

# **Technische Universität Dresden**

Fakultät Elektrotechnik und Informationstechnik

Institut für Regelungs- und Steuerungstheorie

## **Studienarbeit**

### **Konzeption, Aufbau und Inbetriebnahme eines selbststabilisierenden einachsigen Fahrzeuges**

vorgelegt von: Bolorkhuu Dariimaa

geboren am: 26. Juni 1989 in Choibalsan

Betreuer: Dipl.-Ing C. Knoll  
Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack  
Tag der Einreichung: 11. September 2014

**Aufgabenstellung zur Studienarbeit**  
für Herrn cand. ing. Bolorkhuu Dariimaa

**Konzeption, Aufbau und Inbetriebnahme eines  
selbststabilisierenden einachsigen Fahrzeuges**

Am Institut für Regelungs- und Steuerungstheorie werden nichtlineare unteraktuierte mechanische Systeme aus Regelungstechnischer Sicht untersucht. Ein Beispiel hierfür ist ein Fahrzeug mit nur einer Achse. Die konstruktiv bedingte Instabilität (Schwerpunkt oberhalb der Achse) muss dabei durch eine geeignete Regelung kompensiert werden. Ziel der Studienarbeit ist die Realisierung eines funktionsfähigen Prototyps eines solchen einachsigen Fahrzeuges.

Im Einzelnen sind dazu folgende Teilaufgaben zu bearbeiten:

- Herleitung der Bewegungsgleichungen und Implementierung eines Simulationsmodells
- Entwicklung eines Regelungskonzeptes zur Stabilisierung einer instabilen Ruhelage
- Konzeption des mechanischen Aufbaus
- Konzeption und Auslegung der elektrischen und elektronischen Komponenten (Motoren, Leistungselektronik, Sensorik, Steuerungselektronik)
- Aufbau des Prototyps
- Implementierung der Regelung auf dem Prototyp
- Dokumentation der Ergebnisse

Betreuer:

Dipl.-Ing. C. Knoll

Verantwortlicher Hochschullehrer:

Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack

Bearbeitungszeitraum:

01.04.2014 – 31.07.2014



Dresden, den 28. März 2014

Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack

# **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die von mir am heutigen Tage an der Fakultät Elektrotechnik und Informationstechnik eingereichte Studienarbeit zum Thema

## **Konzeption, Aufbau und Inbetriebnahme eines selbststabilisierenden einachsigen Fahrzeuges**

selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen sind, wurden als solche kenntlich gemacht.

Dresden, 11. September 2014

Bolorkhuu Dariimaa

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Gliederung . . . . .	1
<b>2 Modellbildung, Simulationsmodell, Regelungskonzept</b>	<b>2</b>
2.1 Modellbildung . . . . .	2
2.2 Aufstellen der Bewegungsgleichungen . . . . .	3
2.2.1 Partielle Linearisierung . . . . .	5
2.2.2 Linearisierung um die Ruhelage . . . . .	5
2.3 Regelungskonzept . . . . .	6
2.4 Simulationsmodell . . . . .	7
2.5 Modellbildung, Reglerentwurf des Motors . . . . .	8
2.5.1 Modellbildung . . . . .	8
2.5.2 Reglerentwurf . . . . .	8
<b>3 Mechanischer Aufbau und elektrische Komponenten</b>	<b>10</b>
<b>4 Umsetzung und Messergebnisse</b>	<b>12</b>
4.1 Parameteridentifikation . . . . .	12
4.1.1 Bestimmung des Trägheitsmoments des inversen Pendels . . . . .	12
4.1.2 Bestimmung des Trägheitsmoments der Räder . . . . .	13
4.1.3 Parameteridentifikation der Motoren . . . . .	13
4.2 Auswertung des Inkrementalgebers . . . . .	16
4.3 Softwareumsetzung . . . . .	17
4.4 Software-Aufbau . . . . .	18
4.5 Ergebnisse . . . . .	19
<b>5 Zusammenfassung und Ausblick</b>	<b>20</b>
<b>6 Anhang</b>	<b>21</b>
<b>Literaturverzeichnis</b>	<b>25</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Ein inverses Pendel [21] ist eine der Standardaufgaben aus der Regelungstechnik. Sein Schwerpunkt befindet sich oberhalb der Achse. Ohne eine Regelung ist das Pendel ein instabiles System.

Es ist ein sehr anschauliches Beispiel der Regelungstechnik und deshalb oft in der Literatur zu finden, z. B. in [17], [14]. Die Realisierung erfolgt meist mit einer Schiene. Davon existieren zahlreiche Versuchsstände für die Lehre z.B. an den Universitäten Dresden [1] und Wien [2]. Durch die Schiene ist das System groß, schwer, teuer, und unhandlich. Für eine Demonstration im Hörsaal ist dieses System nicht gut geeignet.

Durch die Entwicklung von Gyroskopen und Beschleunigungssensoren ist es aber inzwischen möglich ein ähnlich anschauliches System zu realisieren, was ohne Schiene auskommt. Thema der vorliegenden Arbeit ist deshalb die Umsetzung eines einachsigen Fahrzeuges mit hohem Schwerpunkt inklusive funktionsfähiger Regelung, siehe Bild 16. Seine Grundfunktionalität findet sich in der praktischen Anwendung in vielen Bereichen. Beispielsweise in der Steuerung von Segways oder humanoider Roboter.

Das Ziel der Regelung bei dem inversen Pendel ist eine Stabilisierung einer Stange in aufrechter Position. Diese Stabilisierung erfolgt durch die Steuerung der Räder mittels geeigneter Vorgabe des Drehmoments, erzeugt durch einen Elektromotor mit pulsweitenmodulierter Eingangsspannung. In dieser vorliegenden Arbeit werden frei verfügbare Python-Bibliotheken (PySerial [13], SciPy [11] darunter NumPy, SymPy, Matplotlib usw.) als Hilfsmittel benutzt.

### 1.2 Gliederung

Im Kapitel 2 werden Bewegungsgleichungen des Systems hergeleitet. Daraus werden die partielle sowie Jakobi-Linearisierung, der Reglerentwurf und die Simulation des Systems geschlussfolgert. Kapitel 3 beschreibt den mechanischen und elektrischen Aufbau des Prototyps. Mit der Identifikation der Regelstrecke, der Parametrierung des Reglers und der Umsetzung der Software, sowie den Ergebnisse beschäftigt sich Kapitel 4. Mit der Zusammenfassung und dem Ausblick endet die Arbeit in Kapitel 5.

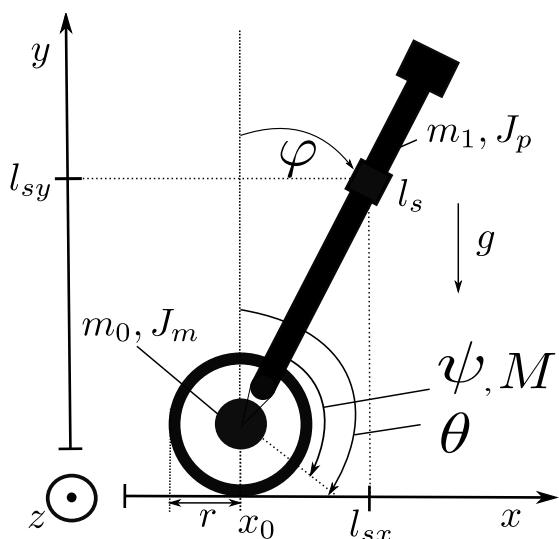
# Kapitel 2

## Modellbildung, Simulationsmodell, Regelungskonzept

Dieses Kapitel umfasst die Modellbildung und das Simulationsmodell, sowie das Regelungskonzept des inversen Pendels.

### 2.1 Modellbildung

Um das Verhalten des inversen Pendels hinreichend genau wiedergeben und untersuchen zu können, wird ein vereinfachtes Modell gebildet welche die Herleitung der Bewegungsgleichungen erheblich erleichtert.



$m_0$	...	Rädermassen
$m_1$	...	Pendelmasse
$J_m$	...	Radträgheitsmoment
$J_p$	...	Pendelträgheitsmoment um den Schwerpunkt
$l_{sx}, l_{sy}$	...	Koordinaten des Pendelschwerpunkts $l_s$
$x_0$	...	zurückgelegter Weg des Rades
$r$	...	Radius des Rades
$g$	...	Erdbeschleunigung
$\varphi$	...	Absolutwinkel des Pendels
$\theta$	...	Rollwinkel des Rades
$\psi$	...	Relativwinkel zwischen Rad und Pendel
$M$	...	im $\psi$ -Gelenk eingeprägter Drehmoment

Abbildung 1 – Schematische Darstellung des inversen Pendels.

Es werden folgende Annahmen getroffen:

- Das inverse Pendel besteht aus einem Antriebssystem (zwei Räder und zwei Motoren)
- Die Bewegung des Systems verläuft nur in der  $x$ - $y$ -Ebene
- Die Masse des Pendels sei um die Drehachse homogen verteilt
- Die Räder werden in  $x$ - $y$ -Ebene wie ein Rad in Abbildung 1 betrachtet
- Die Räder sind mit der Motorwelle starr gekoppelt (kein Getriebespiel)
- Die Räder können nicht gleiten

In der Umsetzung werden die Annahmen nur näherungsweise erreicht.

## 2.2 Aufstellen der Bewegungsgleichungen

Für die Regelungstechnische Umsetzung werden die Bewegungsgleichungen des inversen Pendels benötigt, die mit Hilfe der LAGRANGE-Formalismus hergeleitet werden können. Als Koordinaten werden  $\psi, \varphi, \theta$  benutzt. Die holome Zwangsbedingung ist

$$x = r \theta = r(\psi + \varphi). \quad (2.1)$$

Somit verfügt das Pendel über den Freiheitsgrad  $f = 2$ . Auf dem Schema 1 deutlich zu sehen sind die Freiheitsgrade in  $x$ -Richtung sowie um  $z$ -Achse. Die Kinematik des inversen Pendels lässt sich durch den Relativwinkel  $\psi$  und den Absolutwinkel  $\varphi$  als generalisierte Koordinaten beschreiben, d.h.

$$[\psi \ \varphi]^T =: [q_1 \ q_2]^T =: \mathbf{q}. \quad (2.2)$$

Für die Schwerpunktkoordinaten des Pendels erhält man

$$\begin{aligned} l_{sx} &= r(q_1 + q_2) + l_s \sin(q_2) \\ l_{sy} &= \phantom{r(q_1 + q_2) + } l_s \cos(q_2) . \end{aligned} \quad (2.3)$$

Mit ihnen ergibt sich die kinetische Energie  $T$

$$\begin{aligned} T(\mathbf{q}, \dot{\mathbf{q}}) = & m_0 r^2 \frac{(\dot{q}_1 + \dot{q}_2)^2}{2} + J_m \frac{(\dot{q}_1 + \dot{q}_2)^2}{2} + J_p \frac{\dot{q}_2^2}{2} + \\ & + \frac{m_1}{2} \left[ \left( \frac{d}{dt} \underbrace{(r(q_1 + q_2) + l_s \sin(q_2))}_{l_{sx}} \right)^2 + \left( \frac{d}{dt} \underbrace{(l_s \cos(q_2))}_{l_{sy}} \right)^2 \right] \end{aligned} \quad (2.4)$$

und die potenzielle Energie  $U$

$$U(\mathbf{q}) = m_1 g l_{sy} = m_1 g l_s \cos(q_2). \quad (2.5)$$

LAGRANGE-Funktion  $L$  wird gebildet aus der Differenz von kinetischer und potenzieller Energie [10]

$$L = T - U. \quad (2.6)$$

Damit lassen sich die LAGRANGE-Gleichungen

$$\frac{d}{dt} \left( \frac{dL}{d\dot{q}_i} \right) - \frac{dL}{dq_i} = \tau_i \quad i = 1, 2 \quad (2.7)$$

aufstellen. Konkret erhält man

$$\begin{aligned} L = & m_0 r^2 \frac{(\dot{q}_1 + \dot{q}_2)^2}{2} + J_m \frac{(\dot{q}_1 + \dot{q}_2)^2}{2} + J_p \frac{\dot{q}_2^2}{2} - m_1 g l_s \cos(q_2) + \\ & + \frac{m_1}{2} \left[ \left( \frac{d}{dt} (r(q_1 + q_2) + l_s \sin(q_2)) \right)^2 + \left( \frac{d}{dt} (l_s \cos(q_2)) \right)^2 \right]. \end{aligned} \quad (2.8)$$

Die Bewegungsgleichungen in der Matrizen-Darstellung lauten

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.9)$$

mit der positiv definiten Massenmatrix

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &:= \begin{bmatrix} M_{11}(\mathbf{q}) & M_{12}(\mathbf{q}) \\ M_{21}(\mathbf{q}) & M_{22}(\mathbf{q}) \end{bmatrix} := \\ &= \underbrace{\begin{bmatrix} J_m + (m_0 + m_1)r^2 & J_m + (m_0 + m_1)r^2 + l_s m_1 r \cos(q_2) \\ J_m + (m_0 + m_1)r^2 + l_s m_1 r \cos(q_2) & J_m + J_p + (m_0 + m_1)r^2 + m_1 l_s^2 + 2l_s m_1 r \cos(q_2) \end{bmatrix}}_{\text{Massenmatrix}}. \end{aligned} \quad (2.10)$$

Die Zentrifugal- bzw. Corioliskräfte werden durch die Masse  $m_1$  der Stange, dem Schwerpunktabstand  $l_s$  und dem Absolutwinkel  $\varphi$  bzw.  $q_2$  bestimmt

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) := \begin{bmatrix} C_1(\mathbf{q}, \dot{\mathbf{q}}) \\ C_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} := \underbrace{\begin{bmatrix} -l_s m_1 \dot{q}_2^2 r \sin q_2 \\ -l_s m_1 \dot{q}_2^2 r \sin q_2 \end{bmatrix}}_{\text{Zentrifugal-/Corioliskräfte}}. \quad (2.11)$$

Die Trägheitskraft wirkt allein auf die Masse  $m_1$  der Stange:

$$\mathbf{K}(\mathbf{q}) := \begin{bmatrix} K_1(\mathbf{q}) \\ K_2(\mathbf{q}) \end{bmatrix} := \underbrace{\begin{bmatrix} 0 \\ -g l_s m_1 \sin(q_2) \end{bmatrix}}_{\text{Gelenkmomente durch Gravitation}}, \quad \boldsymbol{\tau} := \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} := \underbrace{\begin{bmatrix} M \\ 0 \end{bmatrix}}_{\text{Antriebsgelenkmoment}} \quad (2.12)$$

$$\begin{bmatrix} M_{11}(\mathbf{q}) & M_{12}(\mathbf{q}) \\ M_{21}(\mathbf{q}) & M_{22}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_1(\mathbf{q}, \dot{\mathbf{q}}) \\ C_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} K_1(\mathbf{q}) \\ K_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}. \quad (2.13)$$

### 2.2.1 Partielle Linearisierung

Wie in (2.9) gezeigt, können die nichtlinearen Bewegungsgleichungen partiell linearisiert werden [12]. Als neuer (virtueller) Eingang  $a$  wird  $\ddot{q}_1$  gewählt. Dadurch ergibt sich die unterlagerte Rückführung

$$\begin{aligned}\tau_1 = & \left[ M_{11}(\mathbf{q}) - M_{12}(\mathbf{q})M_{22}^{-1}(\mathbf{q})M_{21}(\mathbf{q}) \right] a - M_{12}(\mathbf{q})M_{22}^{-1}(\mathbf{q})(C_2(\mathbf{q}, \dot{\mathbf{q}}) + K_2(\mathbf{q}, \dot{\mathbf{q}})) \\ & + C_1(\mathbf{q}, \dot{\mathbf{q}}) + K_1(\mathbf{q}).\end{aligned}\quad (2.14)$$

Somit führt das zu einem partial linearisierten System:

$$\begin{aligned}\ddot{q}_1 &= a \\ \ddot{q}_2 &= -M_{22}^{-1}(\mathbf{q})(C_2(\mathbf{q}, \dot{\mathbf{q}}) + K_2(\mathbf{q}, \dot{\mathbf{q}}) + M_{21}(\mathbf{q})a).\end{aligned}\quad (2.15)$$

Mit dem Zustandsvektor im  $\mathbb{R}^4$

$$\mathbf{x} := [x_1 \ x_2 \ x_3 \ x_4]^T := [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2]^T \quad (2.16)$$

und der Eingangsgröße  $u = a$  wird das eingangsaffine System

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \quad (2.17)$$

in eine nichtlineare Regelungsnormalform

$$\begin{aligned}\dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= a \\ \dot{x}_4 &= -M_{22}^{-1}(\mathbf{x})(C_2(\mathbf{x}, \dot{\mathbf{x}}) + K_2(\mathbf{x}, \dot{\mathbf{x}}) + M_{21}(\mathbf{x})a)\end{aligned}\quad (2.18)$$

transformiert. Aus der Normalform ist das Merkmal deutlich zu sehen, dass die Koordinaten miteinander über die Integratorkette zusammenhängen. Die Nichtlinearität tritt nur in der Ableitung der vierten Koordinate auf.

### 2.2.2 Linearisierung um die Ruhelage

Nach der partiellen Linearisierung werden die nichtlinearen Zustandsgleichungen um die Ruhelage  $\mathbf{x}_0 = [x_{1,0} \ x_{2,0} \ x_{3,0} \ x_{4,0}]^T$  für  $\mathbf{x}_0 = [0 \ 0 \ 0 \ 0]^T$  linearisiert, welche für das inverse Pendel einer aufrechten Position entspricht. Damit ergibt sich die Darstellung im Zustandsraum zu

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & a_{42} & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ b_4 \end{bmatrix} \tilde{u} \quad (2.19)$$

$$a_{42} = \frac{gl_s m_1}{J_p + J_m + l_s^2 m_1 + 2l_s m_1 r + (m_0 + m_1)r^2}$$

$$b_4 = -\frac{J_m + l_s m_1 r + (m_0 + m_1)r^2}{J_p + J_m + l_s^2 m_1 + 2l_s m_1 r + (m_0 + m_1)r^2}$$

bzw. in allgemeiner Darstellung

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A} \tilde{\mathbf{x}}(t) + \mathbf{b} \tilde{\mathbf{u}}, \quad \tilde{\mathbf{x}}(t_0) = \mathbf{x}_0. \quad (2.20)$$

## 2.3 Regelungskonzept

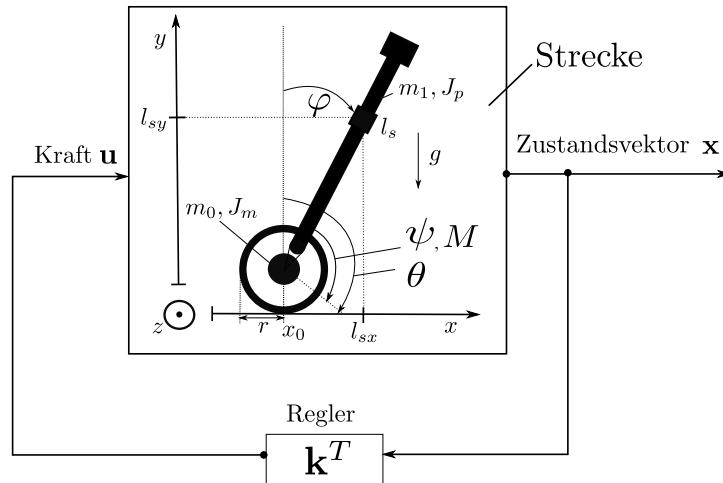
Die Anzahl der Zustandsgrößen des Systems beträgt insgesamt vier. D.h. man hat als Zielgröße mehr als eine Zustandsgröße zu regeln. Es wird eine Regelung der Mehrgrößen bzw. der Zustandsregler benötigt. Um einen Zustandsregler für das System auszulegen, muss das System zustandssteuerbar [17] sein. Um dies zu prüfen kann das Steuerbarkeitskriterium von Kalman oder Hautus angewendet werden. Die Kalmanische Steuerbarkeitsmatrix  $S$  ergibt sich zu

$$S = [\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \mathbf{A}^3\mathbf{b}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & b_4 & 0 & a_{42}b_4 \\ 1 & 0 & 0 & 0 \\ b_4 & 0 & a_{42}b_4 & 0 \end{bmatrix}. \quad (2.21)$$

Das System ist genau dann steuerbar [16], wenn:  $\det(S) \neq 0$ . Zur Prüfung der Steuerbarkeit wird die Determinante von  $S$  berechnet

$$\det(S) = -a_{42}^2 \cdot b_4^2, \quad \Rightarrow \text{Rg}(S) = 4. \quad (2.22)$$

Das System besitzt den vollen Rang. Somit ist das System zustandssteuerbar. Folglich muss sich das System bei beliebiger Radposition und aufrechter Stellung des Pendels um die instabile Ruhelage durch Zustandsrückführung stabilisieren lassen.



**Abbildung 2 – Inverses Pendel mit Zustandsrückführung**

Die Polstellen der Strecke (offenes System) ergeben sich aus

$$\det(s\mathbf{I} - \mathbf{A}) = 0 \quad (2.23)$$

zu

$$s_{1,2}^0 = 0$$

und

$$s_{3,4}^0 = \pm \sqrt{a_{42}}.$$

Aus den Eigenwerten sieht man, dass es eine positive Polstelle gibt. Das ungeregelte System ist also instabil. Durch eine geeignete Rückführung der Zustandsgrößen muss deshalb erreicht werden, dass alle Eigenwerte des geschlossenen Kreises in der linken Halbebene liegen.

## 2.4 Simulationsmodell

Im Folgenden wird die Simulation des Regelsystems durchgeführt.

Als gewünschte Pole des geregelten Systems werden folgende komplexe Zahlen gewählt:  
z. B.  $s_1 = s_2 = -5$ ,  $s_{3,4} = -3 \pm 3j$ .

Diesen Pol-Lagen entspricht das charakteristische Polynom

$$\text{CLCP} = (s + 5)^2 \cdot (s + 3 - 3j) \cdot (s + 3 + 3j). \quad (2.24)$$

Das geregelte System soll folgendes charakteristische Polynom [17] besitzen,

$$\text{CLCP} = \det(s\mathbf{I} - \mathbf{A} - \mathbf{b}(k_1, k_2, k_3, k_4)). \quad (2.25)$$

Die aus dem Abschnitt 4 gewonnenen Parameter werden in  $a_{42}$  und  $b_4$  eingesetzt. Man erhält  $a_{42} \approx 32.257$  und  $b_4 \approx -0.190$ . Aus dem im letzten Abschnitt beschriebenen Verfahren resultieren dann die Reglerverstärkungen  $\mathbf{k}^T = [k_1 \ k_2 \ k_3 \ k_4]^T = [13.95 \ 781.72 \ 10.23 \ 137.42]^T$ .

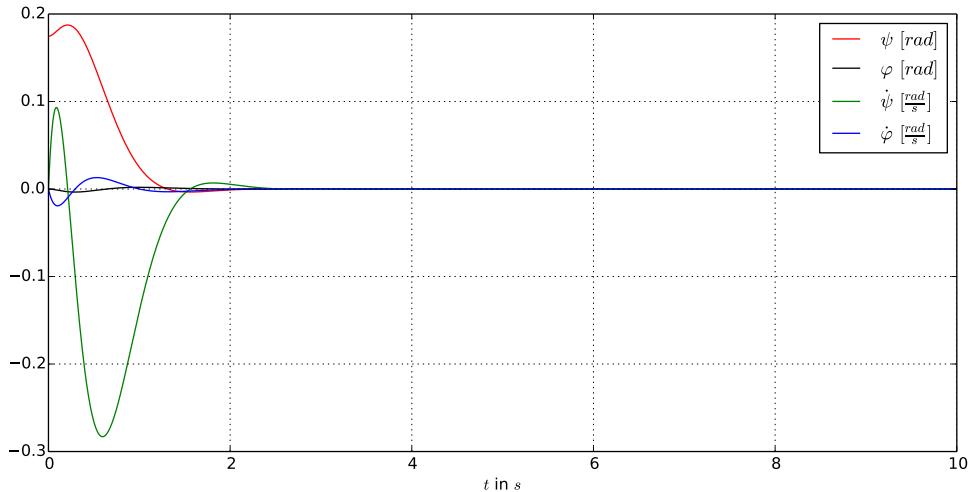


Abbildung 3 – geregeltes System bei Anfangslageabweichung  $\psi(t_0) = 10^\circ$

Je weiter die Polstellen in der linken Halbebene platziert werden, desto größer ist die benötigte Kraft und desto schneller ist die Regelung. Es wird vorausgesetzt, dass die resultierende Kraft  $\mathbf{u}$  nicht in die Sättigung gerät, da die verfügbare mechanische Leistung von Aktuatoren naturgemäß begrenzt ist. Aus der Abbildung 3 sieht man, dass die Regelung nach etwa zwei Sekunden die  $10^\circ$  bzw. ca 0.17rad Lageregelabweichung vollständig kompensiert hat.

## 2.5 Modellbildung, Reglerentwurf des Motors

Im vorherigen Abschnitt wird der Zustandsregler dimensioniert. Aus diesem wird die Sollbeschleunigung  $\ddot{\psi}$  berechnet. Eine separate Motorregelung bzw. unterlagerte PI-Regelung wird deshalb benötigt um die gewünschte Sollbeschleunigung zu gewährleisten.

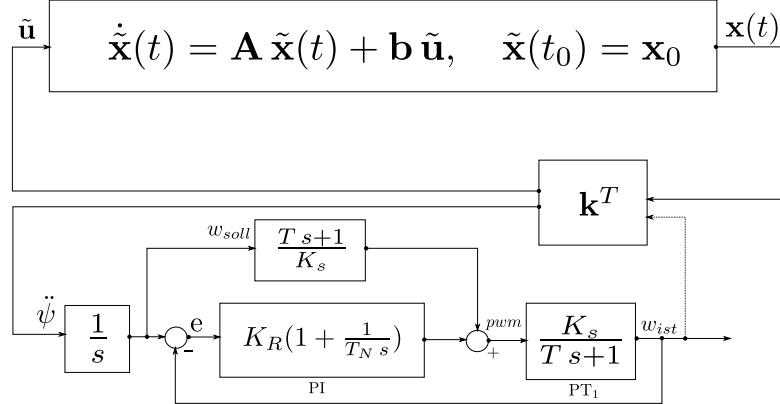


Abbildung 4 – Prinzipskizze des Gesamtregelkreises

### 2.5.1 Modellbildung

Ein Gleichstrommotor hat ein Übertragungsverhalten wie ein PT<sub>2</sub>-Glied [18]. Der Eingang ist eine Spannung bzw. die PWM. Der Ausgang ist die Winkelgeschwindigkeit des Motors. Wenn der Motor ein Getriebe mit hoher Übersetzung besitzt, kann dieser als ein PT<sub>1</sub>-Glied betrachtet werden, weil die Zeitkonstante des Trägheitsmomentes des Getriebes um ein Vielfaches höher ist als das des Motors. Es wird angenommen, dass der Pendel-Motor die selbe Eigenschaft hat. Deshalb ist die Übertragungsfunktion des Gleichstromgetriebemotors:

$$G(s) = \frac{K_s}{T s + 1}. \quad (2.26)$$

Ein PT<sub>1</sub>-Glied ist ein stabiles System. Die Polstelle liegt bei  $s = -\frac{1}{T}$ , wobei die  $T$ -Verzögerungszeit nicht negativ sein kann.

### 2.5.2 Reglerentwurf

Für das betrachtete PT<sub>1</sub>-Glied ist ein P- oder PI-Regler mit Vorsteuerung geeignet um die Verzögerungszeit  $T$  zu kompensieren und um der Sollgeschwindigkeitstrajektorie zu folgen. Ein P-Regler hat die Übertragungsfunktion:

$$K(s) = K_R. \quad (2.27)$$

Ein PI-Regler hat die Übertragungsfunktion:

$$K(s) = K_R(1 + \frac{1}{T_N s}). \quad (2.28)$$

Für die Parameter-Einstellung des Regler existieren zahlreiche Einstellregeln [18]. Um Drehgeschwindigkeit des Motors gezielt gegen Störung zu regeln, ist die Methode Symmetrisches Optimum [18] dafür gut geeignet. Nach Symmetrischem Optimum ist die Verstärkung des PI-Reglers:

$$K_R = \frac{1}{2 K_s} \quad (2.29)$$

und die Nachstellzeit:

$$T_N = 4 T. \quad (2.30)$$

# Kapitel 3

## Mechanischer Aufbau und elektrische Komponenten

Dieses Kapitel beschreibt den mechanischen Aufbau des inversen Pendels. Als Stellgröße soll die Spannung des Antriebssystems dienen. Die zu regelnden Größen sind die Position des Rades und die Position des Pendels. Diese können mit Hilfe des Beschleunigungssensors und des Inkrementalgebers gemessen werden. Deshalb wurden der Beschleunigungssensor [4] und der Inkrementalgeber bzw. die mit Hallsensoren bestückten Motoren [6] ausgesucht, da diese eine sehr genaue Empfindlichkeit und weitere gute Eigenschaften aufweisen. Für das Rahmenmaterial des Roboters werden folgende Anforderungen gestellt:

- einfach verarbeitbar
- günstig
- leicht
- gute mechanische Festigkeit
- gute elektromagnetische Verträglichkeit.

Plexiglas erfüllt genau diese Anforderungen. Davon werden drei Platten benötigt. Eine für den Batteriepack und zwei für die Steuerplatine 25. Die Plexiglasplatten sollen über Gewindestangen miteinander verbunden sein. An der untersten Platte sollen die Ansteuerplatine und die Sensoren, inklusive des Kreiselsensors, [19] welche die Winkelgeschwindigkeit  $\dot{\varphi}$  des Roboters misst, enthalten sein. Die Ansteuereinheiten können auf einer Lochrasterplatine aus Epoxidharz gelötet und über vier Abstandsbolzen mit der Plexiglasplatte festgeschraubt werden 23 24.

Die geschätzte Verlustleistung der elektrischen und mechanischen Komponenten des Roboters beträgt ca. 3W, wobei die mechanische Verlustleistung der zwei Motoren bereits beinhaltet ist. Dafür wird eine Akku oder ein Netzteil benötigt, welche die nötige elektrische Leistung zur Verfügung stellt. Deshalb wurde der 13V-Akku ausgesucht. Die Kapazität beträgt 3000mAh. Der Akku soll auf der obersten Platte angeordnet sein. Damit kann die Akkumasse den Trägheitsmoment des Pendels dominieren. Die Motoren können mit Aluwinkeln an der Bodenplatte befestigt werden. Es handelt sich dabei um Gleichstrom-Getriebemotoren mit 64-Segmenten Radencoder. Das Drehmoment beträgt

0.7767Nm [6]. Die Motoren sind einzeln an einen Motortreiber [7] angeschlossen. Dieser ist ausgelegt auf maximal 10A Spitzenstrom pro Motor, wobei die verwendeten Motoren bis zu 5A aufnehmen (bei Festhalten der Motorachse).

Als Steuereinheit kommt das STM32F4Discovery Board [20] zum Einsatz, da es Open Source Compiler und Linker Sammlung [8] als kostenlose Software für diesen Mikrokontroller zur Verfügung steht. Dieser Mikrokontroller ist einer der leistungsfähigsten und günstigsten, die es derzeit gibt. Der Prozessor verfügt über 1MB Flash, 512kByte RAM und eine mit 168Mhz getakteten ARM-Core, der Floating Point Operationen software- und hardwarebasiert durchführen kann. Festspannungsregler sorgen für eine konstante Eingangsspannung. Diese sind hintereinander angeordnet und stellen die Spannungspegel 12V und 5V sowie 3.3V für die Boardperipherie zur Verfügung. Um hochfrequente Störsignale zu vermeiden, müssen mindestens ein Elektrolytkondensator, sowie ein Keramikkondensator mit einer geringen Kapazität, als Tiefpass parallel vor und nach jedem Spannungsregler eingesetzt werden. Für den Verpolungsschutz wird eine einfacher Diode am Eingang der Vorsorgungsspannung benutzt.



**Abbildung 5**

Zur bidirektionalen Datenübertragung dienen zwei XBee-Module [5], welche eine serielle Punkt-zu-Punkt Kommunikation, entsprechend des Datenblattes über eine Strecke von bis zu ca. 1600m, mit einem Hostrechner ermöglichen. Das Modul verfügt über einen hohen Datendurchsatz von 1Mbps. Für die interne Signalübertragung der Steuerplatine werden elektrisch isolierte Kupferleitungen eingesetzt.

# Kapitel 4

## Umsetzung und Messergebnisse

Dieses Kapitel umfasst die (Hardware-, Software-) Umsetzung und Parameteridentifikation sowie Messergebnisse des geschlossenen Systems.

### 4.1 Parameteridentifikation

Die Parameteridentifikation beschreibt eine experimentelle Analyse des inversen Pendels. Anhand von Messungen in verschiedenen Experimenten kann das Systemmodell mittels mathematischer Methoden identifiziert werden. Damit kann die numerische Simulation und der modellbasierte Reglerentwurf erst ermöglicht werden. Die direkt messbaren Parameter sind beispielsweise die Masse und der Schwerpunkt des Pendels. Diese können ohne großen Aufwand gemessen werden. Die Masse des Pendels beträgt  $m_1 = 1.538 \text{ kg}$ . Der Schwerpunktabstand (Abstand zwischen Achse und Schwerpunkt) des Pendels liegt bei ca.  $l_s = 0.165 \text{ m}$ .

#### 4.1.1 Bestimmung des Trägheitsmoments des inversen Pendels

Bei dem inversen Pendel ist der Trägheitsmoment  $J_p$  nicht direkt messbar. Für die Identifizierung des Trägheitsmoments wird ein Schwingungsversuch durchgeführt.

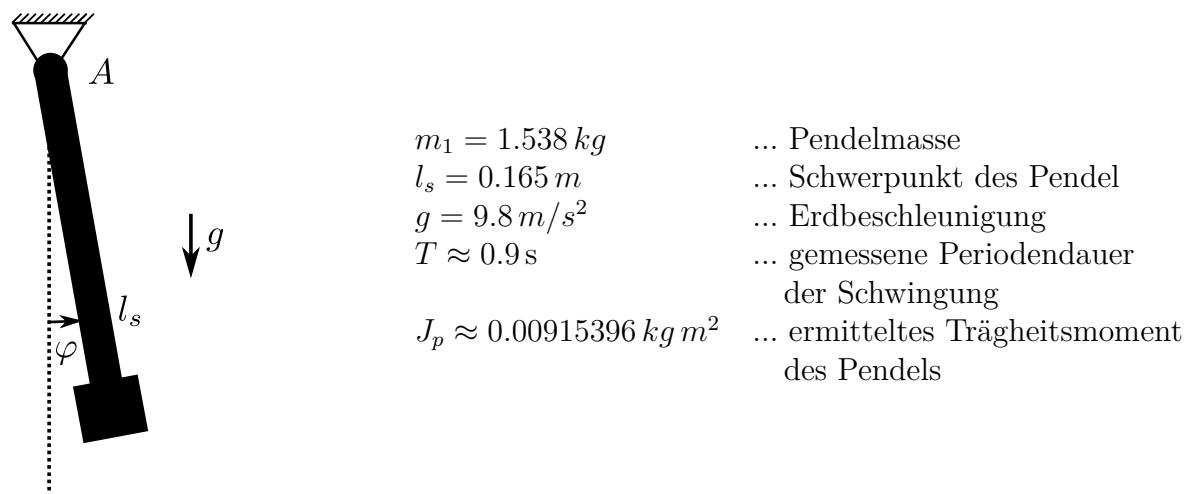


Abbildung 6 – Schematischer Aufbau des Schwingungsversuchs

Um die Drehachse  $A$  wird das inverse Pendel auf Abbildung 6 in Schwingung gesetzt, wobei eine vernachlässigbare Reibung um die Achse  $A$  vorausgesetzt wird. Das gesuchte Trägheitsmoment  $J_p$  ergibt sich zu

$$J_p = J_A - m_1 l_s^2. \quad (4.1)$$

Für den Schwingungsversuch wird die Bewegungsgleichung

$$J_A \ddot{\varphi} + l_s m_1 g \sin \varphi = 0 \quad (4.2)$$

aufgestellt. Um die stabile Ruhelage wird die nichtlineare Bewegungsgleichung (4.2) linearisiert. Für die kleine Auslenkung gilt:  $\sin(\varphi) \approx \varphi$ . Somit

$$\ddot{\varphi} + \underbrace{\frac{l_s mg}{J_A}}_{w_0^2} \varphi = 0, \quad T = \frac{2\pi}{\sqrt{\frac{mgl_s}{J_A}}} \quad (4.3)$$

bekommt man die linearisierte Bewegungsgleichung. Das unbekannte Trägheitsmoment  $J_p$  wird mittels der Gleichung

$$J_p = J_A - m_1 l_s^2 = \frac{l_s m_1 g T^2}{4\pi^2} - m_1 l_s^2 \quad (4.4)$$

ermittelt. Für die gemessene Periodendauer  $T \approx 0.9s$  der Schwingung erhält man  $J_p \approx 0.00915396 \text{ kg m}^2$ .

### 4.1.2 Bestimmung des Trägheitsmoments der Räder

Betrachtet man das Rad als sehr dünne Vollzylinder, ist das Trägheitsmoment  $J \approx \frac{1}{2}mr^2$ . Für die zwei Räder ergibt sich das Gesamtträgheitsmoment:

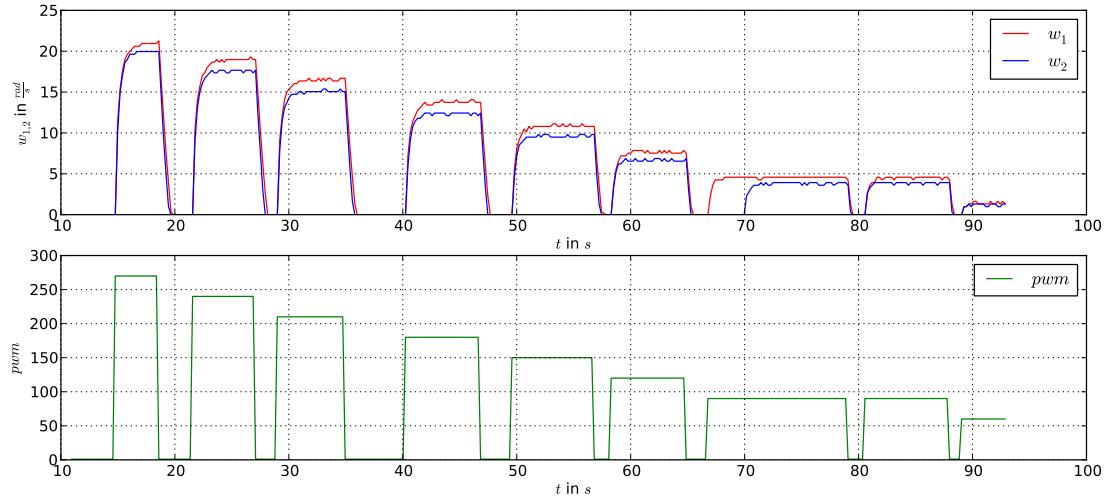
$$J_m = 2 \cdot J = mr^2 = 0.032 \text{ kg} \cdot 0.045^2 \text{ m}^2 = 6.48 \cdot 10^{-5} \text{ kg m}^2. \quad (4.5)$$

### 4.1.3 Parameteridentifikation der Motoren

Eine hinreichend genaue Identifizierung der Motorparameter wird für die Regelung der Motordrehzahl bzw. Motorwinkelgeschwindigkeit benötigt. Betrachtet man das Übertragungsverhalten des Motors als ein PT<sub>1</sub>-Glied. So erhält man folgendes Modell:

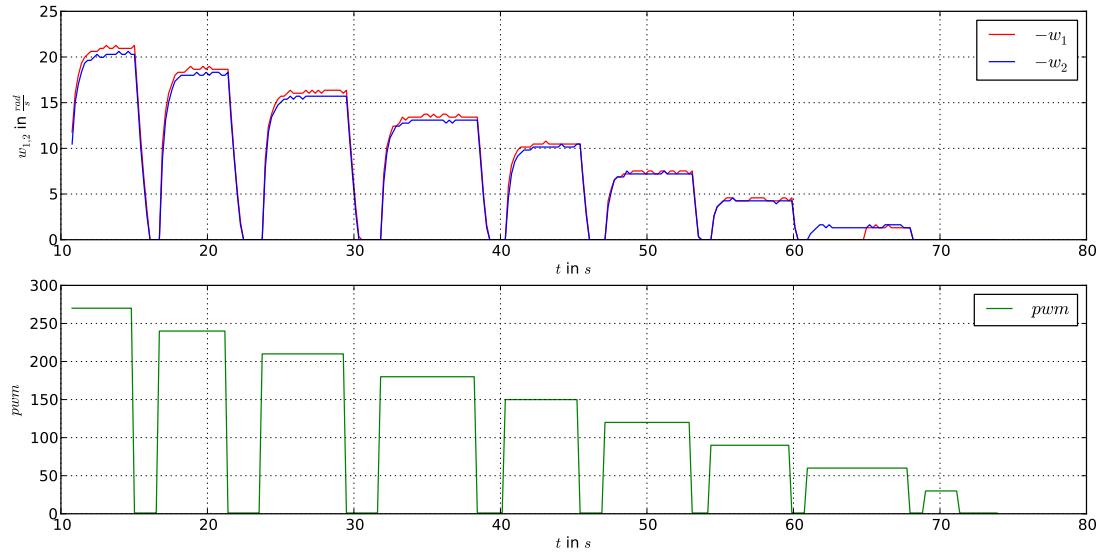
$$G(s) = \frac{K_s}{T s + 1}, \quad (4.6)$$

wobei die Eingangsgröße als pulsweitenmodulierte Spannung bzw. als PWM<sub>1,2</sub> (skaliert im Bereich 0 bis  $2^{10} - 1$ ) und die Ausgangsgröße als Motorwinkelgeschwindigkeit  $w_{1,2}$  festgelegt sind. Die gesuchten Parameter sind  $K_s$  und  $T$ . Dafür werden verschiedene Experimente durchgeführt bei denen am Eingang Sprungsignale mit unterschiedlichen Amplituden angelegt und die Ausgänge beobachtet werden.



**Abbildung 7** – Sprungantworten von Motoren bei verschiedenen Eingangstrepensignalen: Motorrichtung links. Rot-Motor<sub>1</sub>, Blau-Motor<sub>2</sub>

Aus Abbildung 7 ist zu erkennen, dass die beiden Motoren nicht identisch gebaut sind.



**Abbildung 8** – Sprungantworten von Motoren bei verschiedenen Eingangstrepensignalen: Motorrichtung rechts. Rot-Motor<sub>1</sub>, Blau-Motor<sub>2</sub>

Auch in die andere Drehrichtung, Abbildung 8, ergeben sich Unterschiede zwischen den Motoren. Jedoch sind sie weniger stark. Aus der Messung ergibt sich, dass die *pwm*-Verstärkungen nicht linear sind. Das liegt daran, dass das *pwm*-Signal bei den Motoren

jeweils unterschiedliche Offsets  $pwm_{10,20}$  besitzt. Dafür wurden folgende Stichproben aus den Messpunkten entnommen um die Übertragungsfunktion der  $pwm$ -Verstärkungen durch die Regressionskurve anzunähern.

$pwm$	$w_1[\frac{rad}{s}]$	$w_2[\frac{rad}{s}]$	
60	1.41	1.30	
90	4.57	3.9267	
120	7.51	6.5447	
150	10.79	9.49	
180	13.7	12.43	
240	18.95	17.67	

(4.7)

Aus den Stichproben (4.7) wurden für beide Motoren zwei unterschiedliche Funktionen mittels Regressionsanalyse bestimmt:

$$\begin{aligned} pwm_1 &= -6.600529101 \cdot 10^{-5} \cdot pwm^2 + 0.117868254 \cdot pwm - 5.4885 \\ pwm_2 &= -4.510582011 \cdot 10^{-7} \cdot pwm^2 + 0.091970912 \cdot pwm - 4.3036. \end{aligned} \quad (4.8)$$

Es wird mit einer quadratischen Funktion angenähert jedoch kann dies auch mit einer ungenauerer linearen Funktion erfolgen:

$$\begin{aligned} pwm_1 &= 9.80666 \cdot 10^{-2} \cdot pwm - 4.241 \\ pwm_2 &= 9.18355 \cdot 10^{-2} \cdot pwm - 4.295. \end{aligned} \quad (4.9)$$

Somit ist die Verstärkung des  $pwm$ -Signals konstant.

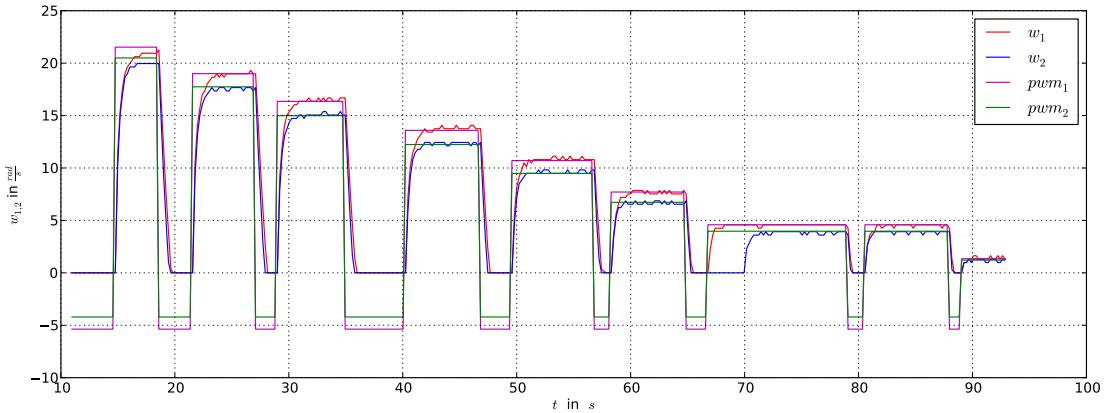
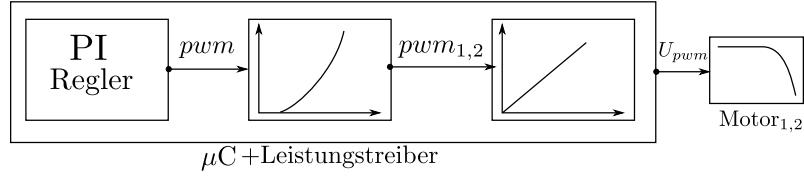
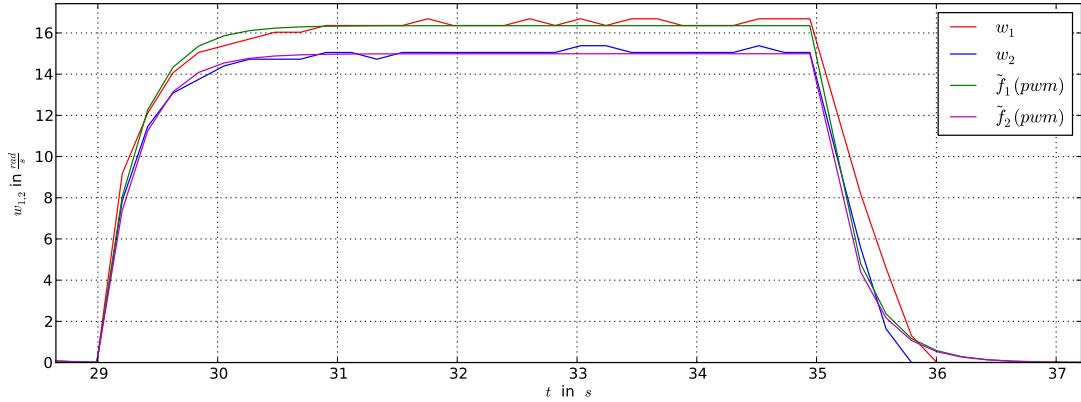


Abbildung 9 – Angenährte Funktion (4.8) von  $pwm_{1,2}$  mittels Regressionsanalyse

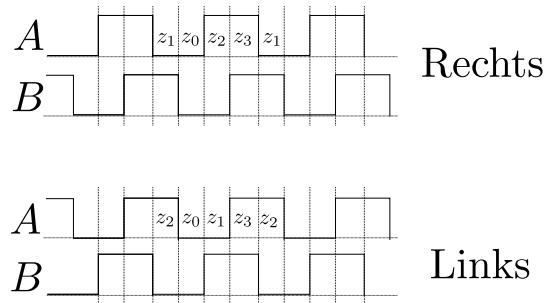
Zu erkennen ist, dass der stationäre Endwert für beide Motoren jeweils mit der Sprunghöhe übereinstimmt.


**Abbildung 10** – Prinzipskizze der PWM-Umrechnung

**Abbildung 11** –  $w_1$ ,  $w_2$ : Messungen,  $\tilde{f}_1$ ,  $\tilde{f}_2$ , angenäherte Funktionen

Aus Abbildung 11 geht hervor, dass die Verzögerungszeit  $T$  bei beiden Motoren gleich  $T = 0.3s$  ist.

## 4.2 Auswertung des Inkrementalgebers

Die Motoren besitzen jeweils einen Quadraturencoder, der zwei Ausgänge  $A$  und  $B$  hat. Die Ausgänge liefern ein  $\pm 90^\circ$  phasenverschobenes Signal an den Mikrokontroller. Für die Auswertung des Inkrementalgebers ist ein Moore-Automat 6 nötig, welcher möglichst kompakt sein sollte, damit nicht viel Rechenleistung verbraucht wird.


**Abbildung 12** – Signalfolgen für Rechts- und Links-Bewegung des Inkrementalgebers

Aus den Signalfolgen ist deutlich zu sehen, dass der Ereignistyp der Eingänge den Mengen

$$A \in \{0, 1\}, \quad B \in \{0, 1\} \quad (4.10)$$

zugeordnet werden kann. Aus den Schalterereignissen werden die Eingänge wie folgt kodiert:

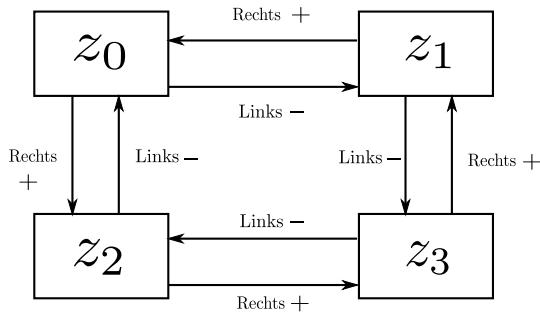
Zustand	A	B	Kodierung
$z_0$	0	0	0
$z_1$	0	1	1
$z_2$	1	0	2
$z_3$	1	1	3

Somit können die Zustandsfolgen für beide Richtungen erstellt werden.

Rechts:  $z_1 \rightarrow z_0 \rightarrow z_2 \rightarrow z_3 \rightarrow z_0 \dots$

Links :  $z_2 \rightarrow z_0 \rightarrow z_1 \rightarrow z_3 \rightarrow z_2 \dots$

Nun kann der Mooreautomat angegeben werden, siehe Abbildung 13.



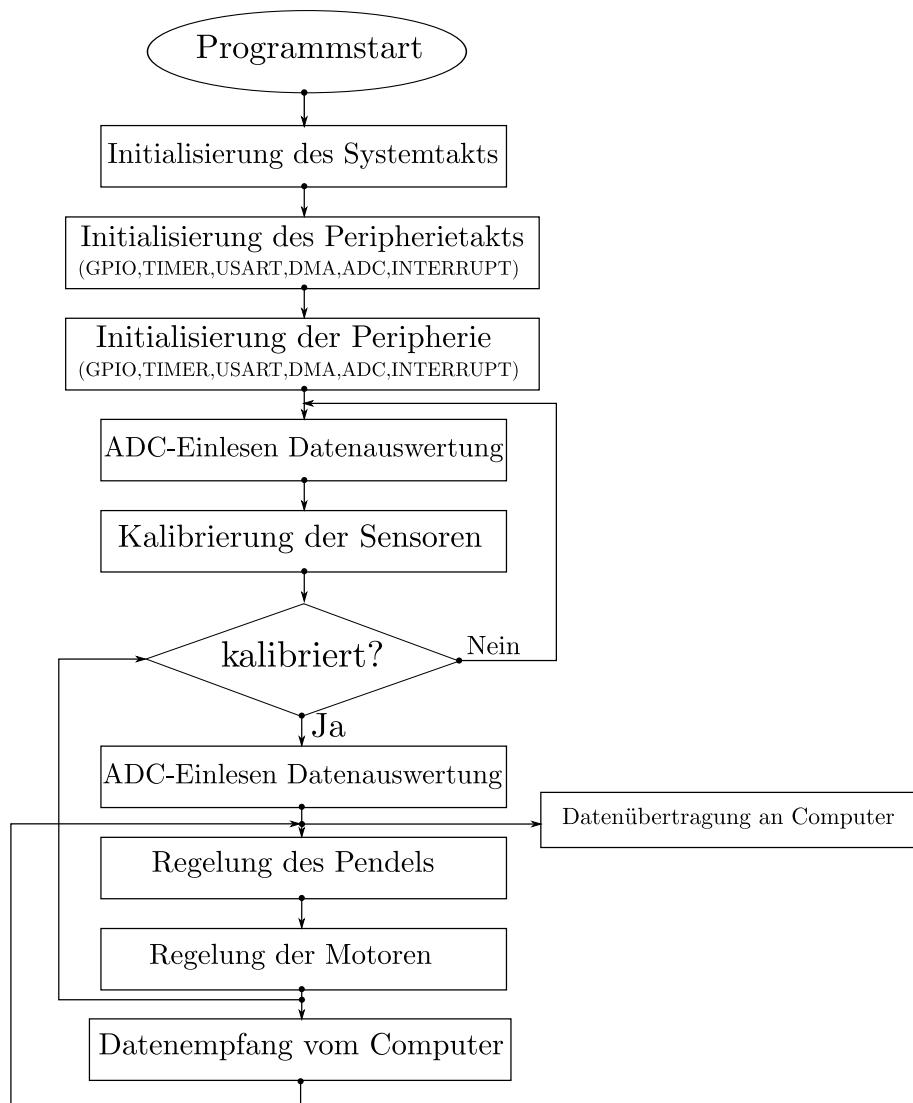
**Abbildung 13** – Mooreautomat,  $+$  ... Zähler inkrementieren,  $-$  ... Zähler dekrementieren

## 4.3 Softwareumsetzung

In diesem Kapitel wird die programmatische Umsetzung der Modelle gezeigt. Dafür wurde ein Programm mittels C-Programmierungssprache geschrieben, der auf dem STM32F4Discovery läuft und die Regelung übernimmt. Als Entwicklungsumgebung kam ein einfacher Texteditor (XCode und Sublime Text) zum Einsatz. Zudem wurde ein Makefile [22] erstellt, welches als Konfigurationsfile für den GNU GCC Compiler und den Linker [8], sowie für den Debugger OpenOCD [15] dient. Die Programmierung erfolgt über den boardeigenen Programmieradapter - STlinkv2. Für diesen existiert ein Open Source Treiber [9].

## 4.4 Software-Aufbau

Zunächst wird die gesamte Peripherie (SYSTEMTAKT, GPIO<sup>1</sup>, USART<sup>2</sup>, ADC<sup>3</sup>, TIMER, INTERRUPT, DMA<sup>4</sup>) initialisiert, siehe Abbildung 14. Danach werden die Sensoren kalibriert indem der Mittelwert aus 100 Samples gebildet wird. Daran schließt sich die Regelung des Pendels und die unterlagerte Regelung bzw. Motorgeschwindigkeitsregelung, welche mit einer Freqenz von 100Hz aufgerufen wird. In dieser werden zunächst die Sensoren eingelesen und daraus wird dann die Roboterpose bestimmt.



**Abbildung 14 – Programmablaufplan**

<sup>1</sup>General Purpose Input Output

<sup>2</sup>Universal Asynchronous Receiver Transmitter

<sup>3</sup>Analog Digital Converter

<sup>4</sup>Direct Memory Access

Diese Daten werden mittels DMA, zwecks Auswertung, an den Hostrechner gesendet. Desweiteren dienen die Daten als Eingabewerte für den Zustandsregler.

## 4.5 Ergebnisse

Die Abbildung 15 verglichen mit der Abbildung 3, sieht man darauf dass die Regelung in etwa zwei Sekunden die Lageabweichungen des Roboters quasi ausgleicht jedoch sind die Anfangsabweichungen bei beiden Abbildungen unterschiedlich.

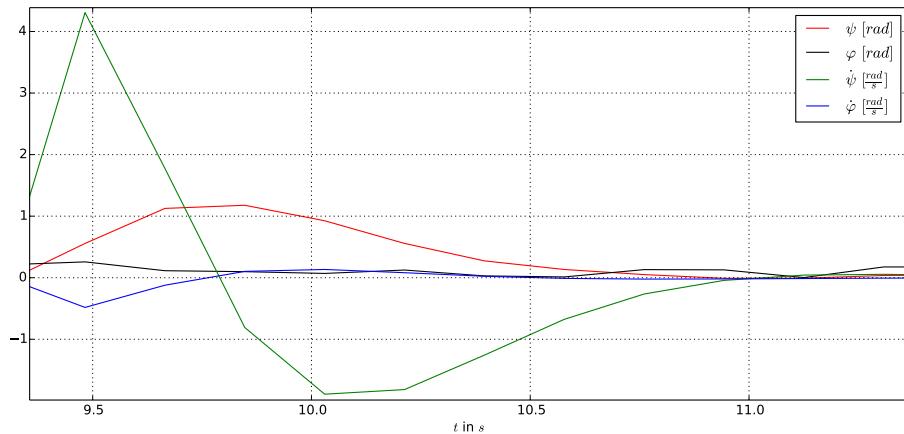


Abbildung 15 – Balanciervorgang gemessen, als Video [3] aufgenommen



Abbildung 16 – Roboter, balancierend

# Kapitel 5

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein funktionsfähiger einachsiger Roboter entwickelt, der mit Hilfe eines Regelkreises selbstständig balanciert. Im ersten Abschnitt wurde die Bewegungsgleichungen und das Regelungskonzept sowie die Simulation von dem System ausgearbeitet. In dem darauf folgenden Kapitel wurde das Lösungskonzept von dem mechanischen und elektrischen Aufbau des Roboters behandelt. Danach wurde der mechanische Aufbau vorgenommen. Im Kapitel 4 wird die Parameter-Identifizierung und Software des Roboters umgesetzt. Anschliessend wurde die Funktionstüchtigkeit des Roboters demonstriert.

Ein Bestand der zukünftigen regelungstechnischen Arbeit sind eine geregelte Lenkung, eine optimale Regelung der Stabilisierung und eine Folgeregelung der Solltrajektorie. Ein 3D-basierter Simulator des realen Systems und eine Abstrahierung der Software sowie eine Integration in das ROS<sup>1</sup> stellen einen Bedarf für den Roboter dar.

---

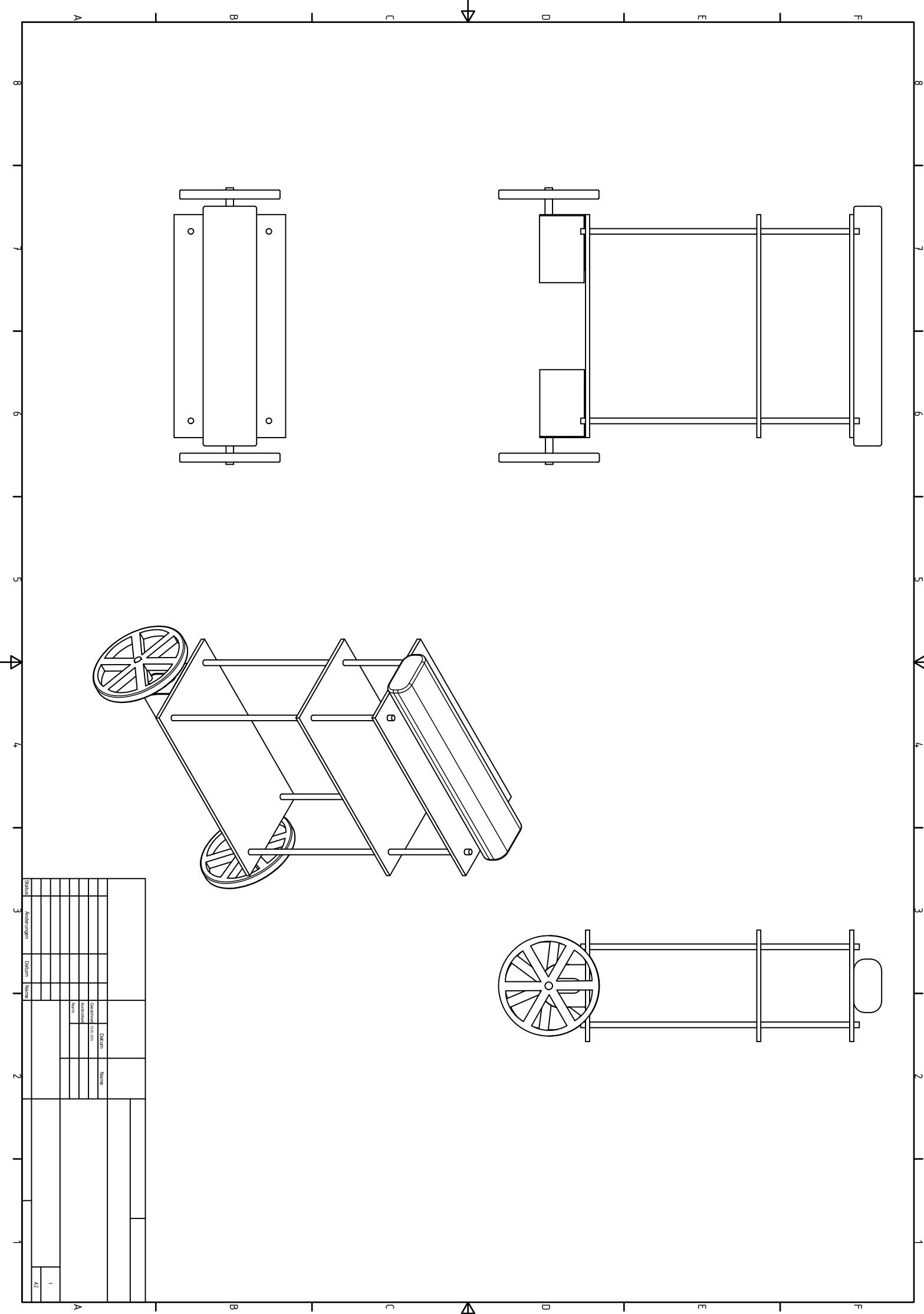
<sup>1</sup>Robot Operating System

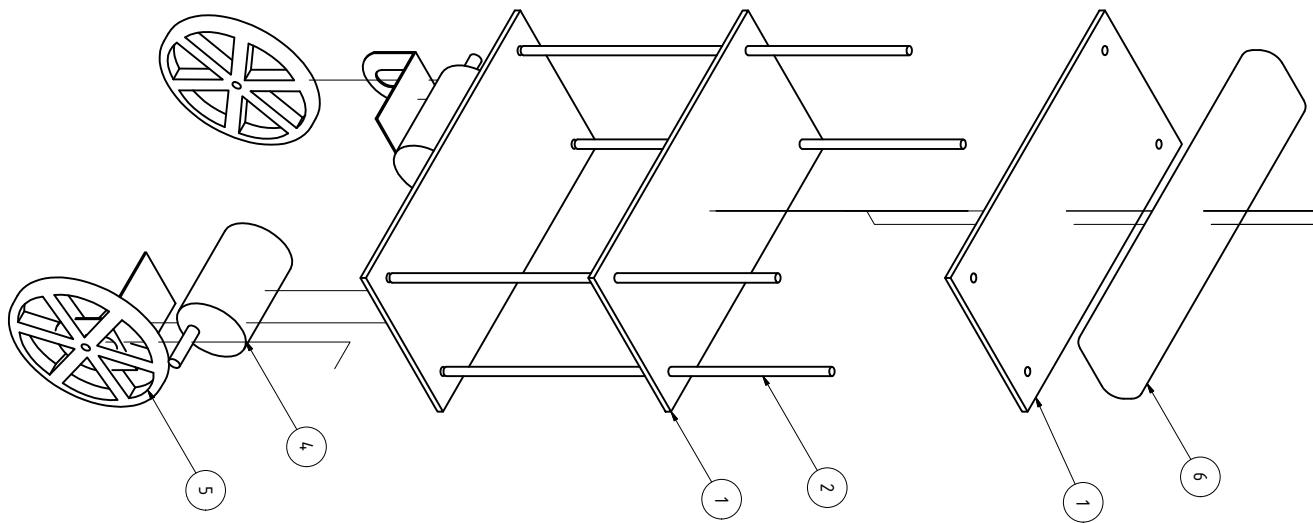
# Kapitel 6

## Anhang

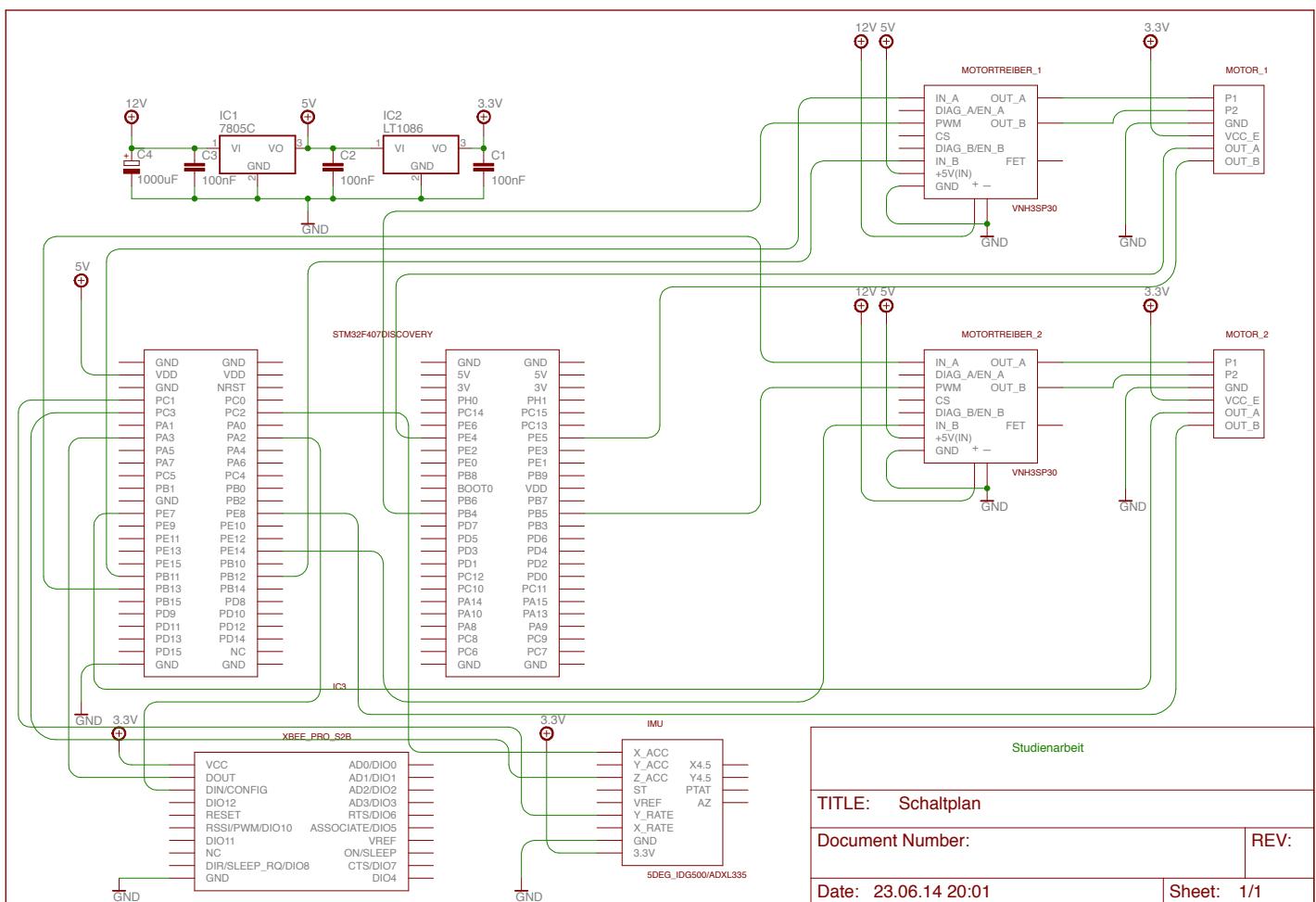
Die Auswertung des Inkrementalgebers in C-Code.

```
1 volatile int8_t old_state1 = 0;
2 volatile int8_t old_state2 = 0;
3
4 void read_encoder(void){
5
6     uint8_t A1 = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_8);
7     uint8_t B1 = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_9);
8
9     uint8_t A2 = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_10);
10    uint8_t B2 = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_11);
11
12    volatile int8_t state1 = (A1<<1)|B1;
13    volatile int8_t state2 = (A2<<1)|B2;
14
15    encoder_counter1=look_up_table[old_state1][state1];
16    encoder_counter2=look_up_table[old_state2][state2];
17
18    old_state1 = state1;
19    old_state2 = state2;
20
21 }
```





TEILELISTE			
OBJEKT	ANZAHL	BAUTEILNUMMER	BESCHREIBUNG
1	3	Platte	
2	4	Gewindestange	
3	2	Motorblech	
4	2	Motor	
5	2	Rad	
6	1	Batterie	



Studienarbeit

TITLE: Schaltplan

Document Number:

REV:

Date: 23.06.14 20:01

Sheet: 1/1

# Literaturverzeichnis

- [1] Institut für Regelungs- und Steuerungstheorie TU-Dresden. <http://www.et.tu-dresden.de/rst/>.
- [2] Instituts für Automatisierungs- und Regelungstechnik TU-Wien. <http://www.acin.tuwien.ac.at/>.
- [3] BOLOR: *Balanciervorgang als Video aufgenommen*. <http://bolorkhuu.com/hobby.html>.
- [4] DEVICES, A.: *Accelerometer ADXL335*. <https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>.
- [5] DIGI INTERNATIONAL, I.: *XBee-WLAN-Modul Serie 2*. [http://ftp1.digi.com/support/documentation/90002160\\_A.pdf](http://ftp1.digi.com/support/documentation/90002160_A.pdf).
- [6] ELECTRONICS, P. R. : *30:1 Metal Gearmotor 37Dx52L mm with 64 CPR Encoder*. [http://www.pololu.com/product/1443/](http://www.pololu.com/product/1443).
- [7] ELECTRONICS, P. R. : *VNH3SP30 Motor Driver Carrier MD01B*. <http://www.pololu.com/product/705>.
- [8] GNU: *GCC, the GNU Compiler Collection*. <https://gcc.gnu.org/>.
- [9] HTTPS://GITHUB.COM/TEXANE: *stm32 discovery line linux programmer*. <https://github.com/texane/stlink>.
- [10] JANSCHEK, K.: *Systementwurf mechatronischer Systeme*. Springer-Verlag Heidelberg, 2009.
- [11] JONES, E., T. OLIPHANT und P. E. A. PETERSON: *SciPy: Open source scientific tools for Python*, 2001–2009.
- [12] KNOLL, C. und K. RÖBENACK: *Maneuver Based Control of the 2-DOF Underactuated Manipulator in Normalform Coordinates*. Institute of Control Theory Faculty of Electrical and Computer Engineering TU-Dresden, Dresden, 2013.
- [13] LIECHTI, C.: *PySerial: serial communication tool for Python*. <http://pyserial.sourceforge.net/>.
- [14] LUNZE, J.: *Regelungstechnik 2*. Springer-Verlag, Berlin, 2010.

- [15] RATH, D.: *Open On-Chip Debugger*. <http://openocd.sourceforge.net/>.
- [16] REINSCHKE, K.: *Skript zur Lehrveranstaltung, Steuerung von kontinuierlicher Prozesse*. TU-Dresden, 2002.
- [17] REINSCHKE, K.: *Lineare Regelungs und Steuerungstheorie*. Springer-Verlag, Berlin, 2006.
- [18] STEUERUNGSTHEORIE TU-DRESDEN, I. FÜR REGELUNGS-UND: *Regelungstechnische Praktikumanleitung*. <http://www.et.tu-dresden.de/rst/>, 2013.
- [19] STMICROELECTRONICS: *analog output yaw rate gyroscope*. <http://www.pololu.com/file/0J150/LISY300AL-gyroscope.pdf>.
- [20] STMICROELECTRONICS: *Discovery kit for STM32F407/417 lines - with STM32F407VG MCU*. <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419?sc=internet/evalboard/product/252419.jsp>.
- [21] WIKIPEDIA: *Inverses Pendel*. [http://de.wikipedia.org/wiki/Inverses\\_Pendel](http://de.wikipedia.org/wiki/Inverses_Pendel).
- [22] WWW.MIKROCONTROLLER.NET: *ARM GCC*. [http://www.mikrocontroller.net/articles/ARM\\_GCC](http://www.mikrocontroller.net/articles/ARM_GCC).