

# hochschule mannheim

Hochschule für Technik und Gestaltung Mannheim

Fachbereich Informatik  
Institut für Robotik

## Diplomarbeit

Sensorgesteuerte Bewegung und  
Bewegungsplanung für ein Roboterbein

**eingereicht von:** Rafael Troilo

**eingereicht am:** 16. April 2007

**Betreuer:** Herr Prof. Dr. Thomas Ihme

**Zweitkorrektor:** Frau Prof. Dr. Miriam Föller

**erstellt bei:** Hochschule Mannheim  
Windeckstraße 116  
68309 Mannheim

# **Eidesstattliche Erklärung**

Hiermit erkläre ich, dass ich die an der Fachhochschule Mannheim vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Sämtliche Stellen der Arbeit, die benutzten Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellenangaben kenntlich gemacht.

Mannheim, den 16. April 2007.

---

Rafael Troilo

# Kurzfassung

Die vorliegende Arbeit hat das Ziel, die Regelung der Beinantriebe eines Schreitroboters zu verbessern und um eine Kraftregelung zu erweitern.

Schreiten durch ein unebenes Gelände ohne detaillierte Kenntnisse der Umgebung, erkennen und überwinden von Hindernissen und das Ausführen von technischen Operationen, wie Testen der Bodeneigenschaften, Manipulation von Objekten, Bohroperationen und Montageoperationen, sind Anforderungen an heutige mehrbeinige Schreitroboter.

Aus diesen Anforderungen heraus müssen sich die Beine eines Laufroboters präzise und schnell positionieren lassen und bei Krafteinwirkungen schnell reagieren. Eine Analyse der vorhandenen Regelung des Lauron IVb brachte hervor, dass ein einschleifiges Regelungsprinzip, aufbauend auf einem Standardwinkelgeber an den Gelenken, auf Grund der groben Auflösung des Gebers und der nichtlinearen lastabhängigen Regelstrecke, der Anforderung einer präzisen Positionierung nicht nachkommt. Durch eine Untersuchung des Regelungssystems und der Erfassung weiterer Sensorinformationen, wie der Drehgeschwindigkeit, konnte ein auf der Kaskadenstruktur basierender Regelalgorithmus entworfen und implementiert werden, mit dem sich diese Anforderung erfüllen lässt. Um schnell auf Krafteinwirkungen reagieren zu können wird gezeigt wie mit Hilfe eines Mikrocontrollers eine Nachgiebigkeitsregelung, sowie eine Dämpfungsregelung realisiert werden kann.

Als Ergebnis dieser Arbeit entstand ein neues Regelungskonzept der Beinantriebe, sowie eine Nachgiebigkeits- und Dämpfungsregelung, implementiert mit Hilfe des Mikrocontrollers.

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>ii</b>
<b>Kurzfassung</b>	<b>iii</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Motivation und Zielsetzung . . . . .	1
1.2 Übersicht . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Laufroboter LAURON IVb . . . . .	4
2.1.1 Aufbau des Laufroboters Lauren IVb . . . . .	4
2.1.2 Koordinatensysteme . . . . .	6
2.1.3 Laufphasen des Schreitzyklus . . . . .	7
2.2 Aufbau eines Roboterbeines . . . . .	8
2.2.1 Kinematischer Aufbau . . . . .	8
2.2.2 Arbeitsbereich . . . . .	9
2.2.3 Sensoren . . . . .	9
2.2.4 Aktoren . . . . .	12
2.2.5 Steuerungs Aufbau . . . . .	12
2.3 Modular Controller Architecture . . . . .	15
2.3.1 MCA-Steuerung des Lauren IVb . . . . .	16
2.4 Grundlagen der Regelungstechnik . . . . .	17
2.4.1 Regelungssystem . . . . .	18
2.4.2 Übertragungsglieder . . . . .	19
2.4.3 PI Regler . . . . .	20
2.4.4 Regelkonzepte . . . . .	21
2.5 Grundlagen der Kinematik . . . . .	23
2.5.1 Direkte Kinematik . . . . .	25
2.5.2 Inverse Kinematik . . . . .	26
<b>3 Neue Regelungskonzepte für Beinantriebe und Kraft</b>	<b>30</b>
3.1 Regelung der Beinantriebe . . . . .	31
3.1.1 Untersuchung am Teilsystem . . . . .	31
3.1.2 Gewinnung zusätzlicher Sensorinformationen . . . . .	37

## Inhaltsverzeichnis

3.1.3	Entwurf des Regelungskonzeptes . . . . .	40
3.1.4	Digitaler PI-Stellungsalgorithmus . . . . .	48
3.2	Kraftregelung der Beine . . . . .	50
3.2.1	Kraftgeführte Bewegung . . . . .	50
3.2.2	Kraftransformation . . . . .	52
3.2.3	Inverse Kinematik . . . . .	53
3.2.4	Einordnung in die Regelstruktur . . . . .	56
<b>4</b>	<b>Implementierung und Integration</b>	<b>57</b>
4.1	Analyse der vorhanden Software des DSP . . . . .	57
4.1.1	Bootloader . . . . .	57
4.1.2	Steuerungsprogramm . . . . .	58
4.2	Implementation Beinregelung . . . . .	59
4.2.1	Erfassen der Geschwindigkeit . . . . .	59
4.2.2	Implementation der Drehzahlregelung . . . . .	60
4.2.3	Implementation der Lageregelung . . . . .	60
4.3	Implementation der Kraftregelung . . . . .	60
4.3.1	Erfassung der Kraftdaten . . . . .	61
4.3.2	Trigonometrische Berechnung . . . . .	62
4.3.3	Berechnungszeit . . . . .	64
<b>5</b>	<b>Ergebnisse</b>	<b>65</b>
5.1	Erweiterungen der DSP-Software . . . . .	65
5.2	Messergebnisse . . . . .	68
5.2.1	Drehzahlregelung des ausgebauten Motors . . . . .	68
5.2.2	Verhalten der Beta-Gelenk Regelstrecke . . . . .	69
5.2.3	Regelung der Beta-Gelenk Regelstrecke . . . . .	71
5.2.4	Kraftwerte . . . . .	74
<b>6</b>	<b>Zusammenfassung</b>	<b>75</b>
	<b>Abbildungsverzeichnis</b>	<b>77</b>
	<b>Literaturverzeichnis</b>	<b>80</b>

# 1 Einführung

Mit Hilfe der natürlichen Fortbewegung, dem Laufen, ist es für Menschen und Tiere möglich, sich frei in ihrer Umwelt zu bewegen. Seit einigen Jahrhunderten arbeiten Ingenieure und Wissenschaftler daran Maschinen das Laufen beizubringen.

Das Laufen bietet besondere Vorteile, denn anders als ein Rad benötigt ein Bein keinen ständigen Bodenkontakt. Dies ermöglicht das Manövrieren auch in zerklüftetem Gelände, wo für rollende Fahrzeuge bereits kein Durchqueren mehr möglich wäre. Die Aufgabenbereiche für Laufroboter sind z.B. das Erkunden von unwegsamen Gelände, das Arbeiten in für Menschen feindlichen Umgebungen oder das Aufsuchen von Minen in Krisengebieten.

## 1.1 Motivation und Zielsetzung

Zur Erforschung von Laufbewegungen wurde im Forschungszentrum für Informatik an der Universität Karlsruhe das Projekt LAURON gestartet. Das LAURON Modell ist ein sechsbeiniger Laufroboter, welcher sich an der Geometrie der indischen Stabheuschrecke orientiert.

Zur Erforschung des Verhaltens von Laufmaschinen in unstrukturierten Umgebungen und deren Einsatz im Bereich technischer Anwendungen, wurde für das Institut für Robotik an der Hochschule Mannheim der Prototyp der sechsbeinigen Laufmaschine *LAURON IVb* aus Karlsruhe angeschafft.

Die Fortbewegung eines Schreitroboters in unstrukturierter Umgebung, speziell in für fahrende Roboter ungeeignetem Gelände, erfordert oft komplexe Laufbewegungen. Je geeigneter eine Laufbewegung für ein Gelände ist, desto sicherer und oft auch schneller vermag sich der Schreitroboter darüber zu bewegen. Laufbewegungen sind das Resultat der definierten Bewegung der Gesamtheit aller Beine des Roboters. Dafür muss sich jedes einzelne Bein so präzise wie möglich regeln lassen, d.h. die Vorgaben der Steuerinstanz möglichst optimal erreichen. Dies zu gewährleisten, ist die Aufgabe eines Beinreglers. Zudem treten beim Schreiten durch eine rauhe Umgebung oft unerwartete Kräfte, wie sie zum Beispiel bei dem Auftreffen auf Hindernissen mit einer Geschwindigkeit deutlich größer als Null entstehen, auf. Diese Kräfte müssen ebenfalls durch ein Regelungssystem abgefangen werden.

Beobachtungen der Beinregler des *LAURON IVb* Modells zeigten das Positionsvorgaben

teilweise nicht vollständig ausgeregelt werden, so dass eine Differenz zwischen der Vorgabe position und der aktuellen Position verbleibt. Diese Abweichung wird umso größer je mehr Belastungen das Bein ausgesetzt ist. Doch gerade bei Bohroperationen treten hohe Belastungen der Beine auf, dennoch wird gerade dort eine präzise Positionierung der Beine verlangt. Zur Ausregelung auftretender Kräfte an den Beinen wurde bereits in einer Vorarbeit von Kai Wetzelsberger [31] ein Regelungssystem entworfen und implementiert. Bedingt durch die Einordnung der Kraftregelung in die oberste Steuerungsinstanz, treten große Signallaufzeiten zwischen Erfassung der Kräfte am Bein bis hin zu ihrer Ausregelung auf.

Das Ziel dieser Arbeit besteht nun darin die vorhandene Regelung unter zu Hilfenahme zusätzlichen Sensorinformationen zu verbessern, so dass sich auch unter schwankender Belastung der Beine genaue Positionsangaben ausregeln lassen. Zudem soll gezeigt werden, ob eine Implementierung der Kraftregelung in den Beinreglern möglich ist, da diese die Signallaufzeiten auf ein Minimum verkürzen würden und somit die Reaktionszeit der Kraftregelung verbessern würde.

## 1.2 Übersicht

Die vorliegende Arbeit gliedert sich insgesamt in sechs Kapitel. In der Einführung wurde auf die Motivation und Zielsetzung dieser Arbeit eingegangen.

Im darauf folgenden Kapitel 2 werden die Grundlagen, auf welche diese Arbeit aufbaut, vorgestellt. In 2.1 wird ein Überblick des Laufroboter Modells *LAURON IVb* gegeben und anschließend detaillierter auf den Aufbau und Steuerung eines einzelnen Beines eingegangen. Des Weiteren werden Grundlagen zur Regelungstechnik sowie zur Bewegungslehre beschrieben.

In Kapitel 3 werden die erarbeiteten allgemeinen Lösungskonzepte für die Antriebsregelung sowie für die Kraftregelung gezeigt. Zunächst wird auf Regelung der Beinantriebe eingegangen. Hierzu wird das vorhandene System untersucht, beschrieben wie sich zusätzliche Sensorinformationen für die Regelung gewinnen lassen, der Entwurf eines neuen Regelungskonzeptes vorgestellt und auf den digitalen Regelungsalgorithmus eingegangen. Im zweiten Schritt wird die Lösung zur Kraftregelung präsentiert. Hierbei werden zwei Methoden der kraftgeführten Bewegung gezeigt, wie sich die von Kraftsensor gelieferten Informationen nutzen lassen, wie man durch die, von der Kraftregelung hervorgebrachten, neuen Positionsangaben im kartesischen Raum wieder auf die einzelnen Gelenkwinkel schließen kann und zuletzt wie sich die Kraftregelung in die neu entworfene Regelstruktur einordnen lässt.

Das nachfolgende Kapitel Implementierung und Integration befasst sich damit, wie sich die neuen Konzepte in die vorhandene Software integrieren lassen und wie Probleme bei der Implementierung umgangen wurden.

## *1 Einführung*

Unter dem Stichpunkt Ergebnisse werden zunächst die Erweiterungen der vorhandenen Software vorgestellt. Es werden neu hinzugekommene Funktionen gezeigt und kurz in ihren Funktionsweisen beschrieben. Anschließend werden in 5.2 konkrete Messergebnisse, welche mit dem neuen System aufgenommen wurden, präsentiert und ihre Aussagen diskutiert.

Das letzten Kapitel Zusammenfassung trägt nochmals alle in dieser Arbeit gewonnen Ergebnisse zusammen und gibt einen Ausblick auf mögliche Erweiterungen und noch offene Fragen.

## 2 Grundlagen

### 2.1 Laufroboter LAURON IVb

Schreiten durch ein unebenes Gelände ohne detaillierte Kenntnisse der Umgebung, erkennen und überwinden von Hindernissen und das Ausführen von technischen Operationen, wie Testen der Bodeneigenschaften, Manipulation von Objekten, Bohroperationen und Montageoperationen, sind Anforderungen an heutige mehrbeinige Schreitroboter.

Um diese Anforderungen umzusetzen wurde Anfang der 90er Jahre das Laufroboter Projekt Lauron (Laufender Roboter Neuronal gesteuert) am Forschungszentrum für Informatik (FZI) in Karlsruhe[15] ins Leben gerufen. Ziel dieses Projektes ist die Entwicklung und praktische Erprobung von Laufsteuerungen. Dabei wurden verschiedene Arten von Neuronalen Netzen untersucht. Die erste Generation des Lauron Projektes wurde 1994 auf der CeBit vorgestellt. Untersuchungen zum ersten Modell Lauron I (Abb.2.1(a)) haben jedoch Schwachstellen im Bezug auf die Sensorausstattung und Mechanik aufgedeckt, die bei der Entwicklung des Nachfolgers Lauron II behoben wurden. Im Laufe der Entwicklung des Lauron Projektes wurden weitere Veränderungen unternommen, so wurden Kraftsensoren an den Beinen montiert und das Steuerungsprinzip, welches ursprünglich auf Neuronale Netze beruhte, durch die hierarchisches MCA<sup>1</sup> Architektur[28] ersetzt. Das neuste Modell des Lauron ist der Lauron IVc (Abb.2.1(b)), welcher 2004 entwickelt wurde. Diese Diplomarbeit baut auf dem *LAURON IVb* Modell auf. Alle nachfolgenden Beschreibungen des Schreitroboters beziehen sich daher auf dieses Modell.

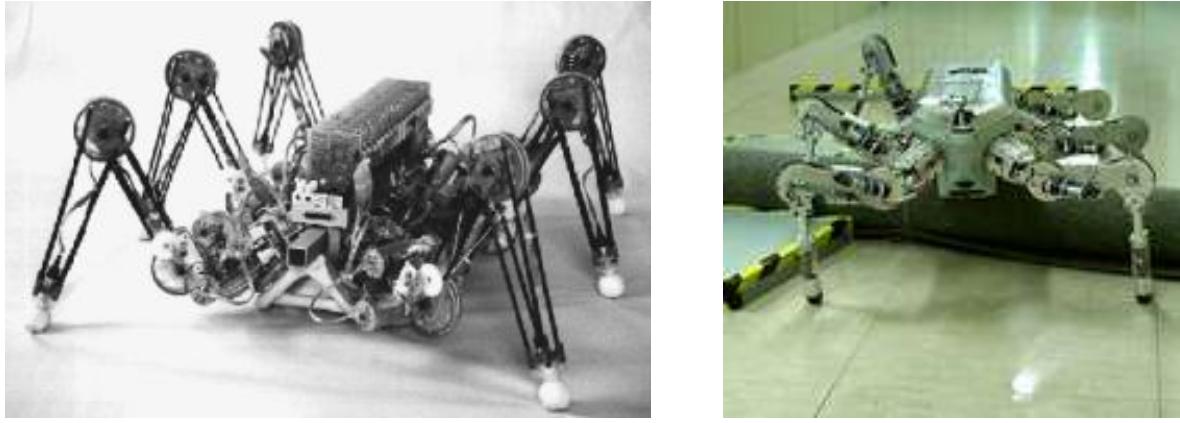
#### 2.1.1 Aufbau des Laufroboters Lauron IVb

Die Grundform des *LAURON IVb* wurde der Natur nachempfunden. Als Vorbild diente hier die indische Stabheuschrecke (*Carausius morosus*) sowohl für den mechanischen Aufbau als auch für die Bewegungskoordination.

Der Körper bildet das Grundgerüst des *LAURON IVb*. An seinen äußeren Seitenflächen sind, wie bei der Stabheuschrecke, sechs Baugleiche Beine befestigt. Wobei die vorderen und hinteren Beine zur Längsachse des Körpers um einen Winkel von 30° angebracht sind, während sich die mittleren Beine parallel zur Längsachse befinden. Zusätzlich sind,

---

<sup>1</sup>Modular Controller Architecture



(a) Lauron I [13]

(b) Lauron IVc [14]

Abbildung 2.1: Lauron Modelle

ebenfalls wie bei der Stabheuschrecke, die mittleren Beine um  $25^\circ$  nach unten Richtung Boden geneigt, während die restlichen Beine um jeweils  $30^\circ$  geneigt sind. Der Schreitroboter kann sowohl in Bezug auf das Symmetrikoordinatensystem (SKS) (siehe 2.1.2.1) in positive x-Richtung(vorwärts) wie auch in negative x-Richtung(rückwärts) gleich schnell Laufen. Zusätzlich beherrscht der Laufroboter noch einen Seitwärtsgang, welcher allerdings bedingt durch die Bauart langsamer ist als der Gang in Vorzugsrichtung vorwärts bzw. rückwärts. Am vorderen Ende der Roboters ist ein Kamerakopfsystem, bestehend aus zwei Kameras, angebracht, welches sich unabhängig von Körper drehen lässt.

Im inneren des Körpers ist das Steuerungssystem untergebracht. Dies besteht aus sieben baugleichen Mikrocontrollern für die Beinregelung und Kopfdrehung und einem Standard Industrie PC/104[3] der Firma Lippert[23].

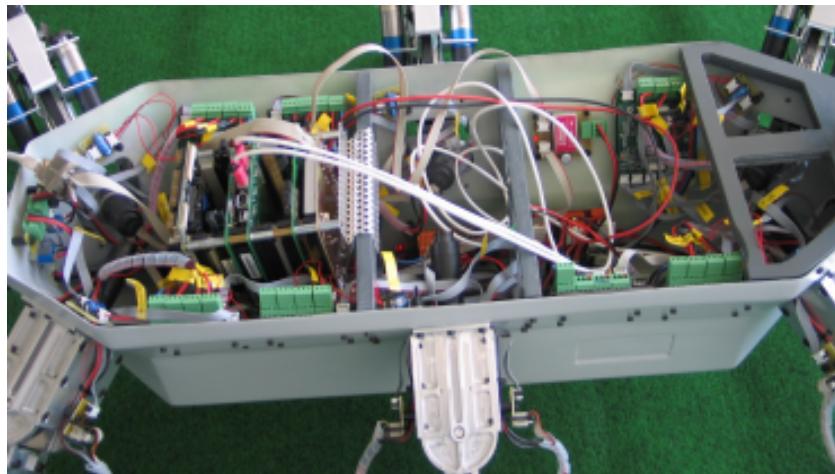


Abbildung 2.2: Innenaufbau des Körpers

## 2.1.2 Koordinatensysteme

Zur Bewegungssteuerung werden verschiedene Koordinatensysteme als Bezugssysteme verwendet. Auf diese beziehen sich die Koordination der Bewegung (Kinematik) und die Sensorinformationen. Bei Lauron IVb werden kartesische Koordinatensysteme verwendet, somit stehen die Koordinatenachsen paarweise senkrecht aufeinander. Diese Koordinatensysteme stehen alle in klar definiertem Bezug zueinander.

### 2.1.2.1 Definition der Koordinatensysteme

Im Folgenden werden die Koordinatensysteme genauer erläutert, da diese für die Koordination der Bewegung und die Darstellung der Sensorinformation essentiell sind. In Abbildung 2.3 ist eine schematische Zeichnung des Zentralkörpers mit eingezeichneten Koordinatensystemen zu sehen.

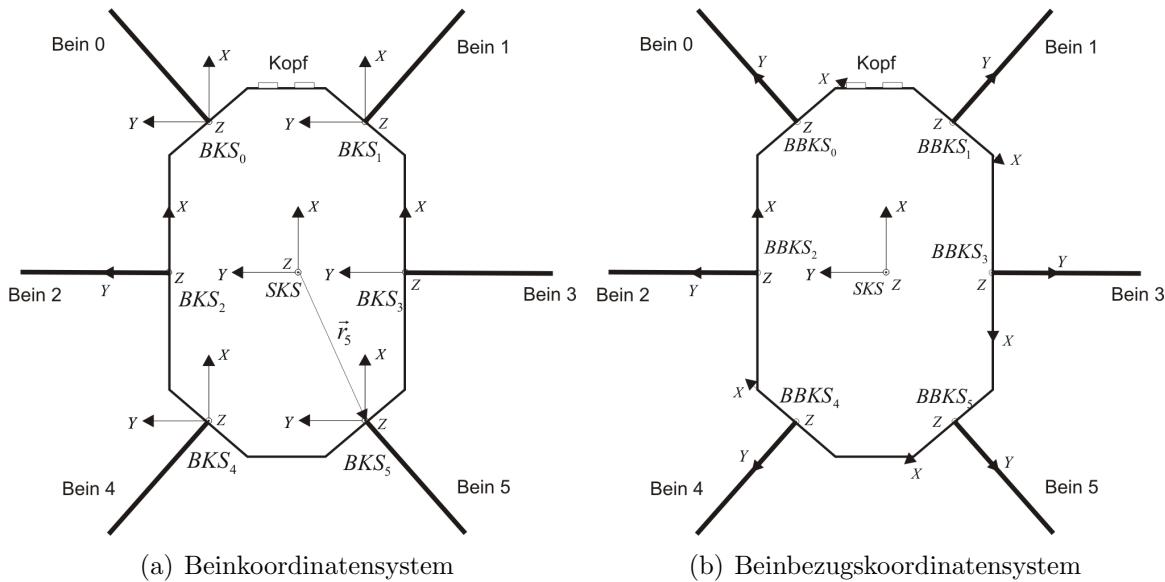


Abbildung 2.3: Koordinatensysteme Lauron IVb mit Ansicht von oben aus [31]

### Das Beinkoordinatensystem (BKS) $\{B^i; X, Y, Z\}$

Alle Beine des Roboters werden mit Indizes versehen. Dabei werden die Beine von vorne links ( $B^0$ ) nach hinten rechts ( $B^5$ ) durchnummieriert. Der Koordinatenursprung ( $B^i$ ) liegt im Beinfestigungspunkt am Zentralkörper. Die  $x$ -Achse zeigt entlang der Zentralkörperlängsachse nach vorne in die Hauptbewegungsrichtung bzw. dem Kopf, die  $z$ -Achse zeigt in Bezug auf den Roboterkörper senkrecht nach oben und die  $y$ -Achse bildet das Rechtssystem (Abbildung 2.3(a)).

### **Das Beinbezugskoordinatensystem (BBKS) {BB;X,Y,Z}**

Das *BBKS* liegt wie auch das *BKS<sub>i</sub>*, im Mittelpunkt der sechs Bohrungen des Zentralkörpers, an denen das  $\alpha$ -Gelenk befestigt ist (Abbildung 2.3(b)). Die *z-Achse* zeigt in Bezug auf den Roboterkörper senkrecht nach oben, die *y-Achse* hat Ihren Ursprung im Mittelpunkt der sechs Bohrungen und zeigt parallel zur Ebene des Roboterkörpers nach außen vom Zentralkörper weg. Die *x-Achse* bildet das Rechtssystem. Somit zeigt die *x-Achse* bei den Beinen  $B^1, B^3, B^5$  nach hinten und bei den linken Beinen  $B^0, B^2, B^4$  nach vorne. Daraus ist ersichtlich, dass das *BBKS* identisch zum *BKS<sub>2</sub>* ist.

### **Das Fußpunktkoordinatensystem (FPKS) {F;X,Y,Z}**

Der Ursprung des *FPKS* liegt im Endpunkt der Beinspitze, die *z-Achse* zeigt in das D-Segment, die *x-Achse* ist beim senkrecht vom Körper weg ausgestreckten Mittelbein parallel zur Zentralkörperlängsachse. Somit identisch mit der *x-Achse* des *BBKS*. Die *y-Achse* zeigt beim senkrecht auf dem Boden stehenden Bein entlang der Waagerechten vom Zentralkörper weg.

### **Das Symmetriekoordinatensystem (SKS) {S;X,Y,Z}**

Im Symmetriepunkt des Zentralkörpers ist das Symmetriekoordinatensystem zu finden. Es besitzt die gleiche Orientierung wie die Beinkoordinatensysteme (Abbildung 2.3).

### **2.1.3 Laufphasen des Schreitzyklus**

Das synchrone Zusammenwirken der Beine in einem Schreitzyklus bewirkt die Fortbewegung des Schreitroboters. Der Schreitzyklus jedes Beines lässt sich in 3 verschiedene Phasen unterteilen.

#### *Transferphase*

In der Transferphase oder auch Schwingphase hebt das Bein vom Boden ab und bewegt sich frei zur nächsten Stützposition.

#### *Auftrittsphase*

Die Auftrittsphase ist der Übergang von der Transferphase in die Stützphase

#### *Stützphase*

Während der Stützphase oder auch Stemmpphase steht das Bein auf den Boden und stützt den Körper. Dabei schiebt es den Körper in die gewünschte Bewegungsrichtung, indem es sich bezüglich des Körpers in die entgegengesetzte Richtung bewegt (bei der

Vorwärtsbewegung setzt das Bein vorne auf dem Boden auf und schiebt sich nach hinten, so dass der Körper nach vorne gestemmt wird).

Jedes Bein befindet sich abwechselnd in einer der drei Phasen.

## 2.2 Aufbau eines Roboterbeines

Die Beine stellen das Fortbewegungsorgan des Laufroboters da. Mit ihnen ist es möglich auf ebenem wie auch auf unebenem Untergrund zu laufen. Ausgestattet mit 10 Sensoren pro Bein sind die Beine ebenfalls der größte Wahrnehmungsapparat des LAURON. Da sich diese Diplomarbeit dem Thema „Sensorgesteuerte Bewegung und Bewegungsplanung eines Einzelroboterbeines“ widmet wird im Nachfolgendem genauer auf den Aufbau eines Einzelbeins des *LAURON IVb* eingegangen. Tabelle 2.1 listet die mechanischen Parameter, wie die zulässigen Winkelbereiche, die Abmessungen der einzelnen Beinsegmente (siehe Abb. 2.4(a)) sowie eine Auflistung der Sensoren pro Bein auf.

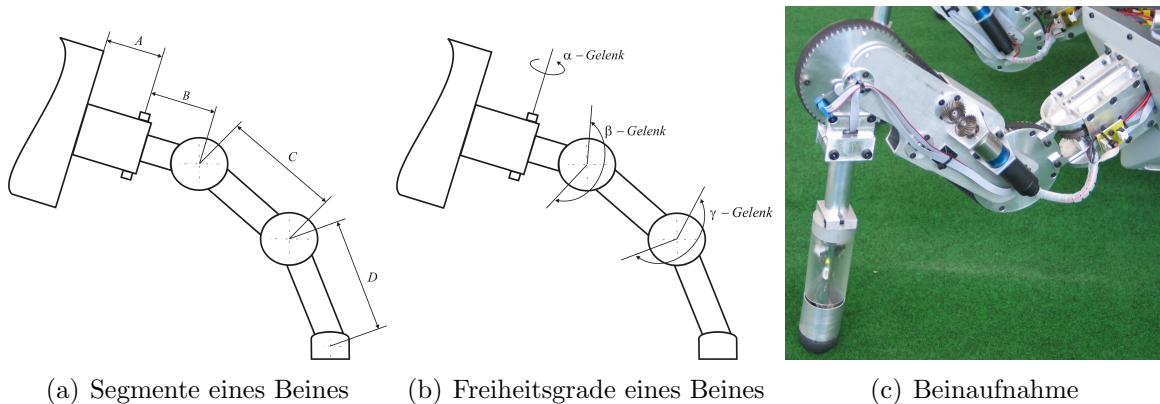


Abbildung 2.4: Aufbau eines Beines

### 2.2.1 Kinematischer Aufbau

Um eine Bewegung im Raum zu erzeugen besitzt jedes Bein entsprechend dem Vorbild der Stabheuschrecke aus der Natur drei Freiheitsgrade. Dadurch kann jede Position innerhalb des Arbeitsbereiches mit der Fußspitze angefahren werden. Die Gelenke des Beins sind Rotationsgelenke, somit werden die Freiheitgrade auch als Rotationsfreiheitsgrade bezeichnet. Beim *LAURON IVb* werden drei dieser Rotationsfreiheitsgrade hintereinander verkettet. Das erste Gelenk realisiert eine Bewegung bezogen auf das BBKS in der x-y-Achse. Das zweite und dritte Gelenk stehen parallel zueinander und bilden die Bewegung y-z-Achse.

Drei Freiheitsgrade genügen hierbei jedoch nicht um auch noch die räumliche Ausrichtung der Fußspitze zu beeinflussen. Aus dieses Grund wurde am untersten Ende des

<b>Beine</b>	
Aktive Freiheitsgrade	3
Arbeitsbereich der Gelenke	
$\alpha$ -Gelenk	$-55^\circ \dots +55^\circ$
$\beta$ -Gelenk	$-50^\circ \dots +110^\circ$
$\gamma$ -Gelenk	$-135^\circ \dots 0^\circ$
Antriebe	Gleichstrommotoren (12V)
Abmessungen eines Beines	
Gesamt	64,5cm
A-Segment	7cm
B-Segment	7,5cm
C-Segment	20cm
D-Segment	30cm
Masse eines Beines ca.	2,2kg
Sensoren	
Gelenkwinkelsensoren	1 optische Inkrementalsensoren pro Beingelenk
Motorwinkelsensor	1 Inkrementalsensor pro Motor
Fußkraftsensoren	3 orthogonale Komponenten im Unterschenkel
Motorstromsensoren	1 pro Gelenkmotor

Tabelle 2.1: Mechanische Parameter des Laufroboters Lauron IVb Beines

Fußes eine Gummihalbkugel angebracht um eine punktförmige Auftrittsfläche zu erhalten. Abbildung 2.4(c) zeigt den Aufbau des Beines.

## 2.2.2 Arbeitsbereich

Die Abbildung 2.5 zeigt den Arbeitsbereich eines Beines. Abbildung 2.5(a) zeigt den Arbeitsbereich in y-z-Ebene im BBKS. Wie zu erkennen ist, wird der Arbeitsbereich durch Kreisbögen beschränkt welche durch die minimalen und maximalen Gelenkwinkel gebildet werden. Abbildung 2.5(b) zeigt den Bereich in der x-y-Ebene. Punkte ausserhalb des Arbeitsbereiches können nicht von dem Fußendpunkt erreicht werden und müssen somit bei der Planung einer Beinbewegung berücksichtigt werden.

## 2.2.3 Sensoren

### Positionssensoren

Um die Gelenkwinkel, die mit Hilfe der Motoren angefahren werden, zu messen, werden Inkrementalsensoren verwendet. Diese Gelenkwinkelpositionssensoren befinden sich

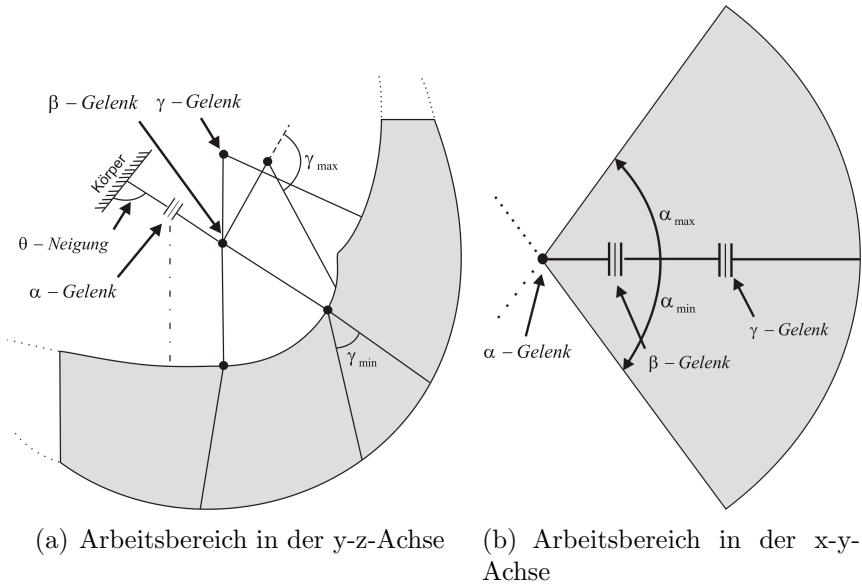


Abbildung 2.5: Arbeitsbereich des Beines

direkt auf der Gelenkachse (Abb.2.6). Die beim Überstreichen eines Winkels auftretenden Impulse werden per Hardware gezählt, so dass der gemessene Winkel schließlich als Zählerwert zur Verfügung steht. Die Umrechnung von Winkelwert  $W$  zu Impulswert  $I$  und umgekehrt wird mittels einer Linearfunktion durchgeführt [16].

$$I = a + W + i_0 \quad (2.1)$$

Dabei ist die Steigung  $a = \frac{1000}{360}$  die Anzahl der Impulse pro Winkelgrad. Der Achsenabschnitt  $i_0$  entspricht dem Impulswert bei einer Gelenkstellung von  $0^\circ$ . Um die initiale Gelenkwinkelposition zu bestimmen wird ein Referenzpunkt angefahren. Dieser befindet sich am mechanischen Anschlag eines Gelenkes. Von dort werden für jede Bewegung die Impulse gezählt.

### **Impulsgeber an der Motorwelle**

Ein direkt an der Motorwelle angebrachter Impulsgeber [5] erzeugt Impulse auf Grund der Drehung der Welle. Hierbei werden pro volle Umdrehung 128 Impulse erzeugt. Diese Impulse werden, wie bei dem Positionssensor, per Hardware gezählt und stehen anschliessend als Zählerwert zur Verfügung.



Abbildung 2.6: Inkrementaler Positionssensor

## Kraftsensoren

Die Kraftsensoren an den Füßen von Lauron IVb bestehen aus Stahlbauteilen mit definierten Biegesetzen und darauf angebrachten Dehnungsmessstreifen (kurz DMS), die in Vollbrücke zusammengeschaltet sind. Die schwachen Signale werden durch einen Differenzverstärker geleitet, bevor die A/D-Wandlung auf den Mikrocontrollern durchgeführt wird. Die Abbildung 2.7(a) zeigt die schematische Darstellung des Verformungskörpers, der wie ein „G“ geformt ist. Wirkt nun eine Kraft in negative  $z$ -Richtung, so wird der linke DMS gedehnt und der rechte gestaucht. Zur Filterung dieser Signale wird ein Medianfilter verwendet um die Ausreißer und das Rauschen in den Griff zu bekommen. Diese Signalwerte werden in Kraftwerte mit der Einheit Newton umgerechnet. Somit können die auf das Bein wirkenden Kräfte ( $x,y,z$ -Richtung) gemessen und in Newton angegeben werden. Die Umrechnung der Signalwerte  $S^i$  der Komponenten  $i \in \{x, y, z\}$  in Kraftwerte  $F^i$  (Newton), wird wie auch bei den Positionssensoren, mit Hilfe einer Linearfunktion durchgeführt [16]:

$$F^i = m^i \cdot (S^i - f_0^i) \quad (2.2)$$

Die Steigung  $m^i$  wird mit Hilfe von Referenzkraftgrößen bestimmt und die Achsenabschnitte  $f_0^i$  ergeben sich aus den Sensorwerten im entlasteten Zustand. Diese Kalibrierung ist notwendig, da die Parameter der Sensoren von Fertigungstoleranzen der verwendeten Bauteile, der Positionierung der DMS und der Verbindung des Dehnungsmessstreifens mit dem Verformungskörper abhängen.

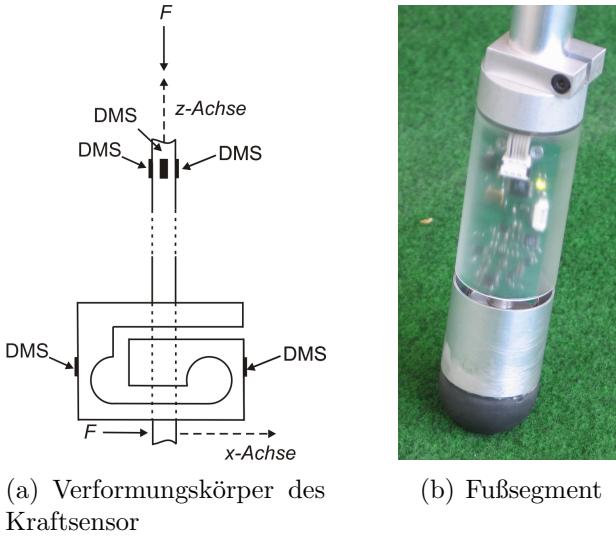


Abbildung 2.7: Fußsensor

## 2.2.4 Aktoren

Für die Umsetzung der Bewegung in den einzelnen Beinen werden so genannte Aktoren benutzt. Standardmäßig werden für Roboter mit einer Masse von unter 100 kg Elektromotoren eingesetzt. Beim Laufroboter *LAURON IVb* werden 12V 17W Gleichstrommotoren der Firma Faulhaber [4] eingesetzt. Sie sind direkt mit einem Planetengetriebe [6] gekoppelt, dass eine Umsetzung von 1:134 realisiert. Zudem besitzen sie einen Impulsgeber, welcher 128 Impulse pro Umdrehung der Motorwelle erzeugt.

## 2.2.5 Steuerungs Aufbau

Die Steuerung eines Beins ist untergliedert in zwei Steuerungsebenen. Eine Hardware nahe Ebene, welche für die Regelung und Umsetzung der Beinbewegung und zur Erfassung von Sensordaten zuständig ist. Realisiert wird dies durch einen DSP56F803 Mikrocontroller der Firma Freescale [10]. Die zweite Ebene plant und generiert die Beinbewegung. Sie befindet sich auf dem Standard Industrie PC/104. Zur Kommunikation der beiden Hardwarekomponenten wird ein Controller Area Network 2.0 (CAN) aus dem Hause Bosch eingesetzt. Nachfolgend werden nun die für die Steuerung des Einzelbeins eingesetzten Hardwarekomponenten vorgestellt.

### 2.2.5.1 Bein/Kopf Controller

Auf dem *LAURON IVb* Modell sind sieben baugleiche Bein/Kopf Controller zur Regelung der Motoren und zur Erfassung der Sensorsignale angebracht. Die Bein/Kopf Controller bestehen aus einer Mikrocontrollerplatine, auf welcher sich der DSP56F803

Mikrocontroller befindet, und einer Motortreiberplatine, welche für die Spannungsversorgung der Motoren und der Mikrocontrollerplatine zuständig ist.

### 2.2.5.2 Mikrocontrollerplatine

Auf der Mikrocontrollerplatine befindet sich der DSP-Mikrocontroller des Typs Freescale DSP56F803[10]. Diese Platine besitzt weiterhin Speichereinheiten (RAM, Flash-ROM), ein Altera FLEX 10K FPGA<sup>2</sup>[1] und Signalverarbeitungseinheiten für Input/Output sowie für die Kommunikation. Der Mikrocontroller übernimmt die Aufgabe der Regelung der Gelenkwinkel, Erzeugung der PWM-Signale zur Ansteuerung der Motoren. Das FPGA dient zur fortlaufenden Auswertung des Absolutcodes der Sensorsignale, welche anschließend in den Registern des FPGA für den Mikrocontroller zur Verfügung stehen, und die Generierung der Statuswerte der Motoren.

### 2.2.5.3 DSP56F803

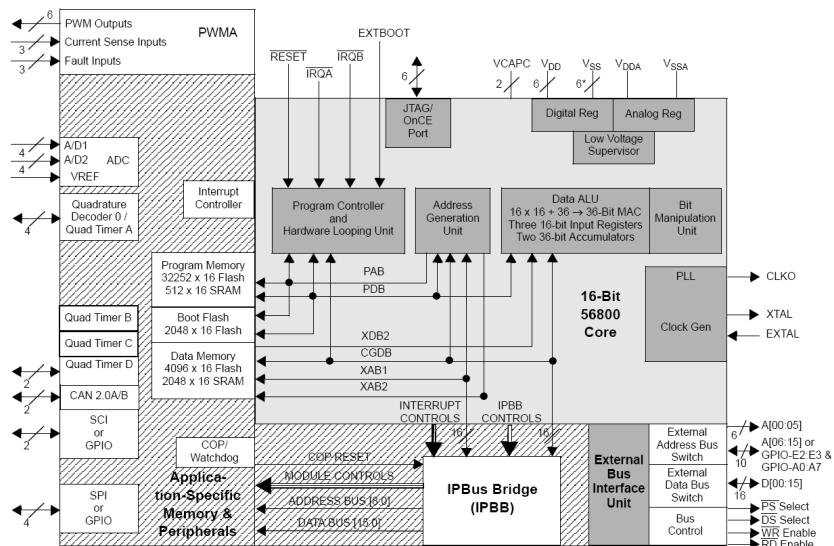


Abbildung 2.8: DSPF803 Blockdiagramm [10]

Der Mikrocontroller 56F803 entstammt der 56800 Prozessorfamilie. Er kombiniert auf einem einzigen 16 Bit Chip, die Prozessorleistung eines Digitalen Signal Prozessors (kurz DSP) und die Funktionalität eines Mikrocontrollers. Der 56F803 enthält die notwendige Peripherie um speziellen Applikationen gerecht zu werden wie z.B. Bewegungskontrolle, Encoding, Fahrzeugsteuerung, Motormanagement und die Industrielle Steuerung und Überwachung von Maschinen. Der 56800 Kern basiert auf der Harvardarchitektur und

---

<sup>2</sup>Field Programmable Gate Array

erlaubt bis zu sechs Operationen pro instruction cyle. Ferner unterstützt der 56F803 die Programmausführung entweder im internen oder externen Speicher. Auf zwei Operanden kann pro Anweisungszyklus vom on-chip RAM zugegriffen werden. Der Controller enthält zwei externe, getrennte Interruptzugänge und bis zu 16 General Purpose Input/Output (kurz. GPIO). Er enthält 31.5K word (16-bit) Programm Flashspeicher und 4K Word Datenflashspeicher (jeweils über die JTAG Schnittstelle programmierbar). Desweiteren enthält er 512 Word Programm RAM und 2K Word Daten RAM. Ein weiteres Feature des 56F803 ist die Integration eines Puls–Weiten–Modulator Moduls (kurz. PWM). Dieses Modul stellt drei unabhängige, individual programmierbare PWM Ausgangssignale zur Verfügung um eine verbesserte Motorsteuerungsfunktionalität zu gewährleisten. Zusätzlich enthält der Controller eine serielle Kommunikationsschnittstelle (kurz. SCI), ein Serial Peripheral Interface (kurz. SPI) und vier Quad Timer. Jede dieser Schnittstellen kann als GPIO genutzt werden, falls ihre regulären Funktionen nicht benötigt werden. Der 56F803 verfügt über eine Controller Area Network Schnittstelle (kurz. CAN Version 2.0) und einen internen Interrupt Controller [10].

### 2.2.5.4 Industrie-PC/104



Abbildung 2.9: PC/104 [23]

PC/104 ist ein Industriestandard [3] in dem die Spezifikationen einer kompakten PC-Architektur definiert sind. Bei diesem PC-Standard handelt es sich um einen Industrie-PC mit sehr geringen Abmessungen, der jedoch mit allen wesentlichen Komponenten eines Standard Desktop PC ausgerüstet ist. Somit unterstützt der Cool RoadRunner III PC/104 der Lippert GmbH [22] embedded PC Applikationen mit der Rechenleistung eines Pentium PCs. Der PC/104 besitzt ein AGP4x Slot, 10/100 BaseT Ethernet und einen AC-97 I/O on-Board Soundchip. Ferner bietet er einen CompactFlash Sockel und ein Ultra DMA-100 EIDE Interface und 512 MB SDRAM Systemspeicher. Ausgestattet ist der PC/104 mit einem Pentium III 933 MHz CPU und bietet somit genug Rechenleistung für eine Vielzahl an Anwendungen. Die Infrastruktur der Hardware wird mit dem VIA TwisterT Chipsatz realisiert, welcher einen integrierten Savage4 Grafikbeschleuniger von

S3 enthält. Dieser besitzt 32MB Grafikspeicher und kann Auflösungen bis zu 1600 x 1200 mit 16.7 Millionen Farben darstellen. Die Platine bietet Anschlussmöglichkeiten für TFT und CRC Monitore und bietet noch einen S-Video Out Anschluss. Zusätzlich zu den üblichen seriellen und parallelen Ports sind zwei USB Ports und ein IrDa auf dem Board integriert. Lokale Speichermedien können über das CompactFlash Modul angeschlossen werden. Systemerweiterungen sind über den PC/104 Bus möglich. Zu den Erweiterungen des PC/104 bei dem *LAURON IVb* Modell gehören eine PC/104-PCA-1 PCMCIA Karte mit zwei Slots zur Aufnahme einer wireless LAN<sup>3</sup> Karte um eine drahtlose Kommunikation zu ermöglichen, eine eNet-CAN-Bus Platine von Phytec zur Kommunikation mit den Bein/Kopf Controllern, eine Firewirekarte (IEEE 1394-Interface) zur Verbindung der Kameras am Kopfteil über Firewire, eine Platine zum Anschluß einer PS/2 Tastatur, Monitor und eine Platine mit Netzteil zur Spannungsversorgung.

## 2.3 Modular Controller Architecture

Modular Controller Architecture (*MCA*)<sup>[28]</sup> <sup>[20]</sup> ist eine am **Forschungszentrum für Informatik**<sup>[15]</sup> (*FZI*) entwickelte Steuerungsarchitektur für autonome Roboter. Für eine effiziente Entwicklung und Realisierung von Roboterprototypen ist es von Vorteil transferfähige Softwarekomponenten zu verwenden; wiederverwendbare Komponenten mit standardisierten Schnittstellen im Hinblick auf erweiterbare Architekturen. *MCA* ist ein modulares Software Framework, welches diese Kernpunkte realisiert. *MCA* ist kein automatisches Codegenerierungstool, weiterhin enthält es keine grafische Programmieroberfläche.

Der MCA Aufbau gleicht einer klassischen hierarchischen Kontrollstruktur. Auf der untersten Ebene befinden sich die Hardwareschnittstellen. Je weiter man sich im System nach oben bewegt, desto abstraktere Einheiten findet man vor. Auf der obersten Ebene des Systems ist die Benutzerschnittstelle zu finden. Generell existieren innerhalb des MCA-Systems zwei Datenstromrichtungen:

- **Sensor–Daten:** Der Fluss der Sensordaten beginnt in den Sensoren und geht über alle Module, welche diese Sensordaten erfordern bis hin zur Benutzerschnittstelle. Die Module zwischen dieser Datenstromrichtung können Sensordaten modifizieren und/oder neue Sensorwerte ableiten.
- **Control–Daten:** Der Controldatenfluss entspringt der obersten Ebene und bewegt sich abwärts zu den Aktoren.

Die MCA Basiseinheit wird durch *Module* bereitgestellt. Diese Module können in *Gruppen* zusammengefasst werden. Die Module innerhalb einer Gruppe kommunizieren über

---

<sup>3</sup>Local Area Network

frei konfigurierbare *Kanten* miteinander. Eine Gruppe kann in eine andere Gruppe gepackt werden oder innerhalb eines *Parts* als main loop eingesetzt werden. Ein Part stellt die Hauptausführungsumgebung innerhalb der Modular Controller Architecture dar. Ausführlichere Beschreibung zum Thema Modular Controller Architecture findet man in [31][28].

### 2.3.1 MCA-Steuerung des Lauron IVb

Zur Steuerung des *LAURON IVb* wurde eine Steuerungssystem auf der Basis der MCA-Architektur verwendet. Insgesamt besteht das System aus 3 Ebenen bzw. Gruppen, welche zu einer virtuellen Gruppe, der sogenannten „virtual MainGroup“ zusammen gefasst werden. Die „virtual MainGroup“ beinhaltet nur zwei Schnittstellen, SensorOutput und ControllerInput. Die 3 Ebenen der Steuerung unterteilen sich in :

- Ebene 1: „BehaviourGroup“.
- Ebene 2: „SignalProcessingGroup“.
- Ebene 3: „DSP–Remote-Part“.

Abbildung 2.10 zeigt die Verkantungen der einzelnen Ebenen. Der ControllerInput der „virtual MainGroup“ ist direkt mit dem ControllerInput der „BehaviourGroup“ verbunden und sind durch den ControllerOutput dieser Gruppen mit dem ControllerInput der nächsten Gruppe „SignalProcessingGroup“ verbunden. Letztlich ist die „SignalProcessingGroup“ über dem ControllerOutput mit dem ControllerInput des „DSP-Remote-Part“ verbunden. Umgekehrt wandern die Sensorinformationen von dem „DSP-Remote-Part“ über die „SignalProcessingGroup“ über die SensorOutput-SensorInput Kante nach oben zur „BehaviourGroup“. Der SensorOutput der „BehaviourGroup“ ist wiederum direkt mit dem SensorOutput der „virtual MainGroup“ verbunden.

#### BehaviourGroup

In der BehaviourGroup<sup>4</sup> wird das Verhalten der Laufmaschine geplant bzw. berechnet.

#### SignalProcessingGroup

In dieser Gruppe wird die Signalverarbeitungen durchgeführt. Zu den Aufgaben dieser Gruppe gehören:

---

<sup>4</sup>engl. behaviour = deutsch Verhalten

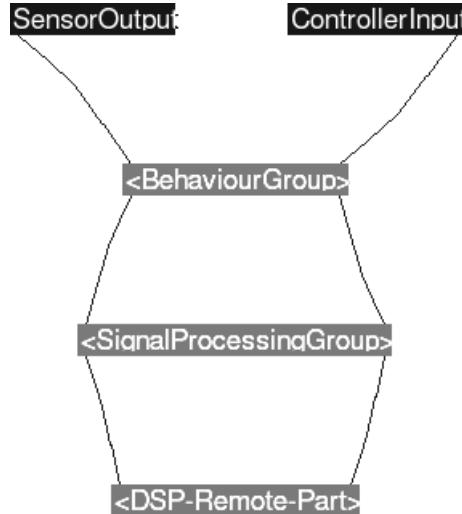


Abbildung 2.10: „Virtual MainGroup“ der MCA–Steuerung

- Glättung der gemessenen Werte mit Hilfe von Median- und Averagefilter.
- Umrechnung der Signale in Einheiten wie Newton, Winkelradian und Milliampere
- Umsetzung von Winkelgeberimpulsen zu Gelenkwinkel und umgekehrt.

### DSP-Remote-Part

Diese Gruppe verbindet sich direkt mit den Mikrocontrollern der Laufmaschine *LAURON IVb* und stellen somit die unterste Ebene des MCA–Steuerung dar. Die Module beinhalten die Soll-Winkel und Reset-Befehle für die zu steuernden Motoren sowie die Ist-Winkel, Motorenströme, Reset-Antworten der Motoren, Sensorsignale der Fußsenso- ren bzw. des Neigungssensors.

## 2.4 Grundlagen der Regelungstechnik

Die Aufgabe einer Regelung besteht darin, eine physikalische Größe in einem technischen System, z.B. die Motordrehzahl, trotz Einwirkung eines Lastmoments, das als Störgröße bezeichnet wird, aufgrund eines Vergleiches zwischen Führungsgröße(Sollwert) und der gemessenen Regelgröße mit Hilfe eines Reglers konstant zu halten. In der Robotik findet

man überall da eine Regelung, wo es gilt eine durch die Software geplante Bewegung, mit Hilfe von Aktoren in eine reale Bewegung umzusetzen. Sie bildet die Schnittstelle zwischen der Software und der Hardware. Hierbei wird zwischen einer Steuerung und einer Regelung unterschieden. Das Steuern ist ein rein vorwärts gerichteter Prozess ohne Rückkopplung. Die Ausgangsgröße wird dabei nicht überwacht und kann sich durch Störungen von außen verändern. Bei einer Regelung hingegen wird die Ausgangsgröße durch eine Rückkoppelung ständig überwacht und bei einer Störung wird versucht diese nachzuregeln.

Das vorliegende Kapitel stellt nun eine Einführung in die Regelungstechnik dar. Dabei sollen die wichtigsten Begriffe und mathematischen Grundlagen aufgezeigt werden.

### 2.4.1 Regelungssystem

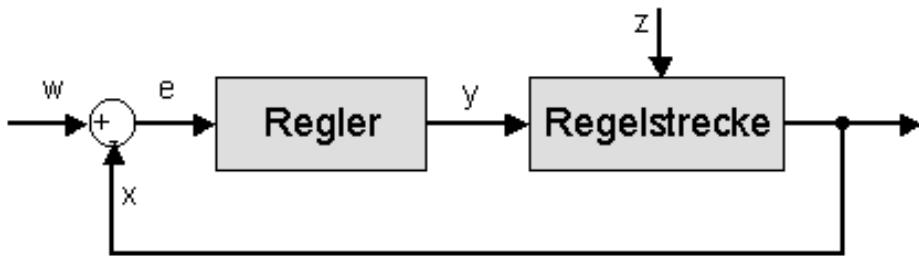


Abbildung 2.11: Allgemeiner Regelungskreis [27]

Abbildung 2.11 zeigt den allgemeinen Grundaufbau eines Regelungssystems bestehend aus:

- *Regler*: Ist der Teil des Regelkreises, der unter Berücksichtigung der dynamischen Eigenschaften der Regelstrecke aus der Regelabweichung die Korrekturmaßnahmen zum Ausregeln ergreift.
- *Regelstrecke*: Ist der Teil des Regelkreises, der vom Regler ausgeregelt werden soll.
- *Führungsgröße (Sollwert)*  $w$ : Vorgegebener Wert, auf dem die Regelgröße durch die Regelung gehalten werden soll. Sie ist eine von der Regelung nicht beeinflusste Größe und wird von außen zugeführt.
- *Regelgröße (Istwert)*  $x$ : Ist die Ausgangsgröße der Regelstrecke, die zum Zweck des Regelns erfasst und zum Vergleich rückgeführt wird.
- *Regelabweichung e*: Differenz zwischen Führungsgröße und Regelgröße  $e = w - x$ , bildet die eigentliche Eingangsgröße des Reglers.

- *Stellgröße y*: Ausgangsgröße der Regeleinrichtung und zugleich Eingangsgröße der Strecke. Sie überträgt die steuernde Wirkung des Reglers auf die Strecke.
- *Störgröße z*: Eine von außen wirkende Größe, die eine Änderung des Istwertes der Regelgröße bewirkt und einen Regelvorgang auslöst.

## 2.4.2 Übertragungsglieder

Die Regelstrecke stellt den zu regelnden Teil dar und umfasst normalerweise ein Reihe von einzelnen Gliedern, sogenannte Übertragungsglieder. Diese Glieder weisen ein bestimmtes Übertragungsverhalten zwischen einer Eingangsgröße und einer Ausgangsgröße auf und werden entsprechend ihrem Zeitverhalten charakterisiert. Nachfolgend werden nun die wichtigsten und in dieser Arbeit verwendeten Glieder vorgestellt:

- *Proportionalglied (P-Glied)*

Die Eingangsgröße stellt sich über einen Faktor  $K$  am Ausgang ein. Für den Fall von  $K = 1$  entspricht die Eingangsgröße gleich der Ausgangsgröße .

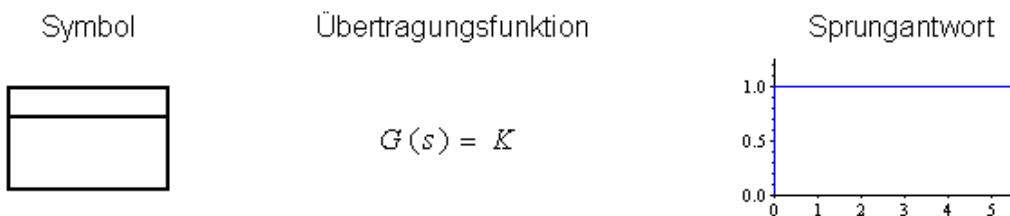


Abbildung 2.12: Proportionalglied (P-Glied) [27]

- *Integrator (I-Glied)*

Übertragungsglied mit einem integrativen Übertragungsverhalten.

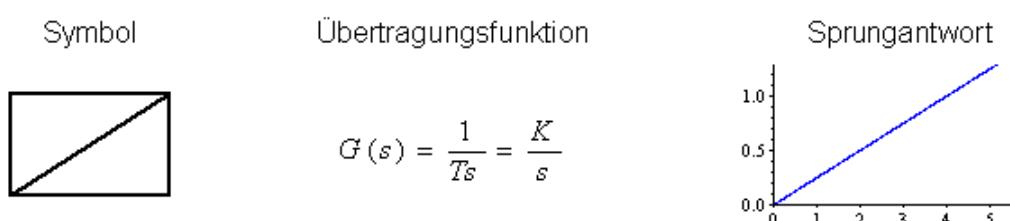


Abbildung 2.13: Integrator (I-Glied) [27]

- *Verzögerungsglied 1. Ordnung (PT1-Glied)*

P-Glied mit einem nicht vernachlässigbaren Zeitverhalten

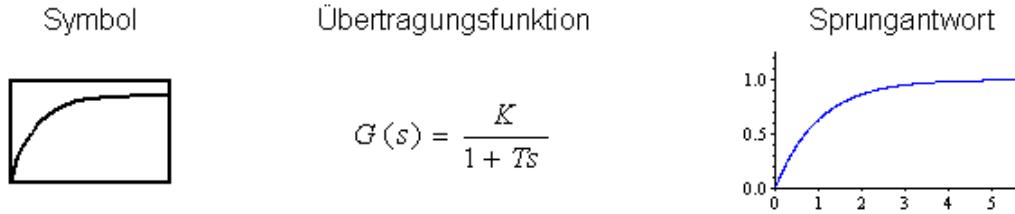


Abbildung 2.14: Verzögerungsglied 1. Ordnung (PT1-Glied) [27]

- *Verzögerungsglied 2. Ordnung (PT2-Glied)*  
Zwei hintereinander geschaltete PT1-Glieder

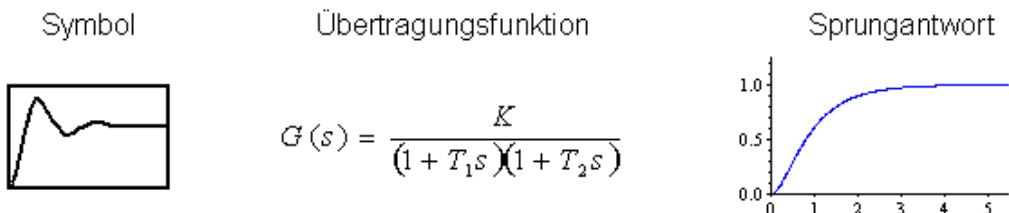


Abbildung 2.15: Verzögerungsglied 2. Ordnung (PT2-Glied) [27]

### 2.4.3 PI Regler

PI-Regler werden häufig in der Industrie als Standardregler eingesetzt. Durch seine Kombination aus proportionalem (P-Glied) und integralem (I-Glied) Verhalten besitzt dieser den Vorteil schnell auf einen Sprung der Führungsgröße oder bei Eintreten einer Störung zu reagieren. Und besitzt trotzdem die Eigenschaft die Regelgröße exakt auszuregeln.

Abbildung 2.16 zeigt das Blockschaltbild eines zeitkontinuierlichen PI-Reglers. Hierbei sind die Übertragungsglieder P und I parallel geschaltet. Die Übertragungsfunktion ergibt sich demnach durch Addition der Übertragungsgleichungen der einzelnen Teilglieder und lautet somit:

$$G_{PI}(s) = K_P + \frac{K_I}{s} \quad (2.3)$$

Um die Gleichung mit der Nachstellzeit auszudrücken, kann die Gleichung (2.3) umgestellt werden zu:

$$G_{PI} = K_P - \cdot \left[ 1 + \frac{1}{T_N s} \right] \quad (2.4)$$

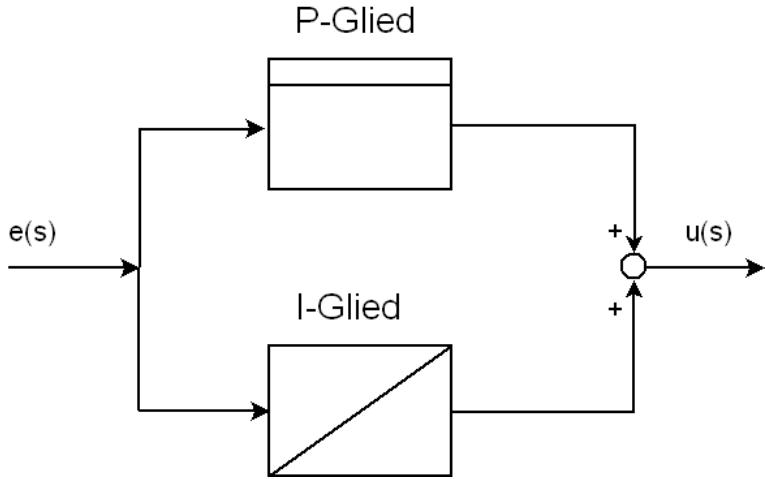


Abbildung 2.16: Blockschaltbild PI-Regler

mit

$$T_N = \frac{K_P}{K_I} \quad (2.5)$$

Die Reglergleichung lautet somit:

$$ut = K_P \cdot \left[ e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau \right] \quad (2.6)$$

Abbildung 2.17 zeigt das Symbol und die dazugehörige Sprungantwort eines PI-Reglers.

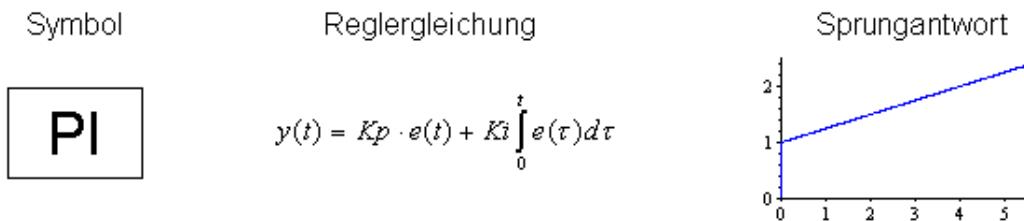


Abbildung 2.17: PI-Regler [27]

#### 2.4.4 Regelkonzepte

In der Regelungstechnik gibt es verschiedene Konzepte wie man einen Regelkreis aufbauen kann. Meist ergibt sich der Aufbau des Regelkreises direkt aus den Anforderungen heraus.

In diesem Abschnitt sollen kurz die wichtigsten und weit verbreitetsten Regelkonzepte vorgestellt werden.

### Einschleifiger Regelkreis

Der einschleifige Regelkreis ist wegen seiner Einfachheit wohl das am häufigsten angewandte Konzept. Er besteht aus einem Regler, ein Vergleicher, wobei dieser meist direkt in dem Regler integriert wird, ein Messwerterfassung (z.B. Lagesensor) und einer Regelstrecke. Hierbei wird nur die zu regelnde Größe erfasst und versucht diese auszuregeln.

### Kaskadenregelung

Bei einschleifigen Regelkreisen, die Regelstrecken mit großen Zeitkonstanten oder Totzeiten haben, werden Störeinflüsse oder Führungsgrößenänderungen nur langsam ausgeregt. Durch Erweitern der Regerstruktur kann das Übergangsverhalten des geschlossenen Regelungssystems wesentlich verbessert werden. Hierzu können weitere Hilfsgrößen erfasst werden, welche separat geregelt werden. Diese zusätzlichen Regelkreise können kaskadiert verbunden werden. Abbildung 2.18 zeigt eine Beispiel einer Lageregelung mit Kaskadenstruktur. Hierbei werden neben der Regelgröße, dem Winkel, zwei weitere Hilfsregelgrößen erfasst: die Winkelgeschwindigkeit und der Ankerstrom.

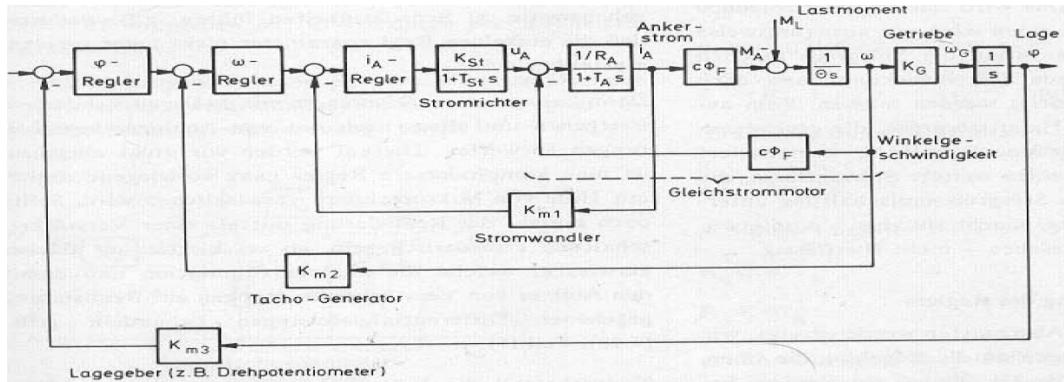


Abbildung 2.18: Lageregelung mit unterlagerter Drehzahl- und Ankerstromregelung [9]

Es entstehen drei überlagerte Regelkreise, bei denen der Lageregler den Sollwert für den Geschwindigkeitsregler und dieser wiederum den Sollwert für den Stromregler liefert. Zur Synthese werden die Regelkreise nacheinander von Innen nach Außen entworfen. Diese Vorgehensweise wird ausführlich in [9] [26][2] und [25] beschrieben.

## Störgrößenaufschaltung

Eine ganz andere Möglichkeit, das dynamische Verhalten einer Regelung durch Rückführung zu verbessern, liegt in der Störgrößenaufschaltung vor. Darunter versteht man die Rückführung einer Störgröße den Bereich des Reglers, mit Ziel, die Einwirkung der Störgröße auf den Regelkreis nach Möglichkeit zu kompensieren. Sofern die Störgröße messbar ist.

## Nichtanalytische Regelungskonzepte

In einigen besonderen Fällen finden zur Lösung von Regelaufgaben auch nicht analytische Ansätze Anwendung. Als gängige Vertreter seien hier die so genannten Fuzzy-Regelungen oder Neuronale Netze erwähnt. Sie werden verwendet, wenn kein mathematisches Modell der Regelstrecke vorliegt. In jüngsten Entwicklungen werden auch hybride Verfahren genutzt, bei denen ein Fuzzy-Algorithmus oder ein Neuronales Netz einen klassischen kontinuierlichen Regler konfigurieren [29].

## 2.5 Grundlagen der Kinematik

Die Kinematik ist die Lehre von der Bewegung von Punkten und Körpern im Raum[32]. Mit ihr kann man das Verhältnis zwischen den Winkeln der einzelnen Gelenke des Roboterbeins und der Position im kartesischen Raum des Fußendpunktes (*End Effector*) beschreiben. Somit kann ein vorgegebener Punkt des *End Effector* im kartesischen Raum in Gelenkwinkel der einzelnen Gelenke umgerechnet werden und umgekehrt mit den einzelnen Gelenkwinkeln die Position des Fußendpunktes im Raum bestimmt werden. Hierbei wird die Berechnung von Gelenkwinkel zu Fußpunkt im kartesischen Raum *direkte Kinematik* genannt und von dem Endpunkt zu den Gelenkwinkeln *inverse Kinematik*. Die

Die kinematische Analysen des Beines können mit Hilfe einer kinematischen Kette durchgeführt werden. Unter einer kinematischen Kette versteht man die Anordnung mehrerer starrer Körper(*Link*) die jeweils mit einer Achse(*Joint*) miteinander verbunden sind. Im Fall des Schreitroboterbeines stellen die einzelnen Segmente die *Links* der kinematischen Kette dar und die Gelenke die *Joints*. Abbildung 2.19 zeigt beispielhaft eine kinematische Kette für einen Roboterarm mit 3 Gelenken.

Um eine kinematische Kette mathematisch beschreiben zu können entwickelten Denavit und Hartenberg ein nach ihnen benanntes Verfahren. Die Beschreibung basiert darauf, dass gegenseitige Lager zweier Segmente durch vier Parameter beschrieben werden können. Diese Parameter werden auch DH Parameter genannt [33] und beschreiben den Übergang eines Gelenkes zum jeweils nächsten Gelenk der kinematischen Kette. Hier-

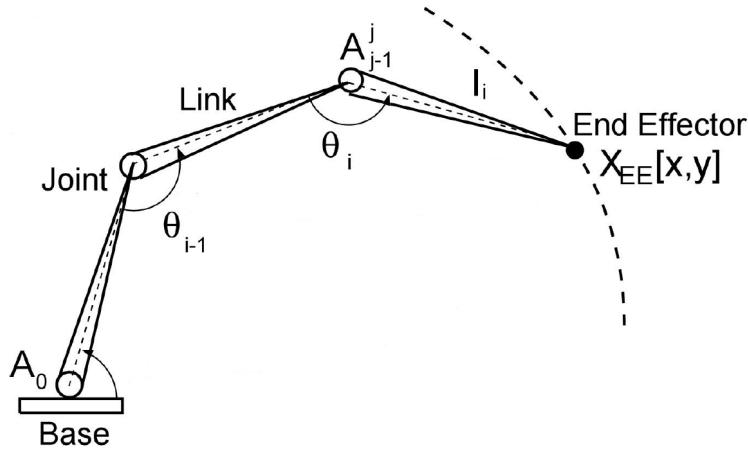


Abbildung 2.19: Kinematische Kette [7]

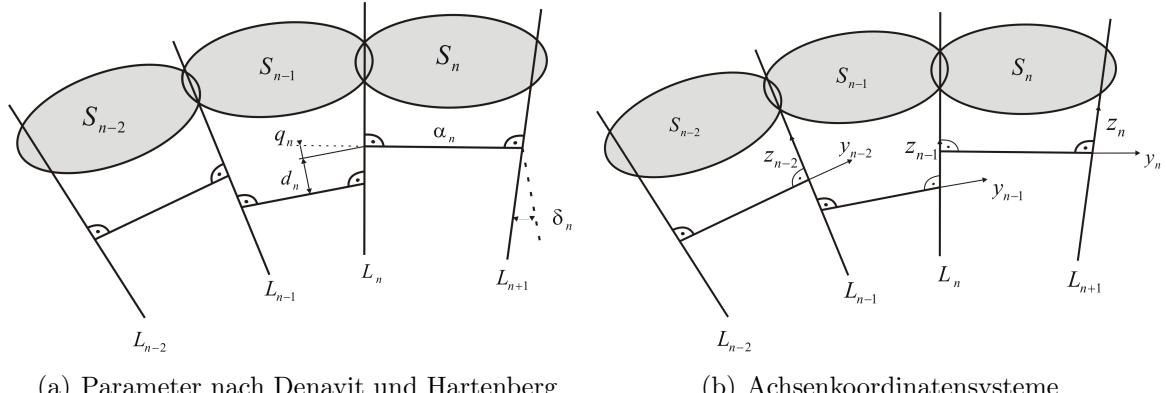


Abbildung 2.20: Denavit-Hartenberg-Verfahren [17]

zu ist es notwendig in jedem Gelenk ein eigenes Koordinatensystem zu definieren, um mit ihrer Hilfe und der DH-Parameter eine Transformationsgleichung aufzustellen zu können. Abb. 2.20 zeigt eine kinematische Kette mit drei Segmenten  $S_{n-2}$  bis  $S_n$ . Parameter  $a_n$  entspricht dem Abstand zweier Gelenke. Sind die Gelenke zueinander versetzt beschreibt  $d_n$  diesen Abstand. Im Fall der LAURON IVb Beine gibt es keine Versätze zwischen den Gelenken, somit bleibt der Parameter in diesem Fall für alle Gelenke Null. Der dritte Parameter  $\delta_n$  gibt die Verdrehung der beiden aufeinander folgenden Gelenke  $L_n$  und  $L_{n+1}$  wieder. Der letzte Parameter  $q_n$  ergibt sich aus der Drehung um die Achse und ist somit gleich dem Winkel des Gelenks. Die ersten 3 Parameter sind konstruktionsbedingt und verändern sich während der Bewegung des Beines nicht. Der vierte Winkel entspricht der aktuellen Stellung des Gelenkes und ist somit variabel.

### 2.5.1 Direkte Kinematik

Die *direkte Kinematik* beschreibt den Übergang der Gelenkwinkeln in den kartesischen Raum des Endeffektors (Abb. 2.21).

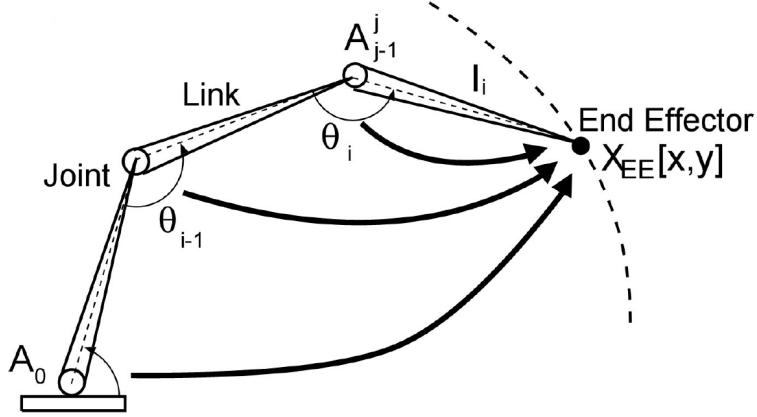


Abbildung 2.21: Direkte Kinematik [7]

Mit der Gleichung

$$\vec{s} = f(\vec{q}) \quad (2.7)$$

kann die direkte Kinematik ausgedrückt werden, wobei der Vektor  $\vec{s} = (x, y, z)^T$  die Position des Endeffektors entspricht und  $\vec{q} = (q_1, q_2, \dots, q_n)^T$ , mit  $n =$  Anzahl der Gelenke, die Gelenkwinkel enthält. Die Funktion  $f$  (2.7) ist eine nichtlineare Funktion. Eine lineare Änderung der Gelenkwinkel entspricht somit keiner linearen Änderung der Position des Endeffektors.

Zur Berechnung der direkten Kinematik kann eine homogene Transformationsmatrix verwendet werden. Zur Aufstellung dieser Matrix müssen zunächst die Transformationsmatrizen für den Übergang zwischen den einzelnen Gelenken ermittelt werden. Mit Hilfe der DH-Parameter kann man diese Transformationsmatrizen (2.8) aufstellen.

$${}_{n-1}^n T = Rot(z_{n-1}, q_n) \cdot Trans(0, 0, d_n) \cdot Trans(a_n, 0, 0) \cdot Rot(y_n, \theta_n) \quad (2.8)$$

Die Transformationsgleichung (2.8) ist hierbei zusammengesetzt aus den Einzeltransformationen:

1. Rotation um die Achse  $z_{n-1}$  um den Winkel  $q_n$

2. Verschiebung entlang der Achse  $z_{n-1}$  um  $d_n$
3. Verschiebung entlang der Achse  $y_n$  um  $a_n$
4. Rotation um die Achse  $y_n$  um den Winkel  $\theta_n$

Da es sich bei dem Roboterbein um vier Segmente handelt, müssen entsprechend vier Transformationsgleichungen (2.8) mit einander multipliziert werden, um mit den Gelenkwinkeln aller Gelenke den Fußendpunkt berechnen zu können.

$${}_4^0 T = {}_1^0 T \cdot {}_2^1 T \cdot {}_3^2 T \cdot {}_4^3 T \quad (2.9)$$

Durch das Ausmultiplizieren der einzelnen Transformationsgleichungen erhält man die homogene Transformationsmatrix (2.10) für das gesamte Bein.

$${}_4^0 T = {}_1^0 T \cdot {}_2^1 T \cdot {}_3^2 T \cdot {}_4^3 T = \begin{pmatrix} {}_4^0 R & {}_4^0 \vec{r} \\ 0 & 1 \end{pmatrix} \quad (2.10)$$

Hierbei beschreibt der Vektor  ${}^0 \vec{r}$  die Position des Endpunktes, relativ zum Anfangskoordinatensystem, in Abhängigkeit der einzelnen Gelenkwinkel.

### 2.5.2 Inverse Kinematik

Im Gegensatz zu der direkten Kinematik versucht die inverse Kinematik, für eine gegebene Position des Endeffektors auf die möglichen Winkel der Gelenke zu schliessen (Abb. 2.22).

Hierbei stellt sich das Problem der Suche nach der Umkehrfunktion bzw. inversen Funktion von (2.7) dar.

$$\vec{q} = f^{-1}(\vec{s}) \quad (2.11)$$

Hierbei treten die Schwierigkeiten auf, dass die meisten Positionen des Endeffektors durch mehrere Konfigurationen der Gelenkwinkel beschrieben werden können. Die inverse Kinematik besitzt somit in den meisten Fällen mehrere Lösungen, wobei einige der Lösungen durch Beschränkungen der Gelenkwinkel nicht verwendbar sind. Zudem gibt es eine unendlich große Menge an Positionen, für welche es keine Lösung der inversen Kinematik gibt.

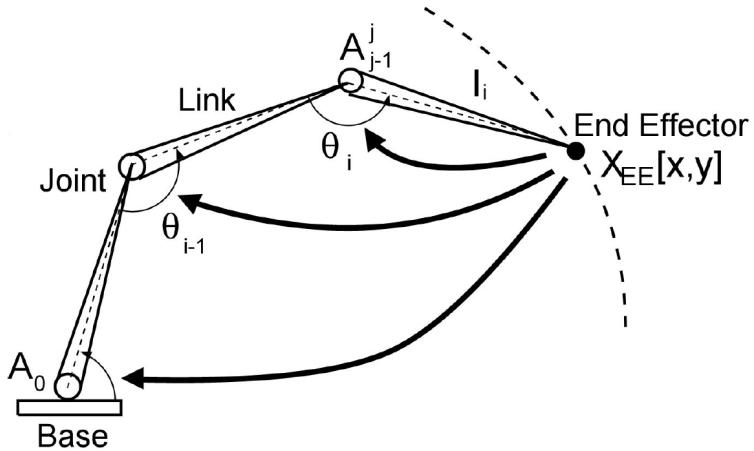


Abbildung 2.22: Inverse Kinematik [7]

Eine weitere Schwierigkeit der inversen Kinematik stellen sogenannte Singularitäten dar. Singularitäten sind Positionen die angefahren werden können, für die es aber unendlich viele Lösungen gibt. Dies ist der Fall, wenn Achsen aufeinander liegen oder mehrere Achsen parallel sind. Es kann somit keine eindeutige Lösung mehr bestimmt werden. Einige Steuerungen brechen das Programm beim Durchfahren eines solchen Punktes ab. Aus diesem Grund wird in der Praxis versucht diese Problem zu vermeiden, indem man die Bewegungsbahn des Endeffektors so abändert, dass die Nähe solcher Singularitäten gemieden werden.

Bei der Funktion  $f$  aus Gleichung (2.7) handelt es sich um eine nichtlineare Funktion. Die Suche nach  $f^{-1}$  entspricht somit der Berechnung eines nichtlinearen Gleichungssystems. Zur Lösung des inversen kinematischen Problems gibt es kein allgemein anwendbares Verfahren. In der Robotik sind jedoch drei Verfahren weit verbreitet.

1. Algebraische Lösung
2. Geometrische/Analytische Lösung
3. Numerische Lösung

#### *Algebraische Lösung*

Die Algebraische Lösung versucht durch sukzessive Invertierung der Denavit-Hartenberg-Transformationsmatrizen und somit Lösung des nichtlinearen Gleichungssystems, die Gelenkwinkel aufgrund der Endeffektorpositon zu ermitteln.

$${}^4T = {}^0_1 T^{-1} \cdot {}^1_2 T^{-1} \cdot {}^2_3 T^{-1} \cdot {}^3_4 T^{-1} \quad (2.12)$$

## 2 Grundlagen

### Geometrische/Analytische Lösung

Aufgrund des Wissens über die Geometrie des Roboters wird versucht, zum Beispiel mit Hilfe von Kosinussatz oder Sinussatz den Gelenkwinkelvektor  $\vec{q}$  zu berechnen.

### Numerische Lösung

Die numerische Lösung der inversen Kinematik sucht nach einer linearen Annäherung an  $f^{-1}$ . Hierbei müssen zuerst Teilgleichungen der Vektorfunktion  $f$  bestimmt werden.

$$\begin{aligned}x &= f_1(q_1, q_2, \dots, q_n) \\y &= f_2(q_1, q_2, \dots, q_n) \\z &= f_3(q_1, q_2, \dots, q_n)\end{aligned}$$

Diese Teilgleichungen können aus dem Vektor  ${}^0\vec{r}$  der Transformationsmatrix (2.10) abgelesen werden. Jede dieser Gleichungen wird nach jeder Variable für die Gelenkstellung partiell abgeleitet:

$$J(\vec{q}) = \left( \frac{\delta f_i}{\delta q_i} \right) = \begin{pmatrix} \frac{\delta f_1}{\delta q_1} & \dots & \frac{\delta f_1}{\delta q_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_3}{\delta q_1} & \dots & \frac{\delta f_3}{\delta q_n} \end{pmatrix} \quad (2.13)$$

Diese  $3 \times n$  Matrix wird auch Jacobimatrix genannt. Die Jacobimatrix entspricht der ersten Ableitung der Funktion  $f$  im Punkt  $\hat{q}$  (2.14), und beschreibt somit die Änderung der Position des Endeffektors ( $\Delta\vec{s}$ ), bei einer gegebenen Änderung der Gelenkstellungen ( $\Delta\vec{q}$ ) im Punkt  $\hat{q}$ .

$$\Delta\vec{s} = J(\hat{q})\Delta\vec{q} \quad (2.14)$$

$$\vec{s} = f(\hat{q}) + J(\hat{q})(\vec{q} - \hat{q}) \quad (2.15)$$

Die Gleichung (2.15) entspricht einer linearen Annäherung an die Funktion  $f$  im Punkt  $\hat{q}$  (Abb.2.23).

Wie bei der Abbildung 2.23 zu erkennen ist, ist die Annäherung nur im näheren Bereich des Punktes  $\hat{q}$  noch akzeptabel. Mit größerem Abstand steigt die Abweichung zur tatsächlichen Funktion  $f$ .

Durch Umstellen der Gleichung (2.14) kann man nun eine Näherung der inversen Funktion  $f^{-1}$  erhalten.

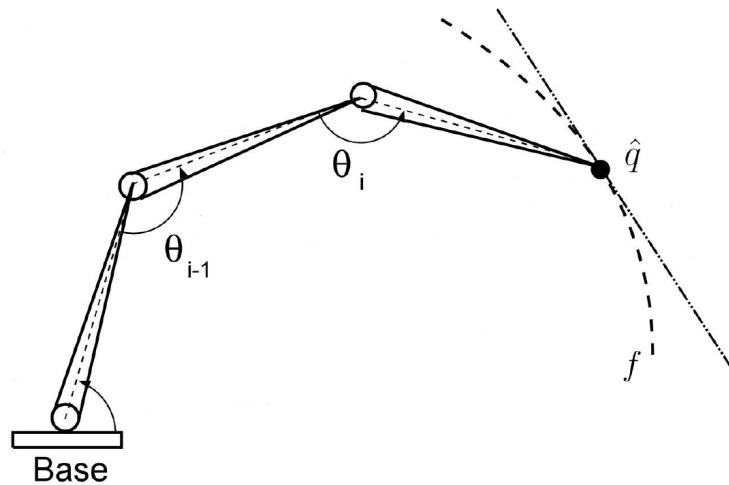


Abbildung 2.23: Annäherung an  $f$  [7]

$$\Delta \vec{q} = J(\hat{q}) \Delta \vec{s} \quad (2.16)$$

$$\vec{q} = \hat{q} + J(\hat{q})(\vec{s} - f(\hat{q})) \quad (2.17)$$

Durch die Gleichung (2.17) ist es nun möglich, bei gegebenen  $\vec{s}$  die dazu gehörenden Gelenkwinkel  $\vec{q}$  zu ermitteln.

# 3 Neue Regelungskonzepte für Beinantriebe und Kraft

In den vorhergegangenen Kapiteln wurden Grundlagen wie der Aufbau des Robotergeines, regelungstechnische Grundlagen und Grundlagen zur Bewegungskoordination beschrieben. In den folgenden Abschnitten werden, aufbauend auf diesen Grundlagen, Konzepte zur sensorgesteuerten Bewegung sowie zur Bewegungsplanung erarbeitet und vorgestellt.

Beobachtungen der Beinantriebsregelung des *LAURON IVb* haben gezeigt, dass sie nicht in hinreichender Genauigkeit die Position des Beines auszuregeln vermag. Hierbei kommt es vor, dass Regelabweichungen von Soll- und Ist-Gelenkwinkel nicht ausgeregelt werden. Beim Schreiten durch ein unebenes Gelände oder bei technischen Anwendungen wie Bohroperationen oder Positionieraufgaben werden jedoch präzise Beinbewegungen gefordert.

Ein Ziel dieser Arbeit ist es eine Verbesserung der Regelung in Hinsicht auf die Positionsgenauigkeit zu erarbeiten und anschließend umzusetzen. Hierzu wird in den folgenden Unterkapiteln zunächst die bereits vorhandene Beinregelung und das zuregelnde System untersucht und anschließend ein neues Regelungskonzept, basierend auf zusätzlich gewonnenen Sensorinformationen, vorgestellt.

In einer Vorarbeit von Kai Wetzelsberger [31] wurden kraftgeführte Bewegungen entwickelt und erfolgreich in der obersten Steuerungsebene (MCA-Steuerung) integriert und getestet. Diese gemessenen Kraftwerte, die für diese Kraftregelung nötig sind, werden von dem Mikrocontroller erfasst, mit Hilfe des CAN-Busses an den Industrie PC/104 geschickt, dort von der DSP-Remote-Part der MCA-Steuerung gelesen und an die Signal-ProcessingGroup weitergereicht, welche wiederum die Werte mittels einer Medianfilters glättet. Dies führt zu einer langen Signallaufzeit was die Reaktionszeit der Kraftregelung verschlechtert. Aus Sicht der Regelung erweist es sich jedoch als vorteilhaft, die Kraftregelung in demjenigen Raum zu realisieren, in dem die physikalischen Sensoren die Informationen liefern. Was kürzeste Regelzeiten und Totzeiten der Regelung ergibt.

Das zweite Ziel dieser Arbeit besteht nun darin die Kraftregelung innerhalb des Mikrocontrollers umzusetzen und damit die Beinregelung zu erweitern. Hierzu werden zunächst die Regelalgorithmen der kraftgeführten Bewegung vorgestellt und anschließend die allgemeinen Schritte erläutert, welche nötig sind eine Kraftregelung durchzuführen.

## 3.1 Regelung der Beinantriebe

Ein Schreitroboter ist ein komplexes System, dessen Hard- und Software unterschiedlichste Aufgaben bewältigen muss. Dies reicht von einfacher Erfassung der Beingelenkpositionen und deren Regelung, über die Planung von Bahnkurven, der Erfassung der Umwelt, bis zur Routenplanung und dem autonomen Lösen von Aufgaben und dem Reagieren auf Ereignisse.

Die Qualität, mit der eine übergeordnete Hierarchieebene ihre Aufgaben erfüllen kann, steht und fällt dabei mit der Güte, mit der die ihr untergeordnete Ebene ihre Aufgabe erfüllt. So wird beispielsweise ein Fußpunkt von dem die Bahnplanung verlangt, er soll sich entlang einer geraden Linie bewegen, dies nur tun, wenn die einzelnen Gelenke die zu jedem Zeitpunkt dafür berechneten Winkel anfahren. In diesem Kapitel wird diese unterste Ebene, die Regelung, also das Nachführen der von einer übergeordneten Regelung/Steuerung vorgegebenen Gelenksollpositionen, behandelt.

### 3.1.1 Untersuchung am Teilsystem

Es hat sich nach Überlegung der Schluss ergeben, dass zur sensorgesteuerten Bewegung ein kontinuierlicher Regler das Mittel der Wahl ist. Doch bevor mit der Überlegung eines geeigneten Regelalgorithmus begonnen werden kann, muss das technische System, die Regelstrecke dahin gehend untersucht werden, ob gewisse grundlegende Voraussetzungen erfüllt sind, die den Einsatz eines kontinuierlichen Reglers erlauben, bzw. ob durch seinen Einsatz die Regelungsaufgabe in der gewünschten Qualität gelöst werden kann. Da das Roboterbein aus 3 gleichen Segmenten mit jeweils einem Motor, einen Inkrementalgeber am Gelenk, einen Impulsgeber an der Motorwelle und Motorstromsensoren auf der Beinkontrollerplatine (siehe Abschnitt 2.2) aufgebaut sind, genügt es nur einen Abschnitt des Beines zu untersuchen und analog auf die anderen zu schließen.

#### 3.1.1.1 Ansteuerung des Motors

Zur Realisierung von beiden Bewegungsrichtungen für die Gelenkantriebe werden Schaltverstärker in Vollbrückenschaltung genutzt, dessen Prinzip Abbildung 3.1 zeigt. Die vier Leistungstransistoren V1 bis V4 sind so zusammengeschaltet, dass sie eine Vollbrücke bilden. Die Dioden leiten die Induktionsspannung der Motorwicklung gegen die Versorgungsspannung  $U_s$  ab. Die Brücke wird beim *LAURON IVb* mit der Vorzeichen-Betrags-Ansteuerung (Sign/Magnitude Control) Methode angesteuert. Das bedeutet, es wird je nach dem ob  $U_A$  positiv oder negativ sein soll, immer nur ein Transistorpaar (V1/V4 oder V2/V3) geschaltet.

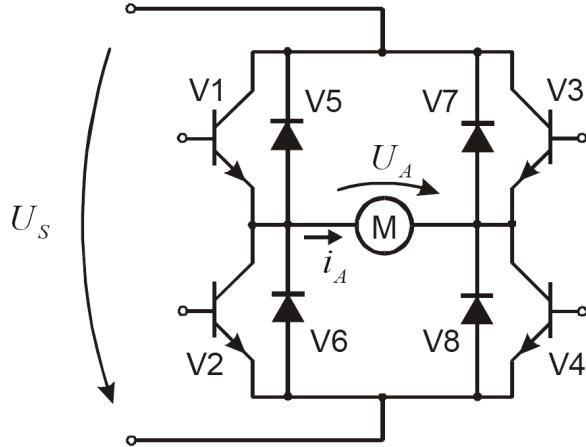


Abbildung 3.1: Brückenschaltung zur Motorsteuerung aus [17]

### 3.1.1.2 Erzeugung der Motorspannung

Zur Erzeugung der Motorspannung wird eine Puls-Weiten-Modulation verwendet. Bei der Puls-Weiten-Modulation bleibt die Periodendauer der aufeinander folgendem Pulse  $n_{PWM}$  konstant. Um eine Änderung des Signals zu erhalten wird die Pulsweite  $n_{ON}$  verändert. Für die Ansteuerung des Motors, werden die Schaltsignale der PWM direkt vom Mikrocontroller erzeugt.(vgl. Kapitel 2.2.5.3) Das Verhalten des Motor ist abhängig von der Periodendauer sowie vom Tastverhältnis. Die PWM wird erzeugt, indem ein Ausgangssignal periodisch eingeschaltet und nach einer festgelegten Zeit wieder ausgeschaltet wird. Maßgeblich ist hierbei die Periodendauer  $n_{PWM}$  und die Einschaltzeit  $n_{ON}$ . Der Anteil der Einschaltzeit an der Gesamtzeit wird als Tastverhältnis ( $X_{PWM}$ ) bezeichnet. Abbildung 3.2a zeigt ein Zeitdiagramm zur PWM-Erzeugung.

$$X_{PWM} = \frac{n_{ON}}{n_{PWM}} \quad (3.1)$$

Die Periodendauer ergibt sich aus:

$$n_{PWM} = n_{ON} + n_{OFF} \quad (3.2)$$

Der Mittelwert der erzeugten Spannung wird mit

$$U_m = U_{OFF} + (U_{ON} - U_{OFF}) \cdot \frac{n_{ON}}{n_{PWM}} \quad (3.3)$$

berechnet. Diesen Zusammenhang verdeutlicht Abbildung 3.2(b).

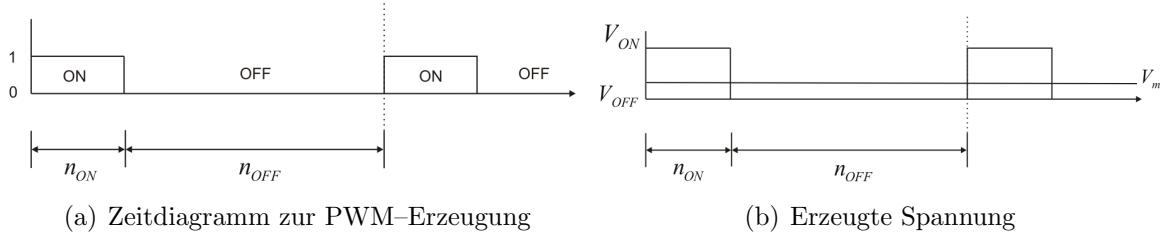


Abbildung 3.2: PWM-Erzeugung

### 3.1.1.3 Zusammenhang von PWM-Wert und Antriebsmoment

Von grundsätzlichen Interesse für jedes Antriebssystem ist der Zusammenhang zwischen der Stellgröße, dem PWM-Wert, und dem durch den Motor erzeugten Antriebsmoment. Wird ein Gleichstrommotor mit einer Gleichspannung angesteuert, so beschreiben folgende Gleichungen den Zusammenhang zwischen Spannung und Antriebsmoment.

$$U_A = U_Q + R_A \cdot I_A + L_A \cdot \dot{I}_A \quad (3.4)$$

$$U_Q = k_1 \cdot \omega \quad (3.5)$$

$$M_I = k_2 \cdot I_A \quad (3.6)$$

$$M_I = M_L + J \cdot \dot{\omega} \quad (3.7)$$

Dabei ist  $U_A$  die Ankerspannung,  $R_A$  der ohmsche Ankerwiderstand,  $L_A$  die Ankerinduktivität und  $U_Q$  die induzierte Spannung, welche über die Motorkonstante  $k_1$  direkt proportional zur Drehzahl  $\omega$  ist. Das erzeugte Moment  $M_I$  ist über die Drehmomentkonstante  $k_2$  direkt proportional zum Ankerstrom  $I_A$ . Gleichung (3.7) ist die mechanische Momentenbeziehung mit dem Lastmoment am Abtrieb  $M_L$  und der tragen Masse  $J$ , unter Vernachlässigung von Reibung.

Im stationären Fall, d.h. wenn das Antriebssystem eingeschwungen ist, ändert sich die Drehzahl nicht mehr und es treten keine Beschleunigungsmomente in Erscheinung. Der letzte Term aus (3.4) verschwindet. In diesem Fall ist das erzeugte Moment  $M_I$  konstant. Stellt man (3.5) nach  $\omega$  um und setzt man (3.4) und (3.6) in diese ein so erhält man eine Gleichung (3.8) welche die Beziehung zwischen der Ankerspannung, dem PWM-Wert, und der Drehzahl beschreibt.

$$\omega = \frac{U_A}{k_1} - \frac{R_A \cdot M_I}{k_1 \cdot k_2} \quad (3.8)$$

Die Kennlinie zeigt einen linearen Verlauf, bis auf den Bereich in welchem das erzeugte Drehmoment nicht ausreicht die Last zu bewegen. Abbildung 3.3 zeigt den idealen Verlauf der Drehzahl bei steigender Ankerspannung.

Beim *LAURON IVb* ist die Ankerspannung keine gleichförmige Spannung, sondern ihr Verhalten wird durch die Ansteuerung der Vollbrücke mit der Puls-Weiten-Modulation,

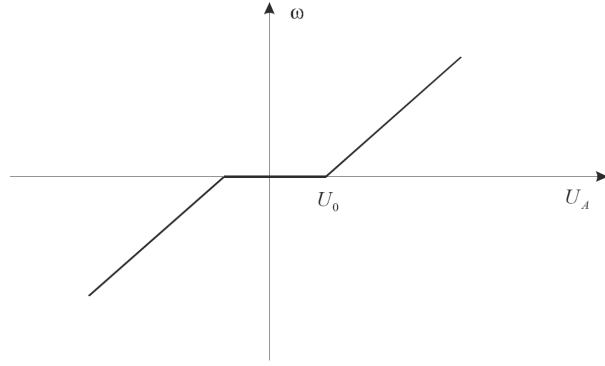


Abbildung 3.3: Ideale Ankerspannung-Drehzahl-Kennlinie aus [17]

dem erzeugten Moment und der Induktivität beeinflusst. Dadurch verschwindet der letzte Term aus der Gleichung (3.4) nicht mehr und es kann der Fall auftreten, dass der Strom während der PWM-Periode wieder den Wert Null annimmt, was als lückender Betrieb bezeichnet wird. Im Zeitintervall  $0 \leq t < t_{off}$  der PWM-Periode ist ein Transistorpaar (z.B. V1/V4 in Abbildung 3.1) aktiv und die Brückenspannung  $U_S$  liegt am Motor an und der Strom beginnt zu fließen, wie in Abbildung 3.4(a) dargestellt. Zum Zeitpunkt  $t = t_{off}$  sperren die Transistoren. Da es sich bei Motoren um ohmsche-induktive Lasten handelt, an denen der Strom nicht springen kann, fließt dieser durch die Freilaufdioden (z.B. V6/V7 in Abbildung 3.1) gegen die Spannungsquelle. Im Intervall  $t_{off} \leq t < t_{ab}$  liegt infolge dessen die negative Spannung  $U_S$  am Motor an. Zum Zeitpunkt  $t_{ab}$  wird der Strom gleich Null, das heißt, der Strom lückt. Die Dioden sperren, die Spannung bricht zusammen und der Motor ist vom Netz getrennt. Das Stromlücken führt zu dem Effekt, das beim *LAURON IVb* die PWM-Drehzahl-Kennlinie keine lineare Steigung besitzt. Abbildung 3.4(b) zeigt den Nicht-linearen Verlauf der Drehzahl. Aufgenommen wurde diese Kennlinie, indem jeweils nach einer Sekunde der PWM-Wert entsprechend um 1V erhöht wurde. Abgetastet wurde die Drehzahl mit der Frequenz von 1ms. Es wurde jeweils der Mittelwert von 100 Abtastungen in das Diagramm aufgenommen. Deutlich zusehen ist der Nicht-lineare Verlauf bei hohen PWM-Werten. Zum Aufnehmen dieser Werte wurde der Motor aus dem Teilsystem Gelenk entfernt und ohne Last, ausgenommen vom Getriebe und Zahnrad (siehe Kapitel 2.2), untersucht. Es ist anzunehmen, dass sich der Nullabschnitt der Kennlinie im eingebauten Fall vergrößert und die Kurve im oberen PWM-Bereich flacher wird.

### 3.1.1.4 Zeitverhalten des Motors

In dem vorherigen Abschnitt wurde der Zusammenhang zwischen dem PWM-Wert und der daraus resultierende Drehzahl des Motors untersucht. Ein weiterer wichtiger Aspekt, welcher für die Regelung des Systems betrachtet werden muss, ist das zeitliche Verhalten des Motors bei einer Veränderung der Ankerspannung. Die Abbildung 3.5 zeigt die Ant-

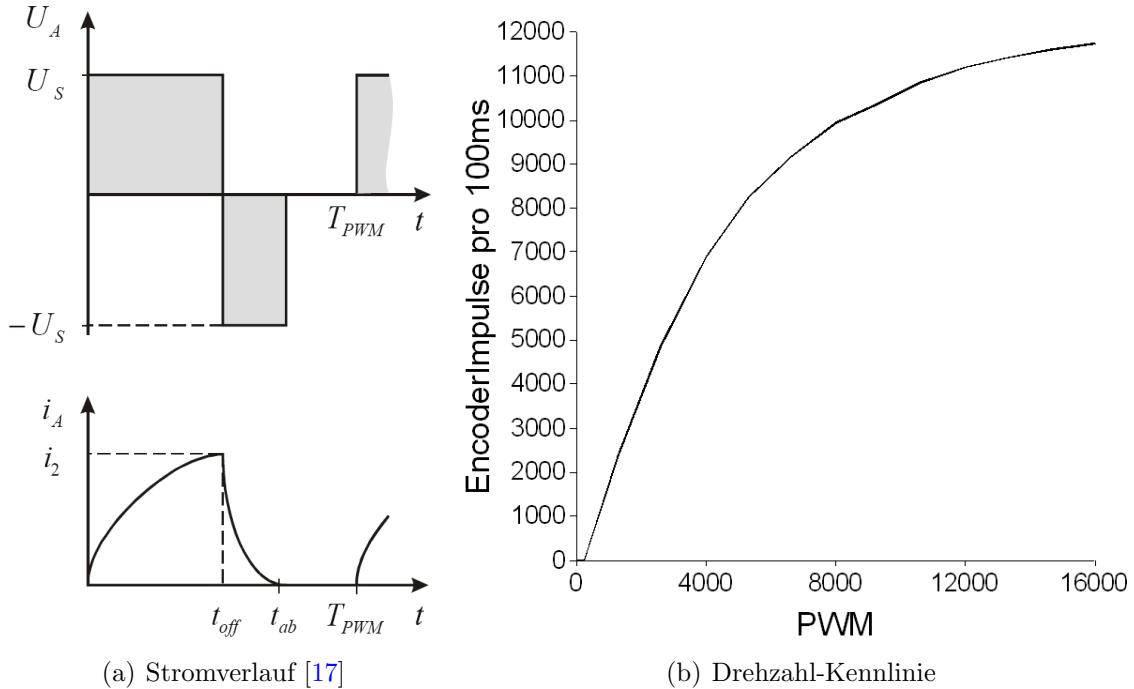


Abbildung 3.4: Verhalten Drehzahlen zu PWM beim LAURON IVb

wort des Motor auf einen Sprung des PWM-Wertes von 0 auf 16000, was 12V entspricht. Aufgenommen wurde diese Kennlinie wieder bei ausgebauten Motor und mit einer Abtastrate von 1ms. Zu sehen ist, dass der Motor ein  $PT_2$  Verhalten (siehe Kapitel 2.4) mit  $T_1 \approx 1ms$  und  $T_2 \approx 15ms$  aufweist.

### 3.1.1.5 Analyse der vorhandenen Regelung

Das System am *LAURON IVb* wird bereits mit einer Lageregelung mit einem einschleifigen Regelkreis betrieben. Es zeigt sich jedoch, dass diese Regelung nicht die geforderte Güte erreicht. Es wird zunächst überprüft, ob dieser Regelungsansatz überhaupt das Problem mit hinreichender Güte lösen kann. Eingesetzt wird bisher ein PID-Regler. Es zeigt sich jedoch, dass die Nutzung des D-Anteils, aufgrund der verwendeten Gelenkwinkelegeber, nicht möglich ist.

Denn diese haben eine Auflösung von  $0.4^\circ$ , was zur Erfassung eines Winkels hinreichend ist, nicht aber wenn das Winkel signal zwischen zwei Tastschritten differenziert werden soll. Um ein zügiges Reagieren des Reglers auf Lageabweichung sicher zu stellen, ist als Tastzeit, also die Zeit, die zwischen zwei Regelschritten vergeht und in der der Regler spätestens eine Regelabweichung erkennen kann, 5ms ein vernünftiger Wert. Um also überhaupt den kleinstmöglichen D-Teil messen zu können, muss sich das Gelenk mit einer Geschwindigkeit von  $0.4^\circ/5ms = 80^\circ/s$  bewegen. Im Normalfall wird diese Geschwindigkeit nicht erreicht werden.

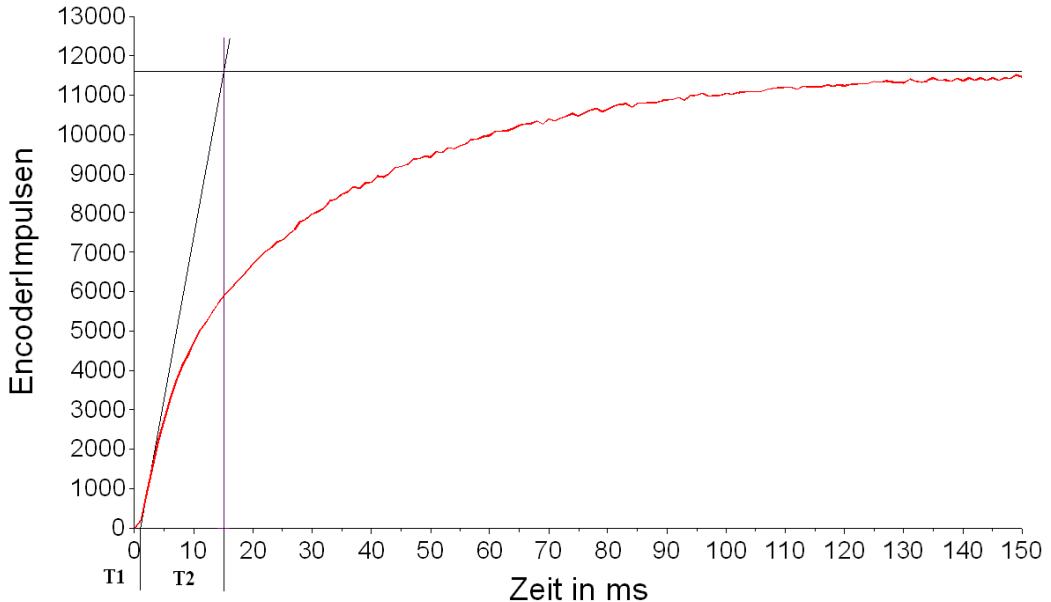


Abbildung 3.5: Zeitverhalten

Im Bezug auf die zu regelnde Größe der Winkel des Gelenkes und auf die Stellgröße, PWM-Wert bzw. Drehzahl des Motors, ergibt sich für die Regelstrecke ein  $IT_1$ -Struktur. Nach [25][26] und [17] eignet sich für eine  $IT_1$ -Strecke nur ein P oder PD - Regler. Somit bleibt für die Lageregelung mit einem einschleifigen Regelkreis nur noch ein P-Regler übrig.

Um die Lageregelung mit nur einen P-Regler zu realisieren wird eine lineares Verhalten der Regelstrecke vorausgesetzt. Es zeigt sich jedoch, wie in Abschnitt 3.1.1.3 beschrieben, dass die Regelstrecke kein lineares Verhalten aufweist. Aus der PWM-Drehzahl-Kennlinie (Abbildung 3.4(b)) ist zu erkennen, dass die Kurve mit einer „Totzone“ beginnt und außerhalb dieser zunächst einen linearer Kurvenanstieg aufweist bevor sie anschließend abflacht. Bei genauen Regelungen in welchen es kleine Differenzen zwischen Soll- und Ist-Gelenkwinkel aus zu regeln gilt, wie es besonders bei der Stützphase vorkommt, zeigt sich das der vom P-Regler generierte PWM-Wert innerhalb dieser Totzone liegt und es zu keiner Drehzahl kommt. In vorherigen Arbeiten [31] [17] wurde versucht die Regelstrecke durch ein Vorschalten einer ausgleichenden Kennlinie zu linearisieren. Um diese Linearisierung zu erreichen wird eine konstante Größe auf den vom P-Regler erzeugten PWM-Wert dazu addiert. Abbildung 3.6 verdeutlicht die vorgehensweise der Linearisierung mittels eines konstanten Offsets.

Die Totzone der PWM-Drehzahl-Kennlinie ist jedoch unter anderem abhängig von der Last, welche auf den Motor wirkt. Während des Schreitens des Roboters treten jedoch ständig Belastungsschwankungen auf. Werden beispielsweise ein oder mehrere Beine im Zuge eines Schrittes angehoben, so müssen die am Boden verbleibenden Beine dasselbe

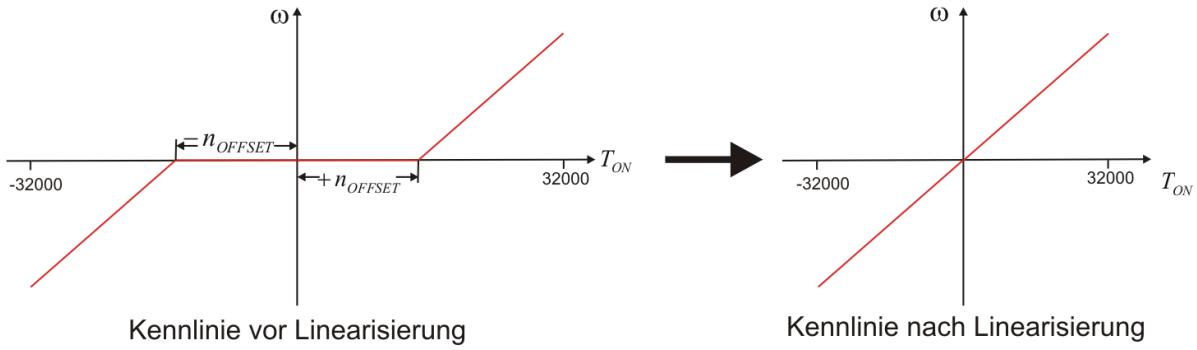


Abbildung 3.6: Kennlinienlinearisierung aus [31]

Gewicht tragen, dass zuvor von allen Beinen getragen wurden. Somit zeigt sich, dass die Linearisierung, mittels eines konstanten Offset, nur für einen bestimmten Belastungsfall gute Ergebnisse erzielt, jedoch nicht den gesamten Bereich abdeckt. Es muss deshalb ein anderer Ansatz verfolgt werden.

### 3.1.2 Gewinnung zusätzlicher Sensorinformationen

Mit einem einschleifigen Regelkreis lässt sich beim vorliegenden System keine befriedigende Lösung erreichen. Wie in 2.4 beschrieben gibt es neben dem einschleifigen Regelkreis noch andere Regelungsmodelle, wie zum Beispiel eine Kaskadenregelung, welche das dynamische Verhalten eines lagegeregelten Systems entscheidend verbessert. Dies setzt jedoch voraus, dass neben der Regelgröße, dem Winkel, weiter Hilfsgrößen gemessen werden können. Wie in 2.2 beschrieben besitzt das Bein, neben der Erfassung des Gelenkwinkels mittels einen Inkrementalgebers, noch weitere Sensoren. Welche Hilfsgrößen verwendet werden und wie sie gewonnen werden, wird in diesem Abschnitt behandelt.

#### 3.1.2.1 Ankerstrom

Auf der Treiberplatine (vgl. Abschnitt 2.2.5.3) wird der Spannungsabfall über 2 Widerständen abgegriffen und dem Analog-to-Digital Converter (ADC) des DSP zugeführt. Durch das Abgreifen zweier Spannungen können Aussagen über die Richtung des Stromflusses gemacht werden. Die Spannungswerte stehen anschließend digital zur weiteren Verarbeitung bereit. Aufgrund von fehlenden Unterlagen im Zeitraum dieser Diplomarbeit, kann keine Aussage gemacht werden an welcher Stelle die Spannungsabfälle erfasst werden, noch wie ihre Werte zu interpretieren sind. Es sollte dennoch an dieser Stelle erwähnt sein, dass beim *LAURON IVb* die Möglichkeit besteht die Ankerströme zu messen und darauf aufbauend eine Stromregelung möglich wäre.

### 3.1.2.2 Winkelgeschwindigkeit

Die Winkelgeschwindigkeit kann entweder über den Gelenkwinkelgeber oder über den auf der Motorwelle angebrachten Geber ermittelt werden.

#### Messung der Winkelgeschwindigkeit

Beide Geber erzeugen durch Drehung, bei Gelenkwinkelgeber durch Drehung des Gelenkes oder durch Drehung der Motorwelle beim dort angebrachten Impulsgeber, Impulse in Abhängigkeit der Weite der Drehung. Der Geber für die Gelenkwinkel liefert 1000 Impulse pro Umdrehung, der Impulsgeber an der Motorwelle 128 pro Umdrehung.

Die Winkelgeschwindigkeit kann nun entweder durch Messung von Frequenz oder Periodendauer der Impulse ermittelt werden. Bei der Frequenzmessung werden die Impulse, die in einem definiertem Zeitintervall auftreten gezählt. Wird die Geschwindigkeit langsamer so wird die Zahl der gemessenen Impulse entsprechend kleiner, wodurch die relative Genauigkeit sinkt und Geschwindigkeitsänderungen nicht mehr detektiert werden können. Deshalb eignet sich die Frequenzmessung nur für hohe bis mittlere Geschwindigkeiten.

Sollen mit hinreichender Genauigkeit geringere Geschwindigkeiten erfasst werden, wird aus diesem Grund die Periodendauermessung verwendet, bei der nun keine Impulse mehr gezählt werden, sondern nur die Zeit  $\Delta T$  zwischen zwei Impulsen gemessen wird. Die gesuchte Geschwindigkeit ist dann  $\approx \frac{1}{\Delta T}$ . In der Anwendung ist ihre Größe jedoch nach unten beschränkt. Wird nämlich  $\Delta T$  größer als die Abtastperiode  $T_S$ , so wird mit dem gleichen Wert in zwei Abtastschritten gearbeitet. Geschwindigkeitssveränderungen können so nicht mehr erfasst werden. Aus diesem Grund wird eine maximale erlaubte Abtastperiode  $\Delta T$  vorgegeben und damit auch die kleinste messbare Geschwindigkeit festgelegt.

Oft genügt es sich auf Frequenz- oder Periodendauermessung festzulegen. Wird jedoch das gesamte messbare Geschwindigkeitsspektrum benötigt, so müssen hybride Verfahren angewandt werden. Das einfachste Verfahren ist das Umschalten zwischen beiden Techniken. In [19] und [29] wird ein kombiniertes Verfahren vorgestellt, mit welchem man die mittlere Drehzahl ermitteln kann. Dieses Verfahren soll hier kurz vorgestellt werden.

Für das Verfahren wird zu der Impulsquelle des Gebers noch eine zusätzliche Impulsquelle mit einer fest vorgegebenen Frequenz  $f_c$  benötigt. In einem Abtastintervall  $T_A$  der Regelung wird die Impulszahl vom inkrementalen Geber  $N_1$  und die Impulszahl  $N_3$  der Referenzfrequenz gezählt. Außerdem werden die Impulse  $N_4$  der Referenzfrequenz zwischen dem letzten Impuls des Gebers und dem Anfang des neuen Abtastintervalls gezählt, siehe Abbildung 3.7.

Mit diesen Informationen kann der Drehwinkel bzw. die Lage der Motorwelle berechnet

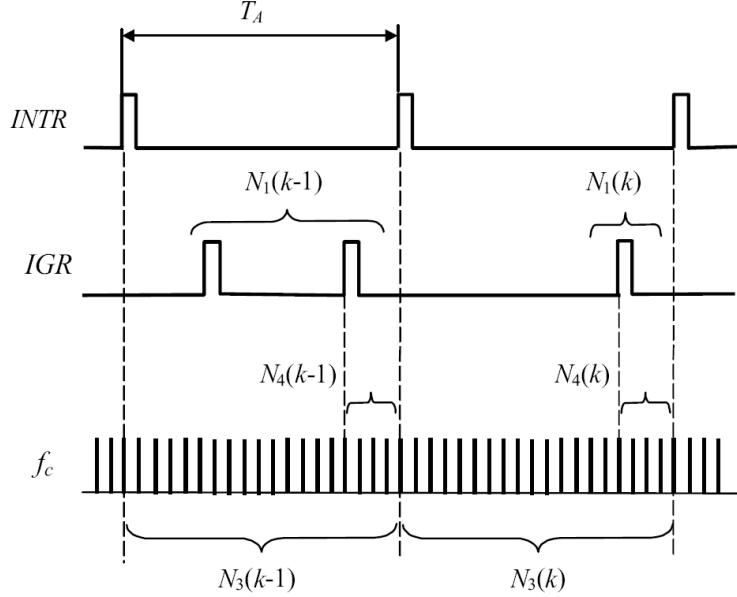


Abbildung 3.7: Drehzahlmessmethode aus [29]

werden.

$$\varphi(k) = \frac{2\pi N_1(k)}{N_Z} \quad (3.9)$$

Mit  $N_Z$  - Impulszahl des inkrementalen Gebers pro Umdrehung.

Weiterhin kann die Zeit, welche die Motorwelle für diese Umdrehung gebraucht hat, ermittelt werden durch:

$$T_\varphi(k) = \frac{N_3(k) - (N_4(k) - N_4(k-1))}{f_c} \quad (3.10)$$

Aus (3.9) und (3.10) kann der Zuwachs des Drehwinkels pro Zeitintervall, die mittlere Drehzahl, berechnet werden:

$$\Omega(k) = \frac{\varphi(k)}{T_\varphi(k)} = \frac{2\pi f_c N_1(k)}{N_Z \cdot (N_3(k) + N_4(k-1) - N_4(k))} \quad (3.11)$$

Durch diesen Algorithmus wird die mittlere Drehzahl bei höheren Drehzahlen aufgrund der Impulszahl des Gebers ( $N_1 > 1$ ) und der entsprechenden Messzeit  $T_\varphi$  berechnet. Bei

niedrigen Drehzahlen wird die Messzeit, und somit  $N_3$ , soweit erhöht bis der nächste Impuls erschienen ( $N_1 = 1$ ) ist. Die mittlere Drehzahl wird dann durch die gemessene Zeit  $T_\varphi$  zwischen zwei Geberimpulse berechnet. Hier ist ebenfalls die Zeit  $T_\varphi$  nach oben begrenzt, wodurch wieder eine kleinste messbare Geschwindigkeit festgelegt wird. Gleichung (3.11) kombiniert folglich die zwei oben erwähnten Drehzahlmessverfahren (Frequenz- und Periodendauermessung) in einem Ausdruck.

#### Auswahl des Gebers

Wie Eingangs erwähnt, eignen sich beide Sensoren vom Prinzip zur Messung der Winkelgeschwindigkeit. Für den Gelenkwinkelgeber würde sich durch seine geringe Auflösung von  $0.4^\circ$  /Impuls, nur die Periodendauermessung anbieten. Eine Vernünftige Abtastzeit für die Geschwindigkeitsregelung sollte nicht größer als 5ms betragen. Damit würde sich für den Gelenkwinkelgeber als Winkelgeschwindigkeitsmesser eine kleinste messbare Geschwindigkeit von  $1000 \frac{\text{Impulse}}{\text{Umdrehung}} \cdot 5\text{ms} = 5 \frac{s}{\text{Umdrehung}}$  ergeben. Kleinere Geschwindigkeiten könnten nicht mehr erfasst werden, was für eine Geschwindigkeitsregelung nicht geeignet ist.

Es empfiehlt sich daher, durch seine hohe Übersetzung, der Impulsgenerator auf der Motorwelle. Wegen seiner hohen Auflösung, eignet er sich hierfür besser die Frequenzmessung bzw. das kombinierte Verfahren. Auch hier ergibt sich wieder eine Mindestgeschwindigkeit von  $128 \frac{\text{Impulse}}{\text{Umdrehung}} \cdot 5\text{ms} \cdot 134_{\text{Getriebe}} \cdot 5_{\text{Gelenk}} = 428,8 \frac{s}{\text{Umdrehung}}$ . Was eine um den Faktor  $\approx 86$  bessere Auflösung als mit dem Gelenkwinkelgeber ergibt.

#### 3.1.3 Entwurf des Regelungskonzeptes

Nachdem die Betrachtung in 3.1.1 zeigte, dass mit einem einschleifigen Regelkreis das Lageregelungsproblem nicht in hinreichender Güte gelöst werden kann, muss nun ein Regelkonzept gefunden werden, das die speziellen Eigenschaften des vorliegenden Systems, also die Ansteuerung der Motoren, besonderst berücksichtigt. In der Literatur [9][26][25][2][30] gilt die mehrschleifige Kaskadenregelung als das effektivste Verfahren für die Regelung elektrischer Antriebe. Dabei wird einer schnellen inneren Drehmoment-Regelung (mit Begrenzungsmöglichkeit) je nach Bedarf eine Beschleunigungs-, Drehzahl- und Lageregelkreis überlagert.

Die inneren Regelkreise arbeiten jeweils am Schnellsten. So muss z.B. bei der Winkelregelung vorausgesetzt werden, dass der unterlagerte Drehzahlregler imstande ist, die ihm vorgegebene Führungsgröße schnell einzustellen. Der Reglerentwurf erfolgt von Innen nach Außen, wobei in jeder Stufe der letzte geschlossene Kreis durch ein vereinfachtes Modell approximiert werden kann. Der Entwurf vereinfacht sich durch dieses iterative Vorgehen, da immer nur ein Teil des Systems betrachtet wird. Analog wird auch bei der Realisierung vorgegangen. Die Regelkreise werden nacheinander in Betrieb genommen,

getestet und gegebenenfalls optimiert.

### 3.1.3.1 Stromregelkreis

Anstelle des Drehmomentes wird der Ankerstrom in der innersten Schleife geregelt. Dies ist zulässig, da nach Gleichung (3.6) das Drehmoment dem Ankerstrom proportional ist. Der Weg vom Drehmoment über die Beschleunigung und der Drehzahl zum Winkel entspricht der natürlichen Folge in dem System, da das Moment die eigentliche Eingangsgröße der mechanischen Bewegung darstellt, während Drehzahl und Winkel durch Integration auseinander hervorgehen.

An dieser Stelle sei darauf hingewiesen, dass der Stromregelkreis im Laufe der Diplomarbeit nicht realisiert wurde, Aufgrund der in Abschnitt 3.1.2.1 erwähnten fehlenden Unterlagen. Dem System steht aber im Prinzip die Möglichkeit offen mit den vorhandenen Motorstromsensoren der nun folgenden Beschreibung des Stromregelkreis zu realisieren, um eventuell eine bessere Regelung zu erhalten.

Stellgröße der Regelstrecke ist die Spannung, die sich über das PWM-Signal an dem Anker des Motors ergibt. Aufgrund dieser Spannung stellt sich ein entsprechender Ankerstrom ein. Der Ankerstrom  $I_A$  ist die Führungsgröße die es zu regeln gilt. Als Störgrößen treten die Unterschiedlichen Lasten des Motors, so wie die in 3.1.1.3 erwähnten Stromlücken auf. Abbildung 3.8 zeigt den Ankerstromregelkreis.

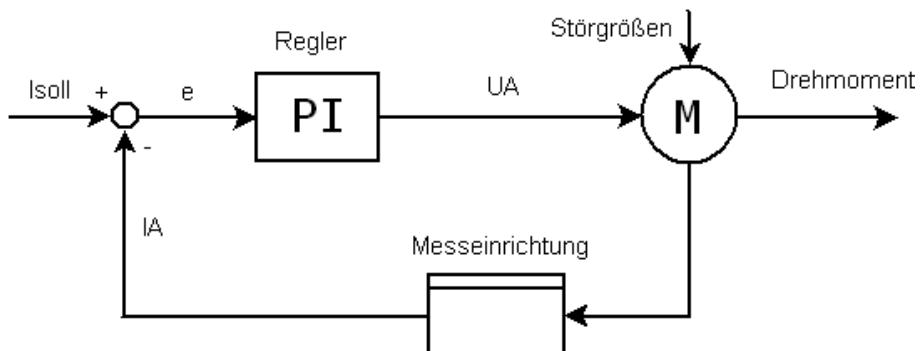


Abbildung 3.8: Modell des Ankerstromregelkreis

Aufgabe des Reglers ist hier, die Regelstrecke zu linearisieren. Dies ist mittels eines einfachen integrierende Reglers, z.B. eines PI-Reglers, erreichbar. Da das dynamische Verhalten des gesamten Antriebssystem direkt von der Reaktionsgeschwindigkeit dieses Regelkreises abhängt, empfiehlt [29] diesen Kreis nicht durch einen gewöhnlichen PI-Regler zu realisieren, sondern durch einen Kompensationsregler mit endlicher Einstellzeit. Solche Regler werden auch Dead-Beat-Regler genannt und garantieren die Einstellung der Regelgröße im Fall eines Führungsgrößensprungs nach  $n$  Abtastschritten. An dieser Stelle wird nicht weiter auf die Theorie des Dead-Beat-Reglers eingegangen, da dieser im Rahmen der Diplomarbeit nicht umgesetzt wurde.

### 3.1.3.2 Geschwindigkeitsregelkreis

Im nächsten Schritt wird die der Stromregelung überlagerte Geschwindigkeitsregelung entworfen. Die Leistungsfähigkeit der unterlagerten Geschwindigkeitsregelung ist entscheidend für die Güte der überlagerten Lageregelung. Der Geschwindigkeits-, so wie auch der Lageregler, muss gegenüber Änderungen der Streckenparameter robust sein, da sich je nach Position der Gelenke des Beins und je nach Phase in der sich das Bein befindet, die Last des Motors und somit die Zeitkonstanten verändern. Die Änderungen der Last werden als Störgröße mit in den Entwurf aufgenommen. Führungsgröße des Reglers stellt die gewünschte Drehzahl des Motors, in Impulse/ms, dar. Bei einem Geschwindigkeitsregler mit unterlagerten Stromregelung ist die Stellgröße des Reglers der, durch den Anker des Motors fließende, Ankerstrom. Da im Rahmen der Diplomarbeit auf eine unterlagerte Stromregelung verzichtet wurde, ist die Stellgröße die Ankerspannung, welche durch einen PWM-Wert erzeugt wird. Der Regelkreis soll, wie jede unterlagerte Regelung, dem Führungssignal möglichst schnell folgen. Wie in Abschnitt 3.1.1.4 beschrieben lässt sich der Motor als eine  $PT_2$  Strecke, bzw. bei Vernachlässigung von  $T_1$ , da die Bedingung  $T_1 \ll T_2$  gegeben ist, als eine  $PT_1$  Strecke mit  $K_P \approx 0,7$  und  $T_1 \approx 0,015$  modellieren. Nach [26][25][27] und [2] eignet sich für so eine Regelstrecke ein P- bzw. PI-Regler am besten. Abbildung 3.9 zeigt das Modell des Geschwindigkeitsregelkreis mit einem PI-Regler.

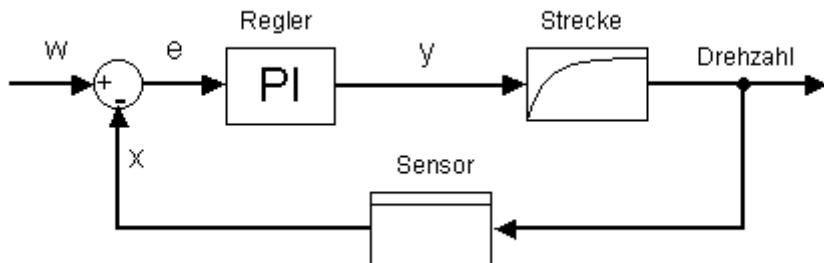


Abbildung 3.9: Modell des Geschwindigkeitsregelkreis aus [27]

### 3.1.3.3 Lageregelkreis

Im letzten Schritt kann nun die Lageregelung entworfen werden. Die Lage bzw. der Gelenkwinkel ist die eigentliche Regelgröße des Systems. Die Aufgabe der Lageregelung ist nun eine vorgegebene Sollposition des Gelenkes anzufahren und bei Störungen zu halten. Führungsgröße des Regelkreises ist der anzufahrende Gelenkwinkel in der Einheit Impulszählerstand. Der aktuelle Istwert wird über den Zähler des Inkrementalgeber am Gelenk bereit gestellt. Hierbei ist zu beachten, dass der Inkrementalgeber nur eine Auflösung von  $0,4^\circ$  pro Impuls besitzt und man somit nicht von einem, im regelungstechnischen

Sinne, stetigen Signal ausgehen kann. Man spricht in solch einem Fall von einer stetig ähnlichen Regelung. Im Bezug auf die Führungsgröße, den zu erreichenden Winkel, und der Stellgröße, die Drehzahl, welche an den unterlagerten Geschwindigkeitsregelkreis weiter gegeben wird, lassen sich die unterlagerten Regelkreise zu einem Block mit einem  $IT_1$  Verhalten zusammenfassen. Für solch eine Regelstrecke eignet sich am besten nach [17][21][26][25][2] ein einfacher Regler mit P-Verhalten.

Somit ergibt sich für die Regelung der Beinantriebe das Regelkonzept durch ein Kaskadenregelung. Damit sind drei ineinander verschachtelte Regelkreise für Motorstrom, Soll-Geschwindigkeit und Soll-Position gemeint. Die Differenz aus Soll-Position und aktueller Position ergibt also die Soll-Geschwindigkeit, die Differenz der Soll-Geschwindigkeit und aktueller Position ergibt den Sollwert für den Strom, der im innersten Regelkreis geregelt wird, oder direkt die Ankerspannung am Motor bei Weglassen der Stromregelung. Abbildung 3.10 zeigt das Gesamtmodell für das zu regelnde System.

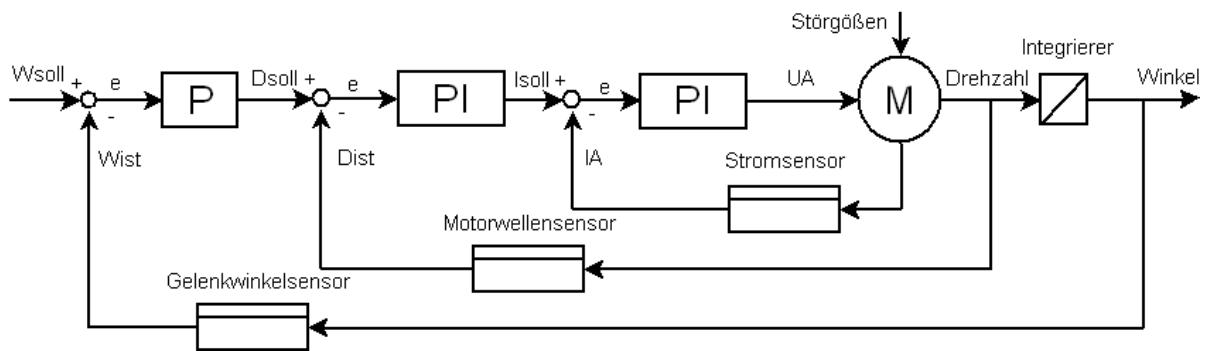


Abbildung 3.10: Modell der Einzelgelenkregelung

### 3.1.3.4 Tastzeiten der Regelkreise

Nachdem die Regelungsstruktur feststeht, ist die Abtastzeit festzulegen, mit der die einzelnen unterlagerten Regler ausgeführt werden. Bei digitalen Entwurfsverfahren [8] ergibt die Abtastzeit sich meist direkt aus den Anforderungen heraus. Da hier der Entwurf nach den Methoden der kontinuierlichen Regelungstheorie vorgenommen wurde, ist es leicht dem Irrtum zu unterliegen, die Tastfrequenz so hoch wie möglich zu wählen, sei ein guter Weg. Grenzen scheinen zunächst nur durch die Verarbeitungsgeschwindigkeit der digitalen Regeleinrichtung gesetzt. Bei der reellen digitalen Regeleinrichtung wird es dann besonders bei differenzierenden und integrierenden Reglern zu Schwierigkeiten kommen. In der Praxis wird dann ein gutes Ergebnis zu erzielen sein, wenn die Tastzeit nicht kleiner als nötig gewählt wird. Dazu wird die kleinste für das Systemverhalten bedeutsame Streckenzeitzkonstante auf die der Regler wirken soll herangezogen. Die Tastzeit muss ein Vielfaches, möglichst eine Größenordnung, kleiner sein.

Zum Entwurf eines überlagerten Regelkreises wird der unterlagerte Regelkreis zu einem

Block zusammengefasst. Die Tastzeit des überlagerten Regelkreises wird nun anhand der Zeitkonstanten des inneren Kreises und der weiteren meist größeren Verzögerungen der offenen Kette bestimmt. Es wird deshalb oft auch der Regler einer inneren Schleife mit entsprechend höherer oder gleicher Frequenz arbeiten, als der einer Äußeren.

Dem Geschwindigkeitsregelkreis liegt die mechanische Anlaufzeitkonstante des Motors zugrunde. Sie beträgt nach [4] 8ms, ist aber durch Trägheit des direkt angeschlossenen Getriebes und durch das Stromlücken deutlich höher. Wie man in Abbildung 3.5 erkennen kann, beträgt sie, beim ausgebauten Motor nur mit Getriebe und Impulsgeber,  $\approx 15\text{ms}$ . Nach [2] sollte die Abtastzeit  $T$  bei einem aperiodischen Fall (siehe Abb. 3.11)  $T \leq 0.1 \cdot T_{ap}$  betragen.

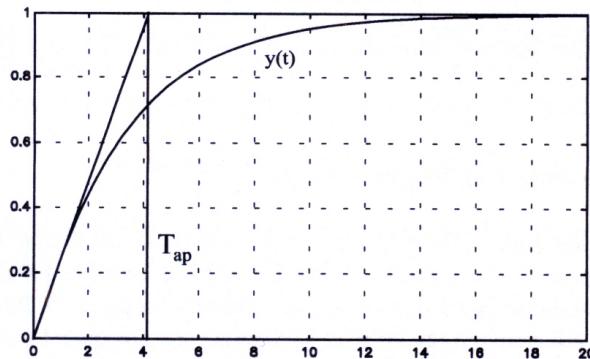


Abbildung 3.11: Abtastzeit im aperiodischen Fall aus [2]

Die Tastzeit wird aber durch die Geschwindigkeitsmessung selbst noch begrenzt. Wird die Abtastzeit zu klein, verschlechtert sich entsprechend 3.1.2.2 die Auflösung des Messwertes. Es ergeben sich störende Quantisierungseffekte. Die Festlegung der Abtastzeit für die Geschwindigkeitsregelung von 2ms stellt hier somit einen Kompromiss dar.

Für die Abtastzeit des Lageregelkreises ist die Zeitkonstante des unterlagerten Regelkreises ausschlaggebend. Auch hier empfiehlt [2] eine Tastzeit  $T$  von  $T \leq 0.1 \cdot T_p$ .

Bezogen auf die Geschwindigkeitsregelung ergibt sich für die Tastzeit ein Wert von 4ms.

### 3.1.3.5 Einstellen der Regelparameter

Für das korrekte Funktionieren der Regelkreise müssen die Parameter der Regler entsprechenden der Natur der Regelstrecke angepasst und eingestellt werden. Es wurden verschiedene Methoden zum Einstellen dieser Parameter entwickelt. Dabei empfiehlt sich nach [24] am Besten das Betragsoptimumverfahren für das Einstellen der Regelparameter der Spannungsregelungen in kaskadierten Regelstrukturen bei elektrischen Antrieben. Da der nach dem Betragsoptimum eingestellte Regelkreis eine besonders gute Dynamik besitzt und nur wenig Überschwingen zulässt [24].

### Betragsoptimierungsverfahren für den Geschwindigkeitsregelkreis

Die Regelstrecke  $G_S(s)$  und der Regler  $G_R(s)$  seien für die folgende Erläuterung durch

$$G_S(s) = \frac{K_S}{(1+sT_1)(1+sT_2)} \quad (3.12)$$

$$G_R = K_P \frac{1+sT_n}{T_n s} \quad (3.13)$$

beschrieben, wobei  $T_2$  die größere der beiden Streckenzeitzkonstanten ist, d.h.  $T_2 > T_1$ . Bei der Regelstrecke  $PT_2$  handelt es sich also offenbar um ein nicht schwingungsfähiges System zweiter Ordnung. Um die Ordnung des gesamten Systems möglichst niedrig zu halten, wird zunächst mit der Wahl der Nachstellzeitkonstante  $T_n$  des Reglers die langsamere Polstelle (größere Zeitkonstante) der Strecke kompensiert. Damit gilt:

$$T_n = T_2 \quad (3.14)$$

Die Übertragungsfunktion des resultierenden geschlossenen Regelkreises ist damit:

$$T(s) = \frac{K_S K_P}{T_2 T_1 s^2 + T_2 s + K_S K_P} \quad (3.15)$$

mit dem dazugehörigen Frequenzgang

$$F_T(j\omega) = \frac{K_S K_P}{(K_S K_P - T_2 T_1 \omega^2) + j\omega T_1} \quad (3.16)$$

beziehungsweise

$$|F_T(j\omega)| = \frac{K_S K_P}{\sqrt{(K_S K_P - T_2 T_1 \omega^2)^2 + (j\omega T_1)^2}} \quad (3.17)$$

$$|F_T(j\omega)| = \frac{K_S K_P}{\sqrt{T_2^2 T_1^2 \omega^4 + [T_2^2 - 2K_S K_P T_2 T_1] \omega^2 + (K_S K_P)^2}} \quad (3.18)$$

Ein optimales Führungsverhalten ist gegeben, wenn der Betrag des Frequenzganges für alle Werte von  $\omega$  den Wert 1 hätte:  $|F_T(j\omega)| \equiv 1$  (d.h. Führungs- und Regelgröße stimmen für alle Frequenzen im Betrag und Phase überein).

### 3 Neue Regelungskonzepte für Beinantriebe und Kraft

Dies ist aber praktisch nicht zu verwirklichen. Ein Grundgedanke, diesem Ideal möglichst nahe zu kommen, ist es nun, den Betrag des Frequenzganges ausgehend von  $\omega = 0$ , bis auf möglichst hohe Werte von  $\omega$  auf den Wert eins zu halten. Um diese Forderung zu erfüllen, müssen möglichst viele Ableitungen

$$\frac{d^n}{d\omega^n} |F_T(j\omega)| \text{ mit } n = 1, 2, \dots \quad (3.19)$$

zu Null gemacht werden. Führt man die Differentiation nach  $\omega$  der Einfachheit halber nicht an  $|F_T(j\omega)|$  selbst, sondern am Quadrat des Kehrwerts durch, so erhält man nach kurzer Rechnung das Ergebnis, das im Ausdruck  $|F_T(j\omega)|$  der Koeffizient bei  $\omega^2$  verschwinden muss. Damit ergibt sich unmittelbar die Einstellvorschrift für den Verstärkungsfaktor des PI-Reglers zu

$$K_P = \frac{1}{2} \cdot \frac{T_2}{T_1 K_S}. \quad (3.20)$$

Damit hat man die Einstellung des Regelkreises nach dem Betragsoptimum. Hierbei beträgt die Dämpfung den Wert  $d = \frac{1}{\sqrt{2}} \approx 0,7$

Im Vergleich zu anderen Optimierungsverfahren stellt die Betragsoptimierung eine Kompromisslösung zwischen dem zu stark gedämpften linearen Optimum und dem zu schwach gedämpften quadratischen Optimum dar.

#### Einstellverfahren für den Lageregelkreis

Für die Lageregelung wird ein P-Regler eingesetzt (vgl. Abschnitt 3.1.3.3). Zur Berechnung des Regelparameter lässt sich folgende Überlegung anstellen [17].

Um ein stabiles System zu erhalten, müssen die allgemeinen Stabilitätsbedingungen für das geschlossene System

$$G_0 + 1 = 0 \quad (3.21)$$

$$G_R \cdot G_S + 1 = 0 \quad (3.22)$$

erfüllt werden [9]. Hierbei ist die  $G_0$  Übertragungsfunktion der offenen Kette,  $G_R$  die Übertragungsfunktion des Reglers und  $G_S$  die Streckenübertragungsfunktion.

Die Regelstrecke für die Lageregelung hat eine  $IT_1$ -Struktur und kann durch die Gleichung

$$G_S = \frac{1}{T_I s \cdot (1 + T_1)} \quad (3.23)$$

### 3 Neue Regelungskonzepte für Beinantriebe und Kraft

mit  $T_I$  als Integrationszeitkonstante und  $T_1$  als Zeitkonstante beschrieben werden.  
Zusammen mit der Gleichung für den Regler

$$G_R = K_R \quad (3.24)$$

wobei  $K_R$  die Übertragungskonstante des Reglers ist, ergibt sich für die geschlossene Kette folgendes Übertragungsverhalten

$$\frac{K_R}{T_I s \cdot (1 + T_1 s)} + 1 = 0. \quad (3.25)$$

Durch die beiden Zeitkonstanten im Nenner ergibt sich ein schwingungsfähiges System

$$\frac{T_I T_1}{K_R} s^2 + \frac{T_I}{K_R} s + 1 = 0. \quad (3.26)$$

Bei dem Koeffizientenvergleich mit der Schwingungsgleichung

$$T_0^2 s^2 + 2DT_0 s + 1 = 0 \quad (3.27)$$

erhält man

$$T_0^2 = \frac{T_I T_1}{K_R} \quad (3.28)$$

$$2DT_0 = \frac{T_I}{K_R} \quad (3.29)$$

Durch Auflösen nach  $T_0$  und anschließendes Quadrieren von (3.29) lässt sich diese mit (3.28) gleichsetzen und nach  $K_R$  umstellen. Somit erhält man eine Berechnungsvorschrift für  $K_R$

$$K_R = \frac{T_I}{4D^2 T_1} \quad (3.30)$$

Für einen Dämpfungswert von  $D \approx 0,7$  erhält man eine Überschwingweite von 2%.

### 3.1.4 Digitaler PI-Stellungsalgorithmus

Beim Entwurf des Regelkonzeptes wie im Abschnitt 3.1.3.2 beschrieben, kommt für die Drehzahl- bzw. Geschwindigkeitsregelung ein PI-Regler zum Einsatz. Dieser Abschnitt befasst sich daher mit der Entwicklung einen digitalen PI- Regelalgorithmus.

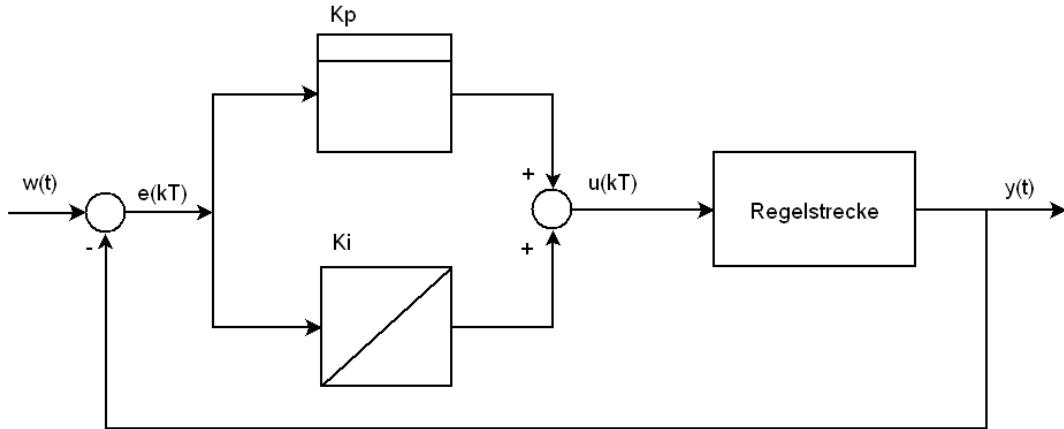


Abbildung 3.12: Wirkungsplan PI-Regler im geschlossenen Kreis

Abbildung 3.12 zeigt den PI-Regler Wirkungsplan des geschlossenen Kreises, wobei folgende Bezeichnungen gelten:

- $w(t)$  Der Führungsgröße/Sollwert z.B. Drehzahl.
- $y(t)$  aktueller Wert/Istwert
- $e(kT)$  Abweichung zwischen Ist- und Sollwert ( $e(t) = w(t) - y(t)$ ).
- $u(kT)$  Die Stellgröße des Reglers z.B. Spannung am Motor/PWM-Signal

Für einen zeitkontinuierlichen Regler (vgl. Kapitel 2.4) lautet die Reglergleichung:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_n} \int_0^t e(t) dt \right] \quad (3.31)$$

Dadurch das der Regler mit Hilfe eines Microcontrollers realisiert werden soll, und Microcontroller stets eine gewisse Abtastzeit der Signale benötigen, muss die Gleichung die Abtastzeit berücksichtigen. Die Gleichung kann mit folgender Näherung, die für kleine Abtastzeiten  $T$  gelten, diskretisiert werden [2].

Das Integral  $\int_T e(t) dt$  kann durch eine Trapezfläche (3.13) angenähert werden.

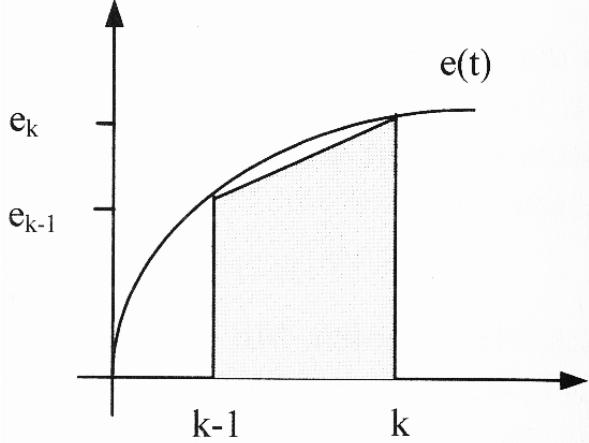


Abbildung 3.13: Integration durch Bildung der Trapezfläche aus [26]

Die Trapezregel lautet

$$\int_T e(t) dt \approx T \cdot \frac{e_k + e_{k-1}}{2} \quad (3.32)$$

aus (3.31) mit (3.32) ergibt sich die genäherte Differenzgleichung des PI-Reglers:

$$u(kT) = K_p \left[ e(kT) + \frac{T}{T_n} \cdot \frac{e_{kT} + e_{[(k-1)T]}}{2} \right] \quad (3.33)$$

Diese Gleichung ist für die Realisierung mit einem Microcontroller ungeeignet, da es sich hier nicht um einen rekursiven Algorithmus handelt. Doch durch bilden von

$$u([(k-1)T]) = K_p \left[ e[(k-1)T] + \frac{T}{T_n} \cdot \frac{e_{[(k-1)T]} + e_{[(k-2)T]}}{2} \right] \quad (3.34)$$

und durch die Differenzbildung von  $u(kT) - u([(k-1)T])$  erhält man einen rekursiven Algorithmus.

$$u(kT) - u([(k-1)T]) = K_p e_{kT} - K_p e_{[(k-1)T]} + \frac{K_p}{T_n} \frac{T}{2} (e_{kT} + e_{[(k-1)T]}) \quad (3.35)$$

$$u(kT) = u([(k-1)T]) + K_p \left( \frac{T}{T_n} \frac{2}{2} + 1 \right) e_{kT} + K_p \left( \frac{T}{T_n} \frac{2}{2} - 1 \right) e_{[(k-1)T]} \quad (3.36)$$

Der Regelalgorithmus (3.36) wird Stellungsalgorithmus genannt [26][2][25][9], und kann zur rekursiven Berechnung der Stellgröße  $u(kT)$  für die gewählte Abtastzeit  $T$  genutzt werden.

## 3.2 Kraftregelung der Beine

Beim Schreiten des Schreitroboters *LAURON IVb* treten im hohen Maße Interaktionen mit der Umgebung auf. Vor allem beim Laufen auf einem unebenen Untergrund, ist es möglich das die Stützstelle mit dem Fuß bereits erreicht wird und seine Geschwindigkeit deutlich größer als Null ist. Diese Situation muss durch die aktivierende Kraftregelung in der Auftrittphase schnell detektiert werden und die kinetische Energie der Beinbewegung von dem Antrieben übernommen wird. Andernfalls kann das Bein beschädigt werden. Wird dagegen die Sollposition des Beines erreicht ohne dass ein Kontakt vorliegt, soll die aktivierende Kraftregelung eine Positionskorrektur in Richtung des Untergrundes erzeugen, um einen Kontakt mit diesem herzustellen.

Eine weitere Situation für eine Kraftregelung tritt in den Stützphasen des Roboters auf. In der Stützphase stehen mehrere Roboterbeine im Kontakt mit dem Untergrund. Das gesamte Robotergewicht wird sich generell ungleich auf die Füße des Roboters verteilen. Es soll aber durch entsprechende Drehmomente der Motoren am Gelenk ausgeglichen werden und die auftretende Querkräfte minimiert werden.

Für beide Fälle wurde bereits von Kai Wetzelsberger [31] Kraftregler, welche im anschließenden Kapitel 3.2.1 vorgestellt werden, entwickelt und erfolgreich in der obersten Steuerungsebene MCA integriert und getestet. Doch aus der Sicht der Regelung erweist es sich als vorteilhaft, die Kraftregelung in demjenigen Raum zu realisieren, in dem die physikalischen Sensoren die Informationen liefern. Dies ergibt kürzeste Regelzeiten und Totzeiten.

Dieser Abschnitt befasst sich daher mit den allgemeinen Konzepten der Kraftregelung für die Umsetzung in die Mikrocontollerebene.

### 3.2.1 Kraftgeführte Bewegung

Die zwei beschriebenen Fälle für eine Kraftregelung der Beine führt zu zwei unterschiedlichen Kraftregelungen bzw. kraftgeführten Bewegungen. Zur Verteilung des Gewichtes auf die Stützbeine und zur Minimierung von Querkräften eignet sich das Verfahren der aktiven Nachgiebigkeit<sup>1</sup> durch seine Einfachheit und durch seine Steuerung, welche rein auf kinematischer Arbeit basiert[17].

---

<sup>1</sup>englisch: Active Compliance

### Nachgiebigkeitsregler

Der Nachgiebigkeitsregler erzeugt eine proportionale Positionsänderung entsprechend einer wirkenden Kraft.

$$\Delta \vec{X} = k_{FN} \cdot (\vec{F}_{soll} - \vec{F}_{ist}) \quad (3.37)$$

mit

$$\begin{aligned}\Delta \vec{X} &= \text{Positionsänderung} \\ k_{FN} &= \text{Nachgiebigkeitsfaktor} \\ \vec{F}_{soll} &= \text{Sollkraft} \\ \vec{F}_{ist} &= \text{Aktuell wirkende Kraft}\end{aligned}$$

Die Gleichung (3.37) ist dem Federkraftgesetz gleichzusetzen und beschreibt somit das Verhalten einer mechanischen Feder. Wird der Nachgiebigkeitsfaktor  $k_{FN}$  größer so hat dies eine größere Nachgiebigkeit zur Folge.

Für den erst beschriebenen Fall, das Auftreffen auf eine Stützstelle mit einer Geschwindigkeit größer Null, eignet sich ein Dämpfungsregler<sup>2</sup> bzw. die aktive Anpassung.

### Dämpfungsregler

Der Dämpfungsregler erzeugt, im Gegensatz zum Nachgiebigkeitsregler, direkt keine Positionsänderung sondern eine Geschwindigkeit in Abhängigkeit einer wirkenden Kraft [17]. Es gilt

$$\dot{\Delta \vec{X}} = k_{F2} \cdot (\vec{F}_{soll} - \vec{F}_{ist}) \quad (3.38)$$

Diese Bezeichnung beschreibt das Verhalten eines Dämpfers, wobei größere Geschwindigkeiten durch größere wirkende Kraftdifferenzen entstehen. Dabei wird eine Bewegung in Richtung der wirkenden Kraft erzeugt. So lange die Kraft wirkt, hält auch die Bewegung an. Es entsteht eine Ausweich- oder Anpassbewegung, die deshalb auch aktive Anpassung genannt wird. Für größere Werte von  $k_{F2}$  wir die Dämpfung geringer.

---

<sup>2</sup>englisch: Damping Control

Abkürzung	Ausführliche Schreibweise
$s\alpha$	$\sin(\alpha)$
$c\alpha$	$\cos(\alpha)$
$s\beta$	$\sin(\beta)$
$c\beta$	$\cos(\beta)$
$s\gamma$	$\sin(\gamma)$
$c\gamma$	$\cos(\gamma)$
$s\theta$	$\sin(\theta)$
$c\theta$	$\cos(\theta)$
$s\beta\gamma$	$\sin(\beta + \gamma)$
$c\beta\gamma$	$\cos(\beta + \gamma)$

Tabelle 3.1: Abkürzende Schreibweisen

### 3.2.2 Krafttransformation

Für die Kraftregelung werden die Kraftsensoren des Beines benötigt um die aktuelle IST-Kraft zumessen. Die Kraftsensoren registrieren hierbei drei senkrecht aufeinander stehende Kraftkomponenten ( $F_1, F_2, F_3$ ). Die gemessenen Kräfte beziehen sich auf den Fußendpunkt, somit ist der Ort und die Richtung des Kraftvektors abhängig von der aktuellen Position des Beines im Raum. Um die Werte sinnvoll interpretieren zu können, müssen sie zuerst im Bezug auf ein gemeinsames System transformiert werden.

Die Sensoren befinden sich im D-Segment des Roboterbeines(vgl. Kapitel 2.2 Abb. 2.4(a)). Die erfassten Kraftkomponenten  $F_1, F_2, F_3$  stehen orthogonal zueinander und sind im Bezug auf die Position des Kraftsensors im Raum ausgerichtet, was der Ausrichtung des D-Segmentes gleichbedeutend ist. Die Kräfte werden im Fußpunktkoordinatensystems (*FPKS*) (vgl. Abschnitt 2.1.2.1) aufgenommen. Die räumliche Orientierung des Kraftangriffspunktes kann mit Hilfe der direkten Kinematik (Kapitel 2.5.1) in das Beinbezugskoordinatensystem (*BBKS*) transformiert werden.

Bei den Fußkräften muss man nun beachten, dass es sich nicht um Koordinaten, sondern um einen Kraftvektor handelt. Beim Übergang in ein anderes Koordinatensystem werden diese von der Translation nicht beeinflusst. Somit genügt hierbei das *FPKS* durch eine Drehung parallel zum *BBKS* zu transformieren. Gleichung (3.39) beschreibt die Rotation der Kräfte am Fußpunkt in die Kräfte im Beinbezugskoordinatensystem. Hierbei wurde übersichtshalber Abkürzungen eingeführt, welche aus der Tabelle 3.1 entnommen werden können.

$$\begin{pmatrix} F_X \\ F_Y \\ F_Z \end{pmatrix} = \begin{pmatrix} c\alpha & s\alpha \cdot s\beta\gamma & s\alpha \cdot c\beta\gamma \\ s\theta \cdot s\alpha & c\theta \cdot c\beta\gamma - s\theta \cdot c\alpha \cdot s\beta\gamma & -c\theta \cdot s\beta\gamma - s\theta \cdot c\alpha \cdot c\beta\gamma \\ -c\theta \cdot s\alpha & s\theta \cdot c\beta\gamma + c\theta \cdot c\alpha \cdot s\beta\gamma & -s\theta \cdot s\beta\gamma + c\theta \cdot c\alpha \cdot c\beta\gamma \end{pmatrix} \cdot \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} \quad (3.39)$$

### 3.2.3 Inverse Kinematik

Mit der im vorherigen Kapitel beschrieben Transformation, stehen nun die Kraftinformationen parallel zum *BBKS* zur Verfügung. Durch die Gleichungen (3.37) und (3.38) können auf Grund dieser Krafteinwirkung, eine neue Position bzw. eine neue Geschwindigkeit des Fußendpunktes, im Bezug auf das *BBKS*, berechnet werden, welche der gewünschten Kraftregelung entspricht.

Die neue Position bzw. Geschwindigkeit des Fußendpunktes müssen nun in Gelenkwinkel bzw. Gelenkwinkelgeschwindigkeiten überführt werden. Dazu muss die Inverse Kinematik benutzt werden, welche in Abschnitt 2.5.2 eingeführt wurde. Auf der *MCA* Steuerungsebene wurde bereits von Bernd Gaßmann[16] die Inverse Kinematik für die Beinsteuierung implementiert. Zu finden ist sie im Modul *KinematicLeg* in der *LegControl* Gruppe des *BehaviourParts* bzw. der *BehaviourGroup* (siehe Kapitel 2.3). Es wurden 2 verschiedene Methoden der Inversen Kinematik implementiert. Eine analytische Methode und eine geometrische.

Es zeigte sich das der geometrische Ansatz zur Berechnung der Inversen Kinematik der des analytischen Ansatzes vorzuziehen ist. Da mit ihm auf einfachster Art die Winkel auf ihr Maximum und ihr Minimum beschränkt werden können. Zudem erlaubt dieser Ansatz, dass bei nicht erreichen eines Punktes trotzdem die Möglichkeit besteht soweit wie möglich an ihn heran zu fahren [16].

Da für die Kraftregelung die Berechnung der Inversen Kinematiks des Roboterbeins essentiell ist, wird nun im folgenden Abschnitt auf den geometrischen Ansatz genauer eingegangen.

#### 3.2.3.1 Geometrischer Ansatz

In diesem Ansatz erfolgt die Berechnung mittels einiger geometrischer Überlegungen. Diese Methode arbeitet auf der Basis der  $\alpha KS$ -Koordinaten, d.h. der Fußpunkt muss zuerst in das  $\alpha KS$  transformiert werden.

$$FP_{\alpha KS} = {}^{\alpha KS} T_{BBKS} \cdot FP_{BBKS} \quad (3.40)$$

wobei  ${}^{\alpha KS} T_{BBKS}$  die Transformation vom Beinbezugskoordinatensystems in das  $\alpha$  - Gelenkkoordinatensystems darstellt.

$${}^{\alpha KS} T_{BBKS} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & L_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.41)$$

### 3 Neue Regelungskonzepte für Beinantriebe und Kraft

mit  $L_0$  konstruktionsbedingte Distanz und  $\theta$  der Neigungswinkel des  $\alpha$ -Gelenks aus der Waagrechten nach unten.

Das  $\alpha$  Gelenk dreht um die y-Achse, während das  $\beta$ - und das  $\gamma$ -Gelenk um die gleiche Ebene drehen, welche senkrecht auf der Drehachse des  $\alpha$ -Gelenks steht. Ist das  $\alpha$ -Gelenk positioniert, kann der Fußpunkt also nur noch innerhalb dieser einen von  $\beta$ - und  $\gamma$ -Gelenk aufgespannten Ebene liegen. Somit muss das  $\alpha$ -Gelenk bei einem vorgegebenen Fußpunkt so gedreht werden, dass die dadurch gebildete Ebene zwischen  $\beta$ - und  $\gamma$ -Gelenk auf dem Fußpunkt liegt. Bei einer Einstellung von 0 Grad am  $\alpha$ -Gelenk ist dies gerade die y-z-Ebene des  $\alpha KS$ . Bei der Projektion des Fußpunktes auf die x-z-Ebene bildet dieser mit der z-Achse und dem Koordinatenursprung, das  $\alpha$ -Gelenk, ein rechtwinkliges Dreieck, dessen Winkel an der Ecke im Ursprung gerade dem Betrag des gesuchten  $\alpha$ -Winkels entspricht. Unter Berücksichtigung des Vorzeichens berechnet sich dieser aus

$$\alpha = \begin{cases} \arctan\left(\frac{x}{z}\right) & , z \neq 0 \\ -\operatorname{sgn}(x) \cdot 90 & , z = 0 \wedge x \neq 0 \\ \text{beliebig} & , z = 0 \wedge x = 0 \end{cases} \quad (3.42)$$

Anschliessend kann der  $\alpha$ -Winkel auf seine Grenzen beschränkt werden durch:

$$\alpha = \min(\max(\alpha, \min \alpha), \max \alpha) \quad (3.43)$$

Mit dem so berechneten  $\alpha$ -Winkel und der Transformationsmatrix  ${}^{\beta KS}T_{\alpha KS}$  können wir zum  $\beta KS$  überwechseln.

$${}^{\beta KS}T_{\alpha KS} = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & L\alpha & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.44)$$

Das  $\beta$ - und das  $\gamma$ -Gelenk drehen jeweils um die x-Achse ihres Gelenks, die parallel zueinander liegen. Nun müssen  $\beta KS(0, 0, 0)_{\beta KS}$ ,  $\gamma KS$  und  $FPKS(x, y, z)_{\beta KS}$  ein Dreieck auf der y-z-Ebene bilden, damit der Fußpunkt erreicht wird. Zum besseren Verständnis sein die Dreiecksseiten definiert:

$$a_{\Delta} = \overline{\beta KS \gamma KS} = L_{\beta} = \text{konstat} \quad (3.45)$$

$$b_{\Delta} = \overline{\gamma KS FPKS} = L_{\gamma} = \text{konstat} \quad (3.46)$$

$$c_{\Delta} = \overline{\beta KS FPKS} = \sqrt{y^2 + z^2} \quad (3.47)$$

### 3 Neue Regelungskonzepte für Beinantriebe und Kraft

Da nur  $c_\Delta$  variable ist, kann ein Dreieck nicht für alle Fußpunkte gebildet werden. Es muss gelten:

$$a_\Delta + b_\Delta \geq c_\Delta \quad (3.48)$$

$$a_\Delta + c_\Delta \geq b_\Delta \quad (3.49)$$

$$b_\Delta + c_\Delta \geq a_\Delta \quad (3.50)$$

Daraus folgt die Einschränkung:

$$|L_\beta - L_\theta| \leq \sqrt{y^2 + z^2} \geq L_\beta + L_\gamma \quad (3.51)$$

Die Dreiecksbildung kann dann durch den Kosinussatz berechnet werden:

$$\beta_\Delta = \arccos\left(\frac{a_\Delta^2 + c_\Delta^2 - b_\Delta^2}{2 \cdot a_\Delta \cdot b_\Delta}\right) = \arccos\left(\frac{L_\beta^2 + y^2 + z^2 - L_\gamma^2}{2 \cdot L_\beta \cdot \sqrt{y^2 + z^2}}\right) \quad (3.52)$$

$$\gamma_\Delta = \arccos\left(\frac{a_\Delta^2 + c_\Delta^2 - b_\Delta^2}{2 \cdot a_\Delta \cdot b_\Delta}\right) = \arccos\left(\frac{L_\beta^2 + L_\gamma^2 - y^2 - z^2}{2 \cdot L_\beta \cdot L_\gamma}\right) \quad (3.53)$$

Beide Winkel liegen, Aufgrund der Arkuscosinus-Funktion, zwischen 0 und 180 Grad. Der gesuchte Winkel  $\beta$  setzt sich nun zusammen aus  $\beta_\Delta$  und dem Winkel der negativen z-Achse mit  $\alpha_\Delta$ , welcher durch die Arkustangens-Funktion berechnet werden kann.

$$\beta = \begin{cases} \beta_\Delta \arctan\left(\frac{y}{-z}\right) & , z < 0 \\ 180 + \beta_\Delta + \arctan\left(\frac{y}{-z}\right) & , z > 0 \\ 90 + \beta_\Delta & , z = 0 \wedge y \neq 0 \\ beliebig & , z = 0 \wedge y = 0 \end{cases} \quad (3.54)$$

$\gamma$  ist der Ergänzungswinkel von  $\gamma_\Delta$  zu 180 Grad und kann somit berechnet werden durch:

$$\gamma = \gamma_\Delta - 180 \quad (3.55)$$

Beide Winkel können anschließend wie  $\alpha$  durch

$$\beta = \min(\max(\beta, \min \beta), \max \beta) \quad (3.56)$$

$$\gamma = \min(\max(\gamma, \min \gamma), \max \gamma) \quad (3.57)$$

auf ihre Maximal- und Minimal-Grenzen beschränkt werden.

Ist die Dreiecksbildung nicht möglich so kann der Fußpunkt nicht erreicht werden. Hierbei soll jedoch beachtet werden das bei der Dreiecksbildung aus Symmetriegründen zwei Dreiecke gebildet werden können, so dass sich die Berechnung (3.52) und (3.53) auf die für das Laufen des *LAURON IVb* notwendige Konstellation beziehen.

### 3.2.4 Einordnung in die Regelstruktur

Die Kraftregelung beeinflusst direkt die Position des Beins, bzw. bei dem Dämpfungsregler eine Geschwindigkeit, welche man als Position in Abhängigkeit der Zeit auffassen kann. Somit fügt sich der Kraftregelkreis als eine weitere Überlagerung in die, im Abschnitt 3.1.3.3, konzeptionierte Regelstruktur der kaskadierenden Lageregelung (Abb. 3.10) ein. Die Abbildung 3.14 zeigt das Regelmodell der überlagerten Kraftregelung mit samt ihren Transformationen.

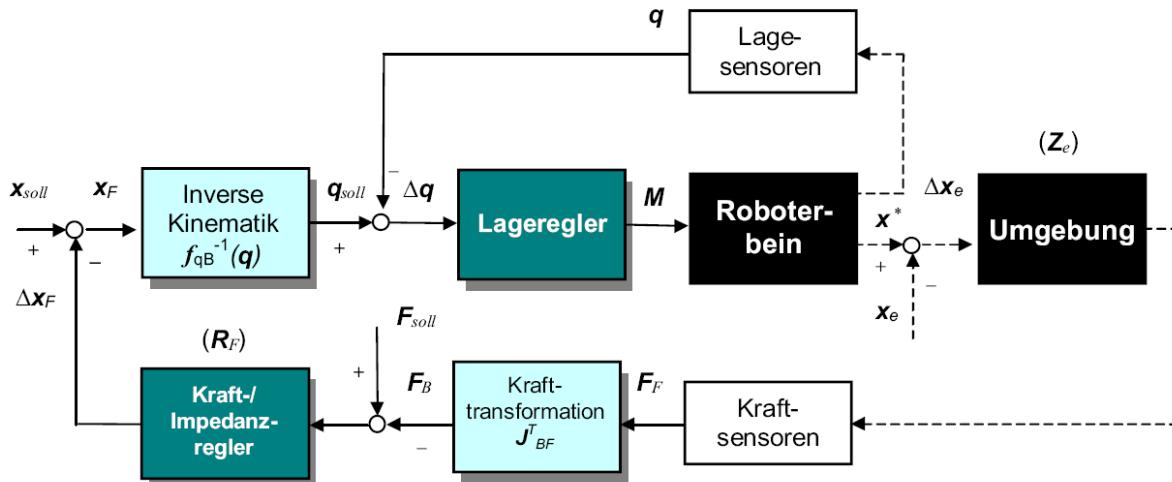


Abbildung 3.14: Modell der überlagerten Kraftregelung aus [29]

# 4 Implementierung und Integration

Im vorherigen Kapitel wurden Konzepte zur Lageregelung und zur Kraftregelung vorgestellt und beschrieben. In diesem Kapitel geht es nun um die Umsetzung der Konzepte in die DSP-Controller Steuerung. Hierbei wird zuerst die vorhandene DSP-Software untersucht, um einen geeigneten Einstiegspunkt der Regelung zu finden. Anschließend wird auf die Umsetzung der Geschwindigkeitserfassung eingegangen, um aufbauend darauf auf die Implementierung der Regler und der Regelkreise eingehen zu können.

Anschließend wird auf die Implementierung der Kraftregelung eingegangen. Besonders die Umsetzung der trigometrischen Funktionen wie Akustangens, Cosinus, Arkuscosinus und Sinus wird genauer betrachtet, da bei der DSP-Controller-Programmierung nicht auf die Standard Bibliotheksfunktionen zurückgegriffen werden kann.

## 4.1 Analyse der vorhanden Software des DSP

Die neu entworfenen Regelungen sollen in die bereits vorhandene DSP-Software integriert werden. Damit dies geschehen kann muss zunächst die vorhandene Software analysiert werden. Die Software gliedert sich in zwei Teilebereiche. Zum Einem eine Routine zum Starten der eigentlichen Applikation und zum Anderen die Hauptroutine zur Steuerung des Beines. Nachfolgend werden diese beiden Bereiche beschrieben.

### 4.1.1 Bootloader

Nach dem Power-ON des DSP-Controllers wird zunächst die Bootloader Routine aus dem BootFlash (\$8000 – \$87FF) des DSP56F803 geladen. Innerhalb dieser Routine wird die Hardware des Controllers initialisiert, sowie der CAN-BUS-Controller des DSP eingerichtet. Nachdem der CAN-BUS eingerichtet ist wartet der DSP auf eine Initialisierungsnachricht über den CAN-BUS. Diese Nachricht dient zu logischen Ordnung der 7 Controller (6 Bein- und 1 Kopfcontroller) auf dem Bus. Dies geschied über ein Daisy Chain Verfahren. Somit bekommen alle Controller nach der Reihenfolge wie sie am Bus angeschlossen sind eine aufsteigende ID zugewiesen.

Nach der Zuweisung der ID befindet sich die Routine in einer Kommunikationsschleife welche auf Nachrichten über den CAN-BUS wartet. Dies ermöglicht es Informationen über die aktuell im Flashspeicher befindliche Software zu bekommen, neue Daten in den

Flashspeicher zu laden und das eigentliche Steuerungsprogramm zu starten.

### 4.1.2 Steuerungsprogramm

Das eigentliche Steuerungsprogramm des DSP wurde mit einem Single-Loop-Verfahren<sup>1</sup> realisiert. Nach weiteren Initialisierungen, wie der des MCA Interfaces, Sensoren und Reglern, wird zyklisch, auf Änderungen der Parameter und Steuerbefehle der MCA-Steuerung (ControllerInput CI) geprüft, PWM-Motor Signale durch die Regler generiert und Sensordaten an das SensorOutputinterface weiter geleitet. Am Ende der Schleife wird auf ein Flag<sup>2</sup> zum Beginn eines neuen Zykluses gewartet. Abbildung 4.1 zeigt nochmals den Ablauf der Single-Loop.

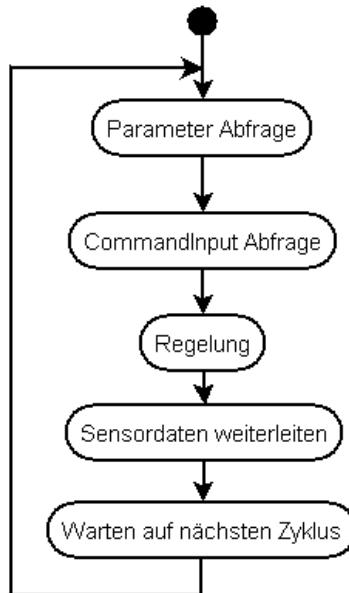


Abbildung 4.1: Single Loop des DSP Programmes

Das Flag zum Beginn eines neuen Zykluses wird mit Hilfe eines Timerinterrupts gesetzt und bei Beginn des Zykluses wieder gelöscht. Der Timerinterrupt wird durch den ersten Kanal des Quad Timer A des DSP gesetzt (siehe 2.2.5.3). Die Zeit in der der Timer den Interrupt auslöst kann mit Hilfe der Parameter zwischen 1 - 200 ms eingestellt werden. In der Standardkonfiguration des *LAURON IVb* wird dieser auf 10ms gesetzt und somit wird die Hauptschleife alle 10ms durchgeführt.

<sup>1</sup>Single-Loop: eine Schleife die periodisch durchlaufen wird

<sup>2</sup>Flag: deutsch Flage bzw. Signal

## 4.2 Implementation Beinregelung

Wie die Analyse des vorhandenen Programms ergab, wird in diesem das Single-Loop Verfahren angewandt, in der zyklisch die Schleife durchlaufen wird. Die vorhandene Regelung wird ebenfalls innerhalb dieser Schleife durchgeführt. Dieser Mechanismus ist für das neue Regelkonzept nicht praktikabel, da es für die Regelung auf feste und konstante Abtastzeiten ankommt, die unabhängig von der Single-Loop Zeit sein sollten. Dies führt dazu, die Regelung, so wie die Erfassung der Geschwindigkeit, aus der Hauptschleife herauszunehmen und sie in einer eigenen Timerinterruptroutine unterzubringen. Hierbei bietet sich der Timer des 2ten Kanals des Quad Timers A an, welche unabhängig vom ersten Kanal die Interruptroutine der Regelung aufrufen kann.

### 4.2.1 Erfassen der Geschwindigkeit

Der Impulswert der Inkrementalgeber an der Motorwelle steht in einem 16bit Register des FPGA zur Verfügung. Der Wert wird entsprechend der Laufrichtung des Motors hoch bzw. herunter gezählt. Zur Ermittlung der Geschwindigkeit kann der Wert aus diesem Register genommen werden. Anschließend muss das Register zurück auf Null gesetzt werden, damit im Register nur die Impulswerte zwischen zwei Abtastungen stehen.

Zur Erfassung der Winkelgeschwindigkeit der Motorwelle eignet sich wie in 3.1.2.2 beschrieben, zwei Verfahren. Die Frequenzmessung und das kombinierte Messverfahren.

Für das kombinierte Messverfahren wird eine zusätzliche Impulsquelle mit einer konstanten Referenzfrequenz benötigt. Auf der Controllerplatine ist keine zusätzliche Referenzfrequenz vorhanden, auf welche zurück gegriffen werden könnte. Somit muss die Referenzfrequenz anders erzeugt werden. Diese kann über einen weiteren Timer realisiert werden. Der DSP56f803 hat eine Taktfrequenz von 80MHz. Der maximale Takt mit dem die Timerzähler herunter bzw. herauf gezählt werden können beträgt die Hälfte der Taktfrequenz. Hier also 40MHz. Ein weiterer Timer könnte also eine Impulsquelle für das kombinierte Verfahren mit einer maximalen Referenzfrequenz von 40MHz erzeugen.

Es zeigt sich jedoch, dass dieser Aufwand keinen wesentlichen Vorteil bringt und es ausreicht das einfache Frequenzmessverfahren einzusetzen. Mit einer Abtastfrequenz von 1ms kann das Frequenzmessverfahren eine minimale Winkelgeschwindigkeit im Gelenk von

$$1_{Impuls} * \frac{1}{1ms} / 128 \frac{1}{U_{mdrehung}} / 134_{U-Getriebe} / 5_{U-Gelenk} = 0.0116 \frac{U_{mdrehung}}{s} \quad (4.1)$$

$$= 4.198 \frac{Grad}{s}$$

erfassen. Dies erscheint völlig ausreichend für die Geschwindigkeitsregelung und es kann

auf den Mehraufwand einer gesonderten Impulsquelle, zu Gunsten der DSP-Ressourcen, verzichtet werden. Die Geschwindigkeitserfassung wurde daher mit Hilfe des Timer des 2ten Kanals des Quad Timer A implementiert. Wobei der Timer alle 1ms die Interruptroutine für die Geschwindigkeitsmessung aufruft. Die Messung an sich geschieht mit dem Frequenzverfahren.

### 4.2.2 Implementation der Drehzahlregelung

Der PI-Regler der Drehzahlregelung konnte nach dem rekursiven Algorithmus wie Gleichung (3.36) umgesetzt werden. Hierzu mussten lediglich die zurückliegende Regelabweichung  $e_{[(k-1)T]}$  sowie die Stellgröße  $u_{[(k-1)T]}$  abgespeichert werden. Die Parameter  $K_P$ ,  $T_N$  sowie  $T$  können von Außen über das MCA-Controller Interface separat für jeden Motor gesetzt werden.

Um auch bei der Drehzahlregelung eine konstante Abtastzeit zu bekommen kann auch hier wieder ein gesonderter Timer, z.B. Kanal 3 des Quad Timer A, verwendet werden. Um aber die Ressourcen des DSP zu sparen, kann die selbe Timerinterruptroutine eingesetzt werden wie die der Geschwindigkeitserfassung. Da der Timer auf 1ms gesetzt wurde kann die Abtastzeit der Drehzahlregelung mit Hilfe einer Zählvariable auf ein Vielfaches von 1ms gebracht werden. Die Zählvariable wird bei jedem Aufruf der Routine um eins hochgezählt. Wie in Kapitel 3.1.3.4 beschrieben, soll die Abtastzeit der Drehzahlregelung 2ms betragen da somit die gemessene Geschwindigkeit mittels eines Averagesfilters geglättet werden kann um möglichst zu starke Quantisierungsfehler zu vermeiden.

### 4.2.3 Implementation der Lageregelung

Die Lageregelung überlagert wie in Kapitel 3.1.3.3 beschrieben die Drehzahlregelung und soll nach 3.1.3.4 eine Tastzeit von 4ms besitzen. Um wiederum sparsam mit den Ressourcen des DSP umzugehen, wird auch die Lageregelung innerhalb der Timerinterruptroutine der Geschwindigkeitserfassung untergebracht. Da es sich bei dem Regler der Lageregelung um einen einfachen P-Regler handelt braucht kein gesonderter Algorithmus zur Regelung implementiert werden. Auch hier kann der Parameter  $K_P$  des Reglers von Außen für jedes Gelenk einzeln eingestellt werden.

## 4.3 Implementation der Kraftregelung

Die Kraftregelung überlagert wiederum den Lageregelkreis. Hierbei macht es jedoch keinen Sinn sie ebenfalls mit in die Timerinterruptroutine der Geschwindigkeitserfassung mit aufzunehmen. Von den Kraftsensoren stehen die ermittelten Kraftwerte alle 8,4ms zur Verfügung [34]. Hierbei bietet sich an, die Kraftregelung innerhalb der Singel-Loop

des Hauptprogrammes zu realisieren. Da diese in der Standardkonfiguration alle 10ms aufgerufen wird. Was für die Kraftregelung eine gute Abtastzeit bedeutet. Um aber die Kraftregelung unabhängig von der einstellbaren Zykluszeit der Hauptschleife zu machen kann auch hier wieder ein eigener Timer verwendet werden. In der Diplomarbeit wurde jedoch auf die Variante mit eigenen Timer verzichtet und die Kraftregelung innerhalb der Hauptschleife realisiert.

### 4.3.1 Erfassung der Kraftdaten

Die Kraftwerte werden über Dehnungsmessstreifen (vgl. Abschnitt 2.2.3) erfasst und mit Hilfe eines AD/Wandlers dem Mikrocontroller bereit gestellt. Die gemessenen Werte sind jedoch stark verrauscht. Abbildung 4.2 zeigt die Messungen der Z-Kraftkomponente, des Kraftsensors. Wie hierbei zu sehen ist schwankt der Wert von +10–−15 und kann so direkt nicht für die Kraftregelung genommen werden.

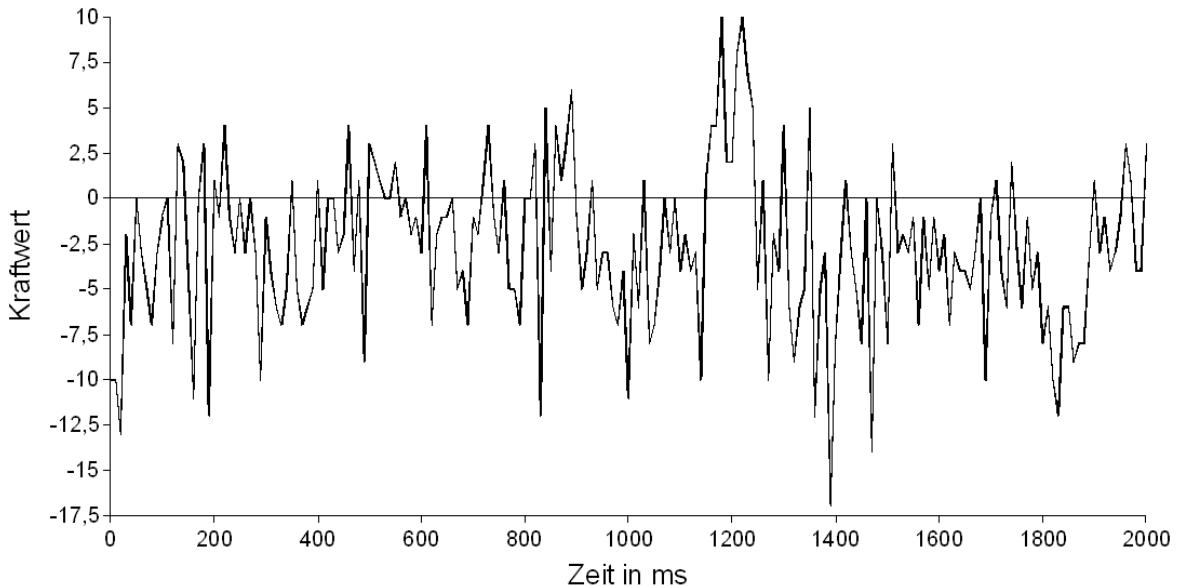


Abbildung 4.2: Z-Kraftwert

Der Wert muss, bevor die Kraftregelung ihn übernehmen kann, geglättet werden. Hierbei bieten sich verschiedene Verfahren wie MedianFilter und AverageFilter usw. an. Um jedoch den Zeitvorteil, den die Umsetzung der Kraftregelung in den Mikrocontroller mit sich bringt, nicht wieder durch eine Filterung, die Zeitbeansprucht, zunichte zumachen wurde bei der Implementierung zur Glättung der Kraftwerte ein Schwellenwertverfahren

benutzt. Gleichung (4.2) beschreibt die Umsetzung des Verfahrens.

$$fth = \begin{cases} f & , \|f\| > th \wedge (f > (fth + th) \vee f < (fth - th)) \\ fth & , \|f\| > th \wedge f < (fth + th) \wedge f > (fth - th) \\ 0 & , |f| \leq th \end{cases} \quad (4.2)$$

wobei,

$$f = \text{gemessener Kraftwert der jeweiligen Komponente } (x,y,z) \quad (4.3)$$

$$th = \text{Schwellenwert, für jede Komponente separat einstellbar} \quad (4.4)$$

$$fth = \text{nach dem Schwellenwertverfahren geglätteteren Kraftwert} \quad (4.5)$$

Dieses Verfahren bieten den Vorteil das auf eine Veränderung der Kräfte sofort reagiert werden kann, da bei diesem Verfahren keine Verzögerungen auftreten. Der Schwellenwert  $th$  kann für jeden Kraftsensor und jede Kraftkomponente separat angepasst werden.

### 4.3.2 Trigonometrische Berechnung

Für die Berechnung der Inversen Kinematik wie auch zur Transformation der Fußkräfte in das BBKS werden trigonometrische Funktionen wie Cosinus, Sinus und Arkustangens benötigt. Die Standardbibliothek innerhalb des CodeWarrior<sup>3</sup> beinhaltet keine „high level floating point math function“ so wie es Cosinus, Sinus und Arkustangens sind[11]. Das Embedded SDK<sup>4</sup> bietet hier jedoch trigonometrische Funktionen an, die auf einen fraktalen Datentyp basieren. Der fraktale Datentyp stellt eine Gleitkommazahl mit einem Integertyp dar. Er wird intern über einen 16bit Integer (*Frac16*) bzw. über einen 32bit Integer (*Frac32*) dargestellt. Dies hat den Vorteil, dass der DSP Controller die gleichen Maschineninstruktionen zur Addition, Subtraktion, Vergleiche und logische Operationen für die Berechnung fraktaler Werte benutzen kann wie die, die er für normale Integer Berechnung verwendet, was zu schnelleren Berechnungen führt [10][12]. Nachteilig ist allerdings das aufgrund der Darstellung die Gleitkommazahl auf den Bereich

$$-1.0 \leq \text{Gleitkommazahl} \leq +1.0 - 2 - (N - 1)$$

begrenzt ist. Wobei  $N$  die Anzahl der Bits für ein Integer sind. Für einen *Frac16* Wert gilt somit

$$-1.0 \leq \text{Gleitkommazahl} \leq 0.99996948242188.$$

---

<sup>3</sup>Entwicklungsumgebung für das DSP-Programm

<sup>4</sup>Software Development Kit

## 4 Implementierung und Integration

Die folgende Gleichung gibt die Beziehung zwischen einem *Frac16* Intergerwert und einer Gleitkommazahl wieder.

$$\begin{aligned} \text{Gleitkommazahl} &= \frac{\text{Frac16}}{2^{15}} \\ \text{Frac16} &= \text{Gleitkommazahl} \cdot 2^{15} \end{aligned} \quad (4.6)$$

Diese Beschränkung muss bei der Berechnung der Inversen Kinematik beachtet werden. Ein weiterer Nachteil dieser Beschränkung liegt darin, dass oft zwischen dem Datentyp *Frac16* bzw. *Frac32* und dem Datentyp *double* gewechselt werden muss um vernünftige Ergebnisse zu erlangen [18].

Probleme machte die Limitierung vor allem bei der Berechnung des  $\alpha$ -Winkels der inversen Kinematik.

$$\alpha = \arctan\left(\frac{x}{z}\right)$$

Das Ergebnis der Berechnung  $\frac{x}{z}$  kann hierbei die Begrenzung des fraktalen Wertes verletzen. Eine Abhilfe schafft hier die Beziehung

$$\arctan\left(\frac{x}{z}\right) = \frac{\pi}{2} - \arctan\left(\frac{z}{x}\right) \quad (4.7)$$

Die Gleichung (3.42) kann nun mit (4.7) zur Berechnung des  $\alpha$ -Winkels mit den Funktionen der Embedded SDK Bibliothek genutzt und direkt umgesetzt werden.

$$\alpha = \begin{cases} \arctan\left(\frac{x}{z}\right) & , z \neq 0 \wedge -1 \leq \frac{x}{z} \leq 1 - 2^{-15} \\ \frac{\pi}{2} - \arctan\left(\frac{z}{x}\right) & , \frac{x}{z} > 1 \vee \frac{x}{z} < -1 \\ \frac{\pi}{4} & , z \neq 0 \\ -\operatorname{sgn}(x) \cdot \frac{\pi}{2} & , z = 0 \wedge x \neq 0 \\ \text{beliebig} & , z = 0 \wedge x = 0 \end{cases} \quad (4.8)$$

Ein weiteres Problem mit der Beschränktheit des Datentyps tritt bei der Berechnung der Dreiecksseite  $c_\Delta$  auf (3.47). Hierbei kann der Ausdruck  $\sqrt{y^2 + z^2}$  die Beschränkung überschreiten. Dieses Problem wurde dadurch gelöst, dass alle Längenangaben mit den Faktor  $\frac{1}{1000}$  bzw. 0.001 multipliziert wurden. Dadurch das sich die Seitenverhältnisse nicht geändert haben, verändert sich nichts bei der Berechnung des Winkels und der Ausdruck  $\sqrt{y^2 + z^2}$  bleibt garantiert innerhalb der Beschränkung.

Trigonometric Function	Oscillator Cycles	Machine Instructions
sinPIx	900	195
cosPIx	1078	244
AsinOverPI	1237	300
AcosOverPI	1375	335
AtanOverPI	2337	597
Atan2OverPI	2606	665

Tabelle 4.1: Rechenzeiten der trigonometrischen Funktionen [12]

Während der Implementierung der Kraftregelung wurde ein Fehler in der Embedded SDK Funktion *Frac16 tfr16CosPIx (Frac16 x)* entdeckt. Ist der Winkel  $x = 0^\circ$  so müsste die Funktion einen Wert von 1 liefern. Wie bereits erwähnt liegt dieser Wert außerhalb der Beschränkung von  $-1 \leq \text{fraktalerWert} < 1$ , nach [12] nimmt der *Frac16* bzw. *Frac32* Datentyp bei Überschreitung seiner Grenzen den maximalen bzw. den minimalen zulässigen Wert an. Dies geschieht bei der *tfr16CosPIx(x)* jedoch nicht. Sie liefert bei  $x = 0$  den Wert  $-1$  was in der fraktalen Darstellung einen Integerwert von 32768 oder Hex 0x8000 beträgt. Dieser Fehler muss mit einer vorhergehenden Überprüfung von  $x = 0$  umgangen werden. Da sonst die nachfolgenden Berechnungen mit einem Falschenwert weiter berechnet würden.

### 4.3.3 Berechnungszeit

Über die Berechnungsdauer der geometrischen Inversen Kinematik können Aussagen anhand der Tabelle 4.1 gemacht werden. Diese Tabelle wurde aus dem Handbuch der Beschreibungen der Mathematischen Funktionen des Embedded SDK [12] entnommen. Sie beschreibt die Anzahl an Oszillatoren Zyklen welche für eine jeweilige Funktion benötigt werden.

Zur Berechnung der geometrischen Inversen Kinematik werden nach Abschnitt 3.2.3.1 2x Arkustangens, 2x Arkuskosinus sowie Cosinus und Sinus benötigt. Die Frequenz des DSP wurde auf 80Mhz eingestellt. Daraus lässt sich nun die Zeit berechnen welche für die trigonometrischen Funktionen der Inversen Kinematik benötigt wird.

$$((1/80\text{Mhz})) * (900_{\text{sin}} + 1078_{\text{cos}} + 2 * 1375_{\text{acos}} + 2 * 2337_{\text{atan}}) = 0.117525\text{ms} \quad (4.9)$$

Zu der Berechnung der Inversen Kinematik werden hierbei nur  $\approx 0.2\text{ms}$  benötigt. Was ihren Einsatz innerhalb des Mikrocontrollers rechtfertigt und es zu keiner Verzögerung der Regelung, minimale Abtastzeit 1ms, kommen wird. Somit kann hier gezeigt werden das eine Implementierung der Kraftregelung innerhalb des Mikrocontrollers machbar ist und dies zu einer Verbesserung der Reaktionszeit der Kraftregelung führt.

# 5 Ergebnisse

In dieses Kapitel werden die in dieser Arbeit entstandenen Erweiterungen des Programms auf dem Mikrocontroller vorgestellt und in ihren Funktionsweise erklärt. Anschließend wird im Abschnitt Messergebnisse die Funktionalität des neuen Regelungskonzeptes gezeigt. Hierzu werden ermittelte Daten des neuen System präsentiert und anschließend ihre Aussagen diskutiert.

## 5.1 Erweiterungen der DSP-Software

Zur Erweiterung der DSP-Software wurden hauptsächlich im Bereich Controller neue Funktionen und Datentypen deklariert. Im folgenden werden nun die wichtigsten Funktionen und Typen beschrieben:

In der Datei *PRegel.h* sowie *PIRegel.h* wurde die neuen Datentypen *tPRegler* und *tPIRegler* definiert. Sie beinhalten den aktuellen Zustand eines Reglers. Des Weiteren wurden die Funktionen:

*void tPIRegler\_Init(tPIRegler PL\_regler)*  
Initialisierung der übergebenen Reglerstruktur

*void tPIRegler\_SetParameter(tPIRegler PL\_regler, int max\_control\_amount, float t0, tParameterType p, tParameterType tn)*  
Setzen der Parameter des übergebenen Reglers, entsprechend für den *PRegler* gibt es eine Funktion die nur die Parameter des PReglers benötigt.

*int tPIRegler\_Control(tPIRegler PL\_regler, float actual, float target)*  
Hier wird  $e(k) = target - actual$  berechnet und die eigentliche Regelung nach der Gleichung (3.36) durchgeführt. Rückgabewert dieser Funktion ist entsprechend  $u(kT)$

zur Kontrolle und Steuerung der Regler definiert.

Die Geschwindigkeitserfassung und die kaskadierende Lageregelung wurden in der *MotorSpeedControl.h* definiert. In ihr befinden sich die Funktionen zur Steuerung der Regelung:

## 5 Ergebnisse

*int StartControlAlpha()*

Starten der Regelung des Alpha Gelenkes, für Beta und Gamma ist jeweils eine gleiche Funktion mit entsprechendem Namen vorhanden. Die Regler des jeweiligen Gelenkes müssen vor dem Start der Regelung initialisiert werden sonst wird die Regelung nicht gestartet. Der Rückgabeparameter gibt hier an, ob die Regelung gestartet werden konnte (= 0) oder ob diese nicht geschehen konnte. In diesem Fall wird ein Fehlercode mit jeweiligen Grund zurückgegeben.

*void StopControlAlpha()*

Stoppt die Regelung des jeweiligen Gelenkes. Zusammen mit *void StartControlAlpha()* kann so die neue Regelung der einzelnen Gelenke individuell gestartet oder gestoppt werden.

*void InitMotorSpeedControl()*

Diese Funktion initialisiert den Timer für die Geschwindigkeitserfassung und initialisiert die für die Lageregelung notwendigen Regler. Hierbei werden jeweils 2 Regler pro Gelenk erstellt. Einen PI für die Drehzahlregelung und einen P für die übergeordnete Lageregelung.

*void stopSpeedControl()*

Mit dieser Funktion ist es möglich die Geschwindigkeitserfassung und die darauf aufbauende Regelung anzuhalten. Dafür wird der Timer der Geschwindigkeitserfassung geschlossen.

*void startSpeedControl()*

Hiermit ist es möglich die neue Regelung wieder zu starten. Zusammen mit *void stopSpeedControl()* kann man nun die neue Regelung je nach Bedarf aus- bzw. wieder dazuschalten.

*void set\_MSC\_PI\_Parameter(tMotor motor, tParameterType p, tParameterType tn, int max\_control)*

Zum Setzen der Parameter der einzelnen Regler wird diese Funktion benötigt. Sie setzt die übergebenen Parameter für den jeweiligen angegebenen Motor. Dazu wird die Funktion *void tPIRegler\_SetParameter(tPIRegler PL\_regler, ...)* aus *PIRegel.h* aufgerufen. Entsprechend für die Parameter der P-Regler der Lageregelung gibt es eine Funktion mit entsprechenden Namen.

*void setTargetPosition(tMotor motor, int target)*

Setzt den anzufahrenden Winkel des übergebenen Motors bzw. Gelenkes. *target* ist die zu erreichende Gelenkstellung in EncoderImpulsen.

*void setSpeed(tMotor motor, int speed)*

Damit sich das Gelenk direkt mit einer bestimmten Geschwindigkeit bewegt kann man mit dieser Funktion die Geschwindigkeit setzen. Hierbei wird die maximale Geschwindigkeit festgelegt, mit welcher sich das Gelenk in Richtung, des durch *void setTargetPosition(...)* angegebene Winkel, bewegen wird. Die Lageregelung

## 5 Ergebnisse

sorgt dafür, dass die Geschwindigkeit bei Erreichen des angegebenen Winkels auf Null verringert wird und damit keine Schäden des Gelenkes entstehen.

Diese Funktionen stellen allesamt die Funktion der neuer Lageregelung sicher. Hierbei wurden jedoch nur die wichtigsten Funktionen beschrieben. Für eine Auflistung aller Funktionen und ihren Beschreibungen kann die *MotorSpeedControl.h* Datei im Modul *Controller* herangezogen werden.

Die für Kraftregelung notwendigen Komponenten sind in der Datei *Kraftregler.h* zu finden. Im Großen und Ganzen sind 4 Funktionen für die Funktionalität der Kraftregelung notwendig:

*void setKinematic(int alpha, int beta, int gamma)*

Diese Funktion berechnet vorab, die für die Krafttransformation (vgl. Abschnitt 3.2.2) benötigten trigonometrischen Variablen,  $c\alpha, s\alpha, s\beta, c\beta, \dots$  (siehe Gleichung (3.39), Tabelle 3.1). Die übergebenen Winkel werden in EncoderImpulsen angegeben und beziehen sich nicht auf die aktuellen Winkel der Gelenke sondern auf die, von den Gelenken anzufahrenden Winkel, welche über die Steuerung vorgegeben werden. Diese Funktion braucht nur bei Änderung der vorgegebenen Winkel neu aufgerufen werden.

*void setDamping()*

Hiermit kann die Art der Kraftregelung beeinflusst werden, entsprechend dazu gibt es die Funktionen *setComplinace()* und *setNoControl()*.

*void setSollForces(int fx, int fy, int fz)*

Die für die Berechnung der Positionsänderung  $\Delta\vec{x}$  des Nachgiebigkeitsreglers Gleichung (3.37) sowie zur Berechnung  $\dot{\Delta\vec{x}}$  des Dämpfungsreglers (3.38) benötigten Soll-Kräfte der einzelnen Kraftkomponenten werden durch diese Funktion gesetzt.

*int Kraftregelung(double kpx, double kpy, double kpz, int \*alpha, int \*beta, int \*gamma, int time)*

Diese Funktion führt die eigentliche Kraftregelung aus. In ihr werden die aktuell gemessenen Kraftwerte abgefragt, anschließend mit den vorab berechneten Werten aus *void setKinematic(int alpha, int beta, int gamma)* die Krafttransformation der Kräfte aus dem *FPKS* in das *BBKS* durchgeführt. Mit den Kraftwerten im *BBKS* werden nun die Berechnungen gemäß den Gleichungen (3.37) und (3.38) ausgeführt. Die berechneten Positionsabweichungen werden im Anschluss mit Hilfe der geometrischen Inversen Kinematik in Gelenkwinkel umgerechnet (vgl. Abschnitt 3.2.3.1). Über die Variablen *alpha*, *beta*, *gamma* werden die errechneten neuen Gelenkwinkel in EncoderImpulsen weiter gegeben. Der Parameter *time* ist für den Dämpfungsregler notwendig. Da hier keine Positionsänderung direkt berechnet wird, sondern eine Geschwindigkeit  $\dot{\Delta\vec{x}}$  welche sich aber über den Ausdruck  $s = v \cdot t$  in eine Positionsänderung überführen lässt.

Die hier beschriebenen Funktionen und Datentypen kamen im Rahmen dieser Diplomarbeit neu zu der Software des DSP hinzu. Weitere Funktionen und Beschreibungen können in den dazu gehörigen Dateien *Kraftregler.h*, *MotorSpeedControl.h*, *PRegler.h* und *PIRegler.h* nachgelesen werden.

## 5.2 Messergebnisse

In diesem Abschnitt werden nun mit Hilfe der neuen Funktionen gewonnene Messergebnisse präsentiert und im Einzelnen diskutiert um die Funktionsweise der neuen Regelungskonzepte zu beweisen.

In den nachfolgenden Abbildungen symbolisiert der blaue Graph den an Motor angelegten PWM-Wert, hierbei bedeutet ein PWM-Wert von  $16000 \approx 12V$  was die maximale Zulässige Motorspannung ist. Der PWM-Wert wird von der Drehzahlregelung eingestellt außer bei den Abbildungen 5.2, hier wurde der PWM-Wert auf konstant 16000 gestellt um die Kennlinien der Regelstrecke aufzunehmen.

Der rote Funktionsgraph gibt hier immer die, von der Geschwindigkeitserfassung, aktuell erfasste Geschwindigkeit wieder. Die Einheit der Geschwindigkeit beträgt hier EncoderImpulsen pro ms. Der besseren Übersicht wegen wurde der Wert der Geschwindigkeit um den Faktor 100 multipliziert. Somit entspricht die Einheit EncoderImpulsen pro 100ms.

Die grüne Kurve zeigt stets die, von der Lageregelung erzeugte, SollGeschwindigkeit für die Drehzahlregelung an. Hierbei ist ebenfalls auf Grund der Übersicht die Einheit EncoderImpulse pro 100ms.

### 5.2.1 Drehzahlregelung des ausgebauten Motors

Das Diagramm 5.1 zeigt die Drehzahlregelung des ausgebauten Motors. Hierbei soll der Motor von 0 auf 100 EncoderImpulsen/1ms gebracht und gehalten werden. Maximale Drehzahl des ausgebauten Motors ist  $\approx 115$  EncoderImpulsen, wie aus der Abbildung 3.5 entnommen werden kann.

Die Einstellparameter für den Drehzahlregelkreis wurden nach Betragsoptimum 3.1.3.5 berechnet:

$$p = 60$$

$$tn = 0.015$$

Die Kurve weist, wie erwartet, ein Einschwingverhalten mit einer Dämpfung von  $\approx 0.6$  auf. Diese Kurve zeigt somit die korrekte Funktionsweise des Drehzahlregelkreises.

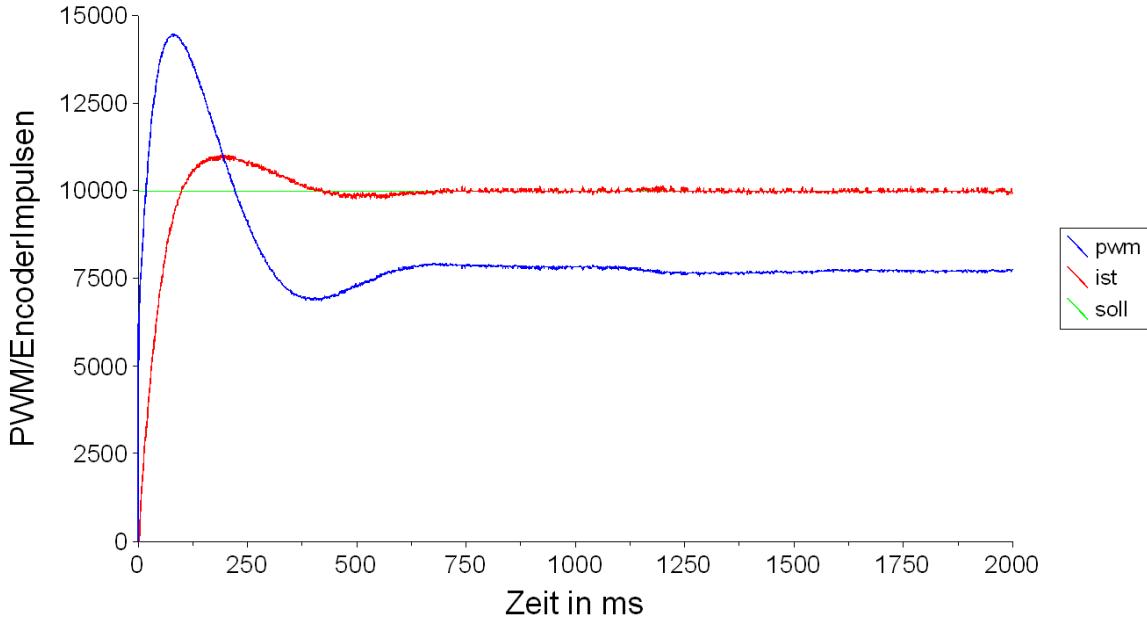


Abbildung 5.1: Drehzahlregelung des ausgebauten Motors

### 5.2.2 Verhalten der Beta-Gelenk Regelstrecke

Wie vermutet zeigt das Spannung/Zeitverhalten des eingebauten Motors einen anderen Verlauf als im ausgebauten Zustand. Abbildung 5.2 zeigt die Kennlinien des Beta-Gelenk Motors (vgl. Abschnitt Aufbau eines Roboterbeines 2.2). Hierbei wurden zwei Kennlinien aufgenommen. Kennlinie 5.2(a) zeigt den Verlauf bei einer Aufwärtsbewegung des Beines von der Gelenkstellung  $-36^\circ$  bis auf  $90^\circ$ . Kennlinie 5.2(b) den umgekehrten Fall von  $90^\circ$  auf  $-36^\circ$ .

Deutlich zu erkennen sind die Unterschiede der beiden Diagramme im Bezug auf ihr Zeit- als auch auf ihr maximales Drehzahlverhalten. Die Kurve der Aufwärtsbewegung besitzt wesentlich größere Zeitkonstanten als bei der Abwärtsbewegung, zudem wird nur eine maximale Geschwindigkeit von  $\approx 95$  EncoderImpulse pro 1ms erreicht. Dies war zu erwarten, da bei der Aufwärtsbewegung der Motor gegen die Gravitation arbeiten muss. Hingegen bei der Abwärtsbewegung das Gewicht der Segmente C und D den Motor förmlich mit antreibt und dadurch eine maximale Geschwindigkeit von  $\approx 121$  EncoderImpulsen erreicht wird, was sogar noch über der maximalen Drehzahl des Motors im ausgebauten Zustand liegt.

Beide Kennlinien weisen eine „Beule“ in ihren Graphen auf. Vermutlich kommt dieses Verhalten durch die hohe Elastizität der Regelstrecke zwischen Motor und Getriebe sowie zwischen Getriebe und Gelenk zustande.

## 5 Ergebnisse

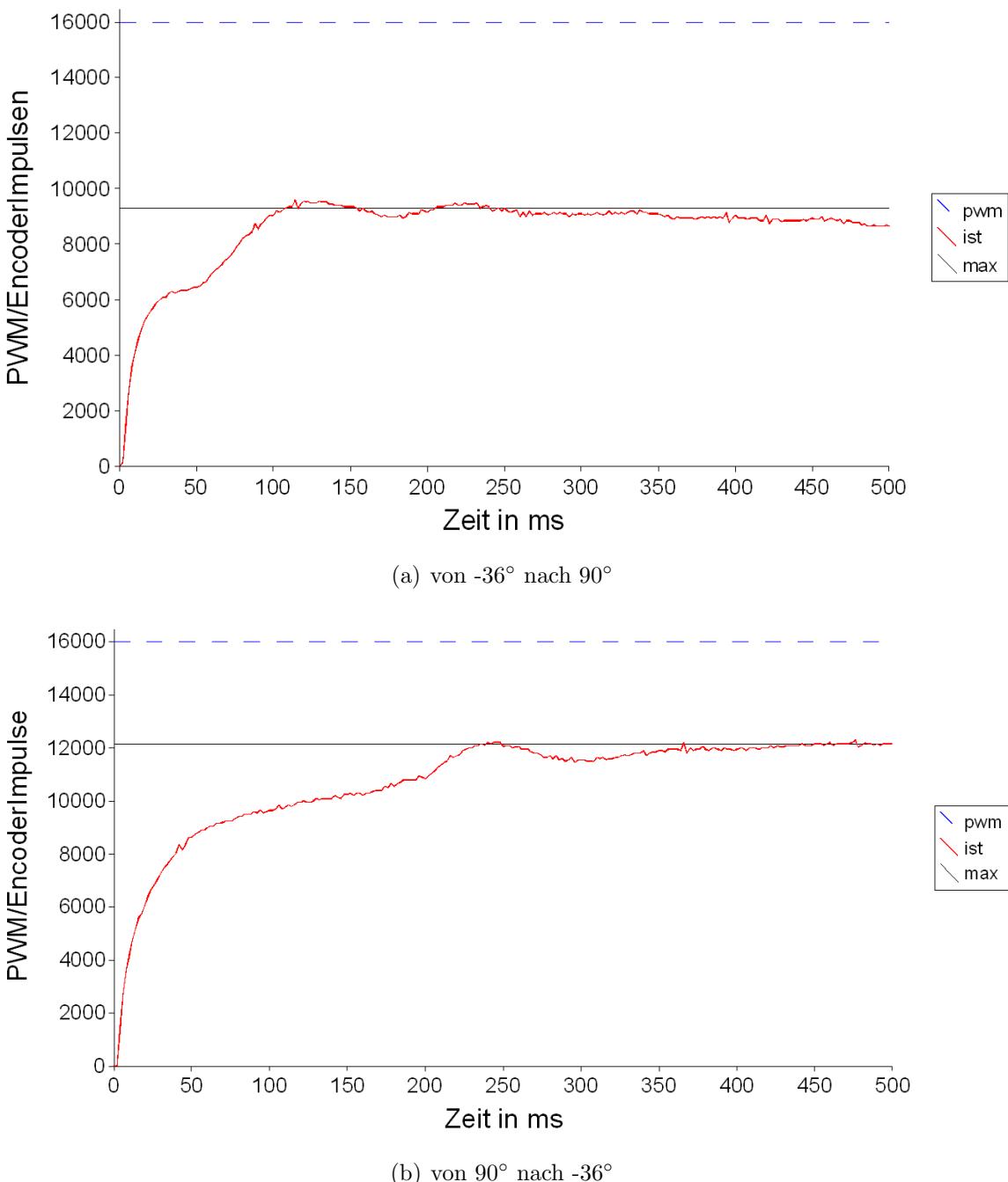


Abbildung 5.2: Drehzahlverhalten der Beta-Gelenk Regelstrecke

### 5.2.3 Regelung der Beta-Gelenk Regelstrecke

Nachdem sich gezeigt hat, dass das zeitliche Verhalten abhängig von der Bewegungsrichtung des Beta-Gelenks ist, wird es schwer sein Einstellparameter zu finden, welche für beide Richtungen ein optimales Regelverhalten darstellen. Abbildung 5.3 zeigt die Drehzahlregelung des Beta-Gelenkes, einmal in Aufwärtsrichtung von  $-36^\circ$  auf  $90^\circ$  und einmal in entgegen gesetzter Richtung.

Aufgrund des veränderten Verhaltens brachten die ermittelten Parameter des ausgebauten Systems keine optimalen Ergebnisse mehr, zudem besitzen die beiden Kennlinien kein reines PT2 bzw. PT1 Verhalten mehr. Das Einstellen der Parameter erfolgte daher durch das Prinzip des einfachen Ausprobierens. Hierbei wurden die Parameter verändert und geschaut wie sich die veränderten Parameter auf die Regelung auswirkten. Als Ausgangspunkt zur Einstellung boten sich die errechneten Werte des ausgebauten System an. Die Parameter wurden nun so lange angepasst bis sich ein ein gutes Regelverhalten bei der Abwärtsbewegung einstellte. Folgende Parameter wurden für die Regelung ermittelt:

$$sp = 55$$

$$tn = 0.01$$

$$pp = 1$$

wobei  $sp$  der P-Parameter des Drehzahlregelkreises darstellt und  $pp$  der P-Parameter des Lageregelkreises. Wie Abb. 5.3(b) zeigt wurden die Parameter so ermittelt, dass sie ein überschwingungsfreies Regelverhalten in der Abwärtsbewegung realisieren. Da bei Abwärtsbewegungen zu erwarten ist, dass eine Stützstelle angefahren wird, sind hier Überschwingungen der Position kritischer als bei der Aufwärtsbewegung. Ein mögliches Verfahren um in beiden Richtungen ein gutes Regelverhalten zu bekommen, wäre Einstellparameter für die Aufwärts- und die Abwärtsbewegung separat zu ermitteln und je nach Bewegungsrichtung den Regler mit den jeweiligen Parametersätzen zu setzen.

Abbildung 5.4 zeigt das Verhalten der Regelung mit gesetzter maximalen Geschwindigkeit von 10EncoderImplusen pro 1ms. 10EncoderImpluse ist eine recht langsame Drehzahl und es treten hierbei verstärkt Fehler durch Quantisierungseffekte der Geschwindigkeitserfassung auf. Ein weiterer Effekt, der in Diagramm 5.4(b) zu beobachten ist, ist der, dass die Regelung, um eine Geschwindigkeit von 10EncoderImpulsen zu halten, den Motor abbremsen muss. Hierbei kommt es zu Spannungsumpolungen (PWM-Wert  $< 0$ ), welche zu starken Bremswirkungen des Motor führen, was wiederum zu Schwankungen in der Drehzahl führt. Alles in Allem kann die Regelung aber dennoch gut eine gesetzte Geschwindigkeit mit  $\pm 2$  EncoderImpulsen halten.

Ebenfalls aus den Diagrammen 5.4 ablesbar ist, dass es bei kleinen Geschwindigkeiten des Gelenke zu keinem Überschwinger in der Position des Beine kommt, egal in welcher Richtung das Gelenk gefahren wird.

## 5 Ergebnisse

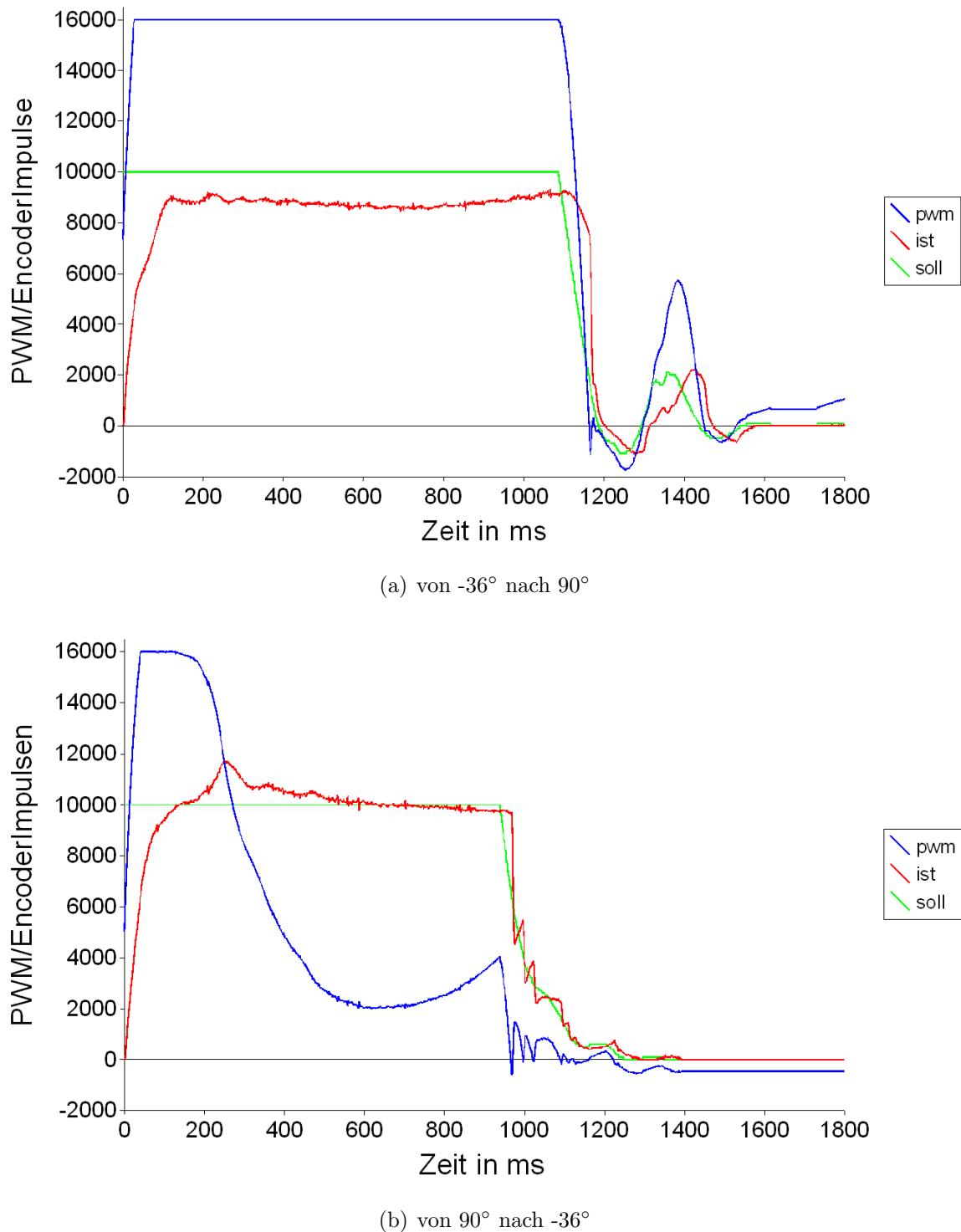


Abbildung 5.3: Drehzahlverhalten der Lageregelung des Beta-Gelenk

## 5 Ergebnisse

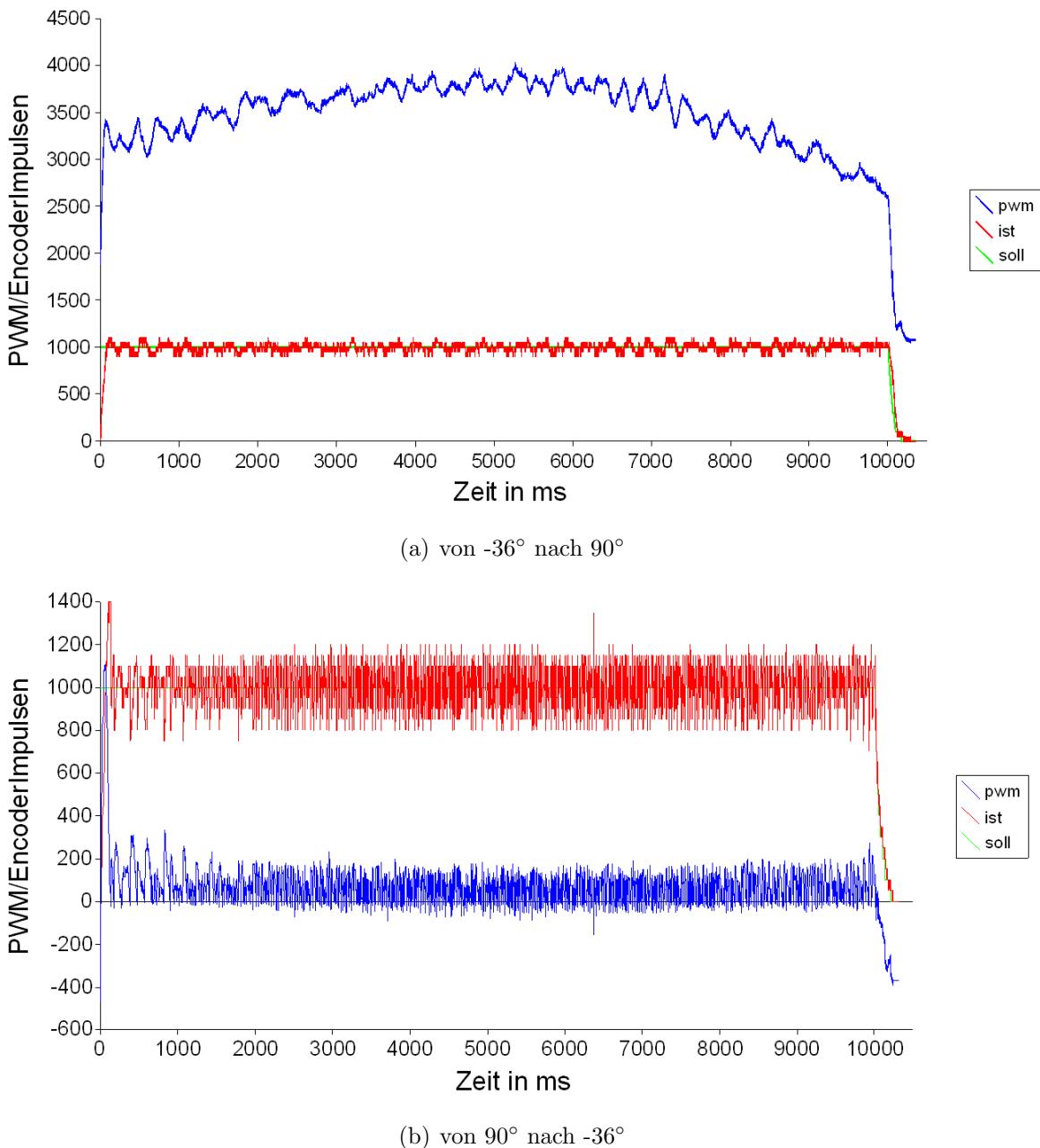


Abbildung 5.4: Drehzahlregelung des Beta-Gelenk

### 5.2.4 Kraftwerte

Für die Kraftregelung werden gemessenen Kräfte gemäß 4.3.1 mit einem Schwellenwertverfahren geglättet. Die Abbildungen 5.5 zeigen hierbei die gemessenen Kräfte(5.5(a)) an der Z-Achse des Kraftsensors und wie sie nach dem Glättung (5.5(b)) zu der Kraftregelung weiter gereicht werden, welche daraus eine Positionsabweichung entsprechend der Krafteinwirkung berechnet. Der Schwellenwert der Glättung wurde hierbei auf 20 eingestellt. Wie zu erkennen ist, wird ein Kraftwert, welcher über dem Schwellenwert liegt, verzögerungsfrei übernommen. Dennoch bietet dieses Verfahren eine nahe zu schwingungsfreies Kraftsignal.

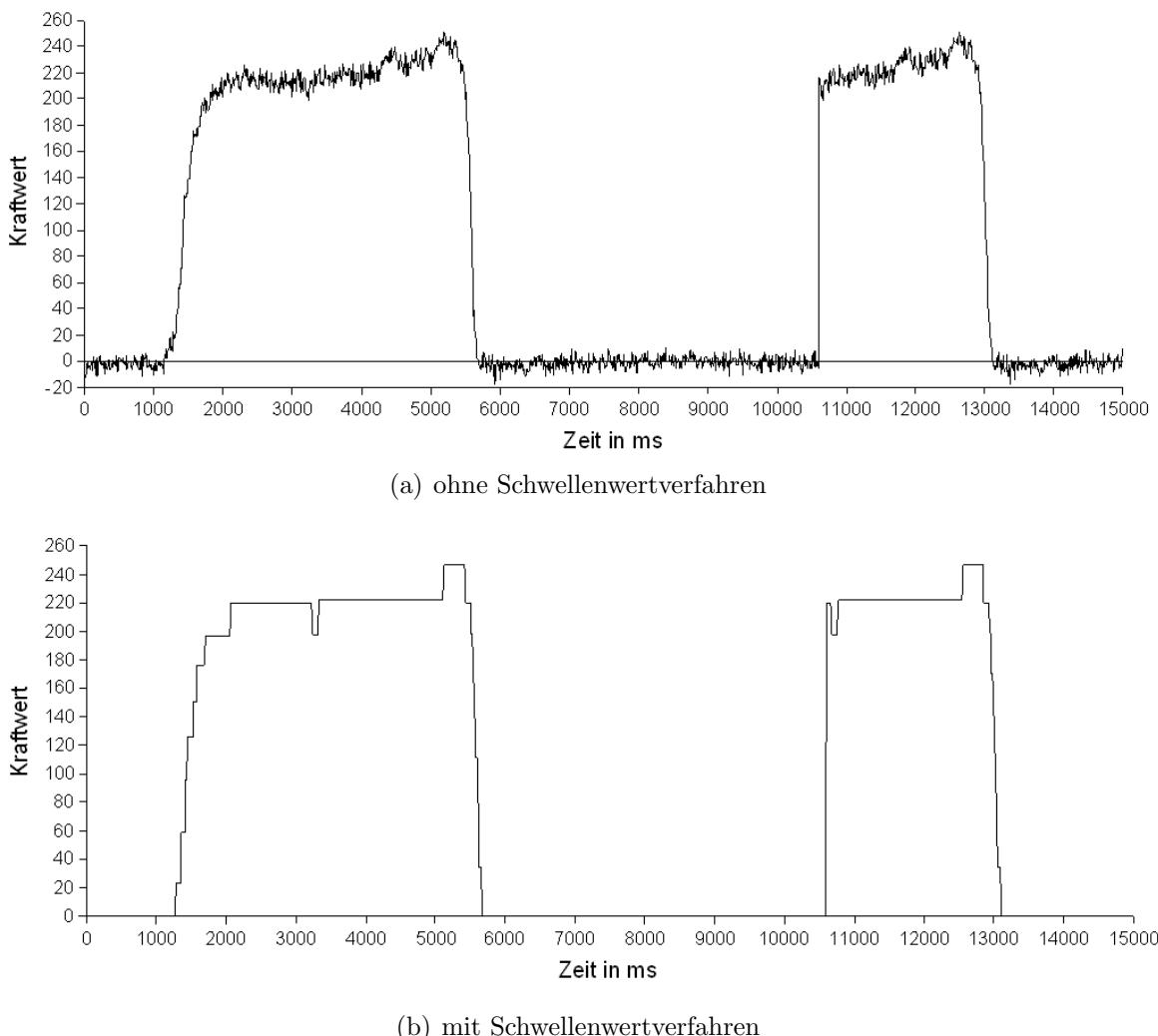


Abbildung 5.5: Z-Komponente Kraftwert Glättung

# 6 Zusammenfassung

Die vorliegende Arbeit beschreibt den Entwurf eines neuen Regelungskonzeptes von Gleichstrommotoren zur Positions- und Geschwindigkeitsregelung eines Beines und ihre Implementierung in das vorhandene System des Laufroboter *LAURON IVb*. Um die Anforderung einer präzisen Positionierung nach Vorgaben des übergeordneten Systems zu erfüllen. Zudem beschreibt sie die Umsetzung einer Kraftregelung der Beine von der übergeordneten Steuerungsebene auf einem Industrie PC hin zu hardwarenahen Steuerungsebene in den DSP56F803. Dies ermöglicht kürzere Reaktionszeiten bei der kraftgeführten Bewegung zu bekommen. Im Ergebnis ist damit eine neue Regelung geschaffen worden, mit der präzise Punkte im Raum angefahren werden können, was für eine genaue Regelung bei technischen Anwendungen wie Bohr- oder Positionieroperationen unverzichtbar ist, und die kürzeste Reaktionszeiten der Kraftregelung mit sich bringt.

Bevor ein neues Regelungskonzept erarbeitet und implementiert werden konnte musste das vorhandene System untersucht werden. Die Motoren des Antriebssystem wurden auf ihre Ansteuerung und auf ihr zeitliches Verhalten hin analysiert. Es wurde gezeigt, dass die vorhandene Regelung auf Grund der groben Auflösung des Winkelgebers und der nichtlinearen lastabhängigen Regelstrecke, den Anforderung einer präzisen Regelung nicht nachkommt und wie mit Hilfe der Impulsgeber an den Motoren, eine Geschwindigkeitserfassung realisiert werden konnte. Dies ermöglichte wiederum den Aufbau einer Drehzahlregelung. Durch die Möglichkeit einer Drehzahlregelung konnte eine kaskadierende Regelstruktur konzipiert und implementiert werden, welche in der Lage ist eine Position auch unter Lastschwankungen genau auszuregeln.

Zur Implementierung der Kraftregelung innerhalb des Mikrocontrollers wurde gezeigt, dass die Berechnung der Inversen Kinematik  $\approx 0.2\text{ms}$  Rechenzeit benötigt, was ihren Einsatz neben der Positionsregelung im Mikrocontroller ermöglicht. Das Problem der stark schwankenden Kraftinformationen wurde mittels eines Schwellenwertverfahrens gelöst, welches es erlaubt verzögerungsfrei auf Kraftänderungen zu reagieren.

Zur Verifikation des neu entworfenen Regelungskonzeptes wurden Messdaten aufgenommen, die das Verhalten der neuen Regelungen beschreiben und bestätigen. Hierbei zeigt es sich, dass das Finden von optimalen Parameter der Regelung, durch die Bewegungsrichtung abhängigen Regelstrecke erschwert wird. In zukünftige Arbeiten könnte, für eine weitere Verbesserung der Regelung, die Regelung um einen Stromregelkreis erweitert werden, welcher das Regelstreckenverhalten verbessert, so dass es kein unterschiedliches Zeitverhalten je nach Bewegungsrichtung mehr gibt.

# Abbildungsverzeichnis

2.1	Lauron Modelle . . . . .	5
2.2	Innenaufbau des Körpers . . . . .	5
2.3	Koordinatensysteme Lauron IVb mit Ansicht von oben aus [31] . . . . .	6
2.4	Aufbau eines Beines . . . . .	8
2.5	Arbeitsbereich des Beines . . . . .	10
2.6	Inkrementaler Positionssensor . . . . .	11
2.7	Fußsensor . . . . .	12
2.8	DSPF803 Blockdiagramm [10] . . . . .	13
2.9	PC/104 [23] . . . . .	14
2.10	„Virtual MainGroup“ der MCA–Steuerung . . . . .	17
2.11	Allgemeiner Regelungskreis [27] . . . . .	18
2.12	Proportionalglied (P-Glied) [27] . . . . .	19
2.13	Integrator (I-Glied) [27] . . . . .	19
2.14	Verzögerungsglied 1.Ordnung (PT1-Glied) [27] . . . . .	20
2.15	Verzögerungsglied 2.Ordnung (PT2-Glied) [27] . . . . .	20
2.16	Blockschaltbild PI-Regler . . . . .	21
2.17	PI-Regler [27] . . . . .	21
2.18	Lageregelung mit unterlagerter Drehzahl- und Ankerstromregelung [9] . . . . .	22
2.19	Kinematische Kette [7] . . . . .	24
2.20	Denavit-Hartenberg-Verfahren [17] . . . . .	24
2.21	Direkte Kinematik [7] . . . . .	25
2.22	Inverse Kinematik [7] . . . . .	27
2.23	Annäherung an $f$ [7] . . . . .	29
3.1	Brückenschaltung zur Motorsteuerung aus [17] . . . . .	32
3.2	PWM–Erzeugung . . . . .	33
3.3	Ideale Ankerspannung-Drehzahl-Kennlinie aus [17] . . . . .	34
3.4	Verhalten Drehzahlen zu PWM beim LAURON IVb . . . . .	35
3.5	Zeitverhalten . . . . .	36
3.6	Kennlinienlinearisierung aus [31] . . . . .	37
3.7	Drehzahlmessmethode aus [29] . . . . .	39
3.8	Modell des Ankerstromregelkreis . . . . .	41
3.9	Modell des Geschwindigkeitsregelkreis aus [27] . . . . .	42
3.10	Modell der Einzelgelenkregelung . . . . .	43

## *Abbildungsverzeichnis*

3.11	Abtastzeit im aperiodischen Fall aus [2] . . . . .	44
3.12	Wirkungsplan PI-Regler im geschlossenen Kreis . . . . .	48
3.13	Integration durch Bildung der Trapezfläche aus [26] . . . . .	49
3.14	Modell der überlagerten Kraf Regelung aus [29] . . . . .	56
4.1	Single Loop des DSP Programmes . . . . .	58
4.2	Z-Kraftwert . . . . .	61
5.1	Drehzahlregelung des ausgebauten Motors . . . . .	69
5.2	Drehzahlverhalten der Beta-Gelenk Regelstrecke . . . . .	70
5.3	Drehzahlverhalten der Lageregelung des Beta-Gelenk . . . . .	72
5.4	Drehzahlregelung des Beta-Gelenk . . . . .	73
5.5	Z-Komponente Kraftwert Glättung . . . . .	74

# Literaturverzeichnis

- [1] ALTERA. *FLEX 10K Embedded Programmable Logic Family*. <http://www.altera.com/literature/ds/dsf10k.pdf> April 2007
- [2] BERGER, Manfred: *Grundkurs der Regelungstechnik*. 1. Books on Demand GmbH, 2001. – ISBN 3-8311-0847-1
- [3] CONSORTIUM, PC/104. *PC/104 Spezifikationen*. <http://www.pc104.org/> April 2007
- [4] FAULHABER: *DC-Micromotors 16mNm, Series 2343 ... CR*. 1. <http://www.faulhaber-group.com>: FaulhaberGroup, März 2007. – [http://www.faulhaber-group.com/uploadpk/e\\_2342CR\\_DFF.pdf](http://www.faulhaber-group.com/uploadpk/e_2342CR_DFF.pdf)
- [5] FAULHABER: *Encoders, Serie IE2 - 512*. <http://www.faulhaber-group.com>: FaulhaberGroup, März 2007. – [http://www.faulhaber-group.com/uploadpk/e\\_IE2512\\_DFF.pdf](http://www.faulhaber-group.com/uploadpk/e_IE2512_DFF.pdf)
- [6] FAULHABER: *Planetary Gearheads 0,7 Nm, Serie 23/1*. <http://www.faulhaber-group.com>: FaulhaberGroup, März 2007. – [http://www.faulhaber-group.com/uploadpk/e\\_231\\_MIN.pdf](http://www.faulhaber-group.com/uploadpk/e_231_MIN.pdf)
- [7] FELLMANN, Jörg. *Rechnerarchitektur und Realisierung grundlegender Bewegungsabläufe für einen zweibeinigen Laufroboter*. April 2007
- [8] FÖLLINGER, Otto: *Lineare Abtastsysteme - Methoden der Regelungstechnik*. 4. Auflage. München : Oldenbourg Verlag GmbH, 1990. – ISBN 3-486-21580-9
- [9] FÖLLINGER, Otto: *Regelungstechnik - Einführung in die Methoden und ihre Anwendung*. 6. Auflage. Heidelberg : Hüthing Buch Verlag, 1990. – ISBN 3-7785-1808-9
- [10] FREESCALE ; MOTOROLA: *DSP56F801-7 User Manual*. <http://www.freescale.com/>: Freescale Semiconductor, März 2007. – DSP56F801-7UM.pdf
- [11] FREESCALE ; MOTOROLA: *DSP56F80x/DSP56F82x Family Targeting Manual*. [www.freescale.com/files/soft\\_dev\\_tools/doc/user\\_guide/Targeting\\_56800.pdf](http://www.freescale.com/files/soft_dev_tools/doc/user_guide/Targeting_56800.pdf): Freescale Semiconductor, März 2007

## Literaturverzeichnis

- [12] FREESCALE ; MOTOROLA: *Embedded SDK, Digital Signal Processing (DSP) Function Library.* <http://www.freescale.com/>: Freescale Semiconductor, März 2007.  
– sdk107.pdf
- [13] FZI. *Lauron I Modell.* <http://earth.kwangwoon.ac.kr/Robots/Lauron.jpg> April 2007
- [14] FZI. *Lauron IVc Modell.* <http://wcms1.rz.tu-ilmenau.de/fakmb/typo3temp/d1a2fddefb.jpg> April 2007
- [15] FZI, Karlsruhe. *Forschungszentrum für Informatik in Karlsruhe.* <http://www.fzi.de/> März 2007
- [16] GASSMANN, Bernd: *Erweiterung einer modularen Laufmaschinensteuerung für unstrukturiertes Gelände.* Karlsruhe, Forschungszentrum Informatik - Universität Karlsruhe (TH), Diplomarbeit, September 2000
- [17] IHME, Thomas: *Steuerung von sechsbeinigen Laufrobotern unter dem Aspekt technischer Anwendungen.* Magdeburg, Otto-von-Guericke-Universität Magdeburg, Dissertation, März 2002
- [18] ISAACS, Andrew ; JACKSON, Gregory: *Autoranging Magnetic Field Meter for 3-Dimensional Elliptically Polarized Fields,* Faculty of Engineering of the University of Manitoba, Final report for the degree of Bachelor of Science, Spring 2003
- [19] ISHIDA, K. ; NAKAMURA, K. ; IZUMI, T. ; OHARA, M.: Microprocessor Control of Convertor-fed DC Motor Drives. In: *IEEE-IAS 17th Annual Meetings' Recording.* San Francisco, 4-7 Oktober 1982, S. 619–623
- [20] K.-U. SCHOLL, J. A. ; GASSMANN, B.: MCA - An Expandable Modular Controller Architecture.
- [21] LANGE, Friedrich: *Adaptiv vorausplanende Steuerung für schnelle sensorbasierte Roboterbewegung.* Karlsruhe, Universität Karlsruhe (TH), Dissertation, Februar 2003
- [22] LIPPERT. *Cool RoadRunner III PC/104-Plus CPU board.* [http://www.lippert-at.de/fileadmin/lippert-at/products/PC\\_104-Plus/Coo%\\_l\\_Roadrunner\\_III/TME-104P-CRR3-R1V11.pdf](http://www.lippert-at.de/fileadmin/lippert-at/products/PC_104-Plus/Coo%_l_Roadrunner_III/TME-104P-CRR3-R1V11.pdf) April 2007
- [23] LIPPERT. *Industrie PC/104.* <http://www.lippert-at.com/> April 2007
- [24] LIU, S.: Drehzahlregelung eines Gleichstrommotors kleiner Leistung / Universität Kaiserslautern, Lehrstuhl für Regelungssysteme. 2006. – Forschungsbericht
- [25] MERZ, Ludwig ; JASCHEK, Hilmar: *Grundkurs der Regelungstechnik - Einführung in die praktische und theoretische Methoden.* 14. Auflage. München : Oldenbourg Wissenschaftsverlag GmbH, 2003. – ISBN 3-486-25960-1

## Literaturverzeichnis

- [26] PHILIPPSEN, Hans-Werner: *Einstieg in die Regelungstechnik - Vorgehensmodell für den praktischen Reglerentwurf.* Leipzig : Fachbuchverlag Leipzig im Carl Hanser Verlag, 2004. – ISBN 3–446–22377–0
- [27] ROBOTERNETZ. *Regelungstechnik.* <http://www.roboternetz.de/wissen/index.php/Regelungstechnik> März 2007
- [28] SCHOLL, Kay-Ulrich ; FZI. *Modular Controller Architecture.* <http://www.mca2.org/> März 2007
- [29] SKLYARENKO, Yevgen G.: *Anwendung neuronaler Netze zur Regelung von nicht-linearen Roboterantrieben.* Magdeburg, Otto-von-Guericke-Universität Magdeburg, Dissertation, März 2002
- [30] WANGEMANN, Jörg: *Entwurf und Implementierung eines Beinreglers für den Schreitroboter LAURON III.* Magdeburg, Otto-von-Guericke-Universität Magdeburg, Studienarbeit, August 2005
- [31] WETZELSBERGER, Kai: *Konzeption und Implementierung von sensorbasiertem Bewegungsalgorithmen für einen Laufroboter.* Mannheim, Hochschule Mannheim - Hochschule für Technik und Gestaltung, Diplomarbeit, September 2005
- [32] WIKIPEDIA. *Kinematik.* <http://de.wikipedia.org/wiki/Kinematik> 2007
- [33] WLOKA, Dieter: *Robotersysteme 1 - Technische Grundlagen.* Berlin Heidelberg : Springer Verlag, 1992. – ISBN 3–540–54739–8
- [34] ZENZ, Christian ; GRÖSSL, Martin. *Kraftmess-Sensor Lauron IVb.* August 2006