

# Simulationsstudie:

## Regelung eines Laserablenkspiegels

Projektarbeit zur Vorlesung Simulationstechniken  
SS 2013

Fakultät 06  
Fakultät für angewandte Wissenschaften und Mechatronik  
der Hochschule München

*vorgelegt von*

**Michael Jost**  
**Sebastian Schleich**

München, Juli 2013

Simulationsstudie eingereicht am: .....

Prüfer:

Prof. Dr. Rainer Froriep  
Prof. Dr. rer. nat. Alfred Kersch



# Inhaltsverzeichnis

<b>1. Einführung</b>	<b>1</b>
<b>2. Aufgabenstellung</b>	<b>5</b>
<b>3. Mathematische Modellbildung</b>	<b>7</b>
3.1. Das System . . . . .	7
3.2. Der Spiegel . . . . .	7
3.3. Der Motor . . . . .	9
3.4. Der Sensor . . . . .	9
3.4.1. Das physikalische Modell des Sensors . . . . .	9
3.4.2. Das lineare Sensormodell . . . . .	10
3.4.3. Das nicht lineare Sensormodell . . . . .	10
<b>4. Programmentwicklung</b>	<b>13</b>
4.1. Motor in Simulink . . . . .	13
4.2. Matlab . . . . .	14
4.2.1. Motor in Matlab . . . . .	14
4.2.2. Sensor in Matlab . . . . .	15
4.2.2.1. Vorbereitungen . . . . .	15
4.2.2.2. Files . . . . .	15
<b>5. Simulationsdurchführung</b>	<b>17</b>
5.1. Simulation . . . . .	17
5.1.1. chap:pidregelung . . . . .	17
5.1.1.1. chap:p <sub>r</sub> egelung . . . . .	17
5.1.1.2. chap:pi <sub>r</sub> egelung . . . . .	17
5.1.1.3. chap:pd <sub>r</sub> egelung . . . . .	18
5.1.1.4. chap:pid <sub>r</sub> egelung . . . . .	18
5.1.2. chap:padaption . . . . .	21
5.1.2.1. chap:p <sub>a</sub> daption . . . . .	21
5.1.2.2. chap:p <sub>a</sub> daptiongalvo . . . . .	22
5.1.2.3. chap:p <sub>a</sub> daptionwerte . . . . .	22
5.1.2.4. chap:p <sub>a</sub> daptionstrom . . . . .	23
5.1.2.5. chap:p <sub>a</sub> daptionfertig . . . . .	25

5.1.3.	chap:sensorregelung . . . . .	26
5.1.3.1.	chap:sensorregelung1 . . . . .	27
5.1.3.2.	chap:sensorregelung2 . . . . .	27
5.1.3.3.	chap:sensorregelung3 . . . . .	29
<b>6.</b>	<b>Diskussion</b>	<b>31</b>
<b>7.</b>	<b>Zusammenfassung</b>	<b>33</b>
<b>A.</b>	<b>Matlab-Files</b>	<b>35</b>
A.1.	msSpiegelundSensor.m . . . . .	35
A.2.	sensorDaten.m . . . . .	39
A.3.	frect.m . . . . .	41
A.4.	gekern.m . . . . .	41
A.5.	sensor.m . . . . .	42
A.6.	posEingabe.m . . . . .	45
<b>B.</b>	<b>Simulink-Files</b>	<b>47</b>
B.1.	1 . . . . .	47
	<b>Literaturverzeichnis</b>	<b>49</b>
	<b>Abbildungsverzeichnis</b>	<b>51</b>
	<b>Tabellenverzeichnis</b>	<b>53</b>
	<b>Eidesstattliche Erklärung</b>	<b>55</b>

# 1. Einführung

Bei der Bearbeitung von z.B. Wafern (z.B. Lasertrimmen von Widerständen), Glas (z.B. Brille) oder Masken für die Lithographie, wird ein Laserstrahl in der Fokusebene durch 2 Spiegel, einer für die X- und einer für die Y-Richtung, abgelenkt. Durch diese Anordnung lassen sich Beschriftungen in 2 Dimensionen ausführen. Wobei eine Fokusebene durch Werte gleicher Intensität und gleicher Strahldurchmesser festgelegt ist. Aber auch in der Medizin, beim Laserstrahlschweißen, in der Raumfahrt, beim Militär und in Barcode Scannern finden Spiegelmotoren Verwendung. Die Spiegel sind auf einer Welle montiert, die über einen Motor bewegt wird. Es gibt verschiedene Größen der Spiegel und der entsprechenden Motoren. Aber nicht nur die Größe der Spiegel und der Motoren unterscheiden sich, sondern auch in der erreichbaren bzw. geforderten Geschwindigkeit und Genauigkeit unterscheiden sich die verschiedenen Ausführungen.

Wird ein Laserstrahl durch zwei Spiegel abgelenkt, so ist die Fokusebene auf eine kleine Fläche beschränkt, siehe Abb. 1.1 und 1.2. Sollen größere Flächen mit dem Laser bearbeitet werden, so müssen die Anlagen verändert werden.

Es werden drei Lösungsansätze vorgestellt.

Der Laserstrahl ist fest nach unten gerichtet und zwei entsprechend große Motoren bewegen den ganzen Laser in X- und Y-Richtung über das Werkstück. Diese Möglichkeit bringt allerdings einige Nachteile mit sich. Zum einen ist die Geschwindigkeit und Genauigkeit von großen Motoren eingeschränkt. Zum anderen kann es die Lebensdauer eines Lasers negativ beeinflussen, wenn er ständig Beschleunigt und Abgebremst wird, und so entsprechenden Kräften die an ihm rütteln ausgesetzt ist. Diese Lösung ist für sehr feine bzw. kleine Strukturen und aus wirtschaftlichen Gründen nicht zu empfehlen.

Eine andere Lösung ist der Einsatz von Robotern in der Fertigung. Dies bringt allerdings den Nachteil, dass die Bewegung drei-dimensional ausgeführt wird und so weitere Ungenauigkeiten miteingebracht werden.

Der dritte Lösungsansatz sieht einen fest eingebauten Laser vor, dessen Laserstrahl nur in einer Richtung abgelenkt wird. Durch die Auslenkung des Laserstrahls in nur einer Richtung, ergibt sich keine Fokusebene sondern eine Fokuslinie. Bewegt sich nun das zu beschriftende Werkstück quer zu der Fokuslinie, kann eine große Fokusebene beschriftet werden, siehe Abb. 1.

In Abb. 1 tritt die Fokuslinie senkrecht aus dem Bild heraus. Es muss dabei sichergestellt sein, dass der Vorschub des Werkstücks auf die Einstellgeschwindigkeit der Spiegelverstellung und somit auf die Beschriftung in der Fokuslinie abgestimmt ist. Durch eine entsprechende Anordnung können so Werkstücke größeren Ausmaßes beschriftet werden.



Abb. 1.1.: 2 Laserablenkspiegel [?]

Die Bearbeitung von Solarpanels oder Werbebeschriftungen auf Folien können so realisiert werden.

Gegenstand dieser Simulationsstudie ist die Regelung eines Motors zur Ablenkung eines Laserstrahls an einem Spiegel.

Es sollen verschiedene Winkel eingestellt werden, wobei die größte Winkeländerung  $20^\circ$  betragen soll. Die größte Winkeländerung soll mit einer Einstellzeit von 1 ms und mit einer Regelgenauigkeit von  $1e-3^\circ$  erreicht werden.

Der Spiegel ist auf der Welle eines Gleichstrommotors befestigt. Da Gleichstrommotoren auf eine Drehgeschwindigkeit und nicht auf einen festen Winkel geregelt werden, ergibt sich hier eine neue Aufgabenstellung. Die Regelung umfasst folgende Bereiche:

- Elektronische Steuerung des Motors
- Mechanische Umsetzung der elektrischen Steuersignale in Drehbewegungen
- Aufnehmen der aktuellen Winkelposition durch einen Sensor und Umwandlung in ein elektrisches Signal

Das Aufnehmen der Winkelposition ist von großer Wichtigkeit. In dieser Simulationsstudie lässt sich die Winkelposition in Simulink direkt ablesen. Wird aber ein reales Bauteil angeschlossen, weichen dessen Parameter von denen der Simulationsstudie ab. Aufgrund dieser Abweichung lässt sich die Winkelposition nicht mehr direkt ablesen, sondern muss

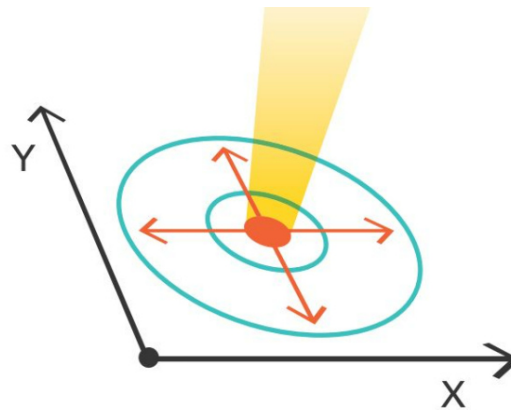


Abb. 1.2.: Fokusebene [?]

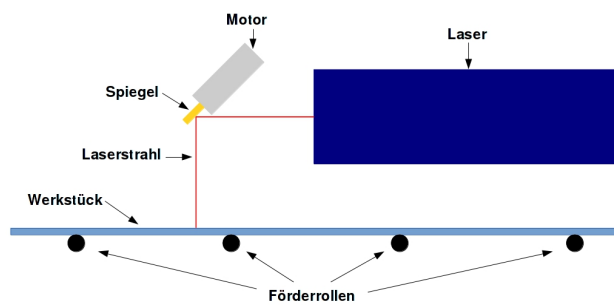


Abb. 1.3.: Fokusline

am Bauteil selbst gemessen werden. Ohne eine genaue Winkelzuordnung ist eine Regelung aber nicht möglich. Aus diesen Gründen wird in Kap. 3.4 intensiver auf die Sensoren eingegangen und es werden verschiedene Sensormodelle vorgestellt.





## 2. Aufgabenstellung

Um größere Flächen eines Werkstücks mit dem Laser zu bearbeiten, soll ein Laserstrahl von einem fest eingebauten Laser mit einem Spiegel abgelenkt werden. Es entsteht so eine Fokuslinie in der das Werkstück beschriftet werden kann. Durch einen Vorschub des Werkstückes kann so eine große Fläche beschriftet werden.

Die Ablenkung des Laserstrahls erfolgt durch einen Gleichstrommotor, auf dessen Welle ein Spiegel fest montiert ist.

In dieser Simulationsstudie soll untersucht werden, ob es möglich ist eine Regelung aufzubauen, die einen Laserablenkspiegel, der von einem Gleichstrommotor bewegt wird, auf eine bestimmte Winkelposition zu bewegen und in entsprechenden Regeldifferenzen zu halten. Es werden folgende, willkürlich gewählten Leistungsmerkmale vorgegeben:

- Verstellung des Spiegels aus der Ruhelage (Mitte) um  $\pm 10^\circ$ . Wobei die Ruhelage des Spiegels den Laserstrahl genau in die Mitte der Fokuslinie auf dem Werkstück ablenkt.
- Um einen maximalen Winkelbereich von  $20^\circ$  abzufahren, darf die Regelung nicht länger als 1 ms benötigen.
- Der einzustellende Winkel soll mit einer Genauigkeit von  $1e-3^\circ$  erreicht und gehalten werden.

In dieser Simulationsstudie wird vorausgesetzt, dass der Abstand des Lasers zum Werkstück keine Rolle spielt. Zudem wird der Fokus des Laserstrahls über den zu regelnden Winkelbereich als konstant angenommen. Der aufeinander abgestimmte Vorschub des Werkstücks und abfahren der Fokuslinie des Lasers wird hier nicht betrachtet, da nur die Ablenkung des Laserstrahls im Zentrum der Studie steht. Ein in der Realität beobachtbarer an- und abstieg der Laserleistung beim an- und abschalten des Lasersstrahls wird hier vernachlässigt.

Die Simulationsstudie deckt folgende Themen ab:

- Bewegung von Magnet, Welle und Spiegel als mechanische Arbeit durch angesetzte Drehmomente
- Drehmomente werden durch Ströme, die Magnetfelder hervorrufen, realisiert
- Positionserfassung durch Auswertung von Lichtintensitäten auf 4 Sensoren
- es müssen verschiedene Parameter wie, Trägheitsmomente von Spiegel und Welle, Drehmomente, induzierte Spannungen und z.B. Lichtintensitäten beachtet werden

Bevor mit der Simulationsstudie begonnen wird, werden einige Vereinfachungen angenommen:

- Spiegel und Drehachse sind eine immer gleich konzentrierte Masse -> gleiche Beschleunigungen
- Luftspalt zwischen Magnet und Spule hat keinen Einfluss -> Luftspalt hat geringere magnetische Kraftflussdichte
- Spiegel ist immer mit Schwerpunkt in der Drehachse -> keine anderen Drehmomente, kein Verbiegen
- Durch verdrehen des Spiegels kann der Laserstrahl nicht vom Spiegel "fallen"(wäre der Spiegel zu weit gedreht, so dass der Laserstrahl nur noch auf eine kleine Ablenkfläche trifft, würde der mittlere Teil des Laserstrahls abgelenkt und der äußere Teil würde am Spiegel vorbei "laufen")
- Lichtquelle hat konstante Beleuchtungsstärke in den Halbraum
- Völlige Abdunkelung des einen Sensors, wenn der andere maximale Helligkeit besitzt
- Alle Bauteile 100% steif
- Erwärmung und dadurch eine Veränderung der Parameter wird nicht beachtet

Es wird mit einem vorgegebenen Gleichstrommotor begonnen, Werte für die Regelparameter heraus zu finden, mit denen sich erste Ergebnisse zeigen. Mit diesen gefundenen Regelparametern wird dann versucht, die Regelergebnisse noch zu verbessern. Als Alternative kommt die s.g. P-Adaption in Betracht. Bei der P-Adaption gibt es im Regelkreis nur einen P-Regler. Diesem P-Regler ist eine Funktion vorgeschaltet, die es über zwei einzugebende Parameter erlaubt, näher an den Sollwert zu gelangen.

## 3. Mathematische Modellbildung

### 3.1. Das System

Der vorliegenden Simulationsstudie wird in folgendes, in Abb. 3.1 dargestellten Systems zugrunde gelegt.

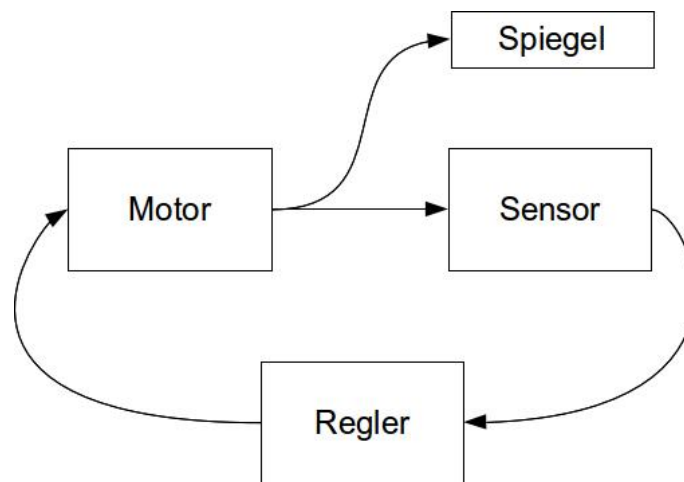


Abb. 3.1.: Allgemeiner Aufbau des simulierten Systems

### 3.2. Der Spiegel

Lineares Modell für die Berechnungen:

$$\Delta\phi = 20^\circ = 0,349rad$$

$$\Delta t = 1ms = 0,001s$$

$$\omega = \frac{\Delta\phi}{\Delta t} = \frac{0,349rad}{0,001s} = 349rad/s$$

(3.1)

Es ergibt sich eine Durchschnittswinkelgeschwindigkeit von 349 rad/s, um einen Winkel von 20° in 1 ms zu überfahren. Dies würde aber eine Anfangs- und Endgeschwindigkeit

voraussetzen. Da der Spiegel aber aus einer Ruhelage beschleunigt und wieder in einer Ruhelage enden soll, wird ein linearer Verlauf der Geschwindigkeit von  $\omega = 0 \text{ rad/s}$  und der doppelten Durchschnittsgeschwindigkeit  $\omega = 698 \text{ rad/s}$  bei der Hälfte der Strecke und bei der Endposition wieder  $\omega = 0 \text{ rad/s}$  der zu fahrenden Strecke angenommen. Daraus folgt eine Beschleunigung von:

$$\begin{aligned}\Delta\omega &= 698 \text{ rad/s} \\ \Delta t &= 0,5 \text{ ms} = 0,0005 \text{ s} \\ \alpha &= \frac{\Delta\omega}{\Delta t} = \frac{698 \text{ rad/s}}{0,0005 \text{ s}} = 1,396 * 10^6 \text{ rad/s}^2\end{aligned}\quad (3.2)$$

Der Spiegel erfährt zu Beginn der Regelung eine Beschleunigung von  $\alpha = 1,396 * 10^6 \text{ rad/s}^2$  um nach der Hälfte der Zeit, also nach 0,5 ms wieder mit dem gleichen Betrag der Beschleunigung abgebremst zu werden.

Modell für den Spiegel:

- Durchmesser: 12 mm  $\rightarrow$  Radius:  $R = 6 \text{ mm}$
- Höhe:  $h = 2 \text{ mm}$
- Gewicht:  $m = 10 \text{ g}$

Das Trägheitsmoment des Spiegels beträgt demnach:

$$\begin{aligned}J &= \frac{1}{4} * m * R^2 + \frac{1}{12} * m * h^2 \\ J &= 1 \frac{4 * 10 * 10^{-3} * (6 * 10^{-3})^2 + \frac{1}{12} * 10 * 10^{-3} * (2 * 10^{-3})^2}{J = 93,3 * 10^{-9} \text{ kgm}^2} \quad (3.3)\end{aligned}$$

Aus den oben berechneten Daten ergibt sich ein Lastmoment von:

$$\begin{aligned}M_L &= J * \alpha \\ M_L &= 93,3 * 10^{-9} \text{ kgm}^2 * 1,396 * 10^6 \text{ rad/s}^2 \\ M_L &= 130,25 * 10^{-3} \text{ Nm} \quad (3.4)\end{aligned}$$

Theoretische Maximale Leistung eines Gleichstrommotors:

$$P = M_L * \omega$$

$$P = 130,25 * 10^{-3} Nm * 698 rad/s$$

$$P = 91 W \quad (3.5)$$

### 3.3. Der Motor

In der Regel werden Laserablenkespiegel über einen Galvo gesteuert. Bei der Bearbeitung dieser Simulationsstudie ergaben sich Probleme, Informationen über die Ansteuerung solcher Galvos zu bekommen. Insofern wird die Simulationsstudie auf der Ansteuerung eines Gleichstrommotors beruhen. Aber auch hierbei konnten jedoch keine Informationen über die Gleichstrommotorparameter KPHI und der Reibungskonstanten bei verschiedenen Herstellern gefunden werden. Um dennoch die Studie durchführen zu können, wird auf die Motorvorgaben aus der Vorlesung Systemtechnik von Prof. Froriep zurück gegriffen.

### 3.4. Der Sensor

Im Folgenden werden der Aufbau, das physikalische Modell, sowie verschiedene mathematische Modelle des Sensors vorgestellt und erläutert.

#### 3.4.1. Das physikalische Modell des Sensors

Der in Abb. 3.1 dargestellte Sensor, gliedert sich nach folgender Darstellung in Abb. 3.2 auf in:

- Einer Lichtquelle: LED mit einer Strahlungsleistung  $\phi_e(\theta)$
- Einer Blende mit einer Transmissionsfunktion  $T(\varphi)$
- Einer Anordnung aus Photodioden, welche die transmittierte Lichtleistung als Spannungssignal  $U_{ph}(\phi)$  bzw.  $U_{ph}(\varphi)$  darstellt

#### Die LED

Die LED wird im folgenden als ein punktförmiger, lambert'scher Strahler betrachtet. Die gesamte Strahlungsleistung  $\phi_e$  wird in den Halbraum  $\omega = 2\pi$  abgestrahlt. Das Spektrum der LED wird in dieser Simulationsstudie nicht berücksichtigt.

#### Die Blende

Die Blende wird im folgenden als masselos und völlig lichtundurchlässig angenommen.

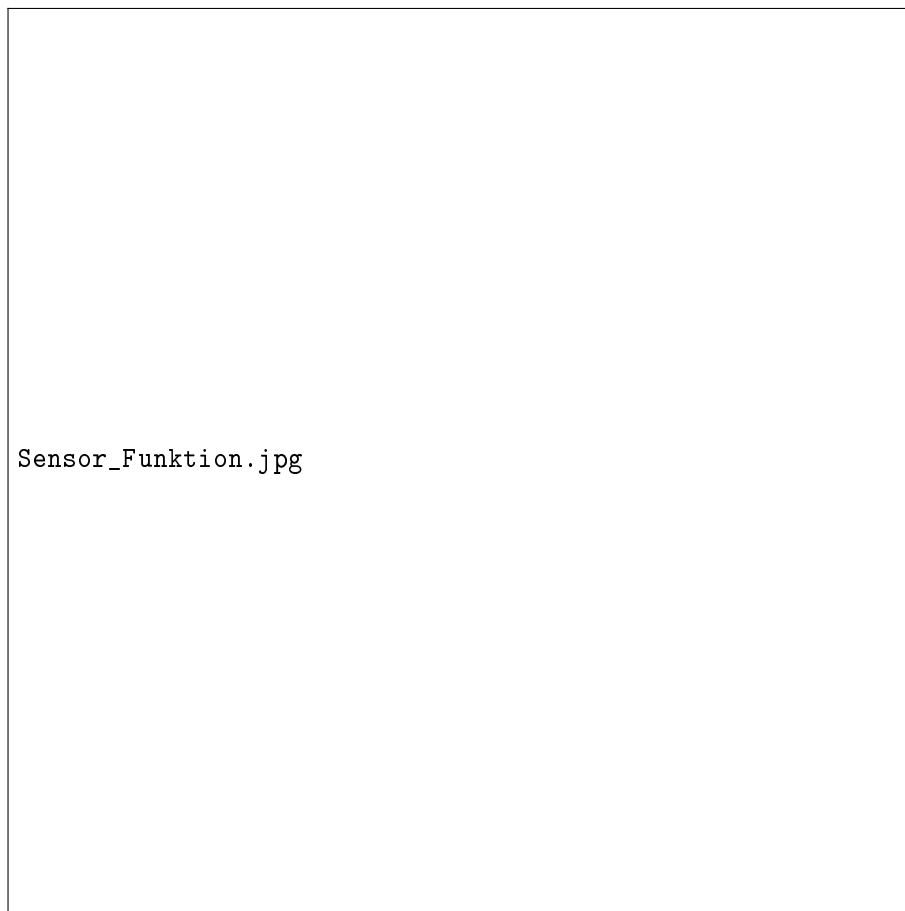


Abb. 3.2.: Allgemeines Funktionsdiagramm des Sensors

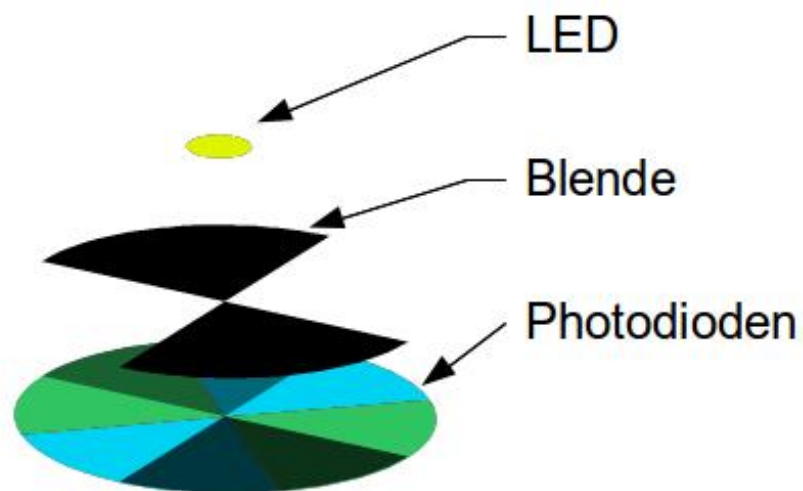


Abb. 3.3.: Allgemeiner Aufbau des Sensors

#### 3.4.2. Das lineare Sensormodell

#### 3.4.3. Das nicht lineare Sensormodell

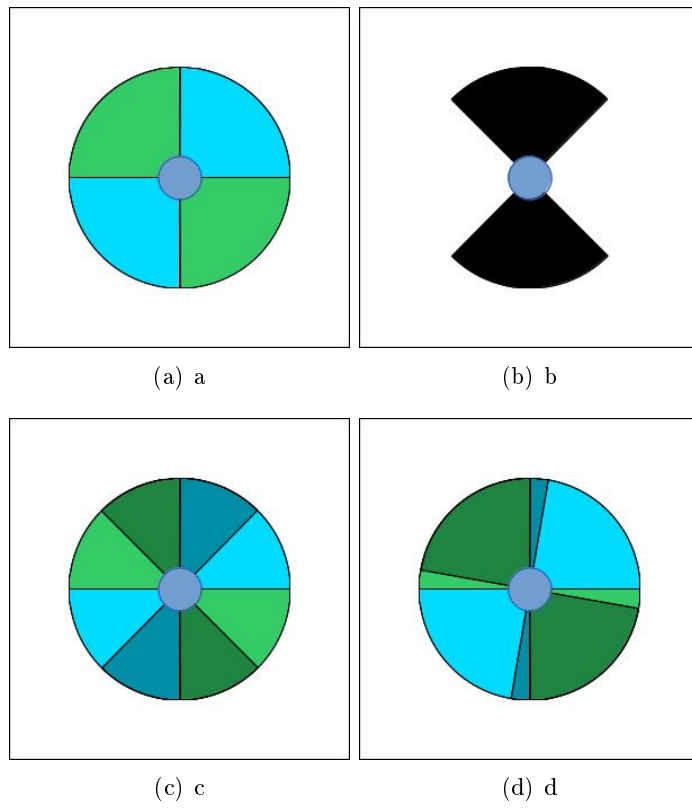


Abb. 3.4.: Sensor Funktionen





## 4. Programmentwicklung

Bei der Programmentwicklung werden die in Kap. 3 aufgestellten Gleichungen mit Matlab und Simulink umgesetzt. Es wird begonnen, die Gleichungen des elektrischen und des mechanischen Teils des Motors in Simulink umzusetzen. Im Anschluss folgt die Implementierung der Werte, des Simulinkprogrammes und des Motors in Matlab. Daran schliesst sich die Umsetzung der Sensoren in Matlab. Wenn die Sensoren mit Matlabfiles eingebunden werden können, werden die Simulink- und Matlabprogramme des Motors entsprechend erweitert.

### 4.1. Motor in Simulink

Es werden die Formeln ?? und ?? aus den Kap. ?? und ?? hergenommen. Durch ein umstellen der beiden Formeln, so dass nur noch erste Ableitungen in beiden Formeln vorkommen, lassen sie sich kombinieren und in Simulink einbinden, da so ein Gleichungssystem nur mit ersten Ableitungen entstanden ist. Um einen besseren Überblick zu bekommen, werden die Formeln hier noch einmal aufgeführt.

$$s i_A = \frac{1}{L_A} (e_A - R_A i_A + u_e) \quad (4.1)$$

$$e_a = K_M * \Phi \omega \quad (4.2)$$

$$s \omega = \frac{1}{J} (M_M - r * \omega - M_L) \quad (4.3)$$

$$M_M = K_M * \Phi * i_A \quad (4.4)$$

$K_M$  und  $\Phi$  sind Motorkonstanten.

Mit der Annahme das

$$x_1 := \omega$$

$$x_2 := i_A(4.5)$$

ist, lässt sich folgendes Gleichungssystem aufstellen:

$$\dot{\omega}_1 = \frac{1}{J}(K_M \Phi x_2 - r * x_1 - M_L)$$

$$\dot{x}_2 = \frac{1}{L_A}(K_M * \Phi x_1 - R_A x_2 + u_e)(4.6)$$

Dieses Gleichungssystem lässt sich jetzt durch die grafischen Elemente in Simulink sehr einfach modellieren.

Wie zu Beginn des Kap. 3.3 erwähnt, war es nicht möglich an verschiedene Werte der Motorkonstanten  $K_M$  und  $\phi$  zu gelangen. Aus diesem Grund wird auf die begleitenden Unterlagen der Vorlesung "Systemtechniken" von Prof. Froriep zurückgegriffen.

Auf dieser Grundlage werden die weiteren Programme entwickelt.

Um eine Regelung aufzubauen, wird noch ein Regler, ein Sollwertgeber und ein Subtrahierer von Ist- und Sollwert benötigt. Diese werden über die Simulinkbibliothek eingebunden und entsprechende Verbindungen werden angelegt. Das fertige Grundprogramm ist in Abb. 4.1 dargestellt.

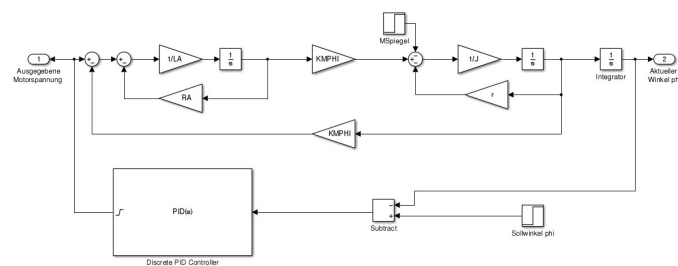


Abb. 4.1.: Simulink Grundprogramm

## 4.2. Matlab

### 4.2.1. Motor in Matlab

Die Programmentwicklung in Matlab gestaltet sich für den Motor als relativ einfach, da, wie oben erwähnt, keine Motordaten gefunden wurden, wird auf das Matlabfile von Prof. Froriep aus der Vorlesung "Systemtechniken" zurückgegriffen. In diesem Matlabfile stehen die Motorkennwerte, die berechnete Trägheit des Spiegels, das berechnete Drehmoment,

die Grenzen für die Plots, Anweisungen für die Plots und den Integrationsalgorithmus. Dieses File ist eine sehr gute Grundlage für die Simulation, welches während der Simulation entsprechend angepasst werden kann.

Als Gründe zur Ausgabe in einem Diagramm, interessieren vor allem die Eingangsspannung  $u_e$ , der aktuelle Winkel  $\phi$ , sowie der Sollwinkel mit seinen Toleranzen. Es werden drei Plots dargestellt. In dem ersten Plot ist die Motorspannung dargestellt. In dem zweiten Plot der aktuelle Winkel  $\phi$ , der direkt von dem Motor abgegriffen wird, sowie der einzustellende Sollwinkel dargestellt. Der dritte Plot enthält auch wieder den aktuellen Winkel  $\phi$ , jedoch mit einer feineren Auflösung um den Sollwinkel, um die Toleranzgrenzen besser erkennen zu können.

#### 4.2.2. Sensor in Matlab

Der Sensor selbst wird nur mit Matlabprogrammen simuliert. Dies ermöglicht verschiedene Sensoren in das Hauptprogramm einzubinden und Änderungen an z.B. den Ausmaßen des Sensors vorzunehmen, ohne das Hauptprogramm ändern zu müssen.

##### 4.2.2.1. Vorbereitungen

Um einen Sensor mit seinen verschiedenen Kenngrößen wie Sensorfläche, Übertragungsverhalten, und weiteres simulieren zu können, werden die verschiedenen Funktionen aus Kap. 3.4 in einzelnen Matlabfiles gespeichert. Dieses macht die Aufgabenlösung zwar komplexer, bietet aber den Vorteil, einzelne Bereiche für sich testen zu können, bevor sie in den Sensor eingebunden werden.

##### 4.2.2.2. Files

Ich würde hier evtl. schreiben, in welche Unterprogramme Du den Sensor aufgeteilt hast. Auch sollten Deine Versuchsprogramme erwähnt und mit in den Anhang kommen. Da steckt viel Arbeit drin und war/ist für die Simulation äußerst wichtig.



## 5. Simulationsdurchführung

### 5.1. Simulation

In diesem Abschnitt werden verschiedene Simulationen durchgeführt. Der Motor, der der Regelung zu Grunde liegt, ist der Gleichstrommotor aus der Vorlesung von Prof. Froriep. Mit diesem Motor soll von einer Nullposition ausgehend ein Winkel von  $20^\circ$  angefahren werden. Dieser Winkel soll innerhalb von einer Millisekunde erreicht werden.

Es wird eine Spannungsbegrenzung von  $\pm 24\text{ V}$  eingeführt, da diese eine in der Fertigung übliche Versorgungsspannung ist.

Zu Beginn wird der Sensor, der das aktuelle Positionssignal liefert, aus der Regelung heraus gelassen. Somit ist es möglich, die Regelung an den Motor anzupassen und sobald diese die Sollwerte erfüllt, werden 3 verschiedene Sensoren das Positionssignal liefern.

#### 5.1.1. chap:pidregelung

Für die verschiedenen P-, PI-, PD- und PID-Regelungen wird der PID-Reglerblock von Simulink verwendet.

Es wird mit einer P-Regelung begonnen, die Sollwerte zu erreichen. Wenn die P-Regelung nicht ausreicht, wird die P-Regelung erst nur um einen I-Anteil und dann nur um einen D-Anteil erweitert. Sollten immernoch keine Zufriedenstellenden Ergebnisse vorliegen, so wird mit einer PID-Regelung versucht, die Vorgaben zu erreichen.

##### 5.1.1.1. chap:p<sub>r</sub>egelung

In Abb. ?? ist das Ergebnis der reinen P-Regelung dargestellt. Es ist zu erkennen, dass nach ca. 7 ms es keine Veränderung des eingesetllten Winkels gibt. Eine Erhöhung des P-Anteils ergibt ein Überschwingen, wie es in Abb. ?? dargestellt ist. In Abb. ?? und ?? ist in der untersten Grafik der Sollwinkel sowie die angegebene Abweichung angezeigt. Wie zu erkennen ist, ist die verbleibende Regeldifferenz noch viel zu groß. Demnach wird mit einem zugefügten I-Anteil zur reinen P-Regelung versucht, die restliche große Regeldifferenz auszugleichen. Für die folgenden Simulationen wird das Matlab-File "*msSpiegelPID.münddasSimulink-FileßSpiegel.slx*" hergenommen.

##### 5.1.1.2. chap:pi<sub>r</sub>egelung

In Abb. 5.4 ist das Ergebnis der PI-Regelung dargestellt. Es ist zu erkennen, dass nach ca. 13 ms es keine Veränderung des eingesetllten Winkels gibt. Eine Erhöhung des P- oder I-

Anteils ergibt ein Überschwingen, wie es in Abb. 5.4 dargestellt ist. In Abb. 5.4 und 5.4 ist in der untersten Grafik der Sollwinkel sowie die angegebene Abweichung angezeigt. Wie zu erkennen ist, ist die verbleibende Regeldifferenz noch viel zu groß. Demnach wird der zugefügte I-Anteil herausgenommen und ein D-Anteil zur reinen P-Regelung hinzugenommen, um so ein besseres Regelergebnis zu erreichen.

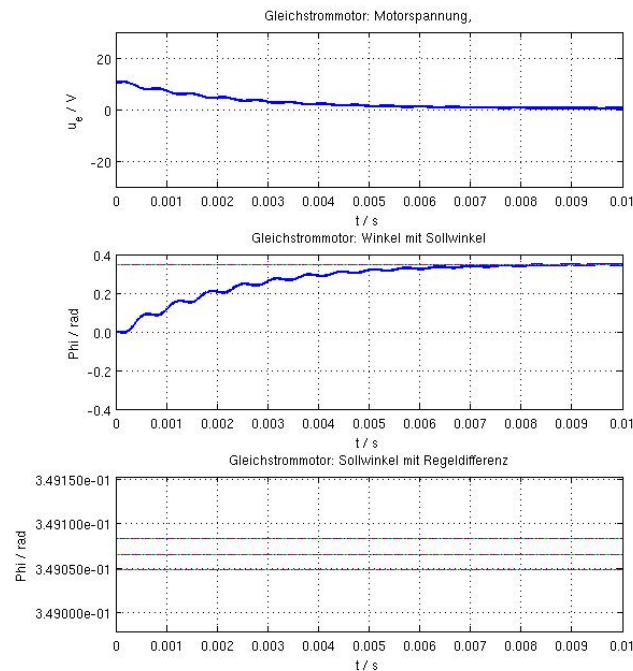


Abb. 5.3.: P-Anteil von 30 und I-Anteil von 17

### 5.1.1.3. chap:pd<sub>regelung</sub>

Auch mit der PD-Regelung werden die Vorgaben noch nicht erfüllt. In Abb. 5.5 ist das Ergebnis der PD-Regelung dargestellt. Es ist zu erkennen, dass nach ca. 7 ms es keine Veränderung des eingesetllten Winkels gibt. Eine Erhöhung des P- oder D-Anteils ergibt ein Überschwingen, wie es in Abb. 5.6 dargestellt ist. In Abb. 5.5 und 5.6 ist in der untersten Grafik der Sollwinkel sowie die angegebene Abweichung angezeigt. Wie zu erkennen ist, ist die verbleibende Regeldifferenz noch viel zu groß.

### 5.1.1.4. chap:pid<sub>regelung</sub>

Nun wird mit einer Kombination der P-, I- und D-Anteile die Regelung betrieben. In Abb. ?? ist das Ergebnis der PID-Regelung dargestellt. Es ist zu erkennen, dass nach ca. 7 ms es keine Veränderung des eingesetllten Winkels gibt. Eine Erhöhung der verschiedenen Reglerparamteranteile ergibt ein Überschwingen, wie es in Abb. ?? dargestellt ist. In Abb. ?? ist in der untersten Grafik der Sollwinkel sowie die angegebene Abweichung angezeigt.

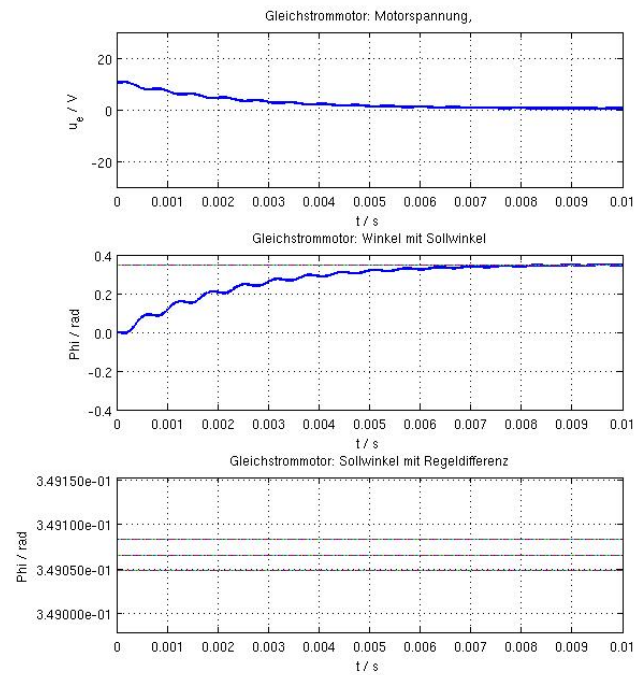


Abb. 5.4.: P-Anteil von 30 und I-Anteil von 17

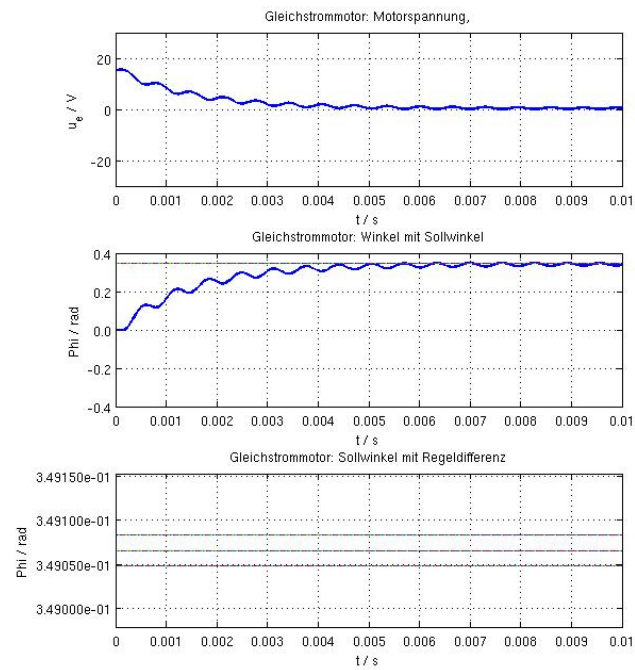


Abb. 5.5.: P=22 - D=1 - N=1

Durch die große Abweichung vom Sollwinkel ist in dieser Grafik kein Graph zu erkennen.

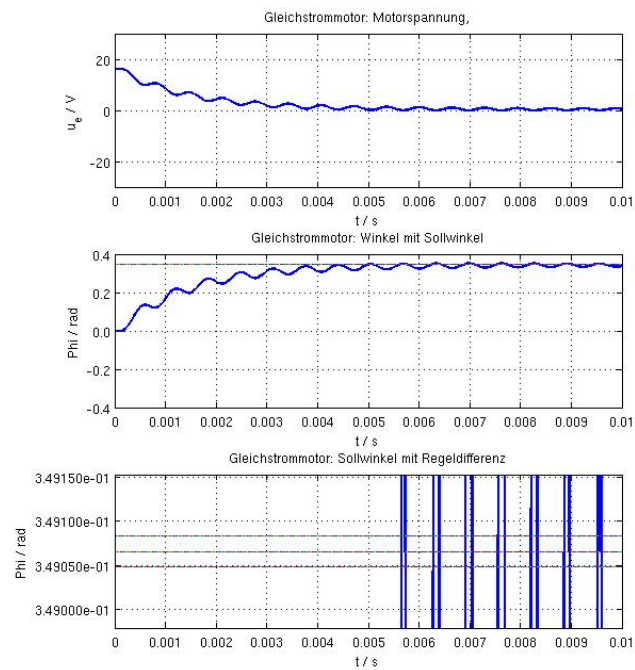


Abb. 5.6.: P=22 - D=1 - N=1

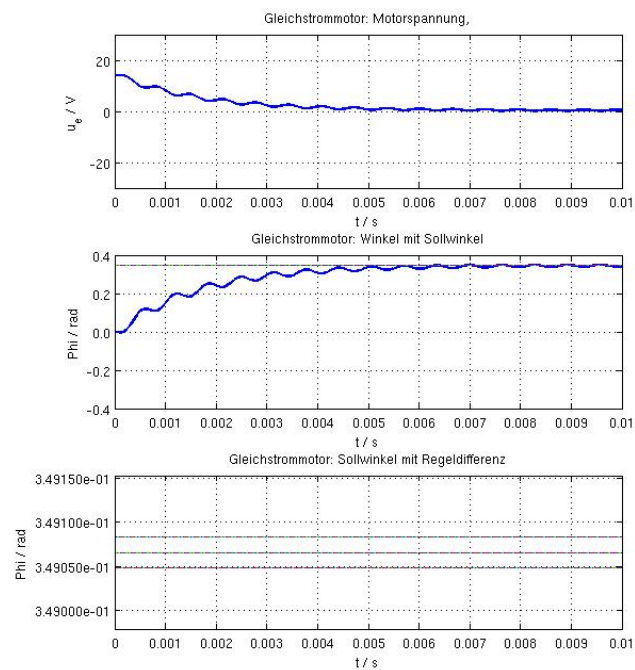


Abb. 5.7.: P=20 - I=15 - D=1 - N=1



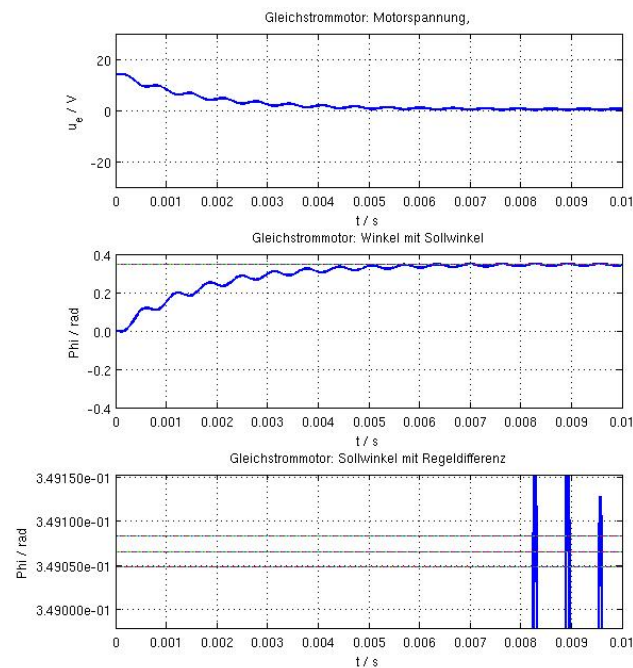


Abb. 5.8.: P=20 - I=16 - D=1 - N=1

### 5.1.2. chap:padaption

Es zeigt sich, dass der P-Anteil den meisten Einfluss, bzw. den größten Erfolg bei der Regelung ausmacht. Durch hinzugefügte I- oder D-Anteile konnte die Regelung nicht verbessert werden. Nach dem die verschiedenen Regler die Vorgaben noch nicht erfüllen konnten, wird nun die P-Adaption eingesetzt. Bei der P-Adaption wird folgende Formel vor den P-Verstärker geschaltet:

$$f = 1 + \frac{c_1 - 1}{(c_2 * e)^2 + 1} \quad (5.1)$$

Dabei muss der Regelkreis folgendermaßen erweitert werden: FrKu: PDF, "Tempo beim Laserzugriff", Artikel im FM Elektronik, Jahrg.111(2003)5, Lugmair, Froiepe, Kuplent, Langhans

#### 5.1.2.1. chap:p\_adaption

Nun kann mit drei verschiedenen Parametern versucht werden, die Sollwerte zu erreichen. Wobei die beiden f-Parameter zu Beginn auf 1 gesetzt werden und erst mit dem Erhöhen des P-Anteils versucht wird, eine gute Regelung zu erhalten. Danach werden die beiden f-Parameter einzeln erhöht bzw. erniedrigt, bis sich das Ergebnis verbessert hat. Eine Anpassung des P-Anteils und danach eine erneute Anpassung der f-Parameter gehört ebenso zur

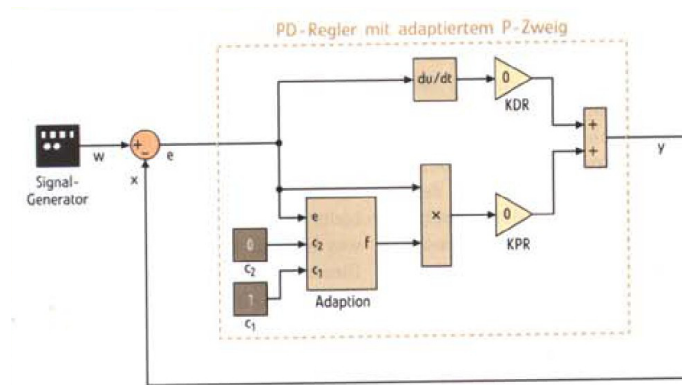


Abb. 5.9.: P-Adaption [?]

Erreichung einer passenden Regelung. Für die folgenden Simulationen wird das Matlab-File "msSpiegel $_{pad}$ .münddasSimulink–FileßSpiegelPad.slx" hergenommen. In Abb. ?? ist zu erkennen, dass der Sollwinkel nie erreicht wird, weil die Parameter  $f_1$  und  $f_2$  nicht angepasst wurden.

#### 5.1.2.2. chap:p $_{adaption}$ galvo

Durch die Verwendung der P-Adaption konnte die Einregelzeit nicht verbessert werden. Es zeigt sich, dass mit dem vorhandenen Gleichstrommotor keine der Vorgaben eingehalten werden können. Um heraus zu finden, welche Daten der Motor aufweisen müsste, um mit einer PID- oder P-Adaption geregelt werden zu können, werden jetzt zusätzlich zu den  $f_1$  und  $f_2$  Parametern auch die Motorparameter geändert. Beispielhaft wurden Werte für den Innenwiderstand und der Induktivität eines Galvos 6230 der Firma Cambridge Technology als Grundlage verwendet [?].

CaTe: PDF, Model 6230H Optical Scanner (Mechanical and Electrical Specifications), Cambridge Technology, 03/07.

Für die folgenden Simulationen wird das Matlab-File "msSpiegel $_{pad}$ Werte.münddasSimulink–FileßSpiegelPad.slx" hergenommen.

In Abb. 5.12 ist eine langsame Annäherung an die zu erfüllenden Vorgaben zu sehen. Jedoch noch nicht in der geforderten Zeit und noch mit zu großen Schwankungen um den Sollwinkel. Es sind folgende Werte aktuell eingestellt:

- Innenwiderstand der Spule:  $1.07 \Omega$  Induktivität der Spule :  $173 \mu H$
- P-Anteil: 330
- $f_1$ : 5
- $f_2$ : 370

#### 5.1.2.3. chap:p $_{adaption}$ werte

Nun werden die Motor- und Spiegelwerte solange verändert, bis sich das gewünschte Ergebnis einstellt. Sollte die Regelung erfolgreich sein, könnte mit den veränderten Werten evtl.

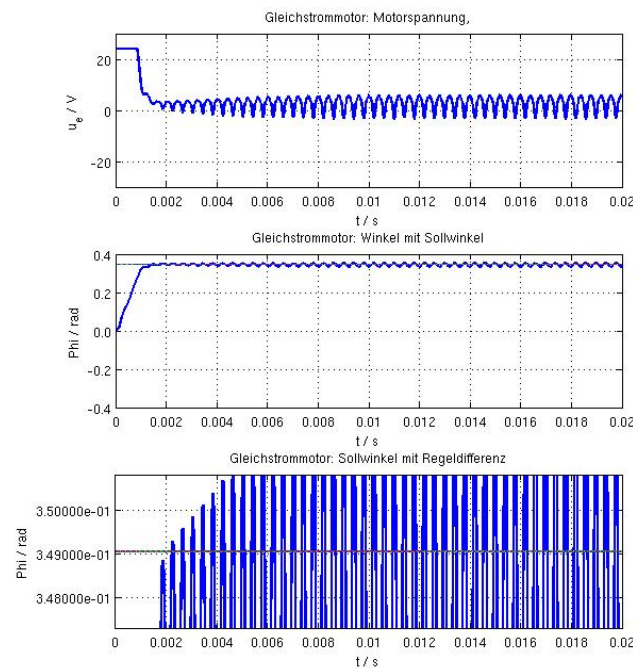


Abb. 5.12.: P-Adaption mit neuen Motorparametern

ein Motor und Spiegel hergestellt werden, der den Anforderungen entspricht.

Für die folgenden Simulationen wird das Matlab-File "`msSpiegelpadNeueWerte.münddasSimulink-FileßSpiegelPad.slx`" hergenommen. Wie in Abb. ?? zu erkennen, ist die Regelung in den geforderten Bereichen

Innenwiderstand der Spule:  $0.1 \Omega$  Induktivität der Spule:  $3 \mu H$

Motorkonstante  $K_{MPHI}$ :  $35e-3 \text{ Vs}$

Reibungskoeffizient:  $6e-5 \text{ Nm*s}$

Trägheitsmoment des Spiegels:  $93.3e-9 \text{ kg * m}^2$

Drehmoment auf den Spiegel:  $130.25e-6 \text{ Nm}$

P-Anteil: 320

$f_1$ : 2

$f_2$ : 160

#### 5.1.2.4. chap:p<sub>a</sub>daptionstrom

Ein Blick auf den Strom liefert allerdings Ergebnisse, die weiterer Überarbeitung der Regelung bedürfen. In Abb. 5.14 ist der Strom der aktuellen Regelung dargestellt. Es fließen Ströme in Höhe von 80 A. Dies ist allerdings sehr hoch, deshalb wird in die bestehende Regelung eine Strombegrenzung von 10 A eingebaut und erneut versucht, die

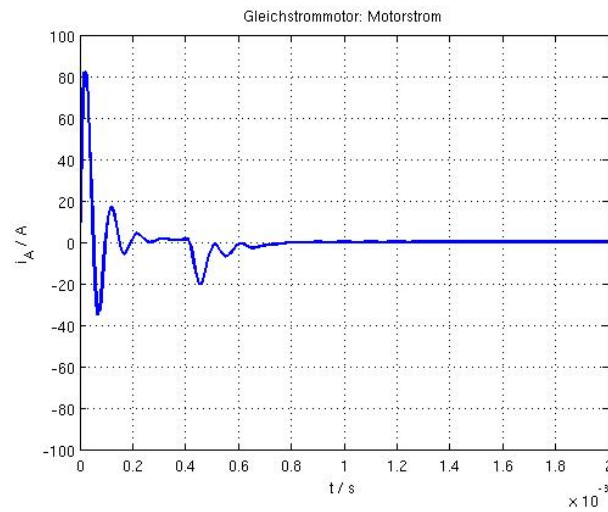


Abb. 5.14.: Stromhöhe während der Regelung

Regelung entsprechende anzupassen. In Abb. 5.15 ist die P-Adaption mit einer Strombe-

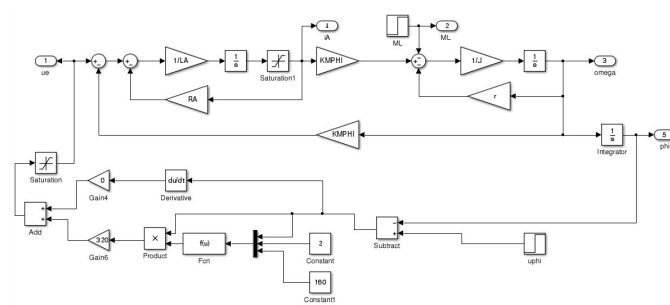


Abb. 5.15.: P-Adaption mit Strombegrenzung

grenzung zu erkennen. Für eine bessere Übersicht, wurden entsprechende Positionen mit einem Namen versehen und der D-Anteil aus der Regelung genommen, da dieser auf Null gesetzt ist, siehe Abb. 5.16. Für die folgenden Simulationen wird das Matlab-File

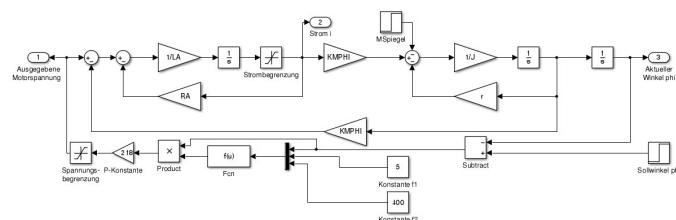


Abb. 5.16.: Stromhöhe während der Regelung

"msSpiegelpadNeueWertestrom.münddasSimulink-FileßSpiegelPadStrom.slx" hergenommen.

Eine Regelung mit den aktuellen Werten zeigt das in Abb. 5.17 zu erkennende Ergebnis. Durch weiteres Anpassen der unterschiedlichen Parameter, konnten die Sollwerte fast erreicht werden. Abb. 5.18 zeigt schon ein sehr gutes Ergebnis.

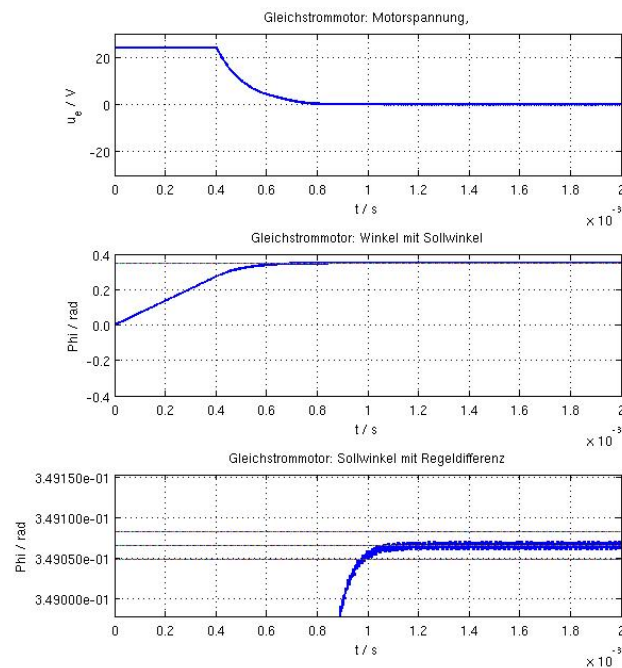


Abb. 5.17.: P-Adaption mit Strombegrenzung

- Neues Trägheitsmoment des Spiegels:  $93.3\text{e-}11 \text{ kg} * \text{m}^2$
- Neues Drehmoment auf den Spiegel:  $30,25\text{e-}6 \text{ Nm}$
- P-Anteil: 218
- $f_1$ : 5
- $f_2$ : 400

#### 5.1.2.5. chap:p<sub>a</sub>daptionfertig

Die Einregelzeit liegt nur noch knapp über der vorgegebenen Zeit, durch weitere Anpassung der Regelparameter soll die vorgegebene Einregelzeit erreicht werden. Abb. ??

- Trägheitsmoment des Spiegels:  $93.3\text{e-}11 \text{ kg} * \text{m}^2$
- Drehmoment auf den Spiegel:  $30,25\text{e-}6 \text{ Nm}$  -Anteil: 250
- $f_1$ : 3
- $f_2$ : 150

In Abb. 5.19 ist zu erkennen, dass durch Anpassung des Trägheitsmoments des Spiegels, des wirkenden Drehmoments auf den Spiegel und der verschiedenen Regelparameter, trotz Spannungs- und Strombegrenzung, die Regelung erfolgreich ist. Der Spiegel zittert zwar etwas um die Position, dies ist aber im angegebenen Toleranzbereich.

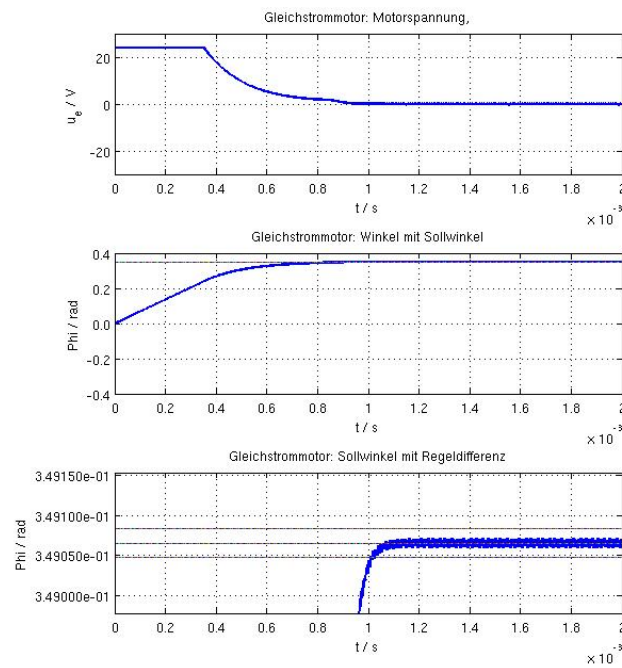


Abb. 5.18.: Angepasste P-Adaption mit Strombegrenzung

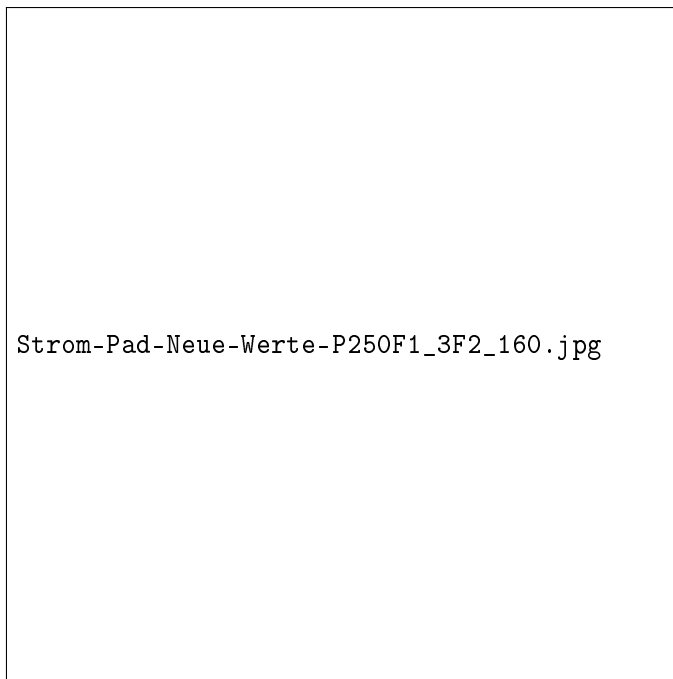


Abb. 5.19.: Angepasste P-Adaption mit Strombegrenzung

### 5.1.3. chap:sensorregelung

Nach dem die Regelung für einen perfekt linear arbeitenden Sensor funktioniert, wird der Sensor aus Kap ?? in die Simulation mit eingebaut. Dafür wird das vorhandene Simulink-

File `ßSpiegelPadStrom.slx` hergenommen und um den Sensor erweitert. Zudem wird der eingegebene Winkel in eine Spannung umgerechnet, da der Sensor eine vom Winkel abhängige Spannung ausgibt. In Abb. ?? ist der gesamte Simulationsaufbau dargestellt, welcher als `ßSpiegelPadStromSensor.slx` gespeichert ist.

Zur Ansteuerung wird das Matlab-File "`msSpiegelpadNeueWertestrom.m`" modifiziert und als "`msSpiegelpadNeueWertestrom.m`" gespeichert. Es werden folgende Daten hergenommen:

Innenradius = 5 mm

Aussenradius = 10 mm

Lastwiderstand = 6000 Ohm

Messbereich =  $20/180 \cdot \pi$  ( $20^\circ$ )

LEDLeistung = 1 W

Umgebungstemperatur = 300 K

nonlinear = 0.0001 (falls ein nichtlinearer Sensor simuliert werden soll)

Es werden drei verschiedenen Simulationen durchgeführt.

In der ersten Simulation wird ein linearer Sensor verwendet. Dies sollte die gleichen Ergebnisse liefern wie der ideale Sensor. Hierbei wird die Regelung, falls nötig, angepasst.

Die zweite Simulation wird ebenfalls mit einem linearen Sensor durchgeführt, nur ist hierbei die Linearität durch eine Faltung der Blende mit der Sensorfläche realisiert worden. Es wird erwartet, dass sich dieser Sensor gleich verhält wie der ideale Sensor.

Bei der dritten und letzten Simulation wird ein nichtlinearer Sensor verwendet. Wobei der Nichtlinearitätsfaktor, der Werte zwischen 0 und 1 annehmen kann, auf den Wert 0,0001 gesetzt wird. Dies sollte ein nahezu lineares Verhalten zeigen.

#### 5.1.3.1. chap:sensorregelung1

Abb. 5.20 zeigt das Regelergebnis des ersten linearen Sensors. Es ist gut zu erkennen, dass die Regelung sogar schneller erfolgt als vorher.

#### 5.1.3.2. chap:sensorregelung2

Abb. 5.21 zeigt das Regelergebnis des zweiten linearen Sensors. Es ist gut zu erkennen, dass eine Regeldifferenz übrig bleibt. Diese lässt sich auch nicht durch verändern der Regelparameter reduzieren.

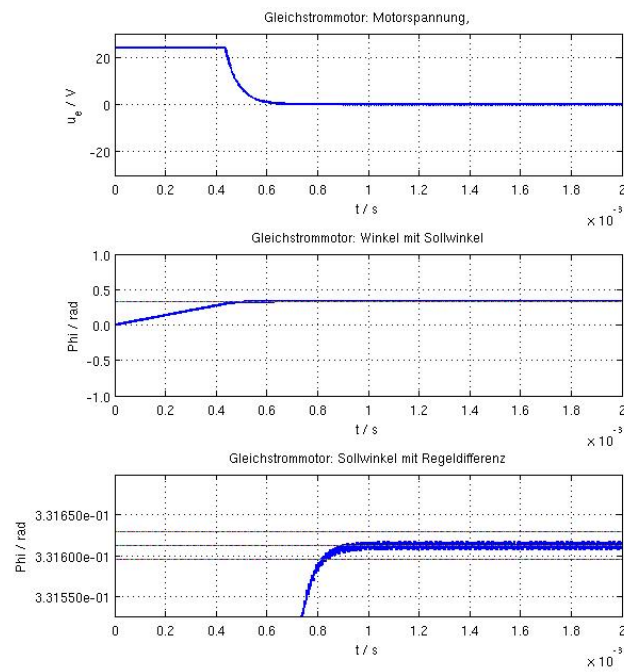


Abb. 5.20.: Erster linearer Sensor

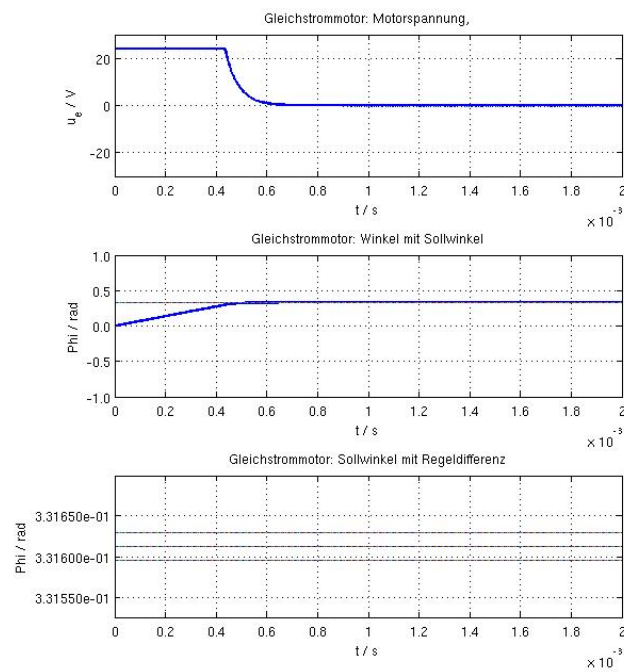


Abb. 5.21.: Zweier linearer Sensor



### 5.1.3.3. chap:sensorregelung3

Abb. 5.22 zeigt das Regelergebnis des nicht linearen Sensors. Auch hier ist eine restliche Regeldifferenz zu erkennen, die sich wiederum nicht durch anpassen der Regelparemeter begleichen lässt.

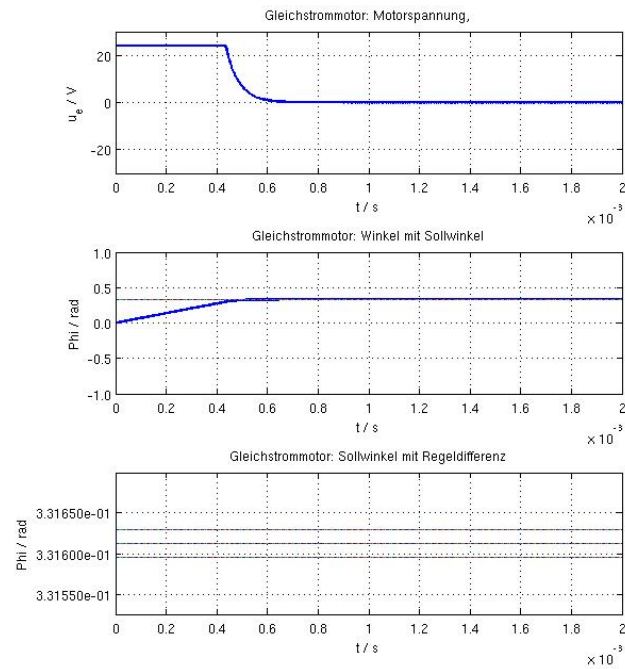


Abb. 5.22.: Linearer Sensor



## 6. Diskussion

text



## 7. Zusammenfassung

text



## A. Matlab-Files

### A.1. msSpiegelundSensor.m

```
1 % msSpiegel_Pad.m      (Matlab/Simulink R2011b)
2 %
3 % Vorgang: Regelung eines Gleichstrommotors zur
4 % Spiegelverstellung
5 % Verfahren: Simulink, mithilfe einer P-Adaption, eingebauter
6 % Strom-
7 % begrenzung, angepasster Motorenwerte und
8 % eingebauter
9 % Sensor
10 %
11 % Unterprogramme: sSpiegelPadStromNeu.slx
12 % sensor.m
13 % posEingabe.m
14 %
15 % #####
16 %
17 % Parameterbeschreibung:
18 %
19 % RA          Ankerwiderstand
20 % LA          Ankerinduktivität
21 %
22 % J           Traegheitsmoment
23 % r           Reibkonstante
24 %
25 % MSpiegel    Drehmoment  $f \frac{1}{4} r$  Spiegel
26 %
27 % KMPHI       Motorkenngrößen
28 %
29 % phi         Sprunghöhe Motorwinkel
30 %
31 % te          Ende des Integrationsintervalls (ab t=0)
32 %
33 % xuX, xoX     Untere/obere Grenze der Grafiken
34 % 0
35 % #####
36 clear all
37 close all
38
39 % Funktionen die benoetigt werden sind hier gespeichert
```

```

37 addpath('/home/michamann/git/syt_ss13/');
38
39 % Auswahl Sensorverhalten
40 global mode
41 mode = 'nonlinear';           % linear1, linear2, nonlinear
42 global Unit
43 Unit = 'rad';
44
45 % Kennwerte fuer Sensor
46 Innenradius =5;              % in mm
47 Aussenradius =10;            % in mm
48 Lastwiderstand =6000;        % in Ohm
49
50 Photodioden = [Innenradius,Aussenradius,Lastwiderstand];
51 Messbereich = 20/180*pi;      % 45° in rad
52 LEDLeistung = 1;             % in W
53 Umgebungstemperatur = 300;    % in K
54 nonlinear = 0.0001;          % Wert zwischen 0 und 1
55
56 % Cell-Array fuer Kennwerte von Sensor
57 global Sensorkonstanten
58 Sensorkonstanten = sensorDaten(Photodioden,...
59                               Messbereich,...
60                               LEDLeistung,...
61                               Umgebungstemperatur,...
62                               nonlinear);
63
64
65 % Angabe der Parameter fuer Simulink fuer die weiteren
    Berechnungen
66 RA=0.1;                      % Innenwiderstand
67 LA=3e-6;                     % Induktivitaet
68 TA=LA/RA;                    % Zeitkonstante T1
69
70 J=93.3e-11;                  % kg m^2 Traegheitsmoment des
    Spiegels
71 r=6e-5;                      % Nm*s Reibung
72
73 KMPHI=35e-3;                 % Vs Motorkonstante
74
75 MSpiegel=30.25e-6;           % Nm Drehmoment fuer Spiegel
76
77 te=.002;                     % end of simulation time
78
79 phi = 19*pi/180;             % einzustellender Winkel
80
81 vu=-30;                      % uu=-30 V
82 vo=30;                       % uo=+30 V
83 iu=-15;                      % iu=-15 A
84 io=+15;                      % io=+15 A

```



```

85     pu1=-1;                                %  $\phi_u = -20^\circ$  in rad 0.4
86     po1=1;                                %  $\phi_o = +20^\circ$  in rad -0.4
87     pu2=phi-0.5e-2*pi/180;                % Diagrammgrenzen fuer
      RegelDifferenz
88     po2=phi+0.5e-2*pi/180;                % Diagrammgrenzen fuer
      RegelDifferenz
89
90     % #####
91
92     % Plot: Eingangssignal u und Winkel phi
93     figure(1)
94     set(gcf,'Units','normal','Position',[.49 .7 .5 .9], ...
95         'NumberTitle','on','Name','u und v ');
96
97     % Integrationsalgorithmus:
98     t0=0;
99     opts=simset('solver','ode45',...
100         'InitialState',[],...
101         'Refine',1,...
102         'MaxStep',.00001);
103
104     [t,x,y]=sim('sSpiegelPadStromSensor',[t0 te],opts);
105
106     % Plots
107     subplot(3,1,1)
108     plot (t,y(:,1),'linewidth',2)
109     axis([0 te vu vo])
110     grid on
111     hold on
112     xlabel('t / s')
113     ylabel('u_e / V')
114     title('Gleichstrommotor: Motorspannung,')
115
116     subplot(3,1,2)
117     plot(t,y(:,2),t,phi,'linewidth',2,'linewidth',2);
118     axis([0 te pu1 po1])
119     grid on
120     xlabel('t / s')
121     ylabel('Phi / rad')
122     YTicks=get(gca,'YTick');
123     set(gca,'YTickLabel',num2str(YTicks(:),'%.1f'));
124     title('Gleichstrommotor: Winkel mit Sollwinkel')
125
126     subplot(3,1,3)
127     plot(t,y(:,2),...
128         t,phi,...
129         t,(phi-1e-3*pi/180),...
130         t,(phi+1e-3*pi/180),...
131         'linewidth',2,...
132         'linewidth',2,...

```

```

133     'linewidth',2,...
134     'linewidth',2);
135 axis([0 te pu2 po2])
136 grid on
137 xlabel('t / s')
138 ylabel('Phi / rad')
139 YTicks=get(gca,'YTick');
140 set(gca,'YTickLabel',num2str(YTicks(:),'%.5e'));
141 title('Gleichstrommotor: Sollwinkel mit Regeldifferenz')
142
143
144 % Möglichkeit, Strom und Winkelspannung auszugeben
145
146 % figure(2)
147 % plot(t,y(:,4),'linewidth',2);
148 % axis([0 te iu io])
149 % grid on
150 % xlabel('t / s')
151 % ylabel('i_A / A')
152 % title('Gleichstrommotor: Motorstrom')
153 % figure(2)
154 % plot(t,y(:,7),'linewidth',2);
155 % axis([0 te vu vo])
156 % grid on
157 % xlabel('t / s')
158 % ylabel('U / V')
159 % title('Sensor: aktuelle Winkelspannung')
160 % figure(3)
161 % plot(t,y(:,6),'linewidth',2);
162 % axis([0 te vu vo])
163 % grid on
164 % xlabel('t / s')
165 % ylabel('U / V')
166 % title('Eingabe: Sollwinkelspannung')
167
168 % Plot der variablen Schrittweite
169 % ht=diff(t)';
170 % ht=[ht ht(end)]';
171 % set(gcf,'Units','normal','Position',[.1 .2 .4 .2], ...
172 %     'NumberTitle','on','Name','h');
173 % plot(t,ht,'x','markersize',9,'linewidth',2);
174 % grid on
175 % xlabel('t / s')
176 % ylabel('h / s')
177 % title('Schrittweite')
178 %end

```

## A.2. sensorDaten.m

```

1  %% Sensorkonstanten = sensorDaten(Photodioden,
2  %                                     Messbereich,
3  %                                     LEDLeistung,
4  %                                     Umgebungstemperatur)
5  % Erzeugt:
6  %     [Out]:
7  %     Array der Sensorkonstanten => Sensorkonstanten
8  %         1 signal_max in A
9  %         2 Lastwiderstand der Idealen Photodiode in Ohm
10 %         3 Azimutwinkel der Photodioden/Maximaler
    Messbereich in rad
11 %         4 signal_max in V
12 %         5 X Koordinaten der linearen Kennlinie
13 %         6 X Koordinaten der nicht linearen Kennlinie
14 %         7 Y- / Funktions-Werte der linearen Kennlinie
15 %         8 Y- / Funktions-Werte der nicht linearen Kennlinie
16 %
17 %     [In]:
18 %     Array der Diodendaten(geometrisch, elektrisch) =>
    Photodioden
19 %         1 Innererradius r1                in mm
20 %         2 Äußererradius r2                in mm
21 %         3 Lastwiderstand                    in Ohm
22 %
23 %         Maximal messbarer Winkel +/- => Messbereich in rad
24 %
25 %         Gesamtleistung der LED => LEDLeistung in W
26 %         Umgebungstemperatur => Umgebungstemperatur in K
27
28 function Sensorkonstanten = sensorDaten(Photodioden,...
29                                         Messbereich,...
30                                         LEDLeistung,...
31                                         Umgebungstemperatur
32                                         ,...
33                                         nonlinear)
34 Sensorkonstanten{1}=0.001; % max Strom
35 Sensorkonstanten{2}=Photodioden(3); % Lastwiderstand
36 Sensorkonstanten{3}=Messbereich; % max Messbereich
37 Sensorkonstanten{4}=Sensorkonstanten{1}*Photodioden(3)*4;%max
    Spannung
38 %Sensorcharakteristik mittels Faltung von Fensterfunktionen
39 xmin = -2*pi;
40 xmax = 2*pi;
41 x = linspace(xmin,xmax,4000);
42
43 c = 1;
44 e = nonlinear;

```

```
45 b = Messbereich*2;
46 pb = 0;
47 ps1 = b/2;
48 g = [];
49 r = [];
50 s1 = [];
51 s2 = [];
52 for i = x
53     r(end+1) = frect(i,b,pb);
54     s1(end+1) = frect(i,b,ps1);
55     s2(end+1) = frect(i,b,-ps1);
56 end
57 r = 1-r;
58 s1 = 1-s1;
59 s2 = 1-s2;
60 f1 = conv(r,s1);
61 f1 = f1./max(f1);
62 x21 = linspace(2*min(x),2*max(x),size(f1,2));
63 f2 = conv(r,s2);
64 f2 = f2./max(f2);
65 Sensorkonstanten{5} = linspace(2*min(x),2*max(x),size(f2,2));
66 for i = x21
67     g(end+1) = gekern(i,e,c);
68 end
69 g = g./max(g);
70
71 gf1 = conv(g,f1);
72 gf1 = gf1./max(gf1);
73 gf2 = conv(g,f2);
74 gf2 = gf2./max(gf2);
75 Sensorkonstanten{6} = linspace(4*min(x),4*max(x),size(gf1,2));
76 f = f1-f2;
77 Sensorkonstanten{7} = f./max(f);
78 gf = gf1-gf2;
79 Sensorkonstanten{8} = gf./max(gf);
80
81 Sensorkonstanten{9} = Umgebungstemperatur;
82
83 end
```

### A.3. frect.m

```
1 function rect = frect(x,breit,pos)
2
3 while abs(x)/(pi/2) > 1
4     x = x - sign(x)*pi;
5 end
6
7 if and(x > (pos-breit/2),x<(pos+breit/2))
8     rect = 1;
9 else
10    rect = 0;
11 end
12
13 end
```

### A.4. gekern.m

```
1 function glatt = gekern(x,e,c)
2 xe = x/e;
3
4 if abs(xe) < 1
5     glatt = c*exp(-1/(1-xe^2));
6 else
7     glatt = 0;
8 end
9
10 glatt = 1/e * glatt;
11
12 end
```

## A.5. sensor.m

```

1  %% signal = sensor( Blendenwinkel, Unit, mode,
    Sensorkonstanten)
2  % Erzeugt:
3  %     [Out]:
4  %     Normiertes Spannungssignal +-5V => signal in V
5  %
6  %     [In]:
7  %     aus Eingangswinkel => Blendenwinkel in
8  %     der angegebenen Einheit => Unit
9  %     Kennmode des Sensors => mode
10 %     Array aus Sensorkonstanten => Sensorkonstanten
11 %         1 signal_max in A
12 %         2 Lastwiderstand der Idealen Photodiode in Ohm
13 %         3 Azimutwinkel der Photodiodennonlinear
14 %         4 signal_max in V
15 %     Unit cases: 'grad', 'mgrad', 'rad', 'mrad'
16 %     mode cases : 'linear', ...
17
18 function signal = sensor(Blendenwinkel)
19
20 global Unit
21 global mode
22 global Sensorkonstanten
23
24 %% Umrechnung der Eingangseinheit
25 switch Unit
26     case 'grad'
27         phiB = Blendenwinkel/180*pi;
28     case 'mgrad'
29         phiB = Blendenwinkel/1000/180*pi;
30     case 'rad'
31         phiB = Blendenwinkel;
32     case 'mrad'
33         phiB = Blendenwinkel/1000;
34 end
35 %Rauschen aus Temperaturspannung
36 % kb = 8.6173324*10^-5; % eV/K
37 % e = 1; % eV
38 % T = Sensorkonstanten{9};
39 % RLast = Sensorkonstanten{2};
40 % zufall = 0.5-rand();
41 % UT = kb*T/e/RLast % in V
42 % Rauschen = UT+UT*zufall
43
44 Rauschen = rand()*Sensorkonstanten{1}/10;
45
46 switch mode
47     case 'einzelGruppe'

```

```

48
49     a=@(phi)1/Sensorkonstanten{3}/2*phi+0.5;% lineares
      Model
50
51     S1=@(phi,d)Sensorkonstanten{1}*(a(phi))+d;% Signal
      Sensor 1
52     S3=@(phi,d)Sensorkonstanten{1}*(a(phi))+d;% Signal
      Sensor 3
53     S2=@(phi,d)Sensorkonstanten{1}*(1-a(phi))+d;% Signal
      Sensor 2
54     S4=@(phi,d)Sensorkonstanten{1}*(1-a(phi))+d;% Signal
      Sensor 4
55
56     S13=@(phi,d)S1(phi,d)+S3(phi,d);
57     S24=@(phi,d)S2(phi,d)+S4(phi,d);
58
59     Iph=@(phi,d)S13(phi,d);% Gesamt Photostrom
60
61     signal=Iph(phiB,Rauschen)*Sensorkonstanten{2};% [Out]
62
63 case 'linear1'
64     a=@(phi)1/Sensorkonstanten{3}*phi+0.5;% lineares Model
65
66     S1=@(phi,d)Sensorkonstanten{1}*(a(phi))+d;% Signal
      Sensor 1
67     S3=@(phi,d)-Sensorkonstanten{1}*(a(phi))+d;% Signal
      Sensor 3
68     S2=@(phi,d)Sensorkonstanten{1}*(1-a(phi))+d;% Signal
      Sensor 2
69     S4=@(phi,d)-Sensorkonstanten{1}*(1-a(phi))+d;% Signal
      Sensor 4
70
71     S13=@(phi,d)S1(phi,d)-S3(phi,d);
72     S24=@(phi,d)S2(phi,d)-S4(phi,d);
73
74     Iph=@(phi,d)S13(phi,d)-S24(phi,d); % Gesamt
      Photostrom
75     signal=Iph(phiB,Rauschen)*Sensorkonstanten{2}; % [Out]
76
77 case 'linear2'
78     AP=abs(Sensorkonstanten{5}-phiB);
79     ind1=find(AP==min(AP));
80     ind2=find(AP==min(AP(AP~=min(AP))));
81     X1=Sensorkonstanten{5}(ind1);
82     X2=Sensorkonstanten{5}(ind2);
83     Y1=Sensorkonstanten{7}(ind1);
84     Y2=Sensorkonstanten{7}(ind2);
85
86     IO=Sensorkonstanten{1}*4;

```

```

87         Iph=@(phiB) I0*((Y2-Y1)/(X2-X1)*phiB+(X2*Y1-X1*Y2)/(X2-
88             X1));
89         signal=Iph(phiB)*Sensorkonstanten{2};
90     case 'nonlinear'
91         AP=abs(Sensorkonstanten{6}-phiB);
92         ind1=find(AP==min(AP));
93         ind2=find(AP==min(AP(AP~=min(AP))));
94         X1=Sensorkonstanten{6}(ind1);
95         X2=Sensorkonstanten{6}(ind2);
96         Y1=Sensorkonstanten{8}(ind1);
97         Y2=Sensorkonstanten{8}(ind2);
98
99         I0=Sensorkonstanten{1}*4;
100        Iph=@(phiB) I0*((Y2-Y1)/(X2-X1)*phiB+(X2*Y1-X1*Y2)/(X2-
101            X1));
102        signal=Iph(phiB)*Sensorkonstanten{2};
103    end
104    %Signalbegrenzung
105    if signal > Sensorkonstanten{4}
106        signal = Sensorkonstanten{4};
107    elseif signal < -Sensorkonstanten{4}
108        signal = -Sensorkonstanten{4};
109    end
110
111 end

```



## A.6. posEingabe.m

```
1 %% sollsignal = posEingabe(phiPos)
2 % Erzeugt:
3 %     [Out]:
4 %     sollsignal
5 %
6 %     [In]:
7 %     sollwinkel
8
9
10 function sollsignal = posEingabe(phi)
11
12 global Unit
13 global Sensorkonstanten
14
15 switch Unit
16     case 'grad'
17         phiPos = phi/180*pi;
18     case 'mgrad'
19         phiPos = phi/1000/180*pi;
20     case 'rad'
21         phiPos = phi;
22     case 'mrad'
23         phiPos = phi/1000;
24 end
25
26 % if abs(phiPos) > Sensorkonstanten{3}
27 %     phiPos = Sensorkonstanten{3}*sign(phiPos);
28 % end
29 %
30 sollsignal = (phiPos*1/(Sensorkonstanten{3})) *
31     Sensorkonstanten{4};
32 end
```



## B. Simulink-Files

### B.1. 1



## Literaturverzeichnis



# Abbildungsverzeichnis

1.1. 2 Laserablenkspiegel [?]	2
1.2. Fokusebene [?]	3
1.3. Fokusline	3
3.1. Allgemeiner Aufbau des simulierten Systems	7
3.2. Allgemeiner Aufbau des Sensors	10
3.3. Sensor Funktions	11
4.1. Simulink Grundprogramm	14
5.3. P-Anteil von 30 und I-Anteil von 17	18
5.4. P-Anteil von 30 und I-Anteil von 17	19
5.5. P=22 - D=1 - N=1	19
5.6. P=22 - D=1 - N=1	20
5.7. P=20 - I=15 - D=1 - N=1	20
5.8. P=20 - I=16 - D=1 - N=1	21
5.9. P-Adaption [?]	22
5.12. P-Adaption mit neuen Motorparametern	23
5.14. Stromhöhe während der Regelung	24
5.15. P-Adaption mit Strombegrenzung	24
5.16. Stromhöhe während der Regelung	24
5.17. P-Adaption mit Strombegrenzung	25
5.18. Angepasste P-Adaption mit Strombegrenzung	26
5.19. Angepasste P-Adaption mit Strombegrenzung	26
5.20. Erster linearer Sensor	28
5.21. Zweier linearer Sensor	28
5.22. Linearer Sensor	29





# Tabellenverzeichnis



# Eidesstattliche Erklärung

Wir versichern hiermit gemäß § 35 Abs.7 der Rahmenprüfungsordnung für Fachhochschulen in Bayern, dass wir die vorliegenden schriftliche Arbeit mit dem Titel:

## **Simulationsstudie: Regelung eines Laserablenkspiegels**

selbständig angefertigt, noch nicht anderweitig für Prüfungszwecke vorgelegt und keine anderen als die angegebenen Hilfsmittel benutzt haben.

Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, haben wir in jedem einzelnen Fall unter genauer Angabe der Quelle (einschließlich Internet sowie anderer elektronischer Datensammlungen) deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für angefügte Zeichnungen, bildliche Darstellungen, Skizzen und dergleichen.

.....

Ort, Datum

.....

Unterschrift (Michael Jost)

.....

Ort, Datum

.....

Unterschrift (Sebastian Schleich)