

Masterarbeit

# Vollaktuierter 2-Gelenkarm Roboter

## Aufbau, Modellierung und Regelung

Dominik Walser BSc

---

Institut für Regelungs- und Automatisierungstechnik  
Technische Universität Graz

Graz, im September 2011

---

## **Kurzfassung**

Gegenstand der hier vorgestellten Arbeit ist die Konstruktion, Modellierung und Regelung eines 2-achsigen Roboterarms mit 2 Freiheitsgraden. Der mechanische Aufbau ist so ausgelegt, dass die Winkel der Drehgelenke, welche den Arbeitsraum des Roboters festlegen, in ihrer Wahl nicht eingeschränkt sind. Anhand der mechanischen Realisierung sind alle Voraussetzungen zur Anwendung von Methoden der modellbasierten Synthese von Regelkreisen erfüllt. Dazu gehören analytische Modellbildung, Parameteridentifikation und weiters die Ermittlung von Regelgesetzen. Die mathematische Beschreibung erfolgt mit Hilfe des Euler-Lagrange Formalismus. Die Daten des Modells werden anschließend durch eine experimentelle Methode zur Parameterermittlung bestimmt.

Ziel der Arbeit ist die Positionsregelung des Endeffektors durch Vorgabe von Trajektorien im Gelenkraum bzw. im kartesischen Raum. Diese erzeugten Trajektorien können mit Hilfe eines Schreibwerkzeugs grafisch erfasst werden. Die Positionieraufgabe wird mit verschiedenen Regelungskonzepten untersucht, die sich durch ihre dynamischen Eigenschaften, Komplexität sowie Robustheit unterscheiden lassen. Stichwörter für die hier verwendeten Ansätze der Regelungskonzepte sind Inverse Dynamic Control im Gelenk- bzw. kartesischen Raum und Robust Control.

## **Abstract**

Purpose of this thesis is the construction, modeling and control of a 2-link robotarm with 2 degrees of freedom. The mechanical assembly is implemented in a way so that the joint angles which define the workspace of the robot, are not limited. On the basis of the mechanical construction all requirements for modelbased controller synthesis are fulfilled. This includes analytic modeling, parameter identification and furthermore the generation of control laws. The mathematical description of the robotic manipulator is derived by the formulation of Euler-Lagrange. Some parameters of the model are determined by a comparison between results from simulations and real measurements.

Goal of this master thesis is the position control of the end-effector in the sense of a given trajectory in the joint- or cartesian space. The task of positioning is carried out by different control techniques with different dynamic behaviour, complexity and robustness. The applied control techniques are Inverse Dynamic Control in joint space or cartesian space and Robust Control.

---

## **Danksagung**

Ich möchte mich an dieser Stelle recht herzlich bei Herrn Univ.-Prof. Dr. Anton Hofer für die freundliche und zielgerichtete Betreuung während der Masterarbeit bedanken. Weiters möchte ich dem gesamten Team des Instituts für ein offenes Ohr für angefallene Fragen danken. Insbesondere gilt mein Dank Herrn Dipl.-Ing. Christoph Schörghuber für die tatkräftige Unterstützung bei der mechanischen Konstruktion des Roboterarms.

Nicht zuletzt gilt mein Dank meiner Familie und Freunden, die stützend hinter mir gestanden und mich im Verlaufe meines Studiums immer wieder zu Höchstleistungen motiviert haben. Diese Masterarbeit ist meiner Freundin gewidmet, die mich während meiner gesamten Studienzeit in Graz aus der Ferne liebevoll unterstützt hat.

---

## Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit mit dem Thema

*Vollaktuierter 2-Gelenkarm Roboter  
Aufbau, Modellierung und Regelung*

selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

---

Ort, Datum

---

DOMINIK WALSER

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Ziel und Gliederung der Arbeit . . . . .	1
<b>2. Mechanischer Aufbau des Roboters</b>	<b>3</b>
2.1. Konstruktion . . . . .	3
2.2. Ansteuerung . . . . .	6
2.3. Quanser-Schnittstelle . . . . .	7
<b>3. Kinematik</b>	<b>8</b>
3.1. Koordinatensysteme . . . . .	8
3.2. Direkte Kinematik . . . . .	9
3.3. Inverse Kinematik . . . . .	10
3.4. Jacobi-Matrix . . . . .	12
3.5. Singularitäten . . . . .	13
<b>4. Dynamische Modellbildung</b>	<b>15</b>
4.1. Methode von Lagrange . . . . .	16
4.1.1. Wahl der generalisierten Koordinaten . . . . .	17
4.1.2. Potentielle- und kinetische Energie . . . . .	17
4.1.3. Lagrangesche Gleichungen 2. Art . . . . .	18
4.2. Dynamik der Antriebe . . . . .	20
4.3. Zustandsraummodell . . . . .	21
4.4. Reibungsmodellierung . . . . .	22
4.4.1. Oberarm . . . . .	24
4.4.2. Unterarm . . . . .	25
4.5. Simulation und Animation . . . . .	25

<b>5. Parameterermittlung</b>	<b>27</b>
5.1. Analytische Identifikation . . . . .	27
5.1.1. Trägheitsmoment des Oberarms . . . . .	29
5.1.2. Trägheitsmoment des Unterarms . . . . .	30
5.2. Experimentelle Parameterbestimmung . . . . .	30
5.3. Vergleich von Simulation und Experiment . . . . .	31
<b>6. Bewegungsplanung</b>	<b>34</b>
6.1. Pfad vs. Trajektorie . . . . .	34
6.2. Trajektorien im Gelenkraum - Punkt-zu-Punkt Bewegung . . . . .	35
6.2.1. Polynom . . . . .	35
6.2.2. Trapez . . . . .	39
6.3. Trajektorien im Arbeitsraum - Pfad Bewegung . . . . .	40
6.3.1. Bahnplanung in Parameterform . . . . .	41
6.3.2. Probleme im Arbeitsraum . . . . .	43
<b>7. Regelungskonzepte</b>	<b>46</b>
7.1. Allgemeine Betrachtung . . . . .	47
7.2. Inverse Dynamic Control im Gelenkraum . . . . .	48
7.2.1. Globale Linearisierung . . . . .	49
7.2.2. PD-Regler . . . . .	49
7.2.3. Beweis der Stabilität . . . . .	53
7.2.4. Implementierung in Simulink . . . . .	55
7.3. Inverse Dynamic Control im Arbeitsraum . . . . .	58
7.3.1. Kinematische Beziehungen . . . . .	58
7.3.2. Regularisierung der Jacobi-Matrix . . . . .	59
7.3.3. Implementierung in Simulink . . . . .	62
7.4. Robust Control . . . . .	64
7.4.1. Idee . . . . .	65
7.4.2. Robustes Regelgesetz . . . . .	66
7.4.3. Stabilitätsbetrachtungen . . . . .	69
<b>8. Zusammenfassung</b>	<b>72</b>
8.1. Fazit . . . . .	72
8.2. Ausblick . . . . .	73
<b>A. Anhang</b>	<b>i</b>
A.1. Motordaten . . . . .	ii
A.2. Technische Dokumentation . . . . .	iv
A.2.1. Inbetriebnahme . . . . .	iv
A.2.2. Quanser Initialisierung . . . . .	v
A.2.3. Parameterbestimmung am realen Modell . . . . .	viii

A.2.4. Implementierung IDC-Konzept im Gelenkraum . . . . .	viii
A.2.5. Implementierung IDC-Konzept im Arbeitsraum . . . . .	xii
A.2.6. Implementierung Robust Control . . . . .	xv
<b>Literaturverzeichnis</b>	<b>xvi</b>

## **Abkürzungsverzeichnis**

DGLn	.....	Differentialgleichungen
IDC	.....	Inverse Dynamic Control - Regelungskonzept
LQR	.....	Linear Quadratic Regulator - Regler mit quadratischem Gütekriterium
LSPB	.....	Linear Segments with Parabolic Blends - Lineare Segmente gemischt mit parabolischen
MGE	.....	Motor-Getriebe-Encoder
MIMO	.....	Multiple Input Multiple Output - Mehrgrößensystem
NBn	.....	Nebenbedingungen
OA	.....	Oberarm
RK	.....	Regelkreis
RTOS	.....	Real Time Operating System - Echtzeitbetriebssystem
SISO	.....	Single Input Single Output - Eingößensystem
SVD	.....	Singular Value Decomposition - Singulärwertzerlegung
TCP	.....	Tool Center Point - Werkzeugmittelpunkt
UA	.....	Unterarm

## Abbildungsverzeichnis

2.1. Tellerschleifringe . . . . .	4
2.2. Laboraufbau des Roboterarms . . . . .	5
2.3. Quanser Terminal Board . . . . .	6
3.1. Roboterkinematik . . . . .	9
3.2. 2-Gelenkarm Roboter . . . . .	10
3.3. Inverse Kinematik . . . . .	11
4.1. 2-Gelenkarm Roboter . . . . .	17
4.2. Modellierung der Antriebe . . . . .	20
4.3. Klassische Reibungsmodelle . . . . .	22
4.4. Reibungskennlinie des Oberarms . . . . .	24
4.5. Reibungskennlinie des Unterarms . . . . .	25
4.6. Animation des Roboterarms . . . . .	26
5.1. Schwerpunktbestimmung des Oberarms . . . . .	28
5.2. Bestimmung des Trägheitsmoments des Oberarms . . . . .	29
5.3. Konzept zur Bestimmung der Reibungsparameter . . . . .	30
5.4. Experiment $u_1=0V, u_2=1V$ . . . . .	32
5.5. Experiment $u_1=24V$ , Unterarm frei . . . . .	33
5.6. Experiment $u_1=15V, u_2=2V$ . . . . .	33
6.1. Zeitfenster für Position, Geschwindigkeit und Beschleunigung eines kubischen Polynoms . . . . .	36
6.2. Zeitfenster für Position, Geschwindigkeit und Beschleunigung aneinandergehängter kubischer Polynome . . . . .	38
6.3. Zeitfenster für Position, Geschwindigkeit und Beschleunigung einer LSPB Trajektorie . . . . .	40
6.4. Kreis in Parameterform . . . . .	41
6.5. Ellipse in Parameterform . . . . .	42
6.6. Gerade in Parameterform . . . . .	42
6.7. Problem Typ 1 . . . . .	43
6.8. Problem Typ 2 . . . . .	44
6.9. Problem Typ 3 . . . . .	45
7.1. Struktur zur Regelung eines SISO-System . . . . .	47

7.2. IDC-Regelkreisstruktur im Gelenkraum . . . . .	48
7.3. Struktur des entkoppelten Modells . . . . .	49
7.4. Struktur des linearen Modells . . . . .	51
7.5. Sprungantwort des Regelkreises, $\omega_n = 50$ . . . . .	52
7.6. Grafische Eingabe der Zielpunkte . . . . .	55
7.7. Vergleich PTP-Bewegung . . . . .	56
7.8. IDC Simulationskoppelplan im Gelenkraum . . . . .	57
7.9. Zusammengesetzte Bewegung . . . . .	62
7.10. IDC Simulationskoppelplan im Arbeitsraum . . . . .	63
7.11. Vergleich Pfad-Bewegung . . . . .	64
7.12. Regelkreisstruktur - Robust Control . . . . .	65
A.1. Messaufbau . . . . .	iv
A.2. Terminal Board mit angeschlossener Peripherie . . . . .	v
A.3. Anbindung des realen Modells in die Simulation . . . . .	vii
A.4. Quanser-Koppelplan zur Parameterbestimmung . . . . .	viii
A.5. Grafische Eingabe der Zielpunkte . . . . .	ix
A.6. Quanser-Koppelplan zur Simulation der PTP-Bewegung . . . . .	xi
A.7. Quanser-Koppelplan zur Simulation von geometrischen Figuren . . . . .	xiii

## **Tabellenverzeichnis**

2.1. Antriebskomponenten . . . . .	4
2.2. Spezifikation der Leistungsverstärker . . . . .	6
5.1. Gewichtsparameter . . . . .	27
5.2. Geometrische Abmessungen . . . . .	28
5.3. Reibungsparameter . . . . .	31
A.1. Daten der Gleichstrommotoren . . . . .	ii
A.2. Daten der Planetengetriebe . . . . .	ii
A.3. Daten der Encoder . . . . .	iii

# 1. Einleitung

## 1.1. Motivation

In der kurzlebigen Zeit von heute streben Ingenieure nach der Automatisierung von Prozessen. Hierbei stellt die Robotik eine wichtige Unterkategorie dar, welche mittlerweile viele Anwendungsgebiete in der Industrie findet. Den größten Einsatzbereich bestreitet nach wie vor die Automobilindustrie, wobei auch die Elektronikindustrie in Form von Bestückungsautomaten von der fortgeschrittenen Robotertechnologie profitiert. Die Klassifikation von Roboterarten reicht von Industrierobotern über autonome Roboter bis hin zu humanoiden Robotern. Bei dem in dieser Arbeit behandelten Roboter handelt es sich um einen Gelenkarm-Roboter, welcher in den Bereich der Industrieroboter eingegliedert werden kann. Die besondere Herausforderung stellt die Ermittlung effizienter Regelalgorithmen zur exakten Steuerung des Manipulators dar. Dabei bildet ein geeignetes mathematisches Modell der Dynamik des Roboters eine Grundvoraussetzung.

## 1.2. Ziel und Gliederung der Arbeit

Diese Masterarbeit beschreibt den mechanischen Aufbau, Modellierung und Regelung eines am Institut für Regelungs- und Automatisierungstechnik der Technischen Universität Graz realisierten Labormodells. Die regelungstechnische Aufgabe besteht in der Ermittlung von Regelkonzepten zur Steuerung des vorliegenden Mehrgrößensystems. Die Zielsetzung der Arbeit ist die präzise Positionsregelung des Roboters unter der Einhaltung gewisser Sollnebenbedingungen. Der Roboter soll einerseits eine kontinuierliche Punkt-zu-Punkt Bewegung und andererseits eine Bewegung entlang eines geometrischen Pfades durchführen können. Durch Implementierung der zweiten Option soll es dem Manipulator ermöglicht werden, geometrischen Figuren wie beispielweise Kreis, Ellipse oder Gerade abfahren zu können. Zur visuellen Kontrolle der geometrischen Bahnen soll weiters eine Möglichkeit gefunden werden, die Trajektorien in geeigneter Weise grafisch festzuhalten.

Die Arbeit setzt sich insgesamt aus 8 Kapiteln mit entsprechender Unterteilung in Abschnitte zusammen. Zu Beginn wird in Kapitel 2 die Konstruktion des mechanischen

Aufbaus besprochen. Es werden die hardware- sowie auch softwareseitigen Komponenten vorgestellt, welche in der Arbeit Anwendung finden. Im darauf folgenden Kapitel 3 wird die Kinematik des Starrkörpermodells behandelt. Anschließend wird im 4. Kapitel das mathematische Modell des dynamisches Systems erfasst. Die zur vollständigen Erfassung benötigte Parameteridentifikation wird im folgenden Kapitel 5 diskutiert. Bevor im 7. Kapitel auf die regelungstechnischen Konzepte eingegangen wird, behandelt Kapitel 6 die Bewegungsplanung des Roboters. Den Abschluss der Arbeit bildet Kapitel 8 mit einer Zusammenfassung und Ausblick auf weitere mögliche interessante Aspekte bzgl. des realisierten Labormodells.

## 2. Mechanischer Aufbau des Roboters

In diesem Kapitel werden der Aufbau und die verwendeten mechanischen Komponenten des Roboters erläutert. Wie der Titel der Arbeit bereits vermuten lässt, handelt es sich um einen vollaktuierten Roboterarm mit 2 Achsen. Folglich sind beide Gelenke mit Motoren ausgestattet. Bei den Achsen handelt es sich um Drehachsen, welche in der Wahl ihrer Gelenkwinkel nicht eingeschränkt sind. Der Manipulator wird in Ober- und Unterarm gegliedert.

Der erste Abschnitt beschreibt die Konstruktion des Gesamtmodells. Es werden die eingesetzten Motoren, Getriebe und Inkrementalencoder vorgestellt. Weiters wird kurz beschrieben, wie die mechanische Entkopplung zwischen Ober- und Unterarm durchgeführt wurde. Zudem zeigt Abschnitt 2.1 den Aufbau des Modells in den Laborräumlichkeiten des Instituts. In Abschnitt 2.2 wird die Ansteuerung des Gesamtaufbaus behandelt. Den Schluss des Kapitels bildet Abschnitt 2.3, in dem kurz die Anbindung der Hardware an die Software verdeutlicht wird.

### 2.1. Konstruktion

Für die Realisierung des Robotermodells wurde nicht auf ein bestehendes Konzept zurückgegriffen, sondern ein eigenständiges Modell erarbeitet. Die Anforderungen an das Modell wurden wie folgt gestellt:

- Modell in horizontaler und vertikaler Lage verwendbar
- Freie bzw. unbeschränkte Wahl der Gelenkwinkel
- Verwendung von bestehenden Hardwarekomponenten im Labor
- Preiswerter Aufbau

Die mechanische Grundkonstruktion basiert auf dem Profilsystem von *Haberkorn Ulmer*. Diese Technik hat sich bereits bei vielen Laboraufbauten des Instituts bewährt. Der Aufbau kann manuell jederzeit in die Horizontallage gebracht werden. Um eine etwaige Verletzungsgefahr zu minimieren, kann an den Enden des Profilgerüsts beispielsweise eine Plexiglasscheibe montiert werden.

Das Herz des Aufbaus bilden jedoch die Antriebe für Ober- und Unterarm des Manipulators. Aufgrund vorhandener Bestände von Motoren und Getrieben wurde auf

Komponenten der Firma *Faulhaber* zurückgegriffen. Als Motoren werden permanente DC-Kleinstmotoren mit Graphitkommutierung verwendet. Diese garantieren hohe Dynamik und eine einfache Regelung aufgrund linearen Verhaltens. Um die benötigten Drehmomente zur Verfügung zu stellen, werden die Motoren mit Präzisionsgetrieben ausgestattet. Die Sensorik bilden 2 optische Inkrementalencoder, die zusammen mit Motor und Getriebe eine kompakte Einheit bilden. Die Typenbezeichnungen der verwendeten Komponenten können Tabelle 2.1 entnommen werden. Für genauere Spezifikationen sei auf den Anhang A.1 sowie das Handbuch [Fau10] verwiesen.

Antrieb OA		Antrieb UA	
Komponente	Bezeichnung	Komponente	Bezeichnung
Motor	3557 024 CS	Motor	2642 012 CR
Getriebe	30/1 4,5Nm	Getriebe	26/1 3,5Nm
Encoder	HEDS 5500 A	Encoder	HEDS 5500 A

Tabelle 2.1.: Antriebskomponenten

Eine weitere Herausforderung stellte die freie Wahl an Gelenkwinkeln dar. Um dieser Anforderung gerecht zu werden, wird ein Tellerschleifring der Firma *LTN Servotechnik* eingesetzt. Dieses elektromechanische Bauteil ermöglicht eine Entkopplung zwischen rotierendem und feststehendem Teil des Manipulators. Somit kann über die rotierende Achse des Oberarmantriebs eine Strom- und Signalübertragung zum Unterarm erfolgen. Der Tellerschleifring besitzt im Gegensatz zu handelsüblichen axialen Ausführungen eine radiale Bauform. Dadurch kann die Wellenbelastung des Oberarms minimiert werden.



Abbildung 2.1.: Tellerschleifringe

## *2. Mechanischer Aufbau des Roboters*

---

In Abbildung 2.2 ist das Labormodell des Roboters in vertikaler Lage ersichtlich. Am unteren Ende des Unterarms kann eine Bohrung festgestellt werden. Diese dient in Verbindung mit einem Gewindestift der Halterung eines Schreibwerkzeugs.

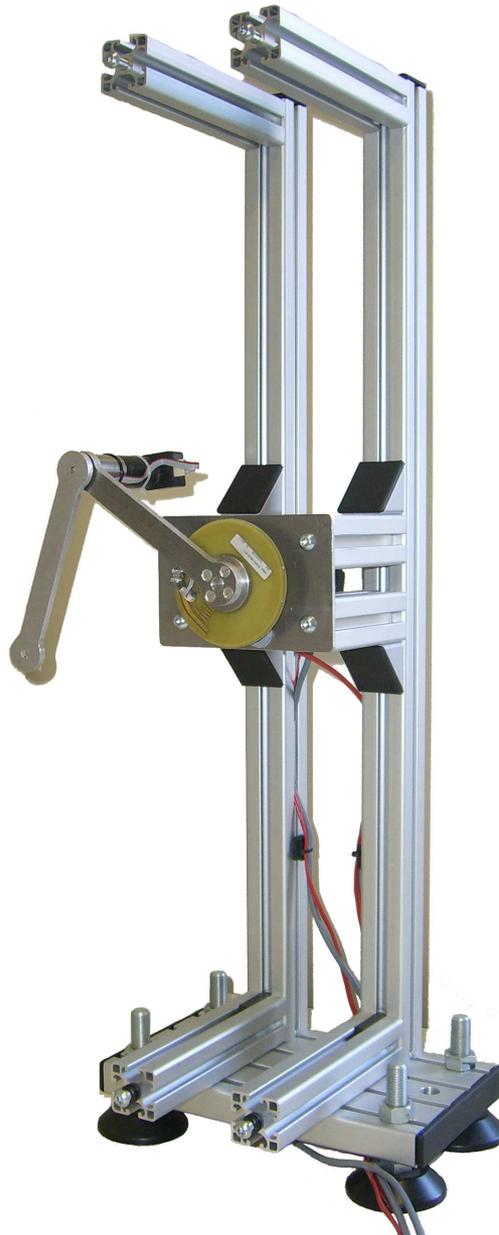


Abbildung 2.2.: Laboraufbau des Roboterarms

## 2.2. Ansteuerung

Die benötigte Spannungsversorgung der beiden Motoren wird durch 2 Leistungsverstärker der Firma *Quanser* zur Verfügung gestellt. Der Ober- und Unterarm wird durch die Modelle UPM-24-05 und PA-103 versorgt.

Motoransteuerung		
Typ	Max. Ausgangsspannung	Max. Ausgangsstrom
UPM-24-05	$\pm 24V$	5.0A
PA-103	$\pm 10V$	3.0A

Tabelle 2.2.: Spezifikation der Leistungsverstärker

Die analogen Ansteuersignale der Leistungsverstärker werden über ein Terminal Board der *E-Serie* von Quanser abgegriffen. Das Board dient weiters als Steckverbindung für die beiden Encoder von Ober- und Unterarm.

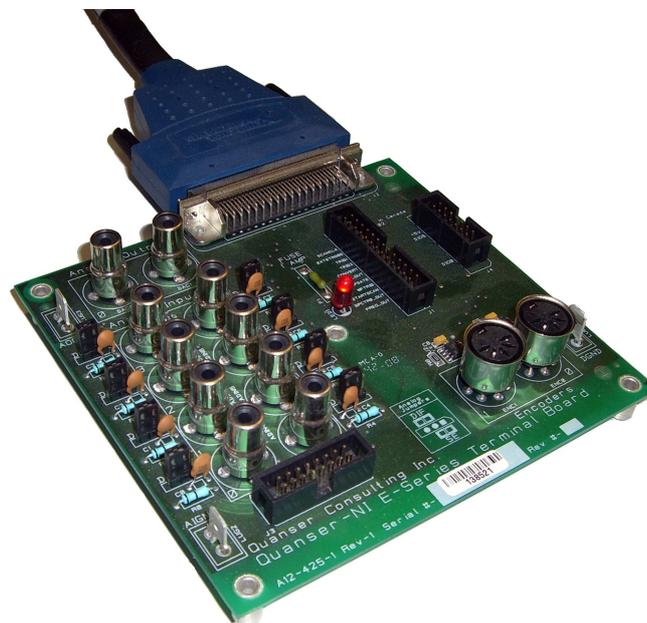


Abbildung 2.3.: Quanser Terminal Board

### 2.3. Quanser-Schnittstelle

Die komplette Steuerung des Modells erfolgt mittels der Messsignalverarbeitungskarte *NI PCI 6251 M-Series* von *National Instruments*. Die Abtastzeit zur Verarbeitung der Daten wird mit  $T_d = 1ms$  festgelegt. Mit Hilfe dieser Schnittstellenkarte wird es möglich, die Roboterapplikation näherungsweise in Echtzeit auszuführen. Die Sensorik kann dadurch permanent ausgewertet und verarbeitet werden. Die Anbindung der Hardware an die Software ermöglicht das Softwarepaket *QuaRC*, welches ebenfalls von der Firma Quanser entwickelt wird. Unter Zuhilfenahme dieses Designwerkzeugs lässt sich ein mathematisch ermitteltes Regelungskonzept umgehend aus Simulink in ein ausführbares, echtzeitfähiges Programm übersetzen. Der in die Programmiersprache *C* kompilierte Code läuft nicht als Task im Windows-System, sondern wird in einem eigenständigen Echtzeitbetriebssystem (*RTOS*) ausgeführt. Dadurch kann deterministisches Verhalten und folglich die verlässliche Abarbeitung der Sensorik und Ansteuerung der Antriebe gewährleistet werden.

## 3. Kinematik

Aus mechanischer Sicht kann ein Manipulator als kinematische Kette von Starrkörpern, welche durch Gelenke miteinander verbunden sind, interpretiert werden. Die Bewegung des Roboters ist dabei durch die Freiheitsgrade der Gelenke festgelegt. Mit Hilfe der *Grübler'schen Formel*<sup>1</sup> kann der Gesamtfreiheitsgrad eines Robotersystems berechnet werden.

Um eine konkrete Bewegung zu erfassen, müssen kinematische Beziehungen aufgestellt werden. Mittels dieser ist eine hinreichend genaue Beschreibung der mechanischen Struktur und der daraus resultierenden Bewegung möglich. Im Gegensatz zum dynamischen Modell, welches im nächsten Kapitel behandelt wird, werden in der Kinematik keine einwirkenden Kräfte und Drehmomente berücksichtigt. Das Kapitel beginnt mit einer kurzen Einführung der verwendeten Koordinatensysteme in Abschnitt 3.1. Anschließend wird in den zwei folgenden Abschnitten die direkte und indirekte Kinematik vorgestellt. Abschnitt 3.4 fokussiert sich auf die Jacobi-Matrix, welche für die Transformation zwischen den verwendeten Koordinatensystemen entscheidend ist. Weiters wird die Geschwindigkeits- und Beschleunigungs-Kinematik<sup>2</sup> und die damit verbundenen Probleme durch Singularitäten behandelt.

### 3.1. Koordinatensysteme

Für die geometrische Beschreibung der Position und Orientierung des Endeffektors muss festgelegt werden, in welchem Koordinatensystem die Daten dargestellt werden sollen.

*Basiskoordinatensystem:* Die Position kann mittels kartesischer Koordinaten (Weltkoordinaten) definiert werden.

*Gelenkkoordinatensystem:* Dieses System ist dadurch gekennzeichnet, dass die Position anhand der Gelenkwinkel angegeben werden kann.

Die Transformation zwischen den Koordinatensystemen ist durch die Abbildungsfunktionen der direkten und indirekten Kinematik gegeben. Abbildung 3.1 veranschaulicht die Transformationsvorschriften für den allgemeinen Fall. Die Parameter  $\alpha$ ,  $\beta$  und  $\gamma$

---

<sup>1</sup>Vgl. [Gfr10, S. 30]

<sup>2</sup>auch differentielle Kinematik 1. und 2. Ordnung genannt

charakterisieren die Orientierung des Koordinatensystems.

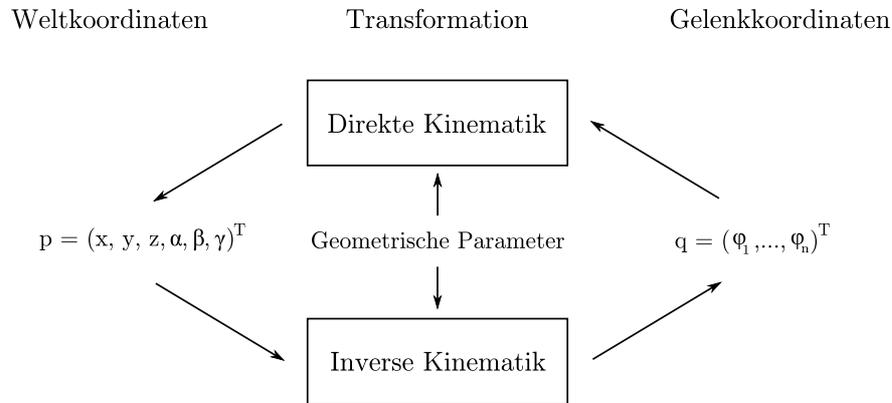


Abbildung 3.1.: Roboterkinematik

Der *Arbeitsraum* des Roboters stellt in diesem Zusammenhang einen besonders wichtigen Begriff in Bezug auf die Einsatzmöglichkeit dar. Er wird durch die Geometrie des Manipulators festgelegt und kann als Summe aller möglichen Punkte im Raum, die vom Roboter angefahren werden können, interpretiert werden. Dabei ist es aufgrund von Beschränkungen in Form von maximalen Drehwinkeln möglich, dass nicht der gesamt mögliche Arbeitsraum für die Bewegung nutzbar ist. In einer solchen Situation gilt es, etwaige Schwierigkeiten bereits bei der Planung der Bewegung zu berücksichtigen. Im hier vorgestellten Modell sind solche Beschränkungen durch die Verwendung eines Schleifrings jedoch nicht gegeben.

## 3.2. Direkte Kinematik

Mit Hilfe der direkten Kinematik oder auch Vorwärtskinematik können die Position und Orientierung des Endeffektors in Weltkoordinaten bei gegebenen Gelenkwinkeln berechnet werden. Aufgrund der relativ einfachen Struktur des Robotermodells können diese direkt angeschrieben werden. Vorgreifend auf die Behandlung des dynamischen Modells mittels generalisierter Koordinaten  $q_i$  in Abschnitt 4.1.1 werden diese Koordinaten in folgender Definition bereits verwendet. Zum besseren Verständnis der Geometrie und der Bedeutung der verwendeten Symbole sei auf Abbildung 3.2 verwiesen.

$$x = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \quad (3.1)$$

$$y = -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) \quad (3.2)$$

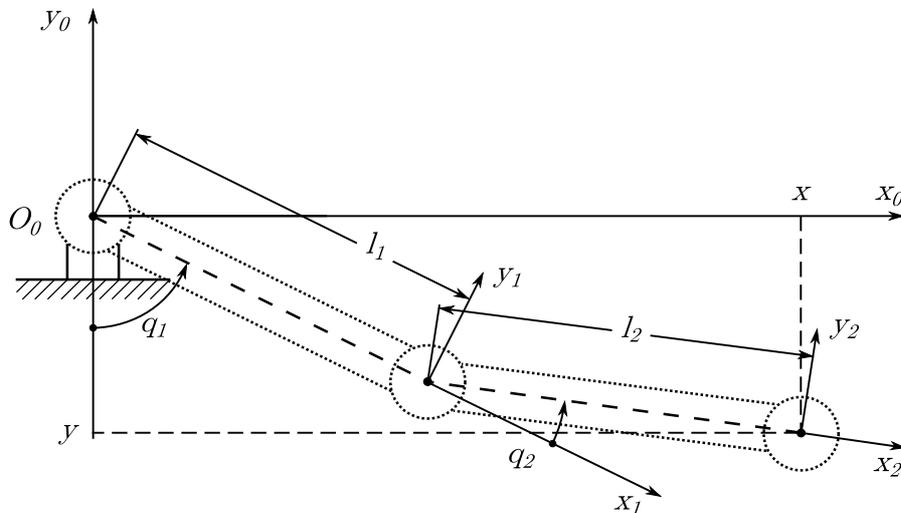


Abbildung 3.2.: 2-Gelenkarm Roboter

Um eine übersichtlichere Form zu erhalten wird die Vektorform eingeführt.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) \end{bmatrix} \quad (3.3)$$

### 3.3. Inverse Kinematik

Die Aufgabe der inversen Kinematik oder auch Rückwärtskinematik besteht in der Bestimmung der Gelenkwinkel für eine gegebene Position und Orientierung des Endeffektors in kartesischen Koordinaten. Aus mathematischer Sicht handelt es sich hierbei um die Lösung eines nichtlinearen Gleichungssystems der Form

$$\mathbf{q} = \mathbf{k}^{-1}(\mathbf{x}). \quad (3.4)$$

Im Vergleich zur direkten Kinematik ist die Ermittlung einer Lösung in diesem Fall deutlich komplexer und oft nicht eindeutig. Dies ergibt sich aufgrund:

- Eines i.A. nichtlinearen Problems, dessen Lösung meist nicht in geschlossener analytischer Form angebar ist.
- Mehrdeutigkeiten in den ermittelten Lösungen.
- Unendlich vielen Lösungen aufgrund von redundanten Konfigurationen.
- Keiner gültigen Lösung im Arbeitsraum des Roboters.

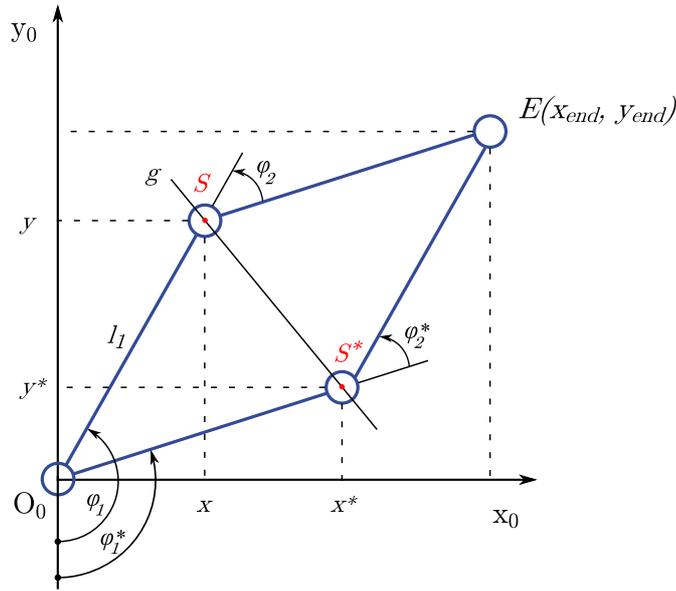


Abbildung 3.3.: Inverse Kinematik

Vor allem für komplexe Roboterstrukturen stellt die Ermittlung der inversen Kinematik eine mathematische Herausforderung dar. Es gilt, die erhaltene Lösung grundsätzlich auf Mehrdeutigkeit und Existenz zu überprüfen. Die inverse Kinematik des 2-Gelenk-Roboterarms birgt ebenfalls eine Mehrdeutigkeit. Aus Abbildung 3.3 ist ersichtlich, dass 2 Lösungen (*Elbow-Down*, *Elbow-Up*) zur Beschreibung der Position des Endeffektors möglich sind.

Der Ansatz zur Berechnung der inversen Kinematik beruht auf einer rein geometrischen Überlegung. Durch Kenntnis der Längen beider Roboterarme können die Punkte  $S$  und  $S^*$  ermittelt werden. Diese ergeben sich aus den Schnittpunkten zweier Kreise mit Mittelpunkt im Ursprung  $O_0$  und im Punkt  $E$  mit den jeweiligen Armlängen als Durchmesser. Mathematisch kann dies wie folgt formuliert werden:

$$\begin{aligned}
 \text{I: } & x^2 + y^2 = l_1^2 \\
 \text{II: } & (x - x_{end})^2 + (y - y_{end})^2 = l_2^2 \\
 \hline
 \text{I-II: } & 2xx_{end} - x_{end}^2 + 2yy_{end} - y_{end}^2 = l_1^2 - l_2^2
 \end{aligned} \tag{3.5}$$

Durch Umformen obiger Gleichung kann die Gerade ermittelt werden, welche die beiden Schnittpunkte miteinander verbindet.

$$g: \underbrace{2x_{end}x}_a + \underbrace{2y_{end}y}_b + \underbrace{l_2^2 - l_1^2 - x_{end}^2 - y_{end}^2}_c = 0 \tag{3.6}$$

Mit Hilfe dieser Geraden  $g$  und der in I angegebenen Formel kann das zu lösende Gleichungssystem angeschrieben werden.

$$\begin{aligned} a \cdot x + b \cdot y + c &= 0 \\ x^2 + y^2 - l_1^2 &= 0 \end{aligned} \tag{3.7}$$

Die hieraus resultierenden Lösungen  $(x, y)$  und  $(x^*, y^*)$  werden für die weitere Berechnung der Gelenkwinkel verwendet. Um korrekte Winkel in allen Quadranten zu gewährleisten, wird die modifizierte Arkustangens-Funktion  $\text{atan2}(y, x)$  verwendet. Durch Subtraktion eines Zusatzterms können der Wertebereich der Funktion und die Festlegung der Winkel in Abbildung 3.2 angepasst werden. Stellvertretend wird in Formel (3.8) die Lösung für  $(x, y)$  angeführt.

$$\begin{aligned} \varphi_1 &= \text{atan2}(y, x) + \frac{\pi}{2} \\ \varphi_2 &= \text{atan2}(y_{\text{end}} - y, x_{\text{end}} - x) - \varphi_1 + \frac{\pi}{2} \end{aligned} \tag{3.8}$$

Allerdings reicht die gerade gezeigte Berechnung der Winkel noch nicht aus, um in allen Quadranten korrekte Winkel zu erhalten. Für den Fall, dass  $x < 0$  und  $y < 0$  liefert die  $\text{atan2}$ -Funktion negative Winkel. Deshalb werden im implementierten Algorithmus Positionen im 4. Quadranten durch eine Abfrage abgefangen und durch eine Addition von  $2\pi$  korrigiert.

### 3.4. Jacobi-Matrix

Die Jacobi-Matrix oder Funktionalmatrix stellt in der Robotik eine wichtige charakteristische Größe des Gesamtsystems dar. Die Analysen von Singularitäten, redundanten Konfigurationen und regelungstechnischen Algorithmen beruhen auf dieser Matrix. Die Ermittlung erfolgt analytisch durch das totale Differential der direkten Kinematik aus Formel (3.3).

$$\begin{aligned} \mathbf{J}_A(\mathbf{q}) &= \frac{\partial \mathbf{x}(\mathbf{q})}{\partial \mathbf{q}} \quad \text{mit } \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\ &= \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) & l_2 \sin(q_1 + q_2) \end{bmatrix} \end{aligned} \tag{3.9}$$

Mit Hilfe der Funktionalmatrix  $\mathbf{J}_A$  kann eine lineare Abbildung der Form

$$\dot{\mathbf{x}} = \mathbf{J}_A \dot{\mathbf{q}} \tag{3.10}$$

angegeben werden, welche auch als Geschwindigkeits-Kinematik bezeichnet wird. Um aus den linearen Geschwindigkeiten  $\dot{\mathbf{x}}$  auf die Gelenkgeschwindigkeiten schließen zu können, muss die Inverse  $\mathbf{J}_A^{-1}$  gebildet werden. Hierfür muss die Matrix regulär und somit vollen Rang besitzen. Eine Rangreduktion in einer entsprechenden Position kann als Verlust eines Freiheitsgrads interpretiert werden. Im Fachjargon spricht man in diesem Fall von einer singulären Position.

$$\mathbf{J}_A^{-1} = \frac{1}{l_1 l_2 \sin(q_2)} \begin{bmatrix} l_2 \sin(q_1 + q_2) & -l_2 \cos(q_1 + q_2) \\ -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \end{bmatrix} \quad (3.11)$$

Für das in Abschnitt 7.3 vorgestellte Regelungskonzept wird eine weitere Transformation der Beschleunigungen benötigt. Zur Ermittlung wird (3.10) nach der Zeit abgeleitet.

$$\ddot{\mathbf{x}} = \mathbf{J}_A \ddot{\mathbf{q}} + \dot{\mathbf{J}}_A \dot{\mathbf{q}} \quad (3.12)$$

Formel (3.10) und (3.12) ermöglichen eine Transformation von Geschwindigkeit und Beschleunigung zwischen Gelenk- und Arbeitsraum. Besonderes Augenmerk muss dabei auf die Determinante der inversen Jacobi-Matrix gelegt werden, da sie eine entscheidende Rolle für die korrekte Transformation darstellt. Der Vollständigkeit halber ist in Formel (3.13) die Ableitung der Jacobi-Matrix angegeben.

$$\dot{\mathbf{J}}_A = \begin{bmatrix} -l_1 \sin(q_1) \cdot \dot{q}_1 - l_2 \sin(q_1 + q_2) \cdot (\dot{q}_1 + \dot{q}_2) & -l_2 \sin(q_1 + q_2) \cdot (\dot{q}_1 + \dot{q}_2) \\ l_1 \cos(q_1) \cdot \dot{q}_1 + l_2 \cos(q_1 + q_2) \cdot (\dot{q}_1 + \dot{q}_2) & l_2 \cos(q_1 + q_2) \cdot (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \quad (3.13)$$

### 3.5. Singularitäten

Eine Roboterkonfiguration, bei der es zu einer Rangreduktion der Jacobi-Matrix kommt, wird als singuläre Position bezeichnet. Die Ermittlung von Singularitäten ist aus vielen Gründen von entscheidender Bedeutung.

- In einer singulären Position ist die Bewegungsfreiheit des Roboters beschränkt.
- Beschränkte Geschwindigkeiten im Arbeitsraum korrespondieren mit unbeschränkten Geschwindigkeiten im Gelenkraum. Selbiges gilt auch für etwaige Kräfte.
- In der Nähe von Singularitäten existiert keine eindeutige Lösung der inversen Kinematik. Es können einerseits unendlich viele oder gar keine existieren.

Singularitäten werden anhand von *Rand-* und *inneren Singularitäten* klassifiziert.

*Randsingularität:* Diese treten am Rand des Arbeitsbereichs bei voll ausgestrecktem bzw. eingezogenem Arm auf.

*innere Singularität:* Bei mehrachsigen Robotern können Singularitäten auch innerhalb des Arbeitsraums auftreten. Im hier behandelten 2-achsigen Roboterarm ist dies jedoch nicht der Fall.

Die Eigenschaft der Regularität einer quadratischen Matrix kann mittels der Determinante überprüft werden. Die Jacobi-Matrix erfüllt im vorliegenden Fall diese Eigenschaft für

$$\det(\mathbf{J}_A) = l_1 l_2 \sin(q_2) \neq 0. \quad (3.14)$$

Somit ist ersichtlich, dass die Inverse nicht existent ist für Konfigurationen

$$q_2 = n \cdot \pi \quad n = 0, \dots, \infty. \quad (3.15)$$

## 4. Dynamische Modellbildung

Dieses Kapitel hat die Ermittlung eines mathematischen Modells zum Ziel, welches die Dynamik des 2-Gelenkarm Roboters beschreibt. Die Behandlung der Dynamik mechanischer Systeme kann durch verschiedene Formalismen erfolgen. Die Methoden basieren dabei auf denselben Grundlagen, allerdings findet die Betrachtung des Modells aus verschiedenen Blickwinkeln statt. In der Robotik finden im Wesentlichen der Euler-Lagrange und der Newton-Euler -Formalismus Anwendung.

Bei Verwendung des Newtonschen Prinzips werden die Bewegungsgleichungen für jeden Körper des Systems einzeln betrachtet. Dieser Ansatz hat den Vorteil, dass die Kinematik und die geometrische Anordnung des Systems ohne weitere Aufbereitung verwendet werden kann. Für die Simulation ergibt sich ein differential-algebraisches Gleichungssystem oder Deskriptor-System<sup>1</sup>. Dieses Gleichungssystem zu lösen führt zu erhöhtem Rechenaufwand und kann je nach Beschaffenheit der Zwangsbedingungen zu numerischen Problemen führen.

Das Lagrangsche Prinzip erfasst die Dynamik unmittelbar für generalisierte Koordinaten. Es müssen zu Beginn die kinematischen Größen auf generalisierte Koordinaten umgerechnet werden. Die Berechnung ist systematisch in kompakter Form durchführbar. Der große Vorteil liegt in der minimalen Anzahl an Gleichungen für die Beschreibung der Bewegung des Systems. Als Ergebnis erhält man  $n$  gewöhnliche Differentialgleichung 2. Ordnung<sup>2</sup>. Der Rechenaufwand zur Lösung des DGL-Systems ist vergleichsweise gering. Dies rührt einerseits aus der minimalen Systemordnung und andererseits aus der Vermeidung von mitgeführten algebraischen Gleichungen her.

Beide Formalismen führen zum selben Ergebnis, bieten allerdings unterschiedliche Einsichten zur Analyse des Modells. Im Rahmen dieser Masterarbeit wird die Modellierung mittels des Lagrangschen Formalismus durchgeführt. Für die weitere Behandlung wird der Roboter als starres Mehrkörpersystem betrachtet. Auf eine zusätzliche Modellierung von Elastizität in den Gelenken wird hier verzichtet.

Die Ermittlung der Bewegungsgleichungen wird mit Hilfe des Programmpackets *Mathematica* rechnergestützt durchgeführt.

---

<sup>1</sup>Ein Deskriptor-System besteht aus gewöhnlichen DGLn und algebraischen NBn.

<sup>2</sup> $n$  entspricht der Anzahl der Freiheitsgrade des Systems

Im ersten Abschnitt dieses Kapitels wird mit Hilfe der Lagrange-Methode das Modell abgeleitet. Anschließend wird in einem weiteren Abschnitt die Dynamik der Gleichstrommotoren, welche für die entsprechenden Drehmomente sorgen, behandelt. Der dritte Abschnitt widmet sich der Reibungsmodellierung für den Ober- bzw. den Unterarm. Schlussendlich wird die Simulation und die zur besseren Veranschaulichung implementierte Animation erläutert.

## 4.1. Methode von Lagrange

Der Lagrange Formalismus ermöglicht die systematische Ermittlung der Bewegungsgleichungen unabhängig vom Bezugskoordinatensystem. Durch die Wahl von  $n$  generalisierten Koordinaten kann die Bewegung jedes einzelnen Arms vollständig beschrieben werden.

Grundsätzlich hat ein System bestehend aus  $N$  Massepunkten im 3-dimensionalen Raum  $(3N - k)$ -Freiheitsgrade, wobei  $k$  die Anzahl der holonomen Zwangsbedingungen beschreibt. Das hier behandelte System eines planaren Roboters beschränkt sich durch die reine Bewegung in der Ebene bereits auf  $(2N - k)$ -Freiheitsgrade. Im allgemeinen Fall von  $N$  Massepunkten definiert eine Gleichung der Form

$$g(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N, t) = 0 \quad (4.1)$$

eine holonome Zwangsbedingung. Die Vektoren  $\vec{r}_i$  entsprechen hierbei Ortsvektoren zu den  $N$  Massepunkten. Falls Gleichung (4.1) nicht explizit von der Zeit abhängig ist, wird sie als skleronome, andernfalls als rheonome Zwangsbedingung bezeichnet. Holonome Zwangsbedingungen führen zu einer Reduktion des Freiheitsgrades, was im weiteren Sinne bedeutet, dass die Ortskoordinaten nicht mehr unabhängig voneinander sind.

Die Zwangsbedingungen des Roboters ergeben sich aus den 2 Gleichungen:

$$x_{S1}^2 + y_{S1}^2 = l_{S1}^2 \quad (4.2)$$

$$x_{S2}^2 + y_{S2}^2 = l_{S2}^2 \quad (4.3)$$

Für das verallgemeinerte System mit  $N = 2$  Massepunkten ergeben sich somit 2 Freiheitsgrade. Zunächst liegt die Problemstellung darin,  $n$  generalisierte Koordinaten zu finden, die diese Freiheitsgrade vollständig beschreiben.

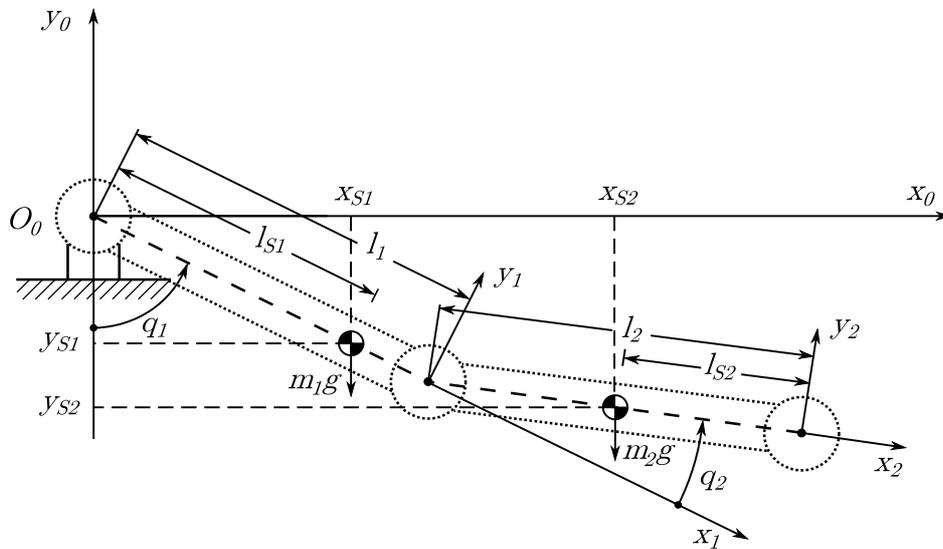


Abbildung 4.1.: 2-Gelenkarm Roboter

#### 4.1.1. Wahl der generalisierten Koordinaten

Die Festlegung der  $n$  generalisierten Koordinaten  $q_1, \dots, q_n$  erfolgt nach folgenden Kriterien:

- Für beliebige Werte der generalisierten Koordinaten sind die holonomen Zwangsbedingungen erfüllt. Das heißt, die Koordinaten  $q_i$  selbst unterliegen keinen Beschränkungen.
- Alle Ortsvektoren können eindeutig durch die generalisierten Koordinaten  $q_i$  dargestellt werden.

Eine mögliche Wahl der  $n = 2$  generalisierten Koordinaten sind die beiden Gelenkwinkel, wie in Abbildung 4.1 dargestellt.

#### 4.1.2. Potentielle- und kinetische Energie

Die kinetische Energie der beiden Roboterarme setzt sich aus 2 Termen zusammen. Der erste Term beschreibt die Energie der Rotationsbewegung um den Schwerpunkt und kann aus dem Trägheitsmoment  $J_i$  für eine Drehung um die Drehachse durch den Schwerpunkt sowie der Winkelgeschwindigkeit  $\dot{\varphi}_i$  gebildet werden. Neben diesem Term beschreibt der zweite Term die Energie der Translation.

Somit erhält man für die kinetische Energie des starren Körpers.

$$T_i = \frac{1}{2} J_i \dot{\varphi}_i^2 + \frac{1}{2} m_i (\dot{x}_{S_i}^2 + \dot{y}_{S_i}^2) \quad i = 1, \dots, n \quad (4.4)$$

Die potentielle Energie ergibt sich zu

$$V_i = -m_i g y_{S_i} \quad i = 1, \dots, n, \quad (4.5)$$

wobei sich die Gewichtskräfte durch den negativen Gradienten der potentiellen Energie ergeben. Die Gesamtenergie  $T$  bzw.  $V$  des Systems ergibt sich aus der Summe der Einzelenergien.

$$T = \sum_{i=1}^n T_i, \quad V = \sum_{i=1}^n V_i \quad (4.6)$$

Um eine Darstellung in generalisierten Koordinaten zu erhalten, müssen die kartesischen Koordinaten ersetzt werden. Durch diese Maßnahme wird die Kopplung zwischen den Roboterarmen berücksichtigt.

$$\begin{aligned} x_{S1} &= l_{S1} \sin(q_1) & x_{S2} &= l_1 \sin(q_1) + l_{S2} \sin(q_1 + q_2) \\ y_{S1} &= -l_{S1} \cos(q_1) & y_{S2} &= -l_1 \cos(q_1) - l_{S2} \cos(q_1 + q_2) \end{aligned} \quad (4.7)$$

$$\begin{aligned} \dot{x}_{S1} &= l_{S1} \cos(q_1) \dot{q}_1 & \dot{x}_{S2} &= l_1 \cos(q_1) \dot{q}_1 + l_{S2} \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ \dot{y}_{S1} &= l_{S1} \sin(q_1) \dot{q}_1 & \dot{y}_{S2} &= l_1 \sin(q_1) \dot{q}_1 + l_{S2} \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \end{aligned} \quad (4.8)$$

### 4.1.3. Lagrangesche Gleichungen 2. Art

Die Dynamik eines starren Mehrkörpersystems mit holonomen Zwangsbedingungen lässt sich mit Hilfe der Lagrangeschen Gleichungen 2. Art darstellen.

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_j \quad j = 1, \dots, n \quad (4.9)$$

Hierbei beschreibt  $Q_j$  die  $j$ -te generalisierte Kraft, und ist durch

$$Q_j := \sum_{i=1}^N \vec{F}_i^* \frac{\partial}{\partial q_j} \quad j = 1, \dots, n \quad (4.10)$$

definiert, wobei  $\vec{F}_i^*$  die am Massepunkt  $i$  angreifende äußere Kraft darstellt.

Weiters bezeichnet  $L$  die Lagrange-Funktion, welche sich aus kinetischer- und potentieller Gesamtenergie zusammensetzt:

$$L = T - V \quad (4.11)$$

Durch Verwendung der Formeln (4.4) - (4.8) kann die Lagrange Funktion berechnet werden.

$$\begin{aligned} L = & g \cos(q_1 + q_2) l_{S2} m_2 + g \cos(q_1) (l_{S1} m_1 + l_1 m_2) \\ & + \frac{1}{2} \left[ J_1 + J_2 + l_{S1}^2 m_1 + 2 \cos(q_2) l_1 l_{S2} m_2 + (l_1^2 + l_{S2}^2) m_2 \right] \dot{q}_1^2 \\ & + \left[ J_2 + \cos(q_2) l_1 l_{S2} m_2 + l_{S2}^2 m_2 \right] \dot{q}_1 \dot{q}_2 + \frac{1}{2} (J_2 + l_{S2}^2 m_2) \dot{q}_2^2 \end{aligned} \quad (4.12)$$

In der Robotik ist es aufgrund der Komplexität der Ausdrücke üblich, eine Darstellung in Matrizen-Schreibweise zu verwenden.

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \boldsymbol{\tau}_R \quad (4.13)$$

Das Gleichungssystem in (4.13) repräsentiert die Differentialgleichungen, mit welchen die Bewegung des Roboters mathematisch beschrieben werden kann. Durch die spezielle Wahl der Koordinaten  $q_1$  und  $q_2$  entspricht der Vektor der generalisierten Kräfte  $\boldsymbol{\tau}$  direkt den Antriebsmomenten der Motoren. Entgegen diesem Moment wirkt ein Reibungsmoment, das an dieser Stelle mit  $\boldsymbol{\tau}_R$  bezeichnet wird. Auf der linken Seite der Gleichung bezeichnet  $\mathbf{M}(\mathbf{q})$  die symmetrische und positiv definite Massenmatrix. Der Vektor  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  beinhaltet Zentrifugal- und Coriolisterme, welche sich aus der Koppung der Winkelgeschwindigkeiten ergeben. Der letzte Term  $\mathbf{g}(\mathbf{q})$  fasst schließlich die Gravitationsterme in einem Vektor zusammen.

Um dem interessierten Leser die Komplexität des Gleichungssystems in (4.13) zu verdeutlichen, sind im Folgenden die Ausdrücke für die linke Seite angegeben.

$$\begin{aligned} \mathbf{M} = & \begin{bmatrix} J_1 + J_2 + l_{S1}^2 m_1 + 2 \cos(q_2) l_1 l_{S2} m_2 + (l_1^2 + l_{S2}^2) m_2 & J_2 + \cos(q_2) l_1 l_{S2} m_2 + l_{S2}^2 m_2 \\ J_2 + \cos(q_2) l_1 l_{S2} m_2 + l_{S2}^2 m_2 & J_2 + l_{S2}^2 m_2 \end{bmatrix} \\ \mathbf{C} = & \begin{bmatrix} -\sin(q_2) l_1 l_{S2} m_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \\ l_{S2} m_2 \sin(q_2) l_1 (\dot{q}_1)^2 \end{bmatrix} \\ \mathbf{g} = & \begin{bmatrix} g \sin(q_1 + q_2) l_{S2} m_2 + g \sin(q_1) (l_{S1} m_1 + l_1 m_2) \\ l_{S2} m_2 g \sin(q_1 + q_2) \end{bmatrix} \end{aligned} \quad (4.14)$$

## 4.2. Dynamik der Antriebe

Beide Gelenke des Roboterarms sind mit permanenten Gleichstrommotoren ausgestattet. Die Modellierung der Dynamik soll anhand Abbildung 4.2 verdeutlicht werden.

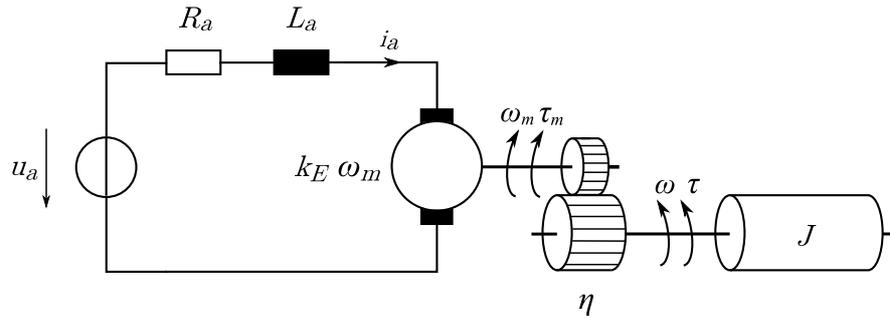


Abbildung 4.2.: Modellierung der Antriebe

Die Abbildung unterliegt der Annahme, dass Getriebebespiel und Elastizität der Gelenke vernachlässigt werden können. Im Folgenden sind die für den DC-Motor üblichen Modellgleichungen angeführt. Die exakte Begriffsdefinition der verwendeten Größen ist Anhang (A.1) zu entnehmen.

DC-Motor:

$$L_a \frac{di_a}{dt} + R_a i_a + k_E \omega_m = u_a \quad (4.15)$$

Motormoment:

$$\tau_m = k_M i_a \quad (4.16)$$

Ideales Getriebe:

$$\eta = \frac{\omega_m}{\omega} = \frac{\tau}{\tau_m} \quad (4.17)$$

Aufgrund der hohen Dynamik des Ankerstromes, welche sich in der sehr kleinen elektrischen Zeitkonstante  $\frac{R_m}{L_m}$  widerspiegelt, kann der Term  $L_a \frac{di_a}{dt}$  in (4.15) vernachlässigt werden. Mit Hilfe dieser Vereinfachung kann ein algebraischer Ausdruck für das Antriebsmoment, bezogen auf die Abtriebseite, ermittelt werden.

$$\tau = \frac{\eta k_M}{R_a} [-\eta k_E \omega + u_a] \quad (4.18)$$

Formel (4.18) kann nun in den Differentialgleichungen (4.13) eingesetzt werden, wodurch das System die Ankerspannungen des Motors als Eingangsgrößen erhält.

### 4.3. Zustandsraummodell

Für die weitere Behandlung der Bewegungsgleichungen ist es zweckmäßig, bestimmte Terme zusammenzufassen. Durch diese Maßnahme kann eine kompakte Notation in der Form eines Zustandsraummodells erreicht werden. Für das Antriebsmoment  $\tau$  wird anhand Formel (4.18) die Matrizen-Schreibweise eingeführt. Hierzu werden zusätzliche Konstanten eingesetzt, um eine anschaulichere Form zu erhalten.

$$\begin{aligned} C_1 &= \frac{-\eta_1^2 k_{M_1} k_{E_1}}{R_{a_1}} & C_3 &= \frac{-\eta_2^2 k_{M_2} k_{E_2}}{R_{a_2}} \\ C_2 &= \frac{-\eta_1 k_{M_1}}{R_{a_1}} & C_4 &= \frac{-\eta_2 k_{M_2}}{R_{a_2}} \end{aligned} \quad (4.19)$$

Weiters wird ein Vektor für die Winkelgeschwindigkeiten  $\dot{\mathbf{q}}$  sowie für die Motorspannungen  $\mathbf{u}$  eingeführt

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad \text{und} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (4.20)$$

wodurch sich das Antriebsmoment neu anschreiben lässt.

$$\boldsymbol{\tau} = \begin{bmatrix} C_1 \dot{q}_1 + C_2 u_1 \\ C_3 \dot{q}_2 + C_4 u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} C_1 & 0 \\ 0 & C_3 \end{bmatrix}}_{\mathbf{C}_{13}} \dot{\mathbf{q}} + \underbrace{\begin{bmatrix} C_2 & 0 \\ 0 & C_4 \end{bmatrix}}_{\mathbf{C}_{24}} \mathbf{u} \quad (4.21)$$

Durch Einsetzen der neu ermittelten Ausdrücke und Umformung auf die Motorspannungen  $\mathbf{u}$  kann folgendes Resultat erzielt werden.

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{D}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{C}_{13}\dot{\mathbf{q}} + \mathbf{C}_{24}\mathbf{u} - \boldsymbol{\tau}_R \\ \underbrace{\mathbf{C}_{24}^{-1} \mathbf{M}(\mathbf{q})}_{\bar{\mathbf{M}}(\mathbf{q})} \ddot{\mathbf{q}} + \underbrace{\mathbf{C}_{24}^{-1} [\mathbf{D}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_R - \mathbf{C}_{13}\dot{\mathbf{q}}]}_{\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})} &= \mathbf{u} \\ \bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{u} \end{aligned} \quad (4.22)$$

Mit den Vektoren  $\mathbf{x}_1$  und  $\mathbf{x}_2$  in der Form

$$\mathbf{x}_1 = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (4.23)$$

gelangt man zur Darstellung der Bewegungsgleichungen im Zustandsraum:

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= -\bar{\mathbf{M}}(\mathbf{x}_1)^{-1} \mathbf{h}(\mathbf{x}_1, \mathbf{x}_2) + \bar{\mathbf{M}}(\mathbf{x}_1)^{-1} \mathbf{u} \end{aligned} \quad (4.24)$$

Obige Gleichungen bilden die Grundlage für die Simulation des Manipulators in Simulink.

#### 4.4. Reibungsmodellierung

Die Beschreibung der Reibung stellt sicherlich einen der schwierigsten Aspekte der Modellbildung dar. Die Effekte, die durch Reibung verursacht werden, beeinflussen das dynamische Verhalten des Roboters enorm. Die typischen Modelle, die in der Literatur vorgeschlagen werden, so genannte klassische Reibungsmodelle, berücksichtigen Coulomb-, Viskose- und Stribeck-Komponenten [Wai07].

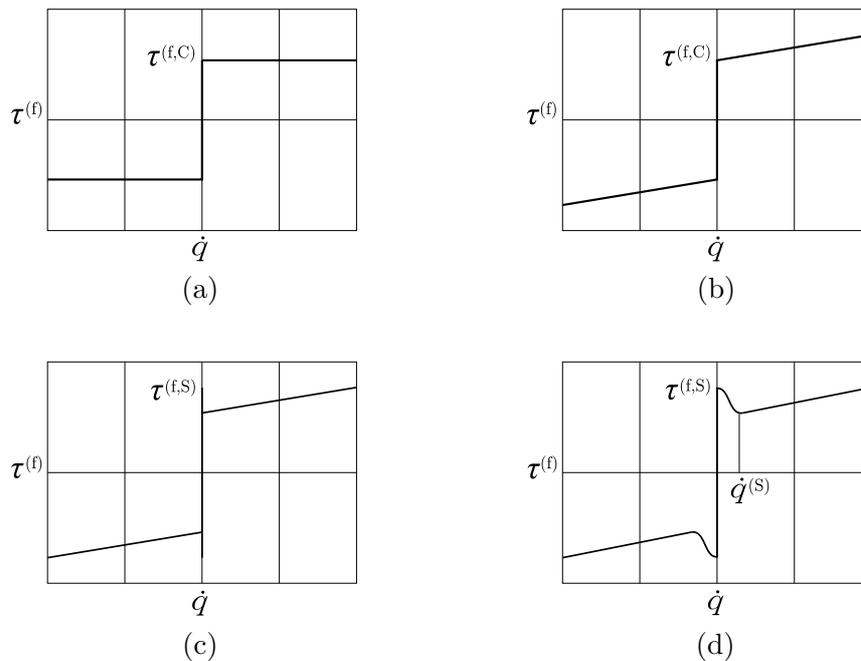


Abbildung 4.3.: Klassische Reibungsmodelle

Diese ermöglichen es, das Reibungsmoment als Funktion der Winkelgeschwindigkeit darzustellen. In der Robotik werden üblicherweise Kombinationen aus den einzelnen Reibungskomponenten verwendet. Das Modell in Abbildung 4.3(b) zeigt eine Kombination aus Coulomb'scher- und Viskoser Reibung. Die Coulomb'sche Reibung kann durch ein konstantes Moment, welches abhängig vom Vorzeichen der Winkel- bzw. Motorgeschwindigkeit ist, beschrieben werden. Die viskose Komponente bewirkt ein der Drehgeschwindigkeit proportionales Reibungsmoment. Die mathematische Formulierung des Reibmodells kann durch

$$\tau^{(f)} = \text{sign}(\dot{q}) \tau^{(f,C)} + c^{(v)} \dot{q} \quad (4.25)$$

angegeben werden.

Der Schweizer Wissenschaftler Euler<sup>3</sup> stellte fest, dass es zwischen Gleitreibung und Haftreibung zu unterscheiden gilt. Er erweiterte das Modell aus Formel (4.25) um einen zusätzlichen statischen Term. Dieser sogenannte Haftreibungsterm wird in Abbildung 4.3(c) berücksichtigt.

$$\begin{aligned} \dot{q} = 0 : & \quad |\tau^{(f)}| = \tau^{(f,S)} \\ \dot{q} \neq 0 : & \quad \tau^{(f)} = \text{sign}(\dot{q}) \tau^{(f,C)} + c^{(v)} \dot{q} \end{aligned} \quad (4.26)$$

Die Modellierung berücksichtigt den Effekt, dass für eine Bewegung aus dem Stillstand mehr Kraft benötigt wird, als bei einem sich bereits bewegenden Körper.

Die letzte Abbildung 4.3(d) zeigt ein Modell, in dem zusätzlich der Stribeck-Effekt berücksichtigt wird. Der deutsche Maschinenbau-Ingenieur Stribeck<sup>4</sup> fand heraus, dass der Übergang zwischen Haftreibung und Coulombscher Reibung bei geschmierten Oberflächen nicht diskontinuierlich verläuft, wie in Abbildung 4.3(c) dargestellt, sondern mit Hilfe einer Mischreibung modelliert werden kann. Armstrong-Hélouvry [AH91] adaptierte das Stribeck-Modell und formulierte das Reibungsmodell mit Hilfe folgenden Ausdrucks:

$$\tau^{(f)} = \text{sign}(\dot{q}) \left( \tau^{(f,C)} + (\tau^{(f,S)} - \tau^{(f,C)}) e^{-|\dot{q}/\dot{q}^{(S)}|^{\delta^{(S)}}} \right) + c^{(v)} \dot{q} \quad (4.27)$$

Hierbei beschreibt  $\dot{q}^{(s)}$  die sogenannte Stribeck-Geschwindigkeit, welche den Bereich der Mischreibung definiert. Die unbekannt Parameter  $\tau^{(f,C)}$ ,  $\tau^{(f,S)}$ ,  $\dot{q}^{(S)}$ ,  $\delta^{(S)}$  und  $c^{(v)}$  müssen aus geeigneten Experimenten ermittelt werden.

---

<sup>3</sup>Leonard Euler (★1707; +1783)

<sup>4</sup>Richard Stribeck (★1861; +1950)

Das Reibungsmodell des 2-Gelenkarm Roboters wurde in einem zyklischen Prozess entwickelt. Um die Komplexität der mathematischen Beschreibung nicht unnötig aufzublähnen, wurde von einem einfachen Modell ausgegangen und dieses iterativ angepasst.

Dabei wurde der folgende Grundsatz für die Modellierung verfolgt:

„So einfach wie möglich, so kompliziert wie nötig. Das einfachste, aber gültige Modell ist stets das Beste.“ [Bre88, S. 7]

Ober- und Unterarm des Roboters wurden dabei getrennt voneinander behandelt. Besonderes Augenmerk wurde auf die Modellierung der Reibung des Unterarms gelegt, da diese für die tatsächliche Position des Endeffektors einen großen Einfluss hat.

#### 4.4.1. Oberarm

Für das Reibungsmodell des Oberarms wurde Formel (4.25) verwendet. In der Simulation des Modells führte die diskontinuierliche Signum-Funktion allerdings zu numerischen Problemen. Deshalb wurde die Funktion durch einen Arkustangens angenähert. Die Arkustangens-Funktion wird in Abhängigkeit einer Konstanten  $a$  gewählt, welche das Übergangverhalten beim Vorzeichenwechsel der Drehgeschwindigkeit festlegt.

$$\tau_{R,oa} = \tau_{R,C_1} \cdot \frac{2}{\pi} \arctan(a\dot{q}_1) + k_{R1}\dot{q}_1 \quad (4.28)$$

Die verwendete Reibungskennlinie des Oberarms ist in Abbildung 4.4 dargestellt.

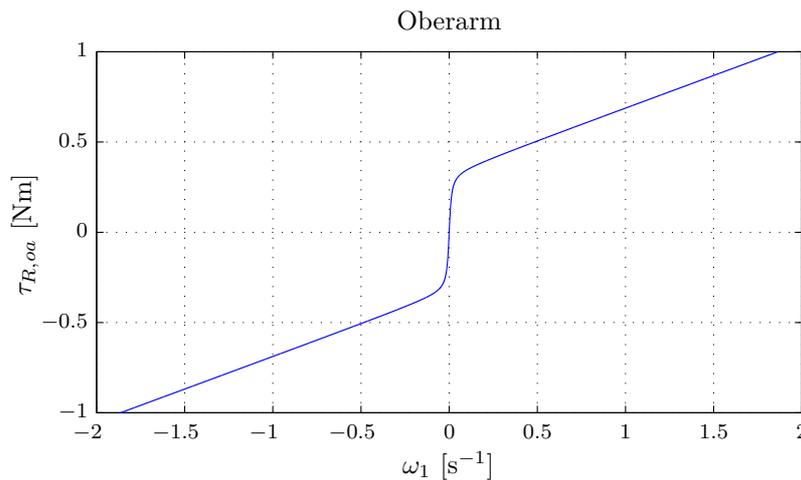


Abbildung 4.4.: Reibungskennlinie des Oberarms

#### 4.4.2. Unterarm

Für die Modellierung wurde ein ähnliches Modell wie in Formel (4.27) verwendet. Wie auch beim Oberarm bilden die ersten 2 Terme Coulomb'sche- und Viskoser-Reibung. Der zusätzliche 3. Term modelliert eine Mischreibung.  $\tau_{R,S_2}$  und  $\alpha$  repräsentieren den Stribeck-Effekt, welcher beim Übergang zwischen Coulombscher- auf Viskose-Reibung zum Tragen kommt. Mit Hilfe des Parameters  $\beta$  kann ein sanfter Übergang zwischen positiven und negativen Winkelgeschwindigkeiten erreicht werden.

$$\tau_{R,ua} = \tau_{R,C_2} \cdot \frac{2}{\pi} \arctan(a\dot{q}_2) + k_{R2}\dot{q}_2 + \tau_{R,S_2} \frac{\tanh(\beta\dot{q}_2)}{\cosh(\alpha\dot{q}_2)} \quad (4.29)$$

Mit Hilfe obiger Formulierung kann die Reibungskennlinie, wie in Abbildung 4.5 dargestellt, ermittelt werden.

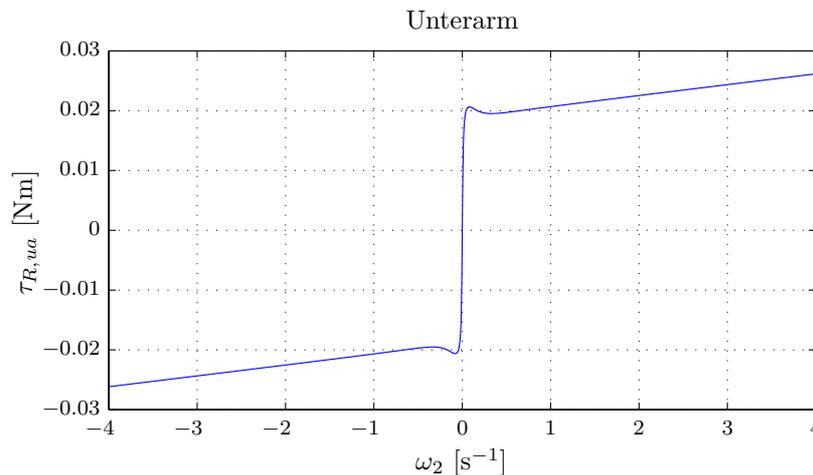


Abbildung 4.5.: Reibungskennlinie des Unterarms

### 4.5. Simulation und Animation

Die Simulation und Analyse des Roboterarms wurden mit Hilfe von *Simulink* durchgeführt. Hierzu wurden die Bewegungsgleichungen aus Mathematica portiert, um sie in MATLAB weiterverwenden zu können. Mit Hilfe dieser war es möglich, eine erste Untersuchung bzgl. der Systemeigenschaften durchzuführen. Die numerische Lösung mittels Simulink liefert zeitliche Verläufe für Gelenkwinkel und Winkelgeschwindigkeiten. Diese können interpretiert werden, allerdings ist die Vorstellung der zusammengesetzten Bewegung mühsam. Aus diesem Grund wurde eine Animation der Bewegung (siehe Abbildung 4.6) programmiert, welche es dem Anwender ermöglicht, die Dynamik des Roboters nachzuvollziehen.

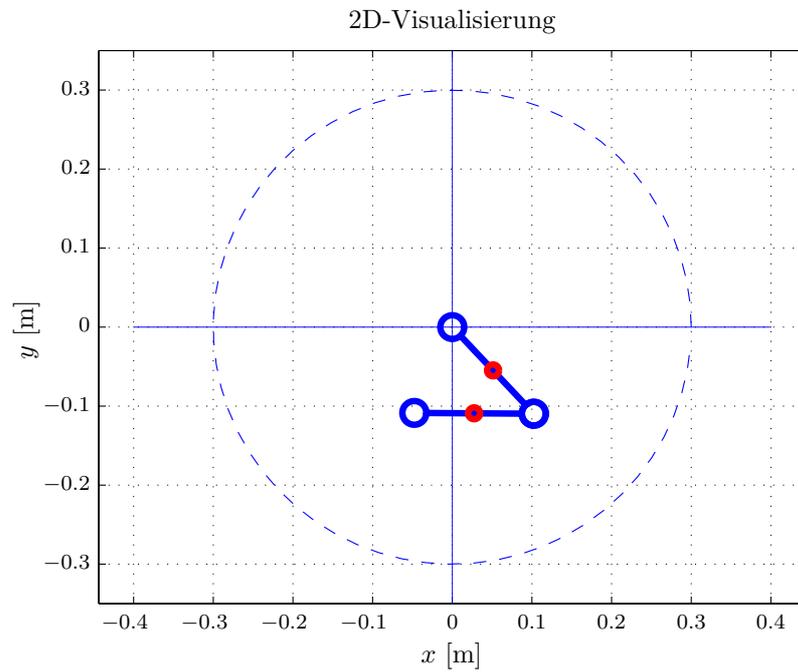


Abbildung 4.6.: Animation des Roboterarms

In obiger Grafik sind beide Roboterarme sowie ein Kreis ersichtlich, welcher den Arbeitsraum des Roboters veranschaulicht.

## 5. Parameterermittlung

Dieses Kapitel behandelt die Bestimmung der zur vollständigen Modellierung benötigten unbekannt Parameter. Die Ermittlung dieser Parameter ist für die Regelung von entscheidender Bedeutung. In den folgenden Abschnitten wird grundsätzlich zwischen analytischer- und experimenteller Parameterermittlung unterschieden.

Zu Beginn werden in Abschnitt 5.1 die Starrkörperparameter bestimmt. Anschließend wird in 5.2 auf die Identifikation der Reibparameter eingegangen. Zum Schluss findet in Abschnitt 5.3 ein Vergleich zwischen dem in Simulink simulierten Modell und den realen Experimenten am Roboter statt.

### 5.1. Analytische Identifikation

Die Bestimmung der Parameter des Starrkörpermodells erfolgt durch physikalische Experimente und Berechnung. Zur vollständigen analytischen Identifikation gehören unter anderem Masse, Schwerpunkt sowie Trägheitsmoment jedes einzelnen Teilkörpers. Für die Messung des Gewichts wurde jeder Teilkörper einzeln und in Kombination mit Motor, Getriebe und Encoder gemessen. Die Masse  $m_Z$  beschreibt das Gesamtgewicht von Motor, Getriebe und Encoder des Unterarms.

Messung	
Parameter	Gewicht in kg
$m_{oa}$	0,077
$m_{ua}$	0,060
$m_Z$	0,254
$m_1 = m_{oa} + m_Z$	0,331
$m_2 = m_{ua}$	0,06

Tabelle 5.1.: Gewichtsparameter

Der Schwerpunkt des Unterarms ist durch seine geometrische Struktur klar vorgegeben und kann sofort bestimmt werden. Bei der Behandlung des Oberarms wird die Annahme getroffen, dass es sich um einen homogenen Stab mit zusätzlicher Masse in der

Antriebsachse des Unterarms handelt.

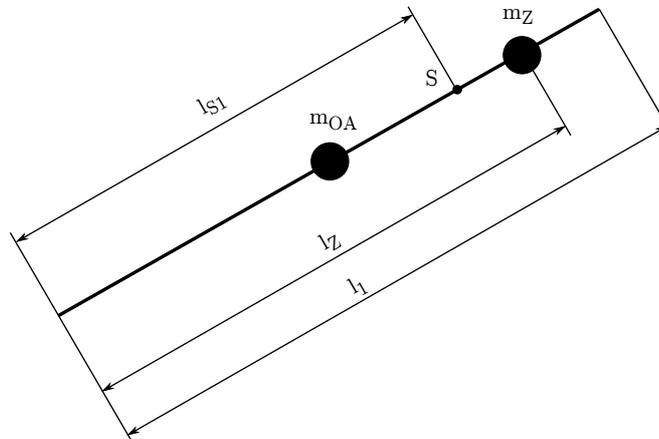


Abbildung 5.1.: Schwerpunktbestimmung des Oberarms

Aus der mechanischen Konstruktion des Roboterarms sind alle geometrischen Abmessungen bereits bekannt. Nachfolgende Tabelle listet die wichtigsten Größen (Länge, Breite, Dicke) für beide Teilkörper auf.

Messung OA		Messung UA	
Parameter	Wert in cm	Parameter	Wert in cm
$l_1$	18,0	$l_2$	18,0
$b_1$	3,0	$b_2$	3,0
$d_1$	0,8	$d_2$	0,6
$l_Z$	16,5		

Tabelle 5.2.: Geometrische Abmessungen

Der Schwerpunktsvektor  $\vec{r}_S$  ergibt sich aus dem gewichteten Mittel der Ortsvektoren aller Massenpunkte  $dm$  zu

$$\vec{r}_S = \frac{\int \vec{r} dm}{\int dm}. \quad (5.1)$$

Formel (5.1) angewandt auf die schematische Darstellung in Abbildung 5.1 ergibt:

$$\begin{aligned} l_{S1} &= \frac{m_{oa} \frac{l_1}{2} + m_Z l_Z}{m_{oa} + m_Z} \\ &= 14,76 \text{ cm} \end{aligned} \quad (5.2)$$

### 5.1.1. Trägheitsmoment des Oberarms

Für die Bestimmung des Trägheitsmoments werden Ober- und Unterarm auf die geometrische Form eines Quaders vereinfacht.

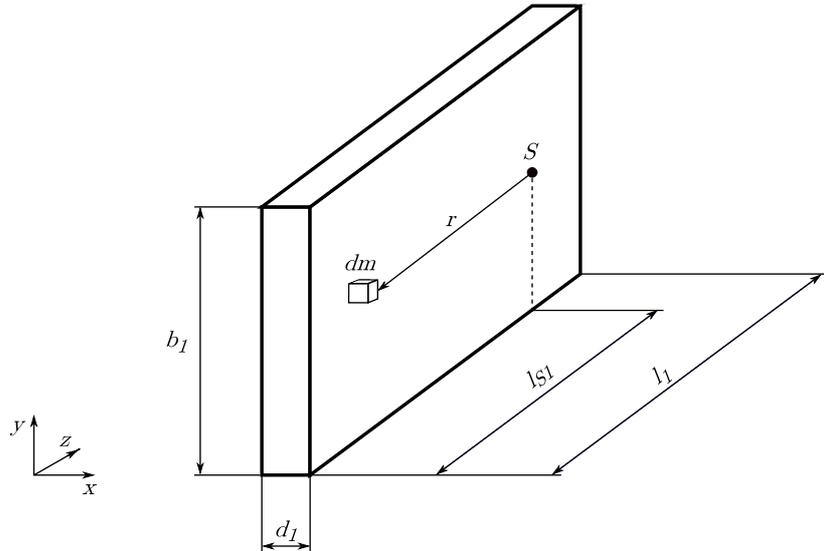


Abbildung 5.2.: Bestimmung des Trägheitsmoments des Oberarms

Mit Hilfe dieser Vereinfachung kann das Trägheitsmoment für beide Arme bestimmt werden, wobei  $M$  die Gesamtmasse des Körpers in Abbildung 5.2 bezeichnet.

$$\begin{aligned}
 J_{oa} &= \int_M r^2 dm \\
 &= \frac{m_{oa}}{b_1 d_1 l_1} \cdot \int_{-(l_1 - l_{S1})}^{l_{S1}} \int_{-b_1/2}^{b_1/2} \int_{-d_1/2}^{d_1/2} (y^2 + z^2) dx dy dz \\
 &= \frac{0,077}{0,008 \cdot 0,03 \cdot 0,18} \cdot 2,6297 \cdot 10^{-7} \approx 4,6872 \cdot 10^{-4} \text{ kgm}^2
 \end{aligned} \tag{5.3}$$

Für die Berechnung des Gesamtträgheitsmoments  $J_1$  muss die Kombination MGE des Oberarms mittels der Übersetzung  $\eta_1$  auf die Abtriebseite umgerechnet werden. Die in Formel (5.4) verwendeten Werte für die Trägheitsmomente werden den Herstellerangaben entnommen (siehe A.1).

$$\begin{aligned}
 J_1 &= J_{oa} + m_Z (l_Z - l_{S1})^2 + (J_{Motor_1} + J_{Getriebe_1} + J_{Encoder_1}) \eta_1^2 \\
 &= 4,6872 \cdot 10^{-4} + 0,254 \cdot (0,0174)^2 + 5,039 \cdot 10^{-6} \cdot 134^2 \\
 &\approx 9,1017 \cdot 10^{-2} \text{ kgm}^2
 \end{aligned} \tag{5.4}$$

### 5.1.2. Trägheitsmoment des Unterarms

Für den Unterarm kann dieselbe Vorgehensweise wie für den Oberarm angewandt werden, allerdings muss der veränderte Schwerpunkt  $l_{S2}$  berücksichtigt werden. Die Kombination MGE des Unterarms wird für die Berechnung als Zylindermantel betrachtet.

$$\begin{aligned}
 J_{ua} &= \frac{m_{ua}}{b_2 d_2 l_2} \cdot \int_{-l_{S2}}^{l_{S2}} \int_{-b_2/2}^{b_2/2} \int_{-d_2/2}^{d_2/2} (y^2 + z^2) dx dy dz \\
 &= \frac{0,06}{12} \cdot (0,03^2 + 0,18^2) \approx 1,665 \cdot 10^{-4} \text{ kgm}^2
 \end{aligned}
 \tag{5.5}$$

$$\begin{aligned}
 J_2 &= J_{ua} + (J_{Motor_2} + J_{Getriebe_2} + J_{Encoder_2}) \eta_2^2 \\
 &= 1,665 \cdot 10^{-4} + 1,231 \cdot 10^{-6} \cdot 14^2 \\
 &\approx 4,0772 \cdot 10^{-4} \text{ kgm}^2
 \end{aligned}
 \tag{5.6}$$

## 5.2. Experimentelle Parameterbestimmung

In diesem Abschnitt werden die noch übrig gebliebenen Reibungsparameter identifiziert. Die Idee zur Ermittlung beruht in dieser Arbeit auf einem experimentellen Ansatz. Abbildung 5.3 zeigt die prinzipielle Aufgabenstellung zur Bestimmung der Reibungsparameter.

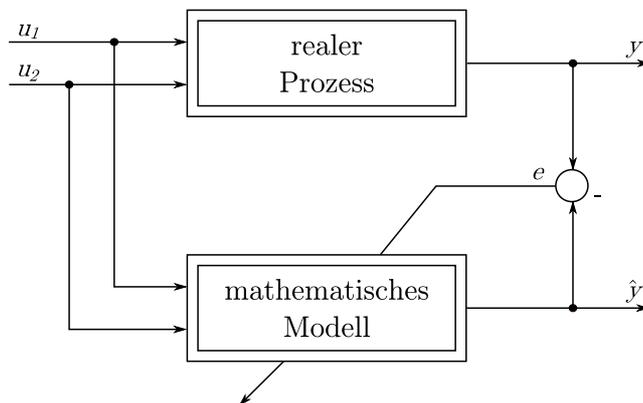


Abbildung 5.3.: Konzept zur Bestimmung der Reibungsparameter

Es werden spezielle Eingangsgrößen  $u_1$  und  $u_2$  auf den Prozess geschaltet, welche bestimmte Teildynamiken ansprechen. Selbige Eingangsgrößen werden auch in das mathematische Modell gespeist. Ziel ist die möglichst genaue Übereinstimmung von mathematischem Modell und Prozess. Die Bildung eines Fehlers  $e$  ermöglicht eine qualitative

Bewertung des Modells. Der Übersicht halber wird in Abbildung 5.3 nur eine Ausgangsgröße zur Bildung des Fehlers verwendet.

Die Eingangsgrößen werden so gewählt, dass eine möglichst entkoppelte Ermittlung der Parameter für Ober- und Unterarm durchgeführt werden kann. Hierzu werden auf einen Arm unterschiedliche Spannungssprünge aufgeschaltet, während der andere unbeeinflusst bzw. fix gehalten wird. Aufgrund der direkten Proportionalität zwischen Ankerspannung  $u$  und Winkelgeschwindigkeit  $\omega$  konnten so die Reibungskennlinien (Abbildung 4.4, Abbildung 4.5) ermittelt werden. Natürlich stellen diese Kennlinien nur eine Näherung der tatsächlichen Verhältnisse dar. Es konnte vor allem für höher werdende Winkelgeschwindigkeiten festgestellt werden, dass weitere nichtlineare Effekte zum Tragen kommen. Daher ist die Modellierung mittels linearem viskosem Term keinesfalls als exakt zu betrachten, sondern als Kompromiss zwischen Komplexität und Modellierungsgenauigkeit. Die aus den Messungen erhaltenen Parameter sind in folgenden 2 Tabellen zusammengefasst.

Messung OA	
Parameter	Wert
$\tau_{R,C_1}$	3,3 $\cdot 10^{-1}$ Nms
$a$	100
$k_{R1}$	3,6 $\cdot 10^{-1}$ Nms

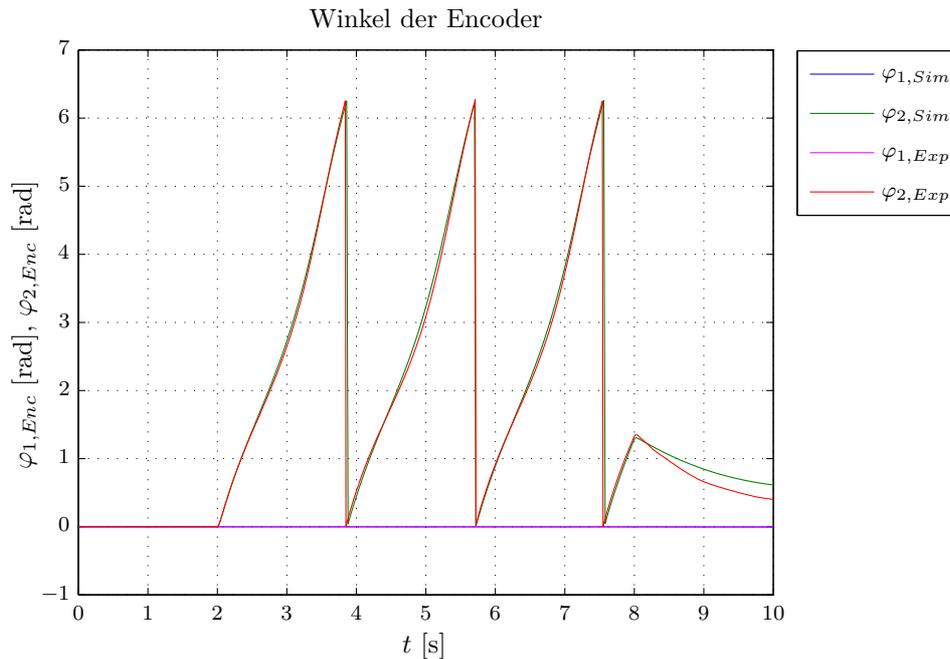
Messung UA	
Parameter	Wert
$\tau_{R,C_2}$	1,9 $\cdot 10^{-2}$ Nms
$a$	100
$k_{R2}$	1,8 $\cdot 10^{-3}$ Nms
$\tau_{R,S_2}$	2,3 $\cdot 10^{-1}$ Nm
$\alpha$	10
$\beta$	100

Tabelle 5.3.: Reibungsparameter

### 5.3. Vergleich von Simulation und Experiment

Mit Hilfe von Tabelle 5.3 kann in weiterer Folge ein Vergleich zwischen realem und mathematischem Modell gezogen werden. Es werden 3 Experimente vorgestellt, welche stellvertretend zur Verifikation der Reibungsparameter dienen.

Das erste zum Vergleich verwendete Experiment widmet sich der Dynamik des Unterarms. Der Oberarm bleibt spannungsfrei ( $u_2=0V$ ), während dem Unterarm eine Spannung von  $u_1=1V$  aufgeschaltet wird. Die Spannungssprünge sind für eine Zeitdauer von 6 Sekunden aktiv. Messtechnisch werden die Winkel  $\varphi_{1,Exp}$  und  $\varphi_{2,Exp}$  der beiden Encoder in Ober- bzw. Unterarm erfasst und in Abbildung 5.4 dargestellt.

Abbildung 5.4.: Experiment  $u_1=0V$ ,  $u_2=1V$ 

Die Beschaltung der Motoren wurde so vorgenommen, dass eine positiv anliegende Spannung eine im *mathematischen Sinn* positive Drehbewegung bewirkt. Hieraus folgt, dass sich durch das Experiment in obiger Abbildung der Unterarm 3 volle Umdrehungen gegen den Uhrzeigersinn bewegt.

Ein weiteres Experiment dient der Untersuchung der Oberarmdynamik. Um die Beeinflussung der Dynamik durch den Unterarm zu minimieren, wurde dieser elektrisch vom restlichen System abgekoppelt. Somit kann die durch die generatorische Wirkung des Motors auftretende rückinduzierte Spannung vermieden werden. Die Reibungsmodellierung durch einen zusätzlichen Term für die Mischreibung macht sich besonders zu Beginn des Experiments in Abbildung 5.5 bemerkbar. Die Abbildung zeigt, dass sich in diesem Bereich eine durch die Reibung verursachte verzögerte Bewegung des Unterarms ergibt. Die Verläufe des Oberarms bilden eine tendenziell ansteigende Abweichung zwischen Simulation und Experiment. Dieser Effekt kann jedoch vernachlässigt werden, da sich die beabsichtigten Bewegungen des Gesamtarms nicht auf mehrere Umdrehungen erstrecken.

Das letzte Experiment soll die Gesamtdynamik des Roboters erfassen. Hierzu werden beide Gelenke mittels Spannungssprüngen angesteuert und deren Winkel gemessen. In der folgenden Abbildung ist jedoch zu beachten, dass ein Versatz von 1 Sekunde zwischen den Sprungsignalen vorliegt. Dadurch ergibt sich der in Abbildung 5.6 zu Beginn dargestellte negative Winkel des Unterarms. Wiederum ist an dieser Stelle anzumerken, dass für die beabsichtigte Anwendung des Roboterarms nur 1 Umdrehung von Bedeutung ist.

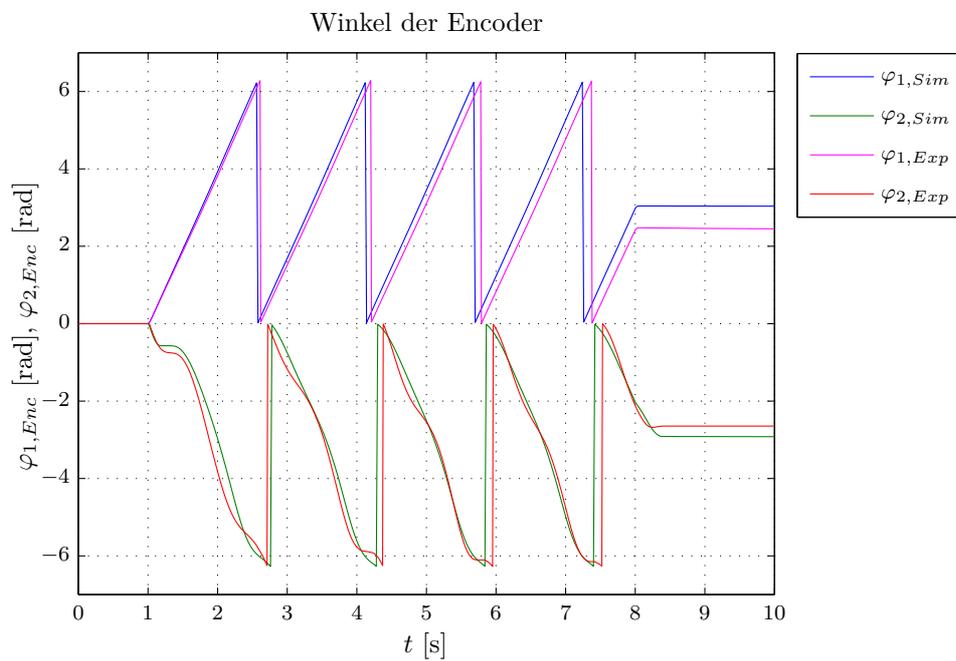


Abbildung 5.5.: Experiment  $u_1=24V$ , Unterarm frei

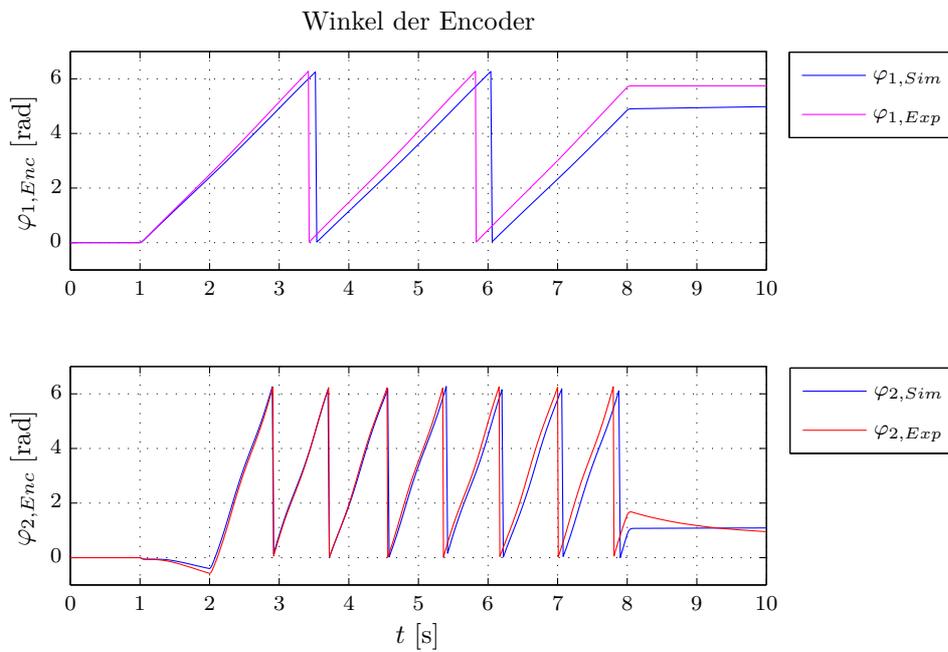


Abbildung 5.6.: Experiment  $u_1=15V$ ,  $u_2=2V$

## 6. Bewegungsplanung

Die bisher behandelten Kapitel 3, 4 und 5 fokussieren sich auf die mathematische Modellierung, sowie die Parameteridentifikation des Roboterarms. Bevor auf die regelungstechnische Problemstellung eingegangen wird, ist es sinnvoll, die Thematik der Bewegungs- bzw. Bahnplanung etwas genauer zu betrachten. Dieses Kapitel stellt Algorithmen zur Planung von Trajektorien in Gelenk- und Arbeitsraum vor. Die Zielsetzung hierbei ist, Referenzgrößen zu ermitteln, welche als Eingangsgrößen für die Regelung dienen. Typischerweise sind für die gewünschte Bewegung die vom Benutzer vordefinierten Parameter entscheidend.

Zu Beginn dieses Kapitels werden die Begriffe *Pfad* und *Trajektorie* erläutert. In den weiteren Abschnitten werden Techniken vorgestellt, die eine Trajektoriengenerierung sowohl für Punkt-zu-Punkt Bewegungen, als auch für die Planung von komplexen Bahnen ermöglichen. In Abschnitt 6.2 wird hierzu die Beschreibung von Trajektorien im Gelenkraum behandelt. Den Schluss des Kapitels bildet die Planung im Arbeitsraum in Abschnitt 6.3.

### 6.1. Pfad vs. Trajektorie

Die einfachste Bewegung, die ein Manipulator ausführen kann, ist die Punkt-zu-Punkt Bewegung. Das Übergangsverhalten zwischen Start- und Endpunkt obliegt dabei den Spezifikationen des Benutzers. Folgende Faktoren können für die Wahl eine bedeutsame Rolle spielen:

- Stellgrößenbeschränkungen des Gelenks
- Vermeidung von mechanischen Resonanzen
- „sanftes“ Übergangsverhalten, im Sinne von mechanischer Belastung
- möglichst schneller Übergang zwischen Start- und Endpunkt
- usw.

Entscheidend für das Übergangsverhalten ist die entsprechende Verwendung von Algorithmen zur Planung der Trajektorie bzw. des Pfads. Um mögliche Missverständnisse zwischen den beiden oft als Synonym verwendeten Fachbegriffen *Trajektorie* und *Pfad*

auszuräumen, wird im Folgenden eine klare Unterscheidung getroffen. Diese wird sinngemäß aus [SS03] übernommen.

Ein *Pfad* ist eine rein geometrische Beziehung im Arbeitsraum des Roboters. In diesem Zusammenhang besteht der Wunsch darin, dass der Roboter penibel dem vordefinierten Pfad folgen soll. Bei der Wahl eines Pfades müssen aufgabenspezifische Randbedingungen, wie beispielsweise Kollisionsfreiheit, berücksichtigt werden.

Eine *Trajektorie* stellt Daten bzgl. des zeitlichen Verhaltens beim Abfahren der Bahn zur Verfügung. Somit kann die Dynamik des Systems berücksichtigt werden.

Die Gesamtheit aus Pfad- und Trajektorienplanung wird in der einschlägigen Literatur als *Bahnplanung* bezeichnet. Im Allgemeinen berechnet ein Bahnplanungs-Algorithmus anhand einer Pfadvorgabe die zeitlichen Verläufe der Gelenk-Trajektorien. Dabei gilt es eine Unterscheidung bzgl. der verwendeten Koordinatensysteme zu treffen. Eine Bewegung kann mittels kartesischer Koordinaten im Arbeitsraum oder auch anhand der Gelenkwinkel im Gelenkraum definiert werden.

### 6.2. Trajektorien im Gelenkraum - Punkt-zu-Punkt Bewegung

Die Planung im Gelenkraum findet in der Praxis Anwendung, wenn ein Zielpunkt möglichst rasch angefahren werden soll. Der räumliche Verlauf der Bewegung des TCP (Tool Center Point - Werkzeugmittelpunkt) ist dabei nicht von Interesse. Die Vorgabe der Zielpunkte kann in Gelenkkoordinaten vorgenommen werden. Intuitiv wird der Benutzer jedoch kartesische Koordinaten bevorzugen, welche durch die inverse Kinematik wiederum in Gelenkwinkel umgerechnet werden können. Anhand der Sollgrößen der Gelenkwinkel errechnet der Trajektorien-Algorithmus die Verläufe für Winkel  $q^d(t)$ , Geschwindigkeit  $\dot{q}^d(t)$  und Beschleunigung  $\ddot{q}^d(t)$ . Die zeitlichen Übergänge werden in der Robotik üblicherweise durch Geschwindigkeitsprofile charakterisiert. Typische hier verwendete Profile sind parabolische- oder auch trapezförmige Verläufe. Je nach Beschaffenheit des Robotersystems müssen bereits an dieser Stelle Grenzwerte der Mechanik und Elektrik in Betracht gezogen werden. In die Planung können weitere Optimierungskriterien, wie z. B. minimale Übergangszeit, Energieeffizienz und mechanische Belastung miteinbezogen werden. Nachfolgend werden nun Algorithmen mit unterschiedlichen Lösungswegen zur Trajektorienbildung behandelt.

#### 6.2.1. Polynom

Ansätze mittels Polynomen sind sinnvoll, wenn „sanfte“ Übergänge zwischen Start- und Endpunkt geplant werden sollen. „Sanft“ bedeutet in diesem Zusammenhang, dass die Geschwindigkeit kontinuierlich zu Beginn erhöht und vor dem Ziel erniedrigt wird.

Diese Gegebenheit zeigt die mittlere Darstellung in Abbildung 6.1. Somit kann eine im mechanischen Sinn gelenkschonende Trajektorie gefahren werden. Den Grad des Polynoms legt der Benutzer durch Vorgabe entsprechender Spezifikationen fest.

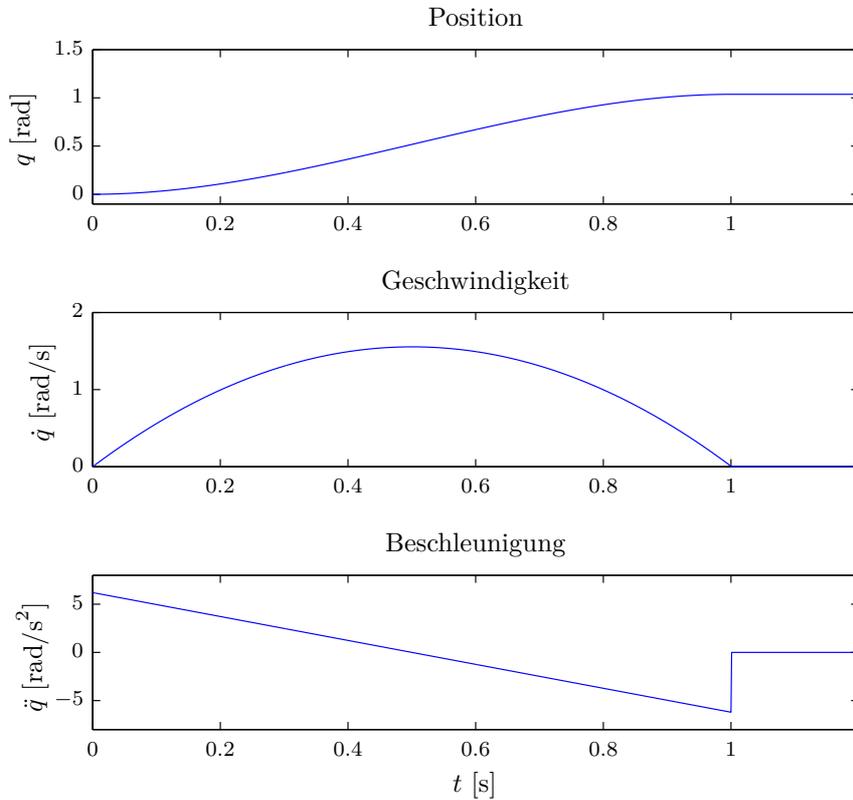


Abbildung 6.1.: Zeitfenster für Position, Geschwindigkeit und Beschleunigung eines kubischen Polynoms

In der vorliegenden Arbeit wird ein kubisches Polynom der Form

$$q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (6.1)$$

verwendet. Aufgrund dieser Wahl ergibt sich ein parabolisches Geschwindigkeitsprofil

$$\dot{q}(t) = 3a_3 t^2 + 2a_2 t + a_1 \quad (6.2)$$

und ein lineares Beschleunigungsprofil.

$$\ddot{q}(t) = 6a_3 t + 2a_2 \quad (6.3)$$

Die einzelnen Profilverläufe sind beispielhaft in Abbildung 6.1 dargestellt. In Formel (6.1) stehen 4 Koeffizienten zur Beschreibung des Polynoms zur Verfügung. Für ei-

ne eindeutige Definition der Polynomkoeffizienten müssen daher 4 Nebenbedingungen formuliert werden. Diese setzen sich aus Start- und Endposition  $q_i(t)$  und den entsprechenden Geschwindigkeiten  $\dot{q}_i(t)$  zusammen.

$$\begin{aligned} \text{Startpunkt :} \quad & q_i(t_0) = q_0 \\ & \dot{q}_i(t_0) = \dot{q}'_0 \end{aligned} \tag{6.4}$$

$$\begin{aligned} \text{Endpunkt :} \quad & q_i(t_f) = q_f \\ & \dot{q}_i(t_f) = \dot{q}'_f \end{aligned} \tag{6.5}$$

Falls der Benutzer zusätzlich Start- und Endbeschleunigung vorgeben will, müssen zwei weitere Nebenbedingungen berücksichtigt werden. Zur Erfüllung aller Forderungen müsste in diesem Fall das Polynom daher mindestens den Grad 5 besitzen.

$$q(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{6.6}$$

Mit Hilfe der NBn aus (6.4) und (6.5) kann ein Gleichungssystem aufgestellt werden, aus dem die Polynomkoeffizienten berechnet werden können.

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_f \\ \dot{q}_f \end{bmatrix} \tag{6.7}$$

In Gleichung (6.7) beschreibt  $t_0$  die Startzeit und  $t_f$  die Endzeit der geplanten Trajektorie. Damit eine Punkt-zu-Punkt Bewegung ermöglicht werden kann, muss das Gleichungssystem noch dahingehend erweitert werden, dass mehrere kubische Polynome zeitlich aneinanderghängt werden können. Hierzu wird die Startzeit  $t_0$  vorerst mit 0 festgelegt.

Somit ergeben sich die Koeffizienten zu:

$$\begin{aligned} a_0 &= q_0 \\ a_1 &= \dot{q}_f \\ a_2 &= \frac{3(q_f - q_0) - (2\dot{q}_0 + \dot{q}_f)t_f}{t_f^2} \\ a_3 &= \frac{2(q_0 - q_f) + (\dot{q}_0 + \dot{q}_f)t_f}{t_f^3} \end{aligned} \tag{6.8}$$

Um nun ein zeitliches Aneinanderhängen zu ermöglichen, ist eine Substitution der Form

$$\begin{aligned} t &\longrightarrow t - t_0 \\ t_f &\longrightarrow t_f - t_0 \end{aligned} \tag{6.9}$$

erforderlich. Es sei angemerkt, dass  $t_0$  nun wiederum die Startzeit der entsprechenden Trajektorie darstellt. Die Substitution in (6.9) kann als Verschiebung der Zeitachse nach links interpretiert werden. Durch Einsetzen von (6.8) und (6.9) kann der Verlauf der Trajektorie sowie deren zeitliche Ableitungen bestimmt werden.

$$\begin{aligned} q^d &= a_3 t^3 + [-3a_3 t_0 + a_2] t^2 + [3a_3 t_0^2 - 2a_2 t_0 + a_1] t \\ &\quad - a_3 t_0^3 + a_2 t_0^2 - a_1 t_0 + a_0 \\ \dot{q}^d &= 3a_3 t^2 + [-6a_3 t_0 + 2a_2] t + 3a_3 t_0^2 - 2a_2 t_0 + a_1 \\ \ddot{q}^d &= 6a_3 t + 2a_2 - 6a_3 t_0 \end{aligned} \tag{6.10}$$

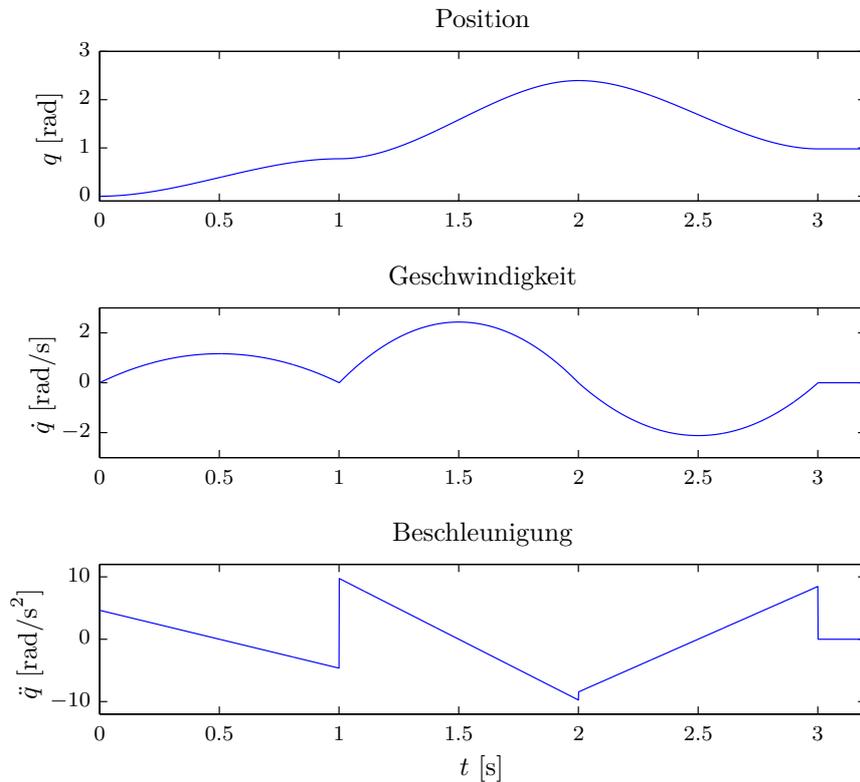


Abbildung 6.2.: Zeitfenster für Position, Geschwindigkeit und Beschleunigung aneinandergehängerter kubischer Polynome

Abbildung 6.2 zeigt die Verläufe von  $q$ ,  $\dot{q}$  und  $\ddot{q}$  eines Gelenks für eine Punkt-zu-Punkt Bewegung. Diese Kurven werden vom Trajektorien-Algorithmus geliefert und stehen der Regelung als Sollreferenzgrößen zur Verfügung.

### 6.2.2. Trapez

Eine alternative Variante stellt die Implementierung mittels LSPB<sup>1</sup> dar, welche eine Kombination aus linearen und parabolischen Geschwindigkeitssegmenten darstellt. Mit Hilfe dieses Profils kann für ein bestimmtes Zeitintervall eine konstante Geschwindigkeit des Manipulators gefahren werden. Dabei besitzt die Trajektorie zu Beginn eine rampenförmige Steigung und zum Schluss eine rampenförmige Verzögerung der Geschwindigkeit. Die NBn bestehen wie im Fall des Polynomansatzes aus Start- und Endposition  $q_i(t)$  sowie den Geschwindigkeiten  $\dot{q}_i(t)$ . Zusätzlich wird die gewünschte konstante Geschwindigkeit  $V$  vorgegeben, die vom Manipulator im linearen Segment gefahren werden soll. Die genaue mathematische Behandlung wird an dieser Stelle nicht durchgeführt. Der interessierte Leser sei auf [SV89] verwiesen. Das trapezförmige Profil kann folgendermaßen beschrieben werden:

$$q^d = \begin{cases} \frac{V}{2t_b}t^2 + q_0 & 0 \leq t \leq t_b \\ Vt + \frac{q_0 + q_f - Vt_f}{2} & t_b \leq t \leq t_f - t_b \\ -\frac{V}{2t_b}t^2 + \frac{Vt_f}{t_b}t - \frac{Vt_f^2}{2t_b} + q_f & t_f - t_b \leq t \leq t_f \end{cases} \quad (6.11)$$

Obiger Ausdruck verdeutlicht die Unterteilung in ein parabolisches-lineares-parabolisches Segment. Die Blendzeit  $t_b$ , welche die Dauer der Geschwindigkeitsrampe festlegt, kann unter Anwendung von

$$t_b = \frac{q_0 - q_f + Vt_f}{V} \quad (6.12)$$

ermittelt werden. Es ist zu beachten, dass die Wahl der Geschwindigkeit  $V$  Beschränkungen unterliegt.

$$\frac{q_f - q_0}{t_f} < V \leq \frac{2(q_f - q_0)}{t_f} \quad (6.13)$$

Werden diese nicht eingehalten, so kann die Trajektorie möglicherweise nicht für die gegebenen Anforderungen realisiert werden. Eine weitere interessante Variante offenbart sich, wenn die Endzeit  $t_f$  nicht genau spezifiziert wird. Die Idee dieser Variante ist

---

<sup>1</sup>Linear Segments with Parabolic Blends

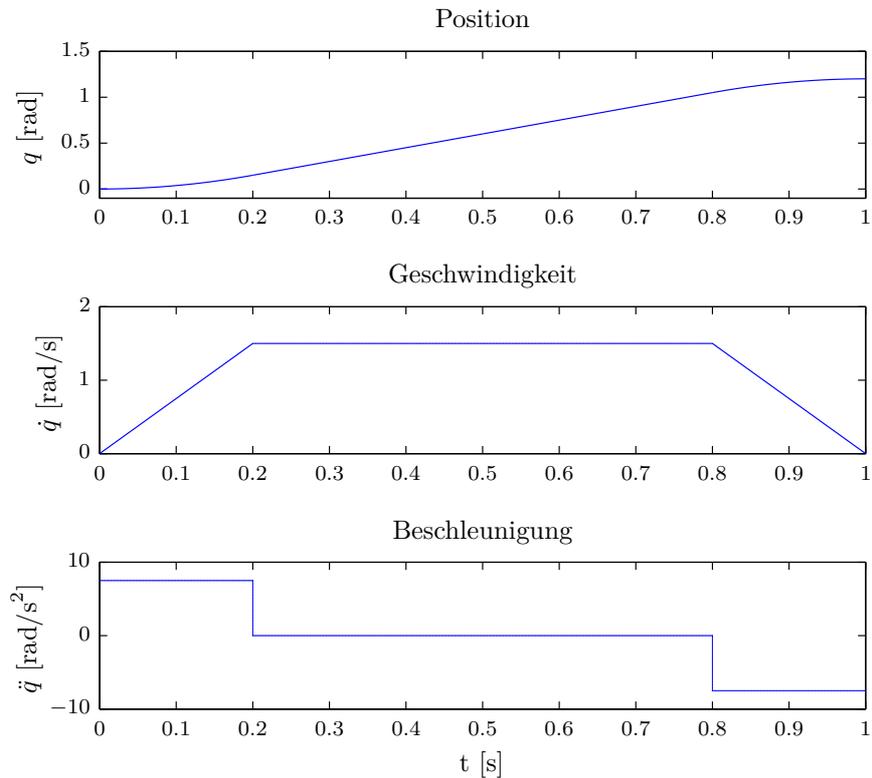


Abbildung 6.3.: Zeitfenster für Position, Geschwindigkeit und Beschleunigung einer LSPB Trajektorie

eine Berechnungsvorschrift zu finden, welche eine minimale Übergangszeitspanne zwischen den Punkten  $q_0$  und  $q_f$  garantiert. Die Trajektorien, die eine solche Anforderung erfüllen, werden auch *Bang-Bang Trajektorien* genannt, da die Minimalanforderung in direktem Zusammenhang mit einer maximalen Stellgröße besteht. Da diese Option allerdings nicht Teil dieser Arbeit ist, sei auf die einschlägige Literatur [SS03] und [SV89] verwiesen.

### 6.3. Trajektorien im Arbeitsraum - Pfad Bewegung

Im Gegensatz zur Planung im Gelenkraum ist im Arbeitsraum der geometrische Verlauf des TCP von besonderer Relevanz. Der Manipulator soll dabei exakt dem vordefinierten Pfad des Anwenders folgen. Die Definition der Bewegung erfolgt mit Hilfe einer rein analytischen Formulierung in Parameterform. Die Solltrajektorien werden ähnlich vorgegeben, wie im Fall der Planung im Gelenkraum mittels Position, Geschwindigkeit und Beschleunigung eines jeden Gelenks. Ziel in dieser Arbeit ist die Planung von geometrischen Bewegungen in Form von Kreisen, Ellipsen und Geraden.

### 6.3.1. Bahnplanung in Parameterform

Durch Verwendung der Parameterdarstellung können die geometrischen Figuren anhand eines einzigen Parameters  $t$  mathematisch dargestellt werden. Der implementierte Trajektorienalgorithmus stellt 3 mögliche Bewegungen zur Auswahl. Dem Benutzer obliegt die Wahl der Position, Größe und Geschwindigkeit, mit der die Figur abgefahren werden soll. Aufgrund der eingesetzten Inkrementalgeber startet jede Bewegung im Ursprung mit den Winkeln  $q_1 = 0$  und  $q_2 = 0$ . Der Algorithmus ermittelt automatisch eine geradlinige Trajektorie, die den Ursprung mit dem Startpunkt verbindet. Im Folgenden werden die 3 implementierten geometrischen Figuren inklusive mathematischer Definition in Parameterform gezeigt.

#### Kreis:

Die Beschreibung der Figur erfolgt durch den Kreisradius  $r$ , den Mittelpunkt  $(x_m, y_m)$  sowie der Winkelgeschwindigkeit  $a$ . In Abbildung 6.4 ist weiters der Startpunkt der Bewegung erkennbar.

$$\begin{aligned}
 x &= x_m + r \cos(at) & y &= y_m + r \sin(at) \\
 \dot{x} &= -ra \sin(at) & \dot{y} &= ra \cos(at) \\
 \ddot{x} &= -ra^2 \cos(at) & \ddot{y} &= -ra^2 \sin(at)
 \end{aligned} \tag{6.14}$$

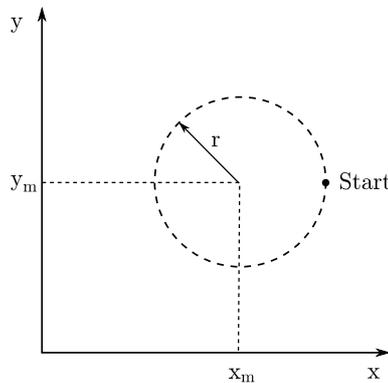


Abbildung 6.4.: Kreis in Parameterform

#### Ellipse:

$$\begin{aligned}
 x &= x_m + a \cos(ct) & y &= y_m + b \sin(ct) \\
 \dot{x} &= -ac \sin(ct) & \dot{y} &= bc \cos(ct) \\
 \ddot{x} &= -ac^2 \cos(ct) & \ddot{y} &= -bc^2 \sin(ct)
 \end{aligned} \tag{6.15}$$

Identisch zur Formulierung des Kreises bezeichnet  $(x_m, y_m)$  den Mittelpunkt. Die Parameter  $a$ ,  $b$  und  $c$  legen die Hauptachsen der Ellipse sowie die Winkelgeschwindigkeit fest.

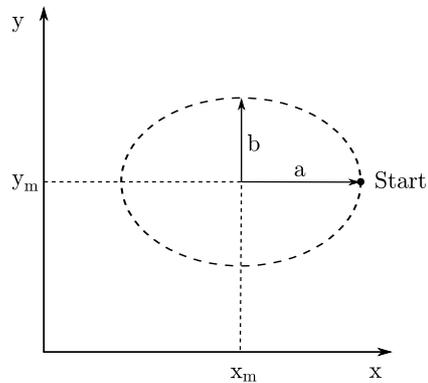


Abbildung 6.5.: Ellipse in Parameterform

#### Gerade:

Die geradlinige Bewegung wird mit Hilfe eines Startpunktes  $\mathbf{p}_0$  und eines Endpunktes  $\mathbf{p}_1$  beschrieben. Aus Gründen der besseren Lesbarkeit wird die Vektorform verwendet. Die Übergangszeit  $\Delta t$  spezifiziert die Geschwindigkeit mit der die Punkte abgefahren werden.

$$\begin{aligned}\mathbf{x} &= \mathbf{p}_0 + \frac{t}{\Delta t} \cdot (\mathbf{p}_1 - \mathbf{p}_0) \\ \dot{\mathbf{x}} &= \frac{1}{\Delta t} \cdot (\mathbf{p}_1 - \mathbf{p}_0) \\ \ddot{\mathbf{x}} &= \mathbf{0}\end{aligned}\tag{6.16}$$

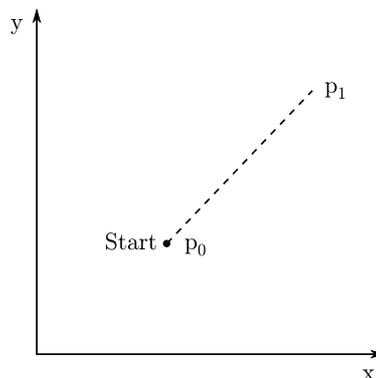


Abbildung 6.6.: Gerade in Parameterform

### 6.3.2. Probleme im Arbeitsraum

Die Festlegung einer Trajektorie im Arbeitsraum birgt einige Schwierigkeiten, die es bei der Planung zu beachten gilt. Im Folgenden werden 3 typische, in der Praxis auftretende Probleme erläutert.

**Problem Typ 1:** Zwischenpunkte eines Pfades nicht erreichbar

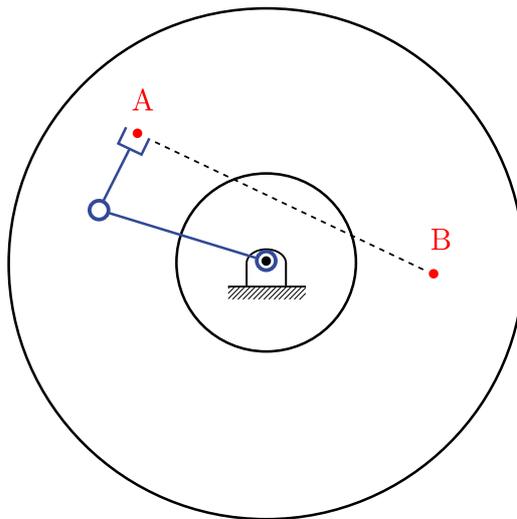


Abbildung 6.7.: Problem Typ 1

Abbildung 6.7 zeigt einen 2-achsigen Roboterarm und den resultierenden Arbeitsraum anhand zweier Kreise. Zwischen den beiden Positionen A und B des Endeffektors ist eine geradlinige Bewegung geplant. Obgleich sich beide Punkte im Arbeitsbereich befinden, kann es durchaus vorkommen, dass Zwischenpunkte der Bewegung physikalisch nicht angefahren werden können. Diesen Umstand veranschaulicht Abbildung 6.7. Das Problem stellen dabei die Zwischenpunkte dar, welche sich außerhalb des gültigen Arbeitsbereichs befinden.

Dieses Fallbeispiel zeigt eine typische Situation, bei der die Bewegungsplanung im Arbeitsraum fehlschlägt. Bemerkenswert ist jedoch, dass für selbige Planung im Gelenkraum keinerlei Probleme auftreten würden. Allerdings könnte in Folge dessen der TCP nicht auf einer Geraden geführt werden. Die Handhabung dieses Problems kann in der Praxis auf verschiedenen Lösungswegen erfolgen. Da die Planung meist offline durchgeführt wird, könnte der Benutzer z. B. bereits an dieser Stelle auf ein Problem hingewiesen werden. Ein weiterer Ansatz wäre den Roboter zu stoppen, wenn eine Bewegung sich außerhalb des definierten Arbeitsraums befindet. Aufgrund der besonderen mechanischen Konstruktion des hier vorgestellten Roboters kann dieser Problemtyp jedoch

nahezu ausgeschlossen werden, da jeder Zwischenpunkt im Arbeitsraum angefahren werden kann.

**Problem Typ 2:** Gelenkgeschwindigkeiten nahe Singularitäten

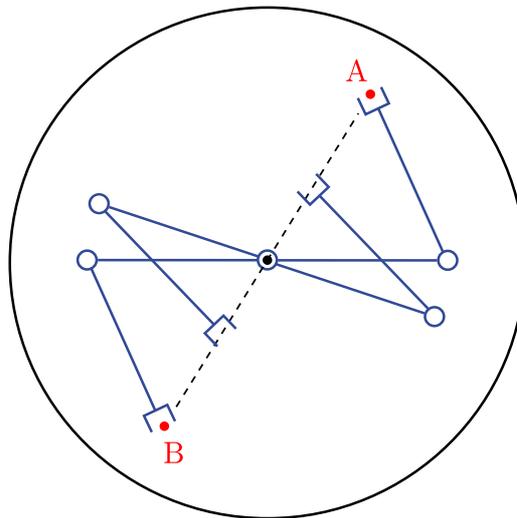


Abbildung 6.8.: Problem Typ 2

In gewissen Bereichen ist es nicht möglich, vorgegebene Geschwindigkeiten im Arbeitsraum in entsprechende Werte für die Gelenke umzurechnen. Die nicht eindeutig lösbar inverse Kinematik führt zu überhöhten Gelenkgeschwindigkeiten. Aufgrund dessen kommt es in weiterer Folge zu großen Abweichungen vom Solltrajektorienverlauf. Wie bereits in Abschnitt 3.5 erläutert, werden Positionen mit diesen Eigenschaften als singular bezeichnet.

In Abbildung 6.8 nähert sich ein 2-achsiger Roboter mit gleich langen Armen der Singularität im Ursprung des eingezeichneten Kreises. Zur besseren Veranschaulichung sind mehrere Zwischenpositionen eingezeichnet. Eine mögliche Strategie zur Behandlung dieses Problemtyps ist die Regularisierung der Jacobi-Matrix nahe den Singularitäten. Das Ziel hierbei ist, die Matrix zu dämpfen und dadurch überhöhte Gelenkgeschwindigkeiten zu vermeiden. In Abschnitt 7.3.2 findet diese Vorgehensweise Anwendung.

**Problem Typ 3:** Punkte nur mit unterschiedlichen Lösungen erreichbar

Abbildung 6.9 stellt einen weiteren Problemtyp dar. Der gezeigte 2-achsige Roboterarm besitzt identische Armlängen. Zielvorstellung ist eine geradlinige Bewegung zwischen den Punkten A und B. Diese geplante Bahn kann nur unter der Voraussetzung angefahren werden, dass die Gelenkwinkel keinerlei Beschränkung unterliegen. Oftmals

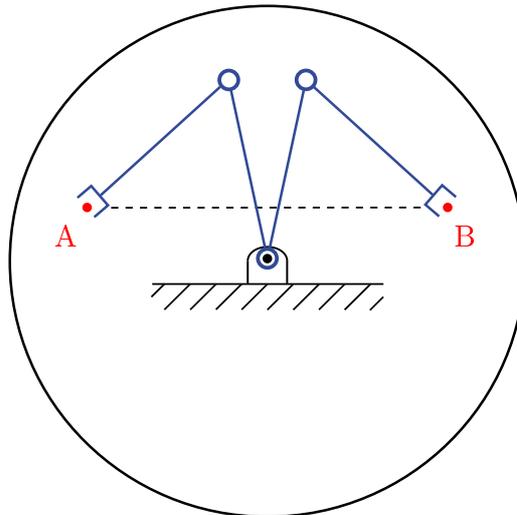


Abbildung 6.9.: Problem Typ 3

ist die mechanische Konstruktion Grund dafür, dass es zu Problemen im Bereich von Kreuzungspunkten zwischen den einzelnen Achsen kommt. Beispielsweise kann die Verkabelung der äußeren Roboterachsen zu einer Beschränkung der Gelenkwinkel führen. Liegt ein solcher Fall vor, so kann der Pfad wie in Abbildung 6.9 dargestellt, nicht mit einer einzigen physikalischen Lösung bewältigt werden. Durch die Verwendung eines Schleifrings und der damit verbundenen Entkopplung der beiden Arme kann diese Art von Problem für den verwendeten Roboter ausgeschlossen werden.

Zusammengefasst kann also gesagt werden, dass es bei der Planung im Arbeitsraum deutlich mehr Aspekte zu beachten gilt, als dies bei der Planung im Gelenkraum der Fall ist.

## 7. Regelungskonzepte

Dieses Kapitel widmet sich der Aufgabe, einen Roboter zu regeln. Den Schwerpunkt bildet die Ermittlung von Stellgrößen für die Gelenkantriebe. Eine geeignete Wahl resultiert in einer bestmöglichen *Tracking-Performance*, da der Systemausgang der Soll-Referenzgröße am Eingang optimal folgen kann. Dabei gilt es, Stellgrößenbeschränkungen zu beachten und diese nicht zu überschreiten. Für die Umsetzung der regelungstechnischen Aufgabe stehen zahlreiche Methoden zur Verfügung. Die Herausforderung liegt in der Auswahl von Konzepten, welche für den entsprechenden Anwendungsfall die besten Ergebnisse liefern. Etliche Aspekte, wie z. B.

- Mechanische Konstruktion des Roboters
- Antriebsstrang der Gelenkachsen
- Anwendungsfall - PTP / Pfad-Bewegung,

können hierbei die Entscheidung des Ingenieurs beeinflussen. Vielfach schränkt bereits das mechanische Design die Wahl von Regelungskonzepten ein. Dies ergibt sich aufgrund der Tatsache, dass sich für verschiedene Typen von Manipulatoren komplett unterschiedliche Anwendungsprobleme ergeben.

Der verwendete Antriebsstrang bildet ebenfalls eine entscheidende Komponente. Dies kann an einem Vergleich zweier Manipulatoren gezeigt werden. Der Erste ist mit permanenterregten DC-Motoren inklusive Getriebeuntersetzung ausgestattet, während der Zweite einen Direktantrieb besitzt. Im ersten Fall führt die Untersetzung zu einem linearisierenden Verhalten der Antriebsdynamik. Damit kann eine Entkopplung der Gelenke hinsichtlich einer Reduktion von nichtlinearen Effekten erreicht werden. Allerdings bewirkt die Untersetzung auch neue nichtlineare Eigenschaften, wie Reibung und Getriebeispiel. Im zweiten Fall treten die nichtlinearen Kopplungseffekte in den Vordergrund. Die Dynamik der Antriebe ist deutlich komplexer, wobei entgegen dem ersten Fall nur geringe Reibungseffekte auftreten.

Beide Antriebslösungen beeinträchtigen unterschiedliche Teilbereiche der Systemdynamik. Daher ist es in Hinsicht auf die Performance zweckmäßig, verschiedene Regelungskonzepte zu implementieren.

Auch der Anwendungsfall spielt eine elementare Rolle, welche Regelungsstrategie bessere Ergebnisse liefert. Generell wird, wie bereits in Abschnitt 6.2 und 6.3 erwähnt, eine

Unterscheidung zwischen Verfahren im Gelenkraum und Arbeitsraum getroffen. Für beide Fälle werden in den folgenden Abschnitten nichtlineare Konzepte vorgestellt und anhand von Versuchen am realen Modell evaluiert. Den Schluss dieses Kapitels bildet eine Erweiterung des Konzepts aus Abschnitt 7.2. Dabei wird die Regelkreisstruktur so verändert, dass ein robustes Verhalten gegenüber ungenau modellierten Dynamiken erzielt wird.

## 7.1. Allgemeine Betrachtung

Die einfachste und naheliegendste Idee zur Regelung eines mehrachsigen Roboters besteht in der Betrachtung jedes einzelnen Gelenks als unabhängiges Teilsystem. Jede Axe wird in Folge dessen als SISO-System<sup>1</sup> aufgefasst. Die gegenseitige Beeinflussung der Gelenke durch nichtlineare Kopplungen wird als Störgrößeneingang  $v$  des Eingrößensystems interpretiert. Abbildung 7.1 zeigt eine schematische Darstellung dieses Sachverhalts.

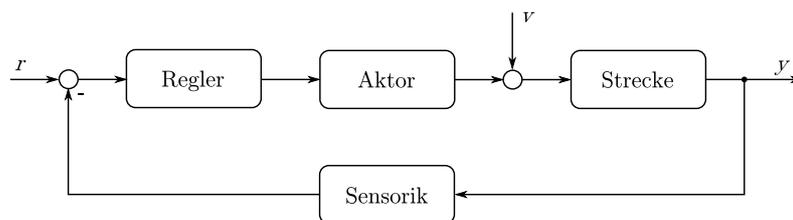


Abbildung 7.1.: Struktur zur Regelung eines SISO-System

Typischerweise werden in der Robotik PD- und PID-Regler eingesetzt. Findet die Betrachtung des Modells im Zustandsraum, statt so sind auch Rückkopplungen mit Zustandsreglern und LQ-Reglern<sup>2</sup> ( $LQR$ ) denkbar. Die Verwendung hängt natürlich vom Anwendungsfall des Manipulators ab.

Das große Manko dieses dezentralen Ansatzes besteht darin, dass die eigentliche Struktur des Systems nicht beachtet wird. Das verkoppelte Mehrgrößensystem wird nicht als solches behandelt, sondern geschickt in ein aus mehreren SISO-Systemen bestehendes Gesamtsystem aufgeteilt. Die Vereinfachung, jedes Gelenk als unabhängig anzunehmen, kann für eine exakte Analyse nicht hingenommen werden. Insbesondere bei *High-Speed Anwendungen* und bei der Verwendung von kleinen Getriebeuntersetzungen spielen die nichtlinearen Kopplungen eine bedeutsame Rolle. Die Betrachtung als reine Störgröße würde in obigem Ansatz zu einer massiven Verschlechterung der *Tracking-Performance* führen.

<sup>1</sup>Ein SISO-System besitzt nur einen Ein- und Ausgang.

<sup>2</sup>LQ-Regler sind optimale Regler hinsichtlich der Erfüllung eines quadratischen Gütekriteriums.

Um die regelungstechnische Aufgabe korrekt anzugehen, müssen nichtlineare Konzepte für den Mehrgrößenfall angewendet werden. Aus diesem Grund werden in der vorliegenden Arbeit Strategien für MIMO-Systeme<sup>3</sup>, wie auch in [SV89] und [SS03] beschrieben, vorgestellt.

## 7.2. Inverse Dynamic Control im Gelenkraum

Die Idee dieses nichtlinearen Konzepts liegt in einer verschachtelten Regelkreisstruktur, welche sich aus 2 Schleifen zusammensetzt. Die innere bewirkt eine globale Linearisierung mittels einer nichtlinearen Rückkopplung, während die äußere überlagerte Schleife ein lineares Regelgesetz implementiert. Der erzielte Effekt der inneren Rückkopplung entspricht vom Prinzip her der Regelung mittels exakter Linearisierung. Aufgrund der speziellen Struktur der Bewegungsgleichungen

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \boldsymbol{\tau}_R \quad (7.1)$$

muss das nichtlineare Rückführungsgesetz allerdings nicht über den komplizierten und mathematisch aufwendigen Formelapparat der exakten Linearisierung ermittelt werden.

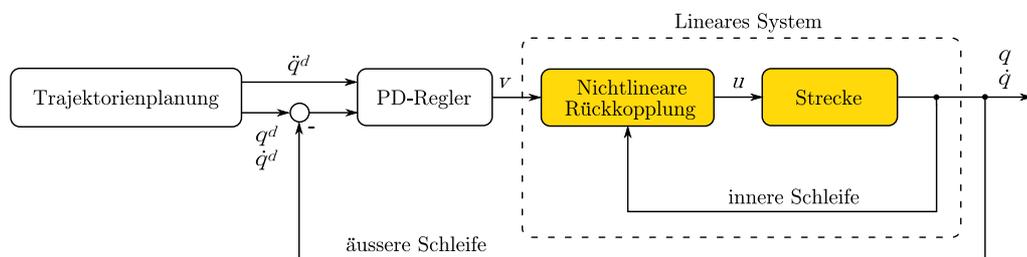


Abbildung 7.2.: IDC-Regelkreisstruktur im Gelenkraum

Die grafische Veranschaulichung zeigt die verschachtelte Struktur dieses Regelungskonzepts. Die Referenzgrößen  $\mathbf{q}^d$ ,  $\dot{\mathbf{q}}^d$  und  $\ddot{\mathbf{q}}^d$  bilden die Eingangsgrößen der äußeren Schleife. Am Systemausgang werden die Gelenkwinkel  $\mathbf{q}$  und die Winkelgeschwindigkeiten  $\dot{\mathbf{q}}$  rückgeführt. Die Steuerung des Roboters findet somit komplett im Gelenkraum statt. Dies schließt die Verwendung von Parametern im Basiskoordinatensystem jedoch keinesfalls aus, da diese ohne Umwege in den Gelenkraum umgerechnet werden können.

<sup>3</sup>Ein MIMO-System ist das Pendant zum SISO-System und besitzt mehrere Ein- und Ausgänge.

### 7.2.1. Globale Linearisierung

Grundgedanke der globalen Linearisierung ist es, ein nichtlineares Rückführungsgesetz der Form

$$\mathbf{u} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \quad (7.2)$$

zu bestimmen, welches eingesetzt in die Bewegungsgleichungen des Roboters die nichtlinearen Koppelterme kompensiert. Generell ist es sehr schwierig, ein solches Gesetz zu finden. Durch die besondere Struktur der Gleichungen

$$\bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} \quad (7.3)$$

ist jedoch sofort einsichtig, dass die Forderungen zur Entkopplung durch folgende Wahl von  $\mathbf{u}$  erfüllt sind.

$$\mathbf{u} = \bar{\mathbf{M}}(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \quad (7.4)$$

Die zusätzlich eingeführte Größe  $\mathbf{v}$  repräsentiert hierbei den neuen Systemeingang. Somit kann das linear entkoppelte Gesamtsystem

$$\ddot{\mathbf{q}} = \mathbf{v} \quad (7.5)$$

mit dem neuen Eingangsvektor  $\mathbf{v}$  angeschrieben werden. Das Ergebnis der Rückführung führt zu einem Doppelintegrierer für jedes einzelne Robotergelenk (Abbildung 7.3).

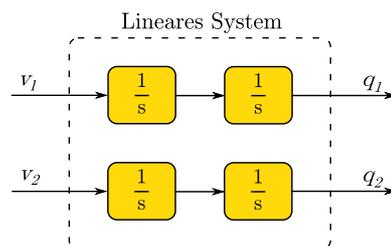


Abbildung 7.3.: Struktur des entkoppelten Modells

### 7.2.2. PD-Regler

Das neue System setzt sich aus 2 entkoppelten SISO-Systemen mit den Eingängen  $v_1$  und  $v_2$  zusammen. Auf jedes dieser Teilsysteme können lineare Regelungsmethoden angewendet werden.

Ferner gilt es ein Regelgesetz zu finden, welches das System aus 2 Doppelintegriern stabilisiert. Aufgrund des primären Anwendungsbereichs des Roboters, Zielpunkte möglichst rasch abzufahren, wird an dieser Stelle ein PD-Regler eingesetzt. Durch die integrierende Strecke ist zudem die stationäre Genauigkeit gesichert. Der Einsatz eines Regelgesetzes der Form

$$\mathbf{v} = -\mathbf{K}_D \dot{\mathbf{q}} - \mathbf{K}_P \mathbf{q} + \mathbf{r} \quad (7.6)$$

resultiert in einem Gesamtsystem mit  $\mathbf{r}$  als Referenzgröße.

$$\ddot{\mathbf{q}} + \mathbf{K}_D \dot{\mathbf{q}} + \mathbf{K}_P \mathbf{q} = \mathbf{r} \quad (7.7)$$

Mit Hilfe des zeitlich geplanten Verlaufs der Trajektorie im Gelenkraum

$$t \rightarrow (\mathbf{q}^d(t), \dot{\mathbf{q}}^d(t)) \quad (7.8)$$

kann der Eingang  $\mathbf{r}$  umformuliert werden.

$$\mathbf{r} = \ddot{\mathbf{q}}^d + \mathbf{K}_D \dot{\mathbf{q}}^d + \mathbf{K}_P \mathbf{q}^d \quad (7.9)$$

Formel (7.9) eingesetzt in (7.6) ergibt das tatsächlich implementierte Regelgesetz.

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D (\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_P (\mathbf{q}^d - \mathbf{q}) \quad (7.10)$$

Bei den verwendeten Matrizen  $\mathbf{K}_P$  und  $\mathbf{K}_D$  handelt es sich um positiv definite  $(n \times n)$ -Diagonalmatrizen. Die Definitheit spielt dabei im Zusammenhang mit dem Stabilitätsbeweis in Abschnitt 7.2.3 eine bedeutsame Rolle. Vorerst seien die Matrizen ohne weitere Erläuterung durch folgende Definition gegeben.

$$\mathbf{K}_P = \begin{bmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{bmatrix} \quad \mathbf{K}_D = \begin{bmatrix} 2\zeta\omega_1 & 0 \\ 0 & 2\zeta\omega_2 \end{bmatrix} \quad (7.11)$$

Das Übertragungsverhalten des geschlossenen Regelkreises entspricht einem System 2. Ordnung. Orientiert man sich nun an der Übertragungsfunktion eines Systems mit dominantem Polpaar

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{\left(\frac{s}{\omega_n}\right)^2 + 2\zeta\left(\frac{s}{\omega_n}\right) + 1} \quad (7.12)$$

und der daraus resultierenden Sprungantwort<sup>4</sup>, so können auch im Fall des Manipulators Parameter für Dämpfung  $\zeta$  und Kennkreisfrequenz  $\omega_n$  eingeführt werden. Mit Hilfe von  $\omega_n$  kann in diesem Zusammenhang die Schnelligkeit des Systems beeinflusst werden. In der Robotik ist es nun üblich  $\zeta = 1$  zu wählen. Diese Annahme resultiert in einer Sprungantwort mit minimalen Überschwüngen, was in weiterer Folge mathematisch bewiesen wird.

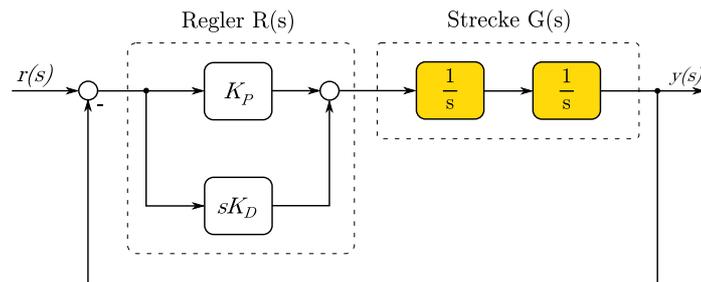


Abbildung 7.4.: Struktur des linearen Modells

Gegeben sei das linearisierte Modell des Roboters im Laplace-Bereich, wie in Abbildung 7.4 dargestellt.

$$\text{Strecke: } G(s) = \frac{1}{s^2}$$

$$\text{Regler: } R(s) = K_P + K_D s$$

(7.13)

$$\text{offener RK: } L(s) = R(s)G(s) = \frac{K_P + sK_D}{s^2}$$

$$\text{geschl. RK: } T(s) = \frac{L(s)}{1 + L(s)} = \frac{sK_D + K_P}{s^2 + sK_D + K_P}$$

Aus dem Vergleich der Übertragungsfunktion des geschlossenen Regelkreises mit Formel (7.12) kann gefolgert werden, dass  $K_P$  und  $K_D$  mittels der Ausdrücke

$$K_D = 2\zeta\omega_n, \quad K_P = \omega_n^2 \quad (7.14)$$

beschrieben werden können.

Die Analyse der Sprungantwort lässt Rückschlüsse auf das Einschwingverhalten des Regelkreises zu. Durch Aufschaltung eines Eingangssprungs  $r(s) = \frac{1}{s}$  kann die Antwort des Systems im Laplace-Bereich berechnet werden.

<sup>4</sup>Vgl. [HD04, S. 201-202]

$$\begin{aligned}
 y(s) = r(s)T(s) &= \frac{1}{s} \cdot \frac{2\zeta\left(\frac{s}{\omega_n}\right) + 1}{\left(\frac{s}{\omega_n}\right)^2 + 2\zeta\left(\frac{s}{\omega_n}\right) + 1} \\
 &= \frac{2\zeta\left(\frac{1}{\omega_n}\right)}{\left(\frac{s}{\omega_n}\right)^2 + 2\zeta\left(\frac{s}{\omega_n}\right) + 1} + \frac{1}{s \left[ \left(\frac{s}{\omega_n}\right)^2 + 2\zeta\left(\frac{s}{\omega_n}\right) + 1 \right]}
 \end{aligned}
 \tag{7.15}$$

Obige Gleichung kann anhand von Tabellen [Int] in den Zeitbereich rücktransformiert werden. Dabei muss eine Unterscheidung bzgl. des Dämpfungsfaktors  $\zeta$  durchgeführt werden.

$$\zeta < 1 : \quad y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cdot \cos\left(\sqrt{1-\zeta^2}\omega_n t + \psi\right)
 \tag{7.16}$$

$$\text{mit } \psi = \arcsin(\zeta)$$

$$\zeta = 1 : \quad y(t) = 1 - e^{-\omega_n t} (1 - \omega_n t)
 \tag{7.17}$$

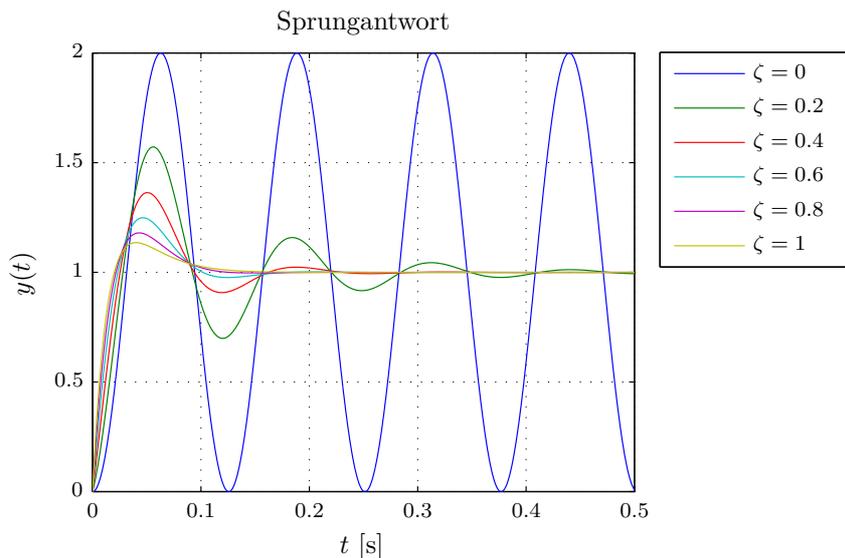


Abbildung 7.5.: Sprungantwort des Regelkreises,  $\omega_n = 50$

Durch eine einfache Extremwertanalyse kann die Überschwingweite  $M_P$ <sup>5</sup> ermittelt werden.

$$\begin{aligned}
 M_P &= 1 - e^{-\omega_n t_{M_P}} (1 - \omega_n t_{M_P}) \\
 \text{mit } t_{M_P} &= \frac{2}{\omega_n}
 \end{aligned}
 \tag{7.18}$$

Aus Abbildung 7.5 ist eindeutig ersichtlich, dass die Sprungantwort für  $\zeta \rightarrow 1$  immer weiter gedämpft wird. Für die in Formel (7.18) berechnete Überschwingweite ergibt sich ein Wert von 1.135, was einem prozentuellen Überschwingen von 13.5% entspricht.

<sup>5</sup> $M_P$  gibt den maximalen Wert der Ausgangsgröße an.

Natürlich wären auch stärkere Dämpfungen möglich, wobei sich diese negativ auf die Anstiegszeit auswirken würden.

### 7.2.3. Beweis der Stabilität

Rufen wir uns nochmals die dynamischen Gleichungen in der Form

$$\mathbf{u} = \bar{\mathbf{M}}(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \quad (7.19)$$

in Erinnerung. Dabei beschreibt  $\bar{\mathbf{M}}(\mathbf{q})$  die symmetrische und positiv definite Massenmatrix. Soll eine Trajektorie  $\mathbf{q}^d(t)$ ,  $\dot{\mathbf{q}}^d(t)$  und  $\ddot{\mathbf{q}}^d(t)$  vorgegeben werden, so kann der Regelfehler wie folgt definiert werden.

$$\mathbf{e} = \begin{bmatrix} \mathbf{q}^d - \mathbf{q} \\ \dot{\mathbf{q}}^d - \dot{\mathbf{q}} \end{bmatrix} \quad (7.20)$$

Um Aussagen über die globale asymptotische Stabilität treffen zu können, muss die Fehlerdynamik des Systems betrachtet werden. Für das Regelgesetz werden zwei neue Matrizen  $\hat{\mathbf{M}}$  und  $\hat{\mathbf{h}}$  eingeführt, um den Umstand zu berücksichtigen, dass es sich bei beiden Kenngrößen in der Praxis um Schätzungen handelt. Aufgrund dieser Annahme kann die Berechnung für den allgemeinen Fall betrachtet werden.

$$\text{System: } \bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} \quad (7.21)$$

$$\text{Regelgesetz: } \mathbf{u} = \hat{\mathbf{M}}\mathbf{v} + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$$

$$\begin{aligned} \text{Gesamtsystem: } \ddot{\mathbf{q}} &= \bar{\mathbf{M}}^{-1}\hat{\mathbf{M}}\mathbf{v} + \underbrace{\bar{\mathbf{M}}^{-1}(\hat{\mathbf{h}} - \mathbf{h})}_{\Delta\mathbf{h}} \\ &= \mathbf{v} + \underbrace{\left[ \bar{\mathbf{M}}^{-1}\hat{\mathbf{M}} - \mathbf{I} \right] \mathbf{v} + \bar{\mathbf{M}}^{-1}\Delta\mathbf{h}}_{\mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})} \\ &= \mathbf{v} + \mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) \end{aligned} \quad (7.22)$$

Für den Eingang  $\mathbf{v}$  des Gesamtsystems wird ein neuer Ausdruck der Form

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{w}(\mathbf{e}) \quad (7.23)$$

gewählt, wobei die Funktion  $\mathbf{w}(\mathbf{e})$  zu einem späteren Zeitpunkt erläutert wird.

Dadurch ergibt sich das Gesamtsystem zu

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}^d + \mathbf{w}(\mathbf{e}) + \mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}). \quad (7.24)$$

Durch Einführung der Matrizen-Schreibweise kann die Fehlerdynamik  $\dot{\mathbf{e}}$  erfasst werden.

$$\dot{\mathbf{e}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{F}} \mathbf{e} - \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{G}} \mathbf{w}(\mathbf{e}) - \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{G}} \mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) \quad (7.25)$$

$$\dot{\mathbf{e}} = \mathbf{F}\mathbf{e} - \mathbf{G}\mathbf{w}(\mathbf{e}) - \mathbf{G}\mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) \quad (7.26)$$

Für eine erste Betrachtung wird angenommen, dass  $\hat{\mathbf{M}}$  und  $\hat{\mathbf{h}}$  die nichtlinearen Kopplungen exakt kompensieren. Dies führt zur Auslöschung des Terms  $\mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})$  in Formel (7.26).

Die Aufgabe besteht desweiteren in der geeigneten Wahl eines Terms  $\mathbf{w}(\mathbf{e})$ , wodurch globale Stabilität des Regelkreises gewährleistet wird.

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0} \quad (7.27)$$

Einfache Wahl:

$$\begin{aligned} \mathbf{w}(\mathbf{e}) &= \mathbf{K}_P (\mathbf{q}^d - \mathbf{q}) + \mathbf{K}_D (\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) \\ &= \begin{bmatrix} \mathbf{K}_P & \mathbf{K}_D \end{bmatrix} \mathbf{e} \end{aligned} \quad (7.28)$$

Gesamtsystem:

$$\begin{aligned} \dot{\mathbf{e}} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{e} - \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{K}_P & \mathbf{K}_D \end{bmatrix} \mathbf{e} \\ &= \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_P & -\mathbf{K}_D \end{bmatrix}}_{\mathbf{A}} \mathbf{e} \end{aligned} \quad (7.29)$$

Um asymptotische Stabilität zu sichern, müssen die Eigenwerte der Matrix  $\mathbf{A}$  negativen Realteil besitzen. Dies führt zu der Bedingung, dass  $\mathbf{K}_P$  und  $\mathbf{K}_D$  durch positiv definite Matrizen beschrieben werden müssen.

$$\mathbf{K}_P, \mathbf{K}_D > 0 \quad (7.30)$$

Es gilt weiters zu beachten, dass die bisherige Analyse nur für perfekte Kompensation ( $\mathbf{z}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) = \mathbf{0}$ ) der nichtlinearen Terme gilt. Andernfalls muss die nichtkompensierbare Dynamik in einem zusätzlichen nichtlinearen Term erfasst werden.

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{k}(\mathbf{e}) \quad (7.31)$$

Durch  $\mathbf{k}(\mathbf{e})$  müssen Verfahren zur Stabilitätsanalyse nichtlinearer Systeme angewendet werden. Eine Methode, die sich vielfach bewährt hat, ist die *direkte Methode* von Ljapunov, welche auch in diesem Fall anwendbar ist. Mit Hilfe dieser Methodik kann durch geschickte Wahl einer Ljapunov-Funktion  $V(\mathbf{x})$  eine Aussage über die Stabilität hergeleitet werden. Eine ausführliche Analyse hierzu kann in [SS03] nachgelesen werden.

#### 7.2.4. Implementierung in Simulink

Dieser Abschnitt erläutert die Realisierung des in Abbildung 7.2 gezeigten Regelungskonzepts mit Hilfe der Blockset-Erweiterung Simulink. In Folge dessen ist eine eingehende Analyse und Simulation des dynamischen Systems möglich. Die implementierte Simulation setzt sich aus 3 Teilen zusammen:

- Initialisierung
- Simulation
- 1D-Visualisierung und Animation

In der Initialisierungsphase werden die statischen Parameter des Modells definiert. Die Eingabe der Zielpunkte (PTP Eingabe) erfolgt im Basiskoordinatensystem mittels einer grafischen Oberfläche. Der Trajektorienalgorithmus berechnet anhand der Eingabe automatisch die erforderlichen Trajektorien im Gelenkraum. Die Zeitspanne zwischen den anzufahrenden Punkten kann vom Benutzer gewählt werden. Abbildung 7.6

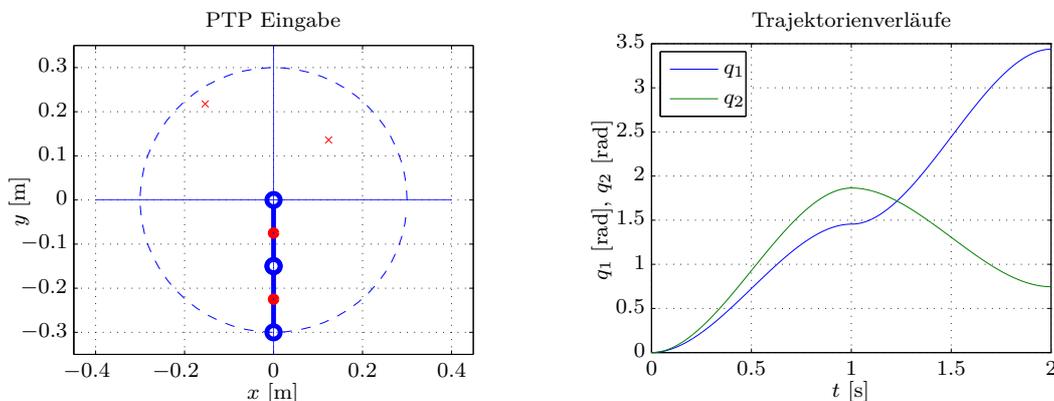


Abbildung 7.6.: Grafische Eingabe der Zielpunkte

veranschaulicht die Eingabe im Arbeitsraum und die resultierenden Verläufe der Gelenktrajektorien. Die Polynomkoeffizienten der Trajektorien werden für die einzelnen Zeitabschnitte in Vektorform abgespeichert.

In der Simulationsphase wird mittels der aktuellen Zeitinformation ein Koeffizientensatz aus dem Polynomvektor gelesen und ausgewertet. Der Simulink-Koppelplan des IDC-Verfahrens ist in Abbildung 7.8 dargestellt. Aufgrund der umfangreichen mathematischen Ausdrücke wurden in erster Linie *MATLAB-Funktionsblöcke* verwendet. Der abgebildete Block *Nonlinear Control* repräsentiert die globale Linearisierung durch eine nichtlineare Zustandsrückführung. Die äußere Schleife realisiert im Block *Robust Compensator* das lineare Regelungsgesetz und stabilisiert damit das Gesamtsystem. Für die tatsächliche Regelung des Modell musste ein weiterer Koppelplan entworfen werden, welcher die Anbindung an die Hardware sichert. Die beiden Versionen unterscheiden sich hauptsächlich durch die Verwendung unterschiedlicher Funktionsblöcke. Die Firma Quanser stellt zu diesem Zweck eine werkseigene Bibliothek zur Verfügung, wodurch die Hardware einfach in das modellbasierte Konzept integriert werden kann.

Aufbauend auf die regelungstechnische Implementierung des Konzepts wurden Experimente am realen Modell erprobt. Wie bereits angemerkt, startet jede Bewegung mit den Winkeln  $q_1 = 0$  und  $q_2 = 0$ . Die nachfolgende Abbildung 7.7 zeigt einen Vergleich zwischen Referenz und Experiment. Dabei wurden verschiedene Punkte im Arbeitsraum in einem Zeitintervall von einer Sekunde angefahren. Der Roboter befindet sich dabei in horizontaler Lage. Die geringe Abweichung zwischen den Verläufen aus Simulation und realem System spiegelt die überaus gute Performance der realisierten Regelung wieder.

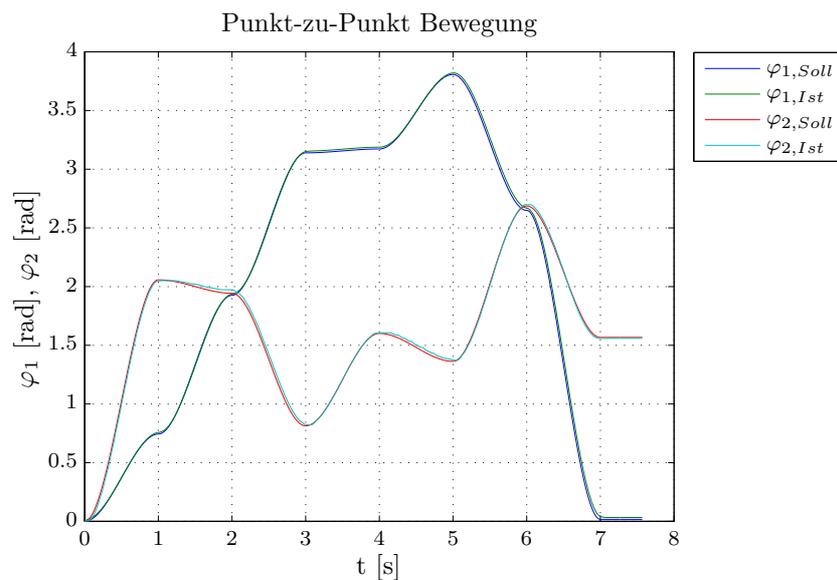


Abbildung 7.7.: Vergleich PTP-Bewegung

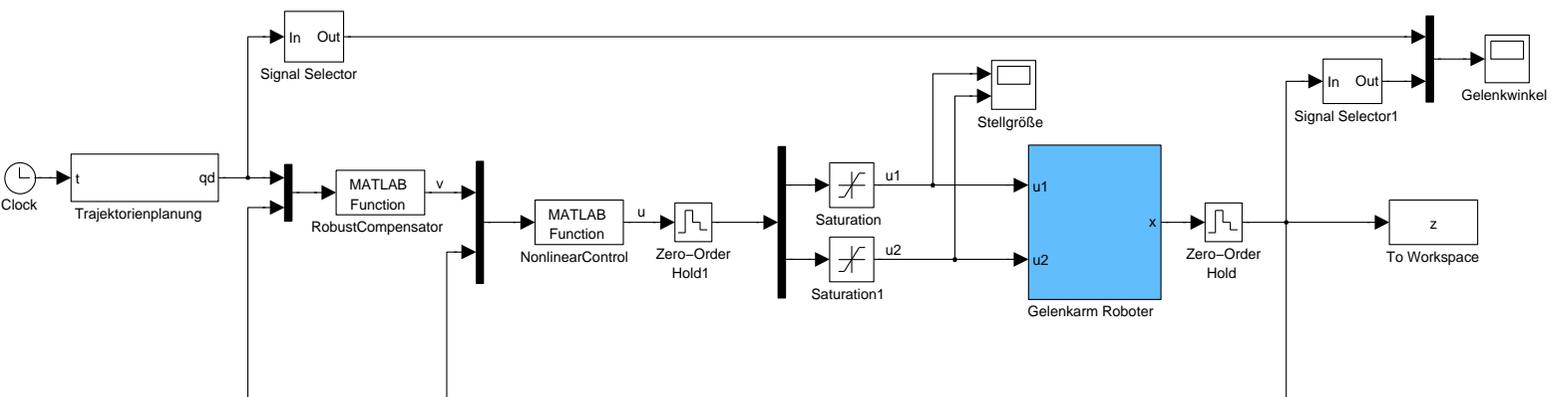


Abbildung 7.8.: IDC Simulationskoppelplan im Gelenkraum

### 7.3. Inverse Dynamic Control im Arbeitsraum

Grundsätzlich ist die Idee dieses Konzepts identisch zu dem in Abschnitt 7.2. Es wird wiederum eine Struktur mit 2 Schleifen eingesetzt. Die Umsetzung des Regelgesetzes findet jedoch nicht im Gelenkraum, sondern komplett im Arbeitsraum des Manipulators statt. Dadurch kann auch die Bewegung, wie in der Praxis meist üblich, im Arbeitsraum geplant werden. Der Vorteil dabei liegt klar auf der Hand, da die Referenztrajektorien nicht mehr als zeitliche Größen vorliegen müssen, sondern direkt in Parameterform eingespeist werden können. Den nachteiligen Effekt, den diese Idee jedoch mit sich bringt, ist der deutlich höhere Berechnungsaufwand. Der gesamte Formelapparat der direkten und differentiellen Kinematik 1. und 2. Ordnung wird für die Implementierung benötigt. Weiters ergeben sich auch zusätzliche Probleme in der Planung, welche in Abschnitt 6.3.2 bereits ausführlich untersucht wurden.

#### 7.3.1. Kinematische Beziehungen

Die globale Linearisierung, unter Zuhilfenahme einer nichtlinearen Rückkopplung, führt auch in diesem Fall zu einem System bestehend aus 2 Doppelintegrieren.

$$\ddot{\mathbf{q}} = \mathbf{v} \quad (7.32)$$

Anhand der bereits in Abschnitt 3.4 abgeleiteten Kinematikbeziehungen kann ein neuer Ausdruck für die Gelenkbeschleunigungen  $\ddot{\mathbf{q}}$  definiert werden.

$$\ddot{\mathbf{q}} = \mathbf{v} = \mathbf{J}_A^{-1} \cdot [\ddot{\mathbf{x}} - \dot{\mathbf{J}}_A \dot{\mathbf{q}}] \quad (7.33)$$

Die Umsetzung des Regelgesetzes erfolgt nun im Arbeitsraum, wodurch der Referenzeingang  $\mathbf{r}$  umformuliert werden muss. Dieser setzt sich nun aus den Komponenten für Beschleunigung, Geschwindigkeit und Position im Basiskoordinatensystem zusammen.

$$\mathbf{r} = \ddot{\mathbf{x}}^d + \mathbf{K}_D \dot{\mathbf{x}}^d + \mathbf{K}_P \mathbf{x}^d \quad (7.34)$$

Angelehnt an das Gesamtsystem in Formel (7.7) kann in abgewandelter Form folgender Ansatz für das mathematische Modell im Arbeitsraum getroffen werden:

$$\ddot{\mathbf{x}} + \mathbf{K}_D \dot{\mathbf{x}} + \mathbf{K}_P \mathbf{x} = \mathbf{r} \quad (7.35)$$

Das implementierte Regelgesetz kann durch Substitution der Beschleunigung  $\ddot{\mathbf{x}}$  in Formel (7.33) erreicht werden.

$$\mathbf{v} = \mathbf{J}_A^{-1} \cdot \left[ \ddot{\mathbf{x}}^d + \mathbf{K}_D(\dot{\mathbf{x}}^d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}^d - \mathbf{x}) - \dot{\mathbf{J}}_A \dot{\mathbf{q}} \right] \quad (7.36)$$

Die Struktur und Dimension der Reglermatrizen  $\mathbf{K}_P$  und  $\mathbf{K}_D$  sind ident zum vorherigen Fall im Gelenkraum. Es ist wiederum aus Sicht der Stabilität darauf zu achten, dass die Matrizen positiv definit gewählt werden. Wie bereits vermutet werden kann, führt das eingesetzte Regelgesetz zu einem enormen Berechnungsaufwand. In jedem Abtastzeitschritt müssen die Inverse Jacobi-Matrix  $\mathbf{J}_A^{-1}$  aus Formel (3.11) sowie die zeitliche Ableitung  $\dot{\mathbf{J}}_A$  aus (3.13) ausgewertet werden. Außerdem stellen die im Arbeitsraum auftretenden singulären Positionen eine besondere Herausforderung dar.

### 7.3.2. Regularisierung der Jacobi-Matrix

Die hier nun vorgestellte Methodik repräsentiert eine Möglichkeit die Robustheit des Regelkreises gegenüber Singularitäten zu erreichen [CCS96]. Der Grundgedanke der „Regularisierung der Jacobi-Matrix“ wurde bereits früh aus der numerischen Mathematik aufgegriffen und für Singularitätsbetrachtungen in der Robotik verwendet. Vom mathematischen Standpunkt aus wird mittels der Regularisierung numerische Robustheit erreicht.

Der Ansatz beruht auf der Idee, die Jacobi-Matrix nahe Singularitäten mit Hilfe eines Faktors zu dämpfen. Dadurch können überhöhte Gelenkgeschwindigkeiten, welche für die exakte Steuerung überaus störend sind, vermieden werden. Diese Problematik fällt im Modell insbesondere ins Gewicht, da der Roboter seine Bewegungen bereits aus einer singularen Position startet ( $q_1 = 0, q_2 = 0$ ). Deshalb ist die Verwendung einer angemessenen Methode bei diesem Regelungskonzept unumgänglich.

Der 1. Schritt besteht in der Definition einer Berechnungsvorschrift zur Ermittlung der gedämpften Form der inversen Jacobi-Matrix. Hierzu wird eine neue Größe  $\mathbf{J}_A^\#$  eingeführt.

$$\mathbf{J}_A^\# = \mathbf{J}_A^T \cdot \left( \mathbf{J}_A \mathbf{J}_A^T + \lambda^2 \mathbf{I} \right)^{-1} \quad (7.37)$$

Die Matrix  $\mathbf{J}_A^\#$  wird in der Literatur als *gedämpfte least-squares Inverse* oder auch *DLS-Inverse* bezeichnet. In Formel (7.37) stellt  $\mathbf{I}$  die Einheitsmatrix und  $\lambda$  den Dämpfungsfaktor dar. Es ist deutlich ersichtlich, dass sich für  $\lambda = 0$  die ursprüngliche Inverse der Jacobi-Matrix ergibt, sofern diese existiert. In Anbetracht der Tatsache, dass eine Dämpfung nur nahe und direkt in einer singulären Position notwendig ist, muss  $\lambda$  dynamisch in Abhängigkeit der Endeffektorstellung ermittelt werden. Je geringer die

Entfernung zu einer Singularität ist, umso stärker muss die Matrix gedämpft werden. Für den Wert von  $\lambda$  muss jedoch ein Kompromiss getroffen werden. Kleine Werte führen zu einer akkuraten Lösung und sichern somit gute Tracking-Performance. Allerdings bringt diese Wahl wenig Robustheit gegenüber naheliegender Singularitäten mit sich. Werden große Werte für  $\lambda$  gewählt, so ist zwar Robustheit gegeben, allerdings fehlt es dem Endeffektor an Genauigkeit.

Ein effizientes Werkzeug zur Untersuchung von singulären Positionen stellt die *Singulärwertzerlegung* (SVD für Singular Value Decomposition) dar. Mit ihr ist die mathematische Definition einer geometrischen Zone um die Singularität möglich. Hierzu wird die Jacobi-Matrix  $\mathbf{J}_A$  in das Produkt dreier spezieller Matrizen zerlegt

$$\mathbf{J}_A = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \quad (7.38)$$

wobei  $\mathbf{U}$  und  $\mathbf{V}$  in diesem Fall orthonormale  $(n \times n)$ -Matrizen und  $\mathbf{u}_i$  und  $\mathbf{v}_i$  deren  $i$ -te Spaltenvektoren darstellen. Für diese Matrizen gilt also die Eigenschaft

$$\mathbf{U}^T \mathbf{U} = \mathbf{I} \text{ und } \mathbf{V}^T \mathbf{V} = \mathbf{I}. \quad (7.39)$$

Die Matrix  $\mathbf{\Sigma}$  bezeichnet eine  $(n \times n)$ -Diagonalmatrix, welche in der Hauptdiagonale die Singulärwerte  $\sigma_i$  beinhaltet.

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix} \quad (7.40)$$

Für die Singulärwerte der Jacobi-Matrix gilt:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0 \quad (7.41)$$

Besitzt  $\mathbf{J}_A$  vollen Rang  $r = n$ , so sind alle Singulärwerte von Null verschieden. Liegt eine singuläre Position vor, so kommt es infolgedessen zu einem Rangabfall  $r < n$ , wobei die letzten  $(n - r)$  Singulärwerte 0 sind. Für die DLS-Inverse  $\mathbf{J}_A^\#$  kann nun ebenfalls eine Singulärwertzerlegung durchgeführt werden.

Zum besseren Verständnis wird vorerst nur der Klammerausdruck in Formel (7.37) behandelt:

$$\begin{aligned} \mathbf{J}_A \mathbf{J}_A^T + \lambda^2 \mathbf{I} &= \mathbf{U} \underbrace{\boldsymbol{\Sigma} \mathbf{V}^T \mathbf{V} \boldsymbol{\Sigma}}_{\mathbf{I}} \mathbf{U}^T + \lambda^2 \mathbf{I} = \\ &= \mathbf{U} \boldsymbol{\Sigma}^2 \mathbf{U}^T + \lambda^2 \mathbf{I} = \mathbf{U} \underbrace{(\boldsymbol{\Sigma}^2 + \lambda^2 \mathbf{I})}_{\mathbf{D}} \mathbf{U}^T \end{aligned} \quad (7.42)$$

Die Matrix  $\mathbf{D}$  beschreibt hierbei eine Diagonalmatrix mit Elementen der Form  $\sigma_i^2 + \lambda_i^2$ . Das erhaltene Ergebnis kann nun für die vollständige SVD der gedämpften least-squares Inversen verwendet werden.

$$\begin{aligned} \mathbf{J}_A^\# &= \mathbf{J}_A^T \cdot (\mathbf{J}_A \mathbf{J}_A^T + \lambda^2 \mathbf{I})^{-1} = \mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^T \cdot (\mathbf{U} \mathbf{D} \mathbf{U}^T)^{-1} = \\ &= \mathbf{V} \boldsymbol{\Sigma} \underbrace{\mathbf{U}^T \mathbf{U}^{-T}}_{\mathbf{I}} \mathbf{D}^{-1} \mathbf{U}^T = \mathbf{V} \underbrace{\boldsymbol{\Sigma} \mathbf{D}^{-1}}_{\mathbf{D}^\#} \mathbf{U}^T \end{aligned} \quad (7.43)$$

Für  $\mathbf{D}^\#$  ergibt sich ebenfalls die Form einer Diagonalmatrix, allerdings mit den veränderten Elementen  $\frac{\sigma_i}{\sigma_i^2 + \lambda^2}$  in der Hauptdiagonale. Der Ausdruck für  $\mathbf{J}_A^\#$  lässt sich nun wie folgt anschreiben:

$$\mathbf{J}_A^\# = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \mathbf{u}_i^T \quad (7.44)$$

Aus Formel (7.44) lässt sich ableiten, dass Komponenten im Fall  $\sigma_i \gg \lambda$  nur vernachlässigbar klein beeinflusst werden. Für diesen Umstand kann die Näherung

$$\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \approx \frac{1}{\sigma_i} \quad (7.45)$$

getroffen werden. Aus obiger Formel geht hervor, dass der Dämpfungsfaktor  $\lambda$  somit nur in der unmittelbaren Nähe von Singularitäten Auswirkungen hat. Bewegt sich der Manipulator auf eine singuläre Position zu, so strebt der kleinste Singulärwert  $\sigma_i$  gegen 0 und die damit verbundene Komponente wird mit dem Faktor  $\sigma_i/\lambda^2$  gedämpft. Im Gegensatz dazu würden im Fall der exakten Inversen  $\mathbf{J}_A^{-1}$  die Komponenten stetig an Wert zunehmen. In weiterer Folge würde dies zu überhöhten Gelenkgeschwindigkeiten führen.

Der kleinste Singulärwert  $\sigma_{min}$  gilt als zuverlässige Kenngröße, um die Nähe zu Singularitäten zu detektieren [CCS96]. Auf Basis des ermittelten Wertes für  $\sigma_{min}$  kann die in (7.46) dargestellte Wahl für den Dämpfungsfaktor getroffen werden, wobei  $\varepsilon$  den Bereich der Singularitätszone und  $\lambda_{max}$  die maximale Dämpfung darstellt.

Der Wert für  $\lambda_{max}$  wurde experimentell anhand von Messungen am Roboter ermittelt.

$$\lambda^2 = \begin{cases} 0 & \sigma_{min} \geq \varepsilon \\ \left(1 - \left(\frac{\sigma_{min}}{\varepsilon}\right)^2\right) \lambda_{max}^2 & \text{sonst} \end{cases} \quad (7.46)$$

### 7.3.3. Implementierung in Simulink

Der Aufbau der Simulation ist ähnlich wie im Fall des IDC-Verfahrens im Gelenkwinkelraum. Für die Planung der Trajektorien wurde jedoch die bereits in Abschnitt 6.3.1 besprochene Strategie verwendet. Durch parametrische Beschreibung der Bewegung können so geometrische Figuren wie Kreis, Ellipse und Geraden vom Endeffektor abefahren werden. Weiters können auch zusammengesetzte Bewegungen, wie in Abbildung 7.9 gezeigt, realisiert werden.

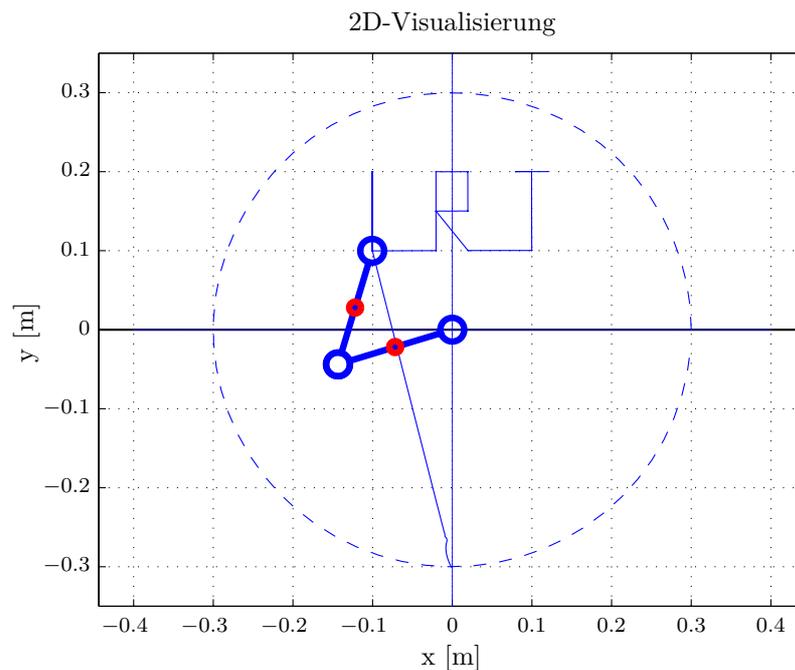


Abbildung 7.9.: Zusammengesetzte Bewegung

In Abbildung 7.10 ist der implementierte Koppelplan ersichtlich. Besonders im linken Teil des Konzepts lässt sich eine grundsätzliche Veränderung gegenüber Abbildung 7.8 feststellen. An dieser Stelle findet mit Hilfe der Blöcke  $Jpa$ ,  $Ja$  und  $k(q)$  eine Transformation vom Gelenk- in den Arbeitsraum nach Formel (3.3), (3.10) und (3.12) statt. Dadurch kann das in Formel (7.36) angegebene Regelgesetz in den Koppelplan integriert werden. Der MATLAB-Block  $invJa$  führt die Rücktransformation in den Gelenkraum und die Regularisierung der Jacobi-Matrix durch. Der im Koppelplan eingesetzte Block

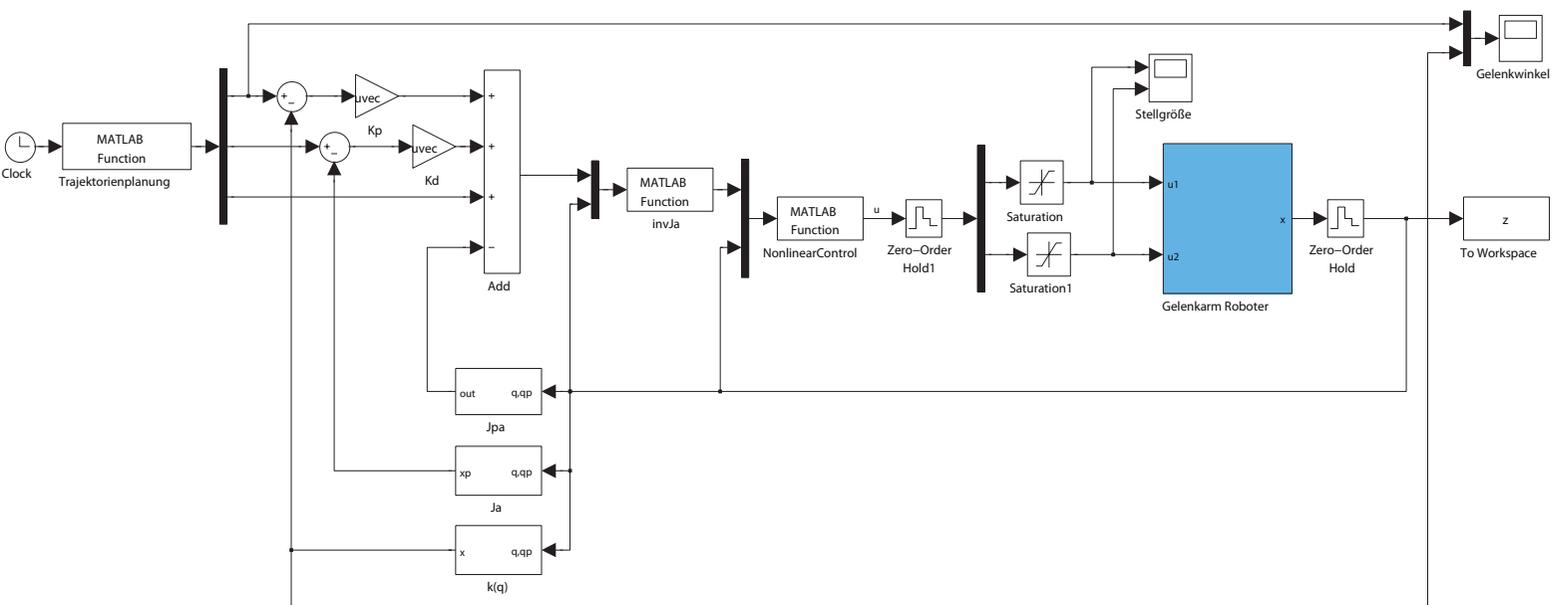


Abbildung 7.10.: IDC Simulationskoppelplan im Arbeitsraum

*Trajektorienplanung* liefert die nötigen Referenzgrößen  $\mathbf{x}^d$ ,  $\dot{\mathbf{x}}^d$  und  $\ddot{\mathbf{x}}^d$  für die Regelung. Die Planung dieser Trajektorien kann im gesamten Arbeitsraum des Roboters vorgenommen werden und unterliegt keinerlei Beschränkungen. Selbst auf Singularitäten muss nicht geachtet werden, da mit Hilfe der Regularisierung ein geeignetes Verhalten des Manipulators erzielt wird.

Nach erfolgreichen Simulationen am theoretischen Modell wurden schließlich Experimente am realen Modell durchgeführt. Die folgende Abbildung 7.11 zeigt die Verläufe der kartesischen Koordinaten  $x$  und  $y$  des Manipulators. Zu Beginn ist gut zu erkennen, wie eine geradlinige Trajektorie vom Ursprung in den Startpunkt der Figur gefahren wird. Aufgrund der Singularität in der Ursprungsposition kommt es hier kurzzeitig zu Abweichungen vom Solltrajektorienverlauf. Ist die Startposition erreicht, führt der Roboter nach einer kleinen zeitlichen Verzögerung die gewünschte Bewegung aus.

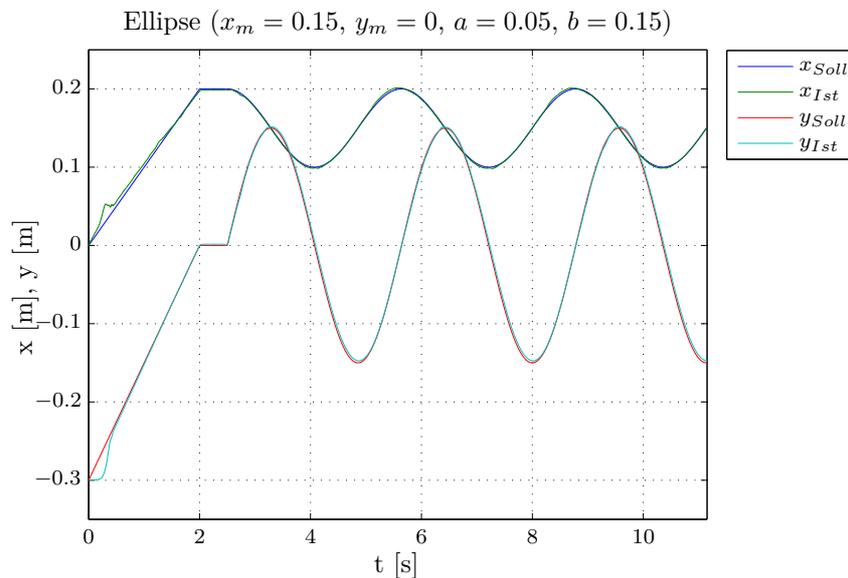


Abbildung 7.11.: Vergleich Pfad-Bewegung

## 7.4. Robust Control

Die exakte Positionsregelung eines Roboters erfordert die Verwendung ausgereifter Konzepte. Weiters muss für den Reglerentwurf und die Simulation auch das mathematische Modell der Strecke ein möglichst gutes Abbild der Realität darstellen. Bisher wurden Konzepte vorgestellt, die unter der Voraussetzung implementiert wurden, dass exakte Kompensation der nichtlinearen Terme vorliegt. Dass dies in der Praxis nur bedingt der Fall sein kann, liegt aufgrund von Parameterschwankungen und Unsicherheiten in der Modellierungsdynamik auf der Hand. Deshalb sind Fragestellungen bzgl. Robustheit und Sensitivität gerechtfertigt.

### 7.4.1. Idee

Das Konzept *Robust Control* besitzt nahezu dasselbe Regelungsgesetz wie die im Abschnitt *Inverse Dynamic Control im Gelenkraum* behandelte Methode. Um Robustheit zu erlangen, wird allerdings ein zusätzlicher Term  $\Delta \mathbf{v}$  addiert, welcher anschließend eingehend diskutiert wird. Dieser Term wird dazu benutzt, mögliche Ungenauigkeiten in der Feedback Linearisierung auszugleichen.

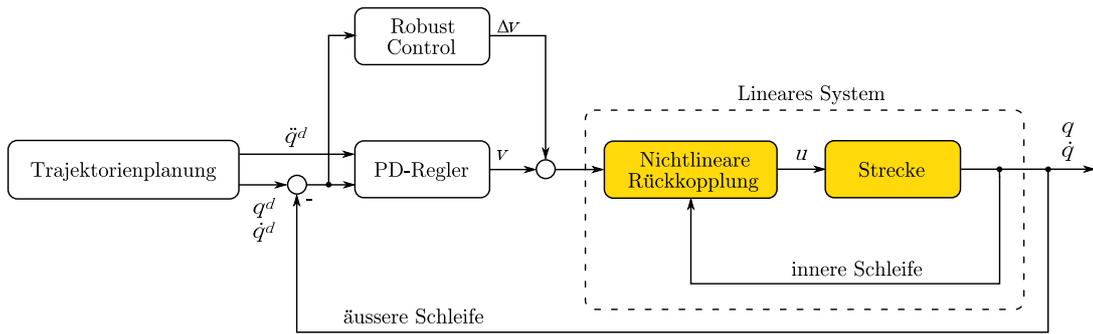


Abbildung 7.12.: Regelkreisstruktur - Robust Control

Hinsichtlich der nicht perfekten Kompensation kann das Regelgesetz anhand von geschätzten Matrizen und dem daraus resultierenden Modellfehlern

$$\begin{aligned}\Delta \mathbf{M} &= \hat{\mathbf{M}}(\mathbf{q}) - \bar{\mathbf{M}}(\mathbf{q}) \\ \Delta \mathbf{h} &= \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})\end{aligned}\quad (7.47)$$

angegeben werden.

$$\mathbf{u} = \hat{\mathbf{M}}(\mathbf{q})\mathbf{v} + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})\quad (7.48)$$

Formel (7.48) eingesetzt in das Gesamtsystem ergibt

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} + \mathbf{h} = \hat{\mathbf{M}}\mathbf{v} + \hat{\mathbf{h}},\quad (7.49)$$

was weiters umgeformt zu folgender Beziehung führt:

$$\begin{aligned}\ddot{\mathbf{q}} &= \bar{\mathbf{M}}^{-1} [\hat{\mathbf{M}}\mathbf{v} + \hat{\mathbf{h}} - \mathbf{h}] \\ &= \mathbf{v} + \underbrace{[\bar{\mathbf{M}}^{-1}\hat{\mathbf{M}} - \mathbf{I}]\mathbf{v} + \bar{\mathbf{M}}^{-1}\Delta\mathbf{h}}_{-\boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})} \\ &= \mathbf{v} - \boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})\end{aligned}\quad (7.50)$$

Der Term  $\boldsymbol{\eta}$  steht für die Unsicherheit in der Modellierung der Dynamik des Roboters. Für den Eingang  $\mathbf{v}$  des Systems gilt nach wie vor die Festlegung

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}^d - \mathbf{q}), \quad (7.51)$$

wobei mit

$$\mathbf{e} = \mathbf{q}^d - \mathbf{q} \quad (7.52)$$

eine neue vereinfachte Darstellung erreicht wird.

$$\mathbf{v} = \ddot{\mathbf{e}} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e} + \ddot{\mathbf{q}} \quad (7.53)$$

Bei Berücksichtigung der Unsicherheit ergibt sich

$$\boldsymbol{\eta} = \ddot{\mathbf{e}} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e}, \quad (7.54)$$

was natürlich Auswirkungen auf die Stabilitätsprüfung hat. Asymptotische Stabilität des Regelfehlers ist in diesem Fall nicht durch die Wahl eines einfachen PD-Reglers gesichert. Zur Untersuchung müssen nichtlineare Methoden angewendet werden.

#### 7.4.2. Robustes Regelgesetz

Der Regelfehler sowie deren zeitliche Ableitungen können zusammengefasst mittels einer geplanten Trajektorie im Gelenkraum in der Form

$$\begin{aligned} \mathbf{e} &= \mathbf{q}^d - \mathbf{q} \\ \dot{\mathbf{e}} &= \dot{\mathbf{q}}^d - \dot{\mathbf{q}} \\ \ddot{\mathbf{e}} &= \ddot{\mathbf{q}}^d - \ddot{\mathbf{q}} = \ddot{\mathbf{q}}^d - \mathbf{v} + \boldsymbol{\eta} \end{aligned} \quad (7.55)$$

dargestellt werden. Mit Hilfe der Definition

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} \quad (7.56)$$

wird die Dynamik des Fehlers kompakt in Matrizen-Schreibweise erfasst.

$$\dot{\xi} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{H}} \xi + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{D}} (\ddot{\mathbf{q}}^d - \mathbf{v} + \boldsymbol{\eta}) \quad (7.57)$$

$$\dot{\xi} = \mathbf{H}\xi + \mathbf{D} (\ddot{\mathbf{q}}^d - \mathbf{v} + \boldsymbol{\eta}) \quad (7.58)$$

Die Matrizen  $\mathbf{H}$  und  $\mathbf{D}$  repräsentieren Blockmatrizen der Dimension  $(2n \times 2n)$  und  $(2n \times n)$ . Durch diese Umstrukturierung kann die Zielvorstellung, einer Solltrajektorie  $\mathbf{q}^d(t)$  zu folgen, neu definiert werden. Es gilt nun ein Regelgesetz  $\mathbf{v}$  zu finden, welches das nichtlineare System in Gleichung (7.58) stabilisiert.

Obgleich die Unsicherheit  $\boldsymbol{\eta}$  in (7.58) nicht bekannt ist, ist es möglich, *worst-case* Grenzen für Schwankungen von  $\boldsymbol{\eta}$  anzugeben. Es werden folgende Annahmen getroffen:

$$\sup_{t \geq 0} \|\ddot{\mathbf{q}}^d\| < Q_M < \infty \quad \forall \ddot{\mathbf{q}}^d \quad (7.59)$$

$$\|\mathbf{I} - \bar{\mathbf{M}}^{-1} \hat{\mathbf{M}}\| \leq \alpha < 1 \quad \forall \mathbf{q} \quad (7.60)$$

$$\|\Delta \mathbf{h}\| \leq \Phi < \infty \quad \forall \mathbf{q}, \dot{\mathbf{q}} \quad (7.61)$$

Die 1. Annahme ist in der Praxis meist erfüllt, da eine sinnvoll geplante Trajektorie keine unendlich hohe Beschleunigung benötigt. Aus Formel (7.50) lässt sich erkennen, dass sich auch die 2. Annahme direkt auf die Unsicherheit  $\boldsymbol{\eta}$  bezieht. Es handelt sich dabei um eine sehr restriktive Annahme, da sie die genaue Modellierung der Massenmatrix  $\hat{\mathbf{M}}$  voraussetzt. Nur unter der Erfüllung dieser Forderung kann dieses Konzept überhaupt angewendet werden. Es zeigt sich jedoch aufgrund der positiven Definitheit der Matrix  $\bar{\mathbf{M}}$ , dass Schranken für  $\bar{\mathbf{M}}$  gefunden werden können, welche folgende Darstellung ermöglichen:

$$0 < \bar{M}_U \leq \|\bar{\mathbf{M}}^{-1}\| \leq \bar{M}_O < \infty \quad \forall \mathbf{q} \quad (7.62)$$

Trifft man nun für  $\hat{\mathbf{M}}$  die Wahl

$$\hat{\mathbf{M}} = \frac{2}{\bar{M}_U + \bar{M}_O} \mathbf{I}, \quad (7.63)$$

kann gezeigt werden, dass Ungleichung

$$\|\bar{\mathbf{M}}^{-1} \hat{\mathbf{M}} - \mathbf{I}\| \leq \frac{\bar{M}_O - \bar{M}_U}{\bar{M}_O + \bar{M}_U} = \alpha < 1 \quad (7.64)$$

gilt. Je genauer also die Approximation der Massenmatrix vorliegt, desto kleiner kann der Faktor  $\alpha$  gewählt werden. Im Idealfall und somit perfekter Kompensation würde sich  $\alpha = 0$  ergeben.

Die Dritte und letzte Annahme bezieht sich auf die Norm des Terms  $\Delta \mathbf{h}$ , wobei dieser eine Funktion der Variablen  $\mathbf{q}$  und  $\dot{\mathbf{q}}$  darstellt. Nehmen wir beispielsweise an, der Regelkreis wäre instabil und könnte die Gelenkgeschwindigkeiten nicht innerhalb von bestimmten Geschwindigkeitsgrenzen beschränken. Theoretisch könnten die Geschwindigkeiten daher ins Unermessliche steigen. Für die praktische Implementierung des Roboters ergibt sich jedoch durch die maximale Drehzahl der Gelenkmotoren eine physikalische Beschränkung. Aufgrund dessen ist in der Realität auch diese Annahme erfüllt.

In Bezug auf die getroffenen 3 Annahmen kann für das Regelgesetz die Wahl

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} + \Delta \mathbf{v} \quad (7.65)$$

getroffen werden. Wobei  $\mathbf{K}_D$  und  $\mathbf{K}_P$  den äußeren Regelkreis stabilisieren und mit Hilfe des zusätzlichen Terms  $\Delta \mathbf{v}$  Robustheit erlangt wird. Die Struktur der beiden Reglermatrizen bleibt gleich wie im Fall Inverse Dynamic Control. Um jedoch eine kompaktere Formulierung zu ermöglichen, wird eine neue Matrix  $\mathbf{K}$  eingeführt.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_P & \mathbf{K}_D \end{bmatrix} \quad (7.66)$$

Dadurch ergibt sich für die Fehlerdynamik in Formel (7.58)

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \underbrace{(\mathbf{H} - \mathbf{DK})}_{\tilde{\mathbf{H}}} \boldsymbol{\xi} + \mathbf{D}(\boldsymbol{\eta} - \Delta \mathbf{v}) \\ &= \tilde{\mathbf{H}} \boldsymbol{\xi} + \mathbf{D}(\boldsymbol{\eta} - \Delta \mathbf{v}). \end{aligned} \quad (7.67)$$

Die Matrix

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_P & -\mathbf{K}_D \end{bmatrix} \quad (7.68)$$

besitzt für positiv definite Reglermatrizen die Hurwitz-Eigenschaft.

Für die Festlegung von  $\Delta \mathbf{v}$  sei vorerst angenommen, es existiere eine Funktion  $\rho(\mathbf{e}, t)$ , welche die folgenden Ungleichungen erfüllt.

$$\|\boldsymbol{\eta}\| < \rho(\mathbf{e}, t) \quad (7.69)$$

$$\|\Delta \mathbf{v}\| < \rho(\mathbf{e}, t) \quad (7.70)$$

Für diese Funktion kann unter Verwendung von (7.50) und (7.65) eine Abschätzung getroffen werden.

$$\begin{aligned}\|\boldsymbol{\eta}\| &\leq \left\| \mathbf{I} - \bar{\mathbf{M}}^{-1} \hat{\mathbf{M}} \right\| \cdot \left\| \ddot{\mathbf{q}}^d + \mathbf{K}\boldsymbol{\xi} + \Delta\mathbf{v} \right\| + \left\| \bar{\mathbf{M}}^{-1} \right\| \|\Delta\mathbf{h}\| \\ &\leq \alpha Q_M + \alpha \|\mathbf{K}\| \|\boldsymbol{\xi}\| + \alpha\rho + \bar{M}_O\Phi\end{aligned}\quad (7.71)$$

Ungleichung (7.71) wird nun nach  $\rho$  aufgelöst, wodurch die Bestimmungsgleichung für  $\rho$  berechnet werden kann.

$$\rho \geq \frac{1}{1-\alpha} \cdot \left[ \alpha Q_M + \alpha \|\mathbf{K}\| \|\boldsymbol{\xi}\| + \bar{M}_O\Phi \right] \quad 0 < \alpha < 1 \quad (7.72)$$

Da  $\tilde{\mathbf{H}}$  eine Hurwitz-Matrix ist, kann ohne Einschränkung der Allgemeinheit die Gleichung

$$\tilde{\mathbf{H}}^T \mathbf{Q} + \mathbf{Q}\tilde{\mathbf{H}} + \mathbf{P} = \mathbf{0} \quad (7.73)$$

angegeben werden, welche auch unter dem Namen Lyapunov-Gleichung bekannt ist. Dieser Ausdruck scheint zunächst aus der Luft gegriffen. Er wird allerdings im nächsten Abschnitt für den Stabilitätsbeweis des Konzepts von entscheidender Bedeutung sein. Für  $\mathbf{P}$  wird eine symmetrische und positiv definite  $(2n \times 2n)$ -Matrix gewählt.  $\mathbf{Q}$  kann als Lösung der Lyapunov Gleichung interpretiert werden. Der in der Literatur [SS03] vorgeschlagene Ansatz zur Wahl von  $\Delta\mathbf{v}$  lautet:

$$\Delta\mathbf{v} = \begin{cases} \rho \frac{\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}}{\|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\|} & \text{für } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| \neq 0 \\ \mathbf{0} & \text{für } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| = 0 \end{cases} \quad (7.74)$$

### 7.4.3. Stabilitätsbetrachtungen

Zur Anwendung der 2. Methode von Lyapunov wird zunächst eine Funktion  $V$  der Form

$$V(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \mathbf{Q} \boldsymbol{\xi} > 0 \quad \forall \boldsymbol{\xi} \neq \mathbf{0} \quad (7.75)$$

gewählt. Aufgrund der entsprechenden Berechnung von  $\mathbf{Q}$  aus (7.73) ist positive Definitheit der Funktion  $V$  gesichert.

Die Lyapunov-Funktion nach der Zeit  $t$  abgeleitet, liefert

$$\begin{aligned}\dot{V}(\boldsymbol{\xi}) &= \dot{\boldsymbol{\xi}}^T \mathbf{Q} \boldsymbol{\xi} + \boldsymbol{\xi}^T \mathbf{Q} \dot{\boldsymbol{\xi}} \\ &= \boldsymbol{\xi}^T \left( \tilde{\mathbf{H}}^T \mathbf{Q} + \mathbf{Q} \tilde{\mathbf{H}} \right) \boldsymbol{\xi} + 2 \boldsymbol{\xi}^T \mathbf{Q} \mathbf{D} (\boldsymbol{\eta} - \Delta \mathbf{v}) \\ &= -\boldsymbol{\xi}^T \mathbf{P} \boldsymbol{\xi} + 2 \boldsymbol{\xi}^T \mathbf{Q} \mathbf{D} (\boldsymbol{\eta} - \Delta \mathbf{v})\end{aligned}\quad (7.76)$$

Für den 1. Term in Formel (7.76) gilt dank der Lyapunov-Gleichung, dass dieser garantiert negativ definit ist. Somit muss für den Beweis der Stabilität nur mehr gezeigt werden, dass dies auch für den 2. Term gilt. Zur besseren Lesbarkeit wird eine zusätzliche Definition eingeführt.

$$\mathbf{w} = \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi} \quad (7.77)$$

Für  $\mathbf{w} = 0$  reduziert sich die Gleichung auf den 1. Term und kann damit für die Stabilitätsbetrachtungen abgehakt werden.

Für  $\mathbf{w} \neq 0$  gilt für

$$\Delta \mathbf{v} = \rho \frac{\mathbf{w}}{\|\mathbf{w}\|}, \quad (7.78)$$

wodurch unter Vernachlässigung des Faktors 2 in Formel (7.76) der 2. Term umgeschrieben werden kann.

$$\begin{aligned}\mathbf{w}^T \left( \boldsymbol{\eta} - \rho \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) &= \mathbf{w}^T \boldsymbol{\eta} - \frac{\rho \mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} \\ \left\| \mathbf{w}^T \boldsymbol{\eta} - \frac{\rho \mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} \right\| &\leq \|\mathbf{w}\| \|\boldsymbol{\eta}\| - \rho \|\mathbf{w}\| \\ &= \|\mathbf{w}\| (\|\boldsymbol{\eta}\| - \rho) \leq 0\end{aligned}\quad (7.79)$$

Es konnte somit gezeigt werden, dass es sich bei  $\dot{V}$  um eine negativ definite Funktion handelt. An dieser Stelle muss der Vollständigkeit halber angemerkt werden, dass globale asymptotische Stabilität nur für den Fall von Lösungstrajektorien des Systems gezeigt wurde. In [Gut79] wird jedoch bewiesen, dass Stabilität auch im allgemeinen Fall gegeben ist.

Für die praktische Implementierung hat sich jedoch gezeigt [SV89], dass die Wahl für  $\Delta \mathbf{v}$  laut (7.74) nicht optimal ist. Wie häufig bei diskontinuierlichen Regelgesetzen kommt es zu hochfrequenten Umschaltungen. Dieses Phänomen ist unter dem Begriff *Chattering* bekannt. Aus diesem Grund wird für die Realisierung eine leicht veränderte Konvention für  $\Delta \mathbf{v}$  getroffen, in der  $\varepsilon$  größer 0 zu wählen ist.

$$\Delta \mathbf{v} = \begin{cases} \rho \frac{\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}}{\|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\|} & \text{für } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| \geq \varepsilon \\ \frac{\rho}{\varepsilon} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi} & \text{für } \|\mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}\| < \varepsilon \end{cases} \quad (7.80)$$

Obige Festlegung für  $\Delta \mathbf{v}$  wurde in das robuste Regelgesetz implementiert und am Roboter erfolgreich getestet. Die Messergebnisse lieferten jedoch nur geringe Veränderungen in der Positioniergenauigkeit des Endeffektors. Daraus lässt sich schließen, dass die Dynamik des Manipulators im mathematischen Modell zufriedenstellend erfasst wurde. In der Praxis würde sich das robuste Regelungskonzept besonders anbieten, wenn Parameterschwankungen nicht auszuschließen sind. Beispielsweise wäre dies bei möglichen Lastwechseln am Endeffektor durch eine zusätzliche Masse möglich. In diesem Fall könnte das robuste Konzept seine Stärken durch Berücksichtigung der Modellierungsunsicherheit  $\boldsymbol{\eta}$  ausspielen.

## 8. Zusammenfassung

### 8.1. Fazit

Im Rahmen der hier vorgestellten Masterarbeit wurde die Positionsregelung eines 2-Gelenkarm Roboters behandelt. Beide Gelenke des Roboters sind vollaktuiert und mit Inkrementalencodern zur Winkelerfassung ausgestattet. Die Konstruktion und Fertigung des Labormodells erfolgte am Institut für Regelungs- und Automatisierungstechnik der Technischen Universität Graz. Standardmäßig befindet sich der Roboter in aufrechter Position, was der vertikalen Arbeitslage entspricht. Die mechanische Konstruktion ermöglicht jedoch auch die horizontale Lage des Manipulators. Mit Hilfe des Lagrange-Formalismus konnte das mathematische Modell des nichtlinearen Systems hergeleitet werden. Die Bewegungsgleichungen des Roboters wurden dabei für den horizontalen- und vertikalen Fall ermittelt. Um ein vollständiges Modell zu erhalten, musste eine Parameteridentifikation durchgeführt werden. Diese erfolgte anhand von Experimenten am realen Laboraufbau. Zur Eruiierung des theoretischen Modells wurde das Simulationswerkzeug Simulink verwendet. Die Simulationsergebnisse wurden zudem durch eine Animation erweitert, um die Dynamik des Roboters besser verstehen zu können. Die Zielsetzung der Regelung war es, Punkt-zu-Punkt Bewegungen und die Bewegung entlang von geometrischen Figuren zu ermöglichen. Die regelungstechnische Grundlage bildeten die Kinematik des Manipulators sowie das Modell aus der dynamischen Modellbildung. In Kapitel 6 wurde gezeigt, wie Trajektorien im Gelenk- und Arbeitsraum geplant werden und welche Probleme dabei auftreten können.

Als erstes Regelungsverfahren wurde das Konzept *Inverse Dynamic Control im Gelenkraum* präsentiert. Dabei wurde mit Hilfe eines nichtlinearen Rückkoppelterms globale Linearisierung erreicht, wodurch ein vereinfachtes lineares Modell ermittelt werden konnte. Das resultierende System wird dann mit Hilfe eines PD-Reglers stabilisiert. Die Referenzgrößen des Regelkreises werden durch die Verläufe von Position, Geschwindigkeit und Beschleunigung der Gelenkkoordinaten beschrieben.

Das zweite Verfahren kann als Erweiterung des IDC-Konzepts im Gelenkraum interpretiert werden. Dabei wurde die Struktur so modifiziert, dass das Regelgesetz zur Steuerung des Manipulators im Arbeitsraum definiert werden konnte. Die dadurch auftretenden singulären Positionen konnten durch eine Regularisierung der Jacobi-Matrix effektiv behandelt werden. Somit kann die Planung im Arbeitsraum ohne Einschränkungen durchgeführt werden. Dieses Verfahren weist im Gegensatz zum ersten einen

hohen Berechnungsaufwand auf. Pro Abtastschritt müssen aufwendige Kalkulationen aufgrund der Koordinatentransformationen durchgeführt werden. Auch die Regularisierung der Jacobi-Matrix benötigt für die Singulärwertzerlegung zusätzliche Rechenkapazität. Beide Konzepte konnten jedoch mit der verfügbaren Hardware am Modell erfolgreich implementiert und getestet werden und lieferten hervorragende Ergebnisse.

Die praktische Realisierung des idealen IDC-Konzepts gestaltet sich dennoch schwierig, da für die globale Linearisierung exakte Kompensation der Nichtlinearitäten vorausgesetzt wird. In der Realität ist die Bestimmung der "exakten" Systemparameter jedoch nur bedingt möglich oder wäre mit enormen Aufwand verbunden. In weiterer Folge ergeben sich daher für die Implementierung folgende Probleme:

- Parameterschwankungen
- Modellunsicherheiten
- Messungenauigkeiten der Encoder

Aus diesem Grund wurde im letzten Abschnitt ein Konzept vorgestellt, welches sich der Thematik Robustheit widmet. Das Regelgesetz wird dabei um einen zusätzlichen Term erweitert, welcher Unsicherheiten in der Modellierung kompensieren soll. Für die Festlegung wurden 3 Annahmen getroffen, die es für den erfolgreichen Einsatz des Konzepts zu erfüllen gilt. Mit Hilfe der 2. Methode von Lyapunov konnte schließlich ein Ansatz zur Wahl des zusätzlichen Faktors gefunden werden.

## 8.2. Ausblick

Aufbauend auf die präsentierten Regelungsmechanismen bietet sich vor allem das 2. Konzept für Erweiterungen in Bezug auf den Berechnungsaufwand an. Zwar können die Transformationen zwischen den einzelnen Koordinatensystemen nicht manipuliert werden, es kann allerdings für die Regularisierung, wie in [Chi93] gezeigt, eine Verbesserung hinsichtlich der benötigten Rechenkapazität bewirkt werden. Dabei wird der minimale Singulärwert nicht mehr mit Hilfe einer kompletten Singulärwertzerlegung ermittelt, sondern über Abschätzungen festgelegt.

Ein weiteres großes Themengebiet, das im Hinblick auf die Parameterschwankungen geradezu prädestiniert wäre, stellt die adaptive Regelung dar. Konzepte und Ideen zu dieser Thematik sind in [SV89] und [SS03] zu finden.

Als spielerische Erweiterung sei noch erwähnt, dass aus dem vollaktuierten Roboter durch einfache mechanische Manipulation ein Pendubot-System [SB95] realisiert werden könnte.

## **A. Anhang**

## A.1. Motordaten

Die folgenden Angaben basieren auf dem Katalog von Faulhaber Antriebssysteme [Fau10].

### DC-Kleinstmotoren

Nr.	Beschreibung	Bez.	3557 024 CS	2642 012 CR
1	Nennspannung	$U_N$	24	12 V
2	Anschlusswiderstand	$R$	5,5	1,45 $\Omega$
3	Abgabeleistung	$P_{2max}$	25,4	22,1 W
4	Wirkungsgrad	$\eta_{max}$	78	78 %
5	Leerlaufdrehzahl	$n_0$	5500	6400 rpm
6	Leerlaufstrom	$I_0$	0,065	0,118 A
7	Anhaltmoment	$M_H$	176	132 mNm
8	Reibungsdrehmoment	$M_R$	2,70	2 mNm
9	Generator-Spannungskonstante	$k_E$	4,3	1,77 mV/rpm
10	Drehmomentkonstante	$k_M$	41,0	16,9 mNm/A
11	Steigung der n-M-Kennlinie	$\Delta n/\Delta M$	31,3	48,5 rpm/mNm
12	Drehmomentkonstante	$L$	850	130 $\mu\text{H}$
13	Drehmomentkonstante	$\tau_m$	16	5,4 ms
14	Rotorträgheitsmoment	$J$	49	11 gcm <sup>2</sup>
15	Winkelbeschleunigung	$\alpha_{max}$	36	120 $\cdot 10^3 \text{rad/s}^2$

Tabelle A.1.: Daten der Gleichstrommotoren

### Planetengetriebe

Nr.	Beschreibung	Bez.	30/1 4,5Nm	26/1 3,5Nm
1	Untersetzungsverhältnis	$\eta$	134:1	14:1
2	Wirkungsgrad	$\eta_{max}$	60	80 %
3	Drehzahl für Dauerbetrieb		4000	4000 rpm
4	Getriebespiel, unbelastet		$\leq 1$	$\leq 1$ °
5	Dauerbetrieb	$M_{max}$	3,5	0,3 Nm
6	Kurzzeitbetrieb	$M_{max}$	4,5	0,4 Nm

Tabelle A.2.: Daten der Planetengetriebe

**Optische Encoder**

Nr.	Beschreibung	Bez.	HEDS 5500 A
1	Impulse pro Umdrehung	N	500
2	Ausgangssignal, rechteckig		2 Ausgänge
3	Betriebsspannung	$V_{CC}$	4,5 ... 5,5 V DC
4	Nennstromaufnahme, Mittelwert	$I_{CC}$	17 mA
5	Trägheitsmoment der Impulsscheibe	J	0,6 $gcm^2$

Tabelle A.3.: Daten der Encoder

## A.2. Technische Dokumentation

### A.2.1. Inbetriebnahme

Für den vollständigen Messaufbau werden folgende Komponenten benötigt:

- Labormodell des Gelenkarm Roboters
- Leistungsverstärker *UPM-24-05*, *PA-103*
- Quanser\_NI *E-Series* Terminal Board
- PC mit installierter *NI PCI 6251 M-Series* Messsignalverarbeitungskarte von National Instruments
- Installierte Softwarepakete: *MATLAB 7.5*, *QuaRC 1.2*

Das folgende Blockschaltbild zeigt den grundsätzlichen Aufbau des Systems.

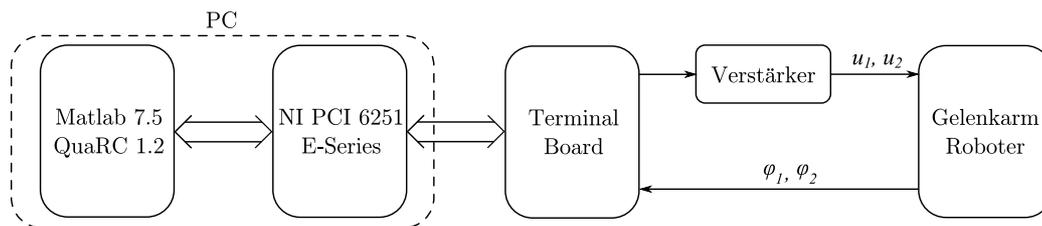


Abbildung A.1.: Messaufbau

Die Spannungsversorgung des Unterarms erfolgt anhand zweier farblich gekennzeichnete Bananenbuchsen an den Leistungsverstärker *PA-103*. Der Gleichstrommotor des Oberarms benötigt den leistungsfähigeren Verstärker *UPM-24-05*, welcher mittels Steckverbindung an den *Load*-Eingang des Verstärkers angeschlossen wird. Im Steckergehäuse befindet sich die Beschaltung eines nichtinvertierenden Verstärkers mit Verstärkungsfaktor  $V = 3$ . Die Steuerleitungen der beiden Verstärker werden durch Cinch-Stecker an das Terminal Board angeschlossen. Dabei belegt der Verstärker des Oberarms den 0. Kanal (*DAC0*) und der Verstärker des Unterarms den 1. Kanal (*DAC1*).

Die beiden Anschlüsse der Inkrementalencoder müssen ebenfalls an das Terminal Board angeschlossen werden. Hierzu wird der Oberarm-Encoder mit dem Anschluss *ENC0* und der Unterarm-Encoder mit dem Anschluss *ENC1* verbunden.

Die Verbindung zwischen dem Quanser Terminal Board und der Messsignalverarbeitungskarte erfolgt anhand eines geschirmten Datenkabels (*Shielded-SHC68-68-EPM*) von National Instruments.

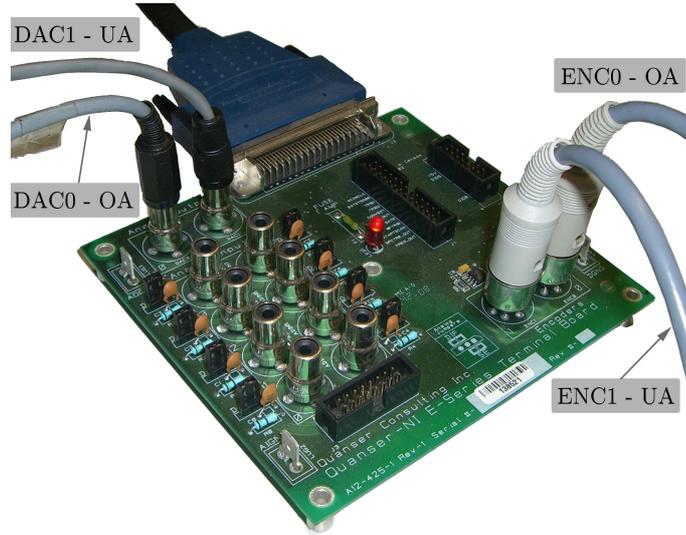


Abbildung A.2.: Terminal Board mit angeschlossener Peripherie

### A.2.2. Quanser Initialisierung

Um die Messdaten der Inkrementalencoder in der Simulation nutzen zu können, müssen diese zuerst aufbereitet werden. Mit Hilfe des Simulink-Blocks *HIL Read Encoder* aus der Quanser-Bibliothek lassen sich die „rohen“ Messdaten in Simulink einlesen. Da dieser Block die Winkel jedoch nicht in Radiant, sondern anhand von Zählimpulsen liefert, muss eine Normierung vorgenommen werden. Beide Datenkanäle der einzelnen Encoder werden erfasst, um eine 4-fach Auswertung zu erreichen. Die Winkelauflösung kann mittels der Strichzahl  $S$ , dem Auswertungsfaktor  $A = 4$  und der Getriebeuntersetzung  $\eta$  wie folgt berechnet werden:

$$\Delta\varphi_i = \frac{2\pi}{S \cdot A \cdot \eta_i} = \frac{2\pi}{cstep_i \cdot \eta_i} \quad \text{für } i = 1, 2 \quad (\text{A.1})$$

Die Strichzahl bzw. die Anzahl der Impulse pro Umdrehung kann aus Tabelle A.3 entnommen werden. Theoretisch müsste sich aufgrund der 4-fach Auswertung eine Auflösung von  $S \cdot A = 2000$  [*Inkrement/Umdrehung*] ergeben. Praktische Messungen zeigten jedoch einen sich fortplanzenden Fehler pro Umdrehung. Aus diesem Grund wurde eine Mittelwertbildung über mehrere Umdrehungen durchgeführt, was zu folgenden Ergebnissen für die Encoder führte:

$$cstep_1 = 1993 \quad cstep_2 = 1970 \quad (\text{A.2})$$

Der tatsächliche Winkel kann mit der Formel

$$\varphi_{i,k} = z_k \cdot \Delta\varphi_i \quad \text{für } i = 1, 2 \quad (\text{A.3})$$

berechnet werden, wobei  $z_k$  den Zählerstand zum Zeitpunkt  $t = kT_d$  repräsentiert.

Als weitere Messgrößen werden die Winkelgeschwindigkeiten der einzelnen Roboterarme benötigt. Da diese messtechnisch nicht erfasst werden, müssen sie anhand der vorhandenen Winkelinformation berechnet werden. Für die Ermittlung von  $\omega = \dot{\varphi}$  kann die Approximation des Differentialquotienten durch den Differenzenquotienten verwendet werden.

$$\dot{\varphi}_i \approx \frac{\varphi_{i,k+1} - \varphi_{i,k}}{T_d} \quad \text{für } i = 1, 2 \quad (\text{A.4})$$

Im  $z$ -Bereich kann obiger Ausdruck mittels der Übertragungsfunktion

$$H(z) = \frac{z - 1}{T \cdot z} \quad (\text{A.5})$$

angegeben werden. Um störende hochfrequente Anteile im Signal zu dämpfen wird dem Differenzierer zusätzlich ein Tiefpass 2. Ordnung nachgeschaltet. Dieser wurde für beide Arme im  $s$ -Bereich mit einer Knickfrequenz von  $\omega_0 = 20Hz$  dimensioniert

$$F(s) = \frac{1}{(1 + \frac{s}{2\pi\omega_0})^2} \quad (\text{A.6})$$

und anschließend in den  $z$ -Bereich transformiert.

Abbildung A.3 zeigt die Einbindung des realen Robotermodells in Simulink. Im linken Teil des Koppelplans werden die Eingangsspannungen  $u_1$  und  $u_2$  der Motoren festgelegt, während im rechten Teil die Zählimpulse der Inkrementalencoder ausgelesen werden. Die Zählerwerte werden anhand Formel (A.3) normiert und liefern dadurch den tatsächlichen Winkel des entsprechenden Roboterarms. Mittels des diskreten Differenzierers und dem nachgeschalteten Tiefpassfilter 2. Ordnung kann die Winkelgeschwindigkeit abgeleitet werden. Im Subsystem *Link1* und *Link2* findet zum Schutz der Hardware eine Begrenzung auf die maximal möglichen Spannungen statt. Weiters wird im Subsystem *Link1* die Spannung  $u_1$  mit dem inversen Verstärkungsfaktor  $V = 1/3$  multipliziert, um dadurch die nachfolgende Verstärkung des nichtinvertierenden Verstärkers zu berücksichtigen.

Bevor die Regelungskonzepte am realen Modell ausprobiert werden, sollte ein kurzer Hardware Funktionstest durchgeführt werden. Hierfür kann die auf der beiliegenden

CD im Ordner *1\_Test des Labormodells* hinterlegte Datei *test\_reales\_modell.mdl* verwendet werden. Mit Hilfe des Simulationsfiles können die Motoren, sowie die korrekte Funktion der Encoder getestet werden.

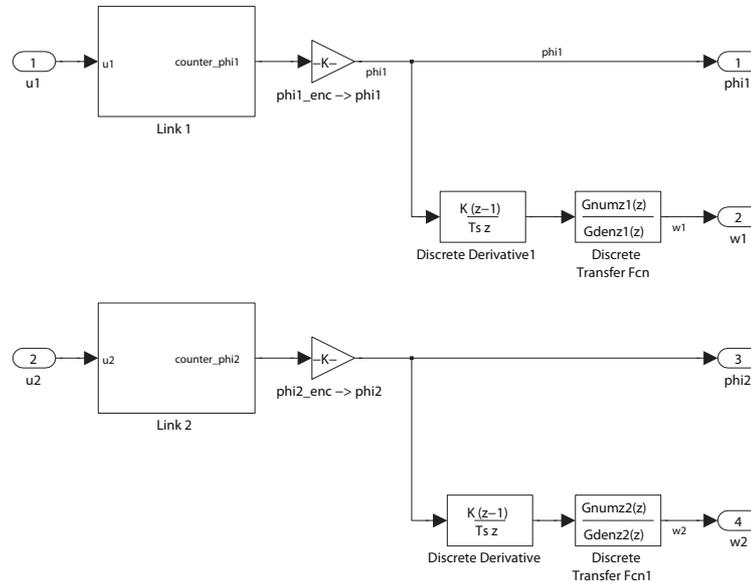


Abbildung A.3.: Anbindung des realen Modells in die Simulation

Bei den verwendeten optischen Encodern des Robotermodells handelt es sich um Inkrementalencoder. Aufgrund dessen kann die Position der Arme nur relativ erfasst werden. Dabei entspricht die Startposition, auf die alle weiteren Positionen inkrementell bezogen werden, immer der aktuellen Lage des Armes nach Ausführung des Programmcodes. Daher muss das Robotermodell zu Beginn der Messung manuell in die Ruhelage ( $q_1 = 0, q_2 = 0$ ) versetzt werden. Um den Oberarm manuell bewegen zu können, muss die Verbindung zum Leistungsverstärker unterbrochen werden. Andernfalls würde die rückinduzierte Spannung des Motors der Bewegung entgegen wirken und folglich zu starker mechanischer Beanspruchung des Modells führen.

Das Robotermodell wurde wie bereits erwähnt für 2 mögliche Lagen realisiert. In den im Folgenden besprochenen Konzepten wird stets von der vertikalen Lage des Manipulators ausgegangen. Diese Annahme spiegelt sich auch in der verwendeten Lösung in der Datei *LagrangeLSGRoboter.m* wieder. Um die horizontale Lage in der Simulation verwenden zu können, muss das Mathematica-Skript für diesen Fall verwendet werden. Es kann allerdings auch durch Manipulation der Erdbeschleunigung  $g$  erreicht werden, dass die Simulation für den horizontalen Fall gilt. Hierzu wird in der Datei *Technische\_Daten.m* die Variable  $g=0$  gesetzt.

### A.2.3. Parameterbestimmung am realen Modell

Für die praktische Bestimmung der Reibungsparameter wurde das Modell in Abbildung A.4 implementiert. Anhand der Eingangssignale  $u_1$  und  $u_2$  können beliebige Spannungen auf das reale Modell geschaltet werden. Um einen direkten Vergleich zwischen Realität und Modell zu erhalten, werden die Signale ebenfalls in das mathematische Modell gespeist. Dadurch kann am Ausgang beider Modelle ein direkter Vergleich durchgeführt werden.

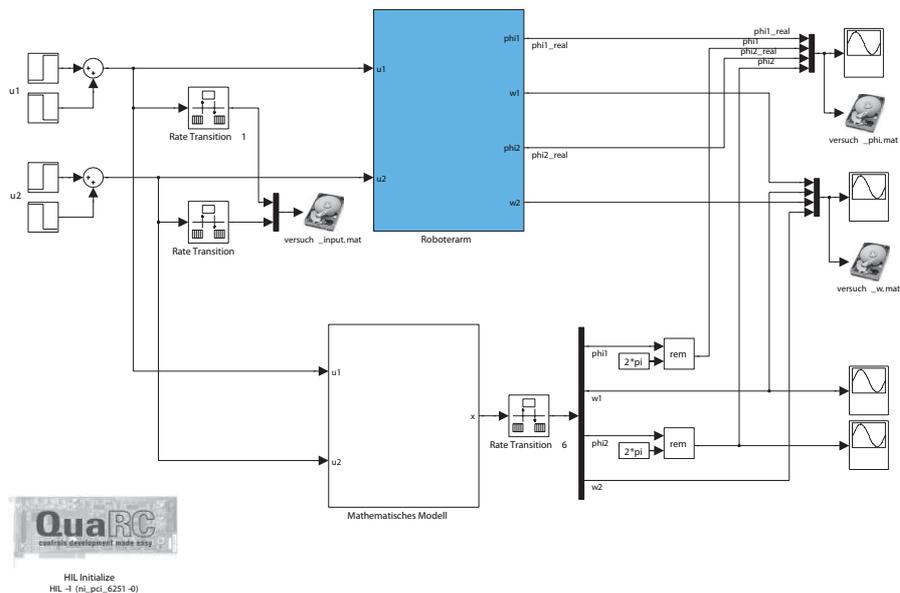


Abbildung A.4.: Quanser-Koppelplan zur Parameterbestimmung

### A.2.4. Implementierung IDC-Konzept im Gelenkraum

Die Realisierung dieses Konzepts hat als Ziel die Punkt-zu-Punkt Bewegung entlang vordefinierter Trajektorien im Gelenkraum. Vom Benutzer werden hierzu beliebige Punkte im Arbeitsraum des Roboters gewählt, welche anschließend mittels der inversen Kinematik automatisch umgerechnet werden. In der Programmstruktur findet eine Trennung zwischen Initialisierung der Trajektorien und Simulation des Regelungskonzepts statt. Zu Beginn werden die Punkte, die vom Roboter angefahren werden sollen, anhand einer grafischen Oberfläche eingegeben. Hierzu wird im Unterordner des Regelungskonzepts das m-File *PathControl.m* verwendet. Die Eingabe kann intuitiv durch Platzierung der Punkte mit der linken Maustaste durchgeführt werden. Die Beendigung der Eingabe erfolgt durch Drücken der rechten Maustaste.

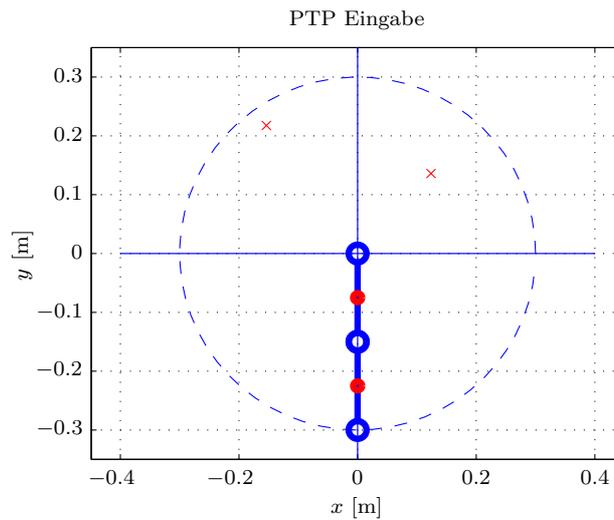


Abbildung A.5.: Grafische Eingabe der Zielpunkte

Mit Hilfe der im Programmcode verwendeten Variablen `dt` kann die Zeitspanne definiert werden, in der die einzelnen Punkte angefahren werden sollen. Dabei muss auf die physikalischen Grenzen des Robotermodells geachtet werden. Weiters kann mit Hilfe der Variablen `t_init` die Zeit zum Anfahren des 1. Punktes gesondert festgelegt werden. Dadurch kann eine kurze Zeitspanne `dt` für Zielpunkte gewählt werden, obwohl sich diese weit entfernt von der Startposition des Roboters befinden. Nach Ausführung des m-Files *PathControl.m* werden die Polynomdaten in die Datei *poly\_data.mat* gespeichert. Die Datei beinhaltet die Polynomkoeffizienten für Position, Geschwindigkeit und Beschleunigung der Trajektorien im Gelenkraum in Matrizenform. Als zusätzlicher Parameter wird die sich aus den Variablen `dt`, `t_init` und der Anzahl der Zielpunkte ergebende Endzeit der einzelnen Trajektorien `tf` gespeichert.

Anhand der generierten Datei *poly\_data.mat* kann die Simulation des Regelungsverfahrens durchgeführt werden. Hierzu wird das Simulationsfile *Regelung\_Quanser.mdl* (siehe Abbildung A.6) geöffnet. Mit Hilfe von Callbacks wird automatisch eine Initialisierung durch Aufruf von *Initialize\_Quanser.m* und *Technische\_Daten.m* durchgeführt. Der Koppelplan wurde größtenteils anhand von *Embedded MATLAB Functions* realisiert. Im Block *Trajektorienplanung* findet die Auswertung der Solltrajektorien durch die Polynomdaten statt. Zur Berechnung wird dem Funktionsblock die aktuelle Simulationszeit übergeben. Der Codeausschnitt A.1 zeigt den implementierten Algorithmus zur Generierung der Solltrajektorien. Mit Hilfe der Variablen `cnt` werden die Polynomkoeffizienten eines Zeitabschnitts selektiert. In den Codezeilen 2 – 20 findet die Auswertung der Polynomdaten mittels des Befehls `polyval` statt. Dieser Befehl berechnet aus den Übergabeparametern den Wert des Polynoms. Die Matrizen `q`, `qp` und `qpp` setzen sich aus 2 übereinander angeordneten Blockstrukturen zusammen, wobei die Obere die Daten des Oberarms und die Untere die Daten des Unterarms beinhaltet.

tet. Nach Beendigung der letzten Trajektorie der zusammengesetzten Bewegung wird mittels der Codezeilen 21 – 28 die letzte Position des Roboters gehalten.

Der Embedded-Block *RobustCompensator* dient der Festlegung der Reglermatrizen  $\mathbf{K}_P$ ,  $\mathbf{K}_D$  und der Implementierung des Regelgesetzes in der Form:

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}^d - \mathbf{q}) \quad (\text{A.7})$$

Zur Festlegung der Matrizen werden die im Programmcode befindlichen Variablen  $\omega_0$  für den Oberarm und  $\omega_1$  für den Unterarm gewählt.

Listing A.1: Trajektorienplanung

```

1  ...
2  if t <= tf(cnt)
3      % Evaluate Trajektorie for Position, Velocity and Acceleration
4      qd1 = polyval(q(cnt,:), t);
5      qd2 = polyval(q(cnt+length(q(:,1))/2,:), t);
6      qdp1 = polyval(qp(cnt,:), t);
7      qdp2 = polyval(qp(cnt+length(qp(:,1))/2,:), t);
8      qdpp1 = polyval(qpp(cnt,:), t);
9      qdpp2 = polyval(qpp(cnt+length(qpp(:,1))/2,:), t);
10 else
11     if t <= tf(end)
12         % Evaluate Trajektorie for Position, Velocity and Acceleration
13         cnt=cnt+1;
14
15         qd1 = polyval(q(cnt,:), t);
16         qd2 = polyval(q(cnt+length(q(:,1))/2,:), t);
17         qdp1 = polyval(qp(cnt,:), t);
18         qdp2 = polyval(qp(cnt+length(qp(:,1))/2,:), t);
19         qdpp1 = polyval(qpp(cnt,:), t);
20         qdpp2 = polyval(qpp(cnt+length(qpp(:,1))/2,:), t);
21     else
22         % Hold Final Position
23         qd1 = polyval(q(cnt,:), tf(end));
24         qd2 = polyval(q(cnt+length(q(:,1))/2,:), tf(end));
25         qdp1 = 0;
26         qdp2 = 0;
27         qdpp1 = 0;
28         qdpp2 = 0;
29     end
30 end
31 ...

```

Der nichtlineare Rückkopplungsterm zur globalen Linearisierung der nichtlinearen Strecke befindet sich im Block *NonlinearControl*. Wie bereits in Abschnitt 7.2.1 erläutert, setzt sich der Term zur Linearisierung wie folgt zusammen:

$$\mathbf{u} = \bar{\mathbf{M}}(\mathbf{q})\mathbf{v} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \quad (\text{A.8})$$

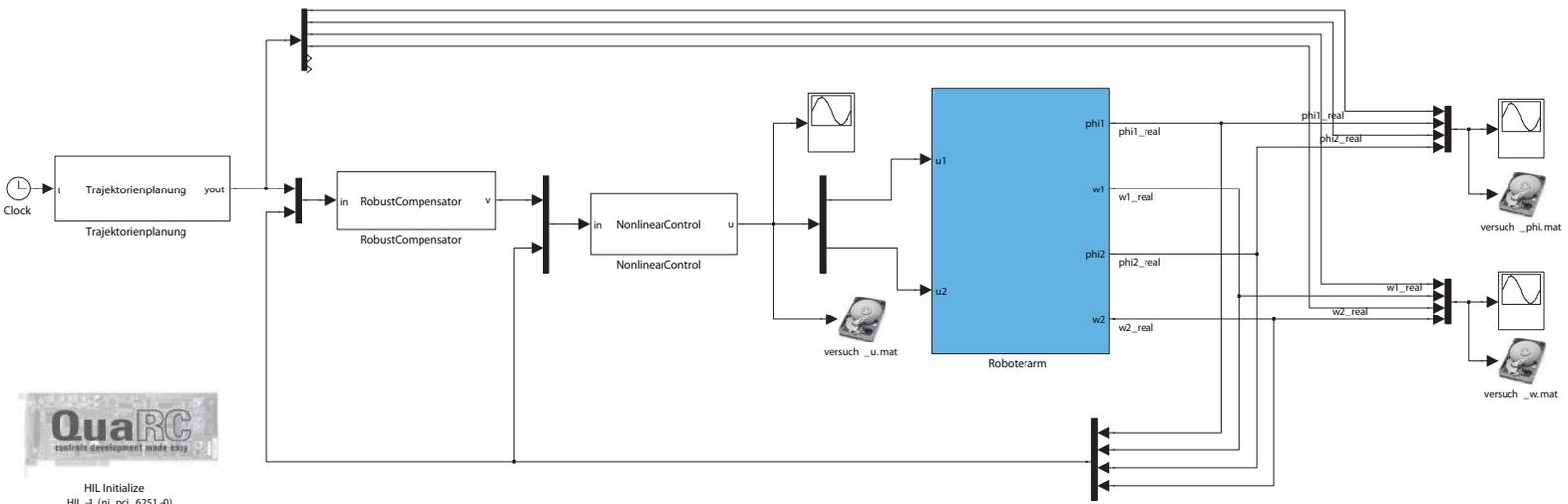


Abbildung A.6.: Quanser-Koppelplan zur Simulation der PTP-Bewegung

Die Ausdrücke für  $\bar{\mathbf{M}}$  und  $\mathbf{h}$  werden aus dem im Unterordner befindlichen Mathematica-Skript in die Datei *FeedbackLinearization* exportiert und können direkt im Block *Nonlinear Control* implementiert werden.

### A.2.5. Implementierung IDC-Konzept im Arbeitsraum

Dieses Regelungsverfahren ermöglicht es dem Manipulator, geometrischen Figuren folgen zu können. Um dies zu gewährleisten, kann die Trajektorienplanung nicht mehr im Gelenkraum anhand von zeitlichen Referenzgrößen vorgenommen werden, sondern muss im Arbeitsraum definiert werden. Zur Durchführung der Simulation des Konzepts wird die Datei *Regelung\_Quanser.mdl* im entsprechenden Unterordner geöffnet. Die Initialisierung der Komponenten wird wiederum automatisch aus dem Simulationsfile mittels Callbacks durchgeführt. Der Benutzer kann daher direkt aus dem Simulink-Koppelplan (siehe Abbildung A.7) nach Übersetzung des Programmcodes die Simulation starten. Aus dem Koppelplan ist ersichtlich, dass die grundsätzliche Struktur bestehend aus 2 verschachtelten Schleifen gleich geblieben ist. Der Block *NonlinearControl* ist daher vollkommen identisch zu dem im vorherigen Abschnitt behandelten Konzept. Allerdings lässt sich erkennen, dass das Regelgesetz nun im Arbeitsraum festgelegt wird.

Der Embedded-Block *Trajektorienplanung* weist 2 Eingänge auf, welche sich aus der Simulationszeit  $t$  und einer vom Benutzer wählbaren Variable **State** ergeben. Mit dieser Variablen kann der Benutzer die gewünschte geometrische Figur festlegen. Dabei wurde folgende Konvention getroffen:

- **State=1:**  
Geometrische Figur, die abgefahren wird, entspricht einem Kreis.
- **State=2:**  
Geometrische Figur entspricht einer Ellipse.
- **State=3:**  
Geometrische Figur entspricht einer Geraden, die periodisch oder einmalig abgefahren wird.
- **Default:**  
Geometrische Figur entspricht einer zusammengesetzten Bewegung. Standardmäßig wird das Kürzel des Instituts für Regelungstechnik *IRT* vom Roboter abgefahren.

Im Programmcode bezeichnet `init_time` die Zeit die dem Manipulator zur Verfügung steht, um aus der Ruhelage ( $q_1 = 0, q_2 = 0$ ) in die Startposition der Figur  $pstart(x, y)$  zu gelangen. Weiters wurde im Code vorgesehen, dass vor Beginn eines geometrischen Pfads eine Pause von  $td = 0.5ms$  eingehalten wird. Durch diese Maßnahme können auftretende mechanische Schwingungen reduziert werden. Der Programmausschnitt in A.2

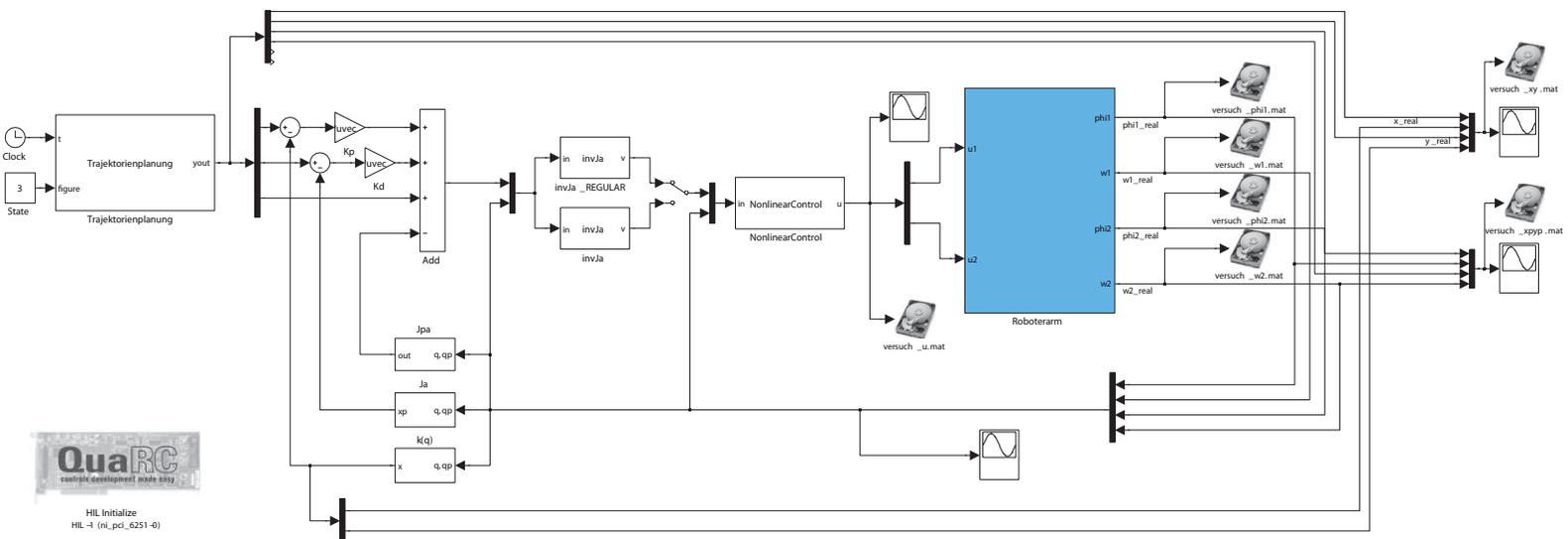


Abbildung A.7.: Quanser-Koppelplan zur Simulation von geometrischen Figuren

zeigt den relevanten Teil im Fall, dass ein Kreis abgefahren werden soll. Der Benutzer kann im gekennzeichneten Bereich (Zeile 5 – 9) die Parameter des Kreises festlegen. Zur Definition der Bewegung stehen dem Benutzer spezielle *Motion-Bausteine* zur Verfügung, welche mittels Matlab-Funktionen implementiert wurden.

- `motion=circle(xm,ym,r,c,t-init_time-td)`
- `motion=ellipse(xm,ym,a,b,c,t-init_time-td)`
- `motion=straight_line(pstart,pend,dt,t-init_time-td)`
- `motion=periodic_line(pstart,pend,dt,t-init_time-td)`
- `motion=periodic_line_delay(pstart,pend,dt,td,t-init_time-td)`
- `motion=irt(pstart,dt,td,t-init_time-td)`

Der Rückgabeparameter `motion` bezeichnet jeweils einen Vektor, welcher sich aus Position, Geschwindigkeit und Beschleunigung zusammensetzt.

Listing A.2: Trajektorienplanung

```

1  ...
2  if figure==1
3
4      %Parameter CIRCLE*****
5      xm=-0.15;
6      ym=0.15;
7      rmax=0.05;
8      r=rmax;    % Schnecke: r=rmax-cnt*1e-5;
9      c=7;      % 2*pi/c Sekunden für 1 Umdrehung
10     %*****
11
12     pstart=[xm+r ym];
13     motion=circle(xm,ym,r,c,t-init_time-td);
14 else
15 ...

```

Die Realisierung des Regelgesetzes erfolgt anhand der Formulierung

$$\mathbf{v} = \mathbf{J}_A^{-1} \cdot \left[ \ddot{\mathbf{x}}^d + \mathbf{K}_D(\dot{\mathbf{x}}^d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}^d - \mathbf{x}) - \dot{\mathbf{J}}_A \dot{\mathbf{q}} \right]. \quad (\text{A.9})$$

Die für die Transformation vom Gelenkraum in den Arbeitsraum benötigten Beziehungen wurden anhand von Funktionsblöcken implementiert. Dadurch konnten die Blöcke ohne weitere Bearbeitung aus dem Simulink-Koppelpan in Abbildung 7.10 übernommen werden. Im mittleren Teil des Quanser-Koppelpans ist ein manueller Schalter erkennbar, mit dessen Hilfe der Benutzer die Regularisierung der Jacobi-Matrix  $\mathbf{J}_A$  ein- oder ausschalten kann. Dadurch können die Auswirkungen dieser Maßnahme direkt am Modell betrachtet werden.

### A.2.6. Implementierung Robust Control

Für die Realisierung dieses Konzepts kann der Quanser-Koppelplan aus Abbildung A.6 herangezogen werden. Die Struktur ist dabei identisch, allerdings findet im Block *RobustCompensator* eine Erweiterung des Codes statt.

Listing A.3: Robust Control

```

1 ...
2 Qm=60;           % maximum acceleration
3 alpha=0.1;      % 0<alpha<1 defines the congruence between the estimate and the real inertia matrix
4 M0=230;        % upper bound of ||Hat_M^(-1)||
5 Phi=0.1;       % upper bound of ||Delta_h||
6 epsilon=0.01;
7
8 if isempty(Q)
9     Q=calcRobustCompensator(Kp,Kd);
10 end
11
12 D=zeros(2,2); eye(2);
13 K=[Kp Kd];
14 zeta=[qd-q; qdp-qp];
15
16 rho=1/(1-alpha)*(alpha*Qm+alpha*norm(K)*norm(zeta)+M0*Phi);
17
18 if norm(D'*Q*zeta) >= epsilon
19     dv=rho*D'*Q*zeta/norm(D'*Q*zeta);
20 else
21     dv=rho/epsilon*D'*Q*zeta;
22 end
23 ...

```

In den Zeilen 2 – 6 werden die Konstanten zur Erfüllung der Annahmen in (7.59) - (7.61) gewählt. Anschließend wird in der Funktion *calcRobustCompensator* mit Hilfe der Wahl einer Einheitsmatrix für  $\mathbf{P}$  die positiv definite Matrix  $\mathbf{Q}$  bestimmt. Die restlichen Codezeilen repräsentieren die Implementierung des besprochenen Konzepts Robust Control. Nach der Berechnung von  $\Delta \mathbf{v}$  kann schließlich das Regelungskonzept in der Form

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}^d - \mathbf{q}) + \Delta \mathbf{v} \quad (\text{A.10})$$

implementiert werden.

## Literaturverzeichnis

- [ABRW09] ANGERMANN, A. ; BEUSCHEL, M. ; RAU, M. ; WOHLFARTH, U.: *MATLAB-Simulink-Stateflow*. Oldenbourg, 2009
- [Ada09] ADAMY, J.: *Nichtlineare Regelungen*. Springer, 2009
- [AH91] ARMSTRONG-HÉLOUVRY, B.: *Control of Machines with Friction*. Springer, 1991
- [Bre88] BREMER, H.: *Dynamik und Regelung mechanischer Systeme*. Teubner, 1988
- [CCS96] CACCAVALE, F. ; CHIAVERINI, S. ; SICILIANO, B.: Singularity-Robust second-order kinematic control of robot manipulators. (1996)
- [Chi93] CHIAVERINI, S.: Estimate of the Two Smallest Singular Values of the Jacobian Matrix: Application to Damped Least-Squares Inverse Kinematics. In: *Journal of Robotic Systems* (1993), S. 991–1008
- [Cra05] CRAIG, J.J.: *Introduction to Robotics*. Pearson Studium, 2005
- [Fau10] FAULHABER: *Katalog Faulhaber Antriebssysteme 2010-2011*, 2010
- [Gfr10] GFRERRER, A.: *Kinematik und Robotik*. TU Graz, Vorlesungsskriptum, 2010
- [Gut79] GUTMAN, S.: Uncertain Dynamical Systems - A Lyapunov Min-Max Approach. In: *IEEE Trans. Aut. Control* (1979), S. 437–443
- [HD04] HORN, M. ; DOURDOUMAS, N.: *Regelungstechnik*. Pearson Studium, 2004
- [Int] INTERNET: *Table of Laplace and Z Transforms and their Properties*. <http://lpsa.swarthmore.edu/LaplaceZTable/LaplaceZFuncTable.html>, Ab-ruf: 28.09.2011
- [LT10] LE-TIEN, L.: *Ansätze zur entkoppelten Regelung von mechanisch gekoppelten Doppelgelenken eines DLR-Medizinroboters*, Technische Universität Dresden, Diss., 2010
- [MWF03] MÜTHER, H. ; WHARAM, D. ; FÄSSLER, A.: *Lagrange Formalismus*. 2003
- [Nat08] NATIONAL INSTRUMENTS: *High-Speed M Series Multifunction Data Acquisition*, 2008

- [Qua09] QUANSER: *QuaRC 1.2 Installation Guide*, 2009
- [SB95] SPONG, M. ; BLOCK, D.: The Pendubot: A Mechatronic System for Control Research and Education. In: *IEEE CDC* (1995)
- [SS03] SCIAVICCO, L. ; SICILIANO, B.: *Modelling and Control of Robot Manipulators*. Springer, 2003
- [Sta09] STARK, G.: *Robotik mit MATLAB*. Hanser, 2009
- [SV89] SPONG, M. W. ; VIDYASAGAR, M.: *Robot Dynamics and Control*. John Wiley & Sons, 1989
- [Wai07] WAIBOER, R. R.: *Dynamic Modelling, Identification and Simulation of Industrial Robots*, Netherlands Institute for Metals Research, Diss., 2007. [http://doc.utwente.nl/57838/1/thesis\\_waiboer.pdf](http://doc.utwente.nl/57838/1/thesis_waiboer.pdf), Abruf: 28.09.2011