Bachelorthesis

Tien Hung Nguyen

Entwicklung und Aufbau eines Fahrsimulators mit Hilfe einer kostengünstigen Mikrocontroller-Plattform
Tien Hung Nguyen

Entwicklung und Aufbau eines Fahrsimulators mit Hilfe einer kostengünstigen Mikrocontroller-Plattform

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung im Studiengang Mechatronik an der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit:
Jugend- und Bildungswerk der Arbeiterwohlfahrt Hamburg GmbH Abteilung Elektronik
Auf dem Königslande 45
22041 Hamburg

Betreuender Prüfer: Prof. Frerk Haase
Zweiter Prüfer: Prof. Dr.-Ing. Jochen Maaß

Abgegeben am 25. November 2016
Tien Hung Nguyen

Thema der Bachelorarbeit
Entwicklung und Aufbau eines Fahrsimulators mit Hilfe einer kostengünstigen Mikrocontroller-Plattform

Stichworte
Fahrsimulator, Getriebemotor, Arduino UNO, Sabertooth, Monster Motor Shield, Parameteridentifikation des Gleichstrommotors, Positionsregelung

Kurzzusammenfassung
Das Ziel der vorliegenden Bachelorarbeit ist es, einen Fahrsimulator nach Kundenanforderungen zu entwickeln und aufzubauen. In dieser Arbeit wird ein Getriebemotor mit unbekannten Parametern identifiziert und ein PID-Regler für die Positionsregelung entworfen. Der Mikrocontroller empfängt die Game-Daten durch eine Motion Simulator Software, um zwei Motoren anzusteuern, die die Bewegungen des Fahrzeugs im Spiel auf den Fahrsimulator übertragen.

Tien Hung Nguyen

Title of the paper
Development and building of a driving simulator with the help of a cost-effective microcontroller-platform

Keywords
Driving simulator, gear motor, Arduino UNO, Sabertooth, Monster Motor shield, parameter identification of a gear motor, position control

Abstract
The objective of the present bachelor thesis is to develop and build a driving simulator according to customer requirements. In this word, a gear motor with unknown parameters is identified and a PID controller for the position control is designed. The microcontroller receives the game data through a Motion Simulator software to drive two motors, which transfer the vehicle’s movements to the driving simulator.
Danksagung

An dieser Stelle möchte ich all jenen danken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.


Ebenso möchte ich mich bei meinen Eltern bedanken, die durch Ihre Unterstützung mein Studium ermöglicht haben.
Inhaltsverzeichnis

Tabellenverzeichnis ........................................................................................................ 7
Abbildungsverzeichnis .................................................................................................... 8

1 Einleitung .................................................................................................................... 10
  1.1 Motivation .............................................................................................................. 10
  1.2 Ziele der Arbeit .................................................................................................... 10
  1.3 Gliederung der Arbeit ......................................................................................... 11
  1.4 Anforderungen an den Fahrsimulator ................................................................. 12

2 Auslegung des Elektroantriebs .......................................................................... 14
  2.1 Analyse ................................................................................................................ 14
  2.2 Berechnung .......................................................................................................... 15
    2.2.1 Statische Berechnung .................................................................................. 15
    2.2.2 Dynamische Berechnung .......................................................................... 20
  2.3 Auswahl des Motors ............................................................................................ 24

3 Grundlagen der GM .................................................................................................. 25
  3.1 Prinzipieller Aufbau ............................................................................................ 26
    3.1.1 Ständerwicklung ......................................................................................... 27
    3.1.2 Ringanker .................................................................................................... 27
    3.1.3 Kommutator ................................................................................................ 28
  3.2 Art der GM ........................................................................................................... 28
  3.3 Mathematisches Modell der GM ......................................................................... 29
    3.3.1 Grundgleichungen .................................................................................... 29

4 Parameteridentifikation der GM ........................................................................... 32
  4.1 Modellierungsprozess ....................................................................................... 33
  4.2 Methode der kleinsten Quadrate ...................................................................... 34
    4.2.1 Motivation und Grundlagen ..................................................................... 34
  4.3 Arduino Uno Boards ............................................................................................. 38
  4.4 Grundsätzliche H-Brückenschaltung ................................................................. 41
    4.4.1 Mit dem Relais ............................................................................................ 42
    4.4.2 Mit dem BJT Transistor .......................................................................... 43
    4.4.3 Mit dem MOSFET ..................................................................................... 45
    4.4.4 Motor Treiber Monster ............................................................................. 49
  4.5 Inkrementalgeber HEDM-5500 ......................................................................... 51
    4.5.1 Funktion ....................................................................................................... 53
    4.5.2 Signalauswertung ....................................................................................... 53
    4.5.3 Art der Programmierung mit Encoder ....................................................... 54
    4.5.4 Ergebnis der Messung ............................................................................... 56

5 Positionsregelung der GM ..................................................................................... 59
  5.1 Grundlagen der Regelungstechnik .................................................................... 59
5.2 Entwurf eines PID-Reglers .................................................................................. 62
5.2.1 Einstellung nach der Ziegler/Nichols – Methode ........................................... 64
5.2.2 PID Einstellung in Matlab/Simulink ............................................................... 68

6 Fahr Simulator ......................................................................................................... 70
   6.1 Beschreibung des Systems ................................................................................. 70
   6.2 Motor Treiber Sabertooth .................................................................................... 70
   6.3 Motion Simulator Software .................................................................................. 71
   6.4 Sequenzdiagramm .............................................................................................. 73
   6.5 Cockpit .............................................................................................................. 73
   6.6 Netzteil 12VDC .................................................................................................. 74

7 Sicherheitsmaßnahmen ............................................................................................. 75
   7.1 Hardware ........................................................................................................... 75
       7.1.1 Notschalter ................................................................................................. 75
       7.1.2 Überhitzungsschutz ..................................................................................... 75
       7.1.3 Motor Schutz ............................................................................................... 76
   7.2 Software ............................................................................................................ 76

8 Zusammenfassung/Ausblick ..................................................................................... 77
   8.1 Zusammenfassung .............................................................................................. 77
   8.2 Ausblick ............................................................................................................. 77
   8.3 Hinweis zum Anhang .......................................................................................... 78

Literaturverzeichnis ....................................................................................................... 79

Versicherung über Selbstständigkeit ......................................................................... 81
# Tabellenverzeichnis

Tabelle 1: Anforderungsliste des Fahrsimulators mit E: Essentiell; N: Notwendig und W: Wünschenswert ................................................................. 12
Tabelle 2: Motoranforderungen .................................................................................................................. 14
Tabelle 3: Getriebeart .................................................................................................................................. 25
Tabelle 4: Vergleich zwischen Arduino und anderen Mikrocontrollersystemen ........ 39
Tabelle 5: verschiedene Versionen von Arduino ......................................................................................... 40
Tabelle 6: Zustände der GM ....................................................................................................................... 48
Tabelle 7: Vergleich zwischen optischen und mechanischen Inkrementalgebers ..... 52
Tabelle 8: Messdatenerfassung der Drehzahlmessung .............................................................................. 57
Tabelle 9: Vergleich zwischen stetigen und unstetigen Regler ............................................................... 60
Tabelle 10: lineare Regler ......................................................................................................................... 61
Tabelle 11: Die Bestimmung von Kp, Tn und Tv nach Ziegler/Nichols ........................................ 64
Tabelle 12: Vergleich zwischen Sabertooth und Monster ................................................................. 71
Tabelle 13: Motion Simulator Software SimTool und X-Sim ............................................. 72
<table>
<thead>
<tr>
<th>Abbildungsverzeichnis</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbildung 1: Blockdiagramm eines Fahrsimulators mit den Bauteilen</td>
<td>11</td>
</tr>
<tr>
<td>Abbildung 2: Einzeichnen der angreifenden Kräfte von Fahrsimulator</td>
<td>15</td>
</tr>
<tr>
<td>Abbildung 3: Die angreifenden Kräfte des Kurbelgetriebes</td>
<td>17</td>
</tr>
<tr>
<td>Abbildung 4: Die angreifenden Kräfte des Sitzplatzes</td>
<td>18</td>
</tr>
<tr>
<td>Abbildung 5: Geschwindigkeitsplan bei einer allgemein ebenen Bewegung</td>
<td>20</td>
</tr>
<tr>
<td>Abbildung 6: Geschwindigkeitsermittlung des Fahrsimulators im Maßstab 1:1</td>
<td>21</td>
</tr>
<tr>
<td>Abbildung 7: Geschwindigkeitsvektor im Punkt C</td>
<td>23</td>
</tr>
<tr>
<td>Abbildung 8: Getriebemotor 12VDC</td>
<td>24</td>
</tr>
<tr>
<td>Abbildung 9: Motordaten von WG88BD88-1</td>
<td>24</td>
</tr>
<tr>
<td>Abbildung 10: Getriebemotor 12VDC WG88DB88-1</td>
<td>26</td>
</tr>
<tr>
<td>Abbildung 11: Der Trommelanker</td>
<td>27</td>
</tr>
<tr>
<td>Abbildung 12: Der Ringanker nach Paccinotti</td>
<td>28</td>
</tr>
<tr>
<td>Abbildung 13: Nebenschlussmaschine</td>
<td>29</td>
</tr>
<tr>
<td>Abbildung 14: Reihenschlussmaschine</td>
<td>29</td>
</tr>
<tr>
<td>Abbildung 15: elektrisches Ersatzschaltbild der GM</td>
<td>30</td>
</tr>
<tr>
<td>Abbildung 16: Blockschaltbild der GM mit Matlab/Simulink</td>
<td>31</td>
</tr>
<tr>
<td>Abbildung 17: Modell der Identifikation der Getriebemotorparameter</td>
<td>32</td>
</tr>
<tr>
<td>Abbildung 18: Konzept der Parameteridentifizierung der GM</td>
<td>33</td>
</tr>
<tr>
<td>Abbildung 19: Vergleich zwischen Messung und Simulation ohne Estimation</td>
<td>35</td>
</tr>
<tr>
<td>Abbildung 20: Beispiel für die Abhängigkeit zwischen Preis und Leistung</td>
<td>36</td>
</tr>
<tr>
<td>Abbildung 21: System Identifikation Toolbox mit Matlab/Simulink</td>
<td>38</td>
</tr>
<tr>
<td>Abbildung 22: einfache-Brückenschaltung mit Schalter</td>
<td>41</td>
</tr>
<tr>
<td>Abbildung 23: Motor im Rechtlauf</td>
<td>41</td>
</tr>
<tr>
<td>Abbildung 24: Motor im Linklauf</td>
<td>42</td>
</tr>
<tr>
<td>Abbildung 25: Motorsteuerungsrelais</td>
<td>42</td>
</tr>
<tr>
<td>Abbildung 26: Motorsteuerung mit npn Transistor</td>
<td>43</td>
</tr>
<tr>
<td>Abbildung 27: Das Beispiel des PWM-Signals</td>
<td>44</td>
</tr>
<tr>
<td>Abbildung 28: Steuerschaltung mit dem BJT-Transistor</td>
<td>44</td>
</tr>
<tr>
<td>Abbildung 29: Schaltzeichen von MOSFET N- und P-Channel</td>
<td>45</td>
</tr>
<tr>
<td>Abbildung 30: Steuerschaltung von N-MOSFET mit Microcontroller (MCU)</td>
<td>46</td>
</tr>
<tr>
<td>Abbildung 31: Steuerschaltung von P-MOSFET mit Microcontroller (MCU)</td>
<td>47</td>
</tr>
<tr>
<td>Abbildung 32: Die komplette H-Brückenschaltung mit den MOSFETs</td>
<td>47</td>
</tr>
<tr>
<td>Abbildung 33: SparkFun Monster Moto Shield</td>
<td>50</td>
</tr>
<tr>
<td>Abbildung 34: Blockdiagramm von Monster Moto Shield</td>
<td>51</td>
</tr>
<tr>
<td>Abbildung 35: Funktion eines Inkrementalgebers</td>
<td>53</td>
</tr>
<tr>
<td>Abbildung 36: Signalauswertung der Drehrichtung im Uhrzeigersinn und Gegenuhrzeigersinn</td>
<td>53</td>
</tr>
<tr>
<td>Abbildung 37: Modell der Drehzahlmessung</td>
<td>55</td>
</tr>
<tr>
<td>Abbildung 38: Algorithmus der Drehzahlmessung</td>
<td>55</td>
</tr>
<tr>
<td>Abbildung 39: Verbindung zwischen Arduino Uno und Monster</td>
<td>56</td>
</tr>
<tr>
<td>Abbildung 40: Vergleich zwischen Messung und Simulation mit Hilfe der „System Identifikation Toolbox“</td>
<td>57</td>
</tr>
</tbody>
</table>
Abbildung 41: Diagramm eines Regelkreises .......................................................... 59
Abbildung 42: Blockdiagramm des Regelkreises .................................................. 60
Abbildung 43: Vergleich der Reglertypen in einem Regelkreis .............................. 62
Abbildung 44: Das Blockdiagramm des geschlossenen Regelkreises ..................... 63
Abbildung 45: PT3-Regelstrecke mit P-Regler ...................................................... 64
Abbildung 46: Das Blockdiagramm des Systems in Matlab/Simulink ..................... 66
Abbildung 47: Die Sprungantwort mit P-Regler (Kpkrit=154) ............................... 66
Abbildung 48: Die Sprungantwort des geschlossenen Regelkreises mit hoher Überschwingungsweite ................................................................. 67
Abbildung 49: Die Sprungantwort nach der Verwendung des Werkzeugs PID-Tuner in Matlab/Simulink ................................................................. 68
Abbildung 50: Die Sprungantwort des geschlossenen Regelkreises mit Kp=26, Ki=23 und Kd=5 ................................................................. 68
Abbildung 51: Systemkontext des Fahrsimulators ................................................. 70
Abbildung 52: Sequenzdiagramm des Fahrsimulators ......................................... 73
Abbildung 53: Schaltplan des Cockpits ................................................................. 74
Abbildung 54: Fahrsimulator mit Notschalter ....................................................... 75
Abbildung 55: Gehäuse für Steuermodule mit Temperatur-Anzeiger ....................... 75
Abbildung 56: Das Bewegungssystem mit 6 DOF ................................................. 78
1 Einleitung

1.1 Motivation


Im nächsten Schritt soll ein beweglicher Playseat, ein Low-Budget-Fahrsimulator entstehen, der die Bewegung eines Fahrzeugs im Spiel in reale Stuhlbewegungen übersetzt. Dies ist die Motivation zu dieser Bachelorarbeit.

1.2 Ziele der Arbeit

Die Arbeit beschäftigt sich mit der Optimierung eines vorhandenen Fahrsimulators, der im Praktikum aufgebaut wurde. Der vorherige Fahrsimulator hat wegen fehlender mechanischer Berechnung nur mit Einschränkung funktioniert, bzw. das limitierte Gewicht des Spielers ist max. 50 kg, die Spielzeit dauert max. 15 Minuten und die Bewegung des Stuhls war verzögert. In dieser Arbeit werden die Anforderungen von dem Kunden (Firma JBW) gesammelt und durch die mechanische Berechnung werden alle Kräfte und das Drehmoment für den Antrieb berechnet. Danach kann man den Antrieb mit der entsprechenden Leistung auswählen.


Zuerst wird die Funktion des Fahrsimulators durch ein Blockdiagramm beschreiben.
1.3 Gliederung der Arbeit

1.4 Anforderungen an den Fahrsimulator

Durch die Besprechung mit Frau Britta Hinz (Geschäftsführerin) und Herrn Jürgen Witt (Projektleiter) wird die Tabelle mit den Anforderungen des Systems dargestellt.

<table>
<thead>
<tr>
<th>ID</th>
<th>Titel</th>
<th>Priorität</th>
<th>Stakeholder</th>
<th>Anforderungen</th>
</tr>
</thead>
<tbody>
<tr>
<td>FST-01</td>
<td>Positionsregelung</td>
<td>E</td>
<td>Nguyen</td>
<td>Zwei Motoren werden durch den Mikrocontroller angesteuert.</td>
</tr>
<tr>
<td>FST-02</td>
<td>mit PID Regler</td>
<td>N</td>
<td>Nguyen</td>
<td>Der Regelkreis soll mit dem bestimmten PID-Regler arbeiten</td>
</tr>
<tr>
<td>FST-03</td>
<td>Cockpit anzeigen</td>
<td>W</td>
<td>Nguyen</td>
<td>Die Gamedaten (Drehzahl, Geschwindigkeit...) sollen am Cockpit angezeigt werden</td>
</tr>
<tr>
<td>FST-04</td>
<td>Belastbarkeit</td>
<td>E</td>
<td>Nguyen</td>
<td>Das System soll mit einem Spieler mit einem Höchstgewicht von 80 kg arbeiten</td>
</tr>
<tr>
<td>FST-05</td>
<td>Spielzeit und Stabilität</td>
<td>N</td>
<td>Nguyen+Witt</td>
<td>Der Fahrsimulator muss mindestens 30 Minuten spielen</td>
</tr>
<tr>
<td>FST-06</td>
<td>Echtzeit</td>
<td>W</td>
<td>Nguyen</td>
<td>Das System soll mit der maximalen Verzögerung von 1 Sekunde funktionieren</td>
</tr>
<tr>
<td>FST-07</td>
<td>Modularität</td>
<td>W</td>
<td>Nguyen+Witt</td>
<td>Das System soll modular aufgebaut sein</td>
</tr>
<tr>
<td>FST-08</td>
<td>Gehäuse</td>
<td>N</td>
<td>Nguyen+Witt</td>
<td>Das Steuerungsmodul muss in einem Gehäuse montiert werden</td>
</tr>
<tr>
<td>ID</td>
<td>Titel</td>
<td>Priorität</td>
<td>Stakeholder</td>
<td>Anforderungen</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FST-09</td>
<td>mehrere Rennspiele</td>
<td>N</td>
<td>Nguyen</td>
<td>Der Fahrsimulator soll mit mindestens zwei Rennspielen (rFactor und Dirt 2) arbeiten</td>
</tr>
<tr>
<td>FST-10</td>
<td>Gesamtkosten</td>
<td>W</td>
<td>Nguyen+Witt</td>
<td>Das System soll preisgünstig produziert werden können (700€)</td>
</tr>
<tr>
<td>FST-11</td>
<td>Austauschbarkeit</td>
<td>N</td>
<td>Nguyen+Witt</td>
<td>Das System soll wartungsfreundlich sein</td>
</tr>
<tr>
<td>FST-12</td>
<td>Sicherheit</td>
<td>E</td>
<td>Nguyen</td>
<td>Bei Überspannung oder Überhitzung soll das System Alarm geben</td>
</tr>
<tr>
<td>FST-13</td>
<td>Bedienbarkeit</td>
<td>N</td>
<td>Nguyen</td>
<td>Der Fahrsimulator soll einfach zu bedienen sein.</td>
</tr>
</tbody>
</table>
2 Auslegung des Elektroantriebs


2.1 Analyse

In dieser Phase werden die Antriebsanforderungen durch die Besprechung mit Frau Hinz (Geschäftsführung) und Herrn Witt (Ausbilder Metalltechnik) der Firma JBW erfasst. In der folgenden Tabelle wird der Vergleich zwischen einem Getriebemotor 12V und einer Asynchronmaschine gezeigt und in Bezug auf die Anforderungen bewertet:

<table>
<thead>
<tr>
<th>ID</th>
<th>Anforderung</th>
<th>Getriebemotor 12V</th>
<th>Asynchronmaschine</th>
</tr>
</thead>
<tbody>
<tr>
<td>M01</td>
<td>niedrige Abtriebs Drehzahl ohne extra Getriebe</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>M02</td>
<td>Hohes Drehmoment am Abtrieb</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>M03</td>
<td>kleiner und kompakter Bauraum und geringes Motorgewicht</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>M04</td>
<td>Einfache Montage</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>M05</td>
<td>Mit niedriger Versorgungsspannung zu betreiben (wegen Sicherheit)</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>M06</td>
<td>Einfache Positionsregelung</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>M07</td>
<td>Steuerung durch Arduino UNO</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>M08</td>
<td>Niedrige Kosten</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

A: Sehr gut, B: Gut; C: Erreichbar, D: Nicht erreichbar
In der oberen Tabelle hat der Getriebemotor 6 Bewertungen mit A und die Asynchronmaschine hat nur eine Bewertung mit A. Deshalb wird der Getriebemotor 12V in diesem Fahrsimulator verwendet.

Die Bestimmung der Kräfte und der Lastmomente wird im nächsten Abschnitt „statische Berechnung“ behandelt.

2.2 Berechnung

2.2.1 Statische Berechnung


In der folgenden Abbildung wird die technische Zeichnung mit den äußeren und inneren Kräften des Fahrsimulators im kartesischen Koordinatensystem angezeigt.
Die äußeren Kräfte:

\[ F_N : \text{Normalkraft} \]
\[ F_R : \text{Radialkraft} \]
\[ F_T : \text{Tangentialkraft} \]
\[ F_P : \text{Pleuelstangenkraft} \]
\[ F_G : \text{Gewichtskraft} \]

Die inneren Kräfte bei dem Festlager C

\[ C_x : \text{horizontale Lagerkraft} \]
\[ C_y : \text{vertikale Lagerkraft} \]

Die Koordinaten des bestimmten Schwerpunkts G im Koordinatensystem Oxyz

\[ G(x_G, y_G, z_G) = G(0,40; 0,40; 0,25) \]

Die Vorberechnung des Winkels \( \theta \)

Im rechteckigen Dreieck OWB:

\[ \theta = \frac{\pi}{2} - \sigma \]  \hspace{1cm} (Gl. 2.1)

Im rechteckigen Dreieck OWE:

\[ \sigma + \alpha = \frac{\pi}{2} - \beta \to \sigma = \frac{\pi}{2} - (\alpha + \beta) \]  \hspace{1cm} (Gl. 2.2)

Ersetzen (Gl. 2.2) in (Gl. 2.1)

\[ \theta = \alpha + \beta \]  \hspace{1cm} (Gl. 2.3)

Um die Kraftberechnung zu vereinfachen, kann man das System in zwei Teile (Kurbelgetriebe und Sitzplatz) aufteilen:
Das Kurbelgetriebe

Die angreifenden Kräfte des Kurbelgetriebes

Abbildung 3: Die angreifenden Kräfte des Kurbelgetriebes

Nun wird der Winkel $\beta$ betrachtet:

\[ \cos \beta = \frac{F}{F_P} \rightarrow F_P = \frac{F}{\cos \beta} \]  
\[ \tan \beta = \frac{F_N}{F} \rightarrow F_N = F \cdot \tan \beta \]  

Die Beziehung zwischen der Tangentialkraft $F_T$ und den Winkeln $\alpha$ und $\beta$

\[ \sin \theta = \sin(\alpha + \beta) = \frac{F_T}{F_P} \rightarrow F_T = F_P \cdot \sin(\alpha + \beta) \]
• **Der Sitzplatz**

Abbildung 4: Die angreifenden Kräfte des Sitzplatzes

Kräftegleichgewicht in x-Richtung
\[ \sum F_x = 0 = -F_p \cdot \sin \beta + A_x \rightarrow A_x = F_p \cdot \sin \beta \quad \text{(Gl. 2.7)} \]

Kräftegleichgewicht in y-Richtung
\[ \sum F_y = 0 = F_p \cdot \cos \beta - F_G + A_y \rightarrow A_y = F_G - F_p \cdot \cos \beta \quad \text{(Gl. 2.8)} \]

Momenten-Gleichgewicht an dem Punkt E
\[ \sum M_E = F_G \cdot l_2 - A_y \cdot l_3 - A_x \cdot l_1 = 0 \quad \text{(Gl. 2.9)} \]

Die Gleichungen 2.7 und 2.8 werden in der Gleichung 2.9 ersetzt und danach kann man die Pleuelstangenkraft bestimmen:
\[ F_P = F_G \cdot \frac{l_3 - l_2}{l_3 \cdot \cos \beta - l_1 \cdot \sin \beta} \quad \text{(Gl. 2.10)} \]

Die Lagerkräfte Cx und Cy
\[ C_x = F_G \cdot \sin \beta \cdot \frac{l_3 - l_2}{l_3 \cdot \cos \beta - l_1 \cdot \sin \beta} \quad \text{(Gl. 2.11)} \]
\[ C_y = F_G - F_G \cdot \cos \beta \cdot \frac{l_3 - l_2}{l_3 \cdot \cos \beta - l_1 \cdot \sin \beta} \quad \text{(Gl. 2.12)} \]

Die Tangentialkraft wird durch das Ersetzen der Gleichung 2.10 in die Gleichung 2.6 berechnet:
\[ F_T = m \cdot g \cdot \frac{l_3 - l_2}{l_3 \cdot \cos \beta - l_1 \cdot \sin \beta} \cdot \sin(\alpha + \beta) \quad \text{(Gl. 2.13)} \]
Jetzt kann man das Lastdrehmoment des Fahrsimulators bestimmen:

\[ M = m \cdot g \cdot r = m \cdot g \cdot \frac{r \cdot (l_3 - l_2)}{l_3 \cdot \cos \beta - l_1 \cdot \sin \beta} \cdot \sin(\alpha + \beta) \]  
(Gl. 2.14)

Mit \( r \): Länge des Hebelarms

Das Ergebnis mit den gemessenen Bemaßungen:

\( l_1 \, [m] = 0,70 \)
\( l_2 \, [m] = 0,40 \)
\( l_3 \, [m] = 0,60 \)
\( r \, [m] = 0,08 \)
\( \alpha \, [\text{rad}] = \frac{5\pi}{18} \)
\( \beta \, [\text{rad}] = \frac{\pi}{18} \)

Das Gewicht des Systems ist die Summe der Gewichte von Spieler und Stuhl

\[ m = m_{\text{Spieler}} + m_{\text{Stuhl}} = 140 \, kg \]

Mit \( m_{\text{Spieler}} = 100 \, [kg] \) und \( m_{\text{Stuhl}} = 40 \, [kg] \)

Die Stützkräfte und die Kräfte der Kurbel

\( C_x \, [N] = 100,88 \)
\( C_y \, [N] = 801,27 \)
\( F_p \, [N] = 580,96 \)
\( F_T \, [N] = 503,13 \)

Das errechnete Drehmoment

\[ M \, [Nm] = 40 \]

Im dem Fahrsimulator verwendet man zwei Motoren, d.h. das minimale Lastdrehmoment für jeden Motor wird halbiert.

\[ M_{\text{min}} = \frac{M}{2} = 20 \, [Nm] \]
2.2.2 Dynamische Berechnung


Mit Hilfe der Euler-Gleichung wird die Geschwindigkeit im Punkt B bestimmt und der vorliegende Geschwindigkeitsvektor \( \vec{V}_B \) resultiert folglich aus der vektoriellen Addition der Geschwindigkeiten der Teilbewegungen.

\[
\vec{V}_B = \vec{V}_A + re_\varphi \quad oder \quad \vec{V}_B = \vec{V}_A + \vec{V}_{BA}
\]

In folgender Abbildung wird der Geschwindigkeitsplan des Fahrsimulators gezeigt.

Abbildung 5: Geschwindigkeitsplan bei einer allgemein ebenen Bewegung\(^2\)

---

\(^1\) Homepage: [https://www.geogebra.org](https://www.geogebra.org)

Abbildung 6: Geschwindigkeitsermittlung des Fahrsimulators im Maßstab 1:1
• Geschwindigkeit im Punkt A

Mit der vorgegebenen Antriebsdrehzahl \( n = 95 \text{ rpm} \) der Kurbel kann man zunächst die Winkelgeschwindigkeit im Punkt A berechnen:

\[
w = 2\pi n = 2\pi \times \frac{2\pi}{60} \times 95 \text{ rpm} = 62,5 \text{ rad/s}
\]

Der Betrag der Geschwindigkeit im Punkt A ist zu ermitteln:

\[
V_A = AA0 \times w = r \times w = 0,08m \times \frac{62,5 \text{ rad}}{s} = 5 \text{ m/s}
\]

Der Geschwindigkeitsvektor \( \overrightarrow{V_A} \) steht senkrecht der Bahnnormalen \( n_A \) und der Geschwindigkeitsplan wird mit dem Maßstab 1:1 konstruiert. Die Länge des dargestellten Vektors \( \overrightarrow{V_A} \) mit dem Betrag 5 m/s ist 10 Zentimeter.

• Geschwindigkeit im Punkt B

Mit Hilfe der Euler-Gleichung wird die Geschwindigkeit im Punkt B ermittelt:

\[
\overrightarrow{V_B} = \overrightarrow{V_A} + \overrightarrow{V_{BA}}
\]

Terme \( \overrightarrow{V_A} \) ist vollständig gegeben (Betrag und Richtung) und bei den anderen Termen \( \overrightarrow{V_B} \) und \( \overrightarrow{V_{BA}} \) sind nur die Richtungen bekannt. \( \overrightarrow{V_B} \) senkrecht der Bahnnormalen \( n_B \) und \( \overrightarrow{V_{BA}} \) senkrecht der Richtung \( \overline{AB} \). Von der Spitze des Vektors \( \overrightarrow{V_A} \) zeichnet man eine Linie senkrecht der Strecke AB. Durch den Schnittpunkt zwischen den Wirkungslinien \( \overrightarrow{V_B} \) und \( \overrightarrow{V_{BA}} \) ist der Geschwindigkeitsvektor \( \overrightarrow{V_B} \) mit dem bekannten Betrag zu ermitteln. Die Länge des Geschwindigkeitsvektors \( \overrightarrow{V_B} \) beträgt 7,13 Zentimeter. Danach wird der Betrag von \( \overrightarrow{V_B} \) durch den Maßstab von \( \overrightarrow{V_A} \) (5 m/s := 10 Zentimeter) berechnet:

\[
V_B = \frac{5 \text{ m/s}}{10 \text{ cm}} \times 7,13\text{ cm} = 3,6 \text{ m/s}
\]
• Geschwindigkeit im Punkt C

![Abbildung 7: Geschwindigkeitsvektor im Punkt C](image)

Die Geschwindigkeit im Punkt C ist mit Hilfe der Euler-Gleichung zu ermitteln:
\[ \overrightarrow{V_C} = \overrightarrow{V_A} + \overrightarrow{V_{CA}} \quad \text{bzw.} \quad \overrightarrow{V_C} = \overrightarrow{V_B} + \overrightarrow{V_{CB}} \]

Durch den Schnittpunkt von zwei Bahnnormalen nA und nB kann man den Momentanpol P, der den Schnittpunkt der Bahnnormalen aller Punkte in einer Lage der bewegten Ebene ist, bestimmen. Die Verbindungslinie zwischen dem Punkt C und dem Momentanpol P entspricht der Bahnnormalen nC, auf der der Geschwindigkeitsvektor \( \overrightarrow{V_C} \) senkrecht steht. An dem Punkt C wird der Geschwindigkeitsvektor \( \overrightarrow{V_A} \) mit der Länge 10 Zentimeter dargestellt, von der Spitze dieses Vektors \( \overrightarrow{V_A} \) wird der Vektor \( \overrightarrow{V_{CA}} \) (senkrecht der Strecke AC) gezeichnet. Durch den Schnittpunkt zwischen den Wirkungslinien \( \overrightarrow{V_C} \) und \( \overrightarrow{V_{CA}} \) ist der bestimmte Geschwindigkeitsvektor \( \overrightarrow{V_C} \) zu ermitteln. Die Länge des Vektors \( \overrightarrow{V_C} \) beträgt 7,55 Zentimeter. Der Betrag des Vektors \( \overrightarrow{V_C} \) wird berechnet:

\[ V_C = \frac{5 m}{10 cm} \times 7,55 \text{ cm} = 3,8 \frac{m}{s} \]

Durch die dynamische Berechnung bestimmt man alle Geschwindigkeitsvektoren \( \overrightarrow{V_A} \), \( \overrightarrow{V_B} \) und \( \overrightarrow{V_C} \) mit den bekannten Richtungen und Beträgen.
2.3 Auswahl des Motors

Aus den Ergebnissen der statischen Berechnung und dem Preisvergleich zwischen den Herstellern wurde der Getriebemotor WG88BD88-1 mit dem maximalen Drehmoment 60 Nm der Firma IMC-Drives ausgewählt.

Abbildung 8: Getriebemotor 12VDC

In folgender Tabelle befindet sich die technischen Daten des ausgewählten Motors:

Abbildung 9: Motordaten von WG88BD88-1
3 Grundlagen der GM


- Die Richtungsumkehr ist problemfrei durch einfaches Umpolen im Betrieb möglich
- Die Ansteuerung ist sehr einfach und hat nur einen geringen regulungstechnischen Aufwand
- Er ist preiswert
- Die 12 VDC Gleichstrom Spannungsversorgung schützt den Spieler vor einer gefährlichen Hochspannung, ein Vorteil im Vergleich zum Wechselstrommotor


Je nach Einsatzbereich lässt sich der Getriebemotor in verschiedene Arten unterteilen. In folgender Tabelle wird der Getriebemotor mit den unterschiedlichen Getrieben aufgezeigt:

Tabelle 3: Getriebeart

<table>
<thead>
<tr>
<th>Getriebe</th>
<th>Schnecken</th>
<th>Stirnrad</th>
<th>Planeten</th>
<th>Kegelrad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td><img src="image1" alt="Schnecken" /></td>
<td><img src="image2" alt="Stirnrad" /></td>
<td><img src="image3" alt="Planetengear" /></td>
<td><img src="image4" alt="Kegelrad" /></td>
</tr>
<tr>
<td>Zum Einsatz</td>
<td>Achse der Abtriebswelle senkrecht zur Motor-Drehachse</td>
<td>Achse der Abtriebswelle parallel zur Motor-Drehachse</td>
<td>Achse der Abtriebswelle gleich der Motor-Drehachse</td>
<td>Achse der Abtriebswelle senkrecht zur Motor-Drehachse</td>
</tr>
</tbody>
</table>
In dem Fahrsimulator wird der Getriebemotor mit dem Schneckengetriebe verwendet

Abbildung 10: Getriebemotor 12VDC WG88DB88-1


3.1 Prinzipieller Aufbau


3.1.1 Ständerwicklung

Am Ständerjoch, das das Gehäuse der Maschine bildet, werden die Polschenkel angebracht. Über dem Polschenkel liegt die Erregerwicklung. Der Gleichstrom fließt in diesen Wicklungen, sodass der Erregerfluss $\Phi$ entsteht, der durch die Polschuhe zum Anker geleitet wird. Der gesamte Ständer (Ständerjoch, Polschenkel und Polschuhe) besteht aus ferromagnetischem Material.

3.1.2 Ringanker

Die Funktion des Gleichstrommotors lässt sich aus dem Läufer nach Pacchinotti erkennen und die Läuferform wird als Ringanker definiert.

Dieser besteht aus einem Eisenring, der aus Einzelblechen zylindrisch aufgeschichtet ist, dadurch verringern sich die Eisenverluste bei der Rotation im Ständerfeld. Der Ringanker trägt die Ankerwicklung. In der folgenden Abbildung hat die Ankerwicklung 12 Windungen.

---

3 Vgl. Prof. Dr. Jens Ginzel: Lehrskript Elektrische Antrieb, S.16
Die Bürste teilt sich vom Anker aus in zwei parallele Zweige auf. Eine Spannung wird an die Bürste angelegt und der Strom wird über die Bürste der Ankerwicklung zugeführt. Mit dem Ständermagnetfeld und der stromdurchflossenen Ankerwicklung wird die Kraft (Lorenz-Kraft) erzeugt. Aufgrund dieser Kraftwirkung in der jeweils entgegengesetzten Richtung am unteren und oberen Ende des Ringankers, entsteht ein Drehmoment, welches wiederum eine Rotationsbewegung des Ringankers verursacht.

Wegen der äußeren Wicklung mit den verschiedenen Winkelstellungen ist das Drehmoment unterschiedlich. Das Drehmoment hängt von der Windungszahl ab. Je größer die Anzahl, desto gleichmäßiger ist das erzeugte Drehmoment.

### 3.1.3 Kommutator


### 3.2 Art der GM

Es gibt verschiedene Arten von Gleichstrommotoren, die sich in der Form der Erregung unterscheiden. Werden die Permanentmagnete zur Erzeugung des Hauptflusses verwendet, so spricht man von einem permanenterregten Gleichstrommotor. Wenn die

---

4 Vgl. Prof. Dr. Jens Ginzel: Lehrskript Elektrische Antrieb, S.17
Erregerwicklung an einer gesonderten Stromversorgung angeschlossen ist, so handelt es sich um einen fremderregten Gleichstrommotor.


![Abbildung 13: Nebenschlussmaschine](image1)
![Abbildung 14: Reihenschlussmaschine](image2)

### 3.3 Mathematisches Modell der GM

Um den Gleichstrommotor zu simulieren, muss das mathematische Modell des Gleichstrommotors erstellt werden. Danach kann mit der Software Matlab/Simulink das Blockschaltbild aufgebaut werden.

#### 3.3.1 Grundgleichungen

Man legt eine Klemmenspannung \( U_a \) an. \( Ra \) und \( La \) bezeichnen den Ankerwiderstand bzw. die Ankerinduktivität. Durch die Drehung des Ankers entsteht eine induzierte Spannung im Anker. Im Ersatzschaltbild wird diese Spannung \( U_q \) bezeichnet.

Kirchhoffsches Gesetz:

\[
U_a = IRa + La \frac{dt}{dt} + Uq \quad \text{(Gl. 3.1)}
\]

Die induzierte Spannung hängt von der Maschinenkonstante \( K \) ab:

\[
U_q = Kw \quad \text{(Gl. 3.2)}
\]

Ersetzt die Gleichung (3.2) in (3.1):

\[
U_a = IRa + La \frac{dt}{dt} + Kw \quad \text{(Gl. 3.3)}
\]

Das elektrische Drehmoment des GM:

\[
Ma = KI \quad \text{(Gl. 3.4)}
\]

Die Bewegungsgleichung des GM:

\[
J \frac{dw}{dt} = Ma - Mr \quad \text{(Gl. 3.5)}
\]
\[
J \frac{dw}{dt} = KI - Bw \quad \text{(Gl. 3.6)}
\]

mit den Parametern

- \( U_a \): Ankerspannung in V
- \( U_q \): induzierte Spannung in V
- \( Ra \): Ankerwiderstand in Ω
- \( La \): Ankerinduktivität in H
- \( Mr = Bw \): viskoses Reibmoment
- \( J \): Trägheitsmoment in kgm²
- \( K \): Maschinenkonstante
In Simulink realisiert man die Differenzengleichung (DGL) am besten, indem man sie nach der höchsten vorkommenden Ableitung auflöst.

Die Gleichung (3.1) wird umgestellt:

\[
\frac{dl}{dt} = \frac{u_a}{L_a} - \frac{R_a}{L_a} I - \frac{K}{L_a} w
\]

(Gl. 3.7)

Mit der DGL (3.6) und (3.7) kann man das Blockschaalbild des GM aufbauen. Mit der Software Matlab/Simulink wird der Gleichstrommotor simuliert.

Abbildung 16: Blockschaltbild der GM mit Matlab/Simulink

Dieses Blockschaltbild wird in dem nächsten Kapitel verwendet, um die Parameter der GM zu identifizieren.
4 Parameteridentifikation der GM

Im vorherigen Kapitel wurden die Grundlagen und die Funktion der GM erklärt und das mathematische Modell wurde aufgebaut. Dieses Kapitel handelt von der Parameteridentifikation.

Um ein System zu simulieren, werden die Modellgleichungen und Modellparameter zuerst bestimmt. Häufig gibt es das Problem, dass die Parameter der Maschinen nicht vorhanden sind oder es gibt kaum Möglichkeiten, die Parameter zu messen. In dieser Arbeit müssen alle Parameter des Getriebemotors bestimmt werden. Die folgende Abbildung zeigt das Experiment der Systemidentifikation für den Gleichstrommotor.

Im Industriebereich nutzt man dafür ein Datenerfassungsgerät, damit die Daten der Messung von der zeitabhängigen Drehzahl mit der bestimmten Eingangsspannung aufgenommen werden können. Aber das Datenerfassungsgerät ist teuer (z.B. das NI USB-6509 vom Hersteller National Instruments kostet 495€) und steht im JBW nicht zur Verfügung.


Die folgende Abbildung zeigt das Verfahren der Parameteridentifikation
4.1 Modellierungsprozess

Bevor die Messung der GM durchgeführt werden kann, muss man die Beziehung zwischen der Ankerspannung $U_a$ und der Winkelgeschwindigkeit $w$ bestimmen. Die Ankerspannung $U(t)$ ist eine zeitabhängige Eingangsfunktion des Messsystems und die Ausgangsfunktion ist die Winkelgeschwindigkeit $w(t)$.

Die Übertragungsfunktion beträgt:

$G(t) = \frac{w(t)}{U(t)}$ \hspace{1cm} (Gl. 3.8)

Um das Übertragungsglied zwischen $U(t)$ und $w(t)$ zu bekommen, muss der Strom $I$ aus der Gleichung (3.6) eliminiert werden:

$I = \frac{J}{K} \frac{dw(t)}{dt} + \frac{B}{K} \cdot w(t)$ \hspace{1cm} (Gl. 3.9)

Die Gleichung (3.8) muss integriert werden:

$\frac{dl}{dt} = \frac{J}{K} \frac{dw(t)}{dt^2} + \frac{B}{K} \cdot w(t)$ \hspace{1cm} (Gl. 3.10)

Ersetzen (3.9) und (3.10) in der Gleichung (3.3):

$U(t) = \frac{I \cdot J}{K} \cdot \frac{dw(t)}{dt^2} + \frac{R \cdot J + L \cdot B}{K} \cdot \frac{dw(t)}{dt} + \left( K + \frac{R \cdot B}{K} \right) \cdot w(t)$ \hspace{1cm} (Gl. 3.11)

Die Koeffizienten werden zusammengefasst

$U(t) = a \cdot \frac{dw(t)}{dt^2} + b \cdot \frac{dw(t)}{dt} + c \cdot w(t)$ \hspace{1cm} (Gl. 3.12)

Mit $a = \frac{I \cdot J}{K}$; $b = \frac{R \cdot J + L \cdot B}{K}$; $c = \left( K + \frac{R \cdot B}{K} \right)$

Die obere Differentialgleichung wird in den Laplace-Bereich transformiert:

$U(s) = W(s) \cdot \left( as^2 + bs + c \right)$ \hspace{1cm} (Gl. 3.13)
Die Übertragungsfunktion ist:

\[ G(s) = \frac{W(s)}{U(s)} = \frac{1}{(as^2 + bs + c)} \]  

(Gl. 3.14)

Damit ist die Übertragungsfunktion eines Gleichstrommotors bestimmt. Im nächsten Schritt wird die mathematische Methode der kleinsten Quadrate aufgestellt.

### 4.2 Methode der kleinsten Quadrate

#### 4.2.1 Motivation und Grundlagen

Die Gleichung (1.11) beschreibt die Beziehung zwischen der Ankerspannung \( U(t) \) und der Winkelgeschwindigkeit \( w(t) \) in Abhängigkeit von der Zeit \( t \). Nach der Laplace Transformation bekommt man die quadratische (polynomische mit Grad 2) Beziehung zwischen der Ankerspannung \( U(t) \), der Winkelgeschwindigkeit \( w(t) \) und der Zeit \( t \). In diesem Kapitel wird das einfache Messsystem aufgebaut, damit die Ausgangswerte (Winkelgeschwindigkeit) mit dem rechteckförmigen Eingangssignal (Ankerspannung) in der Zeit von 3 Sekunden aufgenommen werden. Weil das Eingangssignal konstant (12V) ist, ist nur die Geschwindigkeit in der Zeit von 10 Millisekunden (Abtastzeit) von Interesse. Die Auswahl der Abtastzeit wird im Weiteren erklärt.

Die rote Grafik beschreibt das mathematische Modell der GM und die blaue Grafik zeigt die gemessenen Messdatenpunkte. Man nutzt die Methode der kleinsten Quadrate, um die rote Grafik möglichst nahe an die blaue Grafik zu verschieben.

In der Stochastik wird die Methode der kleinsten Quadrate meistens als Schätzmethode in der Regressionsanalyse genutzt. Mit dieser Methode kann man die Parameter $Ra$, $La$, $J$, $K$ und $B$ der GM ermitteln.

Nachfolgend sollen drei Formen der Regressionsanalyse dargestellt werden und zwar

- die lineare,
- die quadratische (polynomische mit Grad 2) und
- die exponentielle Regression.
Die allgemeine Funktion der Regression im eindimensionalen Fall lautet

\[ y = f(x) + e \]

mit

- \( y \): die abhängige Variable
- \( x \): die unabhängige Variable
- \( f(x) \): die gesuchte Funktion
- \( e \): der Fehler bzw. das Residuum des Modells

**Quadratische Regression**

Aus der Gleichung (3.13) bekommt man die quadratische Funktion:

\[ y = f(x) = ax^2 + bx + c \]

Man ersetzt die Variable \( x \) mit der Variable \( t \) und \( y \) mit \( w \)

Die Messpunkte sind die Wertepaare \((t_1,w_1), (t_2,w_2), (t_3,w_3), \ldots\), so trifft die passende Polynomgleichung \( w = f(x) = at^2 + bt + c \) in der Regel nicht genau auf die Punkte, sondern es gibt bei jedem Wertepaar einen großer Fehler \( e \)

\[
\begin{align*}
    w_1 &= at_1^2 + bt_1 + c + e_1 \\
    w_2 &= at_2^2 + bt_2 + c + e_2 \\
    w_3 &= at_3^2 + bt_3 + c + e_3 \\
    \ldots \, \\
    wn &= at_n^2 + bt_n + c + en
\end{align*}
\]

Nun muss man diesen Fehler $e$ bestimmen.

$$en = wn - (atn^2 + btn + c)$$  \hfill (Gl. 3.15)

Aus der Gleichung (3.5) kann man erkennen, dass der Fehlerwert entweder positiv oder negativ ist. Wenn der Wert negativ ist, können die Endergebnisse der Fehler nicht genau berechnet werden. Dies bedeutet, dass dieser Wert unbedingt positiv sein muss.

Aus der Gleichung (3.15) quadrieren beide Seiten

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (w_i - at_i^2 - b t_i - c)^2$$

d.h. die Straffunktion ist

$$F(a, b, c) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (w_i - at_i^2 - b t_i - c)^2$$

Um das Ziel der Methode der kleinsten Quadrate zu erreichen, muss die Summe der quadrierten Fehler minimiert werden, sodass $e_1^2 + e_2^2 + e_3^2 + \cdots + e_n^2$ möglichst klein wird.

Die Straffunktion ist die polynomische Gleichung und die Voraussetzung der Minimierung ist die erste Ableitung gleich 0.

Diese Funktion hat drei unbekannte Variablen $a, b, c$. Man muss die partielle Ableitung der Straffunktion mit den Variablen $a, b, c$ berechnen.

$$\frac{\partial F}{\partial a} = \frac{\partial F}{\partial b} = \frac{\partial F}{\partial c} = 0$$

$$\frac{\partial}{\partial a} F(a, b, c) = -2 \sum_{i=1}^{n} (w_i - at_i^2 - b t_i - c)(t_i^2) = 0$$

$$\frac{\partial}{\partial b} F(a, b, c) = -2 \sum_{i=1}^{n} (w_i - at_i^2 - b t_i - c)(t_i) = 0$$

$$\frac{\partial}{\partial c} F(a, b, c) = -2 \sum_{i=1}^{n} (w_i - at_i^2 - b t_i - c) = 0$$

Man schreibt alle Gleichungen in ein lineares Gleichungssystem:

$$\begin{pmatrix} \sum_{i=1}^{n} t_i^4 & \sum_{i=1}^{n} t_i^3 & \sum_{i=1}^{n} t_i^2 \\ \sum_{i=1}^{n} t_i^3 & \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i \\ \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} t_i^2 w_i \\ \sum_{i=1}^{n} t_i w_i \\ \sum_{i=1}^{n} w_i \end{pmatrix}$$  \hfill (Gl. 3.16)

Man ersetzt alle Messwertpaare $(t, w)$ in der Gleichung (3.16) und nutzt den Gauß-Algorithmus, dann können die drei Variablen $a, b, c$ errechnet werden.

In der praktischen Phase kann man auf die obige statische Theorie basieren, damit das mathematische Modell des dynamischen Systems mit Hilfe der „System Identifikation Toolbox“ in Matlab/Simulink von gemessenen Eingangs- und Ausgangswerten dargestellt werden kann. In der Arbeit werden die Drehzahlwerte als

---

Auszugswerte mit der bestimmten Eingangsspannung in Abhängigkeit der Zeit gemessen. Diese Daten werden durch die System Identifikation Toolbox analysiert und verarbeitet und anschließend kann durch das mathematische Blockschaltbild das Modell mit den bestimmten Parametern (Ra, La, J, K und B) dargestellt werden.

Im nächsten Schritt wird die Hardware zur Messung der GM aufgebaut und die Experimente werden realisiert. Dann werden die Messdaten aufgenommen. Dies ist die Vorbereitung für die Parameteridentifikation.

### 4.3 Arduino Uno Boards

Tabelle 4: Vergleich zwischen Arduino und anderen Mikrocontrollersystemen

<table>
<thead>
<tr>
<th>Arduino-Boards</th>
<th>andere Mikrocontrollersysteme</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Kompaktes System</td>
<td>• Das System muss aufgebaut werden</td>
</tr>
<tr>
<td>• Einfache Programmierung mit großer Bibliothek und Programmiersprache C und C++</td>
<td>• Komplizierte Programmierung und Programmiersprache mit C, C++ und Assemblersprache.</td>
</tr>
<tr>
<td>• Entwicklungsumgebung ist Open-Source und funktioniert mit verschiedenen Plattformen</td>
<td>• Ungünstiger Preis und mit eingeschränkten Plattformen</td>
</tr>
<tr>
<td>• Funktion nur mit vorhandenem Mikrocontroller, Ersatz nicht möglich</td>
<td>• Unkomplizierter Ersatz vom Mikrocontroller</td>
</tr>
</tbody>
</table>

Heutzutage wird das Arduinoboard in verschiedenen Bereichen verwendet:
- Elektronik
- Robotik
- Steuerung
- Sensordatenerfassung

In dieser Arbeit wird das Arduinoboard zur Steuerung des Fahrsimulators sowie die Sensordatenerfassung für die Parameteridentifikation der GM verwendet. Das Arduinoboard gibt es in drei Versionen mit einer unterschiedlichen Anzahl von digitalen Ein- und Ausgängen und verschiedenen Mikrocontrollern.

---

*[7] Homepage: http://www.arduino.cc*
4.4 **Grundsätzliche H-Brückenschaltung**


Um die Erklärung der Funktion zu erleichtern, werden vier Transistoren mit vier einfachen Schaltern S1, S2, S3 und S4 ersetzt. In folgender Abbildung ist der einfache Schaltplan der H-Brückenschaltung zu sehen.

Vier Schalter werden durch eine Gleichspannung versorgt. Wenn man einen GM in beide Richtungen betreiben will, muss man dafür sorgen, dass die Stromrichtung durch den GM geändert werden kann.

Wenn beide Schalter S1 und S4 eingeschaltet sind, fließt der Strom über S1 zum Motor und über S4 wieder zurück zur Spannungsquelle. Dies bedeutet, dass der Motor im Rechtslauf dreht.
Wenn beide Schalter S3 und S2 eingeschaltet sind, fließt der Strom über S3 zum Motor und über S2 wieder zurück zur Spannungsquelle. Dies bedeutet, dass der Motor im Linkslauf dreht.

Um den Motor per Mikrocontroller anzusteuern, werden die Schalter mit verschiedenen Bauelementen ersetzt:

- mit dem Relais
- mit dem BJT (Bipolar Junction Transistor)
- mit dem MOSFET (Metal Oxide Semiconductor Field-Effect Transistor)

Die Verwendung dieser Arten ist abhängig von der angeforderten Leistung des Systems.

### 4.4.1 Mit dem Relais

In der folgenden Abbildung werden zwei Relais als zwei Schalter verwendet, damit der Motor in beide Drehrichtungen angesteuert werden kann.

Ein Steuersignal wird von dem Mikrocontroller über einen Transistor gesendet. Dieses Signal steuert das Umschalten von beiden Relais an. In der vorherigen Abbildung wird...

Um die Geschwindigkeit mit dem PWM Signal zu regeln, muss man den „mechanischen“ Kontakt (Relais) durch den „elektrischen“ Kontakt (BJT oder MOSFET) ersetzen.

### 4.4.2 Mit dem BJT Transistor

Folgend werden die grundsätzliche Funktion des BJTs (bipolar Junction Transistor) und dann die Funktion der H-Brückenschaltung mit dem BJT beschrieben.

Der bipolare Transistor wird durch einen elektrischen Strom angesteuert. Auf Grund des Aufbaus unterscheidet man zwei Arten von Bipolar Transistoren:

![Bipolar Transistoren](image)

Die Funktionen von beiden Arten sind identisch, im Folgenden wird die Funktion vom npn-Transistor erklärt:

![Motorsteuerung mit npn Transistor](image)
Wenn die Eingangsspannung per Mikrocontroller mit einem rechteckigen Signal (0 oder 5V) versorgt wird, hat das Signal von der Ausgangsspannung auch eine rechteckige Form (12V oder 0V). Zum Beispiel ist die versorgte Eingangsspannung 0V, dann fließt der Strom über den Widerstand RB nicht und der Strom über den Widerstand RC ist sehr klein. Die Ausgangsspannung Vo ist 12V.

Ist aber die Eingangsspannung 5V und der Strom über den Widerstand RC erreicht den maximalen Wert und die Kollektor-Emitter-Spannung UCE ist gleich 0, dann bedeutet dies, dass die Ausgangsspannung Vo 0V ist. Deshalb ist das Signal von der Ausgangsspannung auch ein rechteckförmiges Signal bzw. PWM-Signal in Abhängigkeit von der Zeit. Wie aus der Gleichung (1.13) ersichtlich, ist die Geschwindigkeit der GM abhängig von der Spannung. Die Geschwindigkeit wird durch diese Spannung angesteuert. Die folgende Abbildung zeigt ein Beispiel des PWM-Signals:

\[ V_m = V_{cc} \cdot \frac{t_{ein}}{t_{ein} + t_{aus}} = p \cdot V_{cc} \]

Oder Motorspannung = Betriebsspannung * Tastverhältnis

Mit p: Tastverhältnis (engl. Duty Cycle)


Die folgende Abbildung zeigt die Schaltung mit dem BJT-Transistor

\[ V_m = V_{cc} \cdot \frac{t_{ein}}{t_{ein} + t_{aus}} = p \cdot V_{cc} \]

Oder Motorspannung = Betriebsspannung * Tastverhältnis

Mit p: Tastverhältnis (engl. Duty Cycle)
Das Umschalten von den Eingängen T1, T2, T3 und T4 wird mit dem Mikrocontroller geregelt. Man verwendet 4 Dioden (Schottky-Diode), damit die Transistoren und der Mikrocontroller vor der induzierten Spannung geschützt werden. Für die obere Schaltung werden zwei pnp-Transistoren benutzt und für die untere Schaltung zwei nnp-Transistoren.

Die H-Brückenschaltung mit dem BJT-Transistor hat folgende Vorteile:
- Hohe Strombelastbarkeit
- Sehr geringer Spannungsabfall
- Geringe Kosten

Nachteil: Die Transistoren müssen mit teuren Schutzdioden betrieben werden. Deshalb verwendet man die H-Brückenschaltung mit einem MOSFET. Bei modernen MOFETs kann auf Schutzdioden verzichtet werden.

4.4.3 Mit dem MOSFET

Mit dem Vorteil der hohen Strombelastbarkeit (d.h. hohes Drehmoment) wird die H-Brückenschaltung mit einem MOSFET in dem Fahrsimulator verwendet.

Die Grundlagen des MOSFETs sind ähnlich wie bei einem BJT-Transistor. Im Unterschied zu den bipolaren Transistoren sind die Anschlüsse mit Gate (G), Drain (D) und Source (S) bezeichnet. Das Gate entspricht der Basis eines Transistors, Drain dem Kollektor und Source dem Emitter. Wie bei den BJT-Transistoren gibt es bei den MOSFETs N- und P-Typen.

![Abbildung 29: Schaltzeichen von MOSFET N- und P-Channel](image)

Wenn eine Gleichspannung bei dem G(Gate)-Anschluss angelegt wird, ist der MOSFET gesperrt oder geleitet. Die Umschaltung des MOSFETs ist abhängig von der Gate-Spannung. Bei einer Gate-Spannung (U_GS) von 0V sperrt der N-MOSFET, weil der Widerstand zwischen Gate- und Source-Anschluss sehr groß ist. Im Gegenteil leitet der P-MOSFET bei dieser Gate-Spannung. Wenn die Betriebsspannung (allgemein von 5V) bei dem Anschluss Gate angelegt wird, verhält es sich genau umgekehrt. Der N-MOSFET leitet und der P-MOSFET sperrt.
Heutzutage wird der MOSFET in der H-Brückenschaltung verwendet, weil die Strombelastbarkeit des MOSFETs höher ist als die des BJT-Transistors. Eine H-Brückenschaltung besteht aus vier MOSFET (zweimal N-MOSFET und zweimal P-MOSFET). Die Anordnung der MOSFET wird in dem nächsten Schritt erklärt.

Nun wird die Schaltung zwischen N-MOSFETs und dem Motor betrachtet. Die folgende Abbildung zeigt diese Schaltung.

Abbildung 30: Steuerschaltung von N-MOSFET mit Mikrocontroller (MCU)

Bei der Gate-Spannung von 0V sperrt der N-MOSFET und der Widerstand zwischen D und S ist somit sehr groß. Die Spannung bei dem Source-Anschluss ist gleich 0V und der Motor dreht sich nicht.

Wenn man die Gleichspannung an den Gate-Anschluss legt, ist der Widerstand zwischen D und S sehr klein und die Spannung bei dem Source-Anschluss ist gleich der Betriebsspannung (12V). Der MOSFET wird geleitet und der Motor dreht sich. Um den MOSFET einzuschalten, ist die Spannung bei dem Gate-Anschluss größer als die Spannung bei Source (d.h. UG>12V, weil die Spannungsversorgung für den Motor 12V beträgt). Die Gate-Anschluss wird mit einem Pin des Mikrocontrollers verbunden und durch das Steuerprogramm kann man den logischen Ausgangswert (High oder Low, bzw. 1 und 0) von diesem Pin ansteuern. Bei dem Wert 1 hat die Gate-Spannung (UG) den gleichen Wert wie die Ausgangsspannung des Pins des Mikrocontrollers. Weil aber die Ausgangssspannung des Mikrocontrollers auf einen maximalen Wert von 5V begrenzt ist, ist eine direkte Verbindung zwischen dem Mikrocontroller und dem Gate-Anschluss nicht möglich. Um dieses Problem zu lösen, wird der N-MOSFET durch einen P-MOSFET ersetzt.
Wenn die Ausgangsspannung (5V) des Mikrocontrollers mit dem Gate-Anschluss verbunden ist, ist die Gate-Spannung (UG) kleiner als die Source-Spannung (US=12V). Dies bedeutet, dass der MOSFET geleitet hat d. h. der Motor wird mit der Spannung 12 V versorgt und er dreht sich um.

Auf diesem Grund muss man zwei P-MOSFET für obere H-Brückenschaltung verwenden und zwei N-MOSFET für die untere Schaltung. In der folgenden Abbildung wird die komplette H-Brückenschaltung gezeigt.

Um das Ein- und Ausschalten der H-Brückenschaltung anzusteuern, werden die vier Eingänge IN1, IN2, IN3 und IN4 mit den Ausgängen vom Mikrocontroller (MCU) verbunden. Dies bedeutet, dass man vier Ausgänge des Mikrocontrollers braucht. Um die Anzahl von den Ausgängen zu reduzieren, kann man die Logik-Bausteine (AND- und NOT-Verknüpfung) verwenden. Außerdem kann die H-Brückenschaltung mit den Logik-Bausteinen vor einem Kurzschluss geschützt werden. Wenn beide MOSFET


Die folgende Tabelle zeigt die Zustände der GM:

<table>
<thead>
<tr>
<th>PWM</th>
<th>DIR</th>
<th>A</th>
<th>B</th>
<th>IN1</th>
<th>IN2</th>
<th>IN3</th>
<th>IN4</th>
<th>MOTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>stoppend</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>stoppend</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>rechtsdrehend</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>linksdrehend</td>
</tr>
</tbody>
</table>

Von der logischen Tabelle wird die logische Ansteuerung durch den Mikrocontroller (Arduino UNO) durchgeführt. Hier ist ein Beispiel-Code für die Ansteuerung der H-Brückenschaltung mit dem Motor Treiber „Monster“ der Firma SparkFun:

```c
void motorGo(uint8_t motor, uint8_t direct, uint8_t pwm)
{
    if (motor <= 1)
    {
        if (direct <= 4)
        {
            // Set inA[motor]
            if (direct <= 1)
                digitalWrite(inApin, HIGH);
            else
                digitalWrite(inApin, LOW);
```
// Set inB[motor]
if ((direct==0)||(direct==2))
    digitalWrite(inBpin, HIGH);
else
    digitalWrite(inBpin, LOW);

analogWrite(pwmpin, pwm);
}
}
}

4.4.4 Motor Treiber Monster

Die technische Daten des Motortreibers Monster:

Die maximale Betriebsspannung: 16V
Der maximale Motorstrom: 30A
Der typische Motorstrom: 14A
Der angesteuerte Motor: 2
Der Widerstand des MOSFETS: 19 mOhm
Maximale PWM-Frequenz: 20 kHz

Automatisches Abschalten bei Unter- und Überspannung sowie bei Überhitzung und messbarem Motorstrom

Abbildung 33: SparkFun Monster Moto Shield®

8 Homepage: [http://www.sparkfun.com](http://www.sparkfun.com)
Im vorherigen Text ging es um die Funktion des Mikrocontrollers und des Motortreibers. Um die Parameter der GM zu identifizieren, muss eine Drehzählmessung der GM durchgeführt werden. Danach kann man mit Matlab-Simulink die Datenerfassung dieser Messung zur Parameteridentifikation verwenden. Im Fahrsimulator wird dafür ein Inkrementalgeber vom Hersteller Avago Technologies benutzt.

### 4.5 Inkrementalgeber HEDM-5500

Ein Inkrementalgeber (auch Drehimpulsgeber und Drehgeber genannt) dient der Erfassung von Winkeländerung bei Achsen und Wellen. Ein Inkrementalgeber liefert eine bestimmte Anzahl an Impulsen pro Umdrehung, deshalb wird er häufig als Drehzahlmesser oder zur Positionsbestimmung in der Automatisierungstechnik verwendet. Inkrementalgeber können mit Schleifkontakt, optisch oder magnetisch arbeiten. Im gesamten System verwendet man einen Inkrementalgeber mit Optik für die Parameteridentifikation und einen Inkrementalgeber mit Schleifkontakt (Potenziometer genannt) für den Fahrsimulator.

In folgender Tabelle werden die Vorteile von optischen und „mechanischen“ Inkrementalgebern dargestellt.

---


Abruf 2016-11-15
Tabelle 7: Vergleich zwischen optischen und mechanischen Inkrementalgebern

<table>
<thead>
<tr>
<th>Eigenschaften</th>
<th>Optischer Inkrementalgeber</th>
<th>Potenziometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bauform</td>
<td>möglicherweise</td>
<td>möglicherweise</td>
</tr>
<tr>
<td>Drehrichtung und Drehzahl bestimmen</td>
<td>digital</td>
<td>analog</td>
</tr>
<tr>
<td>begrenzter Drehwinkel</td>
<td>ohne</td>
<td>360°</td>
</tr>
<tr>
<td>Genauigkeit</td>
<td>sehr genau (10 bit)</td>
<td>niedrig</td>
</tr>
<tr>
<td>mit Arduino programmieren</td>
<td>sehr kompliziert</td>
<td>sehr einfach</td>
</tr>
<tr>
<td>Störung</td>
<td>Ja, mit fremden Licht</td>
<td>wenig</td>
</tr>
<tr>
<td>Preis</td>
<td>Nicht günstig (ca. 40€)</td>
<td>Günstig (5€)</td>
</tr>
</tbody>
</table>

Wegen der hohen angefordernten Genauigkeit bei der Drehzahlmessung muss man einen optischen Inkrementalgeber einsetzen. Um den Kostenaufwand des Fahrsimulators gering zu halten, kann man für den Betrieb des Fahrsimulators ein Potenziometer mit geringer Genauigkeit verwenden.

Nun betrachtet man die Grundlagen und die Funktion eines optischen Inkrementalgebers.
4.5.1 Funktion


4.5.2 Signalauswertung

Wegen der Anordnung der zwei Fototransistoren erhält man zwei um 90° elektrisch phasenverschobene Signale (A und B) bei der Drehung der GM.

In einer Drehrichtung ist das Signal A gegenüber dem Signal B um 90° voreilend und die umgekehrte Richtung ist um 90° nacheilend. In der oberen Abbildung sind die Signale A und B dargestellt mit ihren phasenverschobenen Ausgangssignalen. Die Drehrichtung wird unterschieden durch die Voreilung und Nacheilung der Signale.

Vgl. Prof. Dr. Rasmus Rettig: Lehrskript Sensorik, S. 48

4.5.3 Art der Programmierung mit Encoder

Aus der Positionsregelung oder Drehzahlregelung und der Auflösung des Inkrementalgebers gibt es mehrere Varianten der Zählung des Impulses mit dem Mikrocontroller (Arduino UNO) über die Timer.

- **Timer mit Funktion Input Capture**
  Der Arduino Uno mit dem Mikrocontroller ATmega328 hat zwei 8-Bit Timer (Timer 0 und Timer 2) und einen 16-Bit (Timer 1). Ein Timer ist einfach ein bestimmtes Register im Mikrocontroller. Der Timer wird auf- oder abwärtsgezählt, wenn ein Ereignis geschehen ist. Das Signal A oder B wird mit dem Input Capture Pin (ICP, Arduino Uno – Pin 8) verbunden. Bei einem Flankenwechsel von Signal A oder B an einen ICP Pin speichert der Timer seinen aktuellen Wert im ICR (Input Capture Register). Durch die Differenz aus zwei aufeinanderfolgenden Werten im ICR kann man die Frequenz vom Signal A oder B bestimmen, um die Drehzahl zu berechnen. Außerdem kann man einen zusätzlichen Timer zur Drehrichtungsbestimmung verwenden.

  Diese Methode misst die Zeit mit hoher Genauigkeit. Ein Nachteil ist aber die komplizierte Programmierung des Timers.

- **Counter (Zähler)**
  Der Arduino Uno hat auch drei Zähler und zwar zwei 8-Bit Zähler (Zähler 0 und 2) und einen 16-Bit Zähler (Zähler 1). Zwei Ausgangssignale A und B werden mit zwei Zählern vom Arduino Uno verbunden. Dadurch kann man die Impulse von beiden Signalen A und B in der bestimmten Zeit zählen und die Geschwindigkeit der GM wird dann berechnet. Diese Methode dient nur für die Bestimmung der Drehzahl, die Drehrichtung kann nicht bestimmt werden.

- **Externate Interrupt**
  Diese Methode kombiniert die zwei Methoden mit Timer und Zähler. Sowohl die Drehzahl als auch die Drehrichtung der GM kann damit bestimmt werden. Außerdem ist die Programmierung vom Arduino damit vereinfacht und gleichzeitig die Messgenauigkeit gestiegen.

  Bei bestimmten Ereignissen in Prozessoren wird ein sogenanntes Interrupt ausgelöst. Deshalb werden zwei Ausgangssignale A und B von dem Inkrementalgeber an zwei Interrupt-Pins vom Arduino angeschlossen. Der Arduino Uno hat zwei Interrupt-Pins und zwar Pin 2 (INT0) und Pin 3 (INT1). Die Interrupts sind bei einer fallenden (high nach low) oder steigenden Flanke (low nach high) aktiviert. Danach wird eine Funktion, die die Impulse von dem Signal A oder B in der bestimmten Zeit zählt, aufgerufen. Der
Zähler wird inkrementiert oder dekrementiert, wenn der Motor sich im Uhrzeigersinn oder Gegenuhrzeigersinn dreht. In dieser Methode muss man die Taktfrequenz vom Arduino beachten. Die maximale Drehzahl der GM beträgt 100 RPM (oder 10 rad/s) im Leerlauf und die Impulse pro Umdrehung des Inkrementalgeber betragen 1000 CPR. Die maximale Frequenz des Signals A oder B wird berechnet:

\[ f = 100 \text{ rad/s} \times 1000 \frac{\text{Impulse}}{\text{rad}} = 10000 \text{ Hz} = 10 \text{ kHz} \]

Die Taktfrequenz des Arduino Uno muss unbedingt größer sein als der Wert \( f=10\text{kHz} \). Laut Datenblatt des Arduino beträgt seine Taktfrequenz 16MHz und das bedeutet, dass man die Methode mit Encoder (1000 CPR) zur Drehzahlmessung verwenden kann.

**Abbildung 37: Modell der Drehzahlmessung**

In dem folgenden Aktivitätsdiagramm wird ein Algorithmus der Drehzahlmessung gezeigt:

**Abbildung 38: Algorithmus der Drehzahlmessung**
Parameteridentifikation der GM

In dieser Arbeit wird ein Inkrementalgeber mit 1000 CPR (Impulse pro Umdrehung, engl. Count per Revolution) genutzt und es werden insgesamt 4 Zustände der Flanke von beiden Signalen A und B genutzt, um die Impulse zu zählen. Deshalb wird der Drehwinkel mit folgender Formel berechnet:

\[ Drehwinkel = \frac{2\pi \cdot \text{Anzahl der Impulse}}{1000 \text{ CPR} \cdot 4} \]

**4.5.4 Ergebnis der Messung**


Damit die Genauigkeit der Drehzahlmessung erhöht wird, wurde das Experiment fünfmal durchgeführt. Danach wurden in Matlab/Simulink die Ergebnisse der Drehzahlmessung zur Parameteridentifikation der GM genutzt.

Abbildung 39: Verbindung zwischen Arduino Uno und Monster

Die folgende Abbildung zeigt den Vergleich zwischen dem Simulations-Modell und der Messung in Echtzeit:

Abbildung 40: Vergleich zwischen Messung und Simulation mit Hilfe der „System Identifikation Toolbox“

Tabelle 8: Messdatenerfassung der Drehzahlmessung

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ankerwiderstand (RA)</td>
<td>Ω</td>
<td>0,22253</td>
<td>0,21855</td>
<td>0,21740</td>
<td>0,22205</td>
<td>0,21846</td>
<td>0,21980</td>
</tr>
<tr>
<td>Ankerinduktivität (LA)</td>
<td>H</td>
<td>0,00261</td>
<td>0,00276</td>
<td>0,00278</td>
<td>0,00236</td>
<td>0,00261</td>
<td>0,00262</td>
</tr>
<tr>
<td>Maschinenkonstante (K)</td>
<td></td>
<td>0,01025</td>
<td>0,01053</td>
<td>0,01015</td>
<td>0,01026</td>
<td>0,01012</td>
<td>0,01026</td>
</tr>
<tr>
<td>Trägheitsmoment (J)</td>
<td></td>
<td>0,01925</td>
<td>0,01958</td>
<td>0,01882</td>
<td>0,01943</td>
<td>0,01921</td>
<td>0,01926</td>
</tr>
<tr>
<td>Viskose Reibung (B)</td>
<td>kg·m²</td>
<td>0,08011</td>
<td>0,08410</td>
<td>0,08131</td>
<td>0,08001</td>
<td>0,08028</td>
<td>0,08116</td>
</tr>
</tbody>
</table>
Die berechneten Parameter des benutzten Getriebemotors:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Wert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ankerwiderstand (RA)</td>
<td>0,219798</td>
</tr>
<tr>
<td>Ankerinduktivität (LA)</td>
<td>0,002622</td>
</tr>
<tr>
<td>Maschinenkonstante (K)</td>
<td>0,010262</td>
</tr>
<tr>
<td>Trägheitsmoment (J)</td>
<td>0,019259</td>
</tr>
<tr>
<td>Viskose Reibung (B)</td>
<td>0,081164</td>
</tr>
</tbody>
</table>

Die bestimmten Parameter werden im nächsten Kapitel verwendet, um die Parameter vom PID-Regler zu identifizieren.
5 Positionsregelung der GM

5.1 Grundlagen der Regelungstechnik


Prinzip der Regelung

- Messen

Die Regelgröße (aktuelle Position) wird mittels Potenziometer gemessen.

- Vergleichen

Der Wert der Regelgröße wird mit dem Sollwert (eingestellte Position aus dem Game) durch den Arduino UNO verglichen. Die Differenz wird Regelabweichung genannt.

- Stellen

Nach der Berechnung vom Arduino UNO wird die Stellgröße (gewünschte Position) für zwei Motoren bestimmt.

\[\text{Abbildung 41: Diagramm eines Regelkreises}^{11}\]

\[11\text{ Vgl. Unger Jochem: Einführung in die Regelungstechnik, S.15}\]
In der nächsten Abbildung wird das Blockschaltbild eines einfachen Standardregelkreises gezeigt. Es besteht aus zwei Hauptteilen Regler und Regelstrecke.

Abbildung 42: Blockdiagramm des Regelkreises

Die Aufgabe des Reglers ist die Minimierung der Differenz zwischen Soll- und Istwert der Regelgröße. Der Regler stellt einen Ausgleich her. Allgemein wird der Regler nach stetigem und unstetigem Verhalten unterschieden:

<table>
<thead>
<tr>
<th>Stetige Regler</th>
<th>Unstetige Regler</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Generiert aus der Regeldifferenz einen kontinuierlichen Stellwert</td>
<td>- Generiert aus der Regeldifferenz einen diskreten Stellwert.</td>
</tr>
<tr>
<td>Beispiel: P-, I-, PD-, PI-, PID-Regler</td>
<td>Beispiel: Zwei-, Drei-, Mehrpunktregler</td>
</tr>
</tbody>
</table>

Im Fahrsimulator wird die aktuelle Position der GM durch ein analoges Potenziometer ermittelt. Der Mikrocontroller Arduino Uno nutzt diese Werte, um durch einen Regler-Algorithmus die eingestellte Position festzulegen. Daher wird ein stetiger Regler verwendet. In folgender Tabelle handelt es sich um die verschiedenen Arten des stetigen Reglers mit dem Eingangswert e(t) und dem Ausgangswert u(t):
### Tabelle 10: lineare Regler

<table>
<thead>
<tr>
<th>Typ</th>
<th>Verhalten</th>
<th>Formel</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>proportional</td>
<td>$u(t) = K_p \cdot e(t)$</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>integral</td>
<td>$u(t) = K_I \int_0^t e(\tau)d\tau$</td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>proportional und integral</td>
<td>$u(t) = K_p \cdot e(t) + K_I \int_0^t e(\tau)d\tau$</td>
<td></td>
</tr>
<tr>
<td>PD</td>
<td>proportional und differential</td>
<td>$u(t) = K_p \cdot e(t) + K_D \frac{de(t)}{dt}$</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>proportional, integral und differential</td>
<td>$u(t) = K_p \cdot e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt}$</td>
<td></td>
</tr>
</tbody>
</table>


5.2 Entwurf eines PID-Reglers

Um einen gut funktionierenden Regelkreis aufzubauen, muss das dynamische Verhalten der Regelstrecke ermittelt werden. Es heißt das Verhalten des Getriebemotors. Aus der Gleichung (3.11) im Kapitel 3 kann man die Übertragungsfunktion der GM im Laplace-Bereich bestimmen.

\[
G(s) = \frac{W(s)}{U_a(s)} = \frac{K}{(Js+B)(LaB^2+Ra)+K^2}
\]  \hspace{1cm} (Gl 5.1)

Wegen der Positionsregelung interessiert man sich für das Verhältnis des Drehwinkels θ und der Ankerspannung Ua. Deshalb muss die obige Übertragungsfunktion weitergerechnet werden.

\[\text{Abb. 43: Vergleich der Reglertypen in einem Regelkreis}^{12}\]

\[\text{\hspace{1cm} Fr\"{u}hling 2016}^{12}\]

\[\text{\hspace{1cm} Abruf: 2016-11-15}\]
Es gibt die Beziehung zwischen dem Drehwinkel $\theta$ und der Winkelgeschwindigkeit $W$:

$$W = \frac{d\theta}{dt} \text{ oder } \int dW * dt = \theta$$

Die Gleichung (5.1) wird integriert und nochmal im Laplace-Bereich transformatiert. Nun wird die Übertragungsfunktion zwischen dem Drehwinkel $\theta$ und der Ankerspannung $U_a$ ermittelt.

$$G(s) = \frac{\theta(s)}{U_a(s)} = \frac{K}{(Js+B)(L_A+R_A)+K^2} + \frac{1}{s} \left[ \frac{\text{rad}}{V} \right]$$

Alle Werte von den Parametern der GM werden in die Übertragungsfunktion eingesetzt:

$$G(s) = \frac{\theta(s)}{U_a(s)} = \frac{0,01026}{5,05 \times 10^{-5} * s^3 + 4,45 \times 10^{-3} * s^2 + 0,0179 * s}$$

Durch die obige Übertragungsfunktion kann man einfach erkennen, dass die Regelstrecke ein PT3 – Glied (Verzögerungsglied mit 3. Ordnung) besitzt. Jetzt wird ein Blockdiagramm der Positionsregelung der GM erstellt:


5.2.1 Einstellung nach der Ziegler/Nichols – Methode

Bei der Methode nach Ziegler/Nichols werden die Reglerparameter verstellt, damit das System die Stabilitätsgrenze erreichen kann und der Regelkreis fängt an zu schwingen. Damit wird die Periode der Schwingung ermittelt und mit der „Ziegler/Nichols Tabelle“ kann man alle Parameter bestimmen.

Die Vorgehensweise von dieser Methode ist folgende:

1. Der Regler wird zuerst als P-Regler betrieben, d.h. $ki = kd = 0$ und Anfangswert von P-Regler ist gleich ($Kp = 0$)
2. $Kp$ wird langsam erhöht, bis der geschlossene Regelkreis die Stabilitätsgrenze erreicht und eine Dauerschwingung auftritt.
3. Die Verstärkung $Kp$ wird zurückgenommen und bezeichnet als $Kp_{krit}$
4. Die Periodendauer $T_{krit}$ der Schwingung wird gemessen
5. Die Reglereinstellwerte können dann aus „Ziegler/Nichols Tabelle“ entnommen werden:

Tabelle 11: Die Bestimmung von $Kp$, $T_n$ und $Tv$ nach Ziegler/Nichols

<table>
<thead>
<tr>
<th>Regler Typ</th>
<th>$Kp$</th>
<th>$T_n$</th>
<th>$Tv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$0,5 \times Kp_{krit}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PI</td>
<td>$0,45 \times Kp_{krit}$</td>
<td>$0,83 \times T_{krit}$</td>
<td>-</td>
</tr>
<tr>
<td>PID</td>
<td>$0,6 \times Kp_{krit}$</td>
<td>$0,5 \times T_{krit}$</td>
<td>$0,125 \times T_{krit}$</td>
</tr>
</tbody>
</table>

Mit der Nachstellzeit $T_N = \frac{Kp}{Ki}$ und der Vorhaltezeit $T_V = \frac{Kd}{Kp}$

Nun betrachtet man den Regelkreis mit dem Führungsverhalten. Das Führungsverhalten beschreibt die Auswirkung von Führungsgröße $w(t)$ auf die Regelgröße $x(t)$.

Abbildung 45: PT3-Regelstrecke mit P-Regler
Für diesen geschlossenen Regelkreis wird die Führungsübertragungsfunktion ermittelt:

\[ F_w = \frac{F_R F_S}{1 + (F_R F_S)} \]

Die Stabilität des geschlossenen Regelkreises ist abhängig von der genauen Lage der Pole der Übertragungsfunktion. Wenn alle Pole links der imaginären Achse liegen, ist das System stabil. Aus diesem Grund betrachtet man nur die Nullstellen des Nenners, d.h. \( 1 + (F_R \cdot F_S) = 0 \). Der P-Regler beträgt eine Verstärkung \( K_p \) und diese Verstärkung muss bestimmt werden. Nun werden die Verstärkung \( K_p \) und die Übertragungsfunktion der Strecke (Gleichung ...) in der oberen Gleichung eingesetzt.

\[ 1 + \left( \frac{K_p K}{(Js + B)(La s + Ra + K^2) \cdot s} \right) = 0 \]

Und das Nennerpolynom wird bestimmt:

\[ L_A s^3 + (L_A B + R_A J) s^2 + (R_A B + K^2) s + KK_p = 0 \]

Die Bestimmung der Verstärkung \( K_p \) oder die Untersuchung des Regelkreises beruht auf dem Hurwitz – Kriterium. Das reelle Polynom für den 3. Grad nach dem Hurwitz – Kriterium beträgt:

\[ a(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \]

Dann wird die Hurwitzdeterminante gebildet:

\[
H = \begin{vmatrix}
    a_2 & a_0 & 0 & 0 \\
    a_3 & a_4 & 0 & a_0 \\
    0 & a_2 & a_0 & 0 \\
\end{vmatrix}
= \begin{vmatrix}
    (L_A B + R_A J) & K K_p & 0 \\
    L_A J & (R_A B + K^2) & 0 \\
    0 & (L_A B + R_A J) & K K_p \\
\end{vmatrix}
\]

Der geschlossene Regelkreis ist stabil, wenn alle Unterdeterminanten größer als Null sind, d.h.

\[ H_1 = |a_{22}| = |(L_A B + R_A J)| \geq 0 \]

\[ H_2 = \frac{|a_{22}|}{a_3 a_1} = (L_A B + R_A J) * (R_A B + K^2) - (L_A J * KK_p) \geq 0 \]

\[ H_3 = a_3 * H_2 = KK_p * [(L_A B + R_A J) * (R_A B + K^2) - (L_A J * KK_p)] \geq 0 \]

Die Motorparameter betragen:

\[ R_A = 0,219788 \text{ Ohm} \]

\[ L_A = 0,002622 \text{ H} \]

\[ K = 0,010262 \]
\[ J = 0,019259 \text{ kgm}^2 \]
\[ B = 0,081164 \]

Und die Berechnung aller Unter determinanten:

\[ H_1 = 0,0044 > 0 \]
\[ H_2 = 7,9782 \times 10^{-5} - 5,182 \times 10^{-7} \cdot K_p \geq 0 \rightarrow K_p \leq 153,9589 \]
\[ H_3 = 0,010262 \cdot K_p \cdot [7,9782 \times 10^{-5} - 5,182 \times 10^{-7} \cdot K_p] \geq 0 \rightarrow K_p \geq 0 \]

Durch die Bedingungen von \( H_2 \) und \( H_3 \) wird die Verstärkung \( K_p \) ermittelt:

\[ 0 \leq K_p \leq 154 \]

Wenn die Verstärkung \( K_p \) den Maximalwert 154 erreicht, liegt das System an der Stabilitätsgrenze. Es bedeutet, dass die Verstärkung \( K_p \) mit dem Wert 154 genommen wird. Der geschlossene Regelkreis führt die Dauerschwingung aus. Nun wird das System mit Matlab/Simulink simuliert.

Nach dem Simulieren bekommt man die Grafik des Systems:

Die Periodendauer wird gemessen: \( T_{krit} = 0,334 \text{ s} \)
Durch die „Ziegler/Nichols Tabelle“ werden die Vorhaltezeit und Nachstellzeit berechnet:

\[ K_P = 0,6 \times K_{P_{krit}} = 0,6 \times 154 = 92 \]
\[ T_N = 0,5 \times T_{Krit} = 0,5 \times 0,334 \text{s} = 0,1673 \text{s} \]
\[ T_V = 0,125 \times T_{Krit} = 0,125 \times 0,334 \text{s} = 0,0418 \text{s} \]

Nun kann man alle PID – Reglerparameter bestimmen:

\[ K_I = \frac{K_P}{T_N} = \frac{92}{0,1673} = 552 \]
\[ K_D = K_P \times T_V = 92 \times 0,0418 = 4 \]

Alle PID-Reglerparameter werden eingesetzt und das System wird nochmal simuliert.

Abbildung 48: Die Sprungantwort des geschlossenen Regelkreises mit hoher Überschwingungsweite

Durch die obere Grafik der Sprungantwort von dem geschlossenen Regelkreis kann man erkennen, dass der Endwert den stationären Wert 1 erreicht. Aber die Überschwingungsweite (engl. Overshoot), die die größte vorübergehende Sollwertabweichung nach einem Sollwertsprung (z.B. 1 rad) ist, hat einen hohen Wert, d.h. 65,2 %. Im Industriebereich versucht man, dass dieser Wert auf weniger als 10 Prozent reduziert wird. Um dieses Ziel zu erreichen, wird das automatische Tunen von PID-Reglern mit Matlab/Simulink durchgeführt.
5.2.2 PID Einstellung in Matlab/Simulink

Durch die Verwendung eines automatischen Werkzeugs PID-Tuner in Matlab/Simulink kann man die optimalen Werte von drei Regelgliedern (Proportionalglied, Integralglied und Differenzialglied) des PID-Reglers ermitteln. In folgender Abbildung wird das Werkzeug PID – Tuner angezeigt.

Abbildung 49: Die Sprungantwort nach der Verwendung des Werkzeugs PID-Tuner in Matlab/Simulink


Abbildung 50: Die Sprungantwort des geschlossenen Regelkreises mit Kp=26, Ki=23 und Kd=5

Mit dem eingestellten PID-Regler (Kp=26, Ki=23 und Kd=5) besitzt der geschlossene Regelkreis die richtigen Eigenschaften:

- Überschwingungsweite (engl. Overshoot) = 6,67 %
- Endwert (engl. Final value) = 1 (d.h. Ohne Abweichung zwischen Sollwert und Istwert)
- Anstiegszeit (engl. rise time) = 0,13 s (d.h. nach 0,13 Sekunden erreicht das System 90 % des Sollwertes).

In folgendem Kapitel nutzt der Mikrocontroller Arduino Uno diesen PID-Regler mit den Werten Kp=26, Ki=23 und Kd=5, um durch einen Regler-Algorithmus die eingestellte Position festzulegen.
6 Fahrsimulator

6.1 Beschreibung des Systems


![Systemkontext des Fahrsimulators](image)

Abbildung 51: Systemkontext des Fahrsimulators


6.2 Motor Treiber Sabertooth

Um die Anforderung FST 04 (Spieler mit einem Gewicht von 80 kg) erreichen zu können, werden zwei Getriebemotoren mit großer Leistung verwendet. Deshalb braucht der Motor großen Motorstrom und einen Motor-Treiber mit großer Stromversorgung. In dieser Arbeit wird ein Motor-Treiber Sabertooth 2x60 vom Hersteller Dimension Engineering genutzt. Im Vergleich zu dem im Kapitel 4.4.4 vorgestellten Motor-Treiber Monster vom Hersteller SparkFun hat dieser Motor-Treiber Sabertooth mehrere Vorteile:
### Tabelle 12: Vergleich zwischen Sabertooth und Monster

<table>
<thead>
<tr>
<th>Sabertooth 2x60</th>
<th>SparkFun Monster</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Sabertooth" /></td>
<td><img src="image" alt="SparkFun Monster" /></td>
</tr>
<tr>
<td>• Kontinuierlich fließender Strom: 60 A pro Channel.</td>
<td>• Kontinuierlich fließender Strom: 14A pro Channel</td>
</tr>
<tr>
<td>• Versorgungsspannung: 6-30V</td>
<td>• Versorgungsspannung: 6-16V</td>
</tr>
<tr>
<td>• Überstrom- und Überhitzungsschutz</td>
<td>• Überhitzungsschutz</td>
</tr>
<tr>
<td>• Mehrere Eingangsmodule bearbeiten, z.B. Analog, simplified serial und R/C</td>
<td>• Nur Analogsignal bearbeiten</td>
</tr>
<tr>
<td>• Hoher Preis: 160€</td>
<td>• Günstiger Preis: 70€</td>
</tr>
</tbody>
</table>

Durch die vorherige Tabelle kann man erkennen, dass der Motortreiber von Sabertooth den Motor mit 60A versorgt, und der von Hersteller Monster nur mit 14A. Deshalb wird man den Motortreiber von Sabertooth 2x60 verwenden, um zwei Getriebemotoren WG88BD88-1 anzusteuern.

### 6.3 Motion Simulator Software

Um die Bewegung des Fahrsimulators durchzuführen, kann man zwei unterschiedliche Motion Simulator Programme von zwei Entwicklern X-Sim\(^\text{14}\) und SimTool\(^\text{15}\) nutzen. Jede Software hat Vor- und Nachteile und ist abhängig vom Anwender. In diesem Projekt wurde SimTools für die Ansteuerung der Motoren (Bewegung) und für die Ansteuerung des Cockpits benutzt.

---

\(^{14}\) Homepage [http://www.x-sim.de](http://www.x-sim.de)

\(^{15}\) Homepage [http://www.xsimulator.net](http://www.xsimulator.net)
### Tabelle 13: Motion Simulator Software SimTool und X-Sim

<table>
<thead>
<tr>
<th>SimTools</th>
<th>X-Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>![SimTool screenshot]</td>
<td>![X-Sim screenshot]</td>
</tr>
<tr>
<td>• kostenfreie Software                                                   • Zwei Versionen: Kostenfreie Version (mit Internet vorhanden) und offline Version (60 €)</td>
<td></td>
</tr>
<tr>
<td>• unterstützt mehrere Rennspiele (ca. 45 Rennspiele)                      • Unterstützt ca. 50 Rennspiele</td>
<td></td>
</tr>
<tr>
<td>• Parameter von Gamedaten nicht ablesbar                                 • Alle Parameter auszulesen und Diagramm erstellbar</td>
<td></td>
</tr>
<tr>
<td>• mit wenigen Hardware-Steuerungen (Arduino, Sabertooth, Monster) kompatibel • Mit viel verschiedener Hardware (Arduino, Sabertooth, Monster, Hydraulische System...) kompatibel</td>
<td></td>
</tr>
<tr>
<td>• maximal 6 parallele Schnittstellen gleichzeitig zu nutzen               • Max. 2 Schnittstellen</td>
<td></td>
</tr>
</tbody>
</table>

Im nächsten Abschnitt geht es um die Kommunikation zwischen den einzelnen Komponenten des Fahrsimulators.
6.4 Sequenzdiagramm

Mit dem Sequenzdiagramm wird beschrieben, welche Komponenten beteiligt sind, und in welcher zeitlichen Reihenfolge die Informationsaustausche stattfinden. In folgender Abbildung wird das Sequenzdiagramm des Fahrsimulators gezeigt:

![Sequenzdiagramm des Fahrsimulators](image)


6.5 Cockpit

Um die Drehzahl (RPM) und die Geschwindigkeit anzuzeigen, wird ein analoger Tachometer E36 vom Hersteller BMW verwendet, der mit einem Arduino-Mega 2560 angesteuert wird. Die folgende Abbildung zeigt das Blockdiagramm des Cockpits:
Durch das Plugin Game Dash 16 der Simulation Software SimTool empfängt der Arduino Mega die Gamedaten und steuert damit den analogen Tachometer an.

6.6 Netzteil 12VDC


Daten eines Netzteils:
- Max. Strom: 48A
- Spannung: 12VDC
- Max. Leistung: 650 W

16 Game Dash 1.1, URL: https://www.xsimulator.net/community/threads/game-dash.4978 . Abruf: 2016-11-15
7 Sicherheitsmaßnahmen

7.1 Hardware

7.1.1 Notschalter

Um das Gesamtsystem zu schützen, verwendet man einen Notschalter. Wenn das Fahrsimulatorssystem oder der Fahrer/Spieler ein Problem hat, kann man das System schnell ausschalten.

Abbildung 54: Fahrsimulator mit Notschalter

7.1.2 Überhitzungsschutz


Abbildung 55: Gehäuse für Steuermodule mit Temperatur-Anzeiger
7.1.3 Motor Schutz

Während der Funktion des Getriebemotors birgt das Anfassen des Motors eine Verletzungsgefahr für den Spieler. Um dieses zu vermeiden, wird der Getriebemotor in ein Gehäuse eingebaut.

7.2 Software


Die Funktion „ST.motor()“ arbeitet mit zwei Parametern:

**ST. motor (Motor, Power)**

Motor: welcher Motor (1 oder 2) wird angesteuert

Power: $-127 \leq W e r t \leq 127$. Wenn der Wert gleich 0 ist, bleibt der Motor stehen. Und wenn der Wert gleich 127 oder -127 ist, dreht der Getriebemotor sich in die Vorwärts- und Rückwärtsrichtung mit der maximalen Geschwindigkeit. Werte zwischen 0 und + oder -127 ergeben langsamere Bewegungen.
8 Zusammenfassung/Ausblick

Im letzten Kapitel wird eine zusammengefasste Darstellung der Arbeit beschrieben. Dies beinhaltet, wie der Fahrsimulator nach den Anforderungen des Kunden aufgebaut wurde. Im nachfolgenden Unterpunkt “Ausblick” werden Ansatzpunkte beschrieben, die im Anschluss an diese Bachelorarbeit weiterbearbeitet werden können.

8.1 Zusammenfassung

In der Arbeit wurde ein Fahrsimulatorssystem auf Grundlage der vorhandenen Anforderungen entwickelt. Es wurden die Parameter des Getriebemotors mittels der mathematischen Methode der kleinsten Quadrate identifiziert, ohne dass ein besonderes Datenerfassungsgerät für die Drehzahlmessung zur Verfügung stand. Durch die Simulation in Matlab/Simulink konnten unkompliziert alle Parameter des PID-Reglers bestimmt werden. Es können zwei unterschiedliche Motion Simulation Programme genutzt werden, um damit die Gamedaten zu dem Mikrocontroller zu senden. Das System ist offen für alle Rennspiele, dessen Daten von den Motion Simulation Programmen ausgelesen werden können. Das System des Fahrsimulators bewegt sich realistisch, ohne Verzögerungen und parallel zu den Bewegungen des Fahrzeugs im Spiel und der Spieler fühlt sich, als würde er in einem realen Auto sitzen.

Ohne die Arbeitszeit zu berechnen, wurden für die in dieser Arbeit verwendeten Bauteile knapp 300,00 Euro investiert. Mit den Kosten für die vorhandene Sitzkonstruktion wurde das geforderte Budget von 700,- Euro eingehalten. Auf Lötarbeiten wurde weitestgehend verzichtet, um die Wartung zu vereinfachen. Für die Vereinfachung der Bedienung für den Spieler wurde der Startvorgang des Fahrsimulators automatisiert. Durch den modularen Aufbau kann der Fahrsimulator einfach demontiert und transportiert werden.


8.2 Ausblick

8.3 Hinweis zum Anhang

Der Anhang zur Arbeit befindet sich auf CD und ist bei den Prüfern Professor Haase und Professor Maaß einzusehen. Darin enthalten sind die Datenerfassungen der Drehzahlmessung, das Matlab-Skript und der Programmcode, der nur in digitaler Form zur Verfügung steht.

\[17\] Bewegungssystem https://www.xsimulator.net/co-ordinate-system Abruf: 2016-11-15
Literaturverzeichnis


[14] FÖLLINGER, OTTO; DÖRRSCHEIDT, ULRICH KONIGORSKI (Bearb.); BORIS LOHMANN (Bearb.): *Regelungstechnik: Einführung in die Methode und ihre


Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den 25 November 2016
Ort, Datum  
Unterschrift