

Konzeption und Regelung eines Servomotorstands

Maximilian Ebersbach, Janguang Gao, Haoming Wang, Zhenhui Zhou

Zusammenfassung—Diese Arbeit befasst sich mit der Konstruktion und Regelung eines Servomotorstands. Aufbauend auf den theoretischen Inhalten der Veranstaltung Praktikum Regelungstechnik II und des zugehörigen Skripts[QUELLE], entwickelten wir einen von Grund auf neuen Versuchsstand, der die in Matlab simulierten Ergebnisse auf ein reelles Motormodell überträgt. Hierbei übernahmen wir sowohl Einkauf und Konstruktion, wie auch die Implementierung des Versuchs. In den Versuchen wird das Positionierverhalten eines Servomotors mit drei unterschiedlichen Reglern simuliert, diese drei Regler sind ein linearer Lageregler, ein strukturvariabler Regler und ein nichtlinearer Regler. Diese drei Regler wurden von uns auf einem Raspberry Pi implementiert und mit Hilfe eines MotorShields zur Regelung eines Gleichstrommotors verwendet.

Abstract—This paper deals with the design and regulation of a servo motor test bench. Building on the theoretical content of the practical course Praktikum Regelungstechnik II and the associated script[QUELLE], we developed a new test stand, which transfers the in Matlab simulated results to a real motor model. Here we took over both purchasing and construction, as well as the implementation of the experiment. In the experiments, the positioning behavior of a servomotor is tested with three different controllers, these three controllers are a linear position controller, a structure-variable controller and a non-linear controller. These three controllers were implemented by us on a Raspberry Pi connected to an MotorShields to drive a DC motor.

I. EINFÜHRUNG

STATIONÄRE und mobile Roboter finden in der heutigen Industrielandschaft vielfache und variable Einsatzmöglichkeiten. Insbesondere Roboterarme übernehmen die unterschiedlichsten Tätigkeiten, sei es als bewegliche Halterung für Kameras um Bauteile zu untersuchen oder als Werkzeuge um Bauteile zu bewegen und zu bearbeiten. Bei all diesen Aufgaben ist eine hohe Positioniergenauigkeit der einzelnen Achsen des Roboters von größter Bedeutung. Jedes Roboter-gelenk wird dabei separat angetrieben. Elektromotoren haben sich dabei als Standardlösung durchgesetzt, da sie im Vergleich zu anderen Antriebsarten sauber, günstig und ungefährlich sind[vgl. Skript PSS].

Diese Arbeit befasst sich insbesondere damit, einen Versuchsstand zu entwickeln, der es ermöglicht, unterschiedliche Regelungskonzepte zur Positioniergenauigkeit an einem echten Motormodell zu untersuchen. Die verschiedenen Ergebnisse können dann hinsichtlich der Ausregelzeit, dem Überschwingen und der Regelabweichung miteinander verglichen werden. Der Versuchsstand richtet sich dabei an Studierende, die im Rahmen Ihrer Ausbildung in der Ingenieurwissenschaft an einem Motormodell verschiedene Reglertypen testen können. Da sich der Versuchsstand an Studenten richtet, war ein

Hauptaugenmerk der Planung neben den zuvor erwähnten regelungstechnischen Aspekten die visuelle Demonstrierbarkeit, das Sicherheitskonzept und die einfache Zugänglichkeit der Programmierung. Auch muss der Aufbau die nötige Robustheit aufweisen, um bei Fehlbedienung keine Schäden zu verursachen und davonzutragen. Verwirklicht haben wir dies durch einen mobilen Versuchsstand, an dem ein Motor einen angeschlossenen Drehteller mit Gradscheibe dreht. Die Gradscheibe dient dabei rein visuellen Aspekten zur Verdeutlichung des Reglerergebnisses. Die Programmierung umfasst drei unterschiedliche, parametrierbare Reglertypen.

A. Gliederung

Die Gliederung richtet sich nach unserer Reihenfolge der Arbeitsschritte. Zuerst mussten wir die Problemstellung analysieren und uns ein geeignetes Konzept für den Versuchsstand überlegen. Dazu gehört welche Komponenten wir einkaufen wollten, wie wir diese miteinander verbinden und welche Ergebnisse wir basierend auf den Bauteilen erwarten. Somit beschäftigt sich Kapitel II mit der Konzeption und Konstruktion des Versuchsstands.

Als nächster Schritt musste das neue Modell indentifiziert werden und die Regler dafür ausgelegt werden, dazu nutzten wir MatLab und die dazugehörige Erweiterung Simulink. Der Prozess der Reglerauslegung basiert dabei auf der Arbeit und den Matlab-Skripten des Praktikums Regelungstechnik. Dies wird in den Kapiteln III, IV und V erläutert.

In Kapitel VI gehen wir auf die Implementierung des Systems ein, dazu gehören neben der Implementierung der Regler auch die Ansteuerung von Motor und digitalem Encoder.

Im letzten Teil der Arbeit vergleichen wir die am Versuchstand gemessenen Ergebnisse mit den in Matlab simulierten, ziehen darauf basierend unser Fazit und geben einen Ausblick für mögliche zukünftige Arbeiten(Kapitel VII und VIII).

II. KONZEPTION UND KONSTRUKTION DES VERSUCHSSTANDS

Aufgabe des Versuchsstandes ist es, die Simulation des aktuellen Versuchs zur Regelung eines Servomotors im Modul Praktikum Regelungstechnik II durch ein reales Modell zu erweitern. Das im Versuch simulierte Modell besteht aus einem Motor mit 30:1 Untersetzungsgetriebe und einen Tachogenerator. Ein Ausgangspotentiometer gibt die aktuelle Motorposition zurück. Der Motor wird über einen Servoverstärker angetrieben. Dieser wird durch die Gegentaktung eines Vorverstärkers angesteuert. Das Eingangssignal des Vorverstärkers ist wiederum der Ausgang des angeschlossenen Reglers. Der Positionssollwert wird über ein Potentiometer vorgegeben. Somit kann der Regler auf die Ist-Geschwindigkeit und Ist-Position zurückgreifen. Insgesamt handelt es sich um ein

analoges Regelsystem[QUELLE].

Wir haben uns für ein digitales System entschieden, die Sollwertvorgabe und die Stellgrößenberechnung werden in Software verwirklicht und die Messdaten kommen von einem internen, digitalen Encoder. Dies hat den Vorteil, dass wir die Ergebnisse leichter dokumentieren können und neue Regler können schnell implementiert werden. Auch resultiert dies in weniger Komponenten, die verbaut werden müssen.

Anschließend eine Erläuterung der Eigenschaften des Versuchsstands und der einzelnen verwendeten Komponenten.

A. Eigenschaften des Modells

Anforderungen an das Modell waren die regelungstechnischen Notwendigkeiten wie eine hohe Messdynamik und Stelleingriff. Außerdem eine gute Demonstrierbarkeit, günstiger Einkauf der Komponenten und die nötige Sicherheit für die Nutzer des Versuchsstands.

Die verwendeten Bauteile des im Versuchsskript[QUELLE] genutzten Modells sind nicht dokumentiert. Lediglich die Systemparameter der Regelstrecke liegen vor. Insgesamt sahen wir uns dadurch dazu gezwungen, ein komplett neues Modell zu entwickeln. Die benötigten Bauteile kauften wir selbst ein. Aus diesen Gründen haben wir bei den eingekauften Komponenten ein besonderes Augenmerk auf die Verfügbarkeit und den Preis gelegt. Die Anschaffungskosten des gesamten Versuchsstands belaufen sich auf unter 100€. Die Liste der benutzten Komponenten beläuft sich wie folgt:

- *Raspberry PI 3B+*[QUELLE]
- *ASLONG JGY-371 12V Gleichstrommotor mit eingebautem Encoder*[QUELLE]
- *Adafruit DC & Stepper Motor HAT for Raspberry Pi - Mini Kit*[QUELLE]

Der Gleichstrommotor *ASLONG JGY-371* erhält seine Versorgungsspannung von einem *Adafruit DC & Stepper Motor HAT*, welches als *MotorShield* dient. Der Motor dreht eine Drehscheibe, die zur Visualisierung der Motorposition dient. Die Ansteuerung des *MotorShields* übernimmt ein *Raspberry PI 3B+*. Ebenso findet die Auswertung der Signale des Encoders über den *Raspberry PI 3B+* statt. Dieser ist ausgestattet mit dem Betriebssystem *Raspbian VERSIONSNUMMER*. Die Programmierung findet unter *Python VERSIONSNUMMER* statt.

B. Mechanik

Der mechanische Teil des Versuchsstands besteht aus einem Motor, Flansch oder auch Nabe genannt und der Drehscheibe. An der zylindrischen Motorwelle mit seitlicher Planfläche ist der Flansch befestigt, an dem wiederum der Drehteller festgeschraubt wurde. Zum Aufbau des Versuchsstands gehört des weiteren noch eine Platte an deren Unterseite der Motor befestigt ist. Durch eine Bohrung in dieser Platte wird die Motorwelle geführt. Somit befindet sich an der Oberseite der Platte, ohne Kontakt zu dieser, der Drehteller. Die Platte besitzt zusätzlich vier Standfüße. Abbildung 1 zeigt eine CAD-Explosionszeichnung des Versuchsstands ohne Motor. Die Zeichnung wurde mit der Software FreeCAD erstellt.

Für eine technische Zeichnung des Motors siehe [VERWEIS].

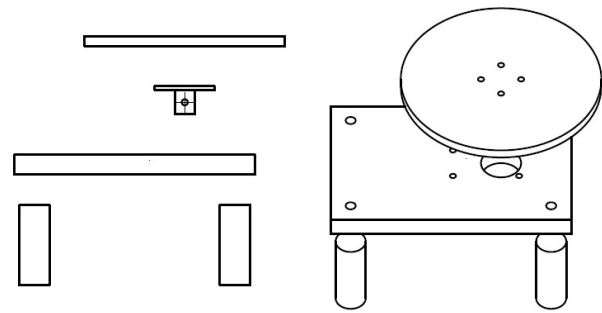


Bild 1. Explosionszeichnung der CAD-Konstruktion des Versuchsstands

Alle Komponenten wurden in der Werkstatt aus Aluminium gefertigt. Als Alternative haben wir mehrere Bauteile mit Hilfe eines CAD-Druckers fertigen lassen. Im Vergleich zu Aluminium ist der Vorteil von Plastikkomponenten aus dem CAD-Drucker die einfache Herstellung und das leichte Gewicht. Jedoch haben wir uns dagegen entschieden, da die Oberflächengüte mangelhaft war und die Plastikbauteile eine geringere Haltbarkeit und Robustheit aufweisen.

C. Motor und Encoder

TABELLE I
LEISTUNGSDATEN DES MOTORS MIT ENCODER[QUELLE]

Nennspannung	12V
Nennstrom	0,6A
Nennndrehzahl	30rpm
Nennmoment	0,73Nm
Getriebe-Untersetzungsverhältnis	200:1
Encoder ppr(Pulse per Round)	11
Versorgungsspannung Encoder	3,3V

Als Servomotor haben wir uns dazu entschieden, den im Vergleich preisgünstigen Gleichstrommotor JGY-1 von ASLONG zu verwenden. Dieser besitzt einen bereits eingebauten, antriebsseitigen digital Encoder. Antriebsseitig bedeutet, dass dieser Encoder die Position des Motors vor dem Getriebe direkt am Rotor misst. Der Motor kann sich in beide Richtungen drehen und der Encoder kann dies auch erfassen. Am Antriebsstrang ist zusätzlich ein Getriebe befestigt. Die Leistungsdaten des 2-phasigen Gleichstrommotors mit Metallbürsten sind Tabelle I zu entnehmen. Seine Versorgungsspannung erhält der Motor über ein Motorshield.

Zusätzlich ist an diesem Motor ist ein Inkrementalencoder angeschlossen. Der Inkrementalencoder liefert eine bestimmte Anzahl von Impulsen pro Umdrehung(PPR) der Motorwelle. Die zwei Ausgänge, Signal A und B genannt, sind Rechteckimpulse mit einer Phasendifferenz von 90°. Die PPR des Inkrementalgebers gibt somit die Anzahl der Rechteckwellen an, die ausgegeben werden, wenn er sich um eine Umdrehung dreht. Sind beide Signale high, ist der Rotor um ein Inkrement gewandert. Abhängig davon, welches Signal 90° voraus- oder nacheilt, lässt sich dann die Drehrichtung bestimmen, siehe Abbildung 2[QUELLE]. Die Anzahl von

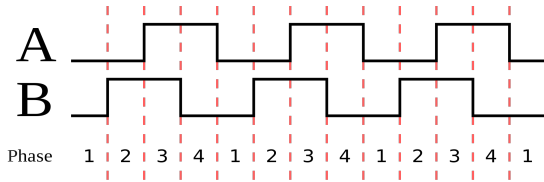


Bild 2. Inkrementalgeber-Quadratur-Rechteckwellenausgang mit Rechtsdrehung[?]

Impulsen pro Umdrehung der Motorwelle zu bestimmen, muss die Anzahl der Impulse pro Umdrehung des Rotors noch mit dem Untersetzungsverhältnis multipliziert werden, hieraus ergibt sich die Auflösung an der Motorwelle zu 2100 Impulse pro Umdrehung.

D. Motorshield und Raspberry Pi

Der Gleichstrommotor wird über die angelegte Spannung gesteuert. In diesem Projekt wird die Spannung am Motor mithilfe der PWM-Technologie erzeugt. Die Erzeugung des PWM-Signals übernimmt dabei ein Motorshield von Adafruit. Dieses Motorshield wird zusätzlich mit externen 12V gespeist. Das Motorshield erzeugt nun Rechteckimpulse mit variabel einstellbarem Tastgrad. Ein Tastgrad von bspw. 50% bedeutet das der Pegel des Rechteckimpulses für die Hälfte einer Periode auf 12V ist, für die andere Hälfte auf 0V. Dies entspricht dann am Motor einer Gleichspannung von 6V[QUELLE]. Der Tastgrad lässt sich über den verbundenen Raspberry Pi einstellen. Dieser wird über die implementierte Regelung berechnet. Die Kommunikation erfolgt über das I2C-Protokoll. Vom Hersteller des Motorshields wird eine Bibliothek zur Ansteuerung des Motorshields mitgeliefert, die wir in unserer Programmierung auch verwenden[QUELLE]. Die Programmierung findet in der Programmiersprache Python statt.

Für die Betriebsplattform dieses Projekts haben wir uns wegen der einfachen Bedienung und Implementierung, der hervorragenden Leistung und des niedrigen Preises für ein Raspberry Pi 3B+ samt Motorshield von Adafruit entschieden.

Siehe Abbildung 3 für die komplette Regelstruktur.

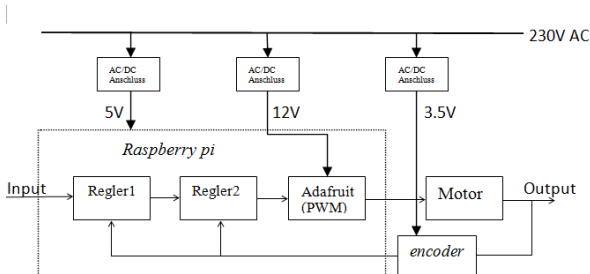


Bild 3. Systemblockdiagramm

III. IDENTIFIKATION DES MODELL

Um dieses System mathematisch zu beschreiben und den Regler zu entwerfen, sollten die Systemparameter vollständig

gemessen und bekannt. Aber manche Parameter können wegen der Zusammenstellung nicht genau oder einfach gemessen. Deshalb wird das System aus Messdaten des Ein- und Ausgangs abgeleitet und identifiziert.

A. Identifikation der Struktur des Modell

Der Servomotor wird durch PWM-Signal gesteuert. Die Spannung und Leistung des Motors hängt von der Frequenz und dem Tastgrad von PWM ab. In dem Fall, dass Tastgrad auf 100% bleibt, ändert die Spannung des Motors sich nicht, wenn die Frequenz sich ändert. Und die Spannung ist linear proportional zu dem Tastgrad. D.h. je kleiner der Tastgrad ist, desto niedriger ist die Spannung. Deshalb könnte die Spannung des Motors als Stellgröße sein, und durch Tastgrad des PWM-Signals linear geändert werden. Bei der niedrigen Frequenz zittert der Motor, während er bei höherer Frequenz glatter drehen kann. Daher wird die Frequenz auf 1,52kHz gestellt, welche die höchst durch Motorshield generiert werden kann. Darüber hinaus wird der durch Encoder messende Winkel als der Ausgang des System gesehen, danach kann die Winkelgeschwindigkeit mittels der Differenz des Winkels in einem sehr kleinen Zeitintervall berechnet werden.

Das System besteht aus zwei Teilsystem, also elektrisches und mechanisches Teilsystem, und kann als LZI-System betrachtet werden. Wie in Abbildung 4 gezeigt. U und I sind als Spannung und Strom des Servomotor, R und L sind der Widerstand und der Induktor der Armatur. U_{ind} ist induzierte Spannung. ω ist Winkelgeschwindigkeit, und J ist Trägheitsmoment der Belastung. Dann gilt es

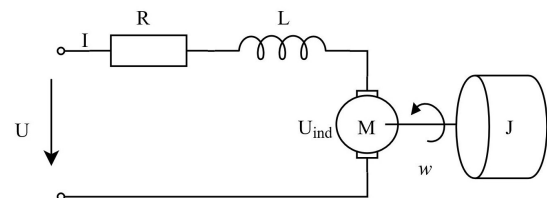


Bild 4. ESB Gleichstrommotor[QUELLE]

$$\begin{cases} U = RI + L \frac{dI}{dt} + U_{ind}, \\ J\dot{\omega} = M_A - D\omega, \\ U_{ind} = c_E \omega, \\ M_A = c_M I \end{cases} \quad (1)$$

wobei c_M und c_E Maschinenkonstante und D die Konstante für Reibung und Belastung[?]. Mit $\dot{\phi} = \omega$ ergibt sich die Gleichung

$$LJ\ddot{\phi} + (LD + RJ)\dot{\phi} + (RD + c_M c_E)\phi = c_M c_E U \quad (2)$$

wobei ϕ der Winkel ist. Wenn der Induktor vernachlässigt ist, gilt es

$$RJ\ddot{\phi} + (RD + c_M c_E)\dot{\phi} = c_M c_E U \quad (3)$$

Schließlich kann das System einfach mit ein PT2-Glied gebildet werden. Und wir haben auch bei der höher als 2. Ordnung in Matlab mittels Messdaten überprüft. Allerdings gibt es keine Verbesserung als PT2-Glied. Somit ist PT2-Glied ausreichend das System zu beschreiben.

B. Identifikation der Parameter des Modell

Wenn die Struktur der Strecke bestimmt wird, dann kann die Parameter mittels Messdaten durch Identifikationsverfahren bestimmt werden. Die Aufgabe liegt darin, die kontinuierliche Übertragungsfunktionen aus den diskreten Messdaten abzuschätzen. Es gibt viele Möglichkeiten, z.B. *The Maximum Likelihood Method*. Bei *The Maximum Likelihood Method* muss ein vernünftiger Anfangsparameterwert bekannt. Daher gibt es eine Beschränkung. Die andere Möglichkeit ist Method based on *State Variable Filters*. Die Messdaten sollten genau zu den jeden diskreten Zielpunkten abgetastet. Und um diese Daten für die Identifikation des kontinuierlichen Systems benutzen zu können, wird *zoh(zero order hold)* zwischen zwei Abtastpunkte verwendet, so dass die Messdaten bis zum nächsten Abtastpunkt konstant bleiben[?]. Man betrachtet das Modell:

$$Y(s) = \frac{B(s)}{A(s)}U(s) \quad (4)$$

also auch gilt es :

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_n y(t) = b_1 u^{(m-1)}(t) + \dots + b_m u(t) \quad (5)$$

Danach wird die Gleichung mit einem kontinuierlichen Tiefpass $L(s)$ gefiltert, so dass man erhält:

$$\begin{cases} z_k(t) = L(s)y^{(k)}(t) \\ w_k(t) = L(s)u^{(k)}(t) \end{cases} \quad (6)$$

Schließlich gilt die Gleichung:

$$z_n(t) + a_1 z_{(n-1)}(t) + \dots + a_n z_0(t) = b_1 w_{(m-1)}(t) + \dots + b_m w_0(t) \quad (7)$$

Ohne Rauschen wird die Parameter durch *Least Squares Method* abgeschätzt:

$$\hat{\theta} = (\Psi^T \Psi)^{-1} \Psi^T z_n \quad (8)$$

Aber wenn das Rauschen berücksichtigt wird, und nicht Weißes Rauschen ist, ist die Abschätzung nicht biasfrei. Deshalb wird *Instrumental Variable (IV) Method* verwendet [?]. Instrumentaler Vektor ζ wird aus Ψ berechnet. Zum Schluss wird die Parameter ausgedrückt:

$$\hat{\theta} = (\zeta^T \zeta)^{-1} \zeta^T z_n \quad (9)$$

Glücklich gibt es schon dieser Algorithmus in Matlab. Man kann mit Befehl `tfest()` anrufen. Weil mit einem I-Anteil das Signal glatter gefiltert, so dass es besser und genauer für Identifikation wäre. Deshalb verwenden wir die Messdaten des Winkel zu identifizieren. D.h. wird die Übertragungsfunktion ausgedrückt:

$$G_s(s) = \frac{k}{s(s+p)} \quad (10)$$

Das Ergebnis ist

$$G_s(s) = \frac{7521}{s(s+39,55)} \quad (11)$$

Dabei wird der Tastgrad als Stellgröße gesehen, und Winkel als Ausgang betrachtet. Man kann die Antwort des identifizierenden Modell mit Messdaten vergleichen. Wie in Abbildung 5 gezeigt.

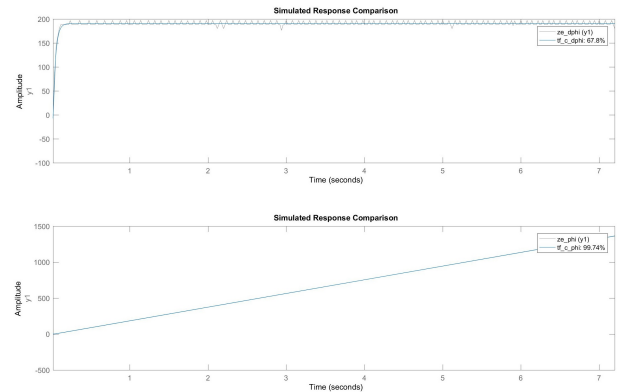


Bild 5. Vergleich von Messdaten und identifiziertem System, bei oben durch Winkelgeschwindigkeit, bei unten durch Winkel

IV. REGLERENTWURF

Die Strecke des Servoantriebs wird durch eine Kombination von PT1-Glied und I-Glied gebildet. Um das Einschwingverhalten des geregelten Systems gut gedämpft und schnell wie möglich zu machen, wird es mittels einer bestimmten Sturuktur der Regelung durchgeführt. Es geht um eine Kaskadenregelung[?]. Die Struktur wird in Abbildung 6 gegeben. Dann wird die PT1-Glied der bisher dargestellten Strecke wird in Innen-Regelkreis zuerst gut gedämpft und stabil geregelt, danach wird die ganze Strecke durch entsprechenden Außer-Regelkreis gut ausgeregelt. Bei Wal der Ausregelung des inneren Reglers spielen die schnelle Dynamik und die ausreichende Dämpfung eine wichtigere Rolle, während die stationäre Genauigkeit weniger wichtig ist[?]. Anschließend wird der Reger mittels eines linearen Reglers, eines Strukturreglers bzw. eines nicht linearen Reglers probiert.

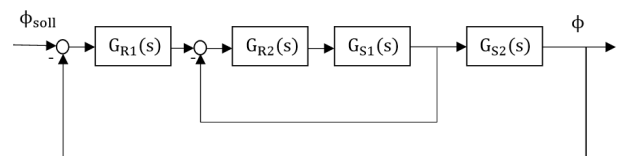


Bild 6. Die Struktur des KaskadenReglers

A. Linearer Lageregler

Bei linearer Regelung wird PD-Regler gesetzt. Der innere Regelkreis lässt sich durch k_d zurückführen. Wenn Kaskadenregelung probiert wird, sollte es zuerst den Stabilitätsbereich bezüglich k_d gefunden werden. Somit kann man durch

allgemeine WOK-Verfahren eines Polynomes versuchen. Das bedeutet, dass man k_d als Parameter auswählen, und finden, wie die Pole des geschlossenen Regelkreises sich ändern, wenn k_d von $-\infty$ bis ∞ läuft. Dann wird ein gut gedämpfter Wert für k_d in diesem Stabilitätsbereich gewählt. Abschließend wird nochmal WOK-Verfahren verwendet, ein entsprechender Wert für k_p des Außer-Regelkreises zu wählen, so dass der geschlossene Regelkreis schnellere Dynamik besitzen kann. Der ganze Regelkreis wird in Abbildung refimg:linear angegeben.

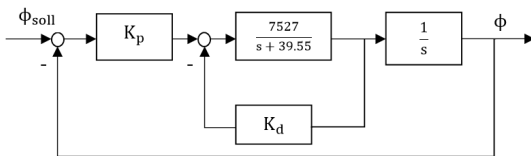


Bild 7. Die Struktur des linearen Reglers

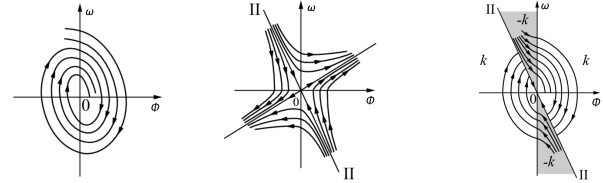
B. Strukturvariabler Regler

Ein Strukturregler kann zwischen verschiedenen Reglern hin und her schalten. Aufgrund der Eigenschaften der Strukturvariablenregelung eignet es sich gut für Servoregelungssysteme[?]. Die Schaltung kann durch eine oder mehr Schaltkurve bestimmen, so dass entsprechender Regler im verschiedenen Bereich aktiviert werden soll. Unter Berücksichtigung der allgemeinen Situation befindet sich im Zustandsraum des Systems $\dot{\mathbf{x}} = f(\mathbf{x})$ eine Schaltgerade $s(\mathbf{x}) = 0$ [?]. Dann wird der Zustandsraum in $s(\mathbf{x}) > 0$ und $s(\mathbf{x}) < 0$ zwei Teilgebiete getrennt. Die Trajektorie kann nicht nur durch Wechseln zwischen den verschiedenen Bereichen sondern auch durch Laufen von beiden Seiten auf die Schaltkurve zum Ende gehen. Die Schaltlogik kann einfach von Mikrokontroller vorgegeben. Wenn die Trajektorie beideseitig auf die Schaltkurve zuläuft, tritt Gleitzustand auf. D.h. muss es gelten:

$$\lim_{s \rightarrow 0} \dot{s} < 0 \quad (12)$$

Bei Gleitzustand besitzt das System eine gute Dynamik, aber es wird immer hochfrequent hin- und hergeschaltet[?], welche schlecht für Motor ist.

In diesem Fall werden zwei lineare Regler benutzt, die beiden nicht unbedingt stabil sind. Nun wird die Zustandsebene durch Zustandsgrößen ϕ und ω aufgespannt. In einem Gebiet sollte einen Regler mit Verstärkung k gesetzt, während in dem anderen Gebiet der Regler mit Verstärkung $-k$ aktiviert werden sollte. Und bei Verstärkung k kann der geschlossene Regelkreis als Oszillatorisch stabil sein, aber bei Verstärkung $-k$ es instabil ist. Dann wenn die beide Regler sich nur in einem entsprechenden Teilgebiet funktionieren lassen, kann die Trajektorie des Gesamtsystems aus der oszillatorisch stabilen und instabilen Trajektorie für k bzw. $-k$ gut verbunden werden und schnell, stabil zum Endwert laufen. Die Trajektorie von für die Regler mit der Verstärkung k bzw. $-k$ und das gesamte Strukturregler lässt sich in Abbildung ?? verdeutlichen. Die


 Bild 8. Von links nach rechts: Die Trajektorie für Regler mit Verstärkung k , Die Trajektorie für Regler mit Verstärkung $-k$, Die Trajektorie für Strukturregler [?]

Regelkreis wird in Abbildung 9 ausgedrückt.

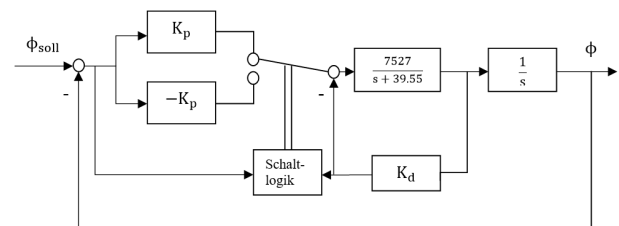


Bild 9. Die Struktur des Strukturreglers

C. Nichtlinearer Regler

Wenn in einem Regelkreis die Nichtlinearität auftritt, kann die Regelung kompliziert sein. Die Nichtlinearität kann z.B. durch Totzone, Zweipunktglied bzw. Dreipunktglied beschrieben werden. Durch Nichtlinearität kann es wahrscheinlich billiger bei dieser Nichtlinearität kann auch zu Dauerschwingung führen, welche schlecht für Regelung und nicht angenehm ist. Man kann diese Dauerschwingung mittels der Harmonischen Balance untersuchen. Die Voraussetzung der Harmonischen Balance ist so ausgedrückt:

$$L(j\omega) = -\frac{1}{N(A)}, \quad (13)$$

wobei $L(j\omega)$ der Frequenzgang des Systems ist, $N(A)$ die Beschreibungsfunktion der Nichtlinearität ist, welche nur von Amplitude A abhängt[?]. Aber die Method liefert nur Näherung der Ergebnisse. Dann muss man noch das System in Simulation genauer betrachten. Abbildung 10 zeigt die Struktur des Reglers.

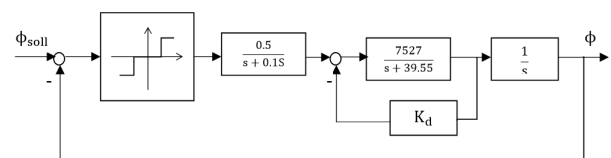


Bild 10. Die Struktur des nicht linearen Reglers

V. SYSTEMSIMULATION IN SIMULINK

Simulink stellt einen Grundsatz an verschiedenen grafischen Blöcke, die die Modellierung des Systems ermöglichen. Darüber hinaus gibt Simulink unterschiedliche Toolboxes, die das

Modell deutlich nachzuschauen und komplexe Struktur des Modells zu ermöglichen. z.B. kann man in diesem versuch mit Sprung aus Toolbox ein Sprungsignal unterschiedliche hoch eingeben. Das Modell hat schon in 11 gegeben worden. Das Modell wird in zwei Teilen unterteilt. Das erste Teil ist Geschwindigkeit-Teil, das mittels eines PT1-Glied beschrieben wird. Das zweite Teil ist das Integral-Teil, welche sich das gedrehte Winkel ergibt. In Simunlink kann man die Strecke wie in Abbildung 11 machen. Im weiteren wird

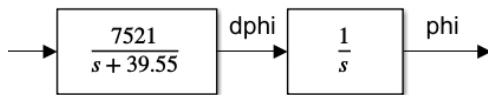


Bild 11. Die gebildete Streck in Simulink

linearer Regler, Strukturregler bzw. nichtlinearer Regler mit der Strecke verbindet und simuliert. In dieser Simulation ist der Eingangssignal ein Sprungsignal, dessen Endwert 60 Grad beträgt.

A. Simulationsergebnisse des linearen Lagereglers

Bei linearem Regler wird das Strutur der Kaskadenregelung verwendet. In Simulink kann man so wie in Abbildung ?? aufbauen. Die Zurückführung der Winkelgeschwindigkeit kann

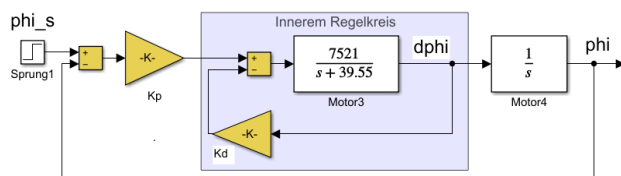


Bild 12. Der aufgebaut Reglkreis mit der Struktur der Kaskadenregelung in Simulink

einen inneren Reglerkreis bilden. Um eine Kaskadenregelung sinnvoll zu entwerfen, sollte die innere Dynamik schneller gewählt werden. Die innere Dynamik ist wie Abbildung 13 gezeigt, Die innere Dynamik besteht aus der geschlossenen

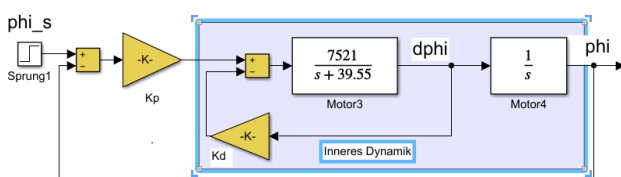


Bild 13. Die innere Dynamik einer Kaskadenregelung

innere Regelkreis sowie Integrtator. Die gesamte Übertragungsfunktion lautet:

$$G(s) = \frac{7521}{s^2 + 39,55s + 7521k_d s}. \quad (14)$$

Um die Übertragungsfunktion durch WOK zu analysieren, ist eine Umformung zu erwarten. Die Gleichung, die alle Punkt von der WOK beschreibt, ist das charakteristische Polynom:

$$N_w = N_0(s) + k \cdot Z_0(s), \quad (15)$$

dann erhält man die WOK-Gleichung:

$$\frac{Z_0(s)}{N_0(s)} = -1. \quad (16)$$

Das charakteristische Polynom der innere Dynamik ist:

$$s^2 + 39,55s + 7521k_d s. \quad (17)$$

Nach der Umformung ergibt sich:

$$k_d \frac{7521s}{s^2 + 39,55s} = -1. \quad (18)$$

Jetzt wird zuerst Eigentliche WOK mit positiver Wert von k_d in Abbildung 14 gezeigt. Es ist leicht zu erkennen, dass mit

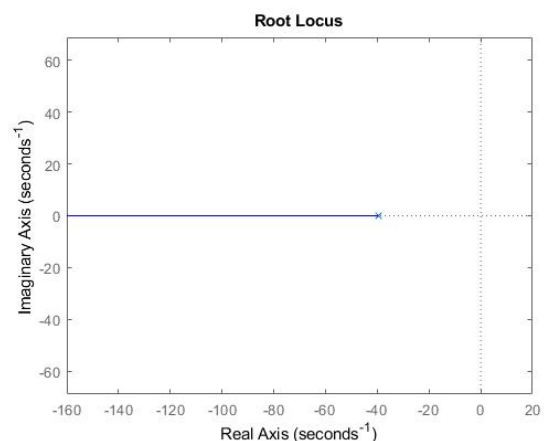


Bild 14. Eigentliche WOK der innere Dynamik

Positiver Wert von k_d alles Pole links von S-Ebene liegen. Die Pole sollte aber nicht so weit liegt, ansonsten entsteht hohe Regelabweichung. Darüber hinaus sollte komplementären WOK mit negativer Wert von k_d untersucht. Das Ergebnis wird in Abbildung 15 verdeutlicht. Für negative Werk von k_d von 0

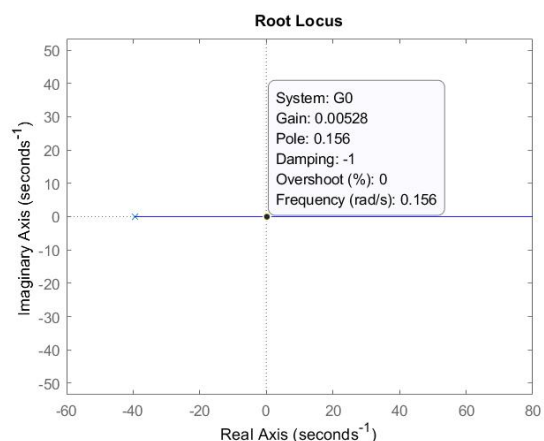


Bild 15. Komplementären WOK der innere Dynamik

bis -0,00528 liegt die Polen der innereren Dynamik in links S-Ebene. Die Polen sind auch sehr langsam.

Nach der Analyse der inneren Dynamik wählt man k_d gleich 0,0035, die Pole der geregelte Innere Dynamik liegt in -65.9,

die relativ schnell ist. Neben der Analyse der inneren Dynamik ist die Stabilität von Außen Dynamik mit dem Parameter k_p zu untersuchen. Der gesamte Regelkreis mit k_d gleich 0,0035 ist wie in folgender Abbildung 16 gezeigt: Die offene

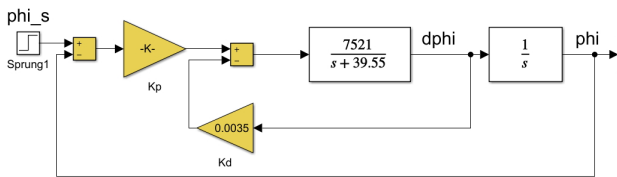


Bild 16. Struktur des außen Regelkreises

Übertragungsfunktion, die im WOK verwendet werden, lautet:

$$G_0(s) = \frac{7521}{s^2 + 65,87s}. \quad (19)$$

Dann wählt k_p gleich 0,144, der schnellst ist und aperiodische Grenzfall erreicht. Die WOK wird in Abbildung 17 gezeigt. Die Pole des geregelten außen Regelkreises beträgt -33,1,

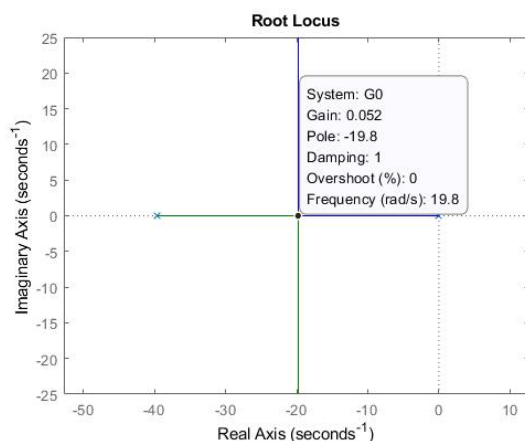


Bild 17. Eigentliche WOK des außen Regelkreises

die langsamer als die Pole der Inneren Dynamik, welche die notwendige Voraussetzung der Linearen Kaskadenregelung erfüllt. Die Simulation des gesamt geregelten System ist in Abbildung 18: Die Ausregelzeit beträgt 0,15 Sekunde, die besonders schnell ist. Außerdem ist die stationäre Genauigkeit erfüllt.

B. Simulationsergebnisse des strukturvariablen Reglers

In der Simulation von Strukturvariablen Regler wird zwei Reglern durch die Schaltung hin und her geschaltet. In diesem Fall kann es in Abbildung 19 gezeigt werden. Die Schaltung wird durch zwei Schaltgeraden $\phi = 0$ und $\dot{\phi} + a\phi = 0$ beschrieben. Die Schaltlogik wählt k als Regelverstärkung, wenn $\phi(\dot{\phi} + a\phi) > 0$ ist (In den hellen Sektoren), oder die wählt $-k$, wenn $\phi(\dot{\phi} + a\phi) < 0$ ist (In den blauen Sektoren). Dann wird die Regelverstärkung k und $-k$ in unterschiedlichen Bereichen der Phaseebene dargestellt, wie in Abbildung 20 (links) gezeigt. Zuerst wird die Trajektorie durch Wechseln zwischen den verschiedenen Bereichen zum Ende gehen. Dabei wählt

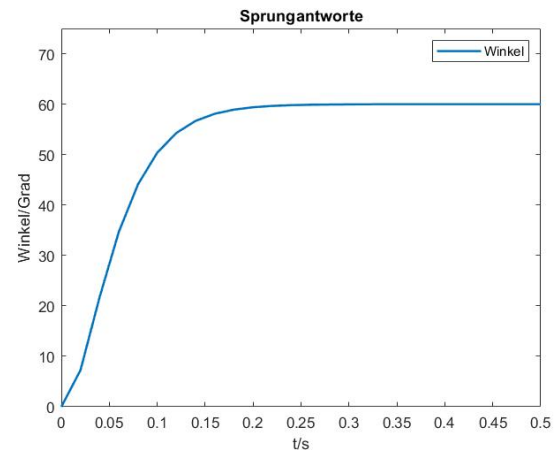


Bild 18. Sprungantworten des geschlossen Regelkreises

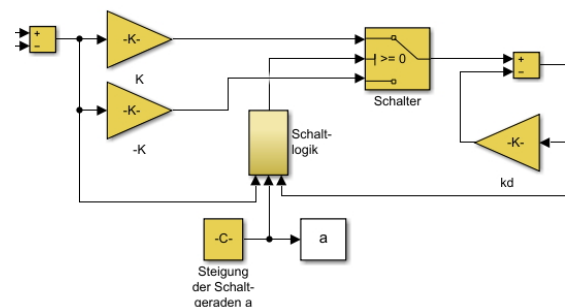


Bild 19. Realisierung des Strukturreglers in Simulink

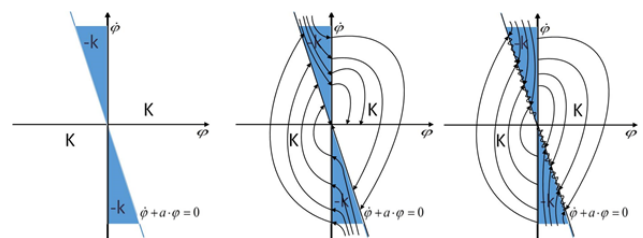


Bild 20. Schaltgerade und Trajektorienverläufe[QUELLE]

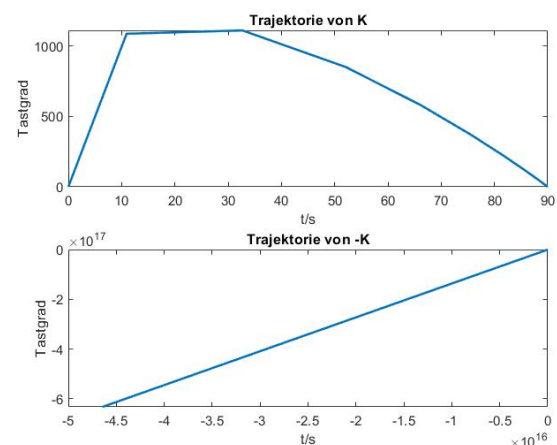


Bild 21. Ergebnisse der erster Simulation

man zuerst k gleich 0,144, k_d gleich 0,0035. Die Trajektorie läuft wie in Abbildung 21. Es ist zu erkennen, dass das ein stabile Trajektorie mit K und instabile Trajektorie mit -K ist. Wählt man eine geeignete Schaltgerade, so kann man das theoretische Verhalten aus Abbildung 20(mittig) erreichen. In der Simulation wird unterschiedliche Steigung a von Schaltgerade probiert. Wenn a zu klein gewählt, dann wird die Sprungantwort stark schwingt. Wenn man aber a sehr groß gewählt, dann tritt große Überschweit auf. Nach zahlreichen Versuche wird a gleich 30 gewählt. Das simulierte Ergebnis ist in Abbildung 22 gezeigt. Die Ausregelzeit beträgt 0,17

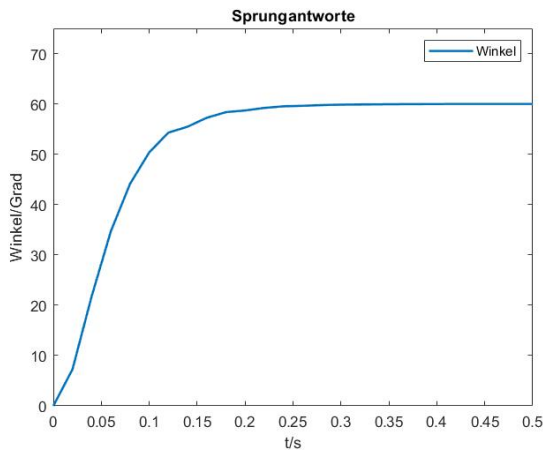


Bild 22. Sprungantworten der Strukturvariablen Regler erster Simulation

sekunde, die auch sehr schnell ist. Außerdem ist die stationäre Genauigkeit erfüllt.

Bei der Simulation von Gleitzustand wählt man k gleich 0,144, k_d gleich -0,01. Die Trajektorie von k und $-k$ ist wie in Abbildung 23 gezeigt. Es ist zu erkennen, dass das ein instabil

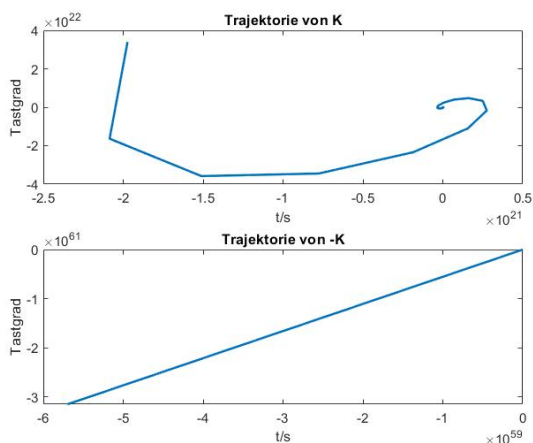


Bild 23. Sprungantworten der Strukturvariablen Regler erster Simulation

Trajektorie mit K und auch ein instabil Trajektorie mit -K ist. Wählt man eine geeignete Schaltgerade, so kann man das theoretische Verhalten aus Abbildung ??(rechts) erreichen. Dies nennt man einen Gleitzustand. In der Simulation wird unterschiedliche Steigung a von Schaltgerade probiert. Wenn a zu klein oder zu groß gewählt, dann ist die Sprungantwort

ganz schlecht und nicht stabil. Nach zahlreiche Versuche wird a gleich 5,8 gewählt. Das simulierte Ergebniss wird in Abbildung 24 gezeigt. Die Ausregelzeit beträgt 0,33 Sekunde, die nur ein bisschen langsam ist. Außerdem ist die stationäre Genauigkeit erfüllt. Aber gibt es einige leichte Schwingungen vor 0,33 Sekunde, die die Motor schaden kann.

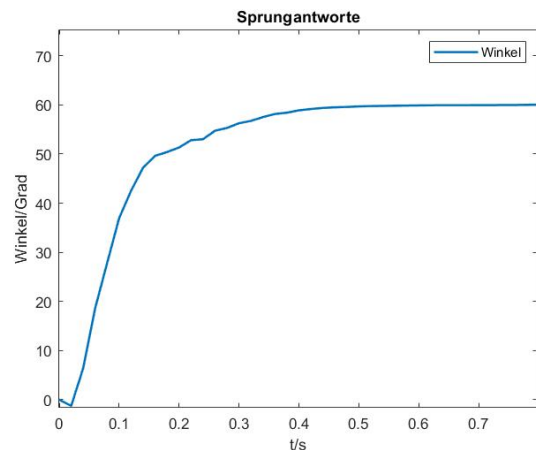


Bild 24. Sprungantworten der Strukturvariablen Regler zweiter Simulation

C. Simulationsergebnisse des nichtlinearen Reglers

In der Simulation von nichtlinearen Regler wird Nichtlinearität im Regelkreis hinzugefügt. Die Regelung mit Nichtlinearität ist besonders billiger. Z.B. wird das Zwei-Punkt Glied durch ein Schalter einfach realisiert. Aber mit Nichtlinearität führt meistens Dauerschwingung zu. Die Amplitude und Frequenz der Schwingung sollten durch Simulation bestimmt werden. Die Struktur des nicht linearen Reglers ist wie Abbildung 25 gezeigt. Hier werden das Zweipunktglied

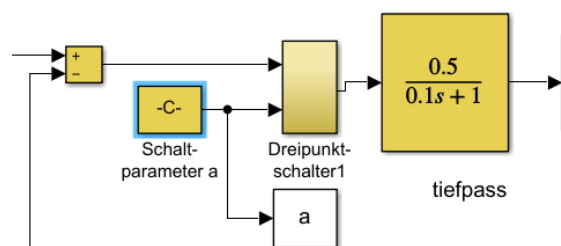


Bild 25. Realisierung des Nichtlinearer Reglers in Simulink

und das Dreipunktglied ausgewählt und probiert. Bei dem Zweipunktglied soll den Parameter a gleich 0 sein. Wenn der Sollwert auf 7 gesetzt wird, ergibt sich das Ergebnis in Abbildung 26. Amplitude der Dauerschwingung beträgt c.a. 2,50 , die Frequenz ist c.a. 0,34 Hz. Wenn a auf 0,5 gesetzt wird, kann das Ergebnis in Abbildung 27 ausgedrückt. Die Amplitude der Dauerschwingung beträgt c.a. 2,45 , die Frequenz ist c.a. 0,32 Hz.

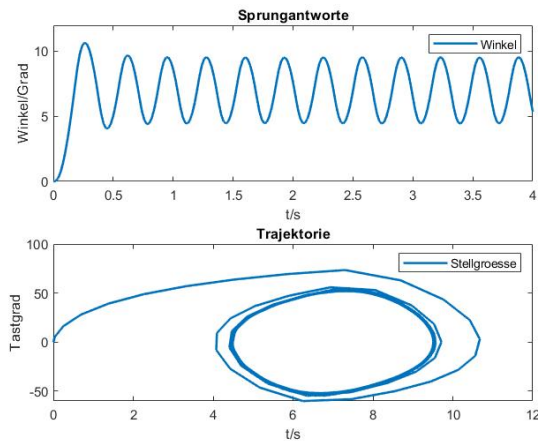


Bild 26. Simulationsergebnisse für Dauerschwingung

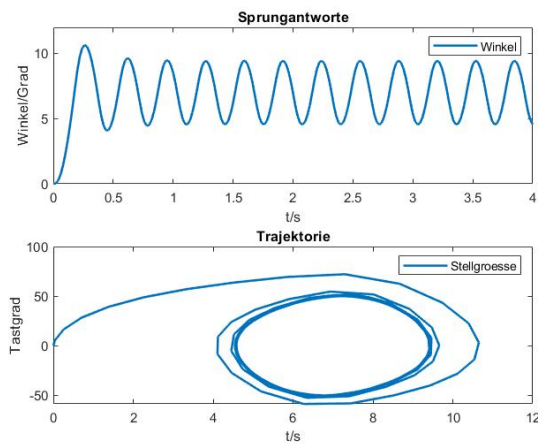


Bild 27. Simulationsergebnisse für Dauerschwingung

VI. IMPLEMENTIERUNG DES SYSTEMS

Wenn das System in Simulink gut geregelt wird, sollte der Regler in dem realen Servomotor umgesetzt. Der Motor wird durch Raspberry Pi über Motorshield mittels PWM gesteuert, so dass der Regler durch Implementierung eines Programms in Raspberry Pi realisiert werden kann. D.h. ist der Regler nun diskret. Dabei wird auch zoh(zero order hold) verwendet. Außerdem gibt es auch Beschränkung der Stellgröße, also der Tastgrad des PWM-Signals, der durch einigen Befehlen geändert werden kann. Der Tastgrad liegt zwischen 0,05 bis 1, wobei bei 0,05 der Motor nicht drehen kann. Deshalb entspricht diese eine Totzone und danach ein Zweipunktglied mit maximalem Wert 1.

Da der den Tastgrad ändernde Befehl relativ länger als normaler Befehl, muss diese aufgewendete Zeit berücksichtigt werden, wenn man genau Abtastzeit setzt.

Die Abtastzeit soll wegen einiger zeitaufwendigen Befehlen, die den Tastgrad des PWM-Signals ändern können, nicht kleiner als 0,015s sein. Daher wählen wir 0,02s aus.

A. Implementierung des Programms zur Regelung

Das Messen des Winkels soll sich ohne Störung immer laufen lassen, so dass die Rückführung immer genau und richtig sein kann. Deshalb wird es bei Implementierung durch ein *Threading* gleichzeitig gemacht, während der Regler arbeitet. Die Realisierung des Reglers wird im diskreten Bereich durchgeführt. Die Rückführung von Winkel und Winkelgeschwindigkeit soll schrittweise gemessen und berechnet werden. D.h. kann der Regler zu jedem Abtast-Schritt mittels der bis hier schon aktualisierten Werte von Winkel bzw. Winkeldifferenz die Abweichung zu dem Sollwert und die durchschnittliche Winkelgeschwindigkeit in diesem kleinen Zeitintervall berechnen. Dann wird die Stellgröße durch Error und Winkelgeschwindigkeit mit der Verstärkung und auf der bestimmten Voraussetzung, also z.B. die Position in der Zustandsebene, ausgerechnet und danach eingesetzt. Natürlich soll die Stellgröße sich bevor Einsetzung überprüfen und in dem zulässigen Bereich legen lassen. Dann werden alle benötigte Größe aktualisiert. Somit geht es zum nächsten Schritt. Jeder Schritt findet zum entsprechenden Abtastzeitpunkt an, kann zu früher noch zu später. Der Ablauf des Programms ist wie Abbildung 28 angezeigt.

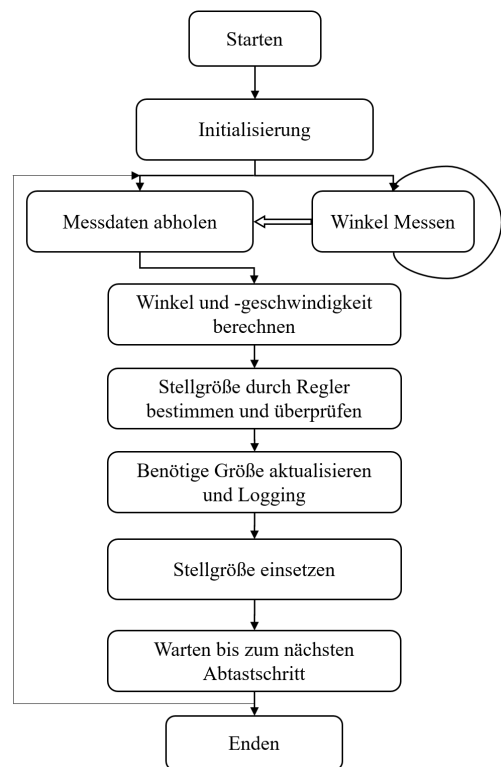


Bild 28. Ablauf des Programms zur Regelung

B. Einsetzen der Regler und die Ergebnisse

Wenn diese Regler getestet werden, setzen wir $\phi_{soll} = 60$. Bei dem linearen Regler wird die Stellgröße, also der Tastgrad, für k-ten Schritt durch

$$u_k = k_p(\phi_{soll} - \phi_k) - k_d\omega_k \quad (20)$$

berechnet. Da es bei Stellgröße eine Beschränkung gibt, kann es in diesem Fall durch

$$u_k = \begin{cases} 1 & u_k > 1 \\ -1 & u_k < -1 \\ 0 & -0,05 < u_k < 0,05 \\ u_k & \text{anderes} \end{cases} \quad (21)$$

Nun wird u_k 0,144 und entsprechend k_d 0,0035 ausgewählt. Der Winkel und der Tastgrad sind in Abbildung 29 gezeigt. Aus der Abbildung kann man die Ausregelzeit um c.a. 0,34s ablesen. Kein Überschwingen tritt auf, und der Endwert auf 59,7 bleibt.

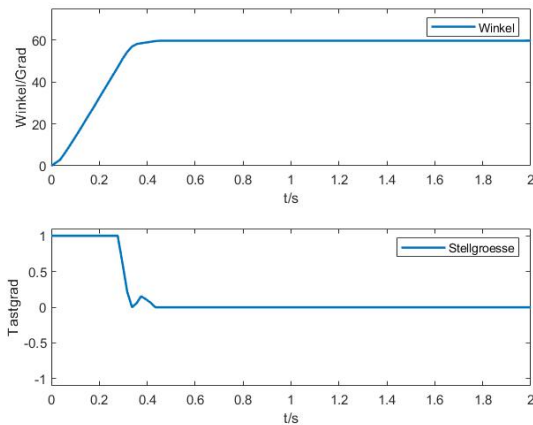


Bild 29. Ergebnisse des linearen Reglers im realen System

Bei dem Strukturregler wird der Tastgrad, für k-ten Schritt durch

$$u_k = \begin{cases} k_p(\phi_{soll} - \phi_k) - k_d\omega_k & \text{i} \\ -k_p(\phi_{soll} - \phi_k) - k_d\omega_k & \text{ii} \\ \text{wenn } \begin{cases} \text{i: } -(\omega - a(\phi_{soll} - \phi))(\phi_{soll} - \phi) \geq 0 \\ \text{ii: } -(\omega - a(\phi_{soll} - \phi))(\phi_{soll} - \phi) < 0 \end{cases} & \end{cases} \quad (22)$$

berechnet, wobei $-a$ die Steigung der Schaltgrade ist. Jetzt man wählt $a = 5,8$, $k_p = 0,144$ und $k_d = -0,01$. Das Ergebnis lässt sich in Abbildung 30 anzeigen. Man kann ablesen, dass die Ausregelzeit 0,6 ist, und Endwert 59,7 beträgt.

Bei dem nichtlinearen Regler lässt die Stellgröße sich für k-ten Schritt durch

$$u_k = u_{k-1}e^{-10T_s} + c_a v_k - b\omega_k$$

$$v_k = \begin{cases} \text{sgn}(\phi_{soll} - \phi_k) & |\phi_{soll} - \phi_k - a| > 0 \\ 0 & |\phi_{soll} - \phi_k - a| \leq 0 \end{cases} \quad (23)$$

berechnen. Wobei $c_a = 5$, $b = 0$ gelten, und T_s Abtastzeit ist, a der Parameter des Dreipunktglieds ist. Danach wird die Stellgröße auch durch die Gleichung (21) überprüft. Wenn $a = 0$ ist, sollte das Dreipunktglied nicht anders als das Zweipunktglied. Das Ergebnis wird durch Abbildung 31 beschrieben. Während $a = 0,5$, führt das Dreipunktglied zur Dauerschwingung. Diese Ergebnisse lässt sich in Abbildung

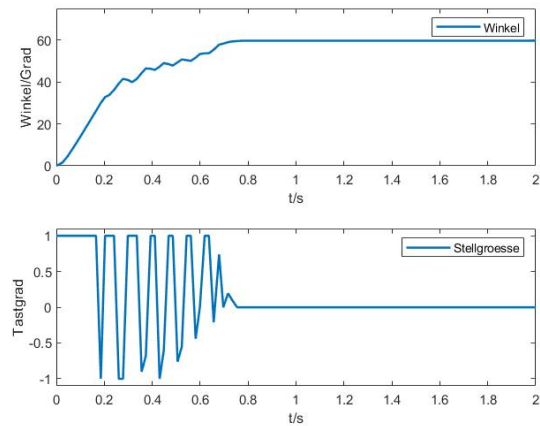


Bild 30. Ergebnisse des Strukturreglers im realen System

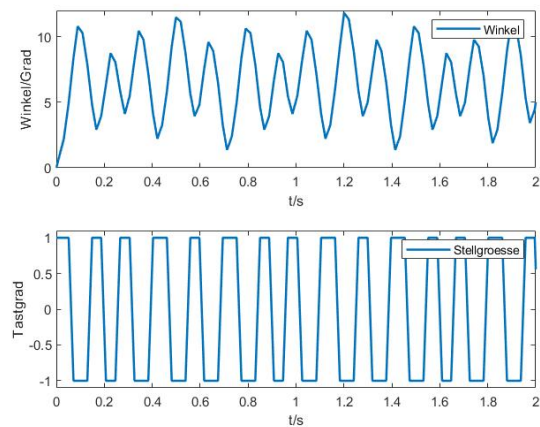


Bild 31. Ergebnisse des nicht linearen Reglers im realen System mit $a = 0$

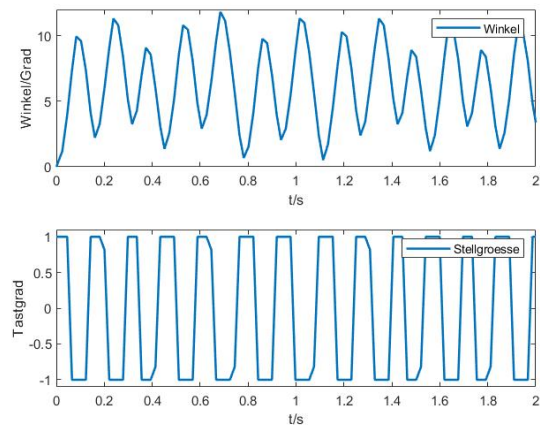


Bild 32. Ergebnisse des nicht linearen Reglers im realen System mit $a = 0,5$

32 anzeigen. Man kann dazu sagen, dass nicht linearer Regler wahrscheinlich billiger wäre, aber es einfacher zu Dauerschwingung führen kann.

Maximilian Ebersbach Biographie Autor A.



VII. ERGEBNISSE IM VERGLEICH VON SIMULATION UND MODELL

Die Ergebnisse aus der Simulation und dem realen System sind nicht perfekt angepasst. Bei linearem Regler wird es in Simulation viel schneller als in dem realen System. Der Grund liegt darin, dass die Stellgröße im realen System eine Beschränkung besitzt. Das bedeutet, dass der Tastgrad als Stellgröße nur zwischen -1 bis 1 möglich sein kann. Allerdings kann die Stellgröße in der Simulation der linearen Regelung 1 oder -1 überschreiten. Deshalb könnte es schneller. Und die Endwert im realen System ist nicht so perfekt genau, weil die Stellgröße eine Totzone besitzt. D.h. wenn der Tastgrad kleiner als 0,05 ist, kann der Motor nicht drehen. Bei Sturkturregler kann es im realen System auch ein bisschen langsamer. Die Geschwindigkeit kann nur am höchsten c.a. 188 Grad/s erreichen. Somit wird die Trajektorie anderes als die in Simulation, d.h. kann die Trajektorie tatsächlich nur in einem Streif laufen. Und die Schaltung ist sehr scharf. Bei nicht linearem Regler kann das reale System komplexiert sein und möglicher zur Dauerschwingung gehen.

Jianguang Gao Biographie Autor B.



Außerdem kann man die Abtastzeit nicht übersehen. Wegen der Begränzung der Implementierung in Programmierung kann die Abtastzeit nicht kleiner als 0,16s sein. Und man kann nicht perfekt genau zu jedem Abtastzeitpunkt, also zu $t_k = k \cdot T_s, k = 0, 1, 2, \dots$ bestimmen, weil der Mikrokontroller in jedem Abtast-Schritt nicht immer dieselbe Befehle ausführen kann. Was weitere ist das Modell, die nicht 100% richtig und genau ist, da z.B. die Reibung vernachlässigt wird.

Haoming Wang Biographie Autor C.



VIII. FAZIT UND AUSBLICK

DANKSAGUNG

Wenn ihr jemanden danken wollt, der Euch bei der Arbeit besonders unterstützt hat (Korrekturlesen, fachliche Hinweise,...), dann ist hier der dafür vorgesehene Platz.

LITERATURVERZEICHNIS

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [2] Deutsche Forschungsgemeinschaft, *Vorschläge zur Sicherung guter wissenschaftlicher Praxis*, Denkschrift, Weinheim: Wiley-VCH, 1998.

Zhenhui Zhou Biographie Autor D.

