

Modellbasierte Softwareentwicklung mit MATLAB für Embedded Controller STM32F4-Discovery

Projektarbeit

im Fachbereich Elektrotechnik-Informationstechnik
Studiengang Fernstudium Elektrotechnik Master of Science
der
Hochschule Darmstadt

**Benedikt Geib
Daniel Hartung
Christof Kötting
Igor Lewin
Jörg Thomasberger**

Betreuer: Prof. Dr. Serge Zacher

Bearbeitungszeitraum: 10. April 2015 bis 25. September
2015

Darmstadt, September 2015

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis.....	6
Tabellenverzeichnis	9
Abkürzungsverzeichnis	10
Vorwort.....	11
1 Einleitung	12
1.1 Überblick Aufgabenstellung.....	12
1.2 Stand der Technik.....	14
1.2.1 Hardware	14
1.2.2 Software.....	19
1.3 Eingeschlagener Realisierungsweg	23
1.3.1 Grundsätzliches Vorgehen.....	23
2 Projektmanagement	24
2.1 Spezifikation des Ergebnisses	24
2.2 Risikoanalyse.....	24
2.3 Anwendungsfälle	24
2.4 Projektplan.....	25
2.5 Projektablauf.....	27
2.6 Meilensteine	27
3 Hardwareentwurf.....	29
3.1 Beschreibung:	31
3.2 Spannungsversorgung:	31
3.3 Sensoren:	32
3.4 H- Brücke:	32
3.5 Kommunikation	32
4 Controller Software.....	33
4.1 Allgemeine Beschreibung	33
4.2 Main.....	33
4.3 Systemhandler	34
4.4 Regulator	35

4.4.1	Aufbau	36
4.4.2	Code-Generierung aus Matlab	36
4.5	Motor Controller	37
4.6	Position	37
4.7	Uart	37
4.8	Timer	37
4.9	Adc	37
5	Systemanalyse	38
5.1	Messaufbau und Auswahl Testsignale	38
5.1.1	Auswahl Testsignale	38
5.1.2	Messaufbau	39
5.2	Systemanalyse ohne Vorkenntnisse	39
5.2.1	Beschreibung der Regelstrecke des Antennenmodells	39
5.2.2	Signalanalyse	41
5.2.3	Untersuchung der Messwerte	44
5.3	Systemanalyse a-priori	49
5.3.1	Struktur des System	49
5.3.2	Verifikation der Struktur	50
5.3.3	Parameterbestimmung	50
5.4	Vergleich und Zusammenführen der Identifikationsergebnisse	53
5.5	Resultierendes Systemmodell	53
5.5.1	Ebene 1	53
5.5.2	Ebene 2 Antenne	54
5.5.3	Ebene 2 Controller	54
5.5.4	Ebene 3 einzelner Regler	55
6	Reglerentwurf	57
6.1	Kompensationsregler	57
6.1.1	Herleitung des Kompensationsreglers	58
6.1.2	Simulation des geschlossenen Regelkreises in Simulink	61
6.1.3	Betrachtung der Nichtlinearität	65
6.1.4	Betrachtung Vorgabe eines Winkels als Sollposition für das Discovery Board	75
6.1.5	Diskretisierung (Digitalisierung) des Kompensationsreglers	76
6.2	Smith-Predictor	84
6.2.1	Funktionsweise Smith-Prädiktor	84
6.2.2	Simulation	85
6.2.3	Zeitdifferenzen	89
6.2.4	Modellabweichungen	90
6.2.5	Zusammenfassung	90

6.2.6	Codegenerierung.....	90
6.2.7	Simulinkmodell	91
6.3	ASA-Controller	93
6.3.1	Parameterberechnung	93
6.3.2	Simulation.....	94
6.3.3	Diskretisierung	96
6.3.4	Verhalten am Antennenmodell.....	97
6.4	PFC-Regler	98
6.4.1	Funktionsweise des PFC-Reglers	98
6.4.2	Auslegung für IT ₁ -Prozess	101
6.4.3	Simulation.....	102
6.4.4	Zusammenfassung	104
6.5	SPFC-Regler.....	104
6.5.1	Funktionsweise des SPFC-Reglers (Simplified PFC)	104
6.5.2	Simulation.....	105
6.5.3	Zusammenfassung	107
7	SIL und PIL	108
7.1	SIL (Software in the Loop): C-Code der Regler mit Simulink Coder generieren und testen	108
7.2	PIL (Software in the Loop): C-Code der Regler mit Embedded Coder generieren und testen	111
7.3	Analyse des generierten Codes.....	113
8	HMI und Mikrocontroller Kommunikation.....	114
8.1	Systematischer Aufbau der Kommunikation.....	114
8.1.1	Sequenzdiagramm der Kommunikation.....	115
8.1.2	Telegrammaufbau.....	115
8.1.3	Telegrammverarbeitung	116
8.2	Matlab GUIDE	118
9	Test und Validierung.....	121
9.1	Grundkonzept	Fehler! Textmarke nicht definiert.
9.1.1	Test der Reglerentwürfe im Bildbereich	Fehler! Textmarke nicht definiert.
9.1.2	Test der Reglerentwürfe nach der Digitalisierung.....	Fehler! Textmarke nicht definiert.
9.1.3	SIL und PIL	Fehler! Textmarke nicht definiert.
9.1.4	Test der Reglerentwürfe am Modell.....	Fehler! Textmarke nicht definiert.
9.2	Validierung der Ergebnisse	Fehler! Textmarke nicht definiert.
10	Zusammenfassung	125
	Anhang A: Projektmanagement.....	126
A1:	Risikoplan	126

A2: Spezifikationsliste	127
A3: Vorgangsliste	128
Anhang B: Code auf dem Controller	135
Anhang C: Pflichtenheft.....	136
Anhang D: Messungen zur Systemidentifikation	137
D1: Messwerte uninterpretiert	137
D2: Systemanalyse ohne Vorkenntnisse	137
D3: Systemanalyse a-priori.....	150
D4: SIL mit Abtastzeit von 1ms für Reglermodell aus Kapitel 5.5.....	150
D5: PIL mit Abtastzeit von 10ms	154
D6: Validierungstestläufe	158
Glossar	168
Literaturverzeichnis	169
Erklärung	170
Stichwortverzeichnis.....	171

1 Einleitung

1.1 Überblick Aufgabenstellung

Ziel dieser Projektarbeit ist es die Positionsregelung einer Antenne mit unterschiedlichen Reglertypen zu entwickeln und diese in ein STM32F4 Discovery Board zu implementieren. Dabei sollen die folgenden Reglertypen zur Auswahl kommen:

- Kompensationsregler
- Smith-Predictor
- ASA-Controller
- Predictive Function Control (PFC)
- SPFC (Simplified PFC)

Das Antennenmodell besteht aus folgenden Komponenten:

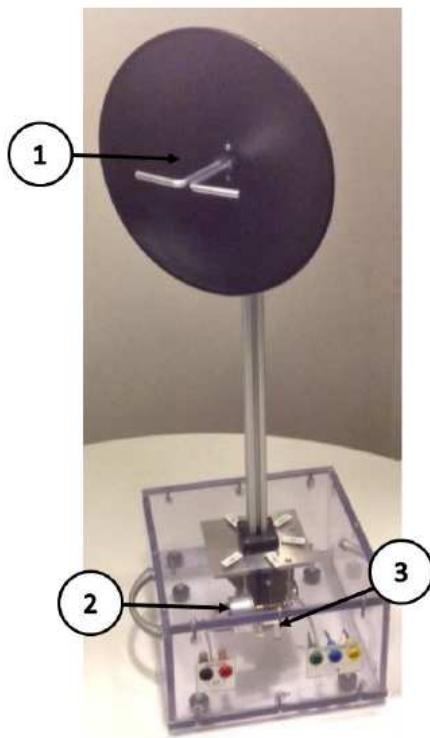


Abbildung 1-1 Antennenmodell

Antenne (siehe Nr. 1): Die Antenne ist direkt mit der Welle des Potentiometers verbunden. Somit entspricht die Antennenposition der Position der Welle, also dem Drehwinkel der Welle.

Gleichstrommotor (siehe Nr. 2): Von diesem Motor existiert kein Datenblatt. Es wird somit angenommen, dass es sich um einen Bürsten-kommutierten permanent erregten Gleichstrommotor handelt.

Potentiometer (siehe Nr. 3): Es handelt sich um ein Drehpotentiometer. Die Spannung am Schleifer fällt oder steigt linear und stetig proportional zum Drehwinkel der Welle.

Getriebe: An den Gleichstrommotor ist ein Getriebe angebracht. Über das Getriebe existieren keine Datenblätter. Es wird als verlustfrei angenommen.

Weitere Hardwarebestandteile sind

- STM32F4 Discovery Kit mit aufsteckbarem Piggy Board
- Serielle Kommunikation über USART mit Treiberbaustein FTDI (FT232)
- H-Brücke

Folgende Software kommt zum Einsatz:

- MATLAB/Simulink
- MATLAB/Simulink Coder
- MATLAB/Embedded Coder
- Embedded Coder Support Package for STM32F4
- Coocox CoIDE
- CMSIS von ARM Cortex

Mit der einzusetzenden Hardware und Software sollen folgende Aufgaben realisiert werden:

- Die Regelstrecke (Antenne) beschreiben, simulieren und testen.
- Mit Model in the Loop unter Simulink den Regelkreis simulieren und testen
- Für den simulierten Regelkreis eine Visualisierung mit Matlab erstellen und testen
- Software in the Loop, d. h. C-Code für Regler mit Simulink Coder generieren und testen.
- Processor in the Loop, d. h. Regler mit dem Simulink Modell der Strecke testen.
- C-Code unter Simulink für die Regler mit Embedded Coder generieren, herunterladen und diesen auf dem Discovery Board implementieren.
- Antenne mit dem STM32F4 Discovery Board regeln.

In dieser Einleitung soll vorab schon mal ein grobes Verständnis für einige der einzusetzenden Hardware und Software Produkte entwickelt werden, bevor dann in den einzelnen Unterkapiteln bei Bedarf eine genauere Betrachtung erfolgt.

richter Brücke. Über einen integrierten Linear Spannungsregler wird die für das Discovery Board benötigte 5V Versorgungsspannung zur Verfügung gestellt. Die benötigten 3V für den Controller werden auf dem Discovery Board selbst erzeugt. Die Versorgungsspannung wird der H-Brücke direkt zur Verfügung gestellt. Da der Motor des Antennen Modells mit max. 12V versorgt werden sollte ergibt sich eine max. Versorgungsspannung für das Piggyback Board von 13,4V.

3.3 Sensoren:

Das Piggyback Board verfügt über 1 HAL Sensor (ACS712) zur Erfassung des momentanen Motorstroms (optional) und einen Potentiometer Anschluss (Poti_A, Poti_B, Poti_C) für die Positionsermittlung der Antenne. Die Potentiometer Anschlüsse sind ebenfalls als 4mm Klinkenbuchsen bestückt. Da der HAL Sensor eine Signalspannung von max. 5V ausgibt wird diese mittels eines Spannungsteilers (R8, R9) auf 3V herabgesetzt. Ein vorgeschalteter Tiefpass Filter dient zur Filterung der Hochfrequenz Anteile aus der Steuerung.

3.4 H- Brücke:

Die H-Brücke besteht aus 2 Halb-Brücken ICs (IR2184). Diese Treiber ICs sind für die Ansteuerung der 4 N-Kanal Mosfets zuständig. Die Integrierte Bootstrap Schaltung versorgt den High Side Fet mit einer höheren Spannung damit dieser geschaltet werden kann. Außerdem ist eine Crossconduction prevention logik sowie eine Dead Time Modul integriert die Kurzschlüsse durch eine fehlerhafte Ansteuerung der Brücke verhindern sollen. Der Motor wird über zwei 4mm Klinkenbuchsen angeschlossen.

3.5 Kommunikation

Für die Kommunikation wird ein FT232RL von der Firma FTDI eingesetzt. Dabei handelt es sich um einen USB zu Uart Converter. Dieser stellt am angeschlossenen PC einen virtuellen Com-Port zur Verfügung über den die Gui mit dem Modell kommunizieren kann.

5 Systemanalyse

In diesem Kapitel wird beschrieben, mit welchem Messaufbau das System ermittelt wird und welche Testsignale dazu warum verwendet werden. Anhand der so entstandenen Messreihen lässt sich das Verhalten des Systems analysieren und die Struktur der Strecke feststellen. Da die Identifikation der Strecke eine zentrale Fehlerquelle darstellt, deren Auswirkungen im weiteren Projektverlauf nur mit viel Aufwand zu korrigieren sind, wird die Identifikation des Systems auf zwei Arten von unterschiedlichen Personen durchgeführt.

Im ersten Verfahren, nachfolgend als „ohne Vorkenntnisse“ benannt, dienen nur die Messreihen als Quelle, ohne dass andere vorliegende Informationen genutzt werden. Daraus werden Struktur und die Parameter der Strecke ermittelt.

Im zweiten Verfahren, nachfolgend als „a-priori“ bezeichnet, werden alle vorhandenen Vorkenntnisse genutzt, um die Struktur der Strecke zu bestimmen. Die Messwerte werden hier im Wesentlichen zur Schätzung der Parameter der Strecke verwendet.

Die Ergebnisse dieser beiden Schätzverfahren werden verglichen und münden in ein gemeinsames Streckenmodell.

5.1 Messaufbau und Auswahl Testsignale

5.1.1 Auswahl Testsignale

Aufgabenstellung ist, die Antenne von einer beliebigen Startposition auf eine Vorgegebene Zielposition zu verfahren. Da es sich um eine mechanische Bewegung senkrecht zur Erdbeschleunigung auf unbewegtem, vibrationsfreiem Untergrund handelt, wird davon ausgegangen, dass das System ohne Stellgrößenaufschaltung stabil in seiner Position verharrt. Dies bestätigt sich nach manueller Bewegung in verschiedene Positionen. Sobald eine Stellgröße ungleich Null aufgeschaltet wird, beginnt die Strecke eine Bewegung, die erst endet, wenn die Stellgröße auf null gesetzt wird.

Daraus und aus der Aufgabenstellung folgt damit, dass der einzig relevante Arbeitspunkt 0V ist.

Durch den Charakter der Strecke erscheint ein Stellgrößensprung als sinnvollstes Testsignal. Um eine sichere Identifikation des Systems sicherzustellen, werden $\pm 2V$, $\pm 4V$, $\pm 6V$, $\pm 8V$ und $\pm 10V$ als Sprunghöhen ausgewählt. Die negativen Sprünge werden nach vorhergehender, positiver Stellgröße gemessen.

5.1.2 Messaufbau

Ein Netzteil versorgt Potentiometer und Motor mit Strom. Der Motor ist über einen Schalter verbunden, um mit diesem einen Sprung generieren zu können. Als Messmittel dient ein Oszilloskop, dass sowohl Motorspannung als auch die Spannung des Potentiometerabgriffs aufzeichnet und auf den Sprung der Motorspannung triggert.

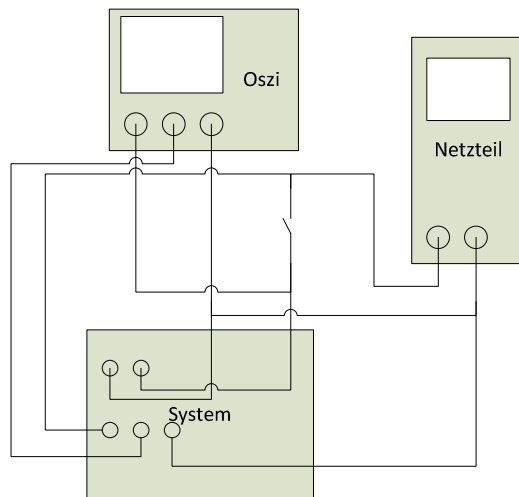


Abbildung 5-1: Messaufbau zur Identifikation; Anschlüsse am Oszilloskop: (v.l.n.r.) Kanal1 Stellgröße, Kanal2 Messwert der Systemantwort, Masse; System: oben: Klemmen für Stellgröße (Motorklemmen); unten: (v.l.n.r.) Masse Messpoti, Mittelabgriff Messpoti, Masse Messpoti

5.2 Systemanalyse ohne Vorkenntnisse

5.2.1 Beschreibung der Regelstrecke des Antennenmodells

In Anlehnung an das Buch „Regelungstechnik für Ingenieure Kapitel 3.7“ wird die Regelstrecke der Antenne hier beschrieben.

Dabei gilt:

R = Ankerwiderstand

J = Trägheitsmoment der rotierenden Massen

$c\Phi_0$ = Erregerfluß

n = Drehzahl der Motorwelle

ω = Winkelgeschwindigkeit Antenne

G = Getriebeübersetzung

Am Gleichstrommotor anliegende Spannung = Eingangsgröße y

Von Antenne zurückgelegter Winkel = Ausgangsgröße φ

Die Ankerinduktivität sei vernachlässigbar klein.

Für die von der Gleichstrommaschine induzierte Spannung gilt $c\Phi_0\omega(s)$.

Somit gilt für den Ankerkreis mit dem Ankerstrom $i(s)$

$$y(s) = i(s) \cdot R + c\Phi_0\omega(s)$$

Der Erregerfluß Φ_0 sei als konstant angenommen. Somit ergibt sich für das elektrische Moment $M_e(s) = c\Phi_0 \cdot i(s)$

Das elektrische Moment $M_e(s)$ ist gleichbedeutend mit dem Antriebsmoment.

Allgemein gilt die Beziehung Antriebsmoment – Widerstandsmoment = Beschleunigungsmoment

In Ruhelage ist das Beschleunigungsmoment 0 Nm und somit ist das Antriebsmoment gleich dem Widerstandsmoment. Das Widerstandsmoment (Gegenmoment) ergibt sich aus dem Massenträgheitsmoment der rotierenden Massen. Dieses Widerstandsmoment muss vom Antriebsmoment überwunden werden, damit sich die Antenne drehen kann. Das Antriebsmoment muss also größer als das Gegenmoment sein. Ist das Antriebsmoment genau so groß wie das Widerstandsmoment (Gegenmoment) findet noch keine Drehbewegung der Antenne statt (stationärer Zustand).

Das Gegenmoment errechnet sich wie folgt: $M_w(t) = J (d\omega(t) / dt)$ bzw. mit dem Differenziationssatz ergibt sich $M_w(s) = J \cdot s \cdot \omega(s) = M_e(s)$

Die zeitliche Winkeländerung der Antenne ist proportional zur Drehzahl des Motors.

Es gilt:

$$d\varphi(t)/dt = n(t) / G$$

$$s \cdot \varphi(s) = n(s) / G \Rightarrow s \cdot \varphi(s) = \frac{1}{G} \cdot \frac{\omega(s)}{2\pi} \Rightarrow \omega(s) = s \cdot \varphi(s) \cdot G \cdot 2\pi$$

Weiterhin gilt: $M_e(s) = c\Phi_0 \cdot i(s) \Rightarrow i(s) = M_e(s) / c\Phi_0 \Rightarrow i(s) = J \cdot s \cdot \omega(s) / c\Phi_0$ Dieses Ergebnis wird in folgende Gleichung eingesetzt.

$$y(s) = i(s) \cdot R + c\Phi_0\omega(s) \Rightarrow y(s) = c\Phi_0\omega(s) + R \cdot J \cdot s \cdot \omega(s) / c\Phi_0$$

$$y(s) = \omega(s) [c\Phi_0 + R \cdot J \cdot s / c\Phi_0] \text{ mit } \omega(s) = s \cdot \varphi(s) \cdot G \cdot 2\pi$$

$$y(s) = s \cdot \varphi(s) \cdot G \cdot 2\pi [c\Phi_0 + R \cdot J \cdot s / c\Phi_0]$$

$$\frac{1}{G2\pi c\Phi_0} \cdot \frac{1}{s} y(s) = \varphi(s) + \frac{R \cdot J \cdot s \cdot \varphi(s)}{c\Phi_0 \cdot c\Phi_0}$$

$$\frac{1}{G2\pi c\Phi_0} \cdot \frac{1}{s} y(s) = \varphi(s) \left[1 + \frac{s \cdot R \cdot J}{(c\Phi_0)^2} \right]$$

Es gilt:

$$Ki = \frac{1}{G2I/c\Phi_0} \text{ und } T1 = \frac{R \cdot J}{(c\Phi_0)^2}$$

Somit ergibt sich:

$$Ki \cdot \frac{1}{s} y(s) = \varphi(s)[1 + sT1]$$

$$G(s) = \frac{\varphi(s)}{y(s)} = \frac{Ki}{s \cdot (1 + sT1)}$$

Somit ergibt sich folgende Differentialgleichung:

$$T1 \cdot \varphi(t)'' + \varphi(t)' = Ki \cdot y(s) \text{ oder } T1 \cdot \varphi(t)' + \varphi(t) = Ki \int y(t) dt$$

Man sieht, dass $T1 \cdot \varphi(t)' + \varphi(t)$ proportional dem zeitlichen Integral der Eingangsgröße y ist: daher die Bezeichnung integrales Verhalten mit Verzögerung 1. Ordnung oder kurz I-T1-Verhalten.

5.2.2 Signalanalyse

Das Potentiometer wurde mit einer festen Spannung von 3 Volt versorgt. Nach Importierung der Messwerte für einen Eingangssprung von 0 auf 2V nach Matlab kann man sich diese mit der folgenden kleinen Matlab Befehlssequenz anschauen lassen.

```
plot (Tsec,Ux,Tsec,Uw)
grid on
legend('Y1','X1',4)
```

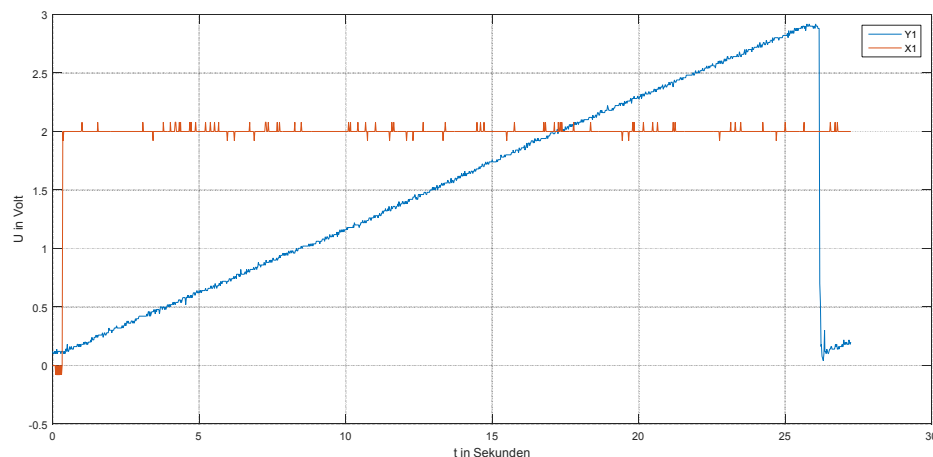


Abbildung 5-2 Potentiometer Ausgangsspannung bei einem Eingangssprung von 0 auf 2 V

Die Messwerte sind zuvor zurecht geschnitten worden. Wenn das Potentiometer seinen mechanischen Drehwinkel von 360° erreicht hat, läuft die Rampe wieder von Null hoch. Die restlichen Messwerte sind gelöscht worden.

Man erkennt, dass beide Signale verrauscht sind.

Daher soll in diesem Kapitel zunächst die Sprungantwort mittels einer Signalanalyse betrachtet werden. Für den Eingangssprung wird keine Signalanalyse durchgeführt. Im Anhang befinden sich alle Matlab Programme für eine Betrachtung bei einem Eingangssprung von 0 auf 2 Volt. Die Matlab Programme für die übrigen Eingangssprünge sind nicht im Anhang dargestellt, da diese nach dem gleichen Prinzip programmiert wurden und sich dabei nur die entsprechenden Werte unterscheiden.

Untersuchung Leistungsdichtespektrum

Es wird das Periodogramm der Potentiometer-Ausgangsspannung untersucht. Ein stärkeres Rauschen entsteht bei 3.008 Hz.

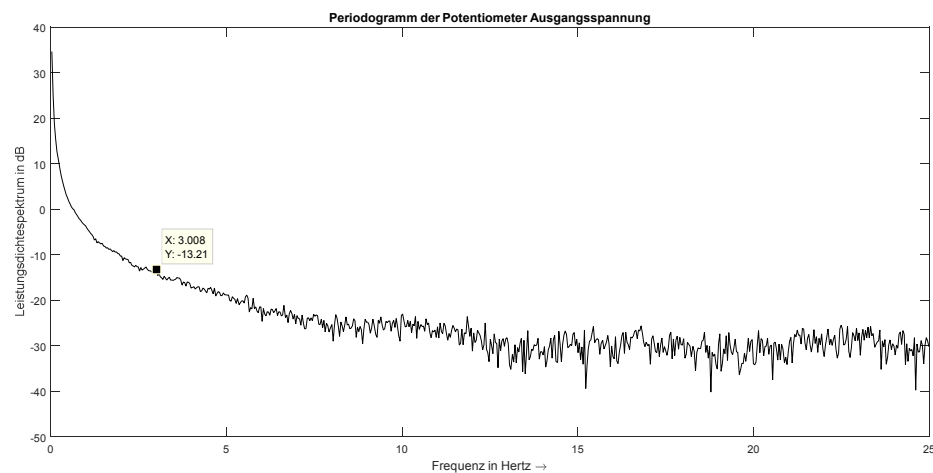


Abbildung 5-3 Periodogramm der Potentiometer Ausgangsspannung

Untersuchung Amplitudenspektrum

Das Amplitudenspektrum der Potentiometer Ausgangsspannung ist im Folgenden dargestellt.

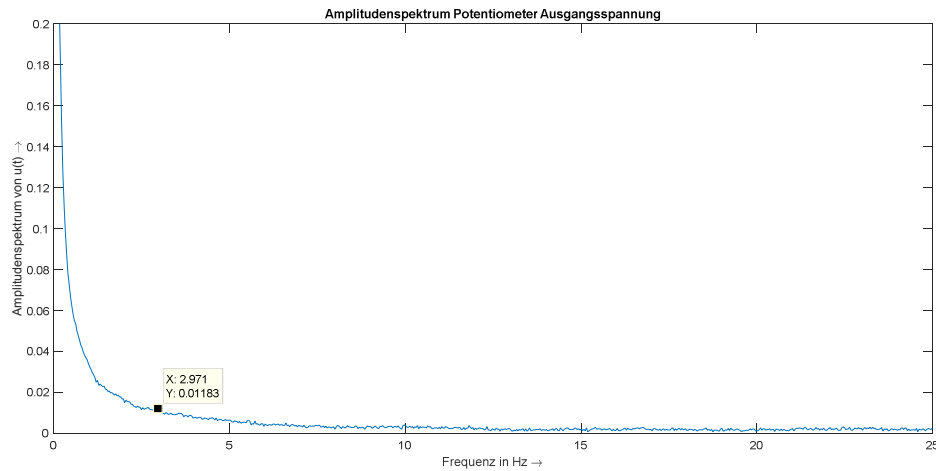


Abbildung 5-4 Amplitudenspektrum der Potentiometerausgangsspannung

Ein Stärkeres Rauschen setzt hier ebenfalls bei ungefähr 3 Hz ein. Die Amplitude ist mit 0,01183V ebenfalls sehr gering.

Filterentwurf

Man muss also nun einen Tiefpass für die Potentiometer Ausgangsspannung mit einer Grenzfrequenz von 3Hz entwickeln. Es handelt sich um ein aperiodisch zeitbegrenztes Signal.

Die Tiefpassentwicklung ist mit dem Matlab fdatool durchgeführt worden. Es sind folgende Kennwerte eingestellt worden:

Grenzfrequenz: 3Hz

Abtastfrequenz: 50Hz

FIR Filter mit einem Kaiser Fenster

Ordnung 10 also 11 Koeffizienten

Auf eine detaillierte Erläuterung zum Filterentwurf wird hier verzichtet und stattdessen auf die entsprechende Literatur verwiesen. Im nächsten Kapitel wird der entsprechende Filter dann auf die verrauschten Signale angewendet.

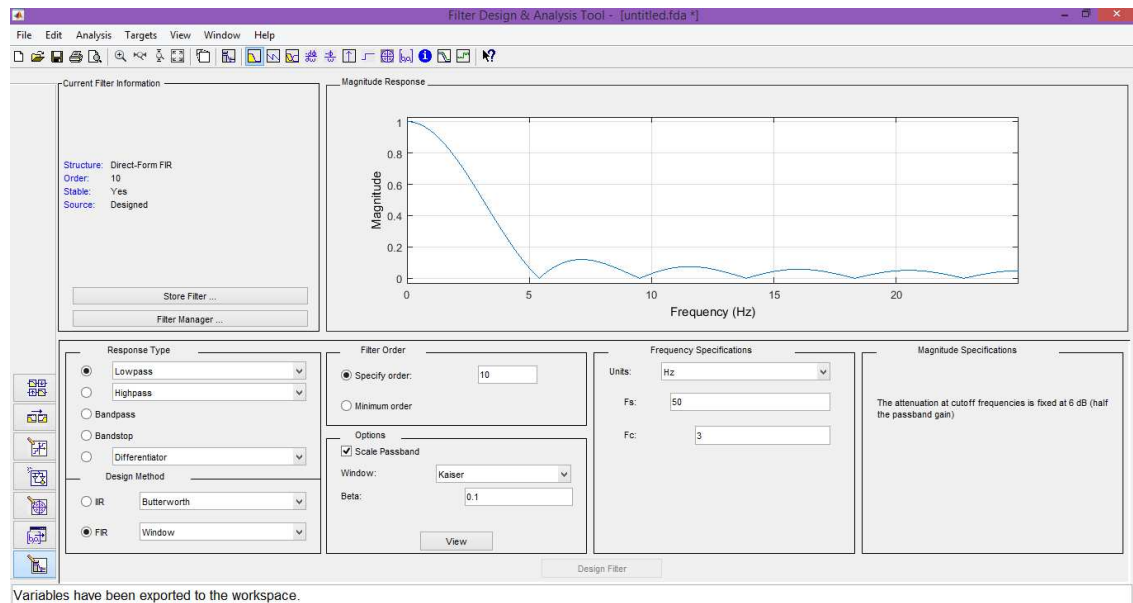


Abbildung 5-5 Filterentwurf mit dem fdatool von Matlab

5.2.3 Untersuchung der Messwerte

In diesem Kapitel werden nun die Messwerte untersucht. Dabei werden sowohl die verrauschten Signale als auch die gefilterten Signale untersucht. Die ermittelten Werte werden miteinander verglichen.

Eingangssprung von 0 auf 2V:

Bei der Betrachtung der Kennlinien erkennt man ebenfalls, dass es sich um eine Regelstrecke mit I Verhalten und einer schwachen Verzugszeit T_1 handelt.

Die Filterkoeffizienten sind aus dem fdatool an den Matlab Workspace übertragen worden und dann auf die verrauschten Signale angewendet worden.

Laut Datenblatt hat das Potentiometer einen wirksamen elektrischen Drehwinkel von 350° . Man erkennt, dass bei ungefähr 25,72 Sekunden sich die Antenne von 0° auf 350° gedreht hat. Danach findet keine weitere Steigung der Potentiometer-Ausgangsspannung statt. Nach ungefähr 26,1 Sekunden ist der mechanische Drehwinkel des Potentiometers von 360° erreicht. Ungefähr von diesem Wert an wird die Potentiometer-Ausgangsspannung wieder von ungefähr 0V auf ungefähr 3V nach einer Rampe hochlaufen. (I-Verhalten), wobei die 3V nie ganz erreicht werden. Es stellt sich eine Abweichung von ungefähr $3V - 2,9V = 0,1V$ ein.

Die Werte K_I und T_1 sind mit der gefilterten Sprungantwort ermittelt worden; ebenso der Wert für die Abweichung. Diese Parameterwerte ließen sich aufgrund der Filterung exakter ermitteln. Die ungefähre Zeit bis zum wirksamen elektrischen und mechanischen Drehwinkel lässt sich durch das geglättete Signal nicht so exakt ermitteln, wie durch das ungefilterte Signal, daher ist sich bei diesem Wert für eine Bestimmung aus dem verrauschten Signal entschieden worden.

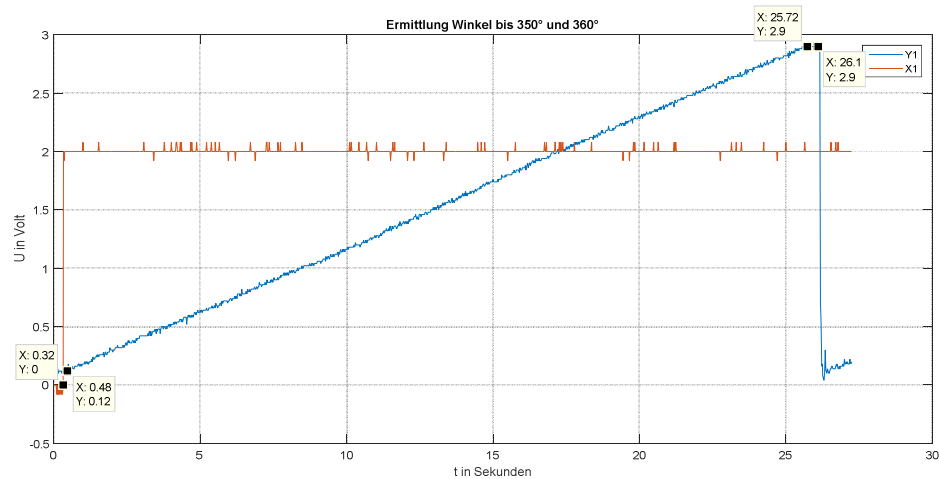


Abbildung 5-6 Ermittlung Winkel bis 350° und 360° bei einem Eingangssprung von 0 auf 2 V

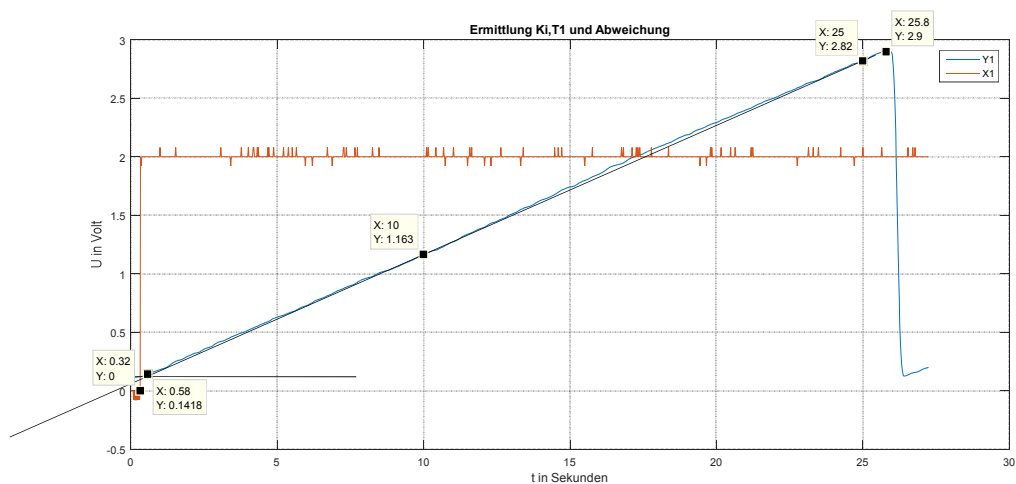


Abbildung 5-7 Ermittlung K_i , T_1 und Abweichung bei einem Eingangssprung von 0 auf 2 V

Ermittlung K_I : $(1,657 \text{ V} / 15 \text{ sec.}) / 2 \text{ V} = 0,05523 \text{ sec}^{-1}$.

Ungefähre Ermittlung der Zeitkonstanten T_1 :

$$T_1 = 0,58 \text{ sec.} - 0,32 \text{ sec.} = 0,26 \text{ sec.}$$

Die Kennwerte der Regelstrecke bei einem Eingangssprung von 0 auf 2 Volt lauten somit:

Ungefähre Zeit bis zum wirksamen elektrischen Drehwinkel

$$\text{von } 350^\circ : 25,72 \text{ sec.} - 0,32 \text{ sec.} = 25,4 \text{ sec.}$$

Ungefähre Zeit bis zum mechanischen Drehwinkel

von 360° : $26,1 \text{ sec.} - 0,32 \text{ sec.} = 25,78 \text{ sec.}$

Ungefähre Ermittlung Abweichung: $3\text{V} - 2,9\text{V} = 0,1\text{V}$

Ungefähre Ermittlung K_I : $(1,657 \text{ V} / 15 \text{ sec.}) / 2 \text{ V} = 0,05523 \text{ sec}^{-1}$.

Ungefähre Ermittlung Zeitkonstante T_1 : $0,58 \text{ sec.} - 0,32 \text{ sec.} = 0,26 \text{ sec.}$

Die Sprungantwort im Zeitbereich lautet somit³:

$$y(t) = K_I * y_0 \left[t - T_1 \left(1 - e^{-\frac{t}{T_1}} \right) \right]$$

$$x(t) = 0,05523 \text{ sec.}^{-1} * 2\text{V} \left[t - 0,26 \text{ sec.} \left(1 - e^{-\frac{t}{0,26 \text{ sec.}}} \right) \right]$$

Abschließend als Probe der Vergleich zwischen der modellierten und der realen Sprungantwort:

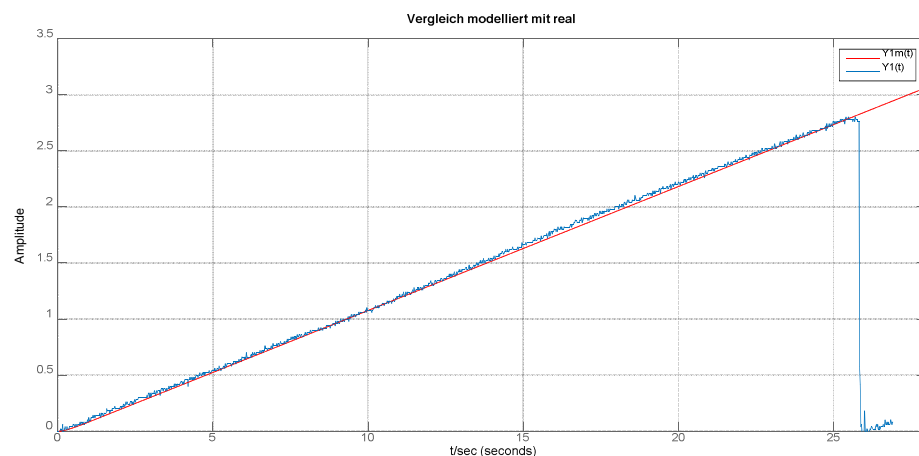


Abbildung 5-8 Vergleich modellierter Sprungantwort mit realer Sprungantwort

Bei den übrigen Messwerten ist genau nach dem gleichen hier zuvor beschriebenen Verfahren vorgegangen worden, so dass das hier in der Dokumentation nicht mehr erläutert werden muss. Alle hierzu erforderlichen Matlab Programme und Diagramme befinden sich zur Nachvollziehbarkeit im Anhang. Die weiteren Parameter der Messwerte sind nun in einer Tabelle zusammen gefasst.

Eingangssprung in V	Abweichung in V	K_I	T_1	Zeit bis	Zeit bis
---------------------	-----------------	-------	-------	----------	----------

³ Auf eine Herleitung wird hier verzichtet und auf die Literatur verwiesen, z.B. Regelungstechnik für Ingenieure

		in sec^{-1}	in sec	350° in sec.	360° in sec.
0 auf 2V	$3\text{V} - 2,9\text{V} = 0,1\text{V}$	0,05523	0,26	25,4	25,78
0 auf 4V	$3\text{V} - 2,903\text{V} = 0,097\text{V}$	0,06154	0,24	11,63	11,79
0 auf 6V	$3\text{V} - 2,899\text{V} = 0,101\text{V}$	0,06248	0,196	7,58	7,68
0 auf 8V	$3\text{V} - 2,901\text{V} = 0,099\text{V}$	0,06421	0,152	5,476	5,544
0 auf 10V	$3\text{V} - 2,896\text{V} = 0,104\text{V}$	0,0638	0,152	4,356	4,424
Mittelwert	0,1002 V	0,061452	0,2		

Zur Kontrolle sind zusätzlich noch zwei negative Sprünge aufgenommen worden.

Das Potentiometer ist weiterhin mit einer festen Potentiometer Spannung von 3 Volt versorgt worden.

Da die Aufnahme von Eingangssprüngen von 0 auf -10V und von 0 auf -2V technisch nicht möglich war, sind auch hier Sprünge von 0 auf 2 Volt und von 0 auf 10 Volt aufgenommen worden. Allerdings sind beim Motor Plus und Minus Anschlüsse vertauscht worden, so dass +10 Volt am Minus-Anschluss angeschlossen ist und 0 am Plus-Anschluss. Die Potentiometer Ausgangsspannung geht somit von 2,9 Volt (350°) auf 0 Volt (0 Grad), da das Potentiometer in die falsche Richtung läuft.

In der folgenden Tabelle sind die beiden Werte gegenüber gestellt. Die aufgenommenen Diagramme befinden sich im Anhang.

Eingangssprung in V	Abweichung in V	K_I in sec^{-1}	T1 in sec	Zeit bis 360° in sec.	Zeit bis 350° in sec.
0 auf 2V	$3\text{V} - 2,911\text{V} = 0,089\text{V}$	0,0585	0,16	23,58	24,2
0 auf 10V	$3\text{V} - 2,906\text{V} = 0,094\text{V}$	0,0665	0,184	4,188	4,276

Beim K_I sieht man keine wesentlichen Unterschiede zu den vorherigen Werten. Die Werte für die Zeitkonstante T1 können ebenfalls so akzeptiert werden, auch wenn diese nicht ganz deckungsgleich mit den Werten für die entsprechenden positiven Sprünge sind. Der Mittelwert der Abweichung liegt mit 0,0915V etwas unter den vorherigen Messwerten. Dies könnte auf Messungenauigkeiten zurückzuführen sein. Es wird daher für die Abweichung ein Wert von 0,1 V angesetzt.

Insgesamt ergibt sich also folgende Übertragungsfunktion:

$$G_s(s) = \frac{K_I}{s(1 + sT1)} = \frac{0,061452 \text{ sec.}^{-1}}{s(1 + s0,2 \text{ sec.})}$$

An Hand der durchgeführten Untersuchungen konnte belegt werden, dass es sich um eine IT-1 Strecke handelt. Ein einfaches Streckenmodell ist hier aus Simulink dargestellt.

Zusätzlich ist die Sprungantwort für einen Eingangssprung auf 10 Volt bei $t = 0$ Sekunden aufgenommen worden. Die Kennlinie bestätigt die gefundene Übertragungsfunktion und das IT-1 Verhalten.

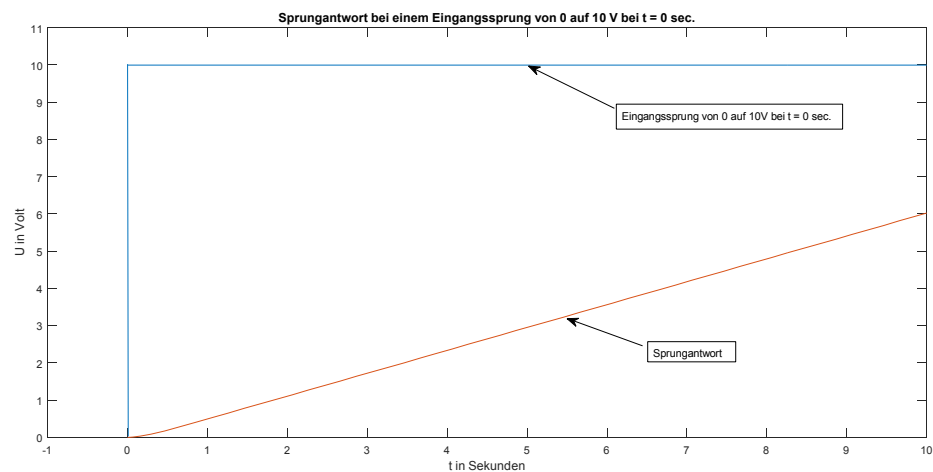
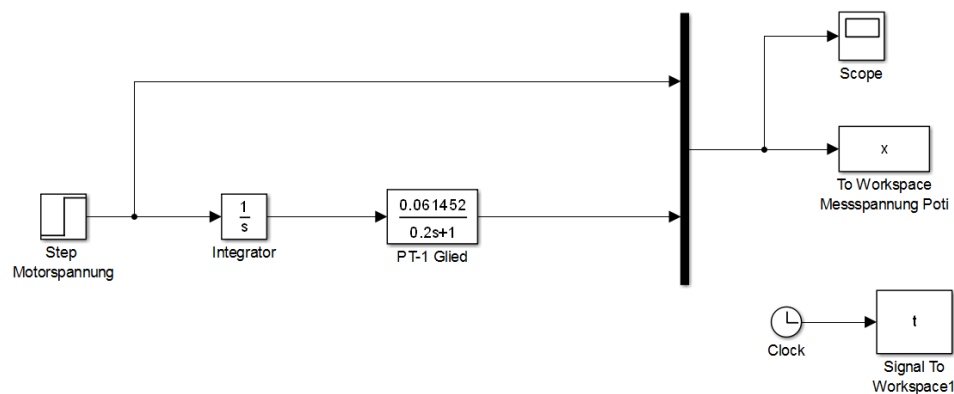


Abbildung 5-9 Streckenmodell und Sprungantwort bei einem Eingangssprung von 0 auf 10 Volt

Abschließend sei noch folgende Bemerkung erwähnt:

Das Potentiometer ist hier mit einer festen Versorgungsspannung von 3 Volt versorgt worden. Diese wurde deswegen gewählt, da der Analog Digital Wandler des Discovery Boards nicht mehr als 3 Volt wandeln kann.

5.4 Vergleich und Zusammenführen der Identifikationsergebnisse

Beide Identifikationsansätze liefern eine IT_1 -Strecke. Für K_1 wurden 0,0203 bzw. (nach Kap. 5.3.3 bereinigt) 0,02038 ermittelt. Diese Abweichung ist so gering, dass das arithmetische Mittel gewählt wird.

Die Werte für T_1 schwanken stärker. Ein Verfahren liefert einen gewählten Wert von 0,12 und das andere einen arithmetischen Mittelwert von 0,1424. In letzterem Wert ist ein Ausreißer enthalten. Ohne diesen ergibt sich ein Wert von 0,128. Dieser ergibt aufgerundet 0,13, was ungefähr den Mittelwert der ermittelten Werte beider Ansätze entspricht. Daher wird dieser Wert übernommen. Bei einem zwischen 0..1 skalierten Ausgang ergibt sich somit folgende Übertragungsfunktion:

$$G_S = \frac{0,0203}{0,13*s^2 + s} \text{ (ohne Einheiten)}$$

Es wurde festgelegt, dass der Systemausgang ein Winkel zwischen 0 und 360° ist.

$$G_S = \frac{0,0203}{0,13*s^2 + s} * 360 = \frac{7,308}{0,13*s^2 + s} \text{ (ohne Einheiten)}$$

5.5 Resultierendes Systemmodell

Das Systemmodell ist zur besseren Übersicht in mehrere Ebenen gegliedert. Diese werden in Matlab Simulink als Subsystems modelliert. Die Ein- und Ausgangssignale entsprechen denen des realen Aufbaus.

5.5.1 Ebene 1

Auf oberster Ebene sind das Controllerboard und das Antennenmodell. Die Schnittstellen entsprechen dem Hardwareaufbau. Einzige Abweichung ist die Referenz-/Betriebsspannung des AD-Wandlers bzw. Potentiometers die zur Verdeutlichung, das es sich um die gleiche Spannung handelt im Modell von außen zugeführt wird, im Aufbau aber von dem Piggyback-Board bereitgestellt wird.

Der Controller hat als Eingänge den Sollwert (in °), das Feedbacksignal des Boards (als Potentiometerspannung in V), die Referenzspannung des AD-Wandlers (entspricht Betriebsspannung des Potentiometers, in V) und einen Enum zur Auswahl des aktiven Regelalgorithmus. Als Ausgang liefert der Block eine Spannung als Stellgröße.

Das Antennenmodell (Antenne) hat als Eingang die Motorspannung in V und die Betriebsspannung des Potentiometers in V. Als Ausgang liefert die Antenne eine Spannung, die proportional zum Winkel ist.

Zur Aufbereitung für die Visualisierung wird die Ausgangsspannung des Potentiometers in einen Winkel umgerechnet.

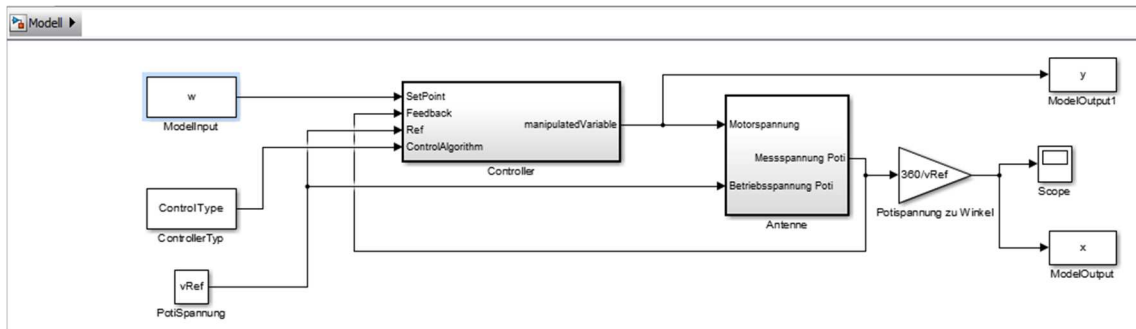


Abbildung 5-11: Resultierendes Systemmodell Ebene 1

5.5.2 Ebene 2 Antenne

Im Block Antenne wird zunächst mittels des in Kapitel 5.4 erstellten Systemmodells die Antwort des Systems für 1 V Potispannung ermittelt. Damit entspricht 1V einer Umdrehung. Dadurch steht vor dem Komma die Zahl der Umdrehungen und hinter dem Komma der Bruchteil einer Umdrehung. Somit kann das Singleturn-Verhalten des Potentiometers einfach nachgebildet werden, indem von dem errechneten Ausgangswert der auf ganze Zahlen abgerundete Wert subtrahiert wird. Damit ist der Sprung des Potentiometers am Ende des Messbereichs abgebildet. Anschließend wird die Antwort noch auf die tatsächliche Potentiometerspannung skaliert.

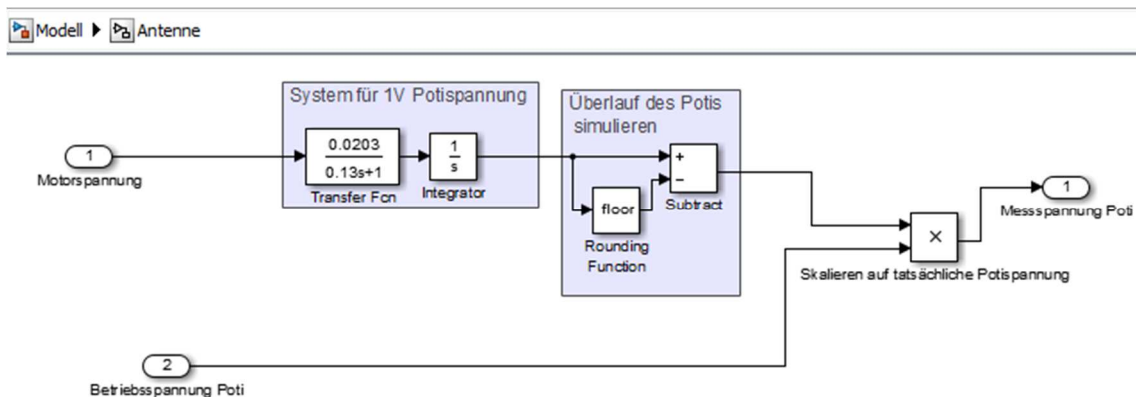


Abbildung 5-12: Resultierendes Systemmodell Ebene 2 Antenne

5.5.3 Ebene 2 Controller

Die Ebene Controller orientiert sich in ihrer Struktur am Klassendiagramm der Controllersoftware.

Im Controller wird zunächst die Potentiometerspannung abstrahiert und das Feedbacksignal zu einem Winkel umgerechnet. Dies erledigen im Klassenmodell die Klassen ADC und Position. Durch diese Abstraktion sind die Reglerentwürfe unabhängig von der tatsächlich verwendeten Potentiometerspannung. Stellt sich bei der Hardwareinbetriebnahme heraus, dass eine höhere oder niedrigere Potentiometerspannung vorteilhaft ist, muss an den Reglerparametern nichts verändert werden. Ebenso falls die

Controllerhardware durch eine andere ersetzt wird, die mit anderen Spannungsbereichen arbeitet (z.B. $\pm 10V$ bei einer SPS).

Das so vorverarbeitete Feedbacksignal und der Sollwert werden auf die einzelnen Regler geleitet. Diese bestimmen die Stellgröße. Über einen Auswahleingang wird bestimmt, welcher Regler die aktive Stellgröße liefert. Im Mikrocontroller wird nur der aktive Regler ausgeführt, in Matlab aus Gründen der einfacheren Implementierung alle und nur das Signal des aktiven Reglers ausgewertet.

Das Ausgangssignal des aktiven Reglers wird dann noch auf einstellbare Schranken begrenzt. Dies simuliert die im realen Aufbau durch die Versorgungsspannung begrenzte Stellgröße. Dieser Block ist außerhalb der Regler, da er auch in der Controllersoftware nachgeschaltet ist. Dies vereinfacht wiederum die Portierung der Reglerentwürfe auf eine andere Controllerhardware mit anderen Spannungsbereichen.

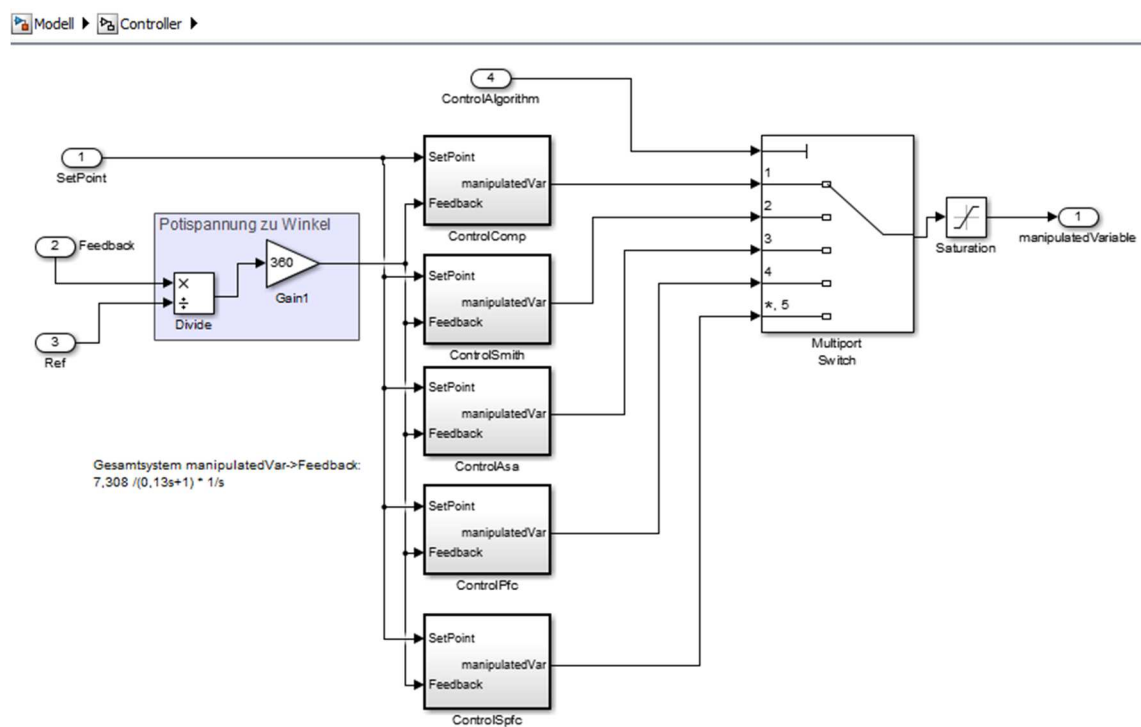


Abbildung 5-13: Resultierendes Systemmodell Ebene 2 Controller

5.5.4 Ebene 3 einzelner Regler

Der einzelne Regler hat als Eingänge Soll- und Istwert und als Ausgang die Stellgröße. Damit liegen für den Reglerentwurf lehrbuchhafte Bedingungen vor.

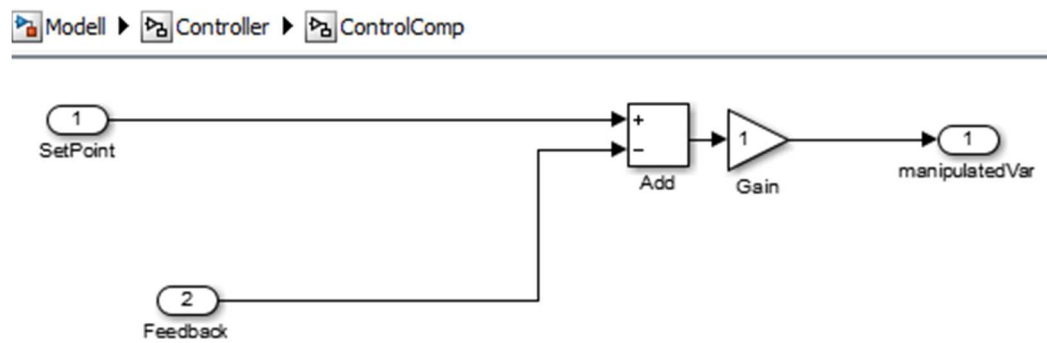


Abbildung 5-14: Resultierendes Systemmodell Ebene 3 einzelner Regler (Abbildung zeigt symbolhaft einen einfachen P-Regler, keinen im Rahmen dieser Arbeit entworfenen Regler)

6 Reglerentwurf

6.1 Kompensationsregler

Als Regelkreis wird ein geschlossener Wirkkreis bezeichnet, der durch fortlaufende Anpassung einer Stellgröße eine Regelgröße so ändert, dass sie eine möglichst geringe Differenz zu einem vorgegebenen Sollwert aufweist. Regelkreise können in den Regler und die Regelstrecke untergliedert werden. Die Regelstrecke ist das System, das geregelt wird.

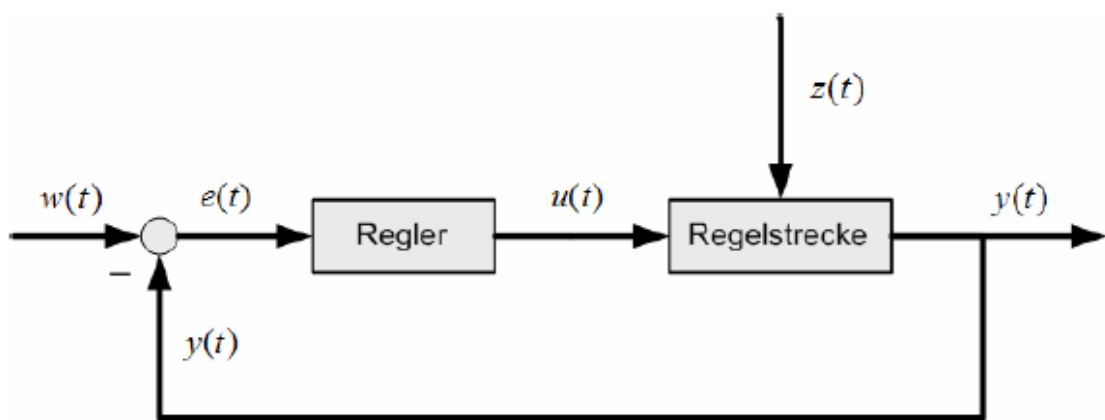


Abbildung 6-1 Grundstruktur eines Regelkreises

$w(t)$ = Führungsgröße, Sollwert

$y(t)$ = Regelgröße, Istwert

$u(t)$ = Stellgröße (Reglerausgang)

$e(t)$ = Regeldifferenz

$z(t)$ = Störgröße

Für jede elektronische Regelung muss die Regelgröße zuerst in ein elektrisches Signal, das Regelsignal $\tilde{y}(t)$, umgewandelt werden, bevor eine Regeldifferenz zum Sollwert ermittelt wird. Auch die Stellgröße wirkt nicht direkt auf die Regelstrecke ein. Meist ist ein verstärkendes oder energiewandelndes Element zwischengeschaltet, das als Stellglied bezeichnet wird. Der Ausgang des Reglers wird somit zum Stellsignal $\tilde{u}(t)$.

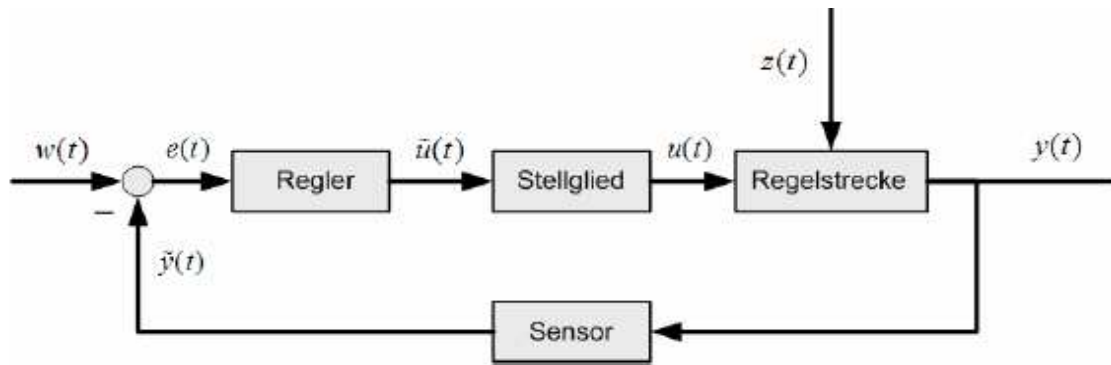


Abbildung 6-2 Erweiterte Grundstruktur des Regelkreises

6.1.1 Herleitung des Kompensationsreglers

Der Kompensationsregler gehört zu den modellbasierten Reglern, die das mathematische Streckenmodell als Bestandteil des Reglers enthalten.

Sind die Übertragungsfunktion der Strecke $G_S(s)$ und die gewünschte Übertragungsfunktion $G_M(s)$ des geschlossenen Regelkreises gegeben, wird der Regler nach folgender Übertragungsfunktion eingestellt.

$$G_R(s) = \frac{1}{G_S(s)} \cdot \frac{G_M(s)}{1 - G_M(s)}$$

Gleichung 6.1.1.1

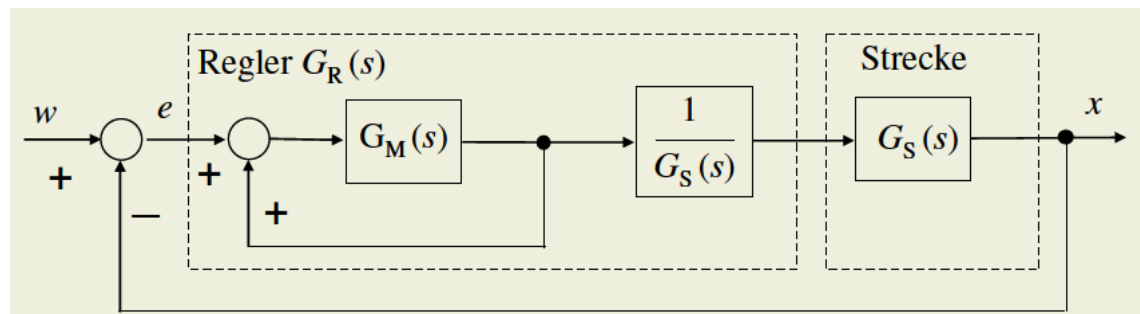


Abbildung 6-3 Wirkungsplan des Regelkreises mit Kompensationsregler

Aus Gleichung 6.1.1.1 folgt, dass die Strecke keine Pol- und Nullstellen in der rechten s -Halbebene besitzen darf. Ebenso darf der Term $1 - G_M(s)$ keine Polstellen in der rechten s -Halbebene besitzen. Sind diese Bedingungen nicht erfüllt, wird der Regler und damit der Regelkreis instabil.

Das gewünschte Verhalten $G_M(s)$ sei hier nun wie folgt definiert:

$$G_{M(s)} = \frac{1}{1 + sT_M}$$

Man erkennt also, dass es sich um ein PT-1 Verhalten handelt. Es wird $T_M = 0,2$ Sek. gewählt, damit der Sollwert der Regelung nach ungefähr einer Sekunde erreicht werden kann. Im Kapitel Systemanalyse ohne Vorkenntnisse wurde herausgefunden, dass es sich bei der Regelstrecke um eine IT-1 Strecke handelt.

$$G_s(s) = \frac{K_I}{s(1 + sT_1)} = \frac{0,061452 \text{ sec.}^{-1}}{s(1 + s0,2 \text{ sec.})}$$

Somit kann der Kompensationsregler wie folgt berechnet werden.

Die Strecke besitzt keine Pole und Nullstellen in der rechten s-Halbebene. Die Übertragungsfunktion des Kompensationsreglers wird nun wie folgt berechnet.

$$G_{R(s)} = \frac{1}{G_{s(s)}} \cdot \frac{G_{M(s)}}{1 - G_{M(s)}} \Rightarrow G_{R(s)} = \frac{1}{\frac{0,061452 \text{ sec.}^{-1}}{s(1 + s0,2 \text{ sec.})}} \cdot \frac{\frac{1}{1 + s0,2 \text{ sec.}}}{1 - \frac{1}{1 + s0,2 \text{ sec.}}}$$

Man sieht, dass der Term $1 - G_{M(s)}$ ebenfalls keine Polstellen in der rechten s-Halbebene besitzt. Regler und Regelkreis müssten also somit stabil sein. Nach weiteren Umformungen ergibt sich für den Kompensationsregler:

$$G_{R(s)} = \frac{s(1 + s0,2 \text{ sec.})}{0,061452 \text{ sec}^{-1}} \cdot \frac{\frac{1}{1 + s0,2 \text{ sec.}}}{1 - \frac{1}{1 + s0,2 \text{ sec.}}} \Rightarrow$$

$$G_{R(s)} = \frac{s(1 + s0,2 \text{ sec.})}{0,061452 \text{ sec}^{-1}} \cdot \frac{\frac{1}{1 + s0,2 \text{ sec.}}}{\frac{1 + s0,2 \text{ sec.} - 1}{1 + s0,2 \text{ sec.}}} \Rightarrow$$

$$G_{R(s)} = \frac{s(1 + s0,2 \text{ sec.})}{0,061452 \text{ sec}^{-1}} \cdot \frac{1}{s0,2 \text{ sec.}} \Rightarrow G_{R(s)} = \frac{s(1 + s0,2 \text{ sec.})}{s0,0122904}$$

$$G_{R(s)} = \frac{(1 + s0,2 \text{ sec.})}{0,0122904} \Rightarrow G_{R(s)} = 81,364 + s16,273 \text{ sec.} \Rightarrow$$

$$G_{R(s)} = \mathbf{81,364(1 + s0,2 \text{ sec.})} = K_{PR}(1 + sT_v)$$

Somit erkennt man, dass es sich bei dem Kompensationsregler um einen **PD Regler** handelt. Dieser PD Regler hat eine Verstärkung K_{PR} von 81,364 und eine Vorhaltzeit T_v von 0,2 Sekunden. Die Vorhaltzeit ist identisch mit der Zeitkonstanten T_1 der IT-1 Strecke.

Abschließend wird das Bode Diagramm des aufgeschnittenen Regelkreises dargestellt:

$$G_0(s) = G_{R(s)} \cdot G_{s(s)} = 81,364(1 + s0,2 \text{ sec.}) \cdot \frac{0,061452 \text{ sec.}^{-1}}{s(1 + s0,2 \text{ sec.})}$$

$$G_{0(s)} = G_{R(s)} \cdot G_{S(s)} = 81,364(1 + s0,2sec.) \cdot \frac{0,061452 sec.^{-1}}{s(1 + s0,2sec.)}$$

$$G_{0(s)} = G_{R(s)} \cdot G_{S(s)} = 81,364 \cdot \frac{0,061452 sec.^{-1}}{s} = \frac{5}{s \cdot sec.}$$

$$G_{0(s)} = G_{R(s)} \cdot G_{S(s)} = \frac{1}{G_{S(s)}} \cdot \frac{G_{M(s)}}{1 - G_{M(s)}} \cdot G_{S(s)}$$

$$\frac{G_{M(s)}}{1 - G_{M(s)}} = \frac{\frac{1}{1 + sT_M}}{1 - \frac{1}{1 + sT_M}} = \frac{\frac{1}{1 + sT_M}}{\frac{1 + sT_M - 1}{1 + sT_M}} = \frac{1}{sT_M} = \frac{5}{s \cdot sec.}$$

Man erkennt, dass die Übertragungsfunktion des aufgeschnittenen Kreises $G_{0(s)}$ unabhängig von der Strecke ist. Weiterhin handelt es sich bei der Übertragungsfunktion des aufgeschnittenen Kreises um ein reines I Verhalten.

```
s=tf('s');
Fs=5/s;
bode(Fs)
```

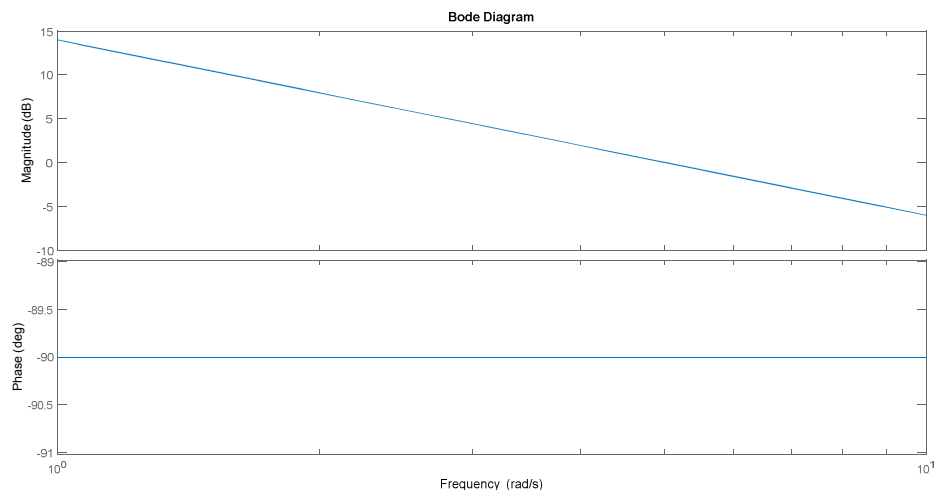


Abbildung 6-4 Bodediagramm des aufgeschnittenen Regelkreises

Wie nicht anders zu erwarten war, ist der Regelkreis stabil. Die Phasenreserve beträgt konstant 90°. Somit ist ebenfalls die oben getätigte Stabilitätsbetrachtung, bei der man Nullstellen und Pole betrachtete, validiert worden.

Im nächsten Abschnitt wird nun der geschlossene Regelkreis in Simulink betrachtet.

6.1.2 Simulation des geschlossenen Regelkreises in Simulink

Im Kapitel 4 Systemanalyse wurde festgestellt, dass es sich bei der Regelstrecke um eine IT-1 Strecke mit $K_I = 0.061452 \text{ sec.}^{-1}$ und einer Zeitkonstante T_1 von 0,2 sec. handelt und sich diese aus folgenden Komponenten zusammensetzt.

- Antenne mit I-Verhalten
- Potentiometer mit P-Verhalten
- Gleichstrommotor mit PT-1 Verhalten
- Getriebe mit P-Verhalten

Die maximale Ausgangsspannung des Potentiometers beträgt 2,9V. Die Antenne soll Positionen von 0 bis 350° anfahren. Somit liefert das Potentiometer bei 1° eine Spannung von 0,00829 Volt. Es wird vorerst ein kontinuierlicher Regelkreis in Simulink entworfen.

Dabei erfolgt die Regelung noch folgendem Prinzip:

Die Antennenposition wird in Matlab in Winkelgrad vorgegeben. Es handelt sich somit um einen Eingangssprung in Winkelgrad.

Das Potentiometer wird zwar mit 3V versorgt, liefert aber nur eine maximale Ausgangsspannung von 2,9V, so dass die Umrechnung der Sprungantwort in Winkelgrad in eine äquivalente Spannung mit 2,9V erfolgt.

Der Spannungswert in Volt kann später bei der realen Regelung über eine serielle Schnittstelle von Matlab an den im Controller implementierten Regler gegeben werden und liegt dort entsprechend digital vor. Bei dieser Lösung wird der Spannungswert auf den PD Regler in Simulink gegeben.

Die Potentiometer-Ausgangsspannung wird als Istwert zurückgeführt und es wird die Regeldifferenz für den PD Regler gebildet. Hier geschieht dies in Simulink, in der realen Anlage geschieht dies im Discovery Board. Die Potentiometer-Ausgangsspannung wird hierzu AD gewandelt. Die AD gewandelte Potentiometer Ausgangsspannung wird zusätzlich über die serielle Schnittstelle zurück nach Matlab gegeben. Dort wird diese zurück in einen Winkel umgerechnet und zur Anzeige gebracht.

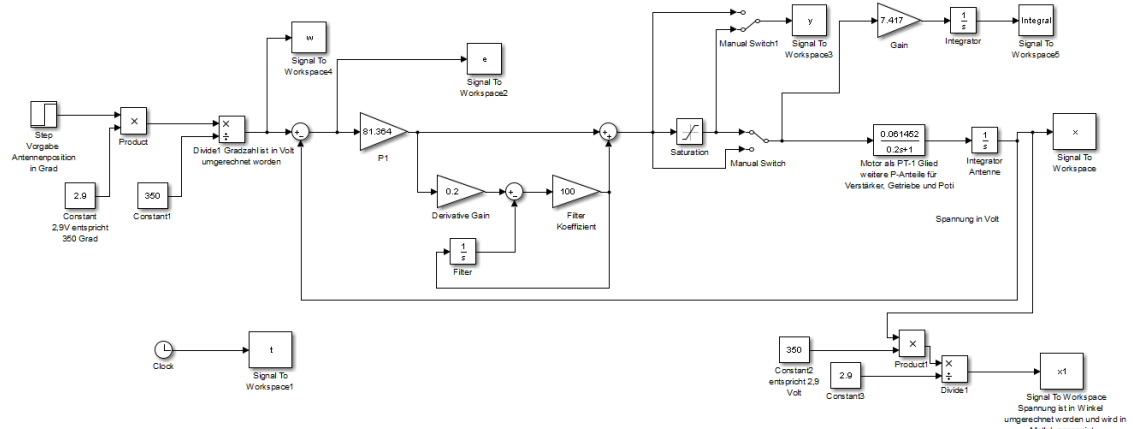


Abbildung 6-5 analoge kontinuierliche Regelung in Simulink

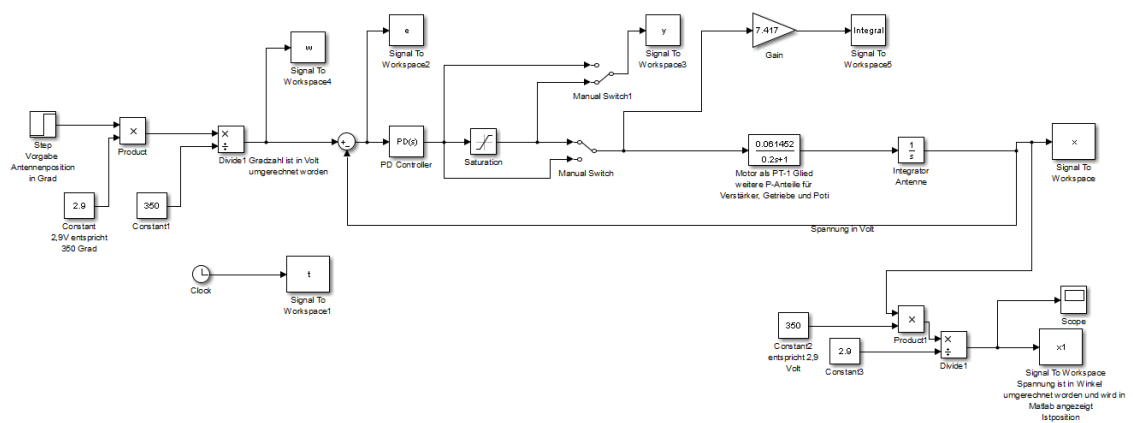
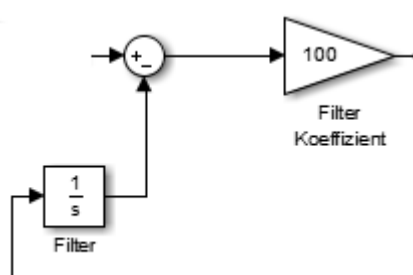


Abbildung 6-6 analoge kontinuierliche Regelung in Simulink alternativ

Abbildung 6.1.5 spiegelt Abbildung 6.1.6 mit der gleichen Funktionalität wieder, jedoch ist die Regelung hier nicht mit der von Matlab vorgegebenen PID Control Box vollzogen worden, sondern durch die folgende Struktur.



$$G(s) = \frac{100}{1 + 100 \frac{1}{s}} = \frac{100}{\frac{s + 100}{s}} = \frac{100 \cdot s}{s + 100} = \frac{1}{1 + 0,01s} \cdot s$$

Bei 100 handelt es sich um von Matlab vorgegebene Filterkoeffizienten. Dem D Anteil⁴ ist somit ein geringer Filter in Form eines PT-1 Gliedes vorgeschaltet worden. Dadurch können Signalstörungen unterdrückt werden. Weiterhin wird dadurch aus dem PD Regler ein PD-T1 Regler. Der D-Anteil sorgt beim PD Regler für einen unendlich steilen Anstieg der Regler Ausgangsgröße. Der vorgeschaltete Filter beseitigt diese Unendlichkeit. Es ergibt sich:

$$G_{R(s)} = K_{PR} \left(1 + \frac{0.2s}{1 + 0.01s} \right) = K_{PR} \left(\frac{1 + 0.21s}{1 + 0.01s} \right)$$

Die Stabilitätsbetrachtung hierzu ergibt mittels Bodediagramm:

$$G_{0(s)} = G_{R(s)} \cdot G_{S(s)} = 81,364 \left(\frac{1 + 0.21s}{1 + 0.01s} \right) \cdot \frac{0,061452}{s(1 + s0,2)}$$

$$G_{0(s)} = \frac{5}{s} \left(\frac{1 + 0.21s}{1 + 0.01s} \right) \cdot \frac{1}{(1 + s0,2)}$$

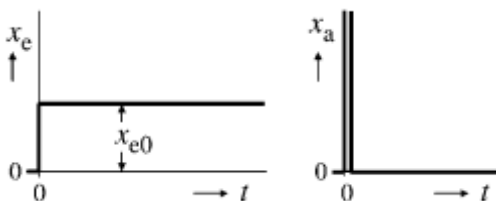
$$G_{0(s)} = \frac{5}{s} \left(\frac{1 + 0.21s}{1 + 0.01s} \right) \cdot \frac{1}{(1 + s0,2)}$$

$$G_{0(s)} = \frac{5}{s} \cdot \frac{1 + 0.21s}{(1 + s0,21 + 0.002s^2)}$$

$$G_{0(s)} = \frac{5 + 1.05s}{(s + s^20,21 + 0.002s^3)}$$

```
s=tf('s');
Fs=(5+1.05*s)/(0.002*s^3+0.21*s^2+s)
bode(Fs)
```

$$^4 f(s) = 1 \text{ für } \delta(t) = \begin{cases} \infty & \text{für } t = 0 \\ 0 & \text{für } t \neq 0 \end{cases}$$



Es gilt für die erste Ableitung des D-Anteils und somit für die Übertragungsfunktion nach Differenziationssatz $L \left[\frac{d^n x(t)}{dt^n} \right] = s^n L[x(t)] = s$

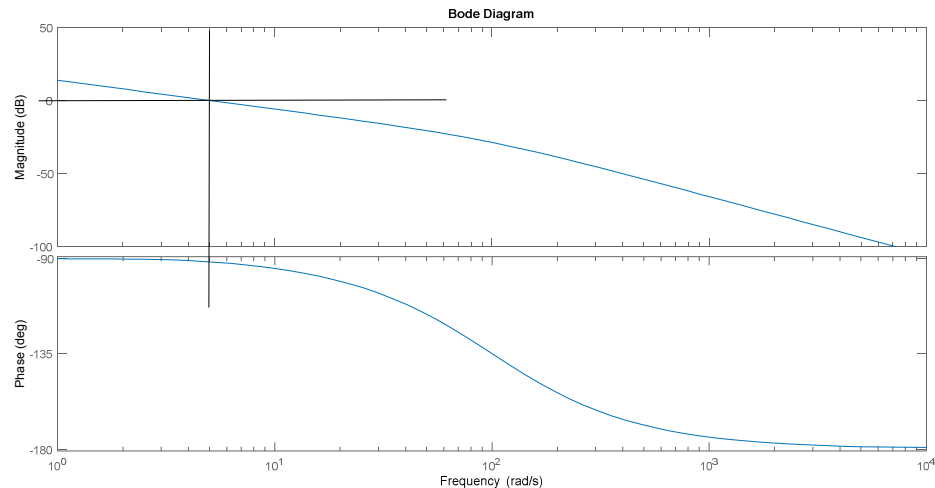
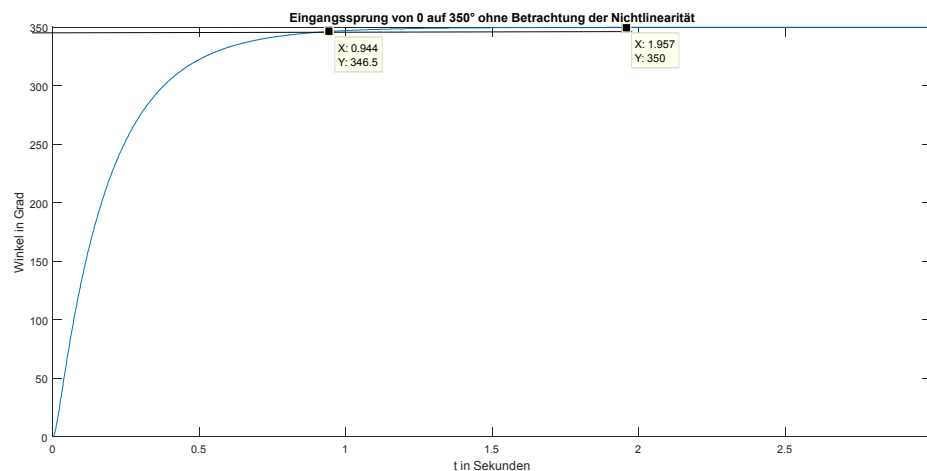


Abbildung 6-7 Bodediagramm mit Matlab Filterung

Der Phasengang läuft asymptotisch gegen 180° . Somit ist die Regelung stabil.

Es wird das Regelverhalten zunächst ohne Nichtlinearität betrachtet:
Es soll eine Regelung von 0° auf 350° durchgeführt werden.

Abbildung 6-8 Eingangssprung von 0 auf 350° ohne Betrachtung der Nichtlinearität

Es lassen sich somit folgende Regelgüteparameter ermitteln:

Ausregelzeit $T_{\text{aus}} = 0,944$ sec.

Erreichung des Sollwertes von 350° nach 1,957 sec.

Die Regelung mit ihrer Ausregelzeit von 0,944 sec. soll hier den Ansprüchen genügen.

Soll der Sollwert schneller, als nach 1,957 Sekunden erreicht werden ist das T_M aus

$$G_{M(s)} = \frac{1}{1+sT_M}$$

entsprechend zu verkleinern.

Eine Störgrößenbetrachtung wurde nicht durchgeführt.

6.1.3 Betrachtung der Nichtlinearität

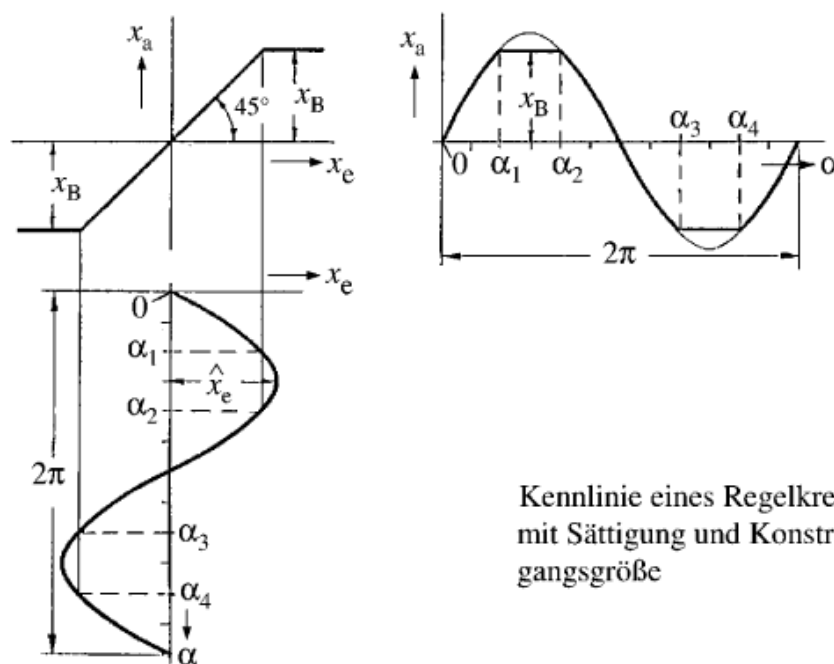
Bei diesem Regelkreis tritt wie bei allen Regelkreisen eine Sättigung auf. So kann z. B. bei einem Verstärker mit dem Verstärkungsgrad K_P die Ausgangsgröße nur einen bestimmten Wert $x_{a_{\max}}$ annehmen. Es gilt die Beziehung $\max. \text{Eingangsgröße} = \max. \text{Ausgangsgröße} / K_P$. Überschreitet die Eingangsgröße ihren Maximalwert, so kann die Ausgangsgröße nicht weiter folgen, der Verstärker ist übersteuert.

Es wird hier die Theorie der harmonischen Balance angewandt. Hier bedient man sich einer Beschreibungsfunktion bei der nur die Grundschiwingung benutzt wird. Die Beschreibungsfunktion ist wie folgt definiert:

$$\text{Beschreibungsfkt.} = \frac{\text{Grundschiwingung am Ausgang}}{\text{Eingangsschiwingung } x_e(\omega)} \Rightarrow N(\hat{x}_e) = \frac{x_{a1}(\omega)}{x_e(\omega)}$$

Die Beschreibungsfunktion ist nur von der Amplitude der Eingangsgröße \hat{x}_e abhängig:

Beschreibungsfunktion eines Gliedes mit Sättigung:



Kennlinie eines Regelkreisgliedes mit Sättigung und Konstruktion der Ausgangsgröße

Abbildung 6-9 Kennlinie Regelkreisglied mit Sättigung und Konstruktion der Ausgangsgröße

Im folgenden Bild ist die Ortskurve der Beschreibungsfunktion eines Gliedes mit Sättigung dargestellt:

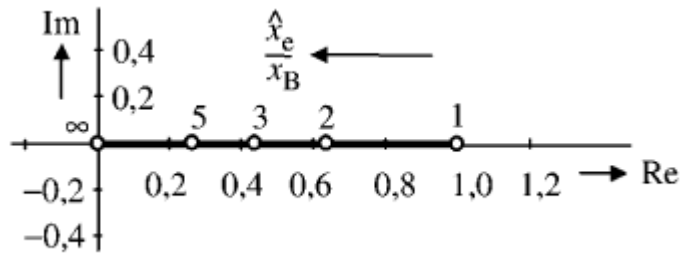


Abbildung 6-10 Ortskurve der Beschreibungsfunktion eines Gliedes mit Sättigung

Das Verfahren der harmonischen Balance betrachtet den geschlossenen Regelkreis an der Stabilitätsgrenze, d. h. es existiert eine stabile Dauerschwingung.

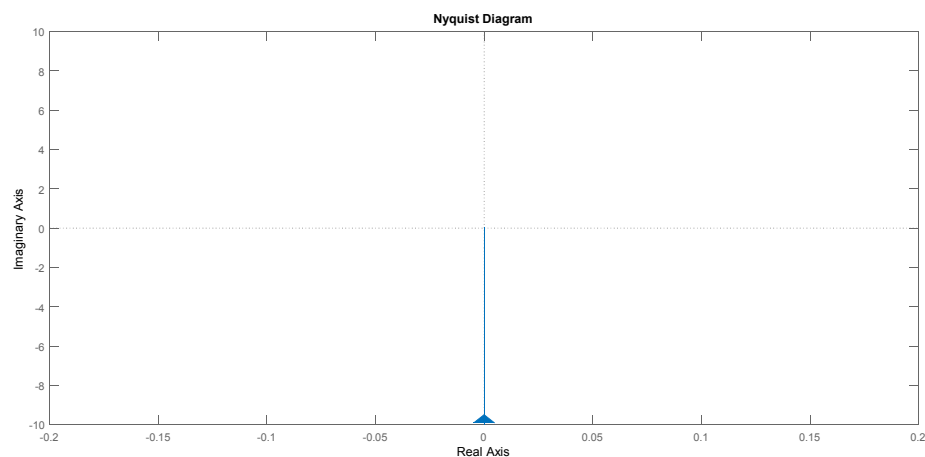
Die Gleichung der harmonischen Balance lautet: $N(\hat{x}_e) = -\frac{1}{G(j\omega)}$.

$$\text{Mit } G_0(s) = \frac{5}{s \cdot \text{sec.}} \Rightarrow -\frac{1}{G_0(j\omega)} = -\frac{j\omega}{5}$$

Man sieht, dass kein Realteil existiert und der negative inverse Frequenzgang keinen Schnittpunkt mit der positiv reellen Achse hat. Es tritt somit keine Dauerschwingung auf.

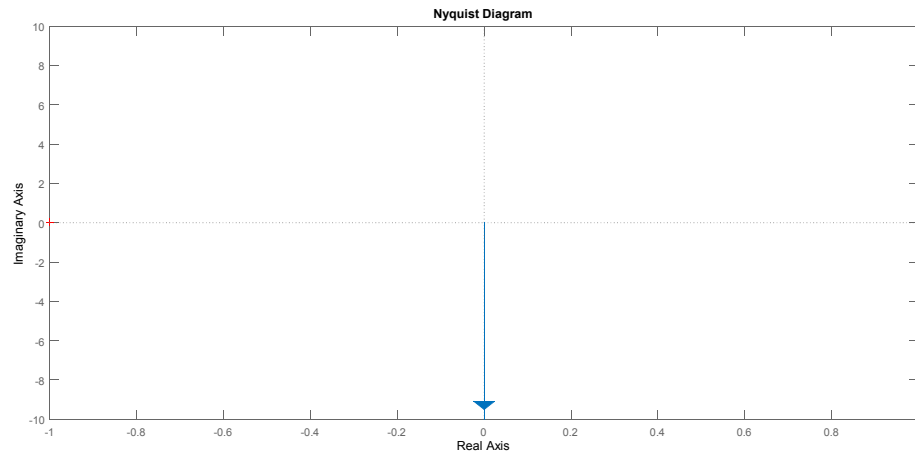
Ortskurve für $G_0(s)$:

```
s=tf('s');
Fs=5/s;
nyquist(Fs)
```

Abbildung 6-11 Ortskurve $G_0(s)$

Ortskurve für $-1 / G_0(s)$:

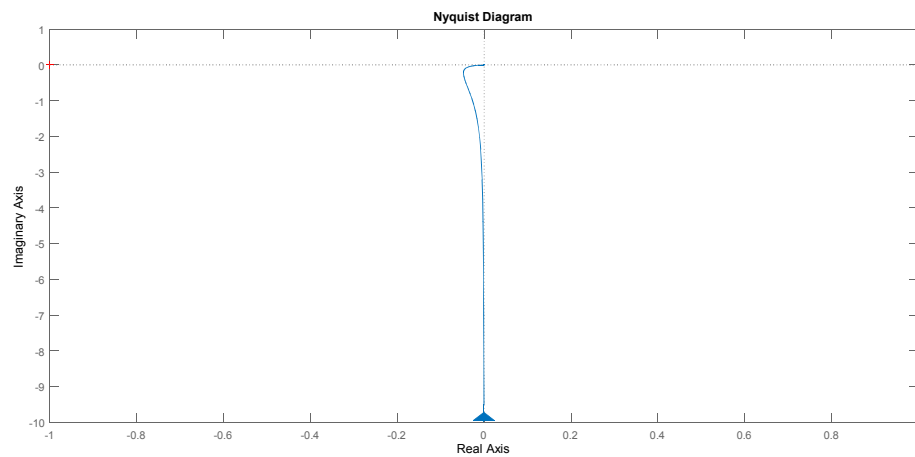
```
s=tf('s');
Fs=-s/5;
nyquist(Fs)
```

Abbildung 6-12 Ortskurve $-1 / G_0(s)$

Unter Berücksichtigung der Filterung mit Matlab ergibt sich:

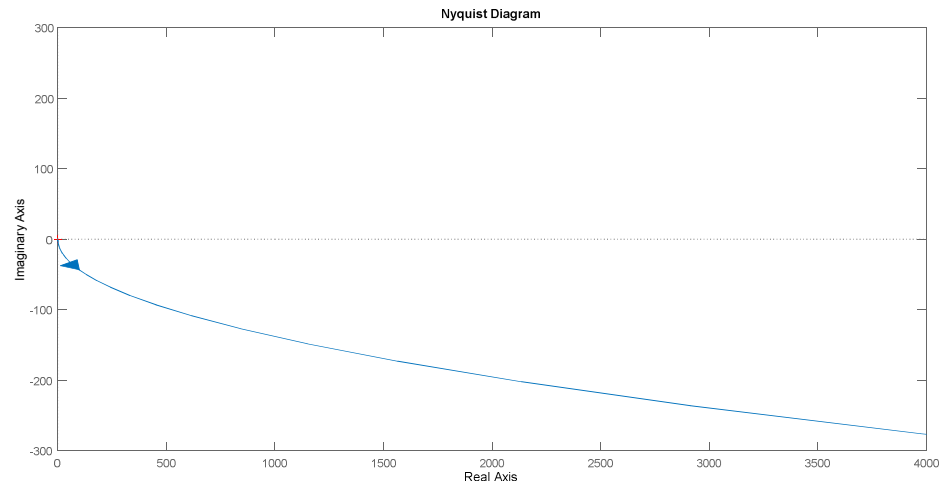
Ortskurve für $G_0(s)$:

```
s=tf('s');
Fs=(5+1.05*s)/(0.002*s^3+0.21*s^2+s);
nyquist(Fs)
```

Abbildung 6-13 Ortskurve für $G_0(s)$ mit Matlab Filterung

Ortskurve für $-1 / G_0(s)$:

```
s=tf('s');
Fs=-(0.002*s^3+0.21*s^2+s)/(5+1.05*s);
nyquist(Fs)
```

Abbildung 6-14 Ortskurve $-1 / G_0(s)$ mit Matlab Filterung

Der negative inverse Frequenzgang hat keinen Schnittpunkt mit der positiv reellen Achse. Es tritt somit keine Dauerschwingung auf.

6.1.3.1 Die ideale PD Regeleinrichtung

Bevor die Nichtlinearität weiter untersucht wird, wird zuvor die ideale PD Regeleinrichtung betrachtet.

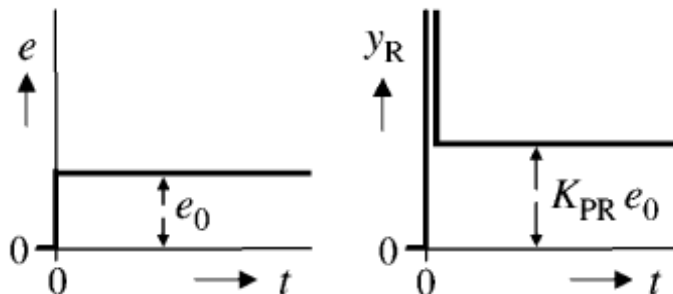


Abbildung 6-15 Eingangssprung und Sprungantwort einer idealen PD Regeleinrichtung

$$Y_R(t) = K_{PR} \left[e(t) + T_v \frac{de(t)}{dt} \right] = 81,364 \left[e(t) + 0,2 \text{sec.} \frac{de(t)}{dt} \right]$$

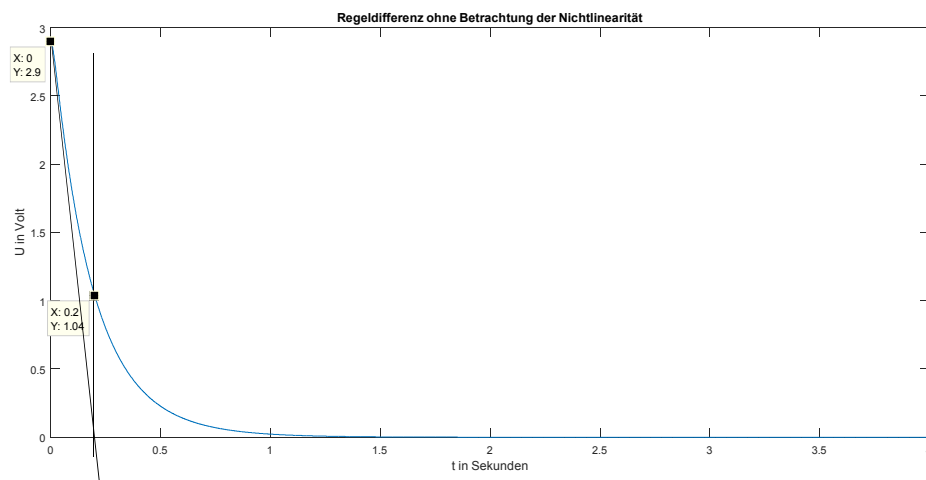
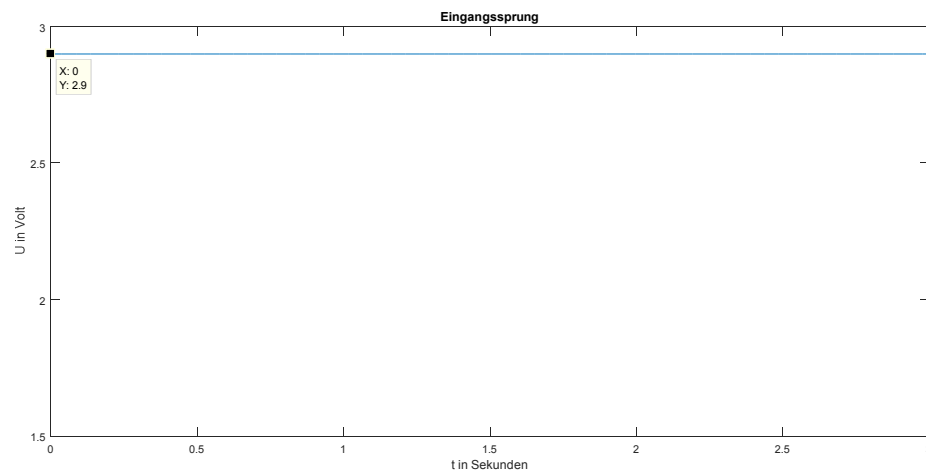
Gleichung 5.1.3.1.1

Man erkennt hier aus der Graphik und der Differentialgleichung folgendes: Die Regeldifferenz steigt bei 0 Sekunden vertikal mit 90° an. Das bedeutet, dass die Ableitung $\frac{de(t)}{dt}$ einen unendlich hohen Wert ergibt. Somit hat der Reglerausgang ebenfalls einen unendlich hohen Wert bei $t = 0$ Sekunden. Man erkennt also, dass in den Reglerausgang direkt der Differentialquotient der Regeldifferenz eingeht. Hier wurde eine konstante Regeldifferenz angenommen, so dass unmittelbar nach dem mit 90° steil ansteigenden

Eingangssprung die Ableitung 0 beträgt. Infolge dessen wird bei dem Reglerausgang nur noch der P Anteil wirksam.

Bei der Antennenregelung hat aber, die Regeldifferenz nicht das in Bild 6.1.15 gezeigte Verhalten. Weiterhin ist zu bemerken, dass es sich auf Grund der geringen Filterung in Matlab nicht um ein reines PD-Verhalten sondern um ein PD-T1 Verhalten handelt.

In den folgenden Bildern werden die Regeldifferenz und der Reglerausgang ohne Betrachtung der Sättigung festgehalten.



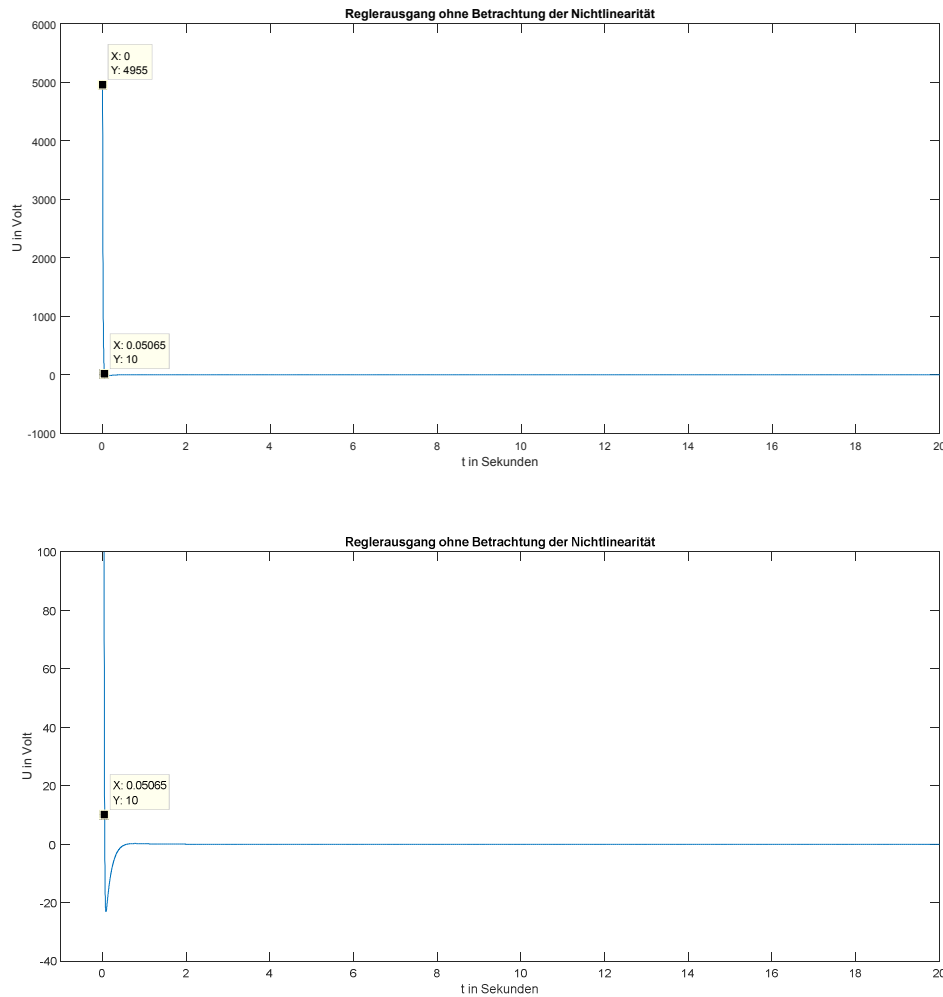


Abbildung 6-16 Eingangssprung, Regeldifferenz und Reglerausgang ohne Betrachtung der Nichtlinearität

Aus dem Bild für den Reglerausgang ist folgendes ersichtlich: Die Regeldifferenz beträgt schlagartig 2,9V. Dadurch steigt der Reglerausgang sehr stark an. Der Maximalwert wird wie folgt berechnet: $K_{PR} \cdot \frac{T_V}{T_1} \cdot \hat{e} = 81,364 \cdot \frac{0,21}{0,01} \cdot 2,9 \text{ V}$. Somit ergibt sich ein Maximalwert von 4955,0676 V.

Danach wird der Ausgang $\tilde{u}_{R(t)}$ negativ. Geht die Regeldifferenz gegen 0 wird schließlich auch der Reglerausgang 0 werden.

Es ergibt sich für den Ausgang der Regelstrecke⁵

$$y(t) = K_I * u_{R(t)} \left[t - T_1 \left(1 - e^{-\frac{t}{T_1}} \right) \right]$$

⁵ $u_{R(t)}$ = Stellgröße

$$y(t) = 0,061452 \text{sec.}^{-1} * u_{R(t)} \left[t - 0,2 \text{sec.} \left(1 - e^{-\frac{t}{0,2 \text{sec.}}} \right) \right]$$

Im folgenden Bild ist dargestellt wie lange das Potentiometer ungefähr zum Erreichen des Endwertes von 350° benötigt wenn der Motor von Beginn an mit der Maximalspannung von 10V versorgt wird. 2,9V entspricht dabei 350° . Somit folgt daraus, dass das Poti ungefähr $4,624 \text{sec.} - 0,328 \text{sec.} = 4,296 \text{sec.}$ zum Erreichen des Endwertes von 350° benötigt. Die Verzugszeit ist hier nicht berücksichtigt worden, da während dieser Zeit die Massenträgheit überwunden werden muss und somit während dieser Zeit keine Drehbewegung stattfindet.

Daraus folgt dass, das Potentiometer für 1 Grad 0,0122 Sekunden benötigt.

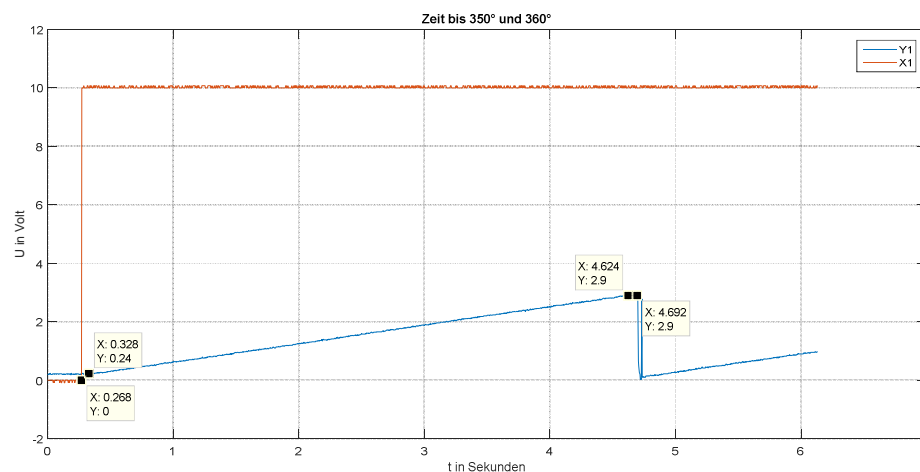


Abbildung 6-17 Zeit Potentiometer bis 350°

6.1.3.2 Regelung mit Sättigung des Stellsignales

Das Stellsignal wird nun auf einen Wert von +10V und -10V begrenzt.

Es soll eine Antennenregelung von 0° auf 350° erfolgen.

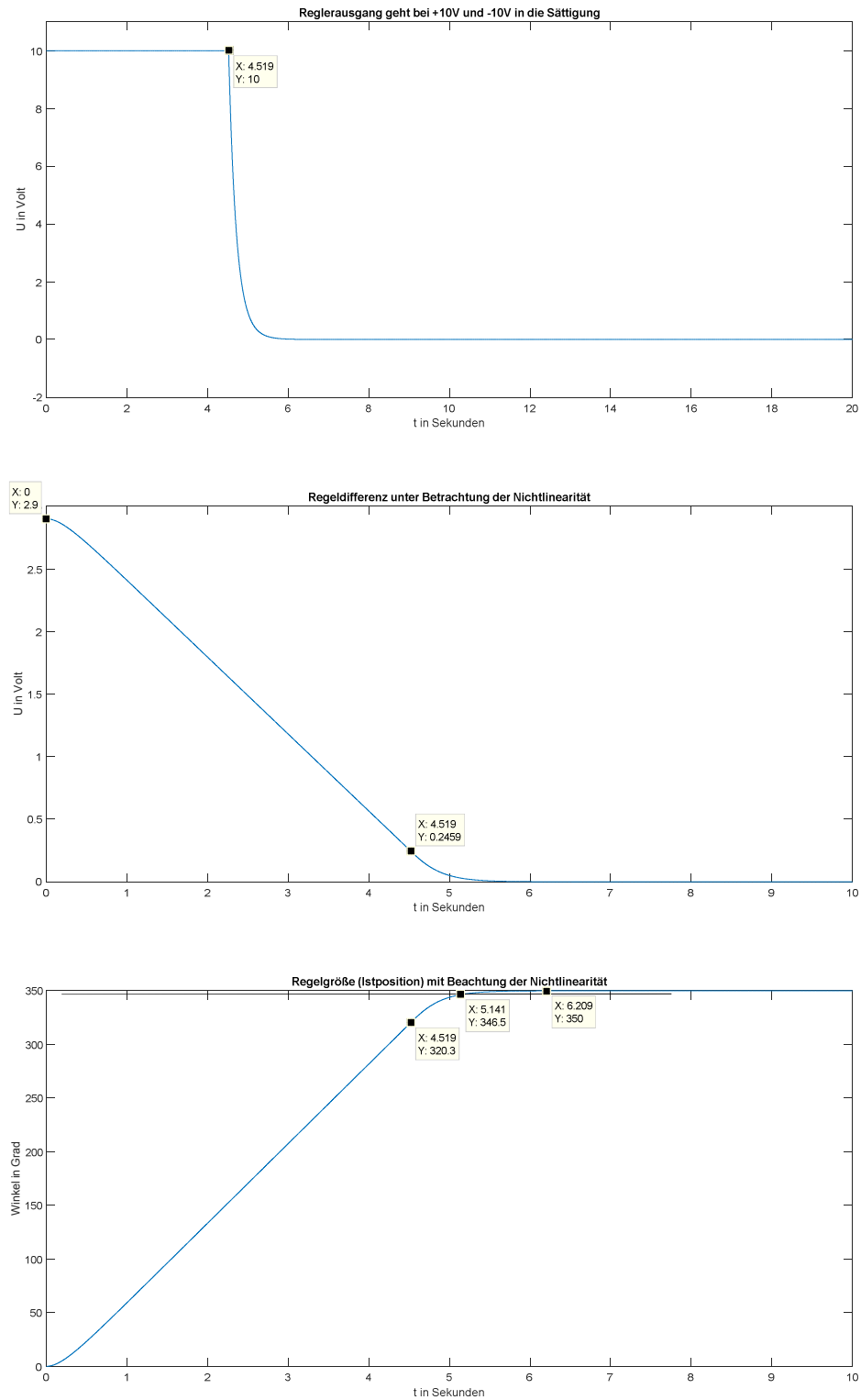


Abbildung 6-18 Reglerausgang, Regeldifferenz und Istposition unter Beachtung der Nichtlinearität

Ausregelzeit $T_{\text{aus}} = 5,141 \text{ sec.}$

Erreichung des Sollwertes von 350° nach $6,209 \text{ sec.}$

Aus diesen Bildern ist folgendes zu erkennen: Der Motor wird 4,519 Sekunden mit 10V versorgt. Oben wurde geschätzt, dass das Poti ungefähr 4,296 sec. zum Erreichen des Endwertes von 350° benötigt. Dieser Wert wird nun genauer ermittelt bzw. unter Betrachtung des Streckenmodells und Simulink Modells ermittelt.

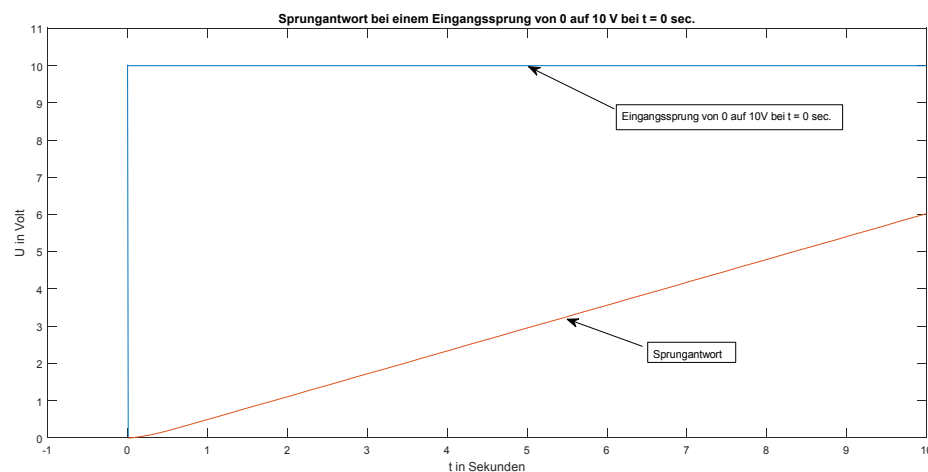
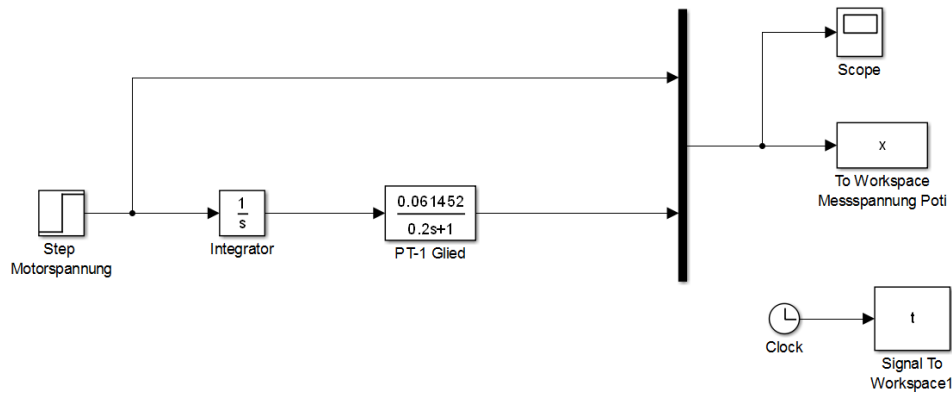


Abbildung 6-19 Streckenmodell aus der Systemanalyse

Die Steigung des Simulink Streckenmodells aus der Systemanalyse beträgt $0,614575$ V/sec. bei einem Eingangssprung von 0 auf 10Volt. Dies ergibt eine Steigung von $74,17^\circ$ pro sec. Somit braucht das Potentiometer unter Berücksichtigung des Simulink Modells 4,72 sec. zum Erreichen des Endwertes von 350° . Regeldifferenz und Istwert befinden sich bis zu dem Wert von 4,519 sec. im linearen Bereich mit einer konstanten Steigung. Je kleiner man die Zeitkonstante T_M wählt, desto größer wird die Reglerverstärkung K_{PR} . Dadurch wird der Regler noch länger in der Sättigung bleiben.

Weiterhin ist erkennbar, dass der Regler durch die Sättigung an Dynamik verliert. Somit wird ein schnelles Ansprechen der Regelung und somit eine schnelle Regelung verhindert.

Es ist angedacht, dass die Antenne z. B. genau von 0° auf 350° oder von 350° nach 0° geregelt werden soll. Im Fall von 0° nach 350° darf die Antenne nicht über 360° hinaus regeln, da in diesem Fall die Istposition größer oder gleich 0 wäre und somit wieder eine sehr hohe Regeldifferenz bei einem Sollwert von 350° anliegen würde. Der Regler würde wieder voll aufmachen und die vollen 10V an den Motor geben. Der Sollwert würde nie erreicht, die Regelung käme nie zur Ruhe und würde schwingen. Das Potentiometer besitzt einen elektrischen Drehwinkel von 350° und einen mechanischen Drehwinkel von 360° . Dies bedeutet, dass ab 350° die Potentiometer Ausgangsspannung nicht weiter ansteigen wird und die Spannung für 350° konstant bis 360° anliegen wird. Somit wäre in diesem Fall ein minimales Überschwingen der Antenne über 350° tolerierbar. Bei einer Regelung auf 0° ist kein Überschwingen tolerierbar, da ansonsten die Potentiometer Ausgangsspannung von 0 Volt direkt auf 2,9 Volt springen wird. Die Regelung würde schwingen und käme nicht zur Ruhe. Gegebenenfalls ist hier ein Puffer einzubauen, so dass man die Antenne nicht auf exakt 0° sondern nur auf $0,5^\circ$ oder 1° regeln kann und nicht darüber hinaus.

Es soll nun weiter untersucht werden, ob durch die Sättigung die Antenne über 350° hinausfahren kann, bzw. in entgegengesetzter Richtung über 0° hinausfahren kann.

Der Reglerausgang geht über die PWM auf die Regelstrecke und somit auf den Gleichstrommotor. Im Bereich der Sättigung wird der Gleichstrommotor konstant mit 10V versorgt. Die Drehzahl des Gleichstrommotors ist proportional seiner angelegten Spannung und somit proportional zum Reglerausgang. Die Winkelgeschwindigkeit des Potentiometers ist proportional zur angelegten Motorspannung und somit ebenfalls proportional zum Reglerausgang. Der Winkel des Potentiometers ist wiederum proportional zur Potentiometer Ausgangsspannung. Das bedeutet, dass der Winkel bzw. die Spannung des Potentiometers bis zur Zeit von 4,519 sec. linear mit konstanter Steigung ansteigen wird, da sich der Winkel aus dem Integral der Winkelgeschwindigkeit ergibt.⁶ Den linearen und konstanten Anstieg bzw. Abfall kann man im Bild der Istposition aber auch im Bild zur Regeldifferenz sehr gut erkennen. Die Regelung bleibt also bis zu einem Winkel von $335,17^\circ$ in der Sättigung. Ob ein Überschwingen der Antennenregelung stattfinden kann, kann ergänzend zur Istposition aus dem Integral des Reglerausganges ermittelt werden. Dies kann man sowohl in Simulink, als auch über die Matlab Funktion *trapz* berechnen. Dabei trifft allerdings nicht die Annahme zu, dass 10 V einer Winkelgeschwindigkeit 10° pro Sekunde entspricht sondern einer Winkelgeschwindig-

⁶ $\int \omega = \varphi$

keit von $74,17^\circ$ pro Sekunde. Beträgt der Reglerausgang 1 Volt entspricht dies einer Winkelgeschwindigkeit von $7,417^\circ$ pro Sekunde. Daher ist das Integral mit einem Faktor von $7,417^\circ$ pro Sekunde zu multiplizieren. (siehe Simulink Modell).

Bei einem Eingangssprung von 0 auf 350° ergibt die Berechnung des Integrals des Reglerausganges 350° . Und zwar gilt dies für alle T_M Werte. Grundsätzlich gilt die Aussage, dass das Integral immer dem Endwert des Eingangssprunges entspricht, also z.B. ergibt ein Eingangssprung von 0 auf -10° ein Integral von -10° . Alle zum Kompensationsregler gemachten Aussagen gelten auch für negative Eingangssprünge. Es kehrt sich lediglich nur das Vorzeichen von plus auf minus um. Auch aus dieser gerade durchgeführten Betrachtung erkennt man, dass kein Überspringen stattfindet.

Das Regelverhalten für unterschiedliche Sollwertsprünge bei einem T_M von jeweils 0,2 sec. ist in der folgenden Tabelle festgehalten worden:

Regelverhalten ohne Berücksichtigung der Sättigung		
Sollwertsprung von	Ausregelzeit in sec.	Erreichung Sollwert in sec.
0° auf $\pm 350^\circ$	0,944	1,957
0° auf $\pm 50^\circ$	0,945	2,044
Regelverhalten mit Berücksichtigung der Sättigung		
Sollwertsprung von	Ausregelzeit in sec.	Erreichung Sollwert in sec.
0° auf $\pm 350^\circ$	5,141	6,209
0° auf $\pm 50^\circ$	1,586	2,727

Man erkennt, dass bei großen Sollwertsprüngen die Sättigung einen wesentlich schlechteren Einfluss auf das Regelverhalten hat, als bei kleinen Sollwertsprüngen. Dies ist auch verständlich, da bei großen Sollwertsprüngen der Regler wesentlich länger in der Sättigung bleibt und somit wesentlich mehr an Dynamik verliert.

Das T_M wurde hier nicht weiter verkleinert, da ein kleineres T_M als 0,2 sec. keine wesentlichen Verbesserungen der Regelung brachte. Das T_M von 0,2 sec. gilt als ausreichend.

6.1.4 Betrachtung Vorgabe eines Winkels als Sollposition für das Discovery Board

Im vorherigen Kapitel wurde unter Matlab der Winkel als Sollposition vorgegeben. Dieser wurde dann in Matlab oder auf dem Discovery Board in eine Spannung umgerechnet, so dass die Spannung als Sollwert zur Verfügung stand. Diese Umrechnung soll

hier nicht erfolgen. Der Sollwinkel soll dem Discovery Board direkt als Winkelwert zur Verfügung stehen. Somit muss auch die Istposition als Winkelwert zur Verfügung stehen. Da das Potentiometer als Istposition einen Spannungswert liefert, muss dieser Spannungswert auf dem Discovery Board in einen Winkel umgerechnet werden. Dieser Winkelwert (Istposition) wird dann dem PD Regler auf dem Discovery Board zur Verfügung gestellt.

Die Regelung mit einem Sollwert und Istwert als Winkel für den PD Regler wird hier zuerst analog kontinuierlich unter Simulink betrachtet. Dabei hat die Regelung das folgende Aussehen.

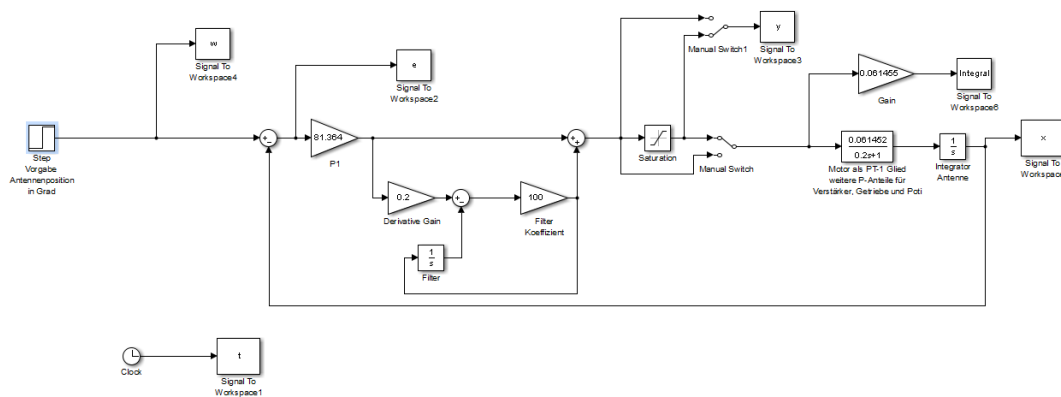


Abbildung 6-20 Regelung für einen Winkel als Sollwert und Istwert

Es wird weiterhin von folgender Annahme ausgegangen:

$$\frac{350^\circ}{2,9 V} = \frac{1206,897^\circ}{10V}$$

Aufgrund dessen, dass bei einer Betrachtung in Grad die Regeldifferenz höher ist, liefert der Reglerausgang auch höhere Werte. Infolge dessen muss die Sättigung entsprechend nach oben hin erhöht werden. Somit muss hier die Sättigung auf $\pm 1206,897$ eingestellt werden. Zur Berechnung des Integrals ist eine Multiplikation mit 0,061455 erforderlich, da hier die Beziehung gilt $74,17^\circ/\text{sec.}$ entspricht $1206,897^\circ$ und $0,061455^\circ/\text{sec.}$ entspricht somit 1° .

Es wurde das gleiche Verhalten wie zuvor bei einer Regelung auf eine Spannung verifiziert.

6.1.5 Diskretisierung (Digitalisierung) des Kompensationsreglers

Damit C-Code generiert werden kann, muss der Regelkreis digitalisiert und somit mindestens für den Regler eine Z- Transformation durchgeführt werden. Die Regelstrecke kann mit einem vorgeschalteten Zero-Order-Hold Glied aus der Simulink Bibliothek

digitalisiert werden. Bei digitalen Reglern handelt es sich immer um Abtastsysteme. Die Erfassung der Messwerte und die Berechnung der Stellwerte erfolgt somit zeitdiskret.

Hierzu muss ein Abtastintervall (Sample Time) festgelegt werden. Je niedriger dieses ausfällt, desto ähnlicher verhält sich der diskrete Regler zum stetigen Regler. Es wird hier 1 ms gewählt. Es handelt sich somit um eine quasikontinuierliche Regelung.

6.1.5.1 Wahl der Abtastzeit

Um die Regelung optimal zu gestalten, muss nach jedem Abtastzeitpunkt möglichst schnell ein aktueller Stellwert berechnet werden. Also hat für den Regler das Messen des Abtastwertes mit anschließender Berechnung des Stellsignals höchste Priorität. Dabei sollte die Rechenzeit des Reglers entsprechend kleiner sein als die Abtastzeit, so dass nach der Rechenzeit des Reglers in der verbleibenden Zeit bis zur einer neuen Abtastung noch genügend Zeit für die Erledigung anderer Aufgaben vorhanden ist.

Ist die Rechenzeit sehr klein gegenüber der Abtastzeit, kann die Rechenzeit vernachlässigt werden. In diesem Fall verlangsamt nur die AD Wandlung die Regelung. Ein analoger Regelkreis wird beim Übergang zu digitaler Regelung für den Fall, dass die Rechenzeit wesentlich kleiner als die Abtastzeit ist, mit einer zusätzlichen Totzeit ergänzt.

Von quasikontinuierlicher Regelung, also von einer kontinuierlichen Behandlung des Regelkreises spricht man, wenn die Abtastzeit viel kleiner als die Zeitkonstante der Regelstrecke ist.⁷

Die Abtastzeit darf nicht so groß gewählt werden, da der Regelkreis wegen großer Totzeiten instabil werden kann. Sie darf aber auch nicht zu klein gewählt werden, da ansonsten der Regler überlastet wird. Bei einer IT-1 Strecke hängt die Abtastzeit von der Verzugszeit T_U (Zeitkonstante T_1) und von der Zeit, bei der die Regelgröße 95% des Beharrungszustandes erreicht hat, ab.

Wie im Kapitel 5 festgestellt gilt für den offenen Regelkreis mit Kompensationsregler

$$G_{0(s)} = G_{R(s)} \cdot G_{S(s)} = \frac{1}{G_{S(s)}} \cdot \frac{G_{M(s)}}{1 - G_{M(s)}} \cdot G_{S(s)}$$

$$\frac{G_{M(s)}}{1 - G_{M(s)}} = \frac{1}{1 - \frac{1}{1 + sT_M}} = \frac{1}{\frac{1 + sT_M - 1}{1 + sT_M}} = \frac{1}{sT_M} = \frac{5}{s \cdot \text{sec.}}$$

Es soll nun untersucht werden, wie groß die Abtastzeit gewählt werden kann, damit der geschlossene Kreis stabil bleibt. Es wird davon ausgegangen, dass die Rechenzeit klei-

⁷ Für eine IT1 Strecke gilt: Abtastzeit $T_A \leq 0,1 \cdot \text{Verzugszeit } T_U$

ner als die Abtastzeit ist. Somit gilt $T_t = 0.5 T_A$. Diese Totzeit muss in den Regelkreis eingefügt werden. Für den kritischen Zustand gilt:

$$\varphi(\omega_{\Pi}) = -\frac{\Pi}{2} + \varphi_{T_t}(\omega_{\Pi}) = -\frac{\Pi}{2} - \omega_{\Pi} T_t = -\Pi \Rightarrow T_t = \frac{\Pi}{2 \cdot \omega_{\Pi}} \Rightarrow T_t = \frac{\Pi}{2 \cdot 5 \text{ sec}^{-1}} = 0.314 \text{ sec.}$$

$T_A = 0.628 \text{ sec.} \Rightarrow$ Der Kreis wird bei $T_A < 0.628 \text{ sec.}$ stabil.

Das zugehörige Bodediagramm für eine Abtastzeit von 1ms sieht wie folgt aus.

```
s=tf('s');
Fs=(5/s)*exp(-s*0.0005);
bode(Fs)
```

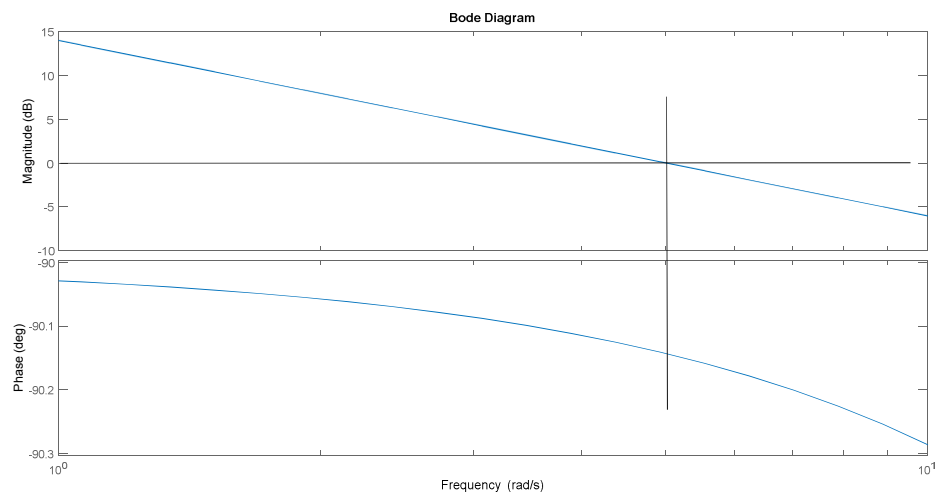


Abbildung 6-21 Bodediagramm für eine Abtastzeit für $T_A = 1 \text{ ms}$

Man sieht, dass der geschlossene Regelkreis ohne Betrachtung der Nichtlinearität stabil ist.

Für die Filterung mit Matlab ergeben die Betrachtungen:

$$G_{0(s)} = \frac{5}{s} \cdot \frac{1 + 0.21s}{(1 + s0.21 + 0.002s^2)} \cdot e^{-sT_t}$$

$$\varphi = -\frac{\Pi}{2} - \omega \cdot T_t + [(\arctan(0.21\omega)) - (\arctan(\frac{0.21}{1-0.000004\omega^2}))]$$

ω = Kreisfrequenz bei der die Amplitude durch Null geht. Aus Bild 6.1.21 ergibt sich hierfür $\omega = 5 \text{ sec}^{-1}$. Für φ muss ein Wert von $-\Pi$ angenommen werden. Auflösen der Gleichung nach T_t ergibt 0,309044. Daraus folgt für die Stabilität $T_A < 0.618 \text{ sec.}$

```
s=tf('s');
Fs=((5+1.05*s)/(0.002*s^3+0.21*s^2+s))*exp(-s*0.0005);
bode(Fs)
```

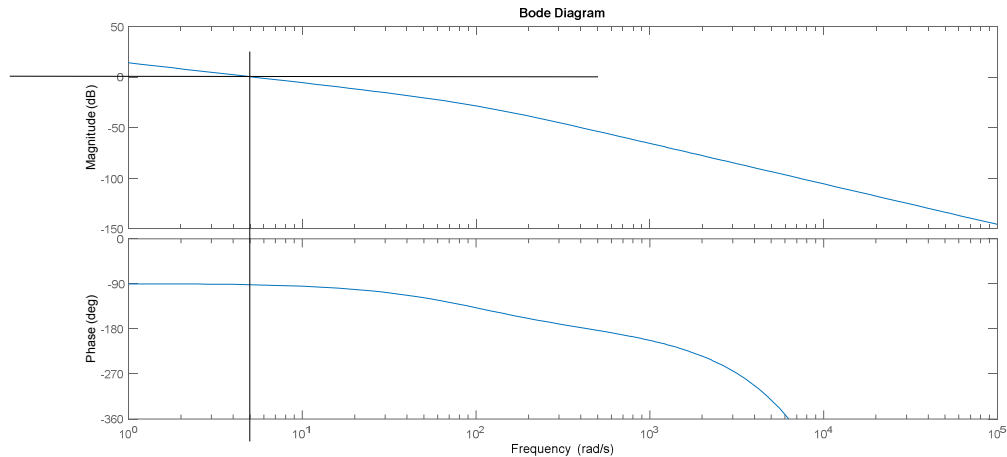


Abbildung 6-22 Bodediagramm für eine Abtastzeit für $T_A = 1$ ms mit Matlab Filterung

Man sieht, dass der geschlossene Regelkreis ohne Betrachtung der Nichtlinearität stabil ist. Die Stabilitätsuntersuchung muss nun ebenfalls noch für die Nichtlinearität überprüft werden. Die Totzeit wird dabei mit $1/(1+sT_t)$ angenähert.

Es wird dazu die Ortskurve für $-1 / G_0(s)$ aufgenommen.

$$\text{Es gilt: } G_0(s) = \frac{1}{sT_M} \cdot e^{-sT_t} = \frac{1}{sT_M \cdot (1+sT_t)}$$

Ortskurve für $-1 / G_0(s)$:

```
s=tf('s');
Fs=0.2*s*(1+0.0005*s);
nyquist(Fs)
```

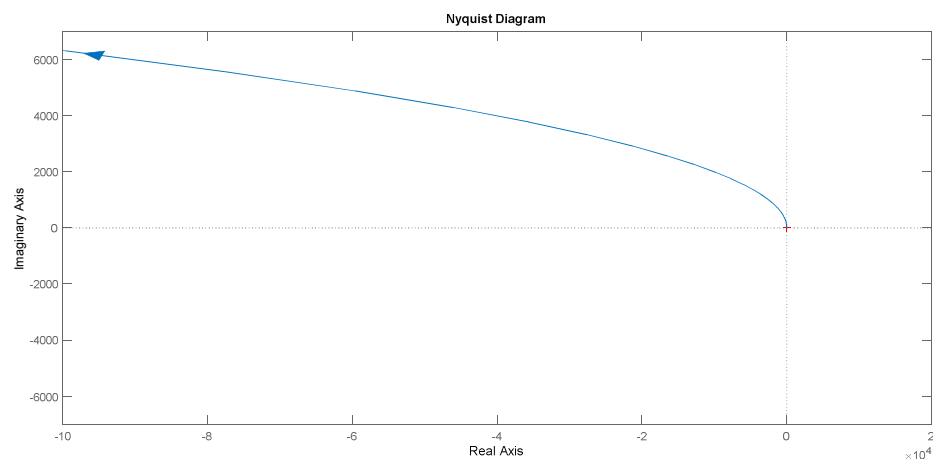


Abbildung 6-23 Ortskurve für $-1 / G_0(s)$:

Für Betrachtung mit Filterung ergibt sich:

```
s=tf('s');
Fs = ((0.002*s^3+0.21*s^2+s)*(1+0.0005*s))/(5+1.05*s);
nyquist(Fs)
```

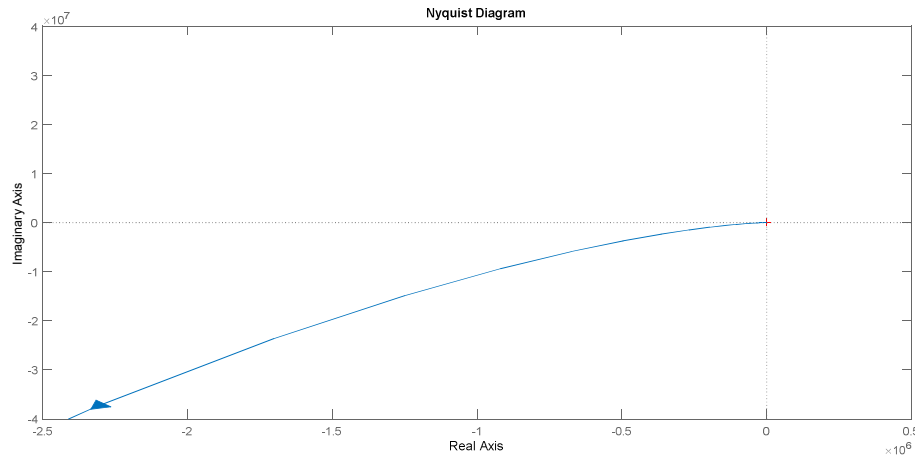


Abbildung 6-24 Ortskurve für $-1 / G_0(s)$ mit Matlab Filterung

Der negative inverse Frequenzgang hat keinen Schnittpunkt mit der positiv reellen Achse. Es tritt somit keine Dauerschwingung auf.

6.1.5.2 Durchführung der Z-Transformation

Als erstes wird der Regelkreis aus Kapitel 5 digitalisiert. Es wird dazu die Beschreibungsform der Z-Transformation angewandt.

Für die Diskretisierung der IT-1 Strecke wird ein Zero-Order-Hold Glied eingesetzt.

Die Digitalisierung des PD Reglers ist unter Simulink direkt in der PD Regler Toolbox realisiert worden. Auf eine Filter Methode ist dabei verzichtet worden, da anders als beim analogen Fall hier von Matlab keine Filterung vorgeschrieben wird und die Filterung für den diskreten Fall keine wesentliche Verbesserung brachte. Es handelt sich somit also um einen reinen diskreten PD Regler.

Der digitalisierte Regelkreis zur Regelung auf einen Winkel ist hier nun dargestellt.

Eine Stabilitätsbetrachtung wurde ebenfalls durchgeführt. Der Regelkreis ist stabil. Die Ausregelzeit und die Erreichung des Sollwertes unterscheiden sich nur geringfügig zum analogen Regelkreis. Der Wert bis zum Verbleiben in der Sättigung ist der gleiche wie beim analogen Regelkreis.

Abschließend sind noch einige Kennlinien sowohl für positive als auch für negative Regeldifferenzen dargestellt.

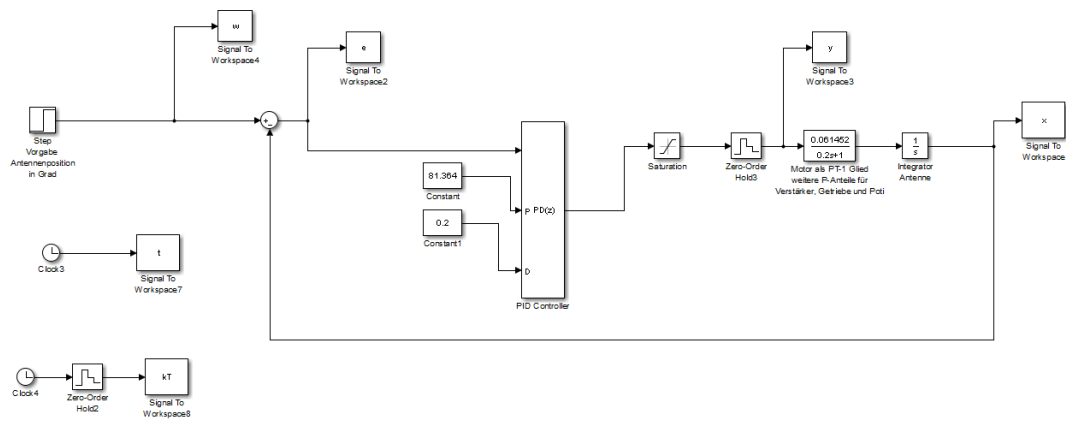


Abbildung 6-25 digitalisierter Regelkreis

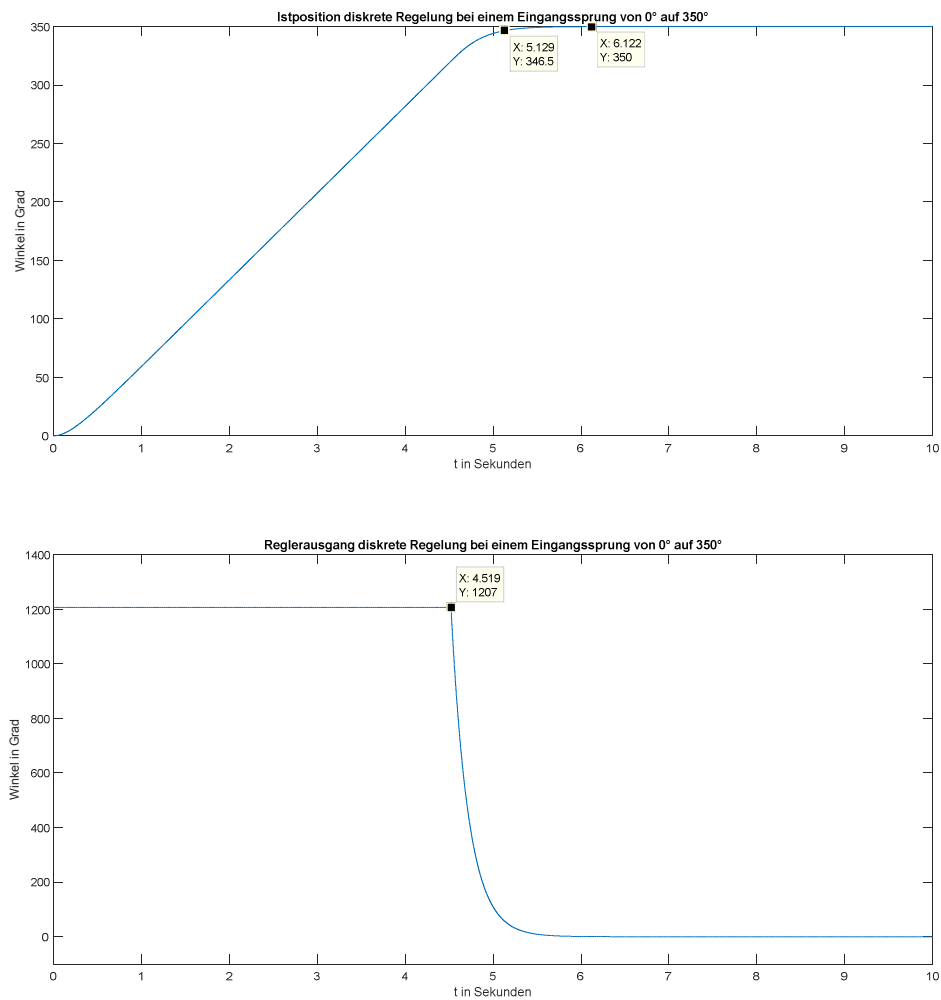


Abbildung 6-26 Istposition und Reglerausgang des digitalen Regelkreises bei einem Eingangssprung von 0° auf 350°.

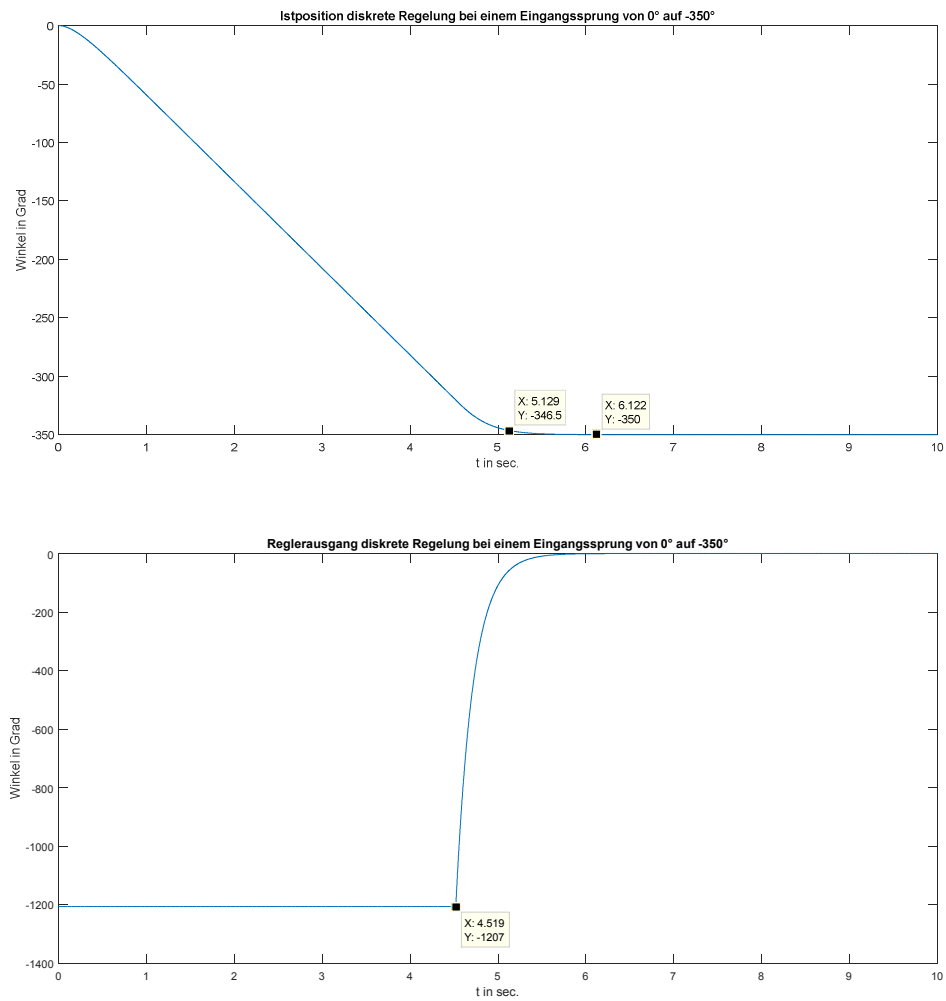
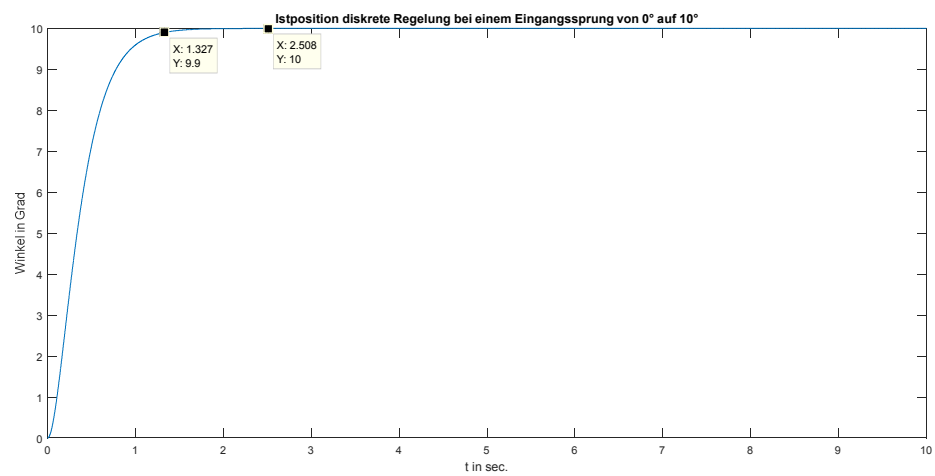


Abbildung 6-27 Istposition und Reglerausgang des digitalen Regelkreises bei einem Eingangssprung von 0° auf -350° .



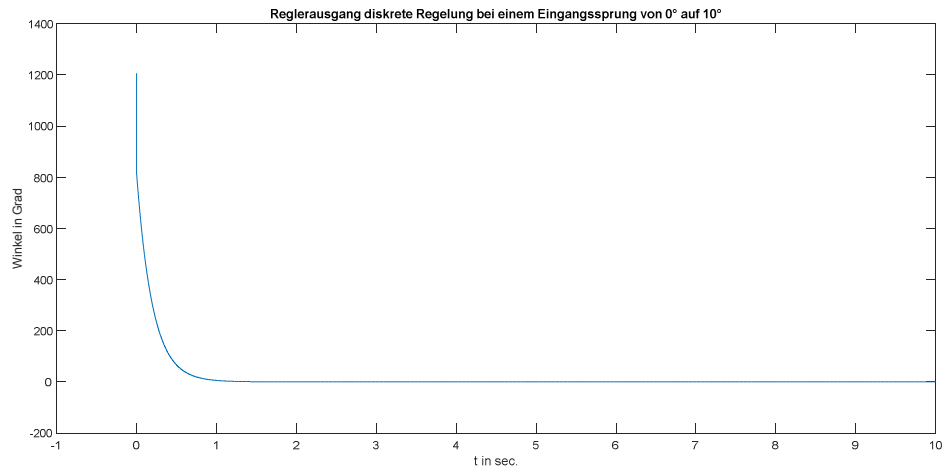


Abbildung 6-28 Istposition und Reglerausgang des digitalen Regelkreises bei einem Eingangssprung von 0° auf 10° .

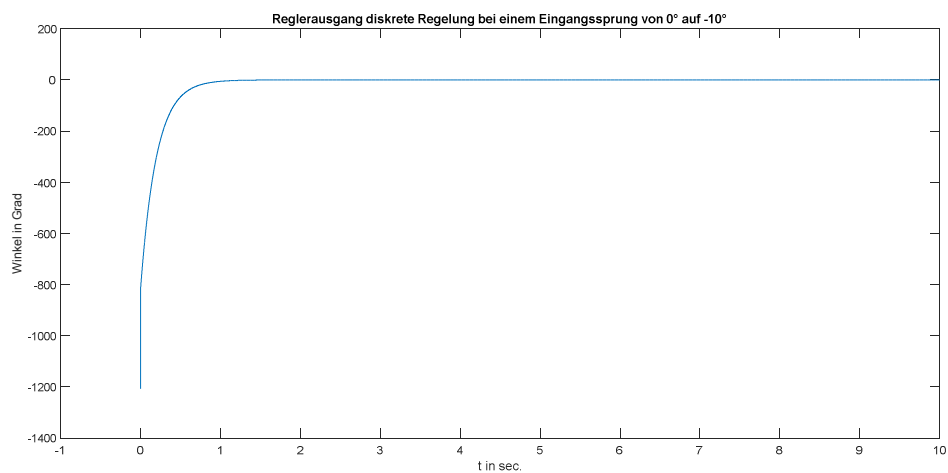
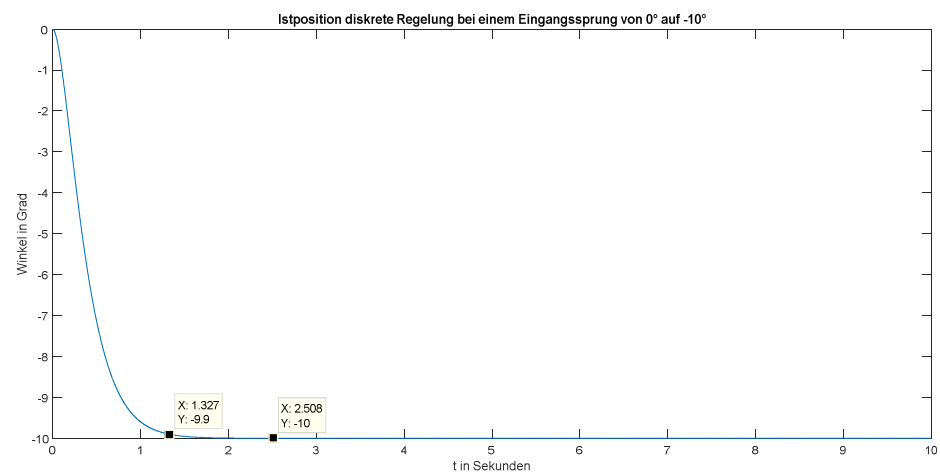


Abbildung 6-29 Istposition und Reglerausgang des digitalen Regelkreises bei einem Eingangssprung von 0° auf -10° .

6.2 Smith-Predictor

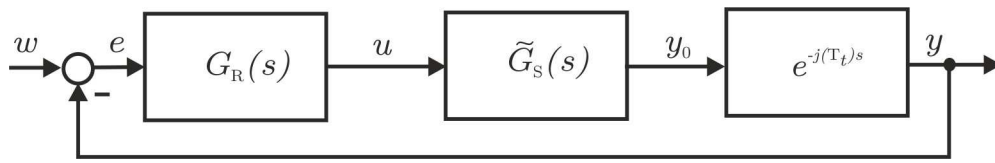


Abbildung 6-30 Regelkreis

Eine standardmäßige Reglerstruktur ist der Abbildung 6-30 Regelkreis zu entnehmen, wobei w die Führungsgröße, e das Fehlersignal, u die Stellgröße, y_0 das Ausgangssignal der verzögerten und y das Ausgangssignal der gesamten Strecke angibt. Der Regler $G_R(s)$ wird größtenteils mit Hilfe einer *PID*-Struktur realisiert, wobei je nach Anwendung der *I* bzw. *D*-Anteil wegfallen kann. Weist ein zu regelndes System $G_S(s) = \tilde{G}_S(s) \cdot e^{-jT_t s}$ nun eine Totzeit T_t auf, kann der Regler $G_R(s)$ zu starke Stellgrößen ausgeben, da diesem keine Information über T_t vorliegt. Spaltet man das System $G_S(s)$ in einen totzeitfreien Anteil mit Verzögerungsglied auf, erhält man den in Abbildung 6-30 stehenden Regelkreis. Um dem Regler das verzögerungsfreie Prozessverhalten bereitzustellen, wird nun der unten abgebildete *Smith-Prädiktor* angewandt. Die verzögerte Reaktion des Systems bleibt erhalten, jedoch gibt der Regler die Stellgröße aus, die zunächst dem unverzögerten Prozess entspricht.

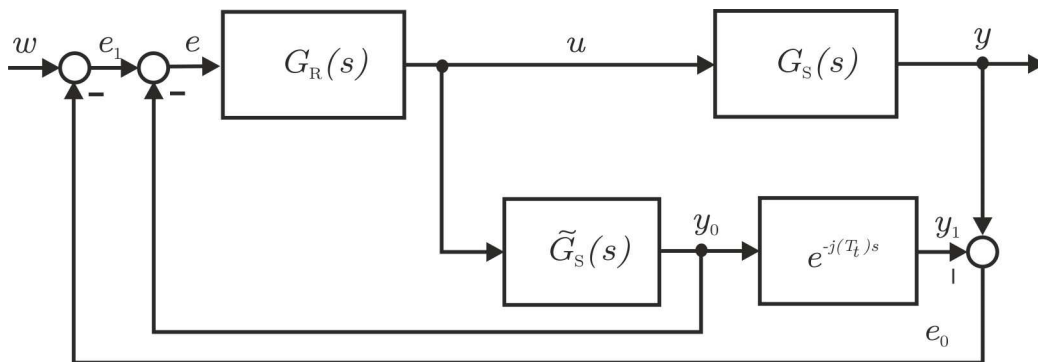


Abbildung 6-31 Smith-Prädiktor

6.2.1 Funktionsweise Smith-Prädiktor

Das System $G_S(s)$ stellt den zu regelnden Prozess mit der Totzeit T_t dar. Die Stellgröße des Reglers $G_R(s)$ wird zusätzlich zum Modell $G_S(s)$ gleichzeitig auf ein verzögerungsfreies Modell $\tilde{G}_S(s)$ der Strecke geführt, wobei daraufhin die Verschiebung um die Totzeit T_t folgt. Die Differenzbildung $e_0 = y - y_1$ hat den Zweck Störungen, die in der Modellierung $\tilde{G}_S(s) \cdot e^{-jT_t s}$ auftauchen, auf den Regler zurückzuführen. Sollte die Modellierung des Prozesses vollständig mit $G_S(s)$ übereinstimmen, entfällt die Notwendigkeit dieser Rückführung. Die Charakteristik besteht nun darin, dass anstelle

des direkt messbaren Ausgangssignals y die um T_t vorgezogene Messgröße y_0 zurückgeführt wird. Somit kann die der Strecke $G_S(s)$ passende Stellgröße u ausgegeben werden.

6.2.2 Simulation

Gegenstand der Betrachtung ist im Folgenden ein System, dessen Übertragungsfunktion im Bildbereich durch ein IT_1 -Glied gegeben ist. Das System kann durch ein I-Glied mit einer Verzögerung um die Zeit T_t angenähert werden, wobei T_t simulativ bestimmt werden kann. Exemplarisch sind zwei Sprungantworten in Abbildung 6-32 Sprungantworten der Regelstrecke sowie dessen Approximation zu sehen.

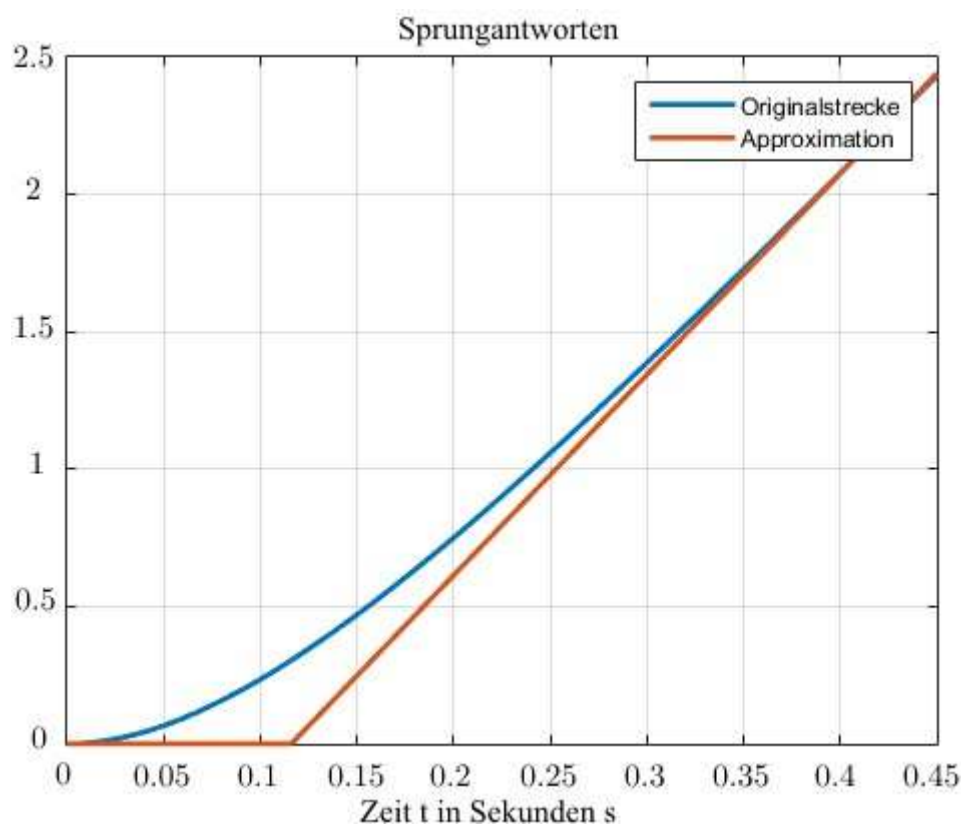


Abbildung 6-32 Sprungantworten der Regelstrecke sowie dessen Approximation

Der Reglerentwurf basiert auf dem Modell der approximierten Strecke, wobei ein P-Regler gewählt wird, da die Strecke bereits über einen I-Anteil verfügt und die Steigung der Sprungantwort konstant ist, sodass kein D-Anteil benötigt wird. Schließt man den P-Regler mit der Strecke $G_S(s)$ zur offenen Kette $G_0(s) = G_R(s) \cdot G_S(s)$ zusammen, erhält man für den Regelkreis die Übertragungsfunktion

$$H(s) = \frac{G_0(s)}{1 + G_0(s)}.$$

Es treten nun zusätzliche Pole $s_{R, \infty 1}, s_{R, \infty 2}, \dots, s_{R, \infty N}$ auf, wobei diese jedoch nicht > 0 und somit nicht instabil sind. Betrachtet man nun das approximierte System $\tilde{G}_S(s) \cdot e^{-jT_t s}$

erhält man konjugiert komplexe Pole $s_{\infty i}, \tilde{s}_{\infty i}^*$, sodass die Sprungantwort Schwingungen aufweist. Nun folgt ein Vergleich der Sprungantworten des geregelten Prozesses innerhalb eines Standardregelkreises sowie unter Verwendung des Smith-Prädiktors (vgl. Abbildung 6-33 Sprungantworten der Regelkreise). Um die Schwingungen zu erklären, erfolgt die Betrachtung der Pol- und Nullstellen der geschlossenen Regelkreise, wobei die konkreten Werte nicht von Relevanz sind.

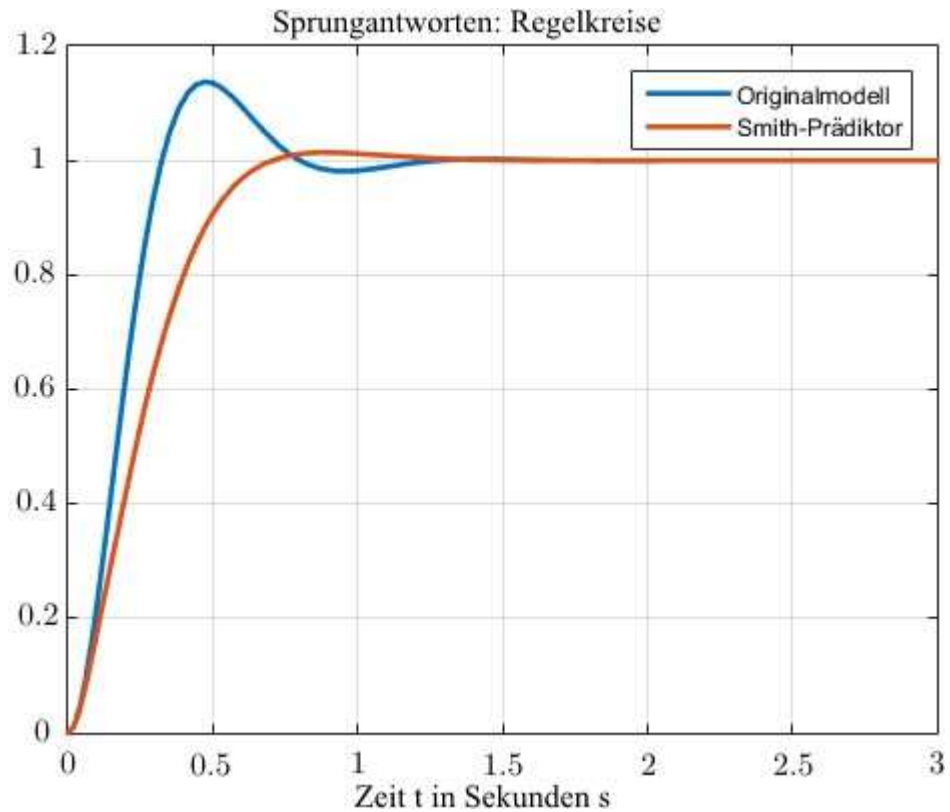


Abbildung 6-33 Sprungantworten der Regelkreise

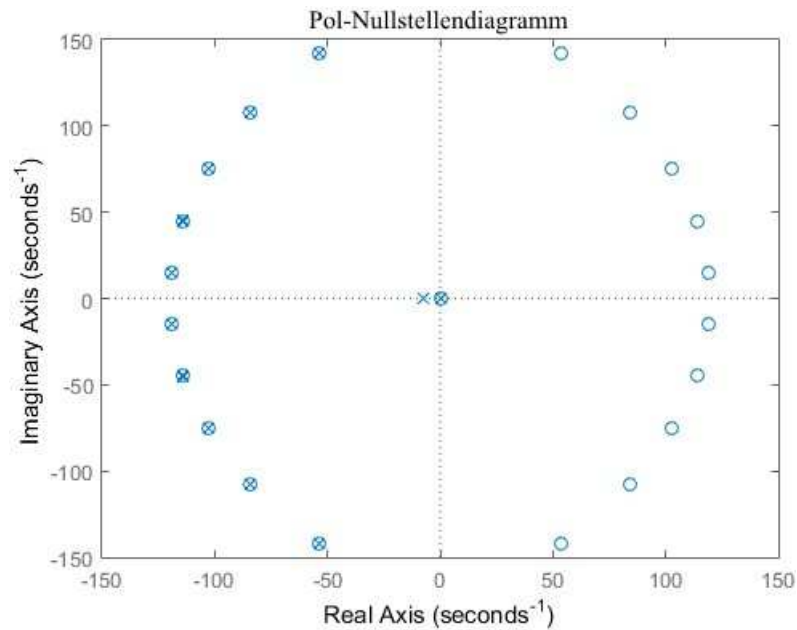


Abbildung 6-34 Pol-Nullstellen-Diagramm: Regelkreis mit Smith-Prädiktor

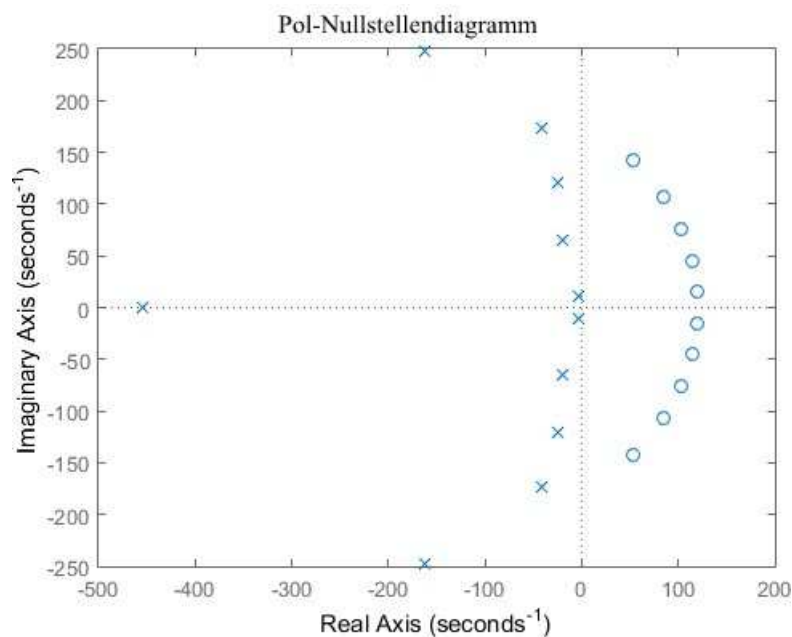


Abbildung 6-35 Pol-Nullstellen-Diagramm: Standardregelkreis mit P-Regler

Die Betrachtung des ersten Diagramms zeigt, dass die konjugiert komplexen Polstellen von den Nullstellen kompensiert werden, sodass der Schwingungsanteil relativ gering ausfällt. Der P-geschlossene Regelkreis besitzt mehrere konjugiert komplexe Polpaare, welche nicht vollständig kompensiert werden und somit einen maßgeblichen Schwingungsanteil hervorrufen. Die Pol- und Nullstellen entstehen durch eine Approximation der Totzeit durch eine Übertragungsfunktion, hier zehnter Ordnung. Es folgen nun Simulationen für verschiedene k_P . Die Simulationen der Abbildung 6-36, Abbildung 6-37

und Abbildung 6-38 zeigen die Ausgangsgrößen der Strecke bei Anwendung eines P-Reglers sowie des Smith-Prädiktors.

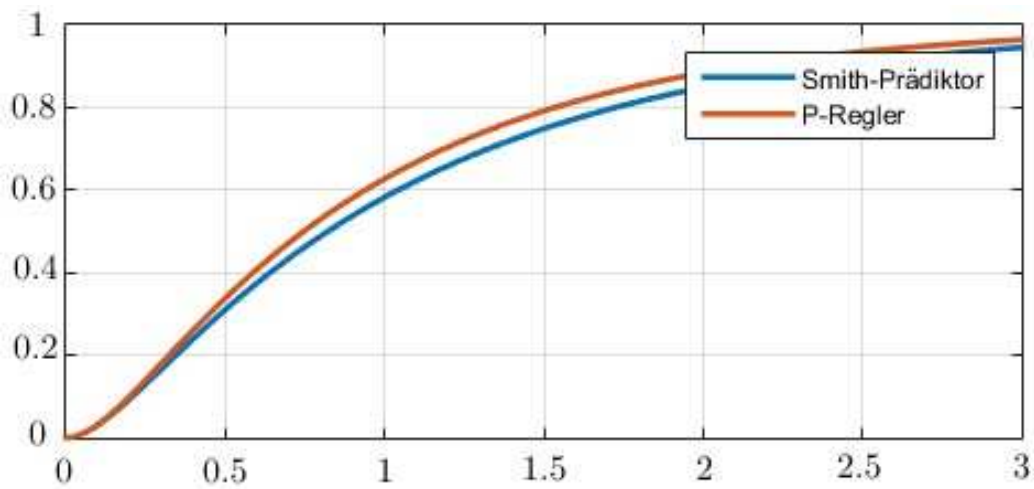


Abbildung 6-36 Simulation: Smith-Prädiktor, P-Regler für $k_P = 0.15$

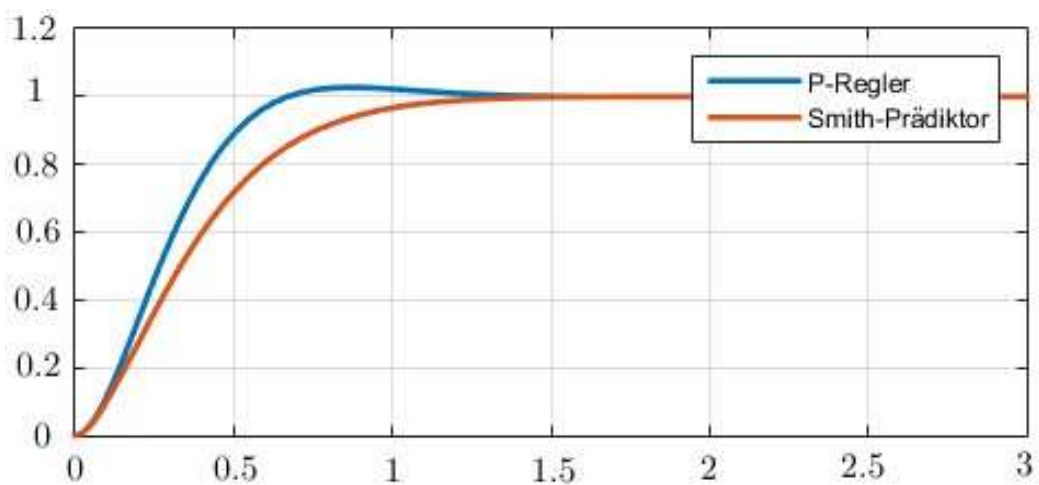
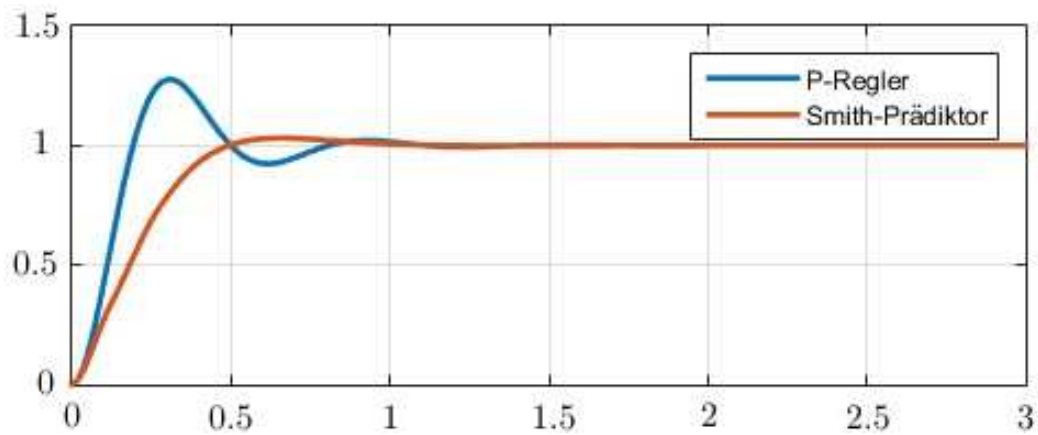
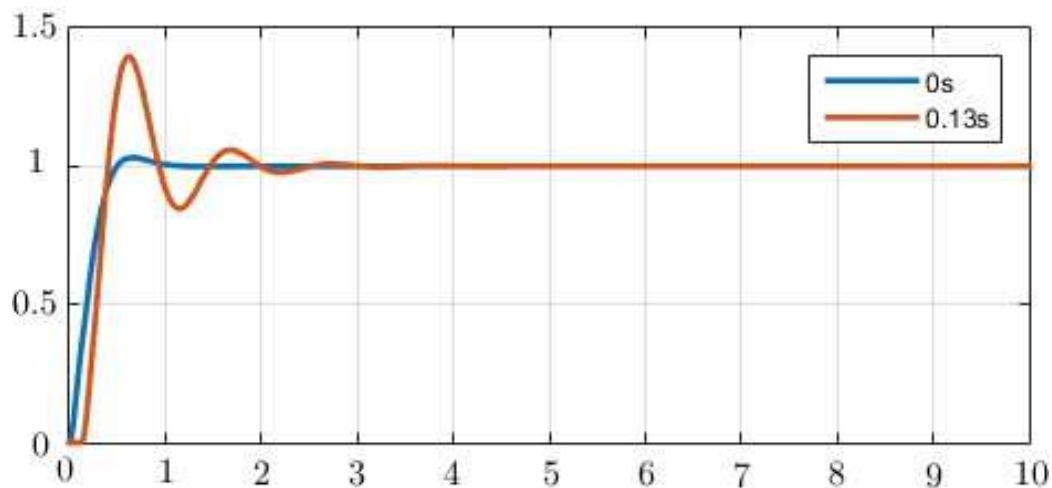


Abbildung 6-37 Simulation: Smith-Prädiktor, P-Regler für $k_P = 0.5$

Der Smith-Prädiktor zeigt für alle k_P einen schwingungsfreien Verlauf.

Abbildung 6-38 Simulation: Smith-Prädiktor, P-Regler für $k_P = 2$

Für große Verstärkungen ist dieser gegenüber den konventionellen Methoden geeigneter, da der Sollwert zusätzlich relativ schnell erreicht wird. Solange die gesamte Totzeit eines Prozesses nicht große Werte annimmt, ist die konventionelle Regelung für alle k_P , die den Regelkreis nicht zum Schwingen bringen, vorzuziehen.

Abbildung 6-39 Simulation: Smith-Prädiktor für $k_P = 2$

6.2.3 Zeitdifferenzen

Die folgenden Abbildungen zeigen Simulationen für Totzeitdifferenzen zwischen dem realen Prozess sowie der Modellierung. Mit Blick auf die

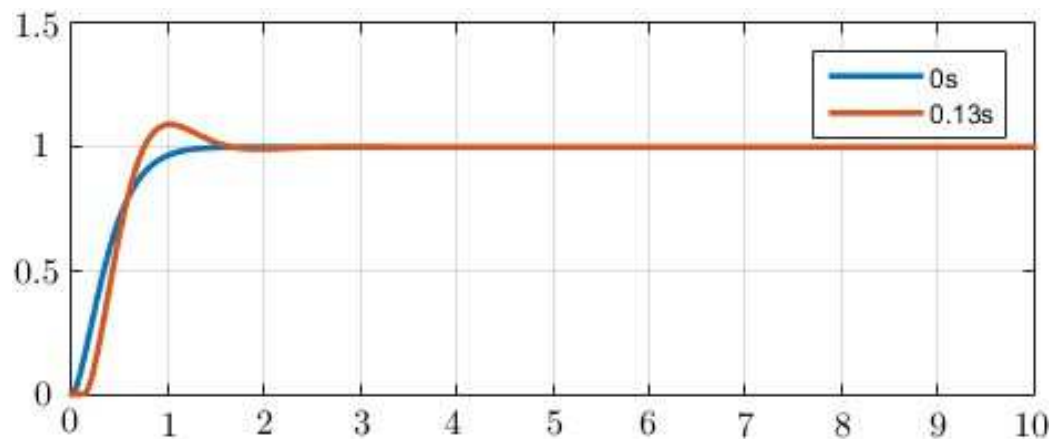
Abbildung 6-40 Simulation: Smith-Prädiktor für $k_P = 0.15$

Abbildung 6-39Abbildung 6-40 wird deutlich, dass sich die Regelung empfindlich auf die Zeitabschätzung verhält. Der Effekt wird für große Reglerverstärkungen größer.

6.2.4 Modellabweichungen

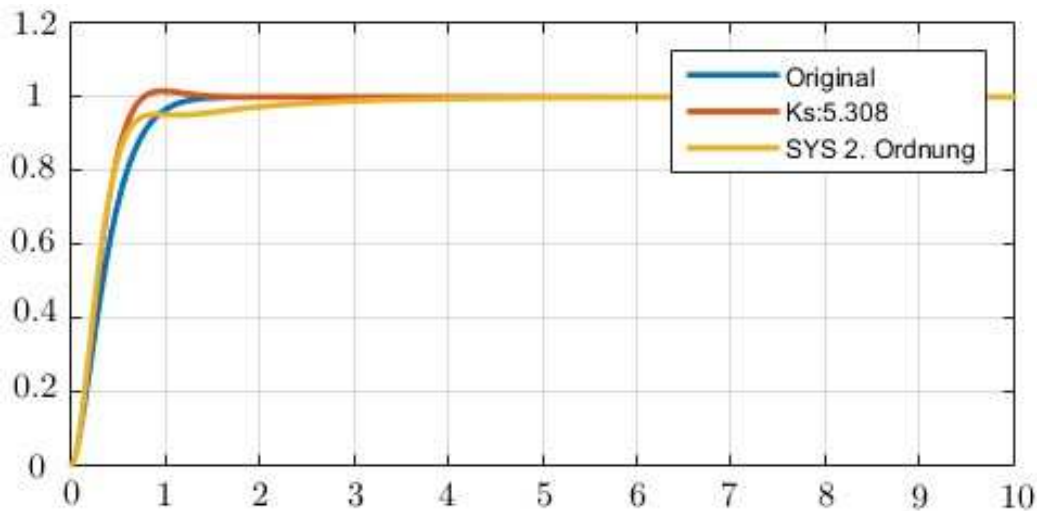
Nun wird das Verhalten für verschiedene Modellabweichungen untersucht. Es folgen Simulationen bei einer Fehleinschätzung der Systemordnung so wie der statischen Verstärkung. Die Simulationen erfolgen ausschließlich für den Smith-Prädiktor, wobei die Abweichungen relativ groß gewählt werden (vgl. Abbildung 6-41). Ein Vergleich zeigt, dass der Smith-Prädiktor trotz der Abweichung ein relativ gutes Verhalten zeigt.

6.2.5 Zusammenfassung

Die Anwendung des Smith-Prädiktors eignet sich nicht für kleine Totzeiten und kleine Reglerverstärkungen. Unter dieser Betrachtung reichen konventionelle Reglerentwurfsmethoden. Bei zunehmenden T_t und Modellunsicherheiten ist die Anwendung des Smith-Prädiktors anzuraten, wobei bei einer Unsicherheit der Totzeiteinschätzung geringe Reglerverstärkungen zu wählen sind.

6.2.6 Codegenerierung

Im Folgenden wird die C-Codegenerierung in der Simulationsumgebung *MATLAB/Simulink* beschrieben, wobei zusätzlich die Tools *MATLAB Coder* und *Simulink Coder* benötigt werden. Die Codeausführung geschieht in der

Abbildung 6-41 Simulation: Smith-Prädiktor für $k_P = 0.5$

Entwicklungsumgebung *CooCox CoIDE*. Die Generierung erfolgt exemplarisch an einem Modell, welches mit Hilfe eines Smith-Prädiktors geregelt wird und wird auf einem Mikrocontroller, hier dem *STM32F4-Discovery-Board*, getestet.

6.2.7 Simulinkmodell

Der Smith-Prädiktor wird im Folgenden für ein Modell aufgestellt, dessen Übertragungsfunktion durch

$$G_S(s) = \frac{7.308}{0.12s^2 + s}$$

gegeben ist. Das Modell kann durch die Funktion

$$\tilde{G}_S(s) = \frac{7.308}{s}$$

mit einer Totzeit von

$$T_t = 0.116s$$

approximiert werden. Die Sprungantworten sind der Abbildung 6-32 zu entnehmen. Für den Regler wird ein P-Regler gewählt, wobei dieser für $P=1$ betrachtet wird. Das entsprechende Simulinkmodell (vgl. Abbildung 6-42) wird für einen beliebigen konstanten Sollwert $w(t) = \bar{w}\sigma(t)$, hier $\bar{w} = 1$, betrachtet,

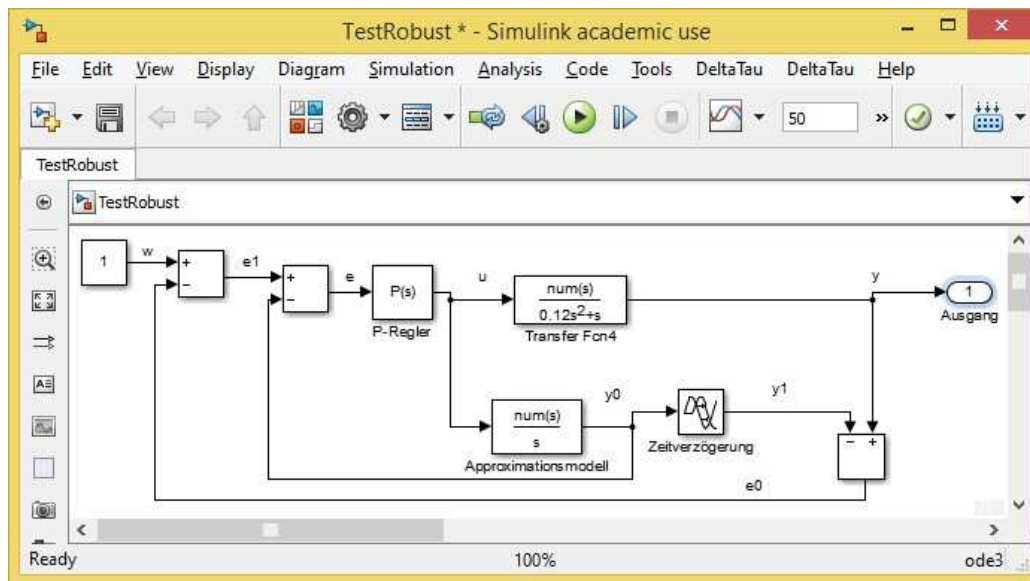


Abbildung 6-42 Simulinkmodell einer Smith-Prädiktor geregelten Regelstrecke mit Zeitverzögerung

wobei $\sigma(t)$ den Einheitssprung bezeichnet. Im Weiteren soll mit Hilfe des Ports *Ausgang* das Ausgangssignal des Regelkreises ausgegeben werden können. Um das Modell auf den Mikrocontroller laden zu können, ist es notwendig, die verwendeten Modelle zu diskretisieren. Dafür wird im Folgenden eine Abtastzeit von 0,001 gewählt. Die Abtastung kann mit Hilfe der MATLAB Funktion $[numd, dend] = bilinear(num, den, fs)$ ausgeführt werden, wobei *num* den Zähler und *den* den Nenner des zu diskretisierenden Modells bezeichnet. *fs* bezeichnet die Abtastfrequenz, sodass für das diskrete Modell der Zähler *numd*, sowie der Nenner *dend* ausgegeben werden.

Tabelle 6-1 Modelltransformation für eine Abtastzeit von $T = 0,001s$

Modell	kontinuierlich	zeitdiskret
$G_s(s)$	$\frac{7.308}{0.12s^2 + s}$	$\frac{3,037e - 5 + 3,028e - 5}{z^2 - 1,992z + 0,9917}$
$\tilde{G}(s)$	$\frac{7.308}{s}$	$\frac{0,007308}{z - 1}$
Verzögerung	$e^{-0.120s}$	z^{-120}

Nach Abschluss der Digitalisierung des Modells erhält man folgendes Modell(vgl. Abbildung 6-43). Für die Code Generierung ist jedoch zu beachten, dass keine Arithmetischen Schleifen in das System eingebaut werden. Diese verhindern eine Generierung

des Modells. In dem vorliegenden Fall wurden diese durch ein Verzögerungsglied 1. Ordnung eliminiert. Das Modell wird nach erfolgreicher Code Generierung in Matlab/Simulink in die CoxCox Umgebung eingebaut und getestet. Weitere Informationen zum Test des Systems sind dem Kapitel 9. Test und Validierung zu entnehmen.

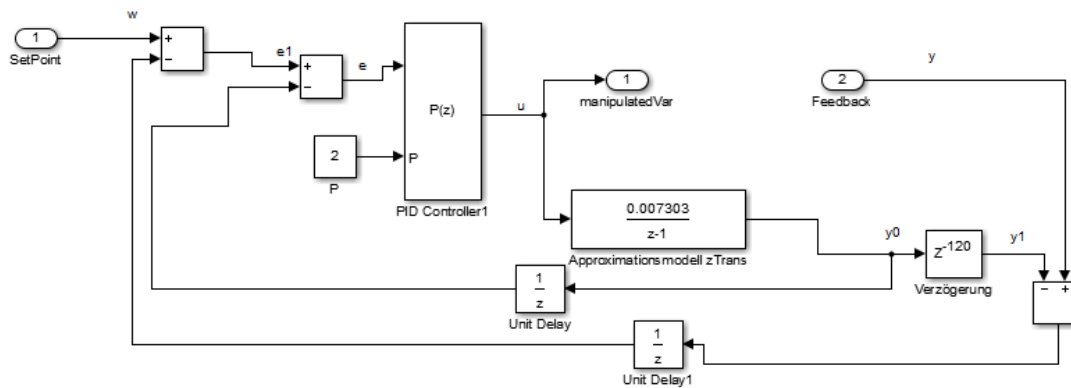


Abbildung 6-43 Digitales Smith-Prädiktor Modell

6.3 ASA-Controller

Die Funktionsweise des ASA-Controller (Antisystem-Approach) kann der Literatur (Zacher⁸) entnommen werden.

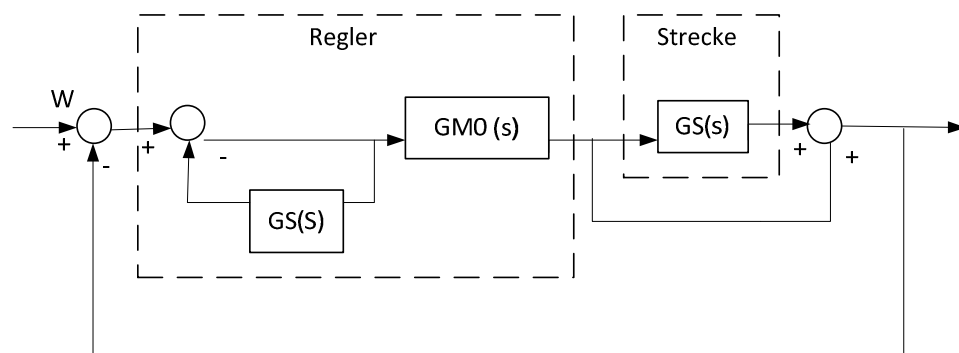


Abbildung 6-44 Struktur des ASA-Reglers

6.3.1 Parameterberechnung

Der Regelstrecke soll mit PT_1 -Verhalten in 1s das Ziel erreichen. Daraus ergibt sich ein T_1 von ca. 0,2s.

⁸ Zacher, Serge; Reuter, Manfred (14. Auflage, 2014): Regelungstechnik für Ingenieure. S.364

$$G_M(s) = \frac{G_{0M}(s)}{1+G_{0M}(s)} = \frac{1}{1+0,2*s} \text{ daraus folgt:}$$

$$G_{0M}(s) = \frac{G_M(s)}{1 - G_M(s)} = \frac{\frac{1}{1 + 0,2 * s}}{1 - \frac{1}{1 + 0,2 * s}} = \frac{5}{s}$$

Damit ergibt sich I-Verhalten der Übertragungsfunktion.

Damit das gewünschte Verhalten bei der gegebenen IT₁-Strecke realisierbar ist, muss $k_R \leq k_s * y_{max}$ gelten. (die maximale Verfahrgeschwindigkeit sollte nicht überschritten werden). Die Stellgröße ist bei 12V Versorgungsspannung maximal $\pm 10V$. Dies ist mit $5 \leq 7,3 * 10$ erfüllt. Hier dürfte Spielraum für künftige Optimierungen bestehen.

6.3.2 Simulation

Nach der Parameterberechnung lässt sich ein Modell des Reglers erstellen.

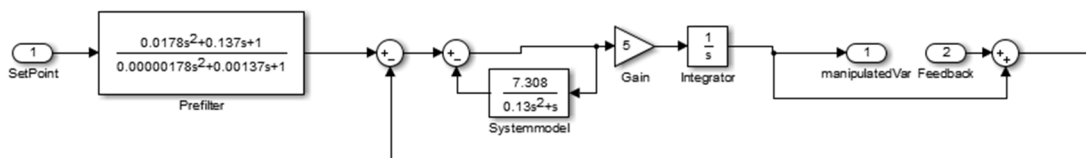


Abbildung 6-45 Struktur des ASA-Reglers in der Simulation

Zunächst wird auf den Vorfilter verzichtet und die Sprungantwort ermittelt. Es zeigt sich ein leichtes Überschwingen, dass beim Regeln auf Positionen in der Nähe des Sprungbereichs kritisch werden könnte. Daher wird versucht, über einen D-Vorfilter die „Erstreaktion“ zu verstärken, um dann langsamer ins Ziel einzulaufen.

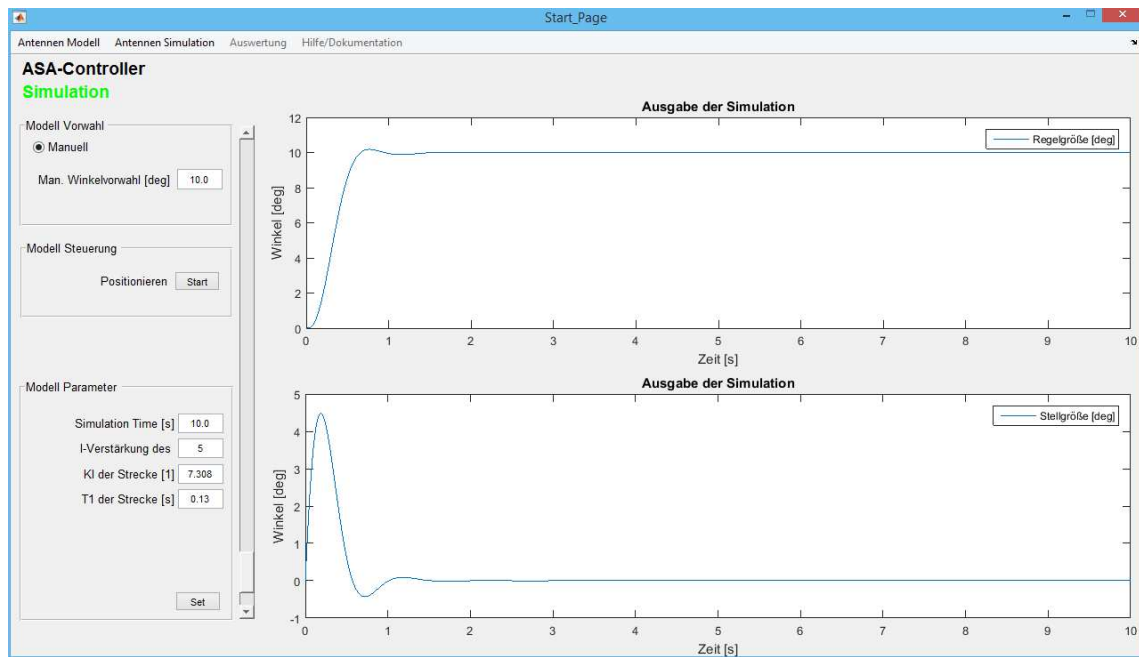


Abbildung 6-46 Sprungantwort des ASA-Reglers ohne Vorfilter

Der Vorfilter wird wie folgt dimensioniert:

$$G_F(s) = \frac{1 + G_S(s)}{G_S(s)} = \frac{1 + \frac{7,3}{0,13 * s^2 + s}}{\frac{7,3}{0,13 * s^2 + s}} = \frac{7,3 + s + 0,13 * s^2}{7,3}$$

$$= 1 + 0,137 * s + 0,0178 * s^2$$

Reine D-Glieder sind nicht realisierbar, also Nenner mit um den Faktor 100 kleineren Zeitwerten einführen.

$$G_F(s) = \frac{1 + 0,137 * s + 0,0178 * s^2}{1 + 0,00137 * s + 0,00000178 * s^2}$$

Der Vorfilter erreicht die gewünschte Wirkung und der Regler schwingt nicht mehr über.

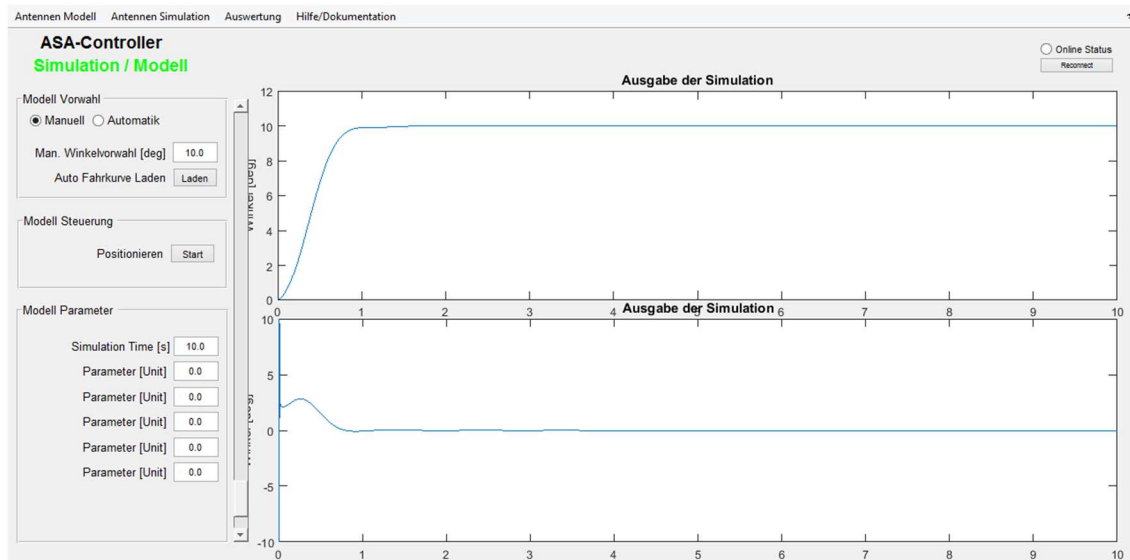


Abbildung 6-47 Sprungantwort des ASA-Reglers mit D-Vorfilter

6.3.3 Diskretisierung

Die Zykluszeit der digitalen Regelung ist mit 1ms deutlich kleiner als die kleinste Zeitkonstante des Regelkreises. Damit kann die digitale Regelung als quasikontinuierlich angenommen werden.

Zur Diskretisierung wird daher die Übertragungsfunktion in den Zeitbereich transformiert.

$$\text{Strecke: } y(t) = 7,303 * \int u(t)dt - T * \frac{dy}{dt}$$

$$\text{GM0: } y(t) = 5 * \int u(t) dt$$

Der Vorfilter sorgt für Instabilität des diskretisierten Regelkreises, da hier Zeitkonstanten kleiner der Abtastzeit vorkommen. Daher wird dieser Filter entfernt.

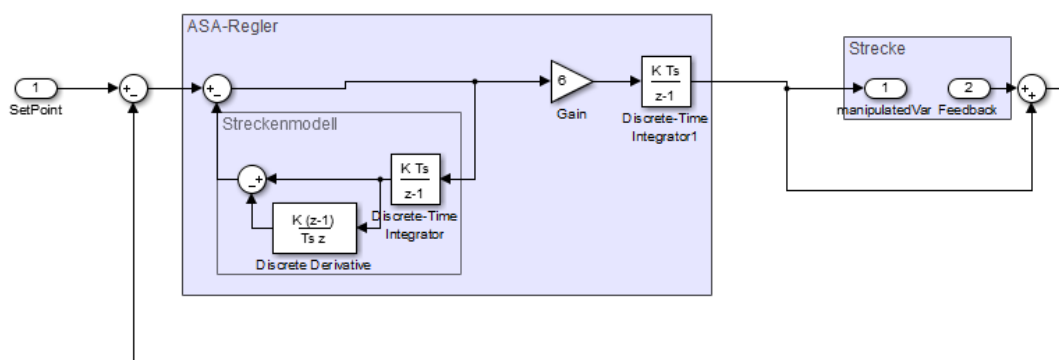


Abbildung 6-48 diskretes Matlab-Modell des ASA-Reglers

6.3.4 Verhalten am Antennenmodell

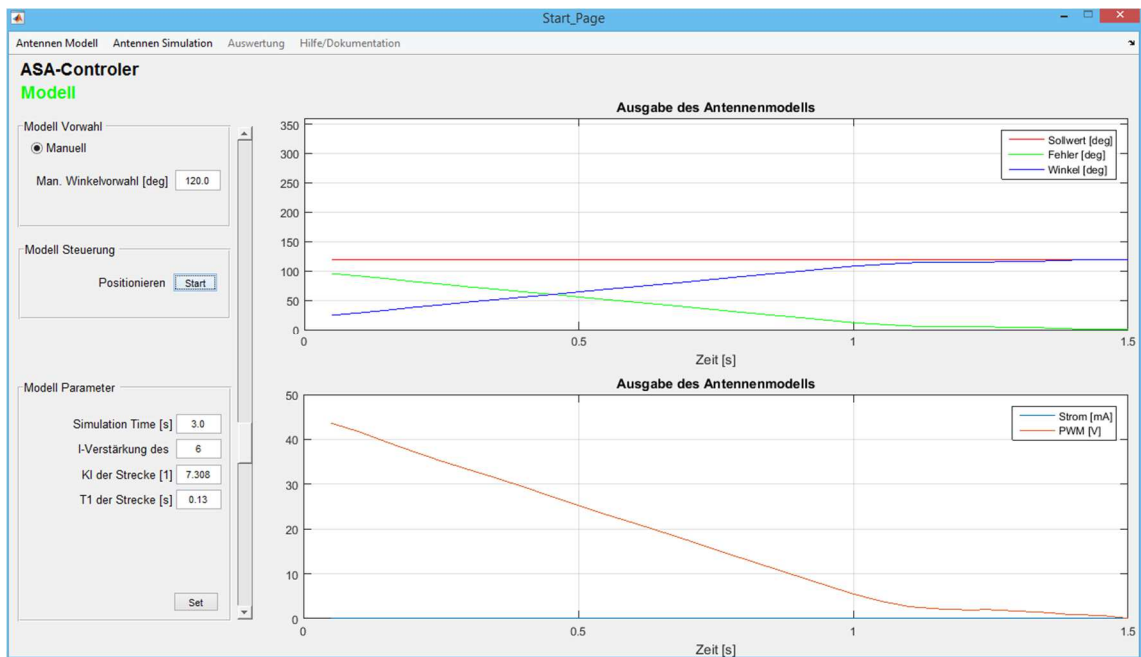


Abbildung 6-49 Sprungantwort des ASA-Reglers

Der Regler führt die Sollposition sauber an, schleicht allerdings ins Ziel. Ein Verkleinern der Zeitkonstante im gewünschten PT_1 -Verhalten sollte dies verbessern. Hier ist ein Spielraum um mehr als den Faktor 10. Mit einem Zehntel der Zeitkonstanten ergibt sich eine deutliche Verringerung der Positionierzeit um 0,35s. Die Verfahrensgeschwindigkeit wird in beiden Fällen durch die Versorgungsspannung des Motors, die als Stellgrößenbegrenzung wirkt, begrenzt.

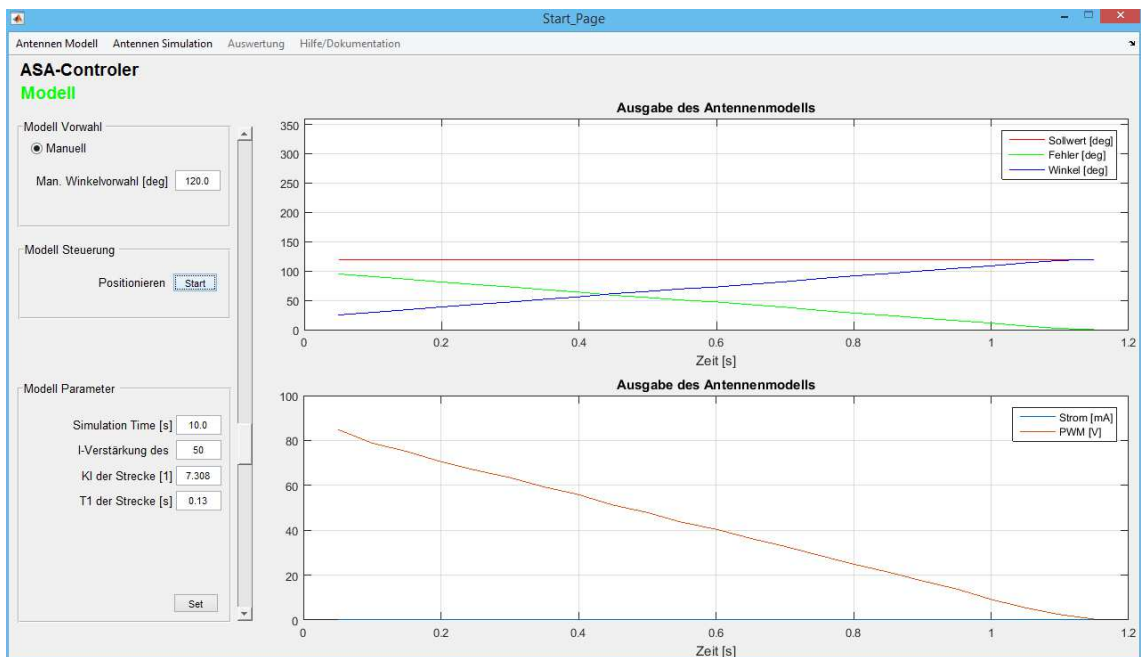


Abbildung 6-50 stark optimierter ASA-Regler

Bei Versuchen eine Zielposition nahe 0° anzufahren, zeigt sich ein Verhalten ähnlich eines Überschwingens über die Sprungstelle (langsamerwerden, und anschließend wieder Beschleunigung auf volles Tempo). Eine Analyse der Winkelmesswerte zeigt, dass nie ein Winkel kleiner $15,6^\circ$ gemessen wird. Die Ursache dieses Messfehlers konnte nicht mehr geklärt werden, da der Fehler zu spät im Projektverlauf identifiziert wurde. Als Abhilfe wird der anfahrbare Bereich in der Spezifikation geändert.

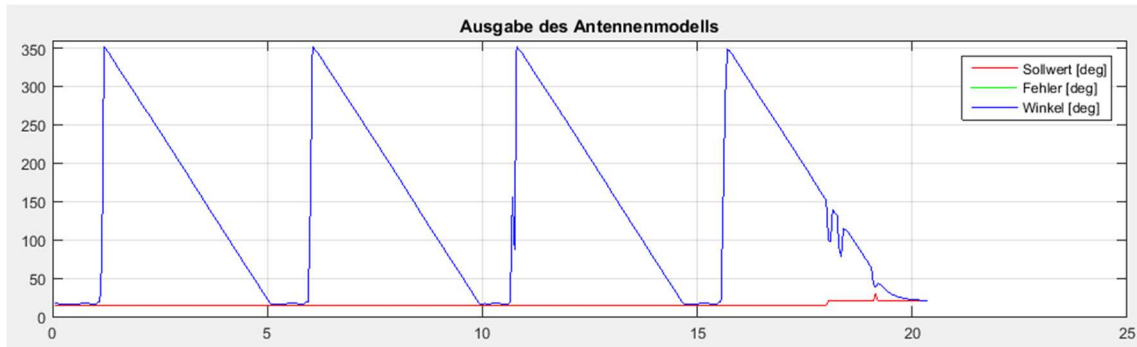


Abbildung 6-51 Abweichung der Winkelmessung (der unregelmässige Signalverlauf am letzten Sägezahn stammt von einem Reset)

6.4 PFC-Regler

6.4.1 Funktionsweise des PFC-Reglers

Predictive Functional Control ist eine Methode zur prädiktiven (vorausschauenden) Regelung von Prozessen. Die Idee der prädiktiven PFC-Regelung wurde in den 70er Jahren von Jacques Richalet für P-Strecken 1. Ordnung mit Verzögerung vorgeschlagen und anschließend in seinem Buch „Predictive Functional Control, Principles and Industrial Applications (Advances in Industrial Control)“ weiterentwickelt.

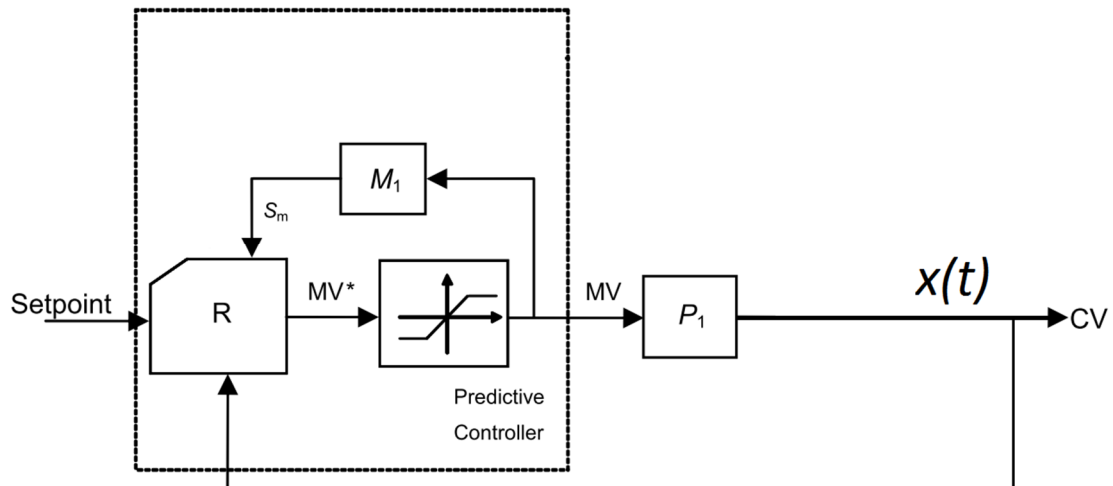


Abbildung 6-52: Schema der PFC-Regelung [Richalet]

Die aktuelle Sprungantwort der Regelstrecke $x(t)$ wird an die Sprungantwort eines vorher gegebenen dynamischen Modells $x_M(t)$

$$T_M \cdot \frac{dx_M(t)}{dt} + x_M(t) = K_{PM} \cdot \hat{y}$$

angepasst.

Die Lösung dieser Gleichung gilt für die gesamte Ausregelzeit T_{Aus} als gewünschte Sprungantwort, die in n Zeitabschnitten (Prädiktionshorizonte) der Länge T_h aufgeteilt wird. Daraus wird die rekursive Formel zur Berechnung des Modellausgangs hergeleitet

$$x_{Mk+h} = \alpha \cdot x_{Mk} + (1 - \alpha) \cdot K_{PM} \cdot y_k$$

wobei $\alpha = e^{-\frac{T_h}{T_{Aus}}}$ ist.

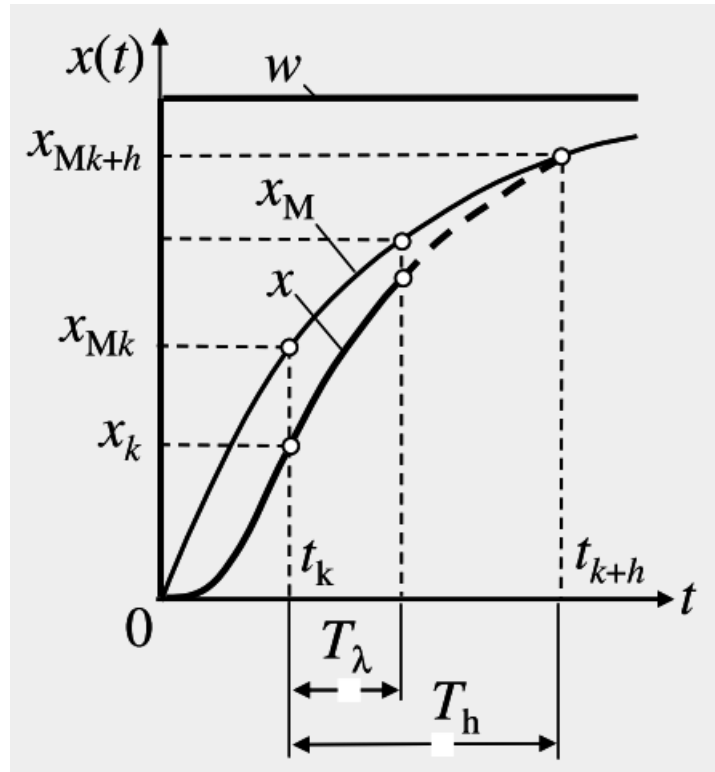


Abbildung 6-53: PFC-Algorithmus

In anderen Worten, zum Abschluss jedes vorherigen Prädiktionshorizonts soll die ausgerechnete Regelgröße x_{Mk} mit dem aktuellen Istwert x_k verglichen werden. Daraus wird die neue Stellgröße y_k für den nächsten Prädiktionshorizont berechnet.

Der Vorteil dieses Verfahrens besteht also darin, dass die möglichen Abweichungen der Modellparameter T_M und K_{PM} von den reellen Parametern am Ende jedes Prädiktionshorizonts erkannt und durch eine geänderte Stellgröße ausgeglichen werden. Die Wahl des Parameters T_λ wird dem Entwickler überlassen. Davon hängt die Wahl des Parameters λ

$$\lambda = e^{-\frac{T_\lambda}{T_{Aus}}}$$

ab und folglich der Stellgröße

$$y_k = \frac{1}{K_{PM}} \cdot \left(x_{Mk} + \frac{1-\lambda}{1-\alpha} \cdot (w - x_k) \right) \quad - \text{PFC-Algorithmus}$$

Für die optimalen Verhältnisse wird es in der Literatur empfohlen:

$$T_\lambda = \frac{1}{5} T_h \quad \text{oder} \quad T_\lambda = \frac{1}{3} T_h$$

6.4.2 Auslegung für IT₁-Prozess

Für die Implementierung und Initialisierung des PFC-Reglers ist es notwendig, dass sowohl der geschlossene Regelkreis als auch der Regler stabil sind. Bei der modellbasierten Regelung, zu der auch die PFC-Regelung zählt, ist die Regelstrecke ein Bestandteil des Reglers. Handelt es sich also bei der zu regelnden Strecke um eine Strecke mit Integralverhalten (wie es in dieser Arbeit der Fall ist), so muss das Streckenmodell M_0 durch Zerlegung in drei stabile Übertragungsglieder M_{11} , M_{12} , M_2 „stabilisiert“ werden.

$$M_0 = \frac{M_1}{1 + M_2}$$

$$M_{11} = \frac{K_1}{1 + s \cdot T}, \quad M_{12} = \frac{K_2}{1 + s \cdot T_{dec}}, \quad M_2 = -\frac{1}{1 + s \cdot T_{dec}}$$

$$K_2 = \frac{K_P \cdot T_{dec}^2}{T_{dec} - T}, \quad K_1 = K_P \cdot T_{dec} - K_2, \quad K_3 = 1, \quad \text{wobei } T_{dec} = \frac{T_{Aus}}{3}$$

Die Glieder M_{11} und M_{12} werden dabei im Vorlauf des Stellsignals und M_2 im Rückkopplungsweig platziert.

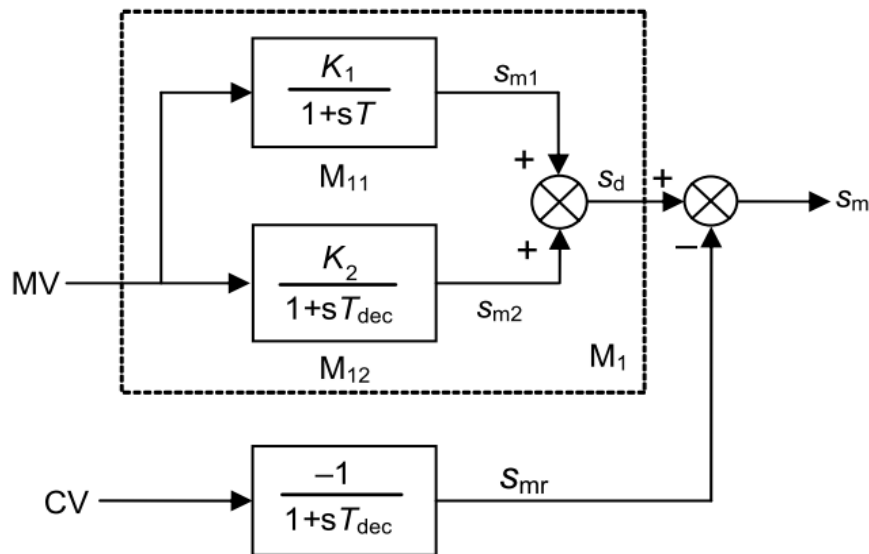


Abbildung 6-54: Zerlegung des IT₁-Modells

Die komplette Gleichung des Modells ist gegeben durch:

$$s_m(k) = s_{m1}(k) + s_{m2}(k) + s_{mr}(k),$$

$$\text{Mit } s_{m1}(k) = a_m \cdot s_{m1}(k-1) + b_m \cdot K_1 \cdot e(k-1),$$

$$s_{m2}(k) = a_s \cdot s_{m2}(k-1) + b_s \cdot K_2 \cdot e(k-1)$$

$$s_{mr}(k) = a_s \cdot s_{m3}(k-1) + b_s \cdot K_3 \cdot spp(k-1)$$

wobei e – Stellsignal, spp – Antwort der Strecke,

$$a_s = e^{-\frac{T_s}{T_{dec}}} \quad , \quad b_s = 1 - a_s \quad , \quad a_m = e^{-\frac{T_s}{T_{Strecke}}} \quad , \quad b_m = 1 - a_m$$

Die Gleichung für die Berechnung des Stellsignals lautet:

$$e(k) = \frac{(SW - spp(k)) \cdot l_h + s_{m1}(k) \cdot b_{mh} + s_{m2}(k) \cdot b_{sh} + s_{mr}(k) \cdot b_{sh} - spp(k) \cdot b_{sh} \cdot K_3}{K_1 \cdot b_{mh} + K_2 \cdot b_{sh}}$$

$$SW - \text{Sollwert}, \quad b_{mh} = 1 - a_m^h, \quad b_{sh} = 1 - a_s^h$$

$$l_h = 1 - \lambda^h = 1 - e^{-\frac{T_s \cdot 3 \cdot h}{T_{Aus}}} \quad - \text{Referenztrajektorie mit Prädiktionshorizont } h$$

6.4.3 Simulation

Die Regelstrecke ist durch ein IT₁-Glieder gegeben, dessen Übertragungsfunktion folgendermaßen aussieht:

$$G_S(s) = \frac{K_S}{s \cdot (s \cdot T_S + 1)} = \frac{7,308}{s \cdot (0,13s + 1)}$$

Um den PFC-Regler optimal einzustellen muss der Regelkreis mit Hilfe von Simulink getestet werden. Da der rekursive Algorithmus unter Simulink eher umständlich umzusetzen ist und das Simulink-Blockschaltbild dabei schnell unübersichtlich wird, wird der eigentliche Regler in MATLAB-Programmiersprache implementiert. Der Quelltext kann in einen Simulink-Block „MATLAB Function“ eingefügt und somit im Simulink-Modell verwendet werden.

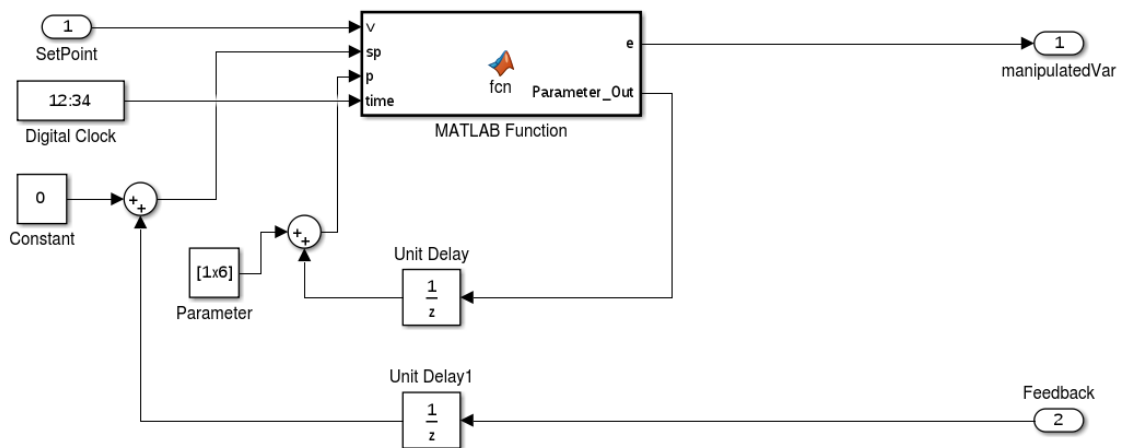


Abbildung 6-55: PFC-Regler im Simulink-Modell

Der PFC-Algorithmus wurde nun für die gegebene Strecke simuliert. Dazu wurde das System durch einen Sollwertsprung von 0 auf die Werte zwischen 0 und 360° ab dem Zeitpunkt $t=0$ getestet.

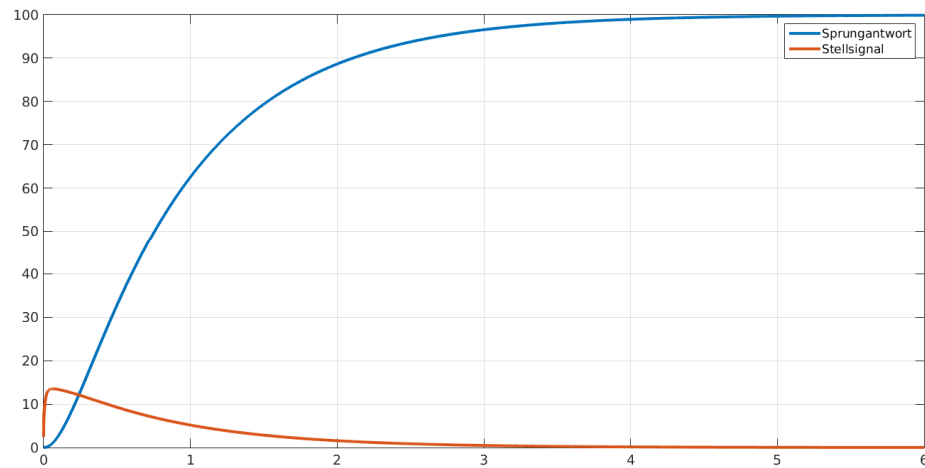


Abbildung 6-56: PFC-Regelung – Sprungantwort und Stellsignal bei Sollwert 100°

Für die Spannungsversorgung des Stellgliedes (Motor) sind maximal 10 Volt zulässig, so dass die Stellgröße mit einem Saturation-Block aus der Simulink-Bibliothek auf diesen Wert begrenzt werden muss.

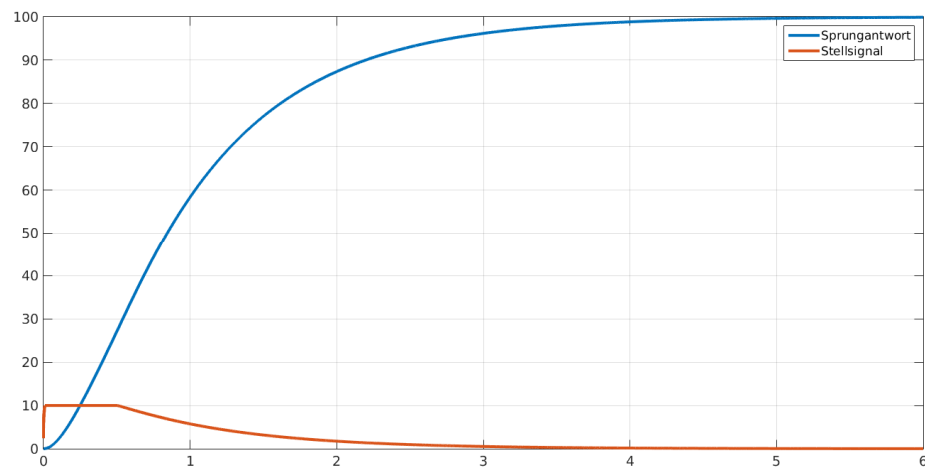


Abbildung 6-57: Signalverlauf mit Stellsignalbegrenzung

6.4.4 Zusammenfassung

Der PFC-Regler eignet sich auch für die Regelung der Strecken mit Integralverhalten. Durch Auswahl der Parameter für T_{Aus} , T_h und T_M kann das Verhalten des Regelkreises beeinflusst werden. Der Zusammenhang zwischen diesen Parametern ist jedoch undurchsichtig und muss während der Simulationsläufe bestimmt werden.

6.5 SPFC-Regler

6.5.1 Funktionsweise des SPFC-Reglers (Simplified PFC)

Der PFC-Regelalgorithmus aus dem vorigen Kapitel kann vereinfacht werden, in dem der Anpassungsschritt und der Prädiktionshorizont gleich gesetzt werden, d.h. $T_\lambda = T_h$.

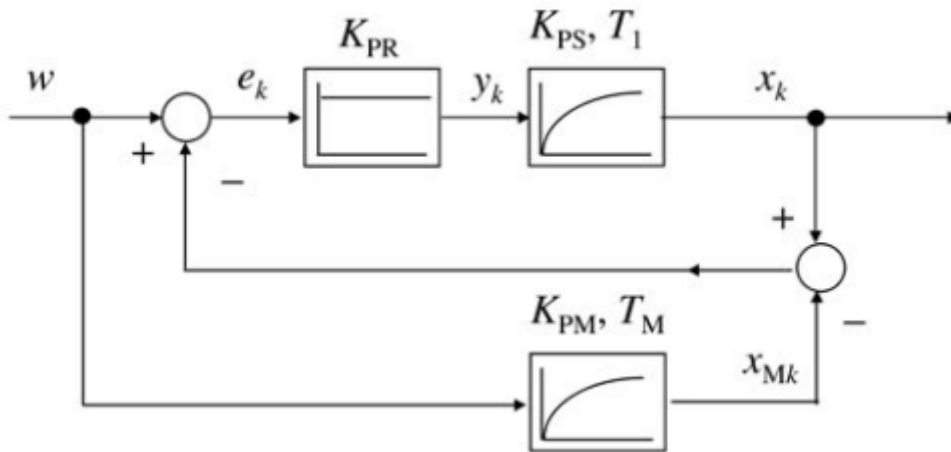


Abbildung 6-58: Wirkungsplan der vereinfachten PFC-Regelung

Die Stellgröße wird nach einer einfacheren rekursiven Formel berechnet:

$$y_k = \frac{1}{K_{PM}} [w - (x_k - x_{Mk})]$$

Die Übertragungsfunktion $G_w(s)$ des geschlossenen Kreises wird dann wie folgt bestimmt:

$$G_w(s) = [1 + G_M(s)] \frac{G_O(s)}{1 + G_O(s)},$$

wobei $G_O(s) = G_R(s) \cdot G_S(s)$

Das SPFC-Verfahren stellt somit die Regelung im geschlossenen Regelkreis mit einem Vorfilter $G_V(s)$ dar. Wird ein bestimmtes Verhalten $G_w(s)$ des geschlossenen Regelkreises gewünscht, so kann die Übertragungsfunktion $G_V(s)$ des Vorfilters wie folgt errechnet werden:

$$G_V(s) = G_W(s) \frac{1 + G_O(s)}{G_O(s)},$$

wobei $G_V(s) = 1 + G_M(s)$

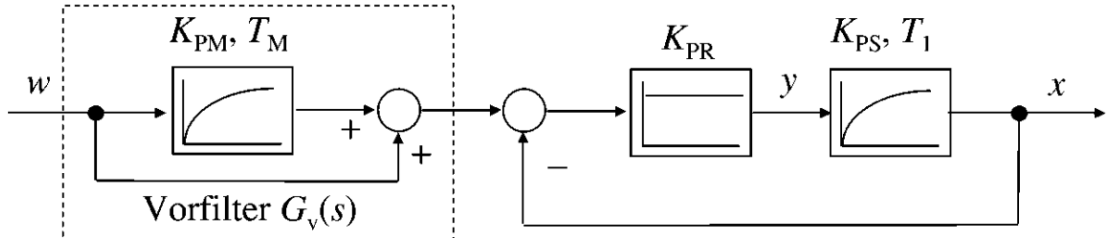


Abbildung 6-59: SPFC-Verfahren als Regelung mit einem Vorfilter

6.5.2 Simulation

Wie bereits erwähnt, ist die Regelstrecke durch ein IT₁-Glieder gegeben, dessen Übertragungsfunktion folgendermaßen aussieht:

$$G_S(s) = \frac{K_S}{s \cdot (s \cdot T_S + 1)} = \frac{7,308}{s \cdot (0,13s + 1)}$$

Der Vorfilter zur Regelung dieser IT₁-Strecke kann allgemein nach folgender Formel berechnet werden:

$$G_V(s) = G_W(s) \frac{K_S \cdot K_P + s \cdot (s \cdot T_S + 1)}{K_S \cdot K_P}$$

Für die Übertragungsfunktion des geschlossenen Regelkreises wird eine PT₂-Struktur gewählt.

$$G_W(s) = \frac{K_W}{(1 + s \cdot T_W)^2}$$

$$G_V(s) = \frac{K_W}{(1 + s \cdot T_W)^2} \cdot \frac{K_S \cdot K_P + s \cdot (s \cdot T_S + 1)}{K_S \cdot K_P}$$

Für $K_W = K_S \cdot K_P = 1$, $T_W = 0,3$ ergibt sich der Vorfilter zu: $G_V(s) = \frac{0,13s^2 + s + 1}{(1 + s \cdot 0,3)^2}$

Die Signalverläufe des Vorfilters, der Stellgröße sowie der Sprungantwort sind in der nachfolgenden Abbildung zu sehen.

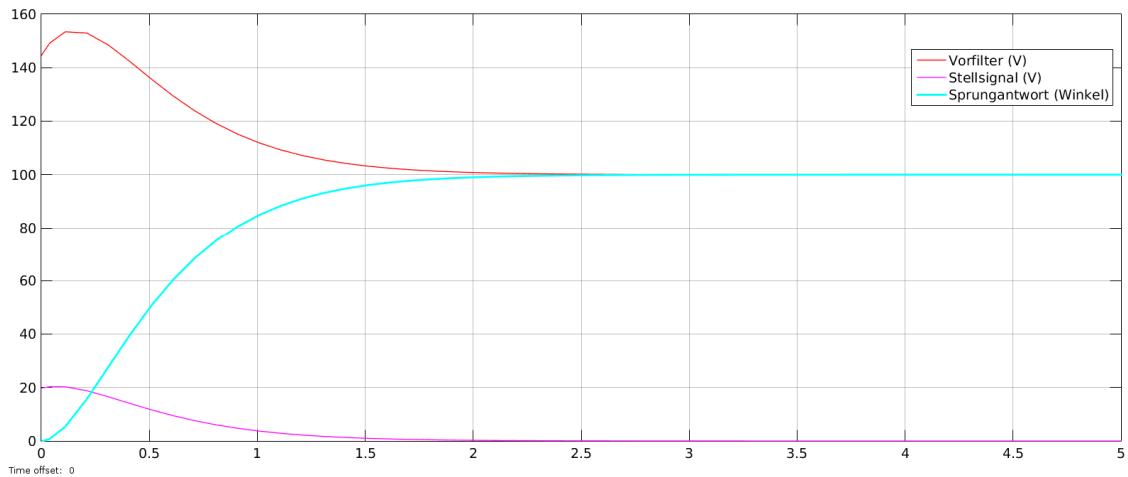


Abbildung 6-60: Vorfilter-, Stellgrößen- und Sprungantwortverlauf

Für die Spannungsversorgung des Stellgliedes (Motor) sind maximal 10 Volt zulässig, so dass die Stellgröße mit einem Saturation-Block aus der Simulink-Bibliothek auf diesen Wert begrenzt werden muss.

Die Änderung der Signalverläufe ist unten zu sehen.

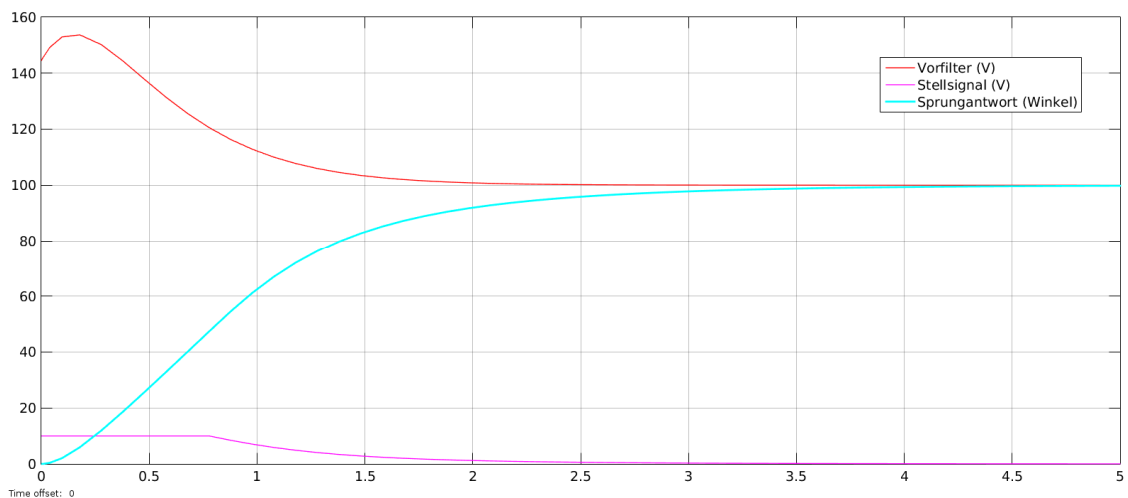


Abbildung 6-61: Signalverlauf nach der Stellgrößenbegrenzung

Für die Generierung des C-Codes muss der Regler nun in den Zeitbereich überführt werden. Auf eine Z-Transformation kann hier verzichtet werden, da die Abtastzeit klein im Vergleich zu der Zeitkonstante der Strecke gewählt ist. Die Regelung kann daher als quasikontinuierlich betrachtet werden.

$$y_R(t) = 0.18 \cdot e^{\frac{t}{0.3}} + 8.29 \cdot t \cdot e^{\frac{t}{0.3}}$$

6.5.3 Zusammenfassung

Mit einem SPFC-Regler kann die Führungsübertragungsfunktion des Gesamtsystems an das gewünschte Verhalten angepasst werden. Durch eine geeignete Wahl der Parameter des geschlossenen Kreises kann die Regelgeschwindigkeit vorgegeben werden. Es ist jedoch zu beachten, dass die schnellere Regelung ein härteres Eingreifen des Stellsignals erfordert. Wird das Stellsignal durch physikalische Einschränkungen des Stellgliedes begrenzt, so kann die Regelung insgesamt langsamer werden. Dadurch weicht der Signalverlauf von dem gewünschten Verhalten ab.

7 SIL und PIL

Bei SIL wird C Code für den Host PC kompiliert und läuft auch auf dem Host PC. Bei PIL wird der C-Code cross kompiliert und läuft auf dem Hardware Target. Beide Simulation laufen in Non Real Time bzw. nicht in Realzeit. Dabei erhalten die SIL/PIL Blöcke Signale vom TOP Modell und laufen auf dem Host PC (SIL) oder auf dem Hardware Prozessor (PIL). Zu Beginn jeder Abtastperiode werden Signale vom Top Modell zu SIL/PIL Blöcken gesendet und verarbeitet.

Das SIL und PIL eignet sich zu Testzwecken im Vorstadium zum realen Testen. Insbesondere bei PIL bietet sich die in the Loop Simulation an, da man einen Test mit dem realen Controller durchführt. Man könnte z. B. die gesamten Regelkomponenten bis auf die Strecke in ein Subsystem packen und für dieses Code generieren. Danach führt man die in the Loop Simulation in Simulink durch und testet die Regelung. Der generierte Code für das Subsystem kann dabei als Black Box betrachtet werden, bei der nur die Ein- und Ausgänge nach außen von Interesse sind. Es ist somit nicht nötig den kompletten Quellcode des Reglers oder der Regler zu analysieren. Eigentlich braucht man später dann nur noch die simulierte Strecke gegen die reale Strecke auszutauschen und die Schnittstellen nach außen entsprechend zu beschalten, z. B mit der Istposition und das reale Verhalten zu beobachten und den Regler daraufhin eventuell entsprechend anzupassen.

Die Simulationen für SIL als auch für PIL werden in dieser Arbeit unter Simulink im Normal Status durchgeführt.

7.1 SIL (Software in the Loop): C-Code der Regler mit Simulink Coder generieren und testen

Zur Codegenerierung wird das folgende in Kapitel 5.5 entwickelte Modell herangezogen. Bei dem Modell ist von einer Zeitkonstanten der Strecke von 0,13 sec. ausgegangen worden. Der Wert des Ki der Strecke beträgt rund 0,06. Daher wurden bei dem Kompensationsregler im Gegensatz zu Kapitel 6 ein KPR von 82.1018⁹ und ein TV von 0,13sec. eingesetzt. Das Systemmodell aus Kapitel 5.5 wird unter dem Dateinamen „Modell_ztrans_ohne_Visu.slx“ gespeichert. Aus dem Controller Block (Subsystem),

⁹ Es ist bei dem Modell aus Kapitel 5.5 vorerst von einer Referenzspannung und somit Poti Spannung von 3 Volt ausgegangen worden. Die Referenzspannung wurde nachher auf 2,95 Volt korrigiert. Für das Verhalten des Kompensationsreglers hat dies aber kaum einen Einfluss, somit kann dies vernachlässigt werden. Es spielt für das Verhalten des Kompensationsreglers kaum eine Rolle ob das Ki der Strecke nun 0,061 oder 0,59 beträgt.

wird dabei Code für das SIL generiert. Dabei wird das Modell zuerst ohne Anbindung an die Visualisierung betrachtet.

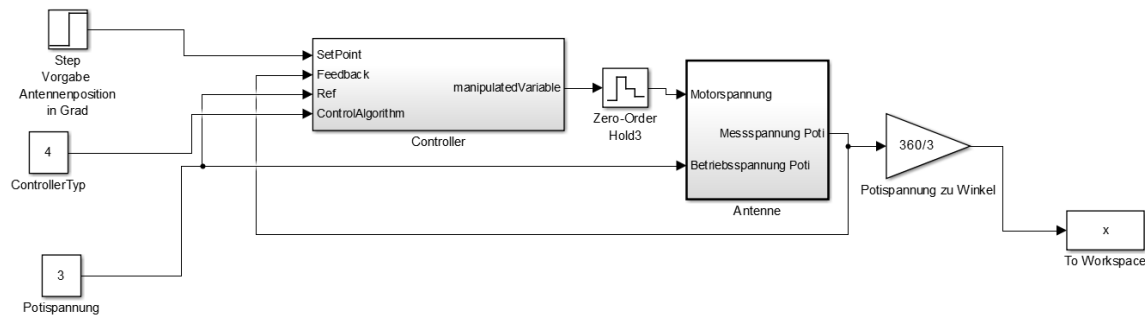


Abbildung 7-1 Systemmodell aus Kapitel 5.5

Dabei sollte man auf folgende Einstellung achten:

Rechter Mausklick auf das Subsystem Controller → Block Parameters (Subsystem)

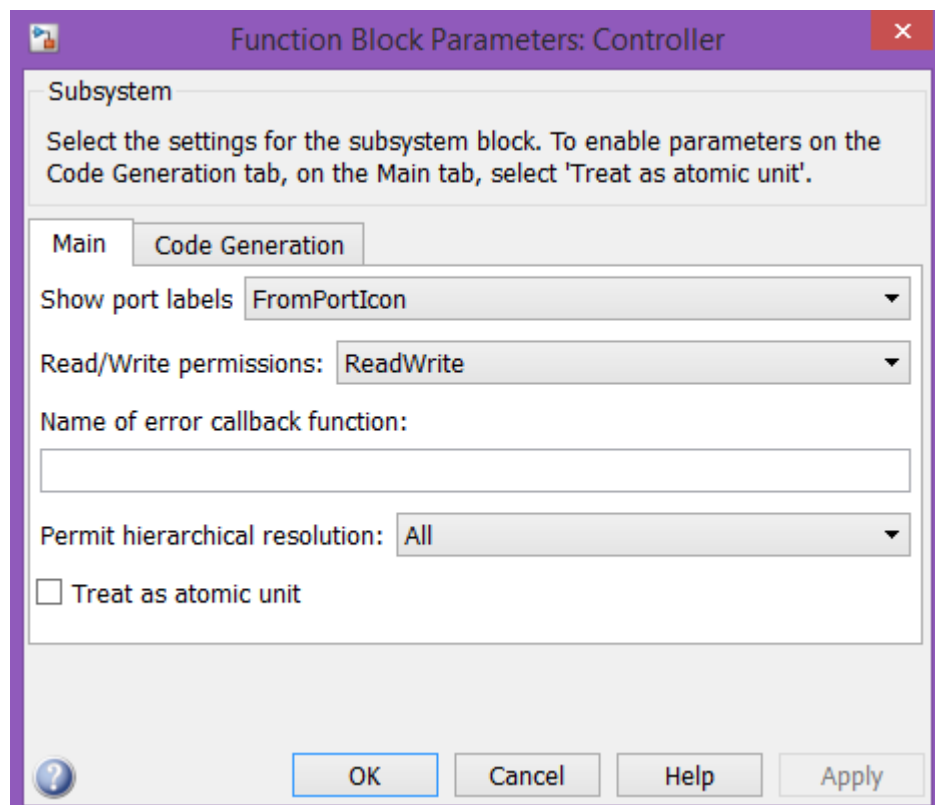


Abbildung 7-2 Einstellungen für das Subsystem

Zusätzlich ist ein Zero-Order-Hold Baustein eingesetzt worden, da für SIL und PIL eine Diskretisierung der Regelung erforderlich ist. Die Abtastzeit der Regelung beträgt 1ms. Für SIL sind zusätzlich weitere Einstellungen am Modell erforderlich. Dazu sind die Configuration Parameters zu öffnen und dort folgende Einstellungen vorzunehmen.

Unter Code Generation ist bei Target Hardware „None“ anzugeben. Als Toolchain ist „Automatically locate an installed toolchain“ auszuwählen. Unter Code Generation/Verification ist unter Create Block „SIL“ einzustellen. Bei allen weiteren Einstellungen wird auf den Foliensatz Software in the Loop und Prozessor in the Loop Simulationen“ von Herrn Zacher verwiesen.

Weiterhin sollten die 5 unterschiedlichen Reglerblöcke bei Sample Time auf -1 gestellt werden. Rechter Mausklick auf das Subsystem für den jeweiligen Regler¹⁰ → Block Parameters (Subsystem) → Sample Time = -1 ¹¹

Die Code Generierung geschieht nun wie folgt: rechter Mausklick auf das Subsystem ⇒ C/C++ Code ⇒ Build this Subsystem ⇒ Build.

Das SIL Subsystem wird gebildet. Es kann nun kopiert und gegen das alte Subsystem eingesetzt werden. Es wird genau an die gleiche Stelle eingesetzt, nachdem das alte Subsystem vorher gelöscht wurde. Das gesamte neue Systemmodell wird unter dem Dateinamen „Modell_ztrans_ohne_Visu_SIL.slx“ gespeichert. Das neue Subsystem beinhaltet den SIL Controller Block, erkennbar an dem SIL Schriftzug auf dem Controller Block. Durch Klick auf den Pfeil in der unteren linken Ecke, kann man in das Innenleben hineinschauen.

Es soll nun vorerst die MIL (Modell in the Loop) Simulation mit der SIL verglichen werden. Weiterhin soll die Abweichung zwischen beiden Simulationsarten, also der Fehler, festgehalten werden. Dazu wird das folgende Matlab Programm eingesetzt und gestartet.

```
close_system('Modell_ztrans_ohne_Visu',0) %für t=0 initialisiert
open_system('Modell_ztrans_ohne_Visu') %Simulink-Modell geöffnet
out=sim('Modell_ztrans_ohne_Visu', 10) %für t=10 ausgeführt
%out ist struct mit 'x' und 'tout'
yout_normal=find(out, 'x');
t_normal=find(out, 'tout');
clear out %struct-variable out löschen

close_system('Modell_ztrans_ohne_Visu_SIL',0) %für t=0 initialisiert
open_system('Modell_ztrans_ohne_Visu_SIL') %Simulink-Modell geöffnet
out = sim('Modell_ztrans_ohne_Visu_SIL',10);
yout_SIL=find(out, 'x')
t_SIL = find(out, 'tout')

%Vergleich Normal-Mode und SIL
machine_precision = eps(single(yout_normal));
tolerance = 4 * machine_precision;
fig1 = figure;
subplot(3,1,1), plot(t_normal, yout_normal), title('Normale Simula-
tion')
```

¹⁰ Also das Subsystem und nicht der konkrete Regler

¹¹ -1 for inherited

```
subplot(3,1,2), plot(t_normal, [abs(yout_normal-yout_SIL) tolerance]), title('Fehler')
subplot(3,1,3), plot(t_SIL, yout_SIL), title('SIL Simulation')
save_system('Modell_ztrans_ohne_Visu_SIL');
```

Die Kennlinien der Regler für einen Sollwert von 350° sind im Anhang dargestellt.

Als nächstes soll die HMI Visualisierung auf die SIL Simulation angewandt werden. Dazu ist wieder das Systemmodell aus Kapitel 5.5 genommen worden. Die Antennensimulation der Visualisierung wird gestartet. Dazu muss die Run.m Datei gestartet werden. Die Visualisierung wird auf das Modell mit dem Namen Modell.slx angewandt. Somit wird der Matlab Workspace mit entsprechenden Daten befüllt. Der Controller Subsystem Block wird nun nach dem oben beschriebenen Verfahren in einen SIL Controller Subsystem Block umgewandelt. Das System wird gespeichert und die Run.m Datei wird erneut gestartet. Im Anhang sind die Bilder der HMI Visualisierung für Sollwertsprünge von 0° auf 350° für die unterschiedlichen Regler dargestellt.

7.2 PIL (Software in the Loop): C-Code der Regler mit Embedded Coder generieren und testen

Zur PIL Implementierung wurde die ST-Link Kommunikation genutzt. Während der Realisierung stellte sich heraus, dass diese Kommunikation sehr langsam war. Daher ist die Abtastzeit auf 0,01 sec. herauf gesetzt worden. Dadurch war sowohl eine schnellere Implementierung des Systems auf das Discovery Board als auch ein schnelleres abarbeiten der Algorithmen auf dem Board möglich.

Die Kommunikationsgeschwindigkeit kann durch die Verwendung der seriellen Schnittstelle verbessert werden.

Im Unterschied zu SIL sind bei PIL folgende Einstellungen vorzunehmen:

Configuration Parameters ⇒ Coder Target ⇒ Target Hardware Resources ⇒ PIL ⇒ PIL communication interface: ST-Link

Configuration Parameters ⇒ Code Generation ⇒ Verification ⇒ Create Block: PIL

Code Generation ⇒ Toolchain: GNU Tools for ARM Embedded Processors

Code Generation ⇒ Target Hardware: STM32 F4-Discovery

Code Generation ⇒ Verification ⇒ Enable portable word sizes deaktivieren

Optimization ⇒ Use division for fixed-point net slope computation: Use division for reciprocals of integers only

Rechter Mausklick auf das Subsystem aus dem PIL Code erzeugt werden soll ⇒ C/C++ Code ⇒ Deploy this Subsystem to Hardware ⇒ Build ⇒ alten Block löschen ⇒ neuen generierten PIL Block einsetzen

Die Kennlinien für PIL werden nach dem gleichen Prinzip aufgenommen, wie die Kennlinien für SIL. Die Kennlinien für PIL werden ebenfalls im Anhang dargestellt. Beim Matlab Programm ist lediglich „Modell_ztrans_ohne_Visu¹²“ durch „Modell_ztrans_ohne_Visu_P“ und „SIL“ durch „PIL“ ersetzt worden.

Zuvor sind noch Kennlinien für das Reglermodell aus Kapitel 6.1.5.2 für eine Abtastzeit von 0,01 sec. und einen Sollwertsprung von 0° auf 10° aufgenommen worden.

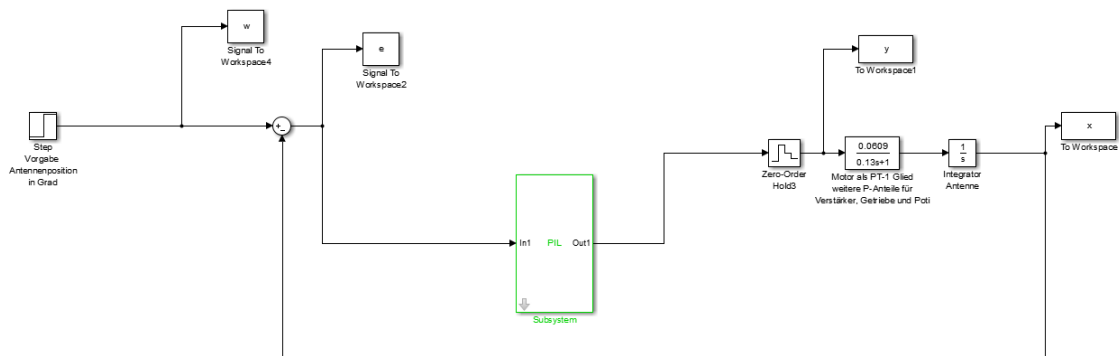


Abbildung 7-3 Reglermodell aus Kapitel 6.1.5.2 für PIL

Dabei wurden der PD Regler und die Sättigung zu einem Subsystem zusammengefasst.

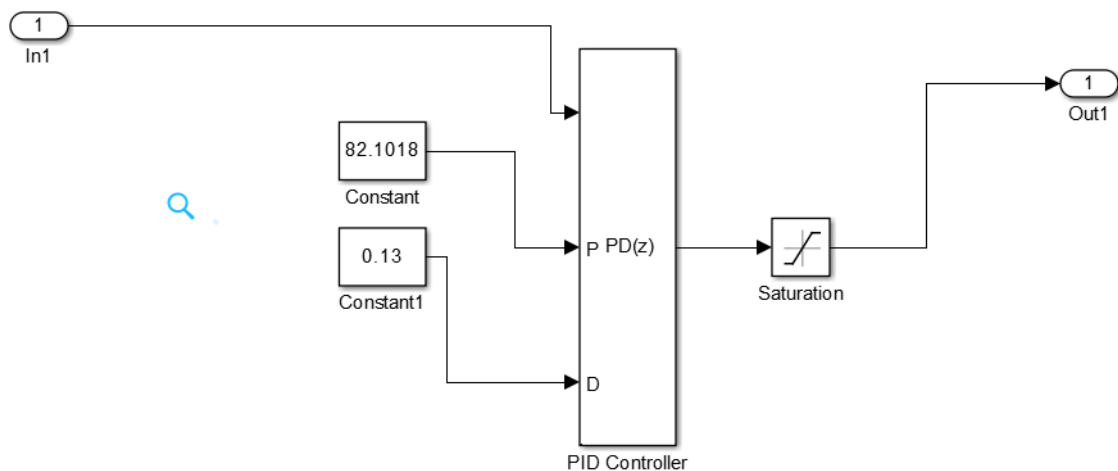


Abbildung 7-4 Inhalt Subsystem bei Reglermodell aus Kapitel 6.1.5.2 für PIL

Istwert und Reglerausgang sind im Anhang dargestellt. Man erkennt, dass sich bei dem Systemmodell aus Kapitel 5.5 eine bleibende Regelabweichung bei einer Abtastzeit von 10ms bei dem Kompensationsregler ergibt. Weiterhin kommt der Istwert nicht zur Ruhe und ist Schwankungen ausgesetzt. Die Regelung kommt nicht zur Ruhe, bzw. erreicht keinen stationären Wert und somit ist der Regler dauerhaft am Regeln. Da der Regler-

¹² Identisch mit Dateinamen des Modells

ausgang eines PD Reglers wie folgt definiert ist $Y_R(t) = K_{PR} [e(t) + Tv \frac{de(t)}{dt}]$ pendelt der Reglerausgang zwischen den beiden Sättigungswerte von -10 und +10. Als Fazit lässt sich sagen, dass das Systemmodell aus Kapitel 5.5 für den Kompensationsregler mit einer Abtastzeit von 10ms nicht geeignet ist.

Auf einer Untersuchung der Regler Smith Prediktor, PFC und SPFC bei einer Abtastzeit von 10ms ist verzichtet worden.

7.3 Analyse des generierten Codes

Die Datei ert_main.c ist die generierte Hauptdatei. Sie beinhaltet die Funktion One_Step (). Diese Funktion beinhaltet den Regelalgorithmus und wird in regelmäßigen Abständen aufgerufen. Der Aufruf könnte innerhalb der Funktionsblocks int_T main(int_T argc, const char *argv[]) erfolgen.

Weiterhin sind in dem Hauptprogramm weitere Dateien eingebunden:

```
#include <stddef.h>
#include <stdio.h>
#include "Controller.h"
#include "rtwtypes.h"
```

Die Datei Controller¹³.h beinhaltet Header Dateien und Definitionen von Datenstrukturen für den Regelalgorithmus. Der Algorithmus aus dem Simulink Modell befindet sich in der Datei Controller.c. In der Datei Controller.c befinden sich die Funktionen Controller_initialize () für die Initialisierung des Algorithmus und Controller_step () zum Ausführen des Algorithmus. Das Modell wird als sequentieller Algorithmus abgebildet. Die Namen der Variablen werden von Blocknamen abgeleitet, deren Berechnungsergebnis sie erhalten. Die Datei rtwtypes.h enthält Definitionen von Standarddatentypen. Der erzeugte Quellcode enthält Standarddatentypen, die über Typendefinition neue Namen erhalten. Ebenso werden Datenstrukturen aus diesen Typen verwendet.

¹³ Das Präfix der Datei, hier Controller, folgt aus dem Namen des Subsystems, aus dem sie generiert wurden.

8 HMI und Mikrocontroller Kommunikation

Die Visualisierung (HMI) wird mit der Matlab GUIDE realisiert. Diese bindet sowohl die Simulationsumgebung als auch die Modellsteuerung mit ein. Die Kommunikation des Modells erfolgt über eine serielle Schnittstelle.

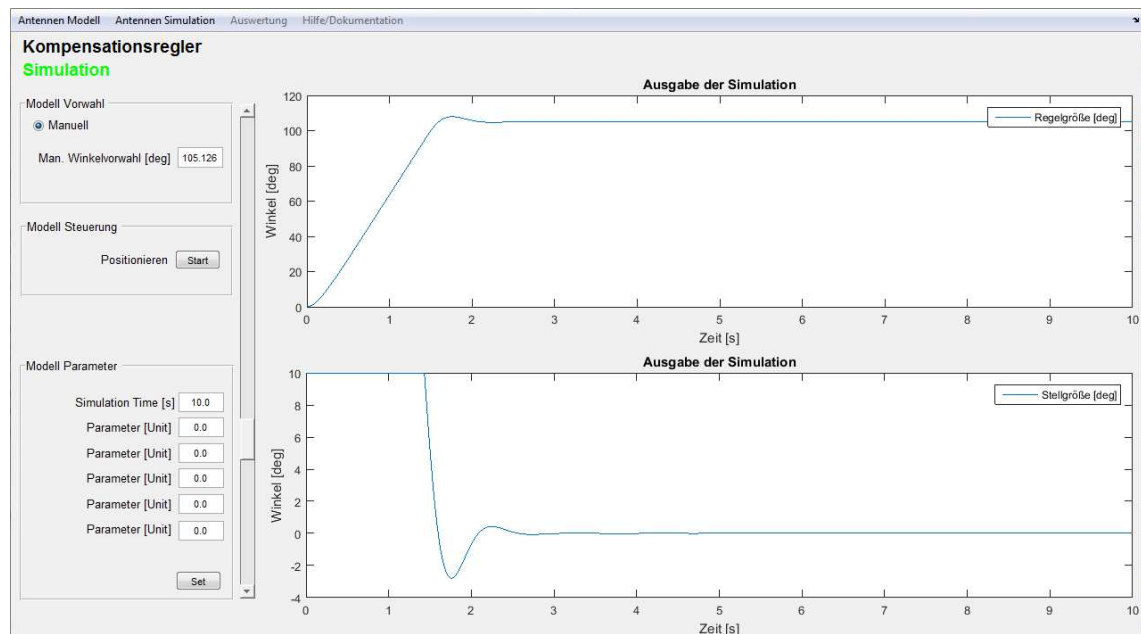


Abbildung 8-1 Visualisierung des Modells

8.1 Systematischer Aufbau der Kommunikation

Die Kommunikation zwischen PC und Controller wird über eine serielle Schnittstelle mittels USART¹⁴ und einem Treiberbaustein realisiert. Konkret ist hierzu der PC mittels USB an einen FTDI (FT232), der sich auf dem Piggyback¹⁵ befindet, angebunden. Der FTDI wiederum ist mit den GPIO¹⁶ - PC10 und PC11 verbunden. Der Treiberbaustein setzt das Spannungsniveau auf die erforderlichen 3,3V herab und organisiert die USB-UART Kommunikation.

Die Inhalte der Kommunikation sind fest definiert. Für den beschriebenen Funktionsumfang sind vier (4) Telegramme definiert. Drei (3) für Kommandos an der Controller und eines (1) für den Empfang. Die genaue Struktur jedes einzelnen Telegramms ist im Anhang wiederzufinden.

¹⁴ Universal Synchronous/Asynchronous Receiver Transmitter

¹⁵ Aufnahme Board für das STM32F4 Discovery

¹⁶ General Purpose Input/Output

Definition der Telegramme

Send_ConCon:

Selektion des gewählten Reglers (1-5).

Send_ConSet:

Übertragen der Parameter. Hier stehen insgesamt 5 freie Felder zu Verfügung, die verwendet werden können.

Send_ConCom:

Übertragen des Sollwerts.

Rec_All:

Empfang aller Systemrelevanten Daten. Systemstatus, Winkel, Strommessung, PWM Spannungspegel, Regler Setpoint, Regler Fehler.

8.1.1 Sequenzdiagramm der Kommunikation

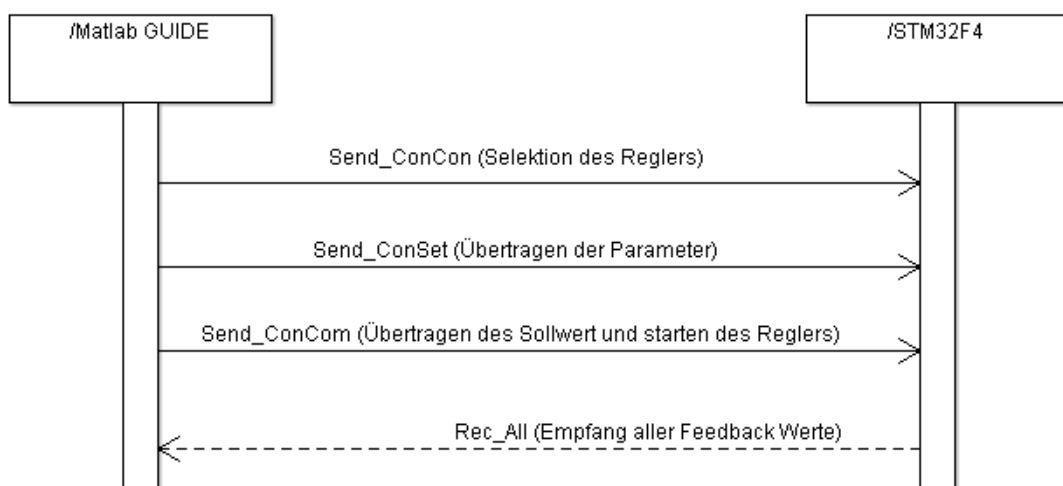


Abbildung 8-2 Sequenzdiagramm Kommunikation

Wie in Abbildung 8-2 Sequenzdiagramm Kommunikation ersichtlich, senden die beiden Telegramme Send_ConCon und Send_ConSet keinerlei Feedback an den Controller. Lediglich das Telegramm Send_ConCom führt dazu, dass die Positionierung gestartet wird und das Telegramm Rec_All wird bis zum Erreichen der Endposition zyklisch im 20ms Takt gesendet.

8.1.2 Telegrammaufbau

Zur Identifikation der verschiedenen Telegramme dient ein Telegramm-Header. Dieser entscheidet, sowohl empfangs- als auch sendeseitig, wie das jeweilige Telegramm zu interpretieren ist. Das Telegramm besteht aus einem String Array, welches wiederum mit Trennzeichen versehen ist. In dem hier vorliegenden Fall wurde das Semikolon - ; - als Trennzeichen verwendet, siehe nachfolgende Tabelle.

Tabelle 8-1 Telegramm Beispiel

Header	Daten	Daten	Daten
Send_ConCom;	1;	0;	0;

8.1.3 Telegrammverarbeitung

Die Interpretation der Telegramme wird je nach System (Matlab oder STM32F4) unterschiedliche gehandhabt.

Verarbeitung der Telegramme in Matlab

Zu Beginn wird in Matlab ein serieller Port geöffnet. Dies erfolgt unter Berücksichtigung der entsprechenden Schnittstellenparameter: Baudrate (115200), DatenBits (8), Modus (Asynchron) sowie dem Terminator (LF/CR).

Das erste Telegramm wird versendet, sobald die Anwahl des Kontrollers getroffen wurde. Im Folgenden wird die Position der Antenne vorgegeben und mit „Start“ auf der Visualisierung bestätigt – es folgt das Telegramm „Send_ConCom“ Position und Start-Kommando. Anschließend begibt sich die Visualisierung in den Empfangsmodus und zeichnet alle einkommenden Telegramme auf. Nach Abschluss der Positionierung wird ein Terminator gesendet, der den Empfangsmodus beendet. Abschließend werden die Daten in einer Excel-Datei gesichert.

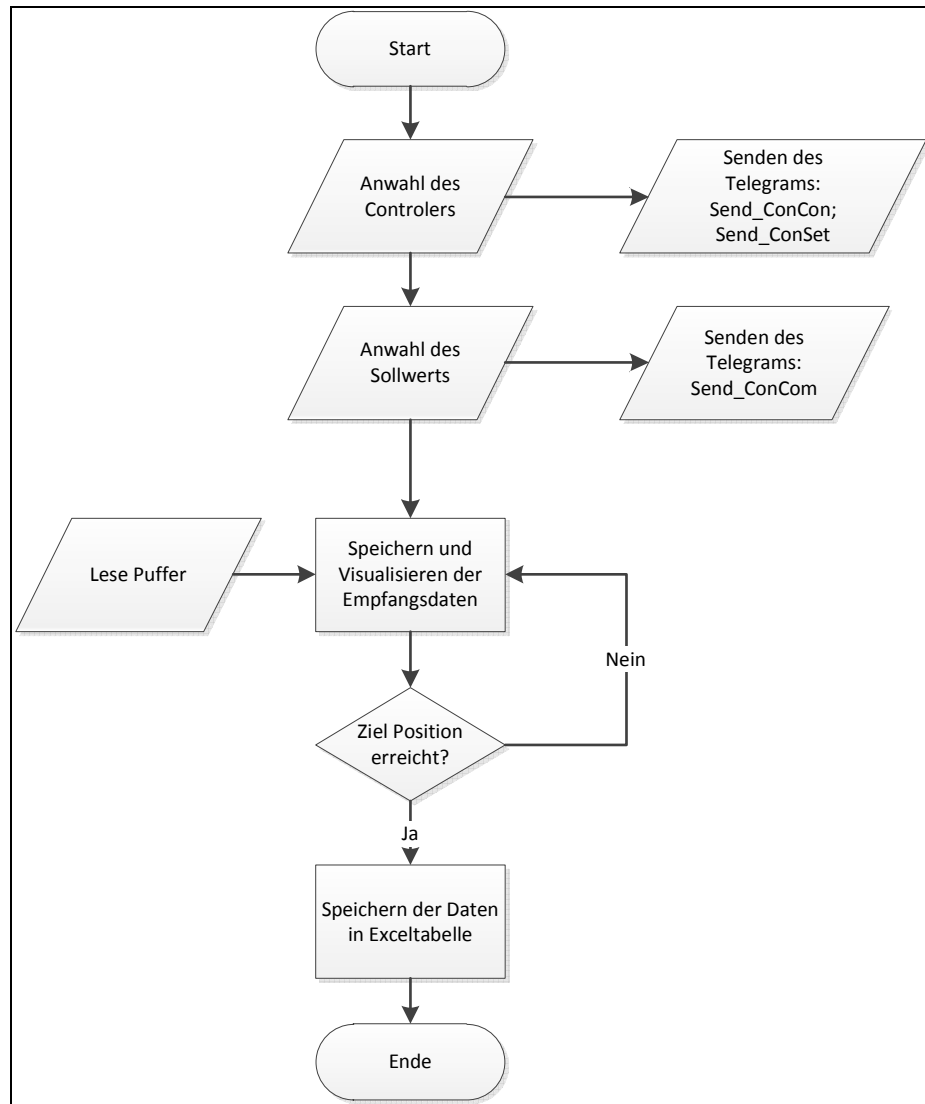


Abbildung 8-3 Kommunikation Matlab

Verarbeiten der Telegramme im Controller (STM32F4)

Für die Verarbeitung der Telegramme auf dem Controller wird eine freie Bibliothek von mikrocontroller-4u.de in der Version 1.4 verwendet. Diese ist zuständig für das Interrupt Handling und das Vorverarbeiten der Daten des USARTs.

Für das Senden und Empfangen wurden unterschiedliche Routinen implementiert. Diese befinden sich in der Uart.c und Uart.h.

Für den Empfang wird zyklisch die Uart.c aufgerufen. Sobald ein Telegramm detektiert wird, wird dieses abgearbeitet. Für das Identifizieren des Telegrams wird zunächst der Header überprüft. Wurde der passende Header erkannt, wird die für ihn definierte Routine aufgerufen. Das könnte beispielsweise das Kopieren des Sollwerts zu dem entsprechenden Regler sein. Wurde der entsprechende Header nicht erkannt, wird eine Fehlermeldung zurück an den Sender geschickt.

Für das Senden der Daten steht ein Ausgangspuffer zu Verfügung. Dieser wird zyklisch alle 50ms mit neuen Daten befüllt und alle 50ms an die Visualisierung gesendet. In dem vorliegenden Fall gibt es zwei Sende-Telegramme. Das Erste meldet eine Störung beim Empfang der Daten, das Zweite sendet alle Informationen zum aktuell laufenden Regel Prozess.

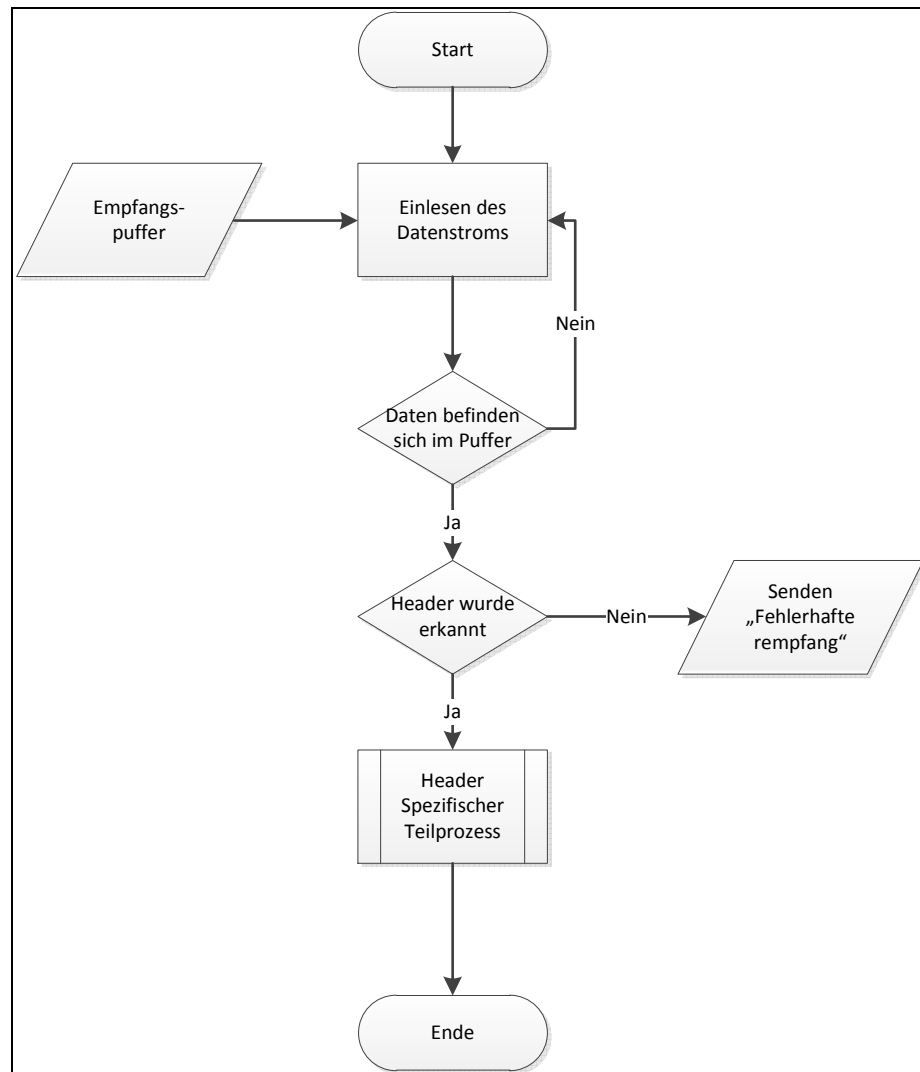


Abbildung 8-4 Kommunikation STM32F4

8.2 Matlab GUIDE

Wie bereits einleitend beschrieben wurde die Visualisierung mit der Matlab GUIDE umgesetzt. Die Visualisierung beinhaltet folgende Funktionen:

- Auswahl der Umgebung Antennenmodell / Simulation
- Winkelvorgwahl
- Starten der Positionierung / Simulation
- Übergabe diverser Parameter (sofern verfügbar)

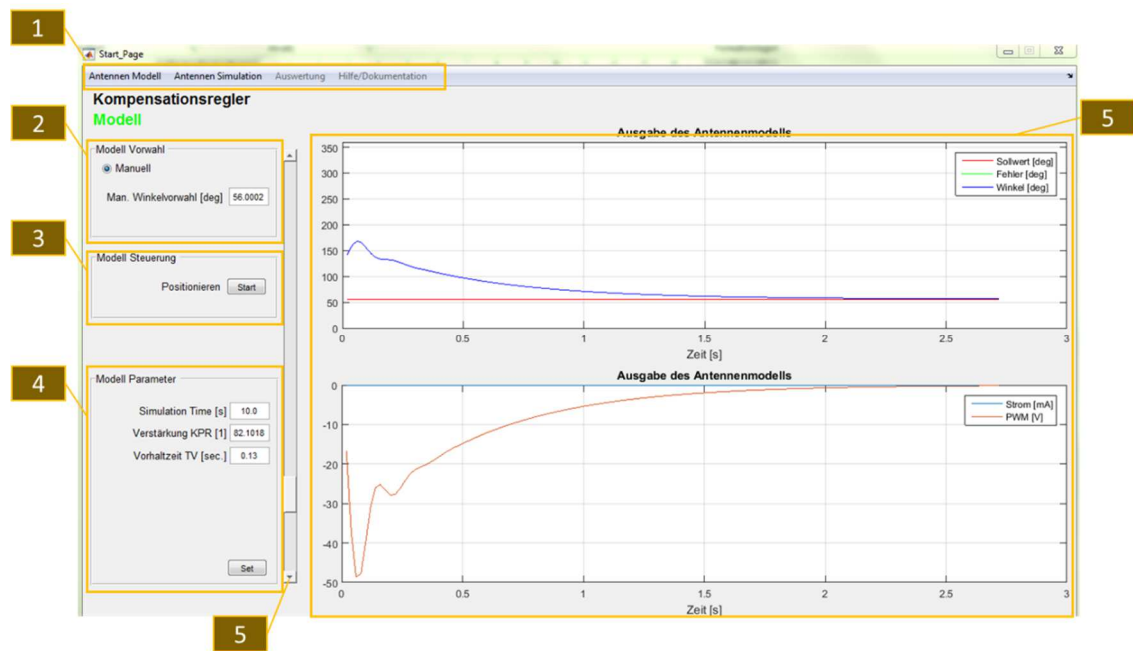


Abbildung 8-5 GUI / Visualisierung

Die Auswahl der entsprechenden Umgebung Simulation oder Modell erfolgt über das Menü Band, siehe [1]. Sobald im Untermenü Antennen Modell der entsprechende Regler gewählt wurde, wird die Visualisierung mit den entsprechenden Parametern dieses Reglers geladen.

Im Folgenden wird die Position durch direkt Eingabe „Man. Winkelvorgabe [deg]“ eingegeben, siehe [2]. Alternativ kann mit dem Slider, der sich zwischen den Datenflächen und den Plots befindet, eingestellt werden, siehe [5].

Die Positionierung wird mit Start bestätigt und je nach Umgebung wird die Positionierung begonnen, siehe [3].

Sollen versuchsweise alternative Parameter herangezogen werden, werden die entsprechend möglichen Parameter geändert und mit „Set“ bestätigt, siehe [4].

Nach Abschluss der Modell Positionierung wird die gesamte Kommunikation in einer Excel Datei gespeichert.

	A	B	C	D	E	F	G
1	Zeit	Sollwert_FB	Winkel_FB	Fehler_FB	PWM_FB	Strommessung_FB	
2	0,02	56,000198	141,740054	-85,739856	-16,78179	0	
3	0,04	56,000198	159,656341	-103,656143	-37,335648	0	
4	0,06	56,000198	168,373857	-112,373659	-48,61982	0	
5	0,08	56,000198	164,943201	-108,943003	-47,703298	0	
6	0,1	56,000198	154,324814	-98,324616	-39,531085	0	
7	0,12	56,000198	143,334642	-87,334444	-30,894767	0	
8	0,14	56,000198	136,165239	-80,16504	-25,972651	0	
9	0,16	56,000198	133,23306	-77,232862	-25,182438	0	
10	0,18	56,000198	132,55857	-76,558371	-26,555964	0	
11	0,2	56,000198	131,85418	-75,853981	-27,818509	0	
12	0,22	56,000198	129,889605	-73,889407	-27,748313	0	
13	0,24	56,000198	126,704127	-70,703928	-26,387765	0	

Abbildung 8-6 Beispiel der gesicherten Excel Datei

10 Zusammenfassung

In dieser Arbeit wurde betrachtet wie die modellbasierte Softwareentwicklung in der Regelungstechnik genutzt werden kann. Die Entwicklung begann in der Simulation und endete am realen Prototyp der Anlage. Als HMI Schnittstelle wurde GUIDE MATLAB eingesetzt.

Während der Arbeit wurden von den einzelnen Teammitgliedern teilweise unterschiedlich Ansätze im Entwicklungsstadium betrachtet. So wurden bei der Systemanalyse die Ansätze „*Systemanalyse ohne Vorkenntnisse*“ und *Systemanalyse a-priori gewählt*“. Es stellte sich heraus, dass beide Ansätze nicht die exakt gleichen Ergebnisse lieferten.

Bei der Reglerentwicklung wurden ebenfalls 2 Ansätze gewählt. Während im Kapitel 5.5 das Single Turn Verhalten des Potentiometers direkt nachgebildet wurde, wurde dieser Ansatz im Kapitel 6.1 bei der Entwicklung des Kompensationsreglers nicht gewählt.

Abschließend wird erwähnt, welche weiteren Möglichkeiten mit diesem Antennenmodell erarbeitet werden könnten.

Die Regelung wurde ohne Störgrößenbetrachtung durchgeführt. Um eine optimale Send- und Empfangsqualität zu erreichen muss eine Antenne in horizontaler und vertikaler Richtung optimal eingestellt sein. Diese optimale Einstellung kann durch Störgrößen, die an der Regelstrecke angreifen beeinträchtigt werden. Eine mögliche Ursache könnten z.B. starke Windböen oder Windgeschwindigkeiten sein. Man könnte also als Nächstes untersuchen wie die einzelnen Regler auf Störgrößen reagieren.

Eine weitere Betrachtungsmöglichkeit ist die Verwendung des SIMULINK HDL Coders mit dessen Hilfe VHDL Code generiert werden kann. Zum Einsatz könnte hierbei ein FPGA¹⁷ Board kommen.

Ebenfalls möglich wäre der Einsatz einer SPS. Hier ließe sich mit Hilfe des Simulink PLC Coders strukturierter Text nach der Norm IEC 61131 generieren. Der Quellcode könnte anschließend auf einer SPS implementiert werden.

¹⁷ Field Programmable Gate Array