ist angebracht, der Torsionsschwinger ist frei drehbar. Achten Sie darauf, dass der Verstärker ausgeschaltet ist!
- Ermitteln Sie mit Hilfe des Nyquist-Kriteriums oder über die charakteristische Gleichung des geschlossenen Kreises, für welche Werte von $k_P = \text{const.}$ ein P-Regler $K(s) = k_P$, d.h. eine Rückführung der Form $U(s) = k_P(W(s) - \Phi_1(s))$, den geschlossenen Kreis stabilisiert. $W(s)$ sei der Sollwert. Welche bleibende Regelabweichung würde bei einem Sollwertsprung auftreten?
- Ermitteln Sie mit Hilfe des Nyquist-Kriteriums oder über die charakteristische Gleichung des geschlossenen Kreises, für welche Werte von $k_P = \text{const.}$ und $T_D = \text{const.}$ ein PD-Regler $K(s) = k_P + T_D s$, d.h. eine Rückführung der Form $U(s) = k_P(1 + T_D s)(W(s) - \Phi_1(s))$, den geschlossenen Kreis stabilisiert. Welche bleibende Regelabweichung würde bei einem Sollwertsprung auftreten?

- d) Führen Sie das MATLAB-Skript `parameter.m` zur Errechnung des Streckenmodells aus (in diesem Skript sind keine Veränderungen notwendig). Öffnen Sie als nächstes das Simulink-Modell `Aufgabe3_pd_sim.mdl`, und bauen Sie sich einen PD-Regler mit Hilfe eines Verstärkungsfaktors und eines Differenzierers. Simulieren Sie die Sprungantwort. Variieren Sie k_P und T_D , und überprüfen Sie die qualitativen und quantitativen Auswirkungen auf die Sprungantwort. Legen Sie den Regler so aus, dass kein Überschwingen auftritt sowie ein Einschwingen in unter 0.5 Sekunden erreicht wird.
- e) Öffnen Sie das Simulink-Diagramm `Aufgabe3_pd.mdl`, worin ein PD-Regler implementiert ist. Vergleichen Sie Ihren PD-Regler mit dem implementierten. Compilieren Sie den Regler mittels des Menüpunktes **Tools – Code Generation – Build Model**. Öffnen Sie danach die Datei `Aufgabe3_pd.cdx` im Verzeichnis `Dspace` durch Doppelklicken im Windows Explorer. Schalten Sie den Verstärker ein.
- Über Control Desk können Sie nun k_P und T_D zur Regelung der Strecke direkt verändern. Die Werte sollten im Bereich $0 < k_P < 8$ und $0 < T_D < 0.2$ bleiben. Beobachten Sie die Auswirkungen auf die Sprungantwort und auf das Stellsignal.
- Versuchen Sie Sprünge von 0,3 rad, 2 rad, 25 rad aufzuschalten.
- Die Signale in Control Desk sind: gemessener Winkel (schwarz), simulierter Winkel (hellblau), tatsächliches Stellsignal (rot), simuliertes Stellsignal (hellrot). Die simulierten Größen kommen von einer parallel laufenden Simulation wofür als Reglerparameter die im MATLAB-Workspace befindlichen Werte verwendet werden.
- f) Der D-Anteil des PD-Reglers könnte durch eine numerische Differentiation ersetzt werden. Dies bringt aber Probleme wie verstärktes Rauschen mit sich. Wie könnten Sie einen PD-Regler, der ja akausal ist, in kausaler Weise noch realisieren, ohne seine Eigenschaften grob zu verändern?
- g) Schalten Sie den Verstärker aus. Verlassen Sie den **Animation Mode** und beenden Sie das Programm auf der dSpace-Karte. Schließen Sie Control Desk. Führen Sie das MATLAB-Skript `weiter.m` aus.

7.8 Polvorgaberegler

In dieser Aufgabe wird ein Polvorgaberegler für den in Abb. 7.8 dargestellten Versuchsaufbau entworfen. Der Regler soll beide Scheiben auf die gleiche gewünschte Position führen. Natürlich besteht nach wie vor nur an der unteren Scheibe eine Stellmöglichkeit durch den Motor. Beachten Sie, dass das Trägheitsmoment J_3 nun wegen der Zusatzmassen deutlich größer ist als ohne, und damit die obere Scheibe zu größeren Schwingungen neigt.

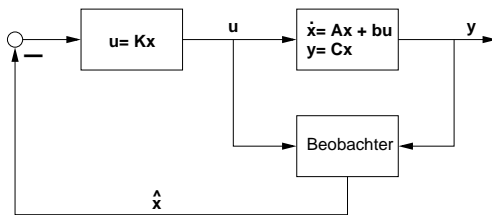


Abbildung 7.7: Schema des Regelkreises mit Zustandsrückführung und Beobachter.

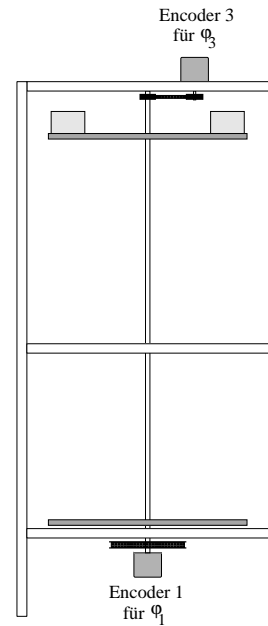


Abbildung 7.8: Versuchsaufbau beim Entwurf des Polvorgabereglers.

Das Modell hierzu ist

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_{12}}{J_1} & -\frac{d_1}{J_1} & \frac{k_{12}}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_{12}}{J_3} & 0 & -\frac{k_{12}}{J_3} & -\frac{d_3}{J_3} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{k}{J_1} \\ 0 \\ 0 \end{bmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) \\ \text{wobei } \mathbf{x} &:= [x_1, x_2, x_3, x_4]^T = [\varphi_1, \dot{\varphi}_1, \varphi_3, \dot{\varphi}_3]^T.\end{aligned}$$

★ **Aufgabe 7.6:**

- a) Welche Voraussetzung muss erfüllt sein, damit mittels einer Zustandsrückführung die Pole des geschlossenen Kreises beliebig vorgegeben werden können. Prüfen Sie, ob diese Voraussetzung hier erfüllt ist.
- b) Da nur die Winkel, nicht aber die Winkelgeschwindigkeiten gemessen werden können, soll ein Zustands-Beobachter der Form

$$\dot{\hat{x}}(t) = \mathbf{A}\hat{x} + \mathbf{b}u + \mathbf{E}(\mathbf{y} - \mathbf{C}\hat{x}) \quad (7.7)$$

konzipiert werden, siehe Abb. 7.7. Angenommen, Sie haben die Matrix \mathbf{E} des Beobachters bestimmt. Wie sieht nun die Zustandsraumrealisierung des Beobachters aus, d.h. wie lauten die Matrizen $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$ des “Beobachter”-Blocks in Abb. 7.7, mit den Eingängen u und \mathbf{y} sowie dem Ausgang \hat{x} ?

- c) Welche Voraussetzung muss erfüllt sein, damit die Fehlerdynamik eines Luenberger-Beobachters beliebig vorgegeben werden kann. Prüfen Sie, ob diese Voraussetzung hier erfüllt ist.

Aufgabe 7.7:

- a) Berechnen Sie die Federkonstante k_{12} der Gesamttorsionsfeder.
- b) Wie viele Pole müssen Sie vorgeben? Welche Messgrößen benötigt eine Zustandsrückführung in diesem Anwendungsfall?
- c) Ändern Sie den Versuchsaufbau so ab, wie er in Abb. 7.8 dargestellt ist, d.h. bringen Sie die obere Scheibe mit 4 Zusatzgewichten (ganz außen) an. Achten Sie darauf, dass der Verstärker ausgeschaltet ist!
- d) Öffnen Sie das Matlabskript `parameter_polvorgabe.m` und ergänzen Sie die fehlenden Zeilen. Legen Sie insbesondere Pole des geschlossenen Kreises fest. Führen Sie das Skript aus.
 Richtlinien zur Polvorgabe: MATLAB kann in diesem Fall nur voneinander verschiedene Pole vorgeben. Probieren Sie Pole aus, die nicht zu weit weg von denen der Strecke sind. Versuchen Sie, die bleibende Regelabweichung zu minimieren.
- e) Ein Beobachter wie in Gleichung (7.7) soll entworfen und implementiert werden. Zur Polvorgabe für die Fehlerdynamik wird ebenfalls der MATLAB-Befehl `place` benutzt. Es gilt: Eigenwerte von A = Eigenwerte von A^T . Der Befehl `place` berechnet die Matrix K so, dass $A - BK$ gewünschte Eigenwerte erhält. Für den Beobachter soll die Matrix E so berechnet werden, dass $A - EC$ gewünschte Eigenwerte erhält. Dies ist äquivalent zur Berechnung der Matrix E^T , so dass $(A - EC)^T = A^T - C^T E^T$ die besagten Eigenwerte erhält. Dies kann direkt mit `place` erreicht werden.
 Öffnen Sie die Matlabdatei `parameter_beobachter.m` und ergänzen Sie die fehlenden Zeilen. Führen Sie das Skript aus.
- f) Fügen Sie im Simulink-Diagramm `Aufgabe4_sim.mdl` Ihre Zustandsrückführung mit dem Beobachter zusammen. Wie und wo bringen Sie Sollwerte ein? Lesen Sie dazu auch die Hinweise im Kapitel 6.5.2 des Versuchs zur Identifikation und Regelung eines

Gleichstromelektromotors nach. Simulieren Sie Sprungantworten. Vergleichen Sie Ihren Gesamtregler mit dem im Simulink-Diagramm **Aufgabe4.mdl**.

- g) Öffnen Sie im Verzeichnis **Dspace** das Experiment **Aufgabe4.cdx** durch Doppelklicken im Windows Explorer. Schalten Sie den Verstärker ein.
 Lenken Sie die Scheiben um ca. 0,3 rad aus. Nehmen Sie die Finger weg und halten Sie die Schaltfläche “Drück mich” gedrückt. Jetzt muß Ihr Regler die Scheiben in die Nulllage zurück drehen. Beachten Sie die Amplitude des Stellsignals (rot). Die Winkelsignale sind schwarz (untere Scheibe) und blau (obere Scheibe).
 Im Eingabefenster **w** können Sie einen Führungsgrößensprung vorgeben. Bei zu großen Sprüngen wird die Stellgröße sehr hoch sein, und die Sicherheitsabschaltung wird wegen zu starker Verdrehung der Torsionsfeder abschalten.
- h) Um trotzdem große Führungssprünge zu ermöglichen, sollte – wie dies bei Robotern oder ähnlichen mechanischen Systemen üblich und angebracht ist – eine Solltrajektorienplanung erfolgen. Hier wird nun eine sehr einfache Trajektorienplanung vorgenommen.
 Wird ein Sollwert vorgegeben, berechnet eine Logikschaltung eine Gerade, die in vorgebarer Zeit den aktuellen Messwert der unteren Scheibe mit dem gewünschten Endwert verbindet. Diese Gerade wird dann als sich über der Zeit verändernder Sollwert vorgegeben.
- Öffnen Sie das Experiment **Aufgabe4.3.cdx**. Geben Sie einen Sollwert **w** ein. Geben Sie außerdem ein Zeitintervall “delta T” ein, innerhalb dessen der Sollwert linear vom aktuellen Winkelwert zum Sollwert verlaufen soll. Mit der Schaltfläche “Drück mich” können Sie den Regler aktivieren. Beachten Sie insbesondere die maximalen Stellamplituden und vergleichen Sie sie mit denen bei Vorgabe eines Sollwertsprungs zuvor.
- i) Schalten Sie den Verstärker aus. Verlassen Sie den **Animation Mode** und beenden Sie das Programm auf der dSpace-Karte. Schließen Sie Control Desk und MATLAB.

7.9 Anhang: Lösung der Schwingungsdifferentialgleichung

Für die Differentialgleichung

$$\ddot{\varphi} = \underbrace{\frac{d}{J}}_{=:2\delta} \dot{\varphi} + \underbrace{\frac{k}{J}}_{=: \omega^2} \varphi \text{ mit den Anfangsbedingungen } \varphi(0) = \varphi_0, \dot{\varphi}(0) = 0$$

sind 2 Lösungen aufgezeigt.

- a) Der allgemeinen Ansatz $\varphi = Ae^{\lambda t}$ führt auf das Charakteristische Polynom $\lambda^2 + 2\delta\lambda + \omega^2 = 0$ mit den Lösungen $\lambda_{1,2} = -\delta \pm j\sqrt{\omega^2 - \delta^2}$.
 Setzt man die Lösung in den allgemeinen Ansatz ein und beachtet, dass $e^{jx} = \cos x + j \sin x$ ist, so kommt man auf

$$\varphi(t) = e^{-\delta t} \left(A_1 \cos \sqrt{\omega^2 - \delta^2} t + j A_2 \sin \sqrt{\omega^2 - \delta^2} t \right).$$

Das Einsetzen der ersten Anfangsbedingung $\varphi(0) = \varphi_0$ ergibt, dass $A_1 = \varphi_0$ ist. Das Ableiten von $\varphi(t)$ ergibt

$$\begin{aligned}\dot{\varphi}(t) = & e^{-\delta t} \left(-\delta \left(\varphi_0 \cos \sqrt{\omega^2 - \delta^2} t + J A_2 \sin \sqrt{\omega^2 - \delta^2} t \right) \right. \\ & \left. + \sqrt{\omega^2 - \delta^2} \left(\varphi_0 \sin \sqrt{\omega^2 - \delta^2} t + J A_2 \cos \sqrt{\omega^2 - \delta^2} t \right) \right).\end{aligned}$$

Setzt man die zweite Anfangsbedingung $\dot{\varphi}(0) = 0$ ein, so erhält man für $A_2 = \frac{\varphi_0 \delta}{J \sqrt{\omega^2 - \delta^2}}$ und somit als Ergebnis

$$\varphi(t) = \varphi_0 e^{-\frac{d}{2J}t} \left(\cos \sqrt{\frac{k}{J} - \frac{d^2}{4J^2}} t + \frac{d}{\sqrt{4Jk - d^2}} \sin \sqrt{\frac{k}{J} - \frac{d^2}{4J^2}} t \right).$$

b) Mit Laplacetransformation ergibt sich die Differentialgleichung zu

$$s^2 \Phi + \frac{d}{J} s \Phi + \frac{k}{J} \Phi = \dot{\varphi}(0) + \left(s + \frac{d}{J} \right) \varphi(0).$$

Aufgelöst nach Φ

$$\begin{aligned}\Phi(s) &= \frac{s + \frac{d}{J}}{s^2 + \frac{d}{J}s + \frac{k}{J}} \varphi_0 = \frac{s + \frac{d}{J}}{\underbrace{s^2 + \frac{d}{J}s + \left(\frac{d}{2J}\right)^2}_{=(s + \frac{d}{2J})^2 = (s + \delta)^2} + \underbrace{\frac{k}{J} - \left(\frac{d}{2J}\right)^2}_{=\omega^2}} \varphi_0 \\ &= \left(\frac{s + \frac{d}{2J}}{(s + \frac{d}{2J})^2 + \frac{k}{J} - \left(\frac{d}{2J}\right)^2} + \frac{\frac{d}{2J\sqrt{\frac{k}{J} - \left(\frac{d}{2J}\right)^2}} \sqrt{\frac{k}{J} - \left(\frac{d}{2J}\right)^2}}{(s + \frac{d}{2J})^2 + \frac{k}{J} - \left(\frac{d}{2J}\right)^2} \right) \Phi_0\end{aligned}$$

und die Laplacerücktransformation ergibt die Lösung

$$\varphi(t) = \varphi_0 e^{-\frac{d}{2J}t} \left(\cos \sqrt{\frac{k}{J} - \frac{d^2}{4J^2}} t + \frac{d}{\sqrt{4Jk - d^2}} \sin \sqrt{\frac{k}{J} - \frac{d^2}{4J^2}} t \right).$$

8

Ansprechpartner

Praktikumskoordinator

Tim Martin
Zimmer 2.236
+49 711 685 66316
tim.martin@ist.uni-stuttgart.de

Betreuer der einzelnen Versuche

- **Invertiertes Pendel**

Tim Martin
Zimmer 2.236
+49 711 685 66316
tim.martin@ist.uni-stuttgart.de

- **Mehrtanksystem**

Steffen Linsenmayer
Zimmer 2.238
+49 711 685 69920
steffen.linsenmayer@ist.uni-stuttgart.de

- **Elektromotor**

Karsten Kuritz
Zimmer 3.244
+49 711 685 67757
karsten.kuritz@ist.uni-stuttgart.de

- **Torsionsschwinger**

Michael Hertneck
michael.hertneck@ist.uni-stuttgart.de

Literaturverzeichnis

- [1] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. Wiley-Blackwell; 2nd Edition edition, 2005.
- [2] M. Horn and N. Dourdoumas, *Regelungstechnik: Rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise*. Pearson Studium, 2004.
- [3] J. Lunze, *Regelungstechnik 1*. Springer Verlag, 1999.
- [4] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, 1999.
- [5] J. Lunze, *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*, 5th ed. Springer, 2008.