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

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 durch die Größe der Spiegel und der Motoren, sondern auch in der erreichbaren bzw. geforderten Geschwindigkeit und Genauigkeit unterscheiden sich die verschiedenen Ausührungen.
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. Ein große finanzielle Investition stellt ein Robotor ebenfalls dar.
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. 1.3.
In Abb. 1.3 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

Abb. 1.1.: 2 Laserablenkspiegel [?]

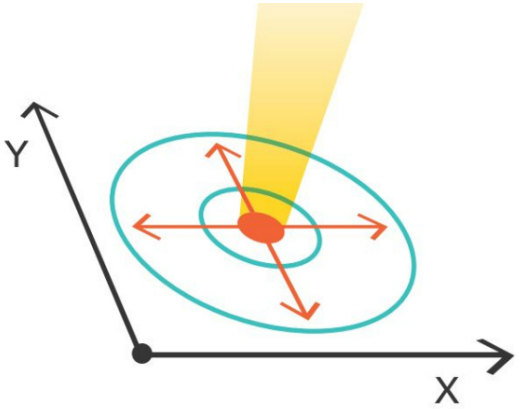


Abb. 1.2.: Fokusebene [?]

entsprechende Anordnung können so Werkstücke größeren Ausmaßes beschriftet werden. Die Bearbeitung von Solarpannels oder Werbebeschriftungen auf Folien können so realisiert werden.

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

Grundsätzlich sollen verschiedene Winkel eingestellt werden können, wobei die größte Winkeländerung 20 ° betragen soll. Die größte Winkeländerung soll mit einer Einstellzeit

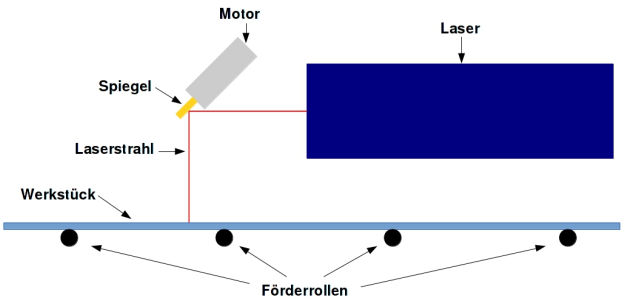


Abb. 1.3.: Fokusline

von 1 ms und mit einer Regelgenauigkeit von 1e-3 ° erreicht werden. Die Simulation wird mit einem fest vorgegebenen Winkel von 20 ° durchgeführt und eine entsprechende Regelung zur Einhaltung der Vorgaben soll entwickelt 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
- Entwicheln und testen verschiedener Sensoren

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. Aufrund dieser Abweichung lässt sich die Winkelposition nicht mehr direkt ablesen, sondern muss 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 vorgstellt.

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 gewählte 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 $1e-3\,^\circ \approx 17\,\mu$rad erreicht und gehalten werden.

In dieser Simulationsstudie wird vorausgesetzt, dass der Abstand des Lasers zum Werkstück unerheblich für die Regelung ist. Zudem wird der Fokus des Laserstrahls über den zu regelnden Winkelbereich als konstant angenommen. Der aufeinander abgestimmte Vorschub des WerkstÃ$\frac{1}{4}$cks 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 ebenso vernachlässigt.

- 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. »»»> 3520102030dedce8edb795c158f30b15688200b4
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 Sensoren
- Es müssen verschiedene Parameter wie, Trägheitsmomente von Spiegel und Welle, Drehmomente, induzierte Spannungen und z.B. Lichtintensitäten betrachtet 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 betrachtet

Es wird mit einem vorgegebenen Gleichstrommotor begonnen, Werte für die Regelparameter zu evaluieren, mit denen sich erste Ergebnisse zeigen. Mit diesen Regelparametern sollen dann die Regelergebnisse verbessert werden. 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, die verbleibende Regeldifferenz zu minimieren. ======= 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 dargestellen Systems zugrunde gelegt.

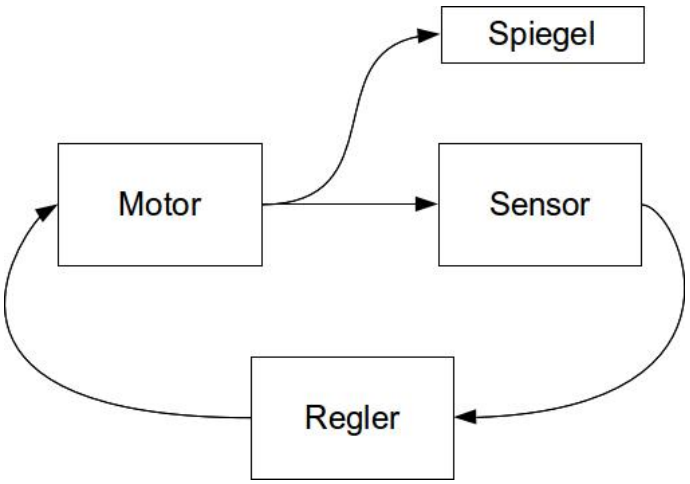


Abb. 3.1.: Allgemeiner Aufbau des simulierten Systems

3.2. Der Spiegel

Lineares Modell für die Berechnungen:

$$\Delta\phi = 20\,^\circ = 0,349\,\mathrm{rad} \tag{3.1}$$

$$\Delta t = 1\,\mathrm{ms} = 0,001\,\mathrm{s} \tag{3.2}$$

$$\omega = \frac{\Delta\phi}{\Delta t} = \frac{0,349\,\mathrm{rad}}{0,001\,\mathrm{s}} = 349\,\mathrm{rad/s} \tag{3.3}$$

Es ergibt sich eine Druchschnittswinkelgeschwindigkeit 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 $\omega = 0\,\mathrm{rad/s}$ und der doppelten Durchschnittsgeschwindigkeit $\omega = 698\,\mathrm{rad/s}$ bei der Hälfte der Strecke und bei der Endposition wieder $\omega = 0\,\mathrm{rad/s}$ der zu fahrenden Strecke angenommen. Daraus folgt eine Beschleunigung von:

$$\Delta\omega = 698\,\mathrm{rad/s} \tag{3.4}$$

$$\Delta t = 0,5\,\mathrm{ms} = 0,0005\,\mathrm{s} \tag{3.5}$$

$$\alpha = \frac{\Delta\omega}{\Delta t} = \frac{698\,\mathrm{rad/s}}{0,0005\,\mathrm{s}} = 1,396 * 10^6\,\mathrm{rad/s^2} \tag{3.6}$$

Der Spiegel erfährt zu Beginn der Regelung eine Beschleunigung von $\alpha = 1,396 * 10^6\,\mathrm{rad/s^2}$ 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 –> Radius: R = 6 mm
- Höhe: h = 2 mm
- Gewicht: m = 10 g

Das Trägheitsmoment des Spiegels beträgt demnach:

$$J = \frac{1}{4} * m * R^2 + \frac{1}{12} * m * h^2 \tag{3.7}$$

$$J = \frac{1}{4} * 10 * 10^{-3}\,\mathrm{kg} * (6 * 10^{-3}\,\mathrm{m})^2 + \frac{1}{12} * 10 * 10^{-3}\,\mathrm{kg} * (2 * 10^{-3}\,\mathrm{m})^2 \tag{3.8}$$

$$J = 93,3 * 10^{-9}\,\mathrm{kgm^2} \tag{3.9}$$

Aus den oben berechneten Daten ergibt sich ein Lastmoment von:

$$M_L = J * \alpha \tag{3.10}$$

$$M_L = 93,3 * 10^{-9}\,\mathrm{kgm}^2 * 1,396 * 10^{6}\,\mathrm{rad/s}^2 \tag{3.11}$$

$$M_L = 130,25 * 10^{-3}\,\mathrm{Nm} \tag{3.12}$$

Theoretische Maximale Leistung eines Gleichstrommotors:

$$P = M_L * \omega \tag{3.13}$$

$$P = 130,25 * 10^{-3}\,\mathrm{Nm} * 698\,\mathrm{rad/s} \tag{3.14}$$

$$P = 91\,\mathrm{W} \tag{3.15}$$

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.

3.3.1. Der elektrische Teil

Der elektrische Teil eines Gleichstrommotors besteht aus einer Spule L_A, ihrem Innenwiderstand R_A, der induzierten Spannung e_A und der Eingangsspannung für den Motor U_e, siehe Abb. 3.2. Es lässt sich folgende Spannungsmasche aufstellen:

$$U_e = u_L + u_R - e_A \tag{3.16}$$

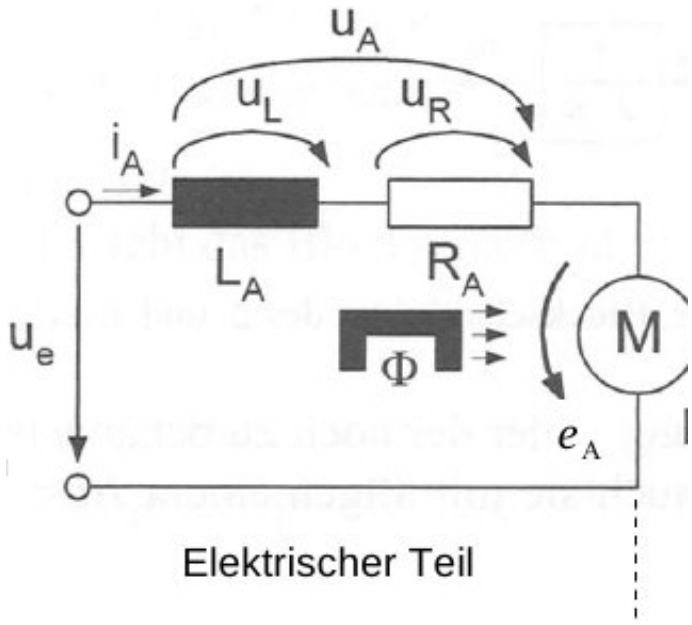


Abb. 3.2.: Elektrischer Teil des Motors

Die Spannung e_A wirkt der Eingangsspannung entgegen und hat ein negatives Vorzeichen. Die Teilspannungen u_L und u_R lassen sich folgendermaßen umschreiben:

$$u_L = L_A * si_A \tag{3.17}$$

$$u_R = R_A * i_A s = \frac{d}{dt} \tag{3.18}$$

Dieses lässt sich in einer Formel ausdrücken:

$$U_e = L_A * si_A + R_A * i_A - e_A \tag{3.19}$$

Für die spätere Integration in Simulink wird die Gleichung 3.19 umgeschrieben:

$$si_A = \frac{1}{L_A}(e_A - R_A i_A + u_e) \tag{3.20}$$

Wobei

$$e_a = K_M * \Phi\omega \tag{3.21}$$

K_M und Φ Motorkonstanten sind.

3.3.2. Der mechanische Teil

Der mechanische Teil eines Gleichstrommotors besteht aus dem gesamten Drehmoment M_B, dem Motormoment M_M, dem Lastmoment M_L des Spiegels. Sowie der Moment-Winkel-Beziehung (Reibung) (siehe Abb. **??**):

$$M_R = r * \omega, \text{ mit } \omega = \frac{d}{dt}\varphi \tag{3.22}$$

Es lässt sich folgende Moment-Bilanzgleichung aufstellen:

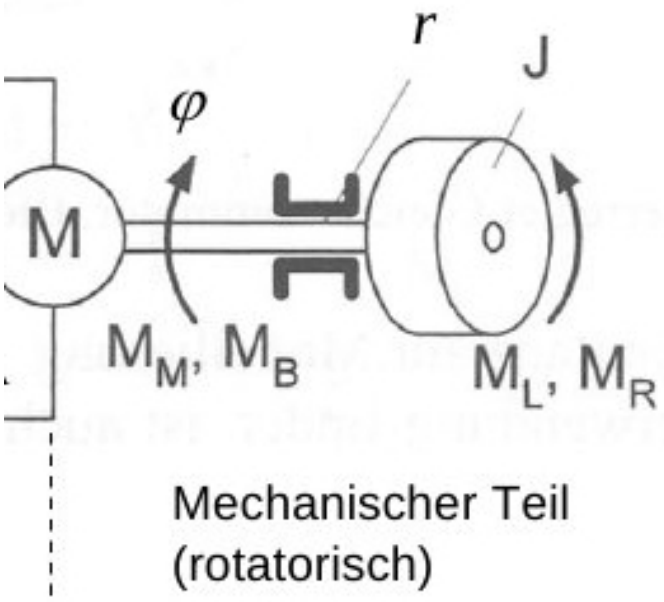


Abb. 3.3.: Mechanischer Teil des Motors

$$M_B = M_M - M_R - M_L \tag{3.23}$$

$$J * \frac{d}{dt}\omega = M_M - r * \omega - M_L \tag{3.24}$$

Für die spätere Integration in Simulink wird die Gleichung 3.24 umgeschrieben:

$$s\omega = \frac{1}{J}(M_M - r * \omega - M_L) \tag{3.25}$$

Wobei

$$M_M = K_M * \Phi * i_A \tag{3.26}$$

K_M und Φ Motorkonstanten sind.

3.3.3. Der ganze Motor

Werden die beiden Einzelaspekte des Motors gleichzeitig betrachtet, so ergbit sich folgender Zusammenhang wie er in Abb. 3.4 dargestellt ist.

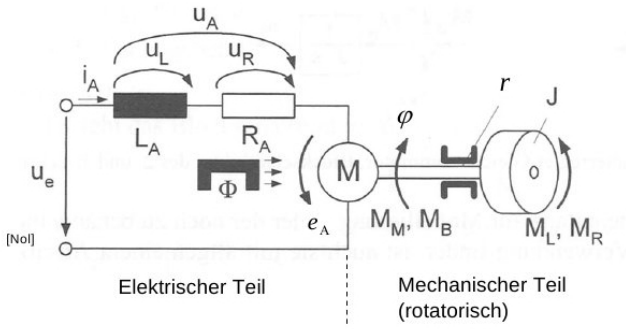


Abb. 3.4.: Aufbau des Motors

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. **??** 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 ϕ_e wird in den Halbraum $\omega = 2\pi$ abgestrahlt. Das Spektrum der LED wird in dieser Simulationsstudie nicht berücksichtigt.
Die Blende
Die Blende wir im folgenden als masselos und völlig lichtundurchlässig angenommen.

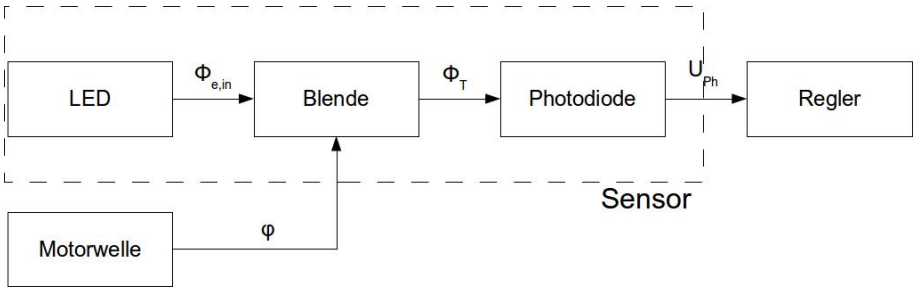


Abb. 3.5.: Allgemeines Funktionsdiagramm des Sensors

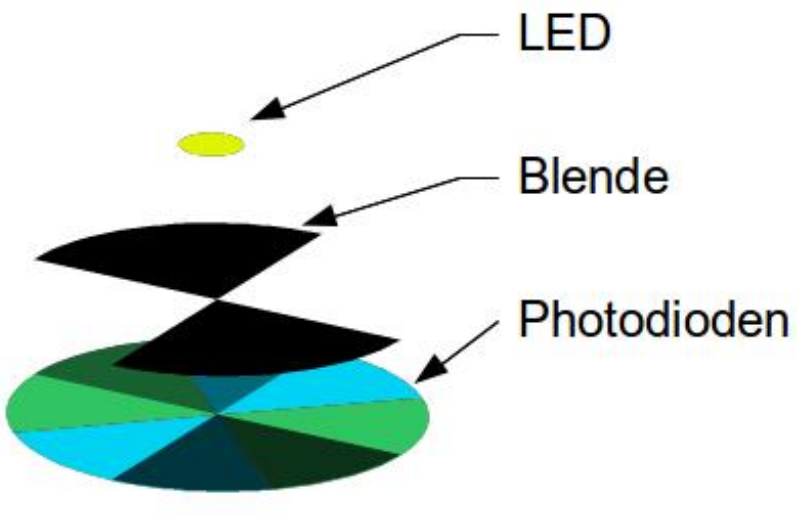
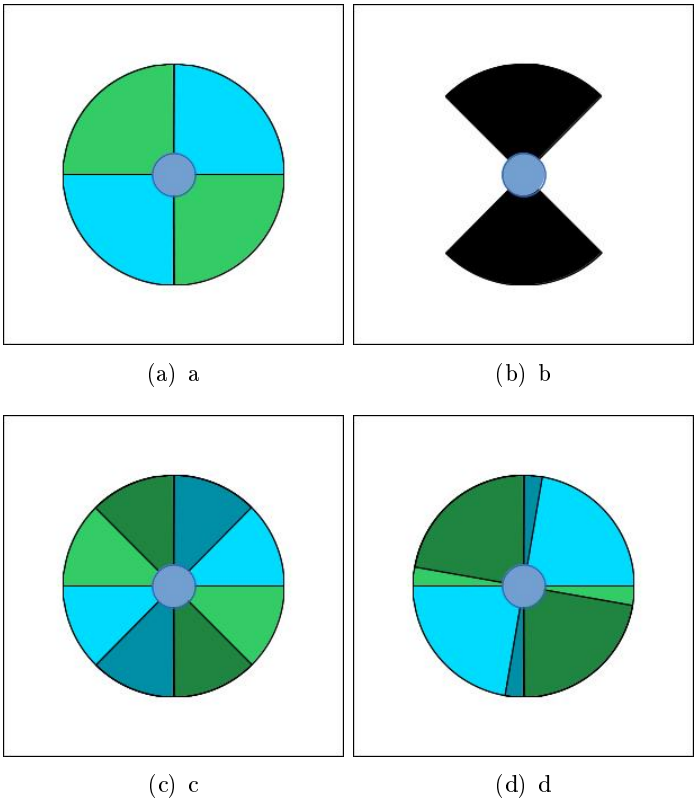


Abb. 3.6.: Allgemeiner Aufbau des Sensors

3.4.2. Das lineare Sensormodell

3.4.3. Das nicht lineare Sensormodell



(a) a (b) b

(c) c (d) d

Abb. 3.7.: Sensor Funktions

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 3.20 und 3.25 aus den Kap. 3.3.1 und 3.3.2 hergenommen. Durch ein umstellen der beiden Formeln, so dass nur noch erste Ableitungen in beiden Formeln vorkommen, lasen sie sich kombinieren und in Simulink einbinden, da so ein Gleichungssytem nur mit ersten Ableitungen entstanden ist. Um einen besseren Überblick zu bekommen, werden die Formeln hier noch einmal aufgeführt.

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

$$e_a = K_M * \Phi\omega \tag{4.2}$$

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

$$M_M = K_M * \Phi * i_A \tag{4.4}$$

K_M und Φ sind Motorkonstanten.
Mit der Annahme das

$$x_1 := \omega \tag{4.5}$$

$$x_2 := i_A \tag{4.6}$$

ist, lässt sich folgendes Gleichungssytem aufstellen:

$$\dot{x_1} = \frac{1}{J}(K_M \Phi x_S ensor_F unktion2 - r * x_1 - M_L) \tag{4.7}$$

$$\dot{x_2} = \frac{1}{L_A}(K_M * \Phi x_1 - R_A x_2 + u_e) \tag{4.8}$$

Dieses Gleichungssytem lässt sich jetzt durch die grafischen Elemente in Simulink sehr einfach modellieren.
Wie zu Begin des Kap. 3.3 erwähnt, war es nicht möglich an verschiedene Werte der Motorkonstanten K_M und ϕ zu gelangen. Aus diesem Grund wird auf die begleitenden Unterlagen der Vorlesung Systemtechniken von Prof. Froriep zurück gegriffen.
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.

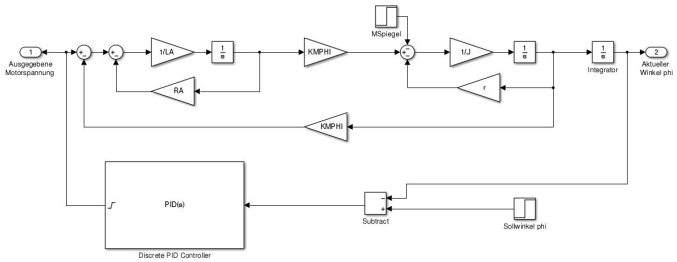


Abb. 4.1.: Simulink Grundprogramm

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, keien Motordaten gefunden wurden, wird auf das Matlabfile von Prof. Froriep aus der Vorlesung SSystemtechnikenßurück gegriffen. In diesem Matlabfile stehen:

- Die Motorkenndaten

- Die berechnete Trägheit des Spiegels
- Das berechnete Drehmoment
- Die Grenzen für die Plots
- Die Anweisungen für die Plots
- Die Anweisungen für den Integrationsalgorithmus.

Dieses File ist eine sehr gute Grundlage für die Simuation, welches während der Simulation entsprechend angepasst werden kann.
Als Größen zur Ausgabe in einem Diagramm, interessieren vor allem die Eingangsspannung u_e, der aktuelle Winkel ϕ, 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 ϕ, der direkt von dem Motor abgegriffen wird, sowie der einzustellende Sollwinkel dargestellt. Der dritte Plot enthält auch wieder den aktuellen Winkel ϕ, 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 und dem Verhalten 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. 3.4in 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 Simuation 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 ° angefahren werden. Dieser Winkel soll innerhalb von einer Millisekunde erreicht werden.
Es wird eine Spannungsbegrenzung von +/-24 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. PID-Regelung

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 erweiteret. Sollten immernoch keine Zufriedenstellenden Ergebnisse vorliegen, so wird mit einer PID-Regelung versucht, die Vorgaben zu erreichen.

5.1.1.1. P-Regelung

In Abb. 5.1 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. 5.2 dargestellt ist. In Abb. 5.1 und 5.2 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 `msSpiegel_PID.m` und das Simulink-File `sSpiegel.slx` verwendet.

5.1.1.2. PI-Regelung

In Abb. 5.3 ist das Ergebnis der PI-Regelung dargestellt. Es ist zu erkennen, dass nach ca. 13 ms es keine Veränderung des eingesetllten Winkels gibt. Eine Eröhung des P- oder I-

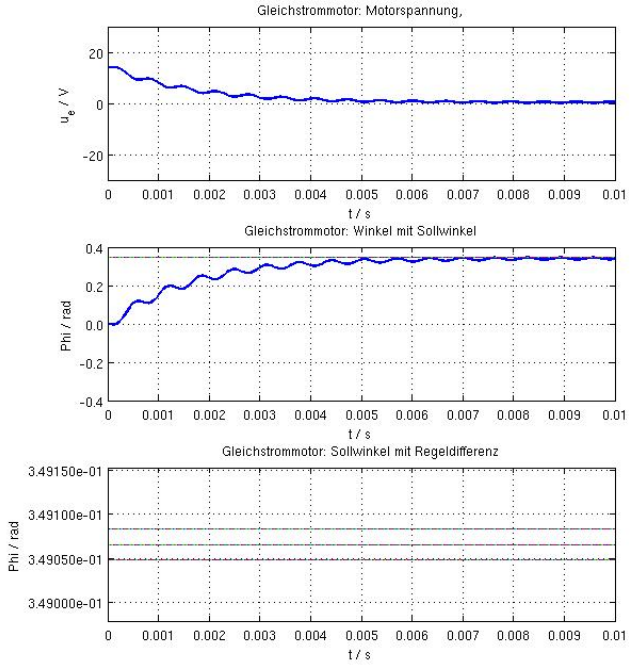


Abb. 5.1.: P-Anteil von 40

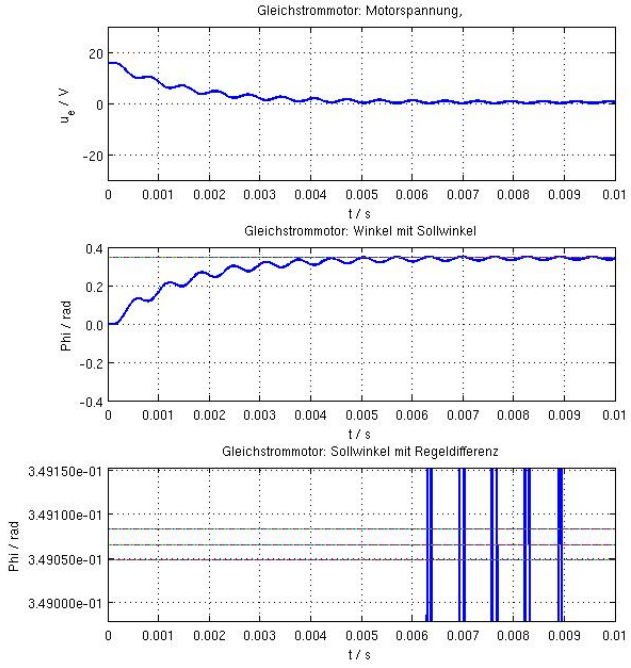


Abb. 5.2.: P-Anteil von 45

Anteils ergibt ein Überschwingen, wie es in Abb. 5.4 dargestellt ist. In Abb. 5.3 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.

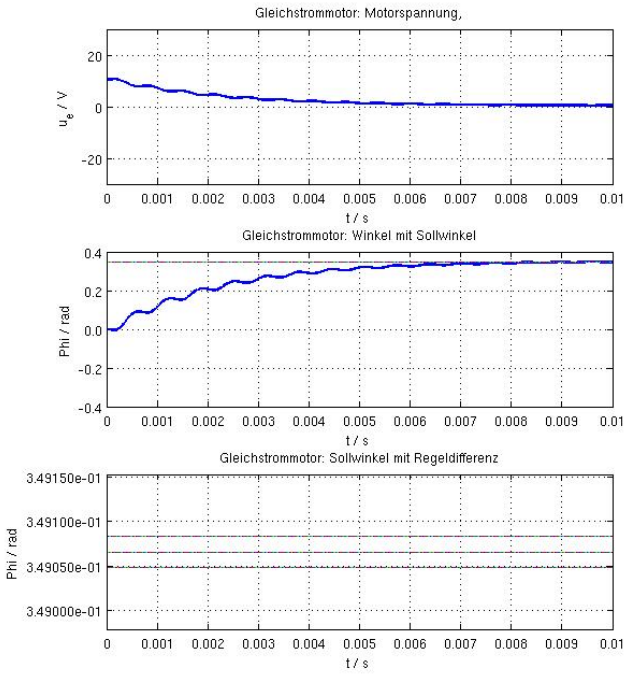


Abb. 5.3.: P-Anteil von 30 und I-Anteil von 17

5.1.1.3. PD-Regelung

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. PID-Regelung

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

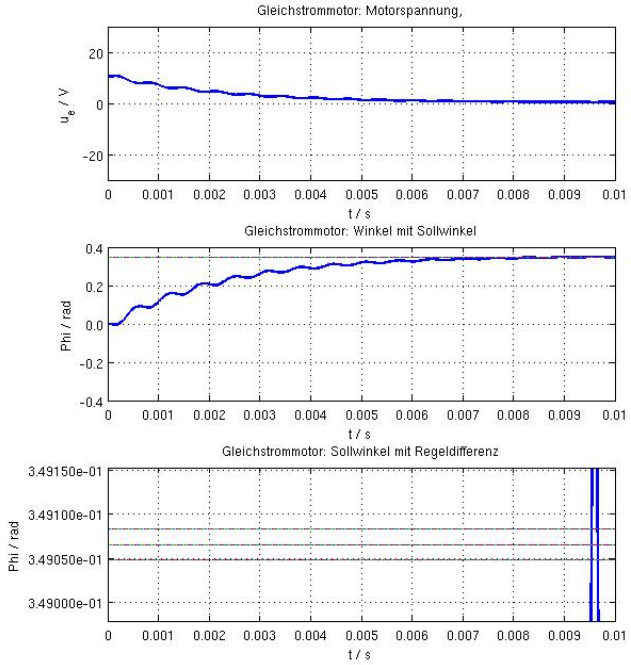


Abb. 5.4.: P-Anteil von 30 und I-Anteil von 18

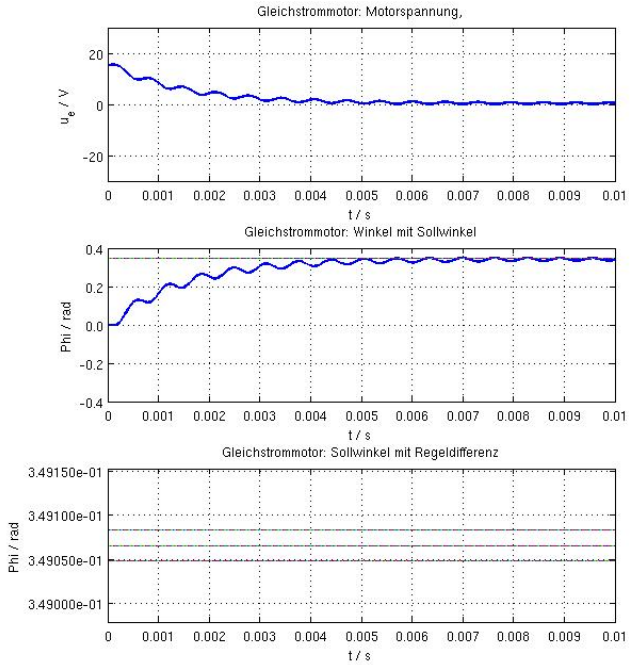


Abb. 5.5.: P=22 - D=1 - N=1

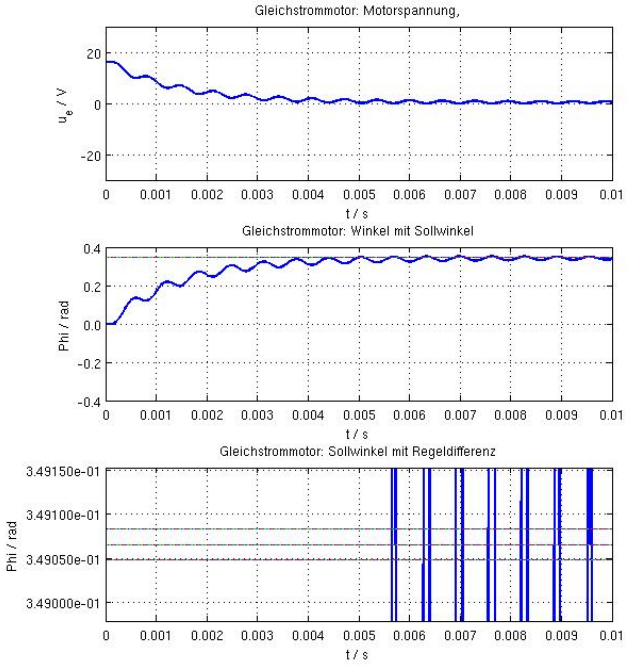


Abb. 5.6.: P=22 - D=1 - N=1

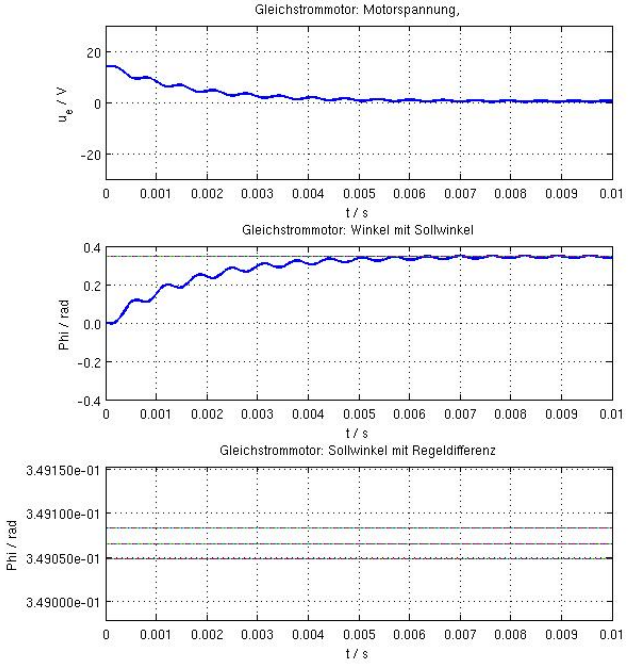


Abb. 5.7.: P=20 - I=15 - D=1 - N=1

5.1.2. P-Adaption

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

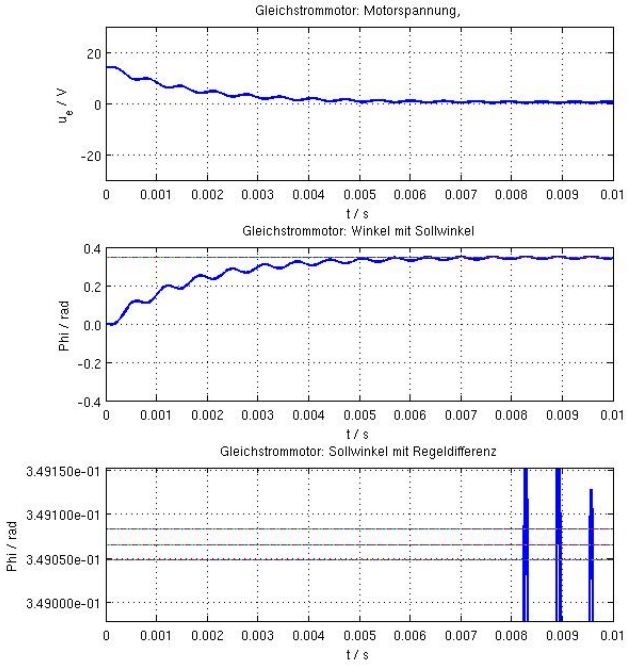


Abb. 5.8.: P=20 - I=16 - D=1 - N=1

verbessert werden. Nach dem die verschiedenen Regler die Vorgaben noch nicht erfüllen konnten, wird nun die P-Adapion eingesetzt. Bei der P-Adaption wird folgende Formel vor den P-Verstärker geschaltet:

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

Dabei muss der Regelkreis folgendermaßen erweitert werden:

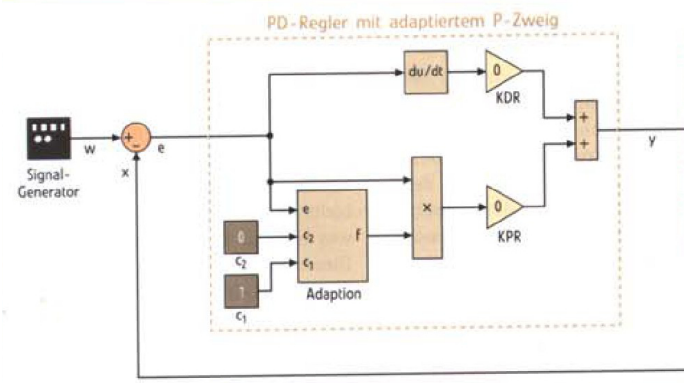


Abb. 5.9.: P-Adaption [?]

Eine Integration in das bestehende Simulinkprogramm ist in Abb. ?? dargestellt.

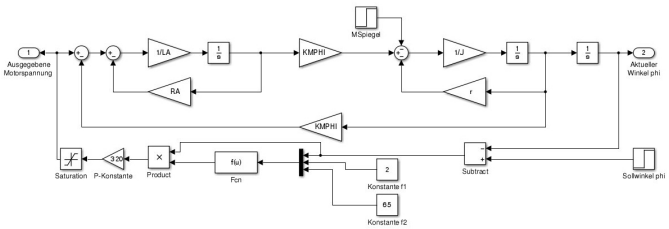


Abb. 5.10.: P-Adaption in Simulink

5.1.2.1. Regelung mit P-Adaption

Nun kann mit drei verschiedenenn Parametern verucht 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 Erebniss verbessert hat. Eine Anpassung des P-Anteils und danach eine erneute Anpassung der f-Parameter gehört ebenso zur Erreichung einer passenden Regelung. Für die folgenden Simulationen wird das Matlab-File `msSpiegel_Pad.m` und das Simulink-File `sSpiegelPad.slx` eingesetzt. In Abb. 5.11

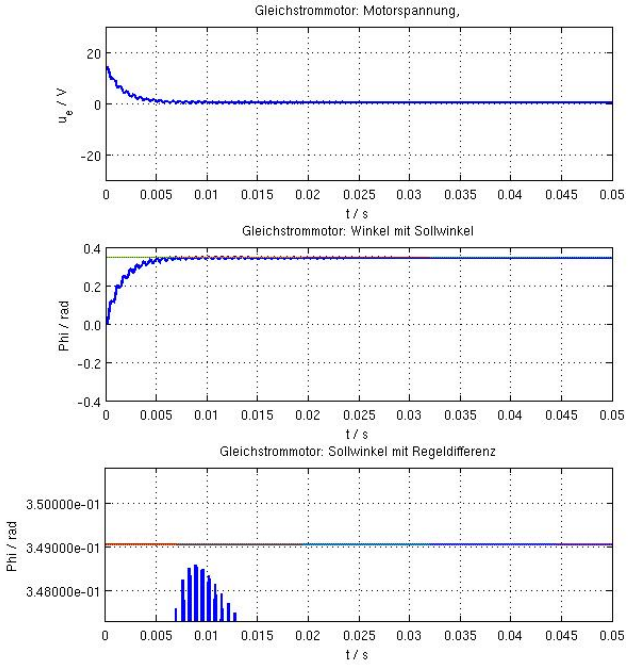


Abb. 5.11.: P-Adaption mit Parametern

ist zu erkennen, dass der Sollwinkel nicht erreicht wird. Wird jedoch der P-Anteil erhöht und mit den beiden f-Parametern weitere Einstellungen probiert, so wird der Sollwinkel nie erreicht. Es kommt zwar zu einem Schwingen um den Sollwinkel, aber dieser kann nicht stabil erreicht werden, siehe Abb. 5.12.

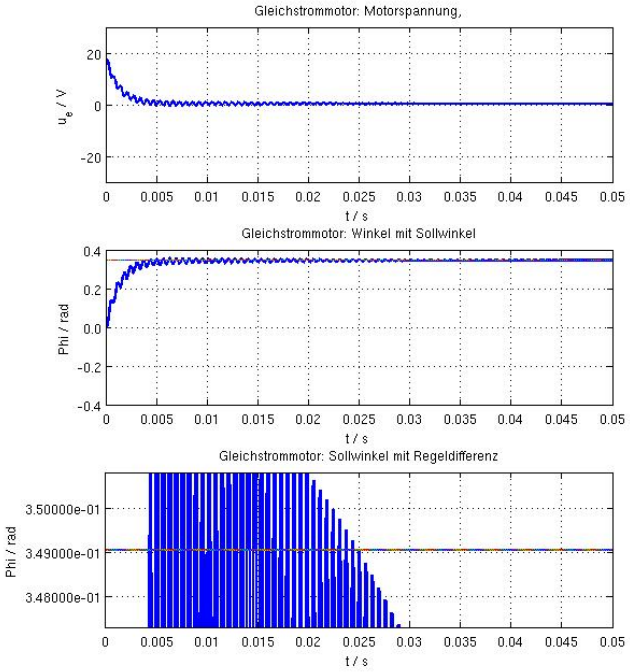


Abb. 5.12.: P-Adaption mit Parametern

5.1.2.2. P-Adaption mit Galvo-Werten

Durch die Verwendung der P-Adaption konnte die Einregelzeit nicht verbessert werden. Es zeigt sich, dass mit dem vorhandenen Gleichstrommoter keine der Vorgaben eingehalten werden können. Um heraus zu finden, welche Daten der Motor aufweisen muss, um mit einer PID- oder P-Adaption geregelt werden zu könnnen, werden jetzt zusätzlich zu den f_1 und f_2 Parametern auch die Motorparameter geändert. Es 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.m` und das Simulink-File `sSpiegelPad.slx` hergenommen.
In Abb. 5.13 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

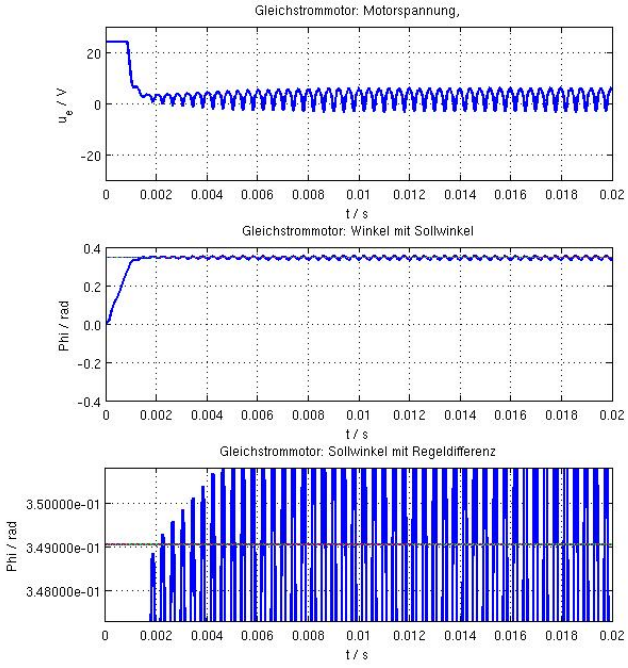


Abb. 5.13.: P-Adaption mit neuen Motorparametern

5.1.2.3. P-Adaption mit neuen Werten

Nun werden die Motor- und Spiegelwerte solange verändert, bis sich das gewünschte Ergebniss einstellt. Sollte die Regelung erfolgreich sein, kann mit den veränderten Werten evtl. ein Motor und Spiegel hergestellt werden, der den Anforderungen entspricht.
Für die folgenden Simulationen wird das Matlab-File `msSpiegel_Pad_Neue_Werte.m` und das Simulink-File `sSpiegelPad.slx` verwendet. Wie in Abb. 5.14 zu erkennen, ist die Regelung in den geforderten Bereichen erfolgreich. Der geforderte Winkel von [20]° ist unter 1 ms in seinen Regeldifferenzen erreicht. Dieser Regelung liegen folgende Werte zu Grunde:

- Innenwiderstand der Spule: $0.1\,\Omega$
- Induktivität der Spule: $3\,\mu$H
- Motorkonstante KMPHI: $35e-3\,$Vs
- Reibungskoeffizient: $6e-5\,$Nms
- Trägheitsmoment des Spiegels: $93.3e-9\,\mathrm{kgm}^2$
- Drehmoment auf den Spiegel: $130.25e-6\,$Nm

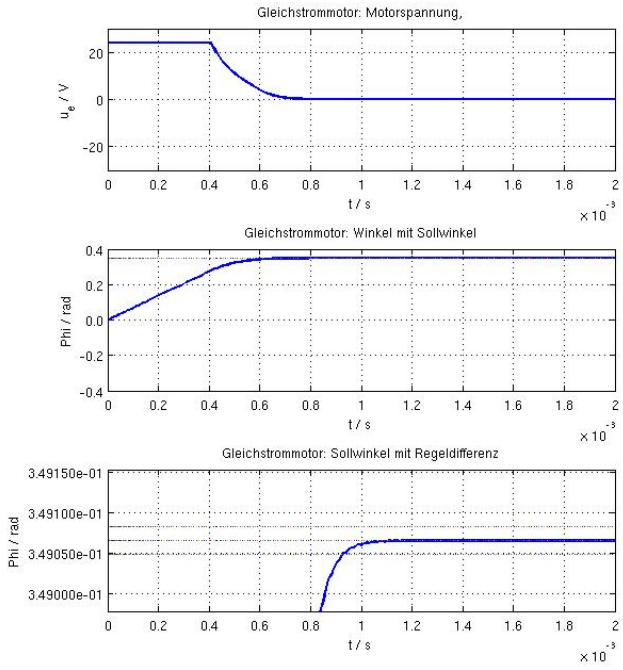


Abb. 5.14.: P-Adaption mit neuen Parametern

- P-Anteil: 320
- f_1: 2
- f_2: 160

5.1.2.4. Strombegrenzung

Ein Blick auf den Strom liefert allerdings Ergebnisse, die weiterer Überarbeitung der Regelung bedürfen. In Abb. 5.15 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 Strombegrenung von 10 A eingebaut und erneut versucht, die Regelung entsprechend anzupassen. In Abb. 5.16 ist die P-Adaption mit einer Strombegrenzung 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. 5.17. Für die folgenden Simulationen wird das Matlab-File `msSpiegel_Pad_Neue_Werte`$strom.mund das Simulink-File$`sSpiegelPadStrom.slx`$hergenommen.$ `Eine Regelung mit den aktuellen Werten zeigt das in Abb. 5.18 zu erkennende Ergebnis. Durch weiteres Anpassen der unterschiedlichen Parameter, konnten die Sollwerte fast erreicht werden. Abb. 5.19 zeigt schon ein sehr gutes Ergebnis.`

- `Neues Trägheitsmoment des Spiegels:` $93.3e-11\,\mathrm{kg}*\mathrm{m}^2$
- `Neues Drehmoment auf den Spiegel:` $30,25e-6\,\mathrm{Nm}$

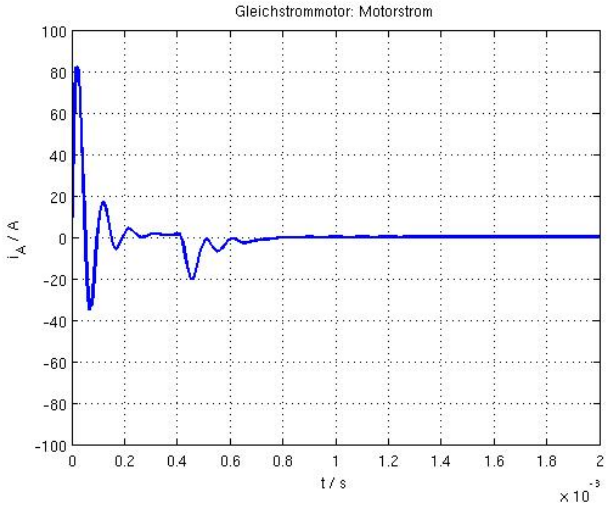


Abb. 5.15.: Stromhöhe während der Regelung

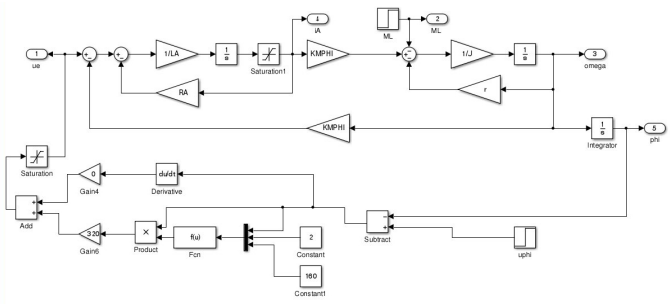


Abb. 5.16.: P-Adaption mit Strombegrenzung

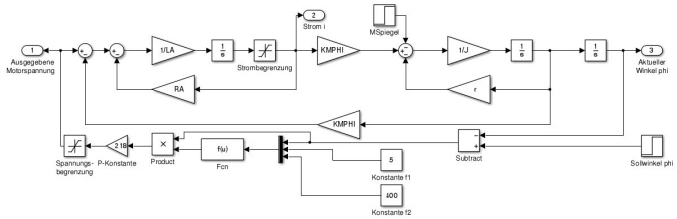


Abb. 5.17.: Stromhöhe während der Regelung

- P-Anteil: 218
- f_1: 5
- f_2: 400

5.1.2.5. Fertige P-Adaption

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

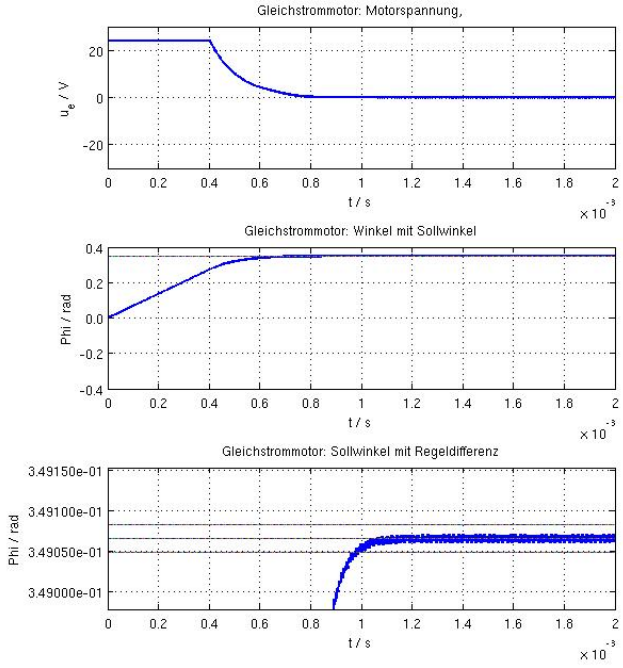


Abb. 5.18.: P-Adaption mit Strombegrenzung

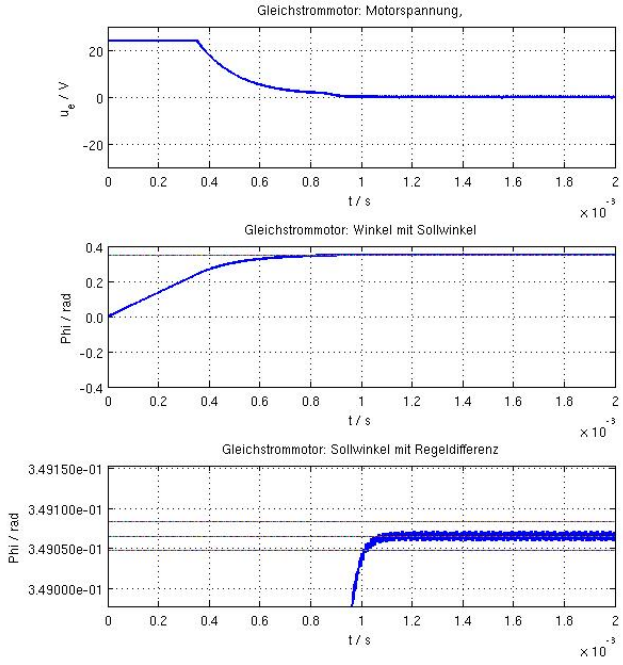


Abb. 5.19.: Angepasste P-Adaption mit Strombegrenzung

- `Trägheitsmoment des Spiegels:` $93.3e-11\,\mathrm{kg}*\mathrm{m}^2$
- `Drehmoment auf den Spiegel:` $30,25e-6\,\mathrm{Nm}$

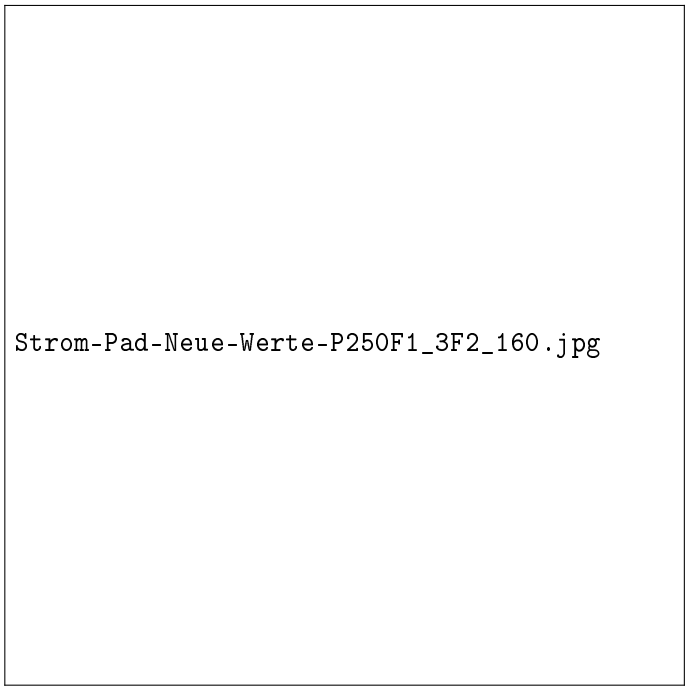


Abb. 5.20.: Angepasste P-Adaption mit Strombegrenzung

- P-Anteil: 250
- f_1: 3
- f_2: 150

In Abb. 5.20 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.

5.1.3. Regelung mit Sensor

Nach dem die Regelung für einen perfekt linear arbeitenden Sensor, der durch ein direktes abgreifen des aktuellen Winkels realisiert wurde, funktioniert, wird der Sensor aus Kap 3.4 in die Simulation mit eingebaut. Dafür wird das vorhandene Simulink-File sSpiegelPadStrom.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. 5.21 ist der gesamte Simulationsaufbau dargestellt, welcher als sSpiegelPadStromSensor.slx gespeichert ist. Zur Ansteuerung wird das Matlab-File msSpiegel_Pad_Neue_Werte_Strom.m modifiziert und als msSpiegelundSens gespeichert. In diesem Matlab-File ist es möglich, verschiedene Kenndaten für den Sensor einzugeben. Es werden folgende Daten hergenommen:

- Innenradius = 5 mm

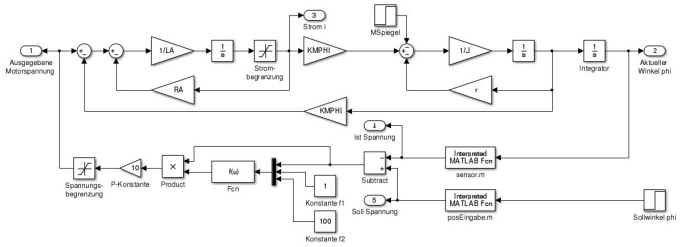


Abb. 5.21.: Simulinaufbau mit Sensor

- Aussenradius = 10 mm
- Lastwiederstand = 6000 Ohm
- Messbereich = 20 °
- LEDLeistung = 1 W
- Umgebungstemperatur = 300 K
- nonlinear = 0.0001 (falls ein nichtlinearer Sensor simuliert werden soll)

Es werden drei verschieden 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. 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. Linearer Sensor 1

Abb. 5.22 zeigt das Regelergebnis des ersten linearen Sensors. Es ist gut zu erkennen, dass die Regelung sogar schneller Erfolgt als vorher.

5.1.3.2. Linearer Sensor 2

Abb. 5.23 zeigt das Regelergebnis des zweiten linearen Sensors. Es ist gut zu erkennen, dass eine Regeldifferenz bleibt. Diese lässt sich auch nicht durch verändern der Regelparameter reduzieren.

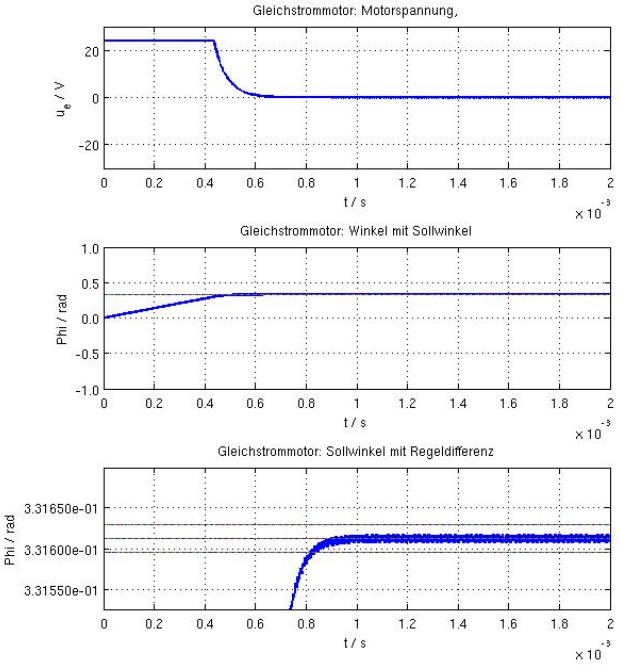


Abb. 5.22.: Erster linearer Sensor

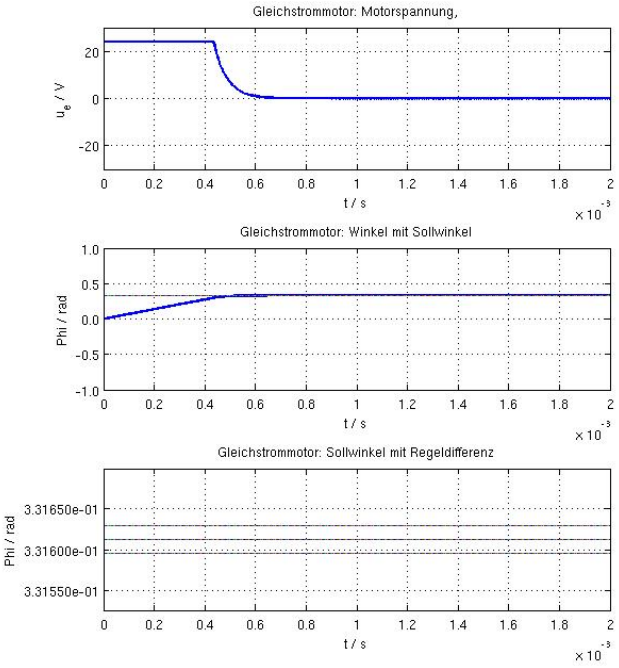


Abb. 5.23.: Zweier linearer Sensor

5.1.3.3. Nichtlinearer Sensor

Abb. 5.24 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.

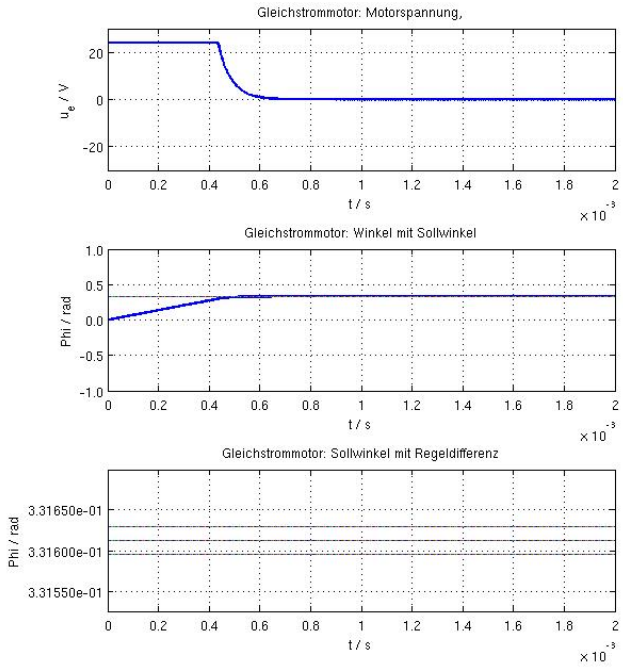


Abb. 5.24.: Linearer Sensor

6. Disskussion

Die Simulationsstudie wurde mit einem vorgegebenen Gleichstrommotor Abb. 3.4 aus Kap. 3.3.3 begonnen. Für diesen Motor wurde ein PID-Regler in Simulink integriert, Abb. 4.1 aus Kap. 4.1, und unterschiedliche Regelparameter getestet. Der Sensor wurde noch nicht integriert, um erst einmal eine Regelung für einen Gleichstrommotor zu bekommen, mit der es überhaupt möglich ist, einen Winkel einzustellen.
Bei der reinen P-Regelung, konnte weder die Einregelzeit, noch die Genauigkeit erreicht werden. Wie in Abb. 5.1 und 5.2 aus Kap.5.1.1.1 zu erkennen ist, beträgt die Zeit, bis der Winkel in der Nähe des Sollwertes ist, ca. 7 ms. Jedoch ist ein große Schwankung des Winkels zu erkennen. Bei kleinerem P-Anteil von P = 40 ist die Abweichunng noch so groß, dass der Winkel in der unteren Grafik nicht erreicht wird. Bei einem etwas größerem P-Anteil P = 45 ist zu erkennen, dass der Sollwinkel zwar überfahren wird, aber nicht stabil bleibt.
Die Erweiterung des reinen P-Reglers um einen I-Anteil ist in den Abb. 5.3 und 5.4 aus Kap.5.1.1.2 zu erkennen. Durch die Erweiterung mit einem I-Anteil, hat sich die Einregelzeit auf ca. 9 ms verlängert. Auch wurde der Sollwinkel nicht erreicht. Eine Erhöhung des P- oder I-Anteils führte dann wieder zu Schwingungen um den Sollwinkel, wie es in Abb. 5.4 exemplarsich für einen leicht erhöhten I-Anteil dargestellt ist.
Die Erweiterung des reinen P-Reglers um einen D-Anteil ist in den Abb. 5.5 und 5.6 aus Kap.5.1.1.3 zu erkennen. Hier ergibt sich das gleiche Problem wie bei der P- und PI-Regelung. Bis zu einer Grenze der P- und D- Anteile, ist die Regelung zu weit von dem Sollwinkel entfernt. Wird nur ein Anteil leicht erhöht, kommt es zu einem Überschwingen um den Sollwinkel. Jedoch hat sich die Zeit, bis der aktuelle Winkel um den Sollwinkel schwingt, auf ca. 6 [ms verkürzt.
Es wurde nun mit einer PID-Regelung versucht, die Vorgaben zu erreichen. In den Abb. 5.7 und **??** aus Kap.5.1.1.4 sind die Ergenisse der PID-Regelung dargestellt. Es zeigt sich das von der P-, PI- und PD-Regelung bekannte Verhalten, dass bis zu entsprechenden Reglerwerten, der aktuelle Winkel zu weit vom Sollwinkel entfernt ist und bei einer nur kleinen Erhöhung eines der Parameter, ein Überschwingen um den Sollwinkel einsetzt. Dies ist exemplarisch für den I-Anteil in Abb. **??** dargestellt. Die Zeit bis zum ersten durchstreichen des Sollwinkels ist etwas über 8 ms.
Mit dieser noch nicht zufriedenstellenden Regelung, wird der PID-Reglerblock in Simulink entfernt und eine P-Adaption in den Regelkreis eingebaut. Abb. **??** aus Kap. 5.1.2 zeigen diesen Aufbau. Mit der vorgeschalteten Funktion vor den P-Verstärker, soll eine geringere

bleibende Regeldiffernz erreicht werden.
Wie aus den Abb. 5.11 und 5.12 aus Kap.5.1.2.1 zu erkennen ist, wird der Sollwinkel noch nicht erreicht. Der aktuelle Winkel gelangt bei kleineren Parameterwerten nicht in die Nähe des Sollwinkels. Bei größeren Parameterwerten dagegen, ergibt sich wieder ein Überschwingen um den Sollwert. Auch mit dieser P-Adaption konnten keine Parameterwerte gefunden werden, mit der eine Regelung bei diesem Gleichstrommotor, den Vorgaben entsprechend realisiert werden konnte.
Um eine erfolgreiche Regelung zu erreichen, werden die Motorparameter angepasst, während die P-Adaption als Regelung behalten wird. Es wird zuerst die Induktivität und der Innenwiderstand eines Galvos eingesetzt, um so erneut Parameter zu finden, die eine erfolgreiche Regelung ermöglichen.
Mit diesen Parametern konnte zwar ein Überschwingen um den Sollwinkel nicht verhindert werden, jedoch hat sich die Zeit, bis zum ersten durchschreiten des Sollwinkels auf ca. 2 ms verkürzt. Abb. 5.13 aus Kap. 5.1.2.2 zeigt den Vorgang.
Da die Anpassung des Motors gut Ergebnisse zeigte, wird nun mit weiterer Anpassung der verschiedenen Motor- und Spiegelwerte versucht, eine erfolgreiche Regelung aufzubauen. Durch die so erhaltenen Parameter kÃ¶nnte es mÃ¶glich sein, entsprechende Bauteil fertigen zu lassen, um so eine Umsetzung in die Wirklichkeit zu realisieren.
Nach entsprechenden Anpassungen konnte eine erfolgreiche Regelung aufgebaut werden. Abb. 5.14 aus Kap. 5.1.2.3 zeigt, dass die Einregelzeit unter 1 ms liegt und sich der aktuelle Winkel in den Regeldifferenzen befindet.
Ein Blick auf den Strom zeigte, dass weitere Anpassungen notwendig sind. Es wurde eine Strombegrenzung eingebaut und neue Regelparameter probiert. Siehe Abb. 5.15 und 5.17 aus Kap. 5.1.2.4.
Nach einigen Versuchen mit Ändern der verschiedenen Parameter, ist wieder eine erfolgreiche Regelung aufgebaut worden. Der Strom ist auf 10 A begrenzt, die Einregelzeit liegt bei 1 ms und der aktuelle Winkel ist in den Regeldifferenzen. Abb. 5.20 aus Kap. 5.1.2.5 zeigt die erfolgreiche Regelung.
Nach dem es nun mÃ¶glich ist, die Vorgaben zu erfÃ¼llen, werden 3 verschiedenen Sensoren in den Regelkreis eingebracht. Es wird mit einem idealen linearen Sensor begonnen. An den schliesst sich ein linearer Sensor an, der Ã¼ber eine Faltung der Blende mit den Lichtsensoren gebildet wird. Der dritte und letzte Sensor stellt einen nichtlinearen Sensor dar.
Abb. 5.21 zeigt den Simulationsaufbau fÃ¼r die verschiedenen Sensoren. Die weitere Implememtierung der verschiedenen Sensoren erfogt Ã¼ber Parameter in den Matlab-Files.
Als der erste lineare Sensor das aktuelle Positionssignal lieferte, musste die Regelung angepasst werden. Mit den neuen Regelparametern konnte sogar eine noch kÃ¼rzere Einregelzeit erreicht werden. Abb. 5.22 aus Kap. 5.1.3.1 zeigt die erfolgreiche Regelung.
Ohne die Regelparameter zu Ändern, wurde der zweite lineare Sensor getestet. Die Regelung ist zwar genauso schnell, jedoch bleibt eine Regeldifferenz Ã¼brig. In Abb. 5.23 aus Kap. 5.1.3.2 ist der Vorgang dargestellt. Die verbleibende Regeldifferenz wird erstmal auf die

diskrete Faltung zurÃ¼ckgefÃ¼hrt, bei der Rechenfehler auftreten kÃ¶nnen.
Nun wurde der dritte Sensor getestet. Wie zu erwarten, verbleibt eine Regeldifferenz. In Abb. 5.24 aus Kap. 5.1.3.3 ist der Vorgang dargestellt. Das ist ein Bild, wie es in einem wirklichen Aufbau z.B. mit genau messbarere Laserablenkung zu entdecken wÃre. Man mÃ¼sste einen Winkel einstellen, und durch die entsprechende Laserablenkung auf den wirklichen Winkel schliessen. In der Simulation gibt es den Vorteil, dass wir uns den wirklichen Winkel anschauen. Dieser wird dann Ã¼ber den nichtlinearen Sensor in ein Spannungssignal gewandelt und dem Regler zugefÃ¼hrt. Insofern wÃ¼rde eine VerÃnderung der Regelparameter keine genauere Regelung bewirken.

7. Zusammenfassung

Es konnte eine Regelung und ein Sensor simuliert werden, mit denen es möglich ist, die Vorgaben aus Kap. 2 zu erfüllen.
Die Simulationsstudie wurde mit einem vorgegebenen Gleichstrommotor begonnen. Für diesen Motor wurde ein PID-Regler in Simulink integriert und unterschiedliche Regelparameter getestet. Der Sensor wurde noch nicht integriert, um erst einmal eine Regelung für einen Gleichstrommotor zu bekommen, mit der es Überhaupt möglich ist, einen Winkel einzustellen.

A. Matlab-Files

A.1. msSpiegelundSensor.m

```
1  %  msSpiegel_Pad.m      (Matlab/Simulink R2011b)
2  %
3  %  Vorgang:    Regelung eines Gleichstrommotors zur Spiegelverstellung
4  %  Verfahren: Simulink, mithilfe einer P-Adaption, eingebauter Strom-
5  %              begrenzung, angepasster Motorenwerte und eingebauter
6  %              Sensor
7  %
8  %  Unterprogramme:  sSpiegelPadStromNeu.slx
9  %                   sensor.m
10 %                   posEingabe.m
11 %
12 % ########################################################
13 %
14 %  Parameterbeschreibung:
15 %
16 %   RA            Ankerwiderstand
17 %   LA            AnkerinduktivitÃ¼t
18 %
19 %   J             Traegheitsmoment
20 %   r             Reibkonstante
21 %
22 %   MSpiegel      Drehmoment fÃ¼r Spiegel
23 %
24 %   KMPHI         MotorkenngrÃ¶sse
25 %
26 %   phi           SprunghÃ¶he Motorwinkel
27 %
28 %   te            Ende des Integrationsintervalls (ab t=0)
29 %
30 %   xuX, xoX      Untere/obere Grenze der Grafiken
31 %0
32 % ########################################################
33 clear all
34 close all
35
36 % 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 Lastwiederstand =6000;         % in Ohm
49
50 Photodioden = [Innenradius,Aussenradius,Lastwiederstand];
51 Messbereich = 20/180*pi;       % 45Â° 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;                      % phiu=-20Â° in rad 0.4
86   po1=1;                       % phio=+20Â° 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
```

```
% ########################################################

% Plot: Eingangssignal u und Winkel phi
figure(1)
set(gcf,'Units','normal','Position',[.49 .7 .5 .9], ...
    'NumberTitle','on','Name','u und v ');

%  Integrationsalgorithmus:
t0=0;
opts=simset('solver','ode45',...
    'InitialState',[],...
    'Refine',1,...
    'MaxStep',.00001);

[t,x,y]=sim('sSpiegelPadStromSensor',[t0 te],opts);

% Plots
subplot(3,1,1)
plot (t,y(:,1),'linewidth',2)
axis([0 te vu vo])
grid on
hold on
xlabel('t / s')
ylabel('u_e / V')
title('Gleichstrommotor: Motorspannung,')

subplot(3,1,2)
plot(t,y(:,2),t,phi,'linewidth',2,'linewidth',2);
axis([0 te pu1 po1])
grid on
xlabel('t / s')
ylabel('Phi / rad')
YTicks=get(gca,'YTick');
set(gca,'YTickLabel',num2str(YTicks(:),'%.1f'));
title('Gleichstrommotor: Winkel mit Sollwinkel')

subplot(3,1,3)
plot(t,y(:,2),...
     t,phi,...
     t,(phi-1e-3*pi/180),...
     t,(phi+1e-3*pi/180),...
     'linewidth',2,...
     'linewidth',2,...
     'linewidth',2,...
     'linewidth',2);
axis([0 te pu2 po2])
grid on
xlabel('t / s')
ylabel('Phi / rad')
YTicks=get(gca,'YTick');
```

```
140 set(gca,'YTickLabel',num2str(YTicks(:),'%.5e'));
141 title('Gleichstrommotor: Sollwinkel mit Regeldifferenz')
142
143
144 % MÃ¶glichkeit, Strom und Winkelspannunge 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

```
%% Sensorkonstanten = sensorDaten(Photodioden,
%                                 Messbereich,
%                                 LEDLeistung,
%                                 Umgebungstemperatur)
% Erzeugt:
%       [Out]:
%       Array der Sensorkonstanten => Sensorkonstanten
%           1 signal_max in A
%           2 Lastwiederstand der Idealen Photodiode in Ohm
%           3 Azimutwinkel der Photodioden/Maximaler Messbereich in rad
%           4 signal_max in V
%           5 X Koordinaten der liniaren Kennlinie
%           6 X Koordinaten der nicht liniaren Kennlinie
%           7 Y- / Funktions-Werte der liniaren Kennline
%           8 Y- / Funktions-Werte der nicht liniaren Kennline
%
%       [In]:
%       Array der Diodendaten(geometrisch, elektrisch) => Photodioden
%           1   Innererradius r1            in mm
%           2   ÃuÃererradius r2              in mm
%           3   Lastwiederstand             in Ohm
%
%       Maximal messbarer Winkel +- => Messbereich in rad
%
%       Gesammtleistung der LED => LEDLeistung in W
%       Umgebungstemperatur => Umgebungstemperatur in K

function Sensorkonstanten = sensorDaten(Photodioden,...
                                        Messbereich,...
                                        LEDLeistung,...
                                        Umgebungstemperatur,...
                                        nonlinear)
Sensorkonstanten{1}=0.001; % max Strom
Sensorkonstanten{2}=Photodioden(3); % Lastwiderstand
Sensorkonstanten{3}=Messbereich; % max Messbereich
Sensorkonstanten{4}=Sensorkonstanten{1}*Photodioden(3)*4;%max Spannung

%Sensorcharakteristik mittels Faltung von Fensterfunktionen
xmin = -2*pi;
xmax = 2*pi;
x = linspace(xmin,xmax,4000);

c = 1;
e = nonlinear;
b = Messbereich*2;
pb = 0;
ps1 = b/2;
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

```
function rect = frect(x,breit,pos)

while abs(x)/(pi/2) > 1
    x = x - sign(x)*pi;
end

if and(x > (pos-breit/2),x<(pos+breit/2))
    rect = 1;
else
    rect = 0;
end

end
```

A.4. gekern.m

```
function glatt = gekern(x,e,c)
xe = x/e;

if abs(xe) < 1
    glatt = c*exp(-1/(1-xe^2));
else
    glatt = 0;
end

glatt = 1/e * glatt;

end
```

A.5. sensor.m

```
%% signal = sensor( Blendenwinkel, Unit, mode, Sensorkonstanten)
% Erzeugt:
%       [Out]:
%       Normiertes Spannungssignal +-5V => signal in V
%
%       [In]:
%       aus Eingangswinkel => Blendenwinkel in
%       der angegebenen Einheit => Unit
%       Kennmode des Sensors => mode
%       Array aus Sensorkonstanten => Sensorkonstanten
%           1 signal_max in A
%           2 Lastwiederstand der Idealen Photodiode in Ohm
%           3 Azimutwinkel der Photodiodennonlinear
%           4 signal_max in V
%       Unit cases: 'grad','mgrad','rad','mrad'
%       mode cases : 'linear', ...

function signal = sensor(Blendenwinkel)

global Unit
global mode
global Sensorkonstanten

%% Umrechnung der Eingangseinheit
switch Unit
    case 'grad'
        phiB = Blendenwinkel/180*pi;
    case 'mgrad'
        phiB = Blendenwinkel/1000/180*pi;
    case 'rad'
        phiB = Blendenwinkel;
    case 'mrad'
        phiB = Blendenwinkel/1000;
end
%Rauschen aus Temperaturspannung
% kb = 8.6173324*10^-5; % eV/K
% e = 1; % eV
% T = Sensorkonstanten{9};
% RLast = Sensorkonstanten{2};
% zufall = 0.5-rand();
% UT = kb*T/e/RLast % in V
% Rauschen = UT+UT*zufall

Rauschen = rand()*Sensorkonstanten{1}/10;

switch mode
    case 'einzelGruppe'

```

```
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);% Gesammt 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); % Gesammt 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          I0=Sensorkonstanten{1}*4;
87          Iph=@(phiB)I0*((Y2-Y1)/(X2-X1)*phiB+(X2*Y1-X1*Y2)/(X2-X1));
88          signal=Iph(phiB)*Sensorkonstanten{2};
89
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-X1));
101         signal=Iph(phiB)*Sensorkonstanten{2};
102 end
103
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

```
%% sollsignal = posEingabe(phiPos)
% Erzeugt:
%       [Out]:
%       sollsignal
%
%       [In]:
%       sollwinkel


function sollsignal = posEingabe(phi)

global Unit
global Sensorkonstanten

switch Unit
    case 'grad'
        phiPos = phi/180*pi;
    case 'mgrad'
        phiPos = phi/1000/180*pi;
    case 'rad'
        phiPos = phi;
    case 'mrad'
        phiPos = phi/1000;
end

% if abs(phiPos) > Sensorkonstanten{3}
%     phiPos = Sensorkonstanten{3}*sign(phiPos);
% end
%
sollsignal = (phiPos*1/(Sensorkonstanten{3})) * Sensorkonstanten{4};

end
```

B. Simulink-Files

B.1. 1

Abbildungsverzeichnis

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)