



Technische Universität Graz



Institut für Regelungs- und
Automatisierungstechnik

Automatisierte Einstellung von PID-Reglern für Prozessanwendungen

Diplomarbeit

vorgelegt von

Martin Draxler

Graz, Jänner 2010

Im Auftrag der Firma



TG GmbH

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am

.....
(Unterschrift)

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die durch ihre Mithilfe zu einem erfolgreichen Abschluss der Diplomarbeit beigetragen haben.

Insbesondere möchte ich mich bei meinem Betreuer, Herrn Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Anton HOFER vom Institut für Regelungs- und Automatisierungstechnik der TU Graz für seine Hilfestellung während des gesamten Zeitraumes meiner Arbeit bedanken.

Außerdem bedanke ich mich bei meinem Firmenbetreuer Herrn Dipl.-Ing. Matthias LUKIC von der Firma TG Mess-, Steuer- und Regeltechnik GmbH, welcher mich während der Durchführung der Diplomarbeit unterstützte und mir während meines Studiums die Möglichkeit gab, praktische Erfahrungen im Bereich Prozessleittechnik von Großanlagen zu sammeln.

Weiters gilt mein Dank Herrn Georg ARNOLD, welcher mir beim Testen meiner Software und durch seine Anmerkungen zur Programmstruktur sehr behilflich war und deshalb auch einen wesentlichen Teil zum Erfolg der Diplomarbeit beigetragen hat.

Insbesondere möchte ich mich auch bei meiner Familie, speziell bei meinen Eltern und meiner Freundin bedanken, die mich in vollster Weise während meines Studiums unterstützten.

Kurzfassung

In heutigen modernen Industrieanlagen werden die meisten Regelungsaufgaben von komplexen Prozessen mit Hilfe von traditionellen Reglern gelöst. Dabei werden am häufigsten PI- und PID-Regler verwendet. Die Parameter für den jeweiligen Regler werden bis dato meist bei der Inbetriebnahme empirisch ermittelt und dies stellt ohne ein mathematisches Modell eine sehr komplexe und zeitaufwendige Aufgabe dar. Um die Reglereinstellung bzw. die Regleroptimierung zu vereinfachen, wurde in einer vorhergehenden Diplomarbeit der Firma TG GmbH bereits ein Programm zur Modellbildung von industriellen Prozessen (Parameterident) entwickelt. Diese Software bildet aus Prozessdaten der Industrieanlage ein mathematisches Modell in MISO-Form, welches in weiterer Folge als Grundlage für eine Berechnung der Reglerparameter und zur Simulation des Regelkreises dient.

Diese Arbeit beschäftigt sich nun mit der automatisierten Einstellung von PID-Reglern für Prozessanwendungen aus der Industrie, um die Effizienz bei der Optimierung zu steigern und um Stillstandszeiten während der Reglerkonfiguration in der Anlage zu verringern. Es wurde eine Entwicklungsumgebung in MATLAB/Simulink erstellt, mit welcher die Einstellparameter des jeweiligen PID-Reglers, basierend auf dem berechneten mathematischen Modell, ermittelt werden können. Mit Hilfe des entwickelten Programms PIDexpert ist es dem Anwender möglich, eine bestimmte Strecke des mathematischen Modells auszuwählen und mit mehreren Entwurfsmethoden verschiedene Regler mit einer gewünschten Struktur entwerfen zu lassen.

Eine Verifikation des Reglerentwurfes ist dadurch möglich, dass man in der Simulation weitere Eingangsgrößen des MISO-Modells als Störgrößen auf den Regelkreis einwirken lässt. Weiters können die Regler auch mit anderen Modellen des gleichen Prozesses, welche durch andere Identifikationsalgorithmen erhalten wurden, validiert und gegebenenfalls angepasst werden.

Abstract

In modern industrial plants most regulation tasks of complicated processes are solved by traditional controllers. Most commonly used are PI- and PID controller. The parameters for the respective controller are usually determined empirically at the time of the plant's startup, which presents a very complicated and time-consuming task in the absence of a mathematical model. To simplify the controller setting or the controller optimization, a program for modeling an industrial process (Parameterident) was developed in a preceding diploma thesis with the company TG GmbH. This software forms a mathematical model in MISO form from process data of the industrial plant, which then serves as the basis for the calculation of control parameters and for the simulation of the control circuit.

The present thesis deals with an automated setting of PID controllers for industrial process applications in order to increase the efficiency of the optimization and to reduce down time in the plant during the controller configuration. A software was developed in MATLAB/Simulink, which determines the parameters of the respective PID controller based on the calculated mathematical model. This program called PIDexpert makes it possible for the user to select a specified control system of the mathematical model and to design controllers with the desired structure by using several design methods.

The obtained controller can be verified by simulation with other input signals of the MISO-model, which causes a perturbation on the closed loop control circuit. Furthermore, the controllers can also be validated with other models of the same process, which were calculated by different identification algorithms, and be adapted when indicated.

Inhaltsverzeichnis

1	Einleitung.....	8
1.1	Motivation	8
1.2	Problem- und Aufgabenstellung.....	9
1.3	Struktur der Arbeit.....	10
2	Grundlage für die Softwareumgebung PIDexpert	11
2.1	Parameterident.....	11
2.1.1	Modellidentifikation mittels Parameterident	12
2.1.2	MISO-System	15
2.2	PID-Regler.....	16
2.2.1	Sprungantwort eines PID-Reglers	17
2.2.2	Prinzipielle Auslegung eines PID-Reglers.....	18
2.3	PID-Regelstrukturen.....	19
2.3.1	PI-D Struktur.....	20
2.3.2	I-PD Struktur.....	21
2.4	Anti-Wind-up	22
2.4.1	Stellgrößenbeschränkung.....	22
2.4.2	Wind-up Effekt.....	22
3	Entwurfsverfahren.....	25
3.1	Ziegler Nichols.....	26
3.1.1	Verfahren 1: „open loop method“	26
3.1.2	Verfahren 2: „closed loop method“	27
3.1.3	Beispiel zum Entwurf mit Ziegler Nichols.....	30
3.2	Chien Hrones Reswick.....	32
3.2.1	Beispiel zum Entwurf mit Chien Hrones Reswick.....	33
3.3	k150-Verfahren	34
3.3.1	Beispiel zum Entwurf mit k150	38
3.4	gKYP-Lemma	39
3.4.1	Beispiel zum Entwurf mit dem gKYP-Lemma	44
3.5	Minimierung von Integralkriterien.....	45
3.5.1	Beispiel zum Entwurf mit Minimierung von Integralkriterien.....	49
3.6	Improved Performance	50
3.6.1	Beispiel zum Entwurf mit Improved Performance	54
4	Aufbau der Software	55
4.1	Vorgaben der Firma TG GmbH.....	55

4.2	Struktur der Entwicklungsumgebung	57
4.2.1	Auswahl der Regelstrecke	57
4.2.2	Auswahl und Konfiguration des Entwurfsverfahrens.....	58
4.2.3	Simulation des erhaltenen Reglers.....	60
4.2.4	Simulation mit Störgrößen	61
4.3	Datenstruktur	62
4.3.1	Modelldaten	62
4.3.2	Reglerdaten.....	63
4.3.3	Kompatibilität vom Parameterident und PIDexpert.....	65
4.4	Aufbau in MATLAB/Simulink	67
4.4.1	PID-Regler.....	68
5	PIDexpert	70
5.1	Erste Schritte	71
5.1.1	Installation der Toolbox.....	71
5.1.2	Starten der Entwicklungsumgebung.....	71
5.2	Auswahl der Regelstrecke	72
5.3	Auswahl und Konfiguration des Entwurfsverfahrens	74
5.3.1	Einstellungen.....	75
5.4	Simulation des erhaltenen Reglers	78
5.4.1	Reglerparameter variieren.....	81
5.5	Simulation mit Störgrößen.....	83
5.6	Menüleiste.....	85
5.6.1	Export.....	86
6	Beispiel aus der Praxis.....	87
6.1	Beschreibung der Anlage	87
6.2	Reglerentwurf.....	89
6.2.1	Ergebnisse der Entwurfsverfahren.....	90
6.2.2	Auswahl des PID-Reglers.....	91
6.2.3	Störgrößensimulation.....	94
7	Zusammenfassung und Ausblick	100
8	Anhang.....	101
8.1	HTML-Report.....	101
8.2	Abbildungsverzeichnis	107
8.3	Tabellenverzeichnis	109
8.4	Literaturverzeichnis.....	110

1 Einleitung

Diese Diplomarbeit beschäftigt sich mit dem Thema der automatisierten Einstellung von PID-Reglern für Prozessanwendungen. Es wird eine Entwicklungsumgebung in MATLAB/Simulink erstellt, welche aus einem mathematischen Modell der Regelstrecke automatisiert einen PID-Regler entwirft und diesen durch Simulation mit Störgrößen überprüft.

1.1 Motivation

Heutzutage werden Industrieanlagen mit einem Prozessleitsystem überwacht und gesteuert. Damit ein wirtschaftlicher und umweltfreundlicher Betrieb möglich ist, bedarf es einer weitgehend automatisierten Anlage. Dazu benötigt man eine Vielzahl von Reglern, welche bei der Inbetriebnahme bzw. nach einem Umbau eingestellt und optimiert werden müssen. In vielen Anlagen wird diese Regelungsaufgabe auch heute noch mit der Hilfe von konventionellen PID-Reglern gelöst, da diese leicht zu handhaben und universal einsetzbar sind. In modernen Leitsystemen werden diese Regler nicht mehr hardwarebasierend sondern bereits softwaretechnisch realisiert.

In Industrieanlagen sind Regelungen von komplexen Prozessen meist als Kaskadenregelungen aufgebaut. Deshalb ist für die Inbetriebnahme der Regler in einer Anlage ein hohes Maß an Erfahrung und Zeit notwendig, um alle physikalischen Zusammenhänge der einzelnen Prozesse zu erfassen und eine akzeptable Regelung zu entwerfen. Klassische Verfahren für den Entwurf, wie zum Beispiel die Einstellverfahren nach Ziegler und Nichols, sind nur mittels eines Eingriffes in den laufenden Betrieb möglich, welcher meist nicht erwünscht ist. Außerdem fordert dieses Verfahren bei der „closed-loop“ Methode, dass der P-Teil des Reglers solange erhöht wird, bis eine ungedämpfte Dauerschwingung der Ausgangsgröße eintritt. Dieser Zustand ist in einer Industrieanlage nicht immer gefahrlos möglich. Auch bei anderen Verfahren ist meist ein Eingriff in den laufenden Betrieb notwendig oder sie können nur im Stillstand der Anlage durchgeführt werden. Aus wirtschaftlichen Gründen wird jedoch stets versucht, die Stillstandszeiten so kurz wie möglich zu halten oder ganz zu vermeiden.

All diese Probleme können durch Ermittlung eines mathematischen Modells der Regelstrecke für den Reglerentwurf entschärft werden. Mit dem ermittelten Modell kann eine rechnergestützte Simulation des Regelkreises durchgeführt werden und verschiedene Parameter des Reglers erprobt werden. Die erhaltenen Reglerparameter dienen danach als Hilfestellung bei der Inbetriebnahme bzw. bei der Optimierung der Anlage und können dort den realen Bedingungen angepasst werden.

Diese Idee hat die Firma TG GmbH in Zusammenarbeit mit der TU Graz aufgegriffen und entwickelte im Zuge einer Diplomarbeit mit dem Titel „Entwicklung einer Software-Umgebung für die Parameteridentifikation

industrieller Prozesse“ ein Programm namens „Parameterident“ zur Identifikation solcher mathematischer Modelle, welche nun als Grundlage für diese Arbeit dienen.

1.2 Problem- und Aufgabenstellung

Im Zuge dieser Diplomarbeit wurde laut Vorgaben der Firma TG GmbH eine Software in MATLAB mit dem Namen „PIDexpert“ entwickelt. Dieses Programm soll dem Anwender die Möglichkeit bieten, mit dem aus der Identifikationsumgebung „Parameterident“ erhaltenen Modell (MISO) eines Prozesses, schnell und einfach einen Regler mit der gewünschten Struktur (P, PI, PID, PD) zu entwickeln. Weiters soll es dem Benutzer möglich sein, das Verhalten des Regelkreises zu simulieren, wobei weitere Eingangsgrößen des MISO-Streckenmodells als Störgrößen verwendet werden können.

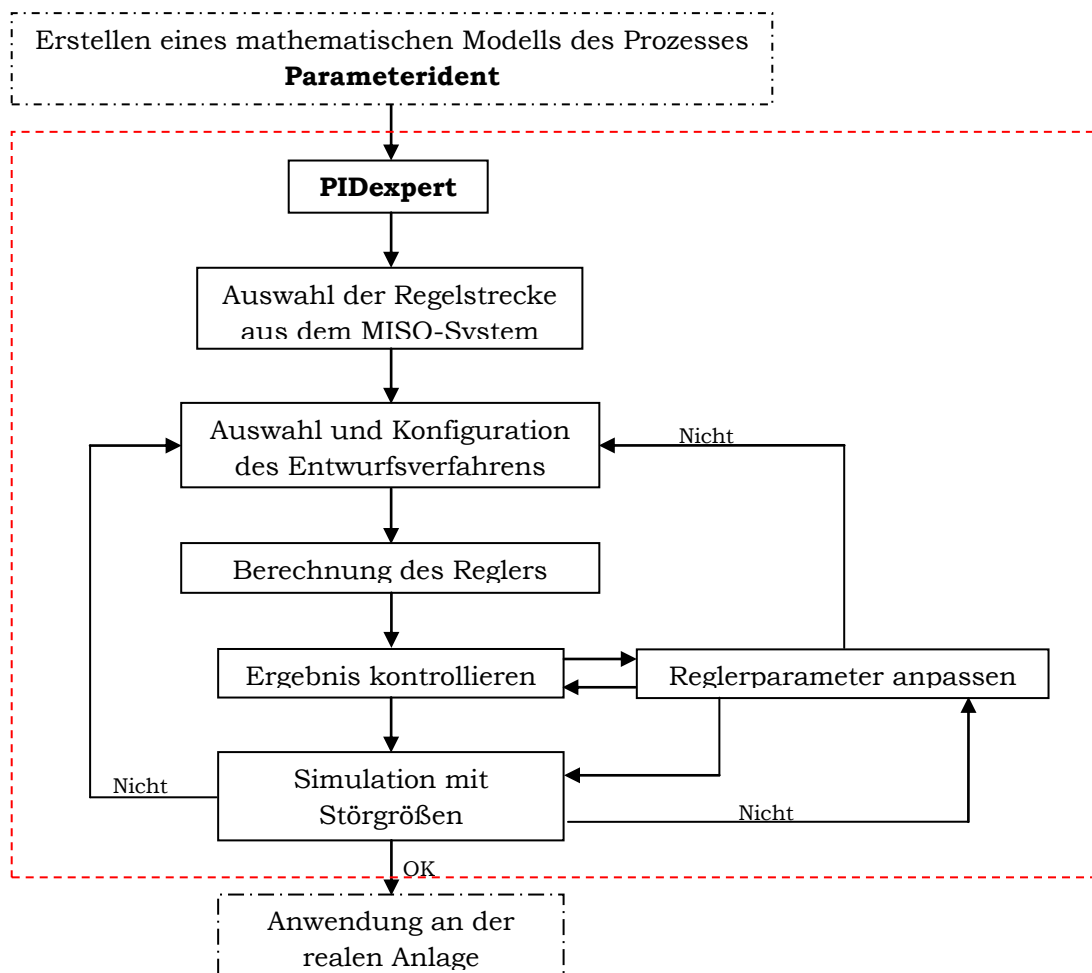


Abbildung 1.1: Ablauf der Entwurfsaufgabe mit PIDexpert

Die Voraussetzung für das Gelingen des zuvor gezeigten Ablaufs ist ein möglichst genaues mathematisches Modell des Prozesses, um Reglerparameter zu bestimmen, welche in der realen Anlage auch ein ähnliches Verhalten wie in der Simulation aufweisen. Dazu muss man bei der Modellbildung alle wichtigen

physikalischen Zusammenhänge, wie z.B.: Totzeiten, integrierendes Verhalten der Strecke usw., berücksichtigen.

1.3 Struktur der Arbeit

Beginnend mit dem 2. Kapitel werden die für das Programm PIDexpert notwendigen Grundlagen erklärt. In Kapitel 3 werden die theoretischen Hintergründe der vorhandenen Regler-Entwurfsmethoden erläutert und anhand eines Testbeispiels verifiziert.

Kapitel 4 befasst sich kurz mit den Vorgaben der Firma TG GmbH und dem Aufbau der Software. Es werden die Simulationsmodelle in MATLAB/Simulink und die Struktur des MATLAB-Programms näher erläutert.

In Kapitel 5 wird die Benutzeroberfläche von PIDexpert im Detail näher erklärt und in Kapitel 6 die Software anhand eines praktischen Beispiels getestet.

2 Grundlage für die Softwareumgebung PIDexpert

Wie bereits in der Einführung erwähnt, dient das mathematische Modell der Identifikationssoftware Parameterident als Grundlage für diese Arbeit. Dieses Kapitel bietet einen Überblick über den Ausgangspunkt dieser Diplomarbeit und behandelt kurz die notwendige Theorie für die Entwicklung der Softwareumgebung PIDexpert.

2.1 Parameterident

Diese Software wurde im Rahmen einer Diplomarbeit und mehrerer Projektarbeiten entwickelt und stetig verbessert. Sie wurde ausschließlich mittels MATLAB entwickelt und bietet die Möglichkeit, aus Prozessdaten Modelle in MISO (Multiple-Input Single-Output) Form mit verschiedenen Identifikations-Algorithmen zu berechnen.

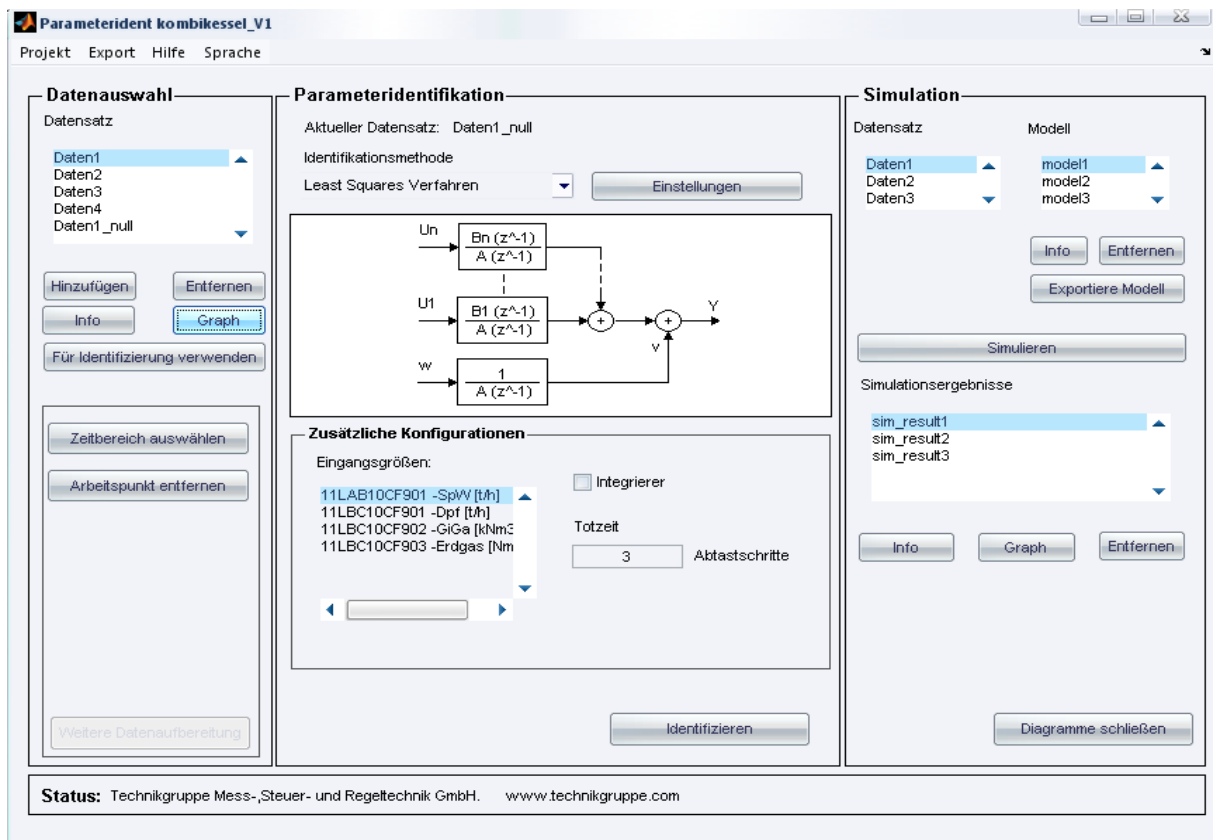


Abbildung 2.1: Benutzeroberfläche Parameterident aus [5]

Ziel dieser Identifikation ist die Erstellung eines mathematischen Modells, welches das dynamische Verhalten eines Prozesses möglichst exakt beschreibt.

2.1.1 Modellidentifikation mittels Parameterident

In der realen Welt sind Strecken meist nichtlinear. Für sehr viele können jedoch mit hinreichender Genauigkeit lineare Modelle erstellt werden, oder zumindest ein linearisiertes Modell um einen Arbeitspunkt angegeben werden.

Kann der Prozess mit hinreichender Genauigkeit durch ein linear zeitinvariantes Modell beschrieben werden, ist es möglich das Ein-/Ausgangsverhalten durch z-Übertragungsfunktionen bekannter Ordnung darzustellen. Der Identifikationsvorgang besteht darin, endlich viele Parameter der Übertragungsfunktion auf Grund von gemessenen Prozessdaten zu finden. Man bestimmt also bei einer vorgegebenen Modellstruktur, die zur endgültigen Festlegung des Modellverhaltens notwendigen Parameter. Diese Vorgangsweise nennt man Parameterschätzverfahren.

Als Eingangsdaten für die Identifikation zur Modellbildung dienen gemessene Prozessdaten, welche dem Programm per Excel-File zur Verfügung gestellt werden.

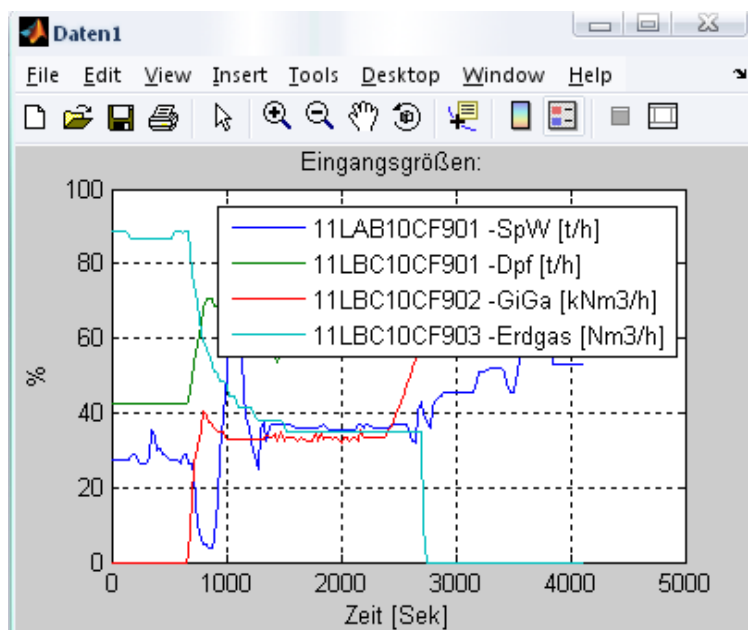


Abbildung 2.2: Eingangsgrößen für die Identifikation

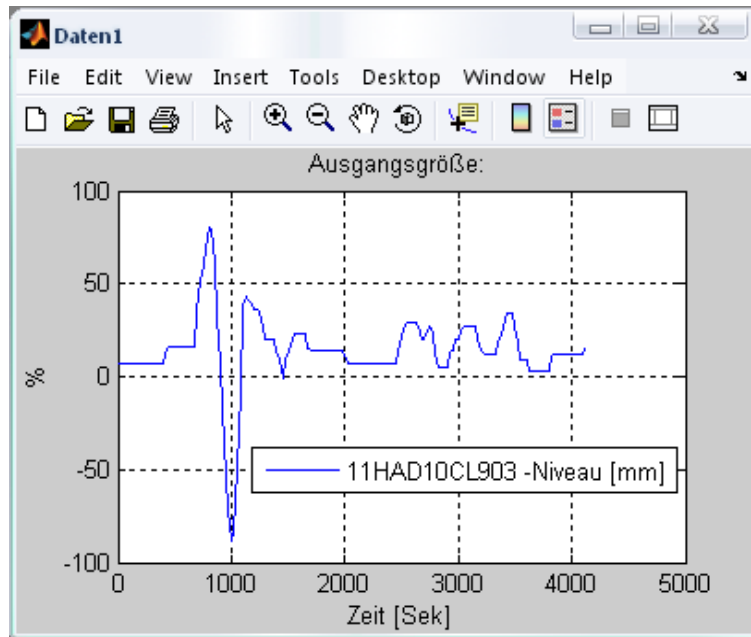


Abbildung 2.3: Ausgangsgröße für die Identifikation

Um ein besseres Ergebnis bei der Identifikation zu erreichen, werden die Daten vor der Identifikation aufbereitet. So zum Beispiel wird durch Berücksichtigung der Anfangswerte der vorhandenen Prozessdaten ein fiktiver Anfangszustand bei Null erzeugt. Somit ist es möglich, die Identifikation mit Verfahren, die keine Anfangswerte schätzen, durchzuführen.

Nach der Aufbereitung der Prozessdaten stehen dem Anwender des Programms zwei verschiedene Identifikationsalgorithmen, um das mathematische Modell zu ermitteln, zur Verfügung:

1. Minimierung der Fehlerquadrate
2. Rekonstruktion der Störgröße

Mittels dieser Identifikationsverfahren erhält man ein LZI-System¹ welches mit z-Übertragungsfunktionen beschrieben werden kann. Für jede Eingangsgröße, welche für den Entwurf herangezogen wurde, wird eine Übertragungsfunktion identifiziert. Diese besitzen ein gemeinsames Nennerpolynom und verschiedene Zählerpolynome. Die Ausgangsfunktion des identifizierten Modells wird durch Addition der Ausgangsgrößen der einzelnen z-Übertragungsfunktionen gebildet.

Nach Beendigung der Identifikation, kann das erhaltene Modell durch Simulationen mittels verschiedener Prozessdaten validiert werden, um eine Aussage über die Qualität des berechneten Modells zu treffen. Die Qualität des identifizierten LZI-Systems hängt nämlich sehr stark von der Einstellung der einzelnen Algorithmen und vom gewählten Arbeitspunkt bei der Messung der Prozessdaten ab. Falls die Validierung des Modells kein zufriedenstellendes Ergebnis liefert, muss der Identifikationsvorgang mit anderen Einstellung

¹ Lineares **Z**eitinvariantes System

wiederholt werden, da nur ein sehr gut identifiziertes Modell, welches annähernd der Realität entspricht, für den Reglerentwurf herangezogen werden sollte.

Ein prinzipieller Ablauf der Identifikationsaufgabe wird in Abbildung 2.4 dargestellt.

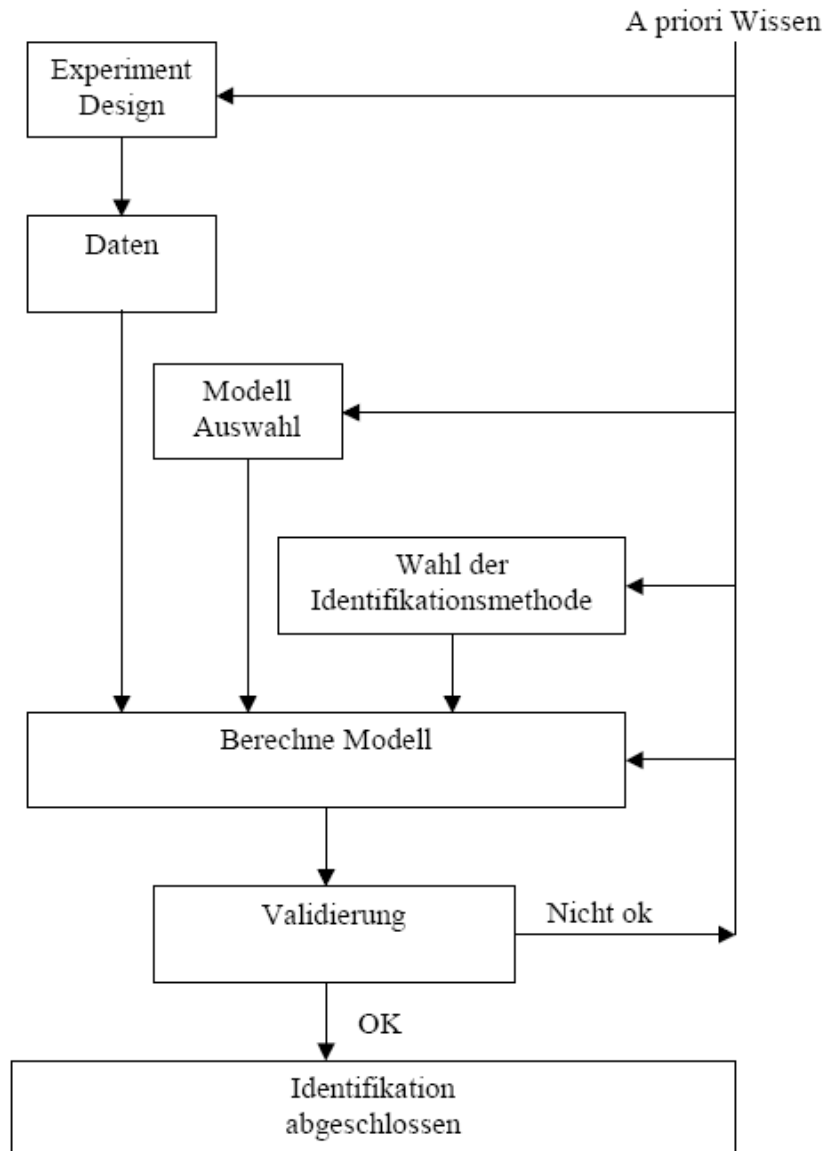


Abbildung 2.4: Ablauf einer Identifikationsaufgabe
entnommen aus [5]

Weitere Informationen zu dem Identifikationsvorgang und der Entwicklungssoftware Parameterident sind in [5] zu finden.

2.1.2 MISO-System

Wie bereits zuvor erwähnt, ist das mathematische Modell, welches man aus dem Programm Parameterident erhält, ein MISO-System. Die Anzahl der Eingänge bestimmt die Anzahl der Übertragungsfunktion, deren Ausgänge zu einem Ausgang zusammengeführt werden.

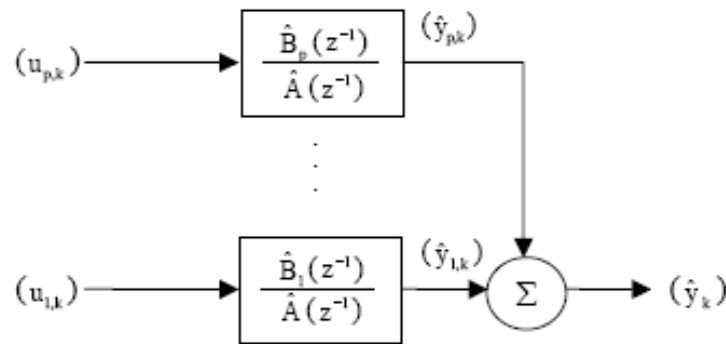


Abbildung 2.5: MISO-System aus [5]

Diese Systemstruktur dient als Grundlage für den weiteren Reglerentwurf. Es wird eine der vorhandenen Übertragungsfunktionen des MISO-Systems als Regelstrecke gewählt und ein Regler mit einer gewünschten Struktur² ermittelt. Die Ausgangsgrößen der restlichen Teilsysteme werden bei der Simulation als Störgröße der Strecke aufgeschaltet.

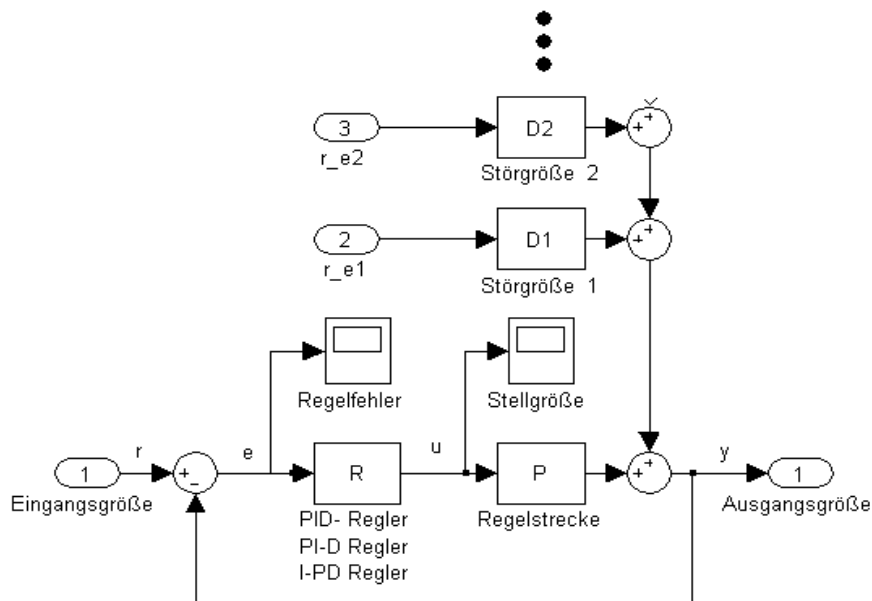


Abbildung 2.6: Prinzipieller Aufbau des Regelkreises

² Struktur: P, PI, PID, PD

2.2 PID-Regler

Die in der Praxis am häufigsten eingesetzte Regelstruktur ist nach wie vor der PID-Regler. Durch die spezifische Wahl der Parameter kann daraus ein P, PI, PD und PID Regler realisiert werden. Für den Aufbau des Reglers gibt es zwei alternative Darstellungsmöglichkeiten, die jedoch inhaltlich identisch sind:

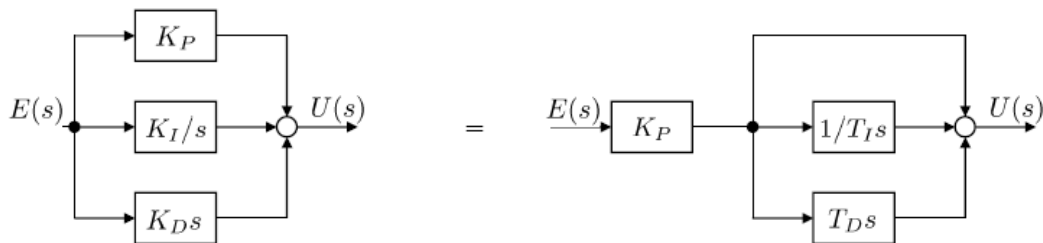


Abbildung 2.7: Darstellungsformen eines PID-Reglers aus [9]

Im Frequenzbereich:

$$R(s) = K_P + K_I \frac{1}{s} + K_D s$$

\nwarrow \uparrow \swarrow
 P- Anteil I- Anteil D- Anteil

$$R(s) = K_P \left(1 + \frac{1}{(T_I s)} + T_D s \right)$$

\nwarrow \uparrow \swarrow
 Proportionale Nachstellzeit Vorhaltezeit
 Verstärkung (Integrierzeit) (Differenzierzeit)

Im Zeitbereich:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{d}{dt} e(t) \qquad u(t) = K_P \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{d}{dt} e(t) \right]$$

P-Anteil: $u(t) = K_P e(t)$

- Beim P-Anteil wird die Regelabweichung³ $e(t)$ proportional verstärkt → Stellgröße ist proportional zur Regelabweichung.

I-Anteil: $u(t) = K_I \int_0^t e(\tau) d\tau$

- Die Stellgröße entspricht dem Integral der vorangegangenen Regelabweichung.
- Langsamer aber vollständiger Abbau der Regelabweichung bei konstanter Führungsgröße.

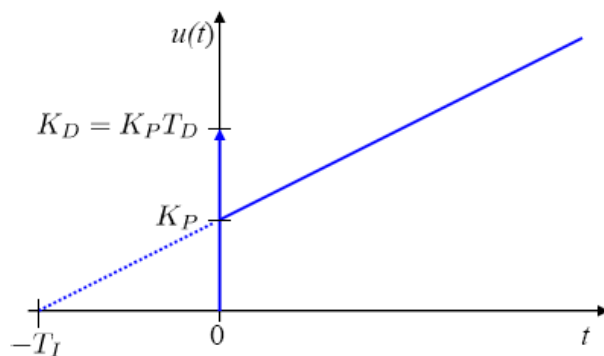
³ Regelfehler

D-Anteil: $u(t) = K_D \frac{d}{dt} e(t)$

- Stellgröße entspricht der Steigung der Regelabweichung
- Sehr schnelle Reaktion. Auch bei kleiner Regelabweichung mit falscher Tendenz aktiv.

Für die in dieser Arbeit entwickelte Software wurde die zweite Darstellungsform mit der Vorhalte- und Nachstellzeit eines PID-Reglers gewählt, da diese Form in den derzeit verwendeten Leitsystemen für Industrieanlagen der Firma TG GmbH zum Einsatz kommt.

2.2.1 Sprungantwort eines PID-Reglers



D-Anteil: Driac-Impuls
multipliziert mit $K_P T_D$.

P-Anteil: Sprungantwort ist
eine Sprungfunktion mit der
Höhe K_P .

I-Anteil: Linearer Anstieg mit
der Steigung K_P/T_I .

Abbildung 2.8: Sprungantwort eines PID-Reglers aus [9]

Das gezeigte Verhalten gilt nur für den idealen Differenzierer. Er benötigt nämlich einen sogenannten Realisierungsterm, da sonst die Übertragungsfunktion im Zähler eine höhere Ordnung als im Nenner aufweisen würde und dies wäre technisch nicht realisierbar. Deshalb wird der Differenzierer als DT_1 Glied aufgebaut, indem zum idealen Differenzierer eine kleine Verzögerung hinzugefügt wird.

$$DT_1 \text{ Glied: } K_P T_D \frac{s}{(1 + T_f s)} \quad \text{für } T_f \ll T_D$$

Eine gute Wahl für den Realisierungsterm des D-Teils liegt im Bereich $T_f = T_D/(8 \dots 20)$. In dieser Arbeit wird in weiterer Folge $T_f = T_D/(20)$ verwendet.

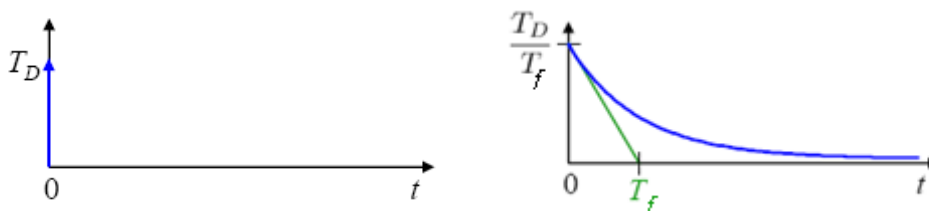


Abbildung 2.9: Idealer und realer Differenzierer aus [9]

2.2.2 Prinzipielle Auslegung eines PID-Reglers

P-Anteil:

Der P-Anteil bestimmt im vorhandenen Regelkreis die Kreisverstärkung und ist somit für die Stabilität von entscheidender Bedeutung. Es ist stets ein Kompromiss bei der Wahl von K_P zwischen einer kleinen bleibenden Regelabweichung und einer schnellen Dynamik einerseits und genügend Abstand zur Stabilitätsgrenze und guter Dämpfung andererseits zu finden.

I-Anteil:

Der Hauptgrund für das Hinzufügen eines Integral-Teils ist die Vermeidung einer bleibenden Regelabweichung. Falls die Strecke von vornherein einen I-Anteil besitzt, kann der I-Anteil im Regler meist weggelassen werden. Dies ist sinnvoll, da der I-Anteil durch seine Phasenabsenkung um 90° destabilisierend wirkt und den Phasenrand entsprechend verkleinert.

D-Anteil:

Die Verwendung eines D-Anteils wird meist vermieden, da er hochfrequente Anteile im rückgekoppelten Messsignal sehr stark verstärkt oder leicht zu einem Überschwingen der Ausgangsgröße führt. Er kommt hauptsächlich bei Regelungsaufgaben zum Einsatz, bei denen eine schnelle Dynamik gefragt ist oder die Strecke durch ein doppeltes I-Verhalten eine starke Phasenrückdrehung aufweist.

2.3 PID-Regelstrukturen

Die klassische Form des PID-Reglers führt in der Praxis zu einigen unerwünschten Nebeneffekten. Sprunghafte Änderungen der Führungsgröße, können aufgrund des P- und des D-Anteils zu betragsmäßig großen Stellgrößenwerten führen. Aus diesem Grund wird bei der praktischen Implementierung in der Industrie die Form des Reglers etwas modifiziert. In den folgenden Abbildungen wird die Problematik verdeutlicht.

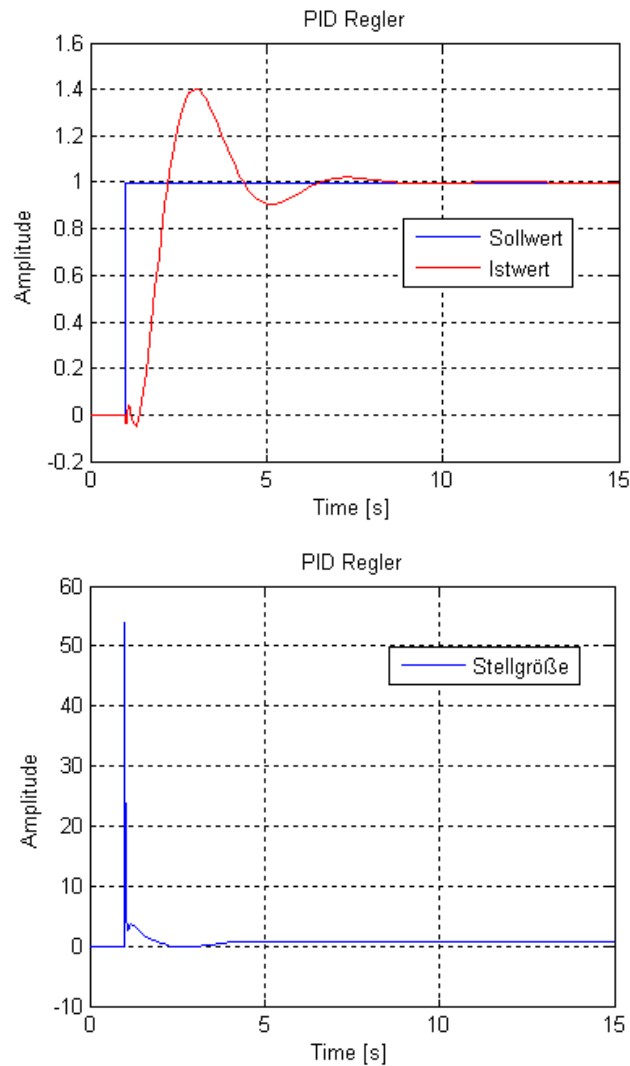


Abbildung 2.10: Problematik der traditionellen PID-Struktur

2.3.1 PI-D Struktur

Eine sprungförmige Änderung des Sollwerts⁴ des Regelkreises führt zu einer extremen Reaktion des D-Anteils, welche nicht gewollt ist. Daher wird der D-Anteil meist nur in der Rückführung eingebaut, um nur die Ableitung der Ausgangsgröße zu bilden, bei der in der Realität meist keine sprungförmige Änderung vorhanden ist.

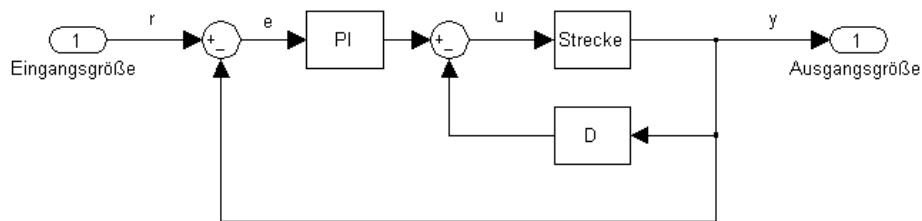


Abbildung 2.11: PI-D Struktur

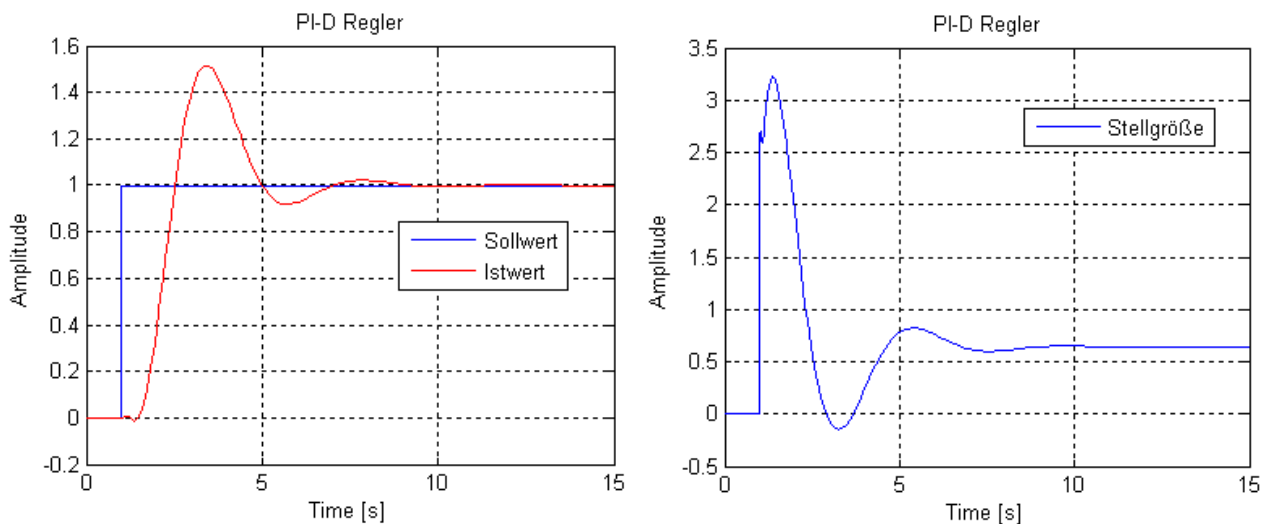


Abbildung 2.12: Verläufe der PI-D Struktur

Aus den Ergebnissen der Simulation mit der PI-D Struktur ist ersichtlich, dass sich der Stellgrößenverlauf des geschlossenen Regelkreises, im Vergleich zur Abbildung 2.10, deutlich verbessert hat. Durch die alternative Struktur des PID Reglers kommt es bei einer sprungförmigen Änderung der Führungsgröße zu betragsmäßig geringeren Stellgrößenwerten. Der Istwert des Regelkreises weist für diese Regelstrecke zwar ein etwas größeres Überschwingen auf, jedoch kann dieses Verhalten durch eine Anpassung der Reglerparameter wieder annähernd kompensiert werden.

⁴ Führungsgröße

2.3.2 I-PD Struktur

Bei dieser Implementierung des PID-Reglers wird nur mehr der Integral-Anteil vom Regelfehler durchlaufen. Auch dadurch führt eine sprunghafte Änderung beim Sollwert zu keinem Sprung in der Stellgröße.

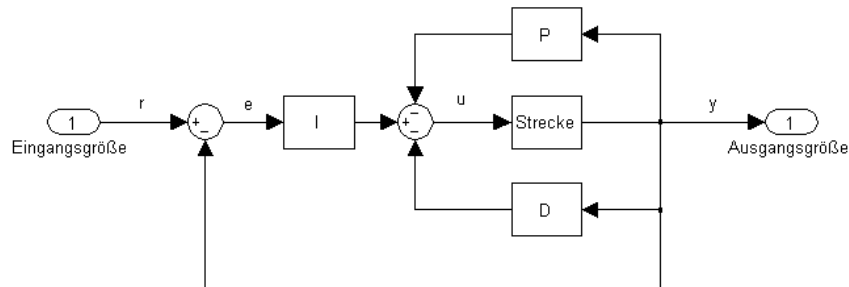


Abbildung 2.13: I-PD Struktur

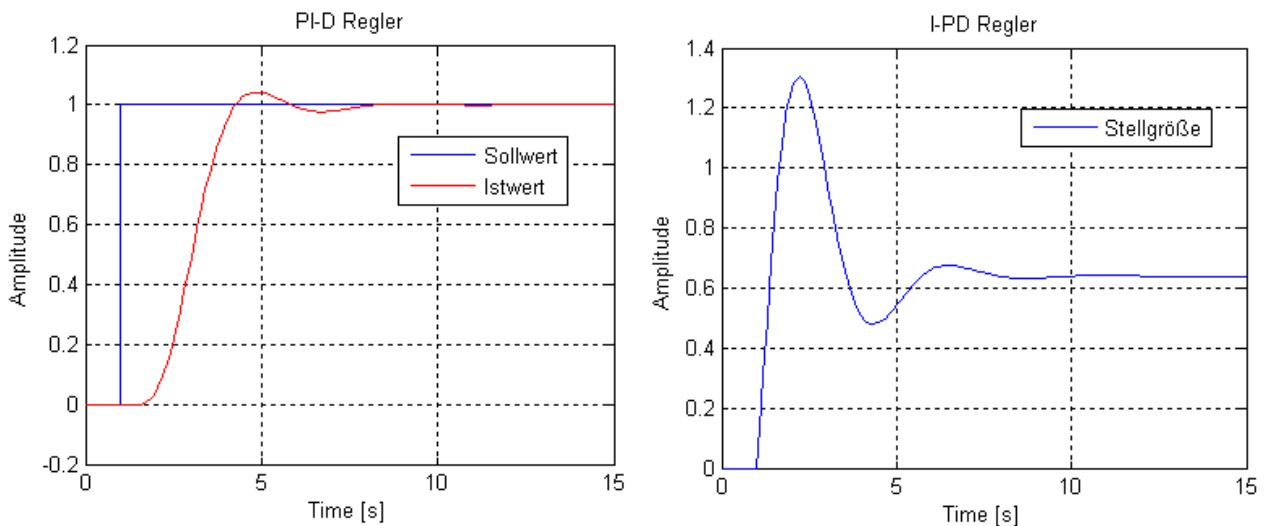


Abbildung 2.14: Verläufe bei I-PD Struktur

Bei der Implementierung des Reglers mit der I-PD Struktur ist aus den Simulationsergebnissen ersichtlich, dass sich für die verwendete Regelstrecke der Verlauf der Stellgröße und des Istwerts im Vergleich zur Abbildung 2.12 weiter verbessert hat. Die Ausgangsgröße weist ein geringeres Überschwingen und ein schnelleres Erreichen des stationären Endwerts auf.

Aus den zuvor gezeigten Simulationsergebnissen kann man anschaulich erkennen, dass sich diese alternativen Regelstrukturen eines PID-Reglers meistens besser für die Verwendung in der Industrie eignen. Reale Stellglieder werden mit den gezeigten Implementierungen bei einer sprungförmigen Änderung der Führungsgröße nicht so stark belastet und somit kann dies sogar zu einer Verlängerung ihrer Lebensdauer beitragen.

2.4 Anti-Wind-up

2.4.1 Stellgrößenbeschränkung

Eine wesentliche Nichtlinearität im Betrieb von Regelkreisen ist die Beschränkung der Stellgröße. Jedes reale Stellglied hat eine obere und untere Grenze (Ventil: offen/geschlossen, Motor: Stillstand/Maximalleistung usw.), über welche die Stellgröße nicht hinaus verändert werden kann. Diese Beschränkung führt zu einem erheblichen Problem, dem sogenannten „Wind-up“ Effekt, bei Reglern mit einem Integral-Anteil.

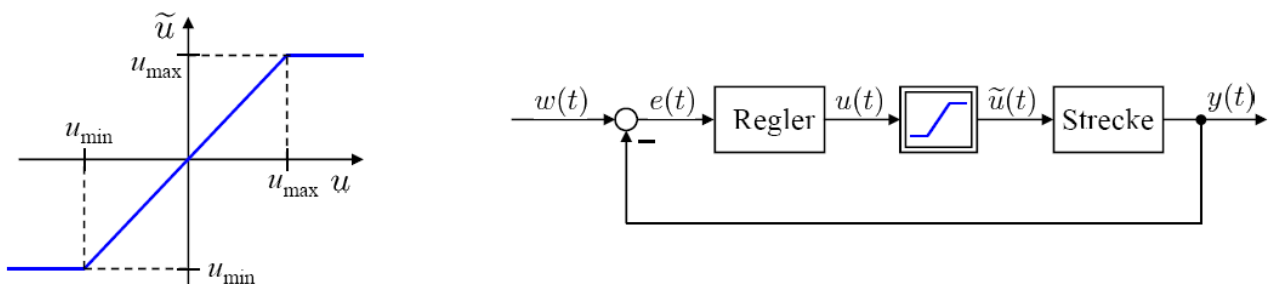


Abbildung 2.15: Stellgrößenbeschränkung bei Regelkreisen aus [9]

2.4.2 Wind-up Effekt

Nimmt man zum Beispiel einen PI-Regler an, bei dem eine positive Stellgröße ausgegeben wird, wächst der I-Anteil an, solange die Regelabweichung positiv ist. Übersteigt nun die Stellgröße die Grenze u_{max} der Stellgrößenbeschränkung, gibt es eine Abweichung zwischen der Wunschstellgröße $u(t)$ und der tatsächlich vom Stellglied ausgegebenen Größe u_{max} . Der I-Anteil des Reglers steigt nun solange weiter an, bis die Regelabweichung null wird. Während der Zeit, in der die Stellgröße an ihrer oberen Beschränkung festgehalten wurde, hat sich die Wunschstellgröße immer weiter nach oben entfernt. Wenn nun die Regelabweichung negativ wird, braucht der I-Anteil lange Zeit, bis er wieder abgebaut ist.

Je kleiner die Begrenzung der Stellgröße ist, desto schlechter wird die Regelgüte des geschlossenen Regelkreises. Der Grund dafür liegt an zwei Faktoren:

- Der Eingriff der Stellgröße wird vermindert.
- Wind-up Effekt: Der I-Anteil integriert lange auf und die Wunschstellgröße hängt geraume Zeit über u_{max} fest.

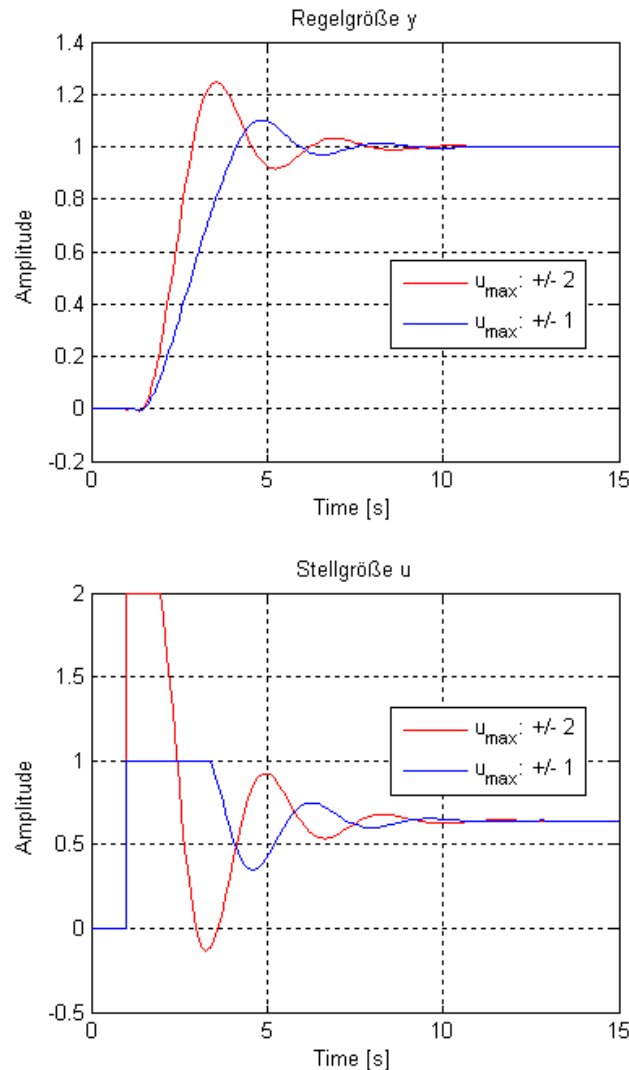


Abbildung 2.16: Verschlechterung der Regelgüte durch Wind-up Effekt

Dieser unerwünscht lange Ausgleichsvorgang kann mittels Anti-Wind-up Maßnahmen verkürzt werden [15]. Mittels einer Rückkopplung der Differenz zwischen der Wunschstellgröße $u(t)$ und der tatsächlichen Stellgröße wirkt man dem Wind-up Effekt entgegen. Der I-Anteil wird nun bei einer Überschreitung der Stellgrößenbeschränkung „abgeschnürt“ und somit die weitere Integration verhindert bzw. abgeschwächt.

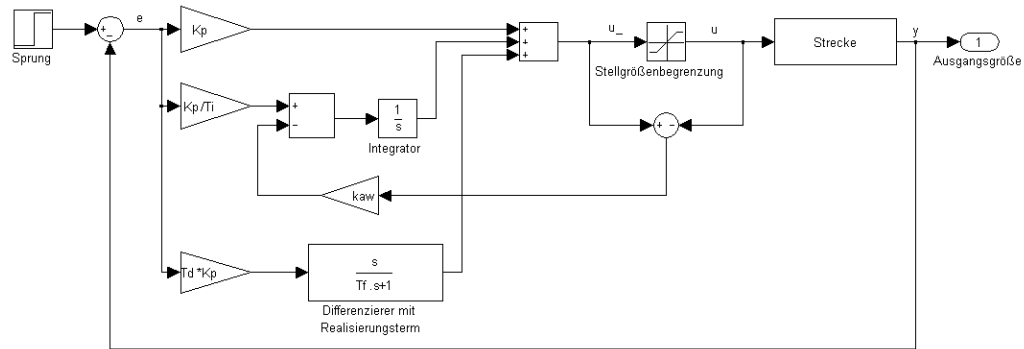


Abbildung 2.17: Strukturbild der Anti-Wind-up Maßnahme

Der Verstärkungsfaktor k_{AW} in der Rückkopplung wird häufig empirisch in der Simulation ermittelt, da noch keine allgemein gültigen Regeln für die Einstellung dieses Faktors vorhanden sind.

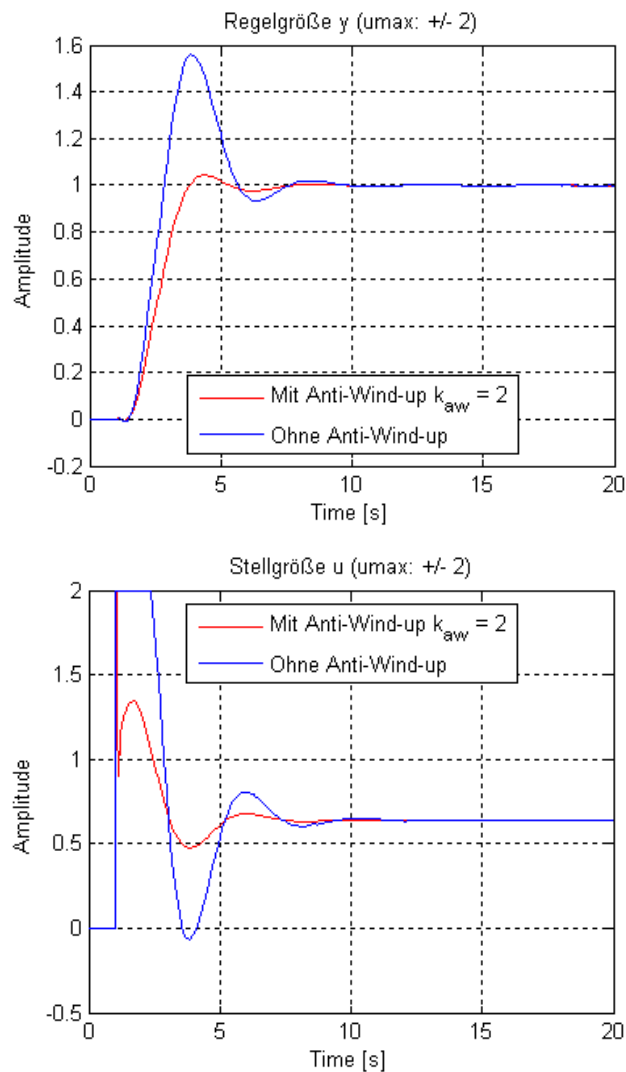


Abbildung 2.18: Verbesserung der Regelgüte durch Anti-Wind-up

3 Entwurfsverfahren

Für die Einstellung eines PID-Reglers gibt es eine Reihe von Entwurfsverfahren, wobei jedes Verfahren meist nur für spezielle Strecken ein gutes Ergebnis liefert. Einige sind speziell für integrierende Strecken, Strecken mit Totzeit oder solche mit einem s-förmigen Verlauf der Sprungantwort entwickelt worden. Deshalb wurde in dieser Arbeit versucht, ein geeignetes Paket aus Verfahren zu implementieren, welches einen großen Bereich der möglichen Regelstrecken abdeckt. Da die meisten Entwurfsverfahren auf einer zeitkontinuierlichen Realisierung eines PID-Reglers beruhen, wird die Simulation zuerst mit einem zeitkontinuierlichen Aufbau durchgeführt. Für den Entwurf wird die gewählte zeitdiskrete Regelstrecke, aus dem MISO-Modell, in eine zeitkontinuierliche umgewandelt, da einige der verwendeten Verfahren ein solches benötigen. Die erhaltenen Reglerparameter können in weiterer Folge, bei einer hinreichend kleinen Abtastzeit des Reglers, auch für die in der Praxis benötigte zeitdiskrete Implementierung des PID-Reglers verwendet und gegebenenfalls angepasst werden. Im ersten Augenblick erscheint diese Vorgangsweise etwas umständlich, jedoch bietet sie den Vorteil, dass der Regler unabhängig von der Abtastzeit des zeitdiskreten Reglers entwickelt werden kann. Man vermag dadurch sofort zu beurteilen, ob das gewählte Verfahren für jeweilige Regelstrecke überhaupt eine Lösung finden kann und muss sich nicht die Frage stellen, ob die Abtastzeit des Reglers für die Simulation zu groß gewählt wurde.

Auf die implementierten Verfahren wird nun in diesem Kapitel etwas näher eingegangen und ihre Anwendbarkeit anhand eines Testbeispiels verifiziert. Für die Simulation des Regelkreises wird ein zeitkontinuierlicher Aufbau des PID-Reglers verwendet.

Als Teststrecke dient hierbei die Übertragungsfunktion

$$P(z) = \frac{-0.00449z + 0.01373}{z^2 - 1.846z + 0.8521} z^{-3}$$

mit einer Abtastzeit von 0.1 Sekunden. Für den Entwurf wird die Regelstrecke mittels des MATLAB-Befehls „d2c“ in ein zeitkontinuierliches Modell konvertiert und die Totzeit wird mittels einer Pade-Approximation⁵ 2. Ordnung angenähert. Somit erhalten wir die zeitkontinuierliche Übertragungsfunktion

$$P(s) = \frac{-0.1s^3 + 3s^2 - 33.33s + 133.3}{s^4 + 21.6s^3 + 166s^2 + 226.1s + 85.33}$$

Die Zeitkonstante für den Realisierungsterm des D-Teils wird bei den meisten Entwurfsverfahren mit $T_f = \frac{T_D}{20}$ definiert. Die Ausnahmen bilden die in weiterer Folge erläuterten Verfahren k150 und gKYP, da die Zeitkonstante im Entwurf optimiert wird bzw. bei der Konfiguration eingestellt werden kann.

⁵ Matlab-Befehl „pade“ z.B.: $e^{-sT_t} \approx \frac{1 - \frac{T_t}{2}s + \frac{1}{12}T_t^2s^2}{1 + \frac{T_t}{2}s + \frac{1}{12}T_t^2s^2}$ Pade-Approximation 2.Ordnung. T_t ...Totzeit

3.1 Ziegler Nichols

Die Ziegler Nichols Design-Methode ist eine der populärsten heuristischen Methoden zur Ermittlung der Parameter eines PID-Reglers⁶. Sie werden anhand von Experimenten mit der Regelstrecke und dem Regelkreis festgelegt. In der Praxis wird diese Entwurfsmethode eigentlich für Regelstrecken, bei denen das mathematische Modell nicht bekannt ist, angewendet. Da in dieser Arbeit die Regelstrecke als Übertragungsfunktion zur Verfügung steht, können die einzelnen Parameter für den Entwurf aus der Sprungantwort bzw. aus dem Bode-Diagramm ermittelt werden. Man unterscheidet grundsätzlich zwischen zwei Verfahren.

3.1.1 Verfahren 1: „open loop method“

Zur Reglerdimensionierung wird hier die Sprungantwort der Strecke analysiert. Wie in Abbildung 3.1 angedeutet, werden die Kenngrößen a und T_v (Verzugszeit) aus dem Verlauf der Sprungantwort abgelesen.

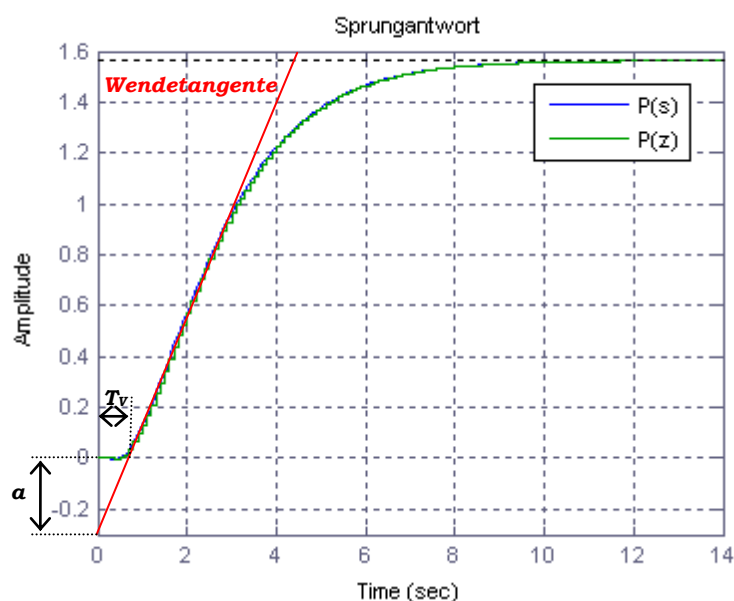


Abbildung 3.1: Bestimmung der Kenngrößen für die „open loop“ Methode

Hierfür ist an die Sprungantwort die Wendetangente zu legen. Mit den gefundenen Parametern kann man aus der folgenden Tabelle die Einstellungen für den gewünschten Reglertyp ablesen.

Reglertyp	K_P	T_I	T_D
P	a^{-1}	-	-
PI	$0.9 a^{-1}$	$3 T_V$	-
PID	$1.2 a^{-1}$	$2 T_V$	$0.5 T_V$

Tabelle 1: Einstellregeln Ziegler Nichols „open loop“

⁶ Nähere Informationen können aus [4], [8], [9] und [11] entnommen werden.

3.1.2 Verfahren 2: „closed loop method“

Hier wird die Dynamik des geschlossenen Regelkreises untersucht. Als Regler wird zunächst ein einfaches P-Glied verwendet.

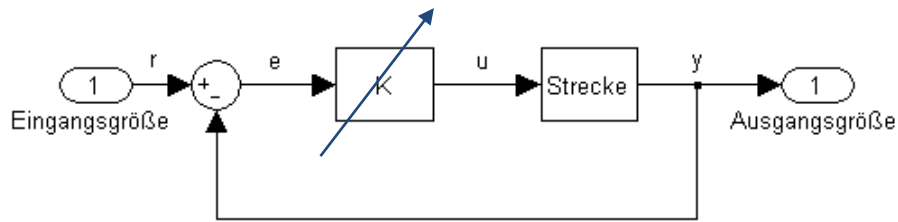


Abbildung 3.2: Regelkreis „closed loop“ Methode

Der Verstärkungsfaktor wird so eingestellt, dass der Regelkreis BIBO-stabil ist. Danach wird der Verstärkungsfaktor solange erhöht, bis die Ausgangsgröße der Strecke bei einer sprungförmigen Änderung der Führungsgröße eine Dauerschwingung vollführt (siehe Abbildung 3.3). Den entsprechenden kritischen Verstärkungsfaktor nennt man K_U (*ultimate gain*), die kritische Periodendauer der Dauerschwingung wird mit T_U (*ultimate period*) bezeichnet. Mit diesen beiden Werten, lassen sich aus der folgenden Tabelle die Reglerparameter berechnen. Dieses Verfahren lässt sich, falls kein Modell vorhanden ist, nur an realen Anlagen, bei denen die Stabilitätsgrenze gefahrlos erreicht werden kann, anwenden!

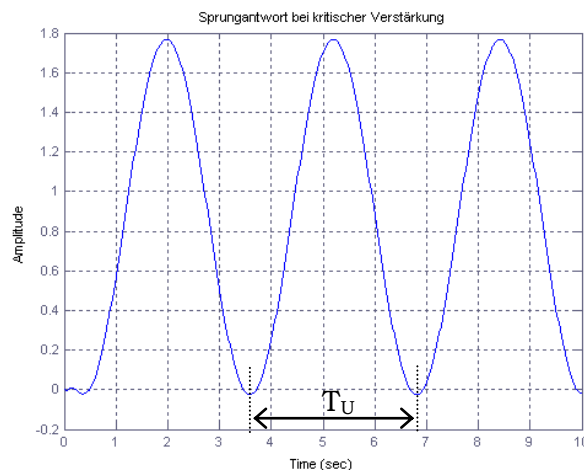


Abbildung 3.3: Dauerschwingen bei der „closed loop“ Methode

In der folgenden Tabelle können die Reglerparameter in Abhängigkeit von K_U und T_U abgelesen werden.

Reglertyp	K_P	T_I	T_D
P	$0.5 K_U$	-	-
PI	$0.4 K_U$	$0.8 T_U$	-
PID	$0.6 K_U$	$0.5 T_U$	$0.12 T_U$
PD	$0.55 K_U$	-	$0.15 T_U$

Tabelle 2: Einstellregeln Ziegler Nichols „closed loop“

Eine weitere Möglichkeit, den kritischen Verstärkungsfaktor und die kritische Periodendauer zu erhalten, bietet, falls für die jeweilige Strecke vorhanden, das Bode-Diagramm der zu regelnden Strecke. Daraus lassen sich relativ einfach die notwendigen Parameter für den Reglerentwurf ablesen. Die kritische Verstärkung ist jene Verstärkung, um welche die Betragskennlinie angehoben werden muss, damit die Stabilitätsgrenze erreicht wird. Dies ist erfüllt, wenn der Amplitudengang die 0 dB Linie genau bei jener Frequenz schneidet, bei der die Phasendrehung -180° beträgt.

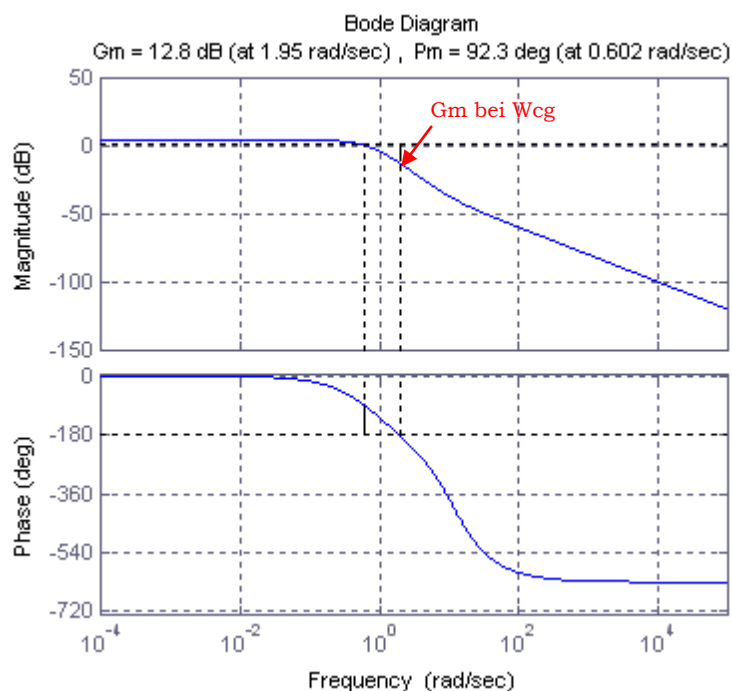


Abbildung 3.4: Ermittlung der Kenngrößen „closed loop“ Methode aus dem Bode-Diagramm

Die kritische Verstärkung für den Entwurf liegt bei diesem Beispiel bei $G_{m,dB} = 12.8 \text{ dB}$ und lässt sich mittels $G_m = K_U = 10^{(12.8/20)} = 4.3360$ in die notwendige Form umrechnen. Die Periodendauer der Dauerschwingung T_U erhält man mittels $T_U = \frac{2\pi}{\omega_{cg}} = \frac{2\pi}{1.95} = 3.2196$.

Die Berechnung der Reglerparameter erfolgt unter MATLAB mit der Funktion:

`function [Kp, Ti, Td, Tf, typ, method] = znPID(P, tfinal, w, typ_R, idx)`

Die Ausgabe-Parameter sind bei allen weiteren Entwurfsverfahren gleich und werden deshalb nur an dieser Stelle einmal erläutert.

Ausgabe-Parameter:

- Kp: Verstärkung
- Ti: Integrierzeit (Nachstellzeit T_N)
- Td: Differenzierzeit (Vorhaltezeit T_V)
- Tf: Zeitkonstante für den Realisierungsterm
- Typ: Reglertyp: P, PI, PID, PD
- method: Name des jeweiligen Entwurfsverfahrens

Eingabe-Parameter:

- P: Verwendete Regelstrecke
- tfinal: Zeitvektor für die Sprungantwort
- w: Frequenzvektor für das Bode-Diagramm
- typ_R: Auswahl des zu entwerfenden Reglers: P, PI, PID, (PD)
- idx: 1 → „closed loop“ Methode
2 → „open loop“ Methode

3.1.3 Beispiel zum Entwurf mit Ziegler Nichols

„open loop“ Methode:

Methode:	Ziegler Nichols „open loop“	
PID	K_P	3.4497
	T_I	1.5088
	T_D	0.3772
	T_f	0.0189
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 3: Reglerparameter Testbeispiel Ziegler Nichols „open loop“

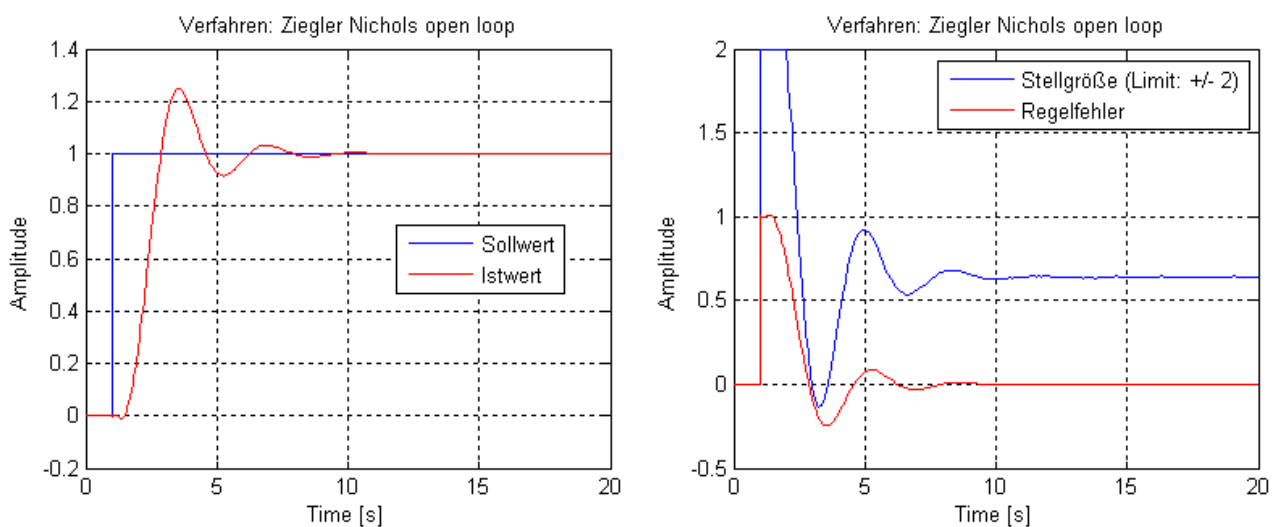


Abbildung 3.5: Simulationsergebnis mit Ziegler Nichols „open loop“

Aus den Ergebnissen des Reglerentwurfs mit der „open loop“-Methode nach Ziegler Nichols, ist ein ca. 20 %iges Überschwingen der Regelgröße ersichtlich. Weiters wird die obere Schranke der Stellgrößenbegrenzung erreicht und die Anti-Wind-up Maßnahme kommt hierbei zu tragen. Dies könnte jedoch gegebenenfalls durch Verwendung einer alternativen Reglerstruktur, wie PI-D oder I-PD, vermieden werden.

„closed loop“ Methode:

Methode:	Ziegler Nichols „closed loop“	
PID	K_P	2.5683
	T_I	1.6098
	T_D	0.4025
	T_f	0.0201
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 4: Reglerparameter Testbeispiel Ziegler Nichols „closed loop“

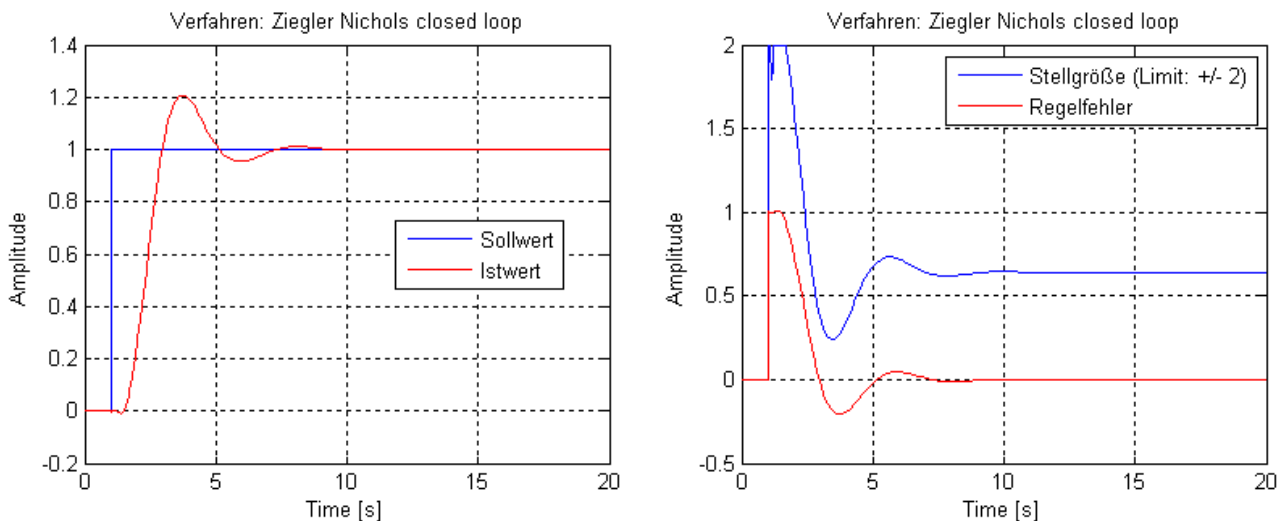


Abbildung 3.6: Simulationsergebnis mit Ziegler Nichols „closed loop“

Die „closed loop“-Methode liefert, im Gegensatz zur „open loop“-Methode, ein etwas besseres Ergebnis. Die Ausgangsgröße des geschlossenen Regelkreises weist ein geringeres Überschwingen und eine kürzere Ausregelzeit auf. Der erhaltene Regler könnte u. U. für die Regelung in der realen Anlage bereits verwendet werden.

3.2 Chien Hrones Reswick

Die Einstellregeln nach Chien Hrones Reswick⁷ sind mit jenen von Ziegler Nichols (open loop) vergleichbar. Der Vorteil bei diesem Entwurf ist, dass man Einfluss auf das Einschwingverhalten der Ausgangsgröße des Regelkreises hat. Man kann beim Entwurf den Wunsch eines aperiodischen Verlaufs oder eines mit 20 Prozent Überspringen vorgeben. Außerdem gibt es die Möglichkeit, den Regler auf ein gutes Führungsverhalten oder auf ein gutes Störverhalten auszulegen. In dieser Arbeit wurden nur jene Einstellregeln, welche ein gutes Störverhalten nach sich ziehen, verwendet, da die Ausgangsgrößen der restlichen Teilsysteme als Störgröße auf den Regelkreis wirken. Weiters wurde bei diesem Entwurfsverfahren noch eine Einstellregel nach Aeström und Hägglund [13] integriert. Dieses benötigt zu den bereits bekannten Parametern zur Reglereinstellung noch die Anstiegszeit. Sie ist hier jene Zeit, welche die Sprungantwort benötigt um von fünf Prozent bis siebenzig Prozent des stationären Endwertes zu gelangen.

	0 % Überspringen Aeström & Hägglund			0 % Überspringen			20 % Überspringen		
Reglertyp	K_P	T_I	T_D	K_P	T_I	T_D	K_P	T_I	T_D
P	$0.3 \alpha^{-1}$	-	-	$0.3 \alpha^{-1}$	-	-	$0.7 \alpha^{-1}$	-	-
PI	$0.6 \alpha^{-1}$	T	-	$0.6 \alpha^{-1}$	$4 T_V$	-	$0.7 \alpha^{-1}$	$2.4 T_V$	-
PID	$0.95 \alpha^{-1}$	$1.4 T$	$0.47 T_V$	$0.95 \alpha^{-1}$	$2.4 T_V$	$0.42 T_V$	$1.2 \alpha^{-1}$	$2 T_V$	$0.42 T_V$

Tabelle 5: Einstellregeln Chien Hrones Reswick

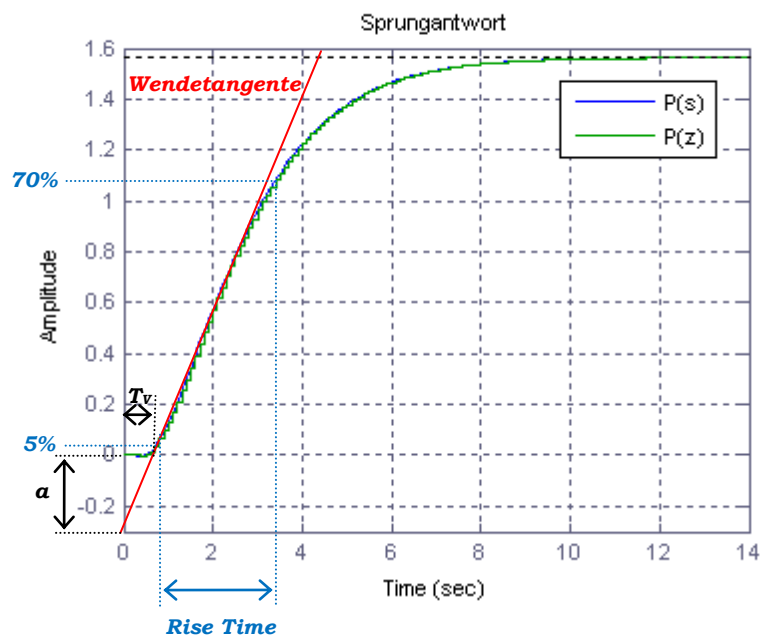


Abbildung 3.7: Ermittlung der Kenngrößen für Chien Hrones Reswick

⁷ Nähere Informationen können aus [4] und [11] entnommen werden

Die Berechnung der Reglerparameter erfolgt unter MATLAB mit der Funktion:

`function [Kp, Ti, Td, Tf, typ, method] = chrPID(P, tfinal, typ_R, overshoot)`

Eingabe-Parameter:

- P: Verwendete Regelstrecke
- tfinal: Zeitvektor für die Sprungantwort
- typ_R: Auswahl des zu entwerfenden Reglers: P, PI, PID
- overshoot: 1 → 0 % Überschwingen, Aeström und Hägglund Methode
2 → 0 % Überschwingen, Chien Hrones Reswick Methode
3 → 20 % Überschwingen, Chien Hrones Reswick Methode

3.2.1 Beispiel zum Entwurf mit Chien Hrones Reswick

Methode:	Chien Hrones Reswick 20 %	
PID	K_P	2.7310
	T_I	1.8105
	T_D	0.3168
	T_f	0.0158
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 6: Reglerparameter Testbeispiel Chien Hrones Reswick

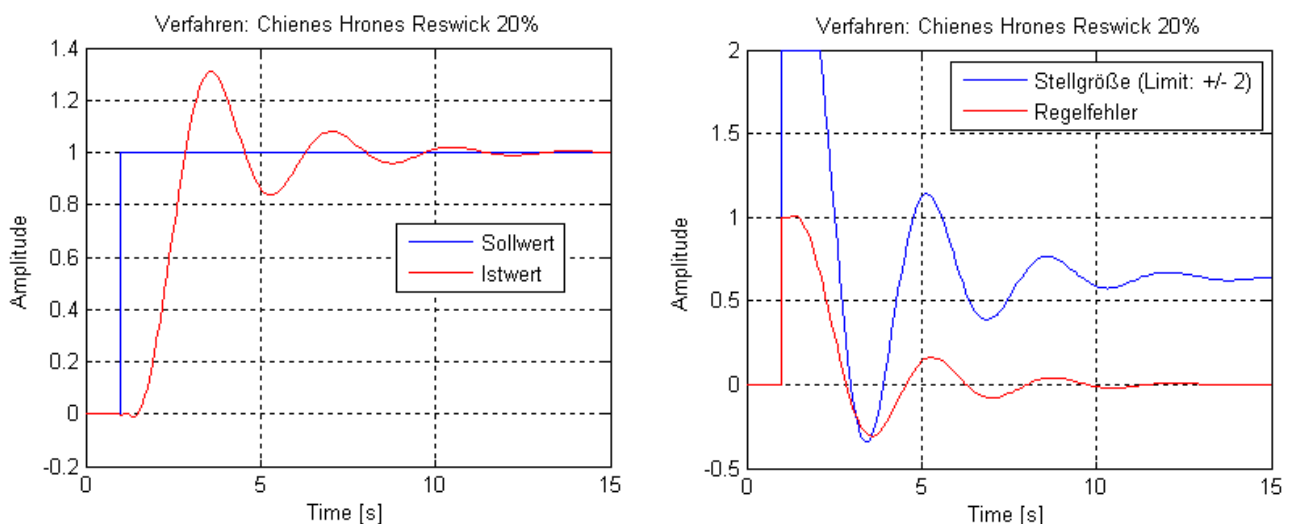


Abbildung 3.8: Simulationsergebnis mit Chien Hrones Reswick

Das Verfahren nach Chien Hrones Reswick liefert ein sehr unbefriedigendes Ergebnis. Der Wunsch von einem 20 %igen Überschwingen konnte nicht erfüllt werden und die Ausregelzeit von ca. 15 Sekunden hat sich, im Vergleich zu den vorherigen Ergebnissen, erheblich vergrößert. Aus diesem Grund eignet sich der erhaltene Regler für die geforderte Aufgabe nicht.

3.3 k150-Verfahren

Das k150-Verfahren beruht auf den Überlegungen aus [1] und bietet eine einfache Möglichkeit zum Entwurf von P-, PI- und PID-Reglern. Jedoch mit der Einschränkung, dass die Regelstrecke eine Phasendrehung von mindestens -150° aufweisen muss⁸. Dieses Einstellverfahren eignet sich sehr gut für Strecken mit Totzeiten, kann speziell mit integrierenden Strecken umgehen und optimiert im Entwurf den Realisierungsterm des D-Anteils.

Vier verschiedene Kriterien werden im Entwurf berücksichtigt:⁹

1.) Kriterium für Störunterdrückung:

$$J_v = \left\| \frac{1}{s} \left(\frac{P(s)}{1 + L(s)} \right) \right\|_{\infty}$$

2.) Kriterium für Robustheit:

$$GM_s = \max \left(\|S(s)\|_{\infty}, \alpha \|T(s)\|_{\infty} \right)$$

3.) Stellgrößen-Kriterium:

$$J_u = \left\| \frac{R(s)}{1 + L(s)} \right\|_{\infty}$$

4.) HF Kriterium:

$$J_{HF} = \left\| s^m \frac{R(s)}{1 + L(s)} \right\|_{\infty}$$

P(s)... Übertragungsfunktion der Regelstrecke

R(s)... Übertragungsfunktion des Reglers

L(s)... Übertragungsfunktion des offenen Kreises $L(s) = R(s) P(s)$

S(s)... Störübertragungsfunktion $S(s) = \frac{1}{1+L(s)}$

T(s)... Führungsübertragungsfunktion $T(s) = \frac{L(s)}{1+L(s)}$

⁸ Streckenordnung ≥ 2 , bzw. eine positive Nullstelle bei Systemen 1. Ordnung

⁹ H_{∞} - Norm: z.B.: $\|G\|_{\infty} = \max_{\omega} |G(j\omega)| \rightarrow$ Maximalwert im Amplitudengang von $G(j\omega)$

Das J_V Kriterium ist ein Maß für die Fähigkeit des geschlossenen Regelkreises mit niederfrequenten Störungen umzugehen. Dieses Kriterium ist eine Alternative im Frequenzbereich für Kriterien basierend auf dem Integral des Regelfehlers, z.B. $IAE = \int_0^\infty |e(t)|dt$, im Zeitbereich. Das Kriterium für Robustheit mit der generalisierten maximalen Sensitivität GM_S ist eine Kombination der Beschränkung des Maximalwertes von $\|S\|_\infty = \max_\omega |S(j\omega)| \leq M_S$ und $\|T\|_\infty = \max_\omega |T(j\omega)| \leq M_T$. Den Komponenten von GM_S entsprechen zwei Kreise in der komplexen Ebene $\|S\|_\infty = M_S = 1.7$ und $\|T\|_\infty = M_T = 1.3$, welche für dieses Verfahren einen Stabilitätsrand im Nyquist-Diagramm für $L(j\omega)$ vorgeben. Die Komponente M_S garantiert, dass der Abstand von $L(j\omega)$ zum kritischen Punkt $-1 + j0$ immer größer als $1/M_S$ ist. Der gewählte Wert für M_T gewährleistet eine Phasenreserve von mindestens 45° . Der Parameter $\alpha = M_S/M_T$ transformiert die Beschränkung $\|T\|_\infty$ zu einer gleichwertigen Beschränkung $\|S\|_\infty$.

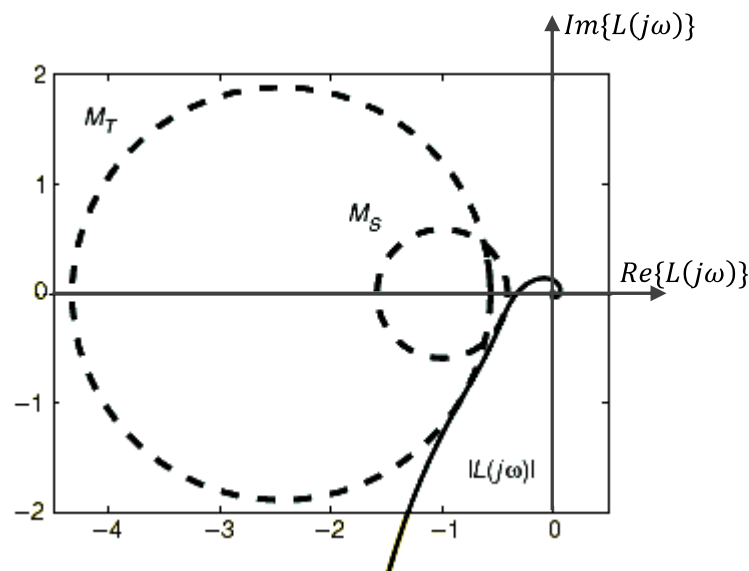


Abbildung 3.9: Darstellung des Stabilitätsrandes für $L(j\omega)$ entnommen aus [1]

	Zentrum	Radius
M_S -Kreis	-1	$\frac{1}{M_S}$
M_T -Kreis	$-\frac{M_T^2}{M_T^2 - 1}$	$\frac{M_T}{M_T^2 - 1}$

Das Kriterium J_U bezieht sich auf die mittleren bis hohen Frequenzen, in deren Nähe sich meist die Bandbreite des geschlossenen Regelkreises befindet. Der Betrag des Frequenzganges von $S_U(s) = \frac{R(s)}{1+L(s)}$ erreicht vorwiegend in diesem Bereich sein Maximum. Für das HF Kriterium sind zwei Ziele, nämlich die Robustheit gegenüber Modellunsicherheiten und die Reduktion des Sensorrauschens, vorrangig. Beruhend auf dem „small gain theorem“ [12], kann bei einer Strecke mit einer signifikanten Modellunsicherheit, durch Kleinhalten der H_∞ -Norm der Übertragungsfunktion $S_U(s)$ die Stabilität des geschlossenen Regelkreises gewährleistet werden. Für nähere Erklärungen der verschiedenen Kriterien wird hier auf [1] verwiesen.

Der traditionelle PID-Regler der Form

$$K_{PID}(s) = K_P \left(1 + \frac{1}{sT_I} + \frac{sT_D}{1 + sT_f} \right),$$

optimiert mit einem Filter und alle Parameter frei wählbar, hat meist komplexe Nullstellen zur Folge. Deshalb wird bei diesem Verfahren der Regler in der Form

$$R_{PID}(s) = k_i \frac{1 + 2\zeta\tau s + (s\tau)^2}{s \left(1 + \frac{s\tau}{\beta} \right)}$$

angeschrieben.

Die vier Entwurfsparameter lauten:

- 1.) ζ Dämpfung
- 2.) τ Zeitkonstante der Reglernullstellen
- 3.) k_i Verstärkung des I-Teils
- 4.) $\beta = k_\infty / (\tau k_i) \rightarrow k_\infty = \lim_{\omega \rightarrow \infty} |R_{PID}(j\omega)| = R_{PID}(\infty)$ HF Verstärkung

Die Einstellregeln dieses Verfahrens verwenden den Parameter $k_{150} = \frac{|G(j\omega_{150G})|}{G(0)}$, wobei mit ω_{150G} jene Frequenz gemeint ist, bei der die Phase der Strecke -150° beträgt. Dieser Parameter charakterisiert die Dynamik der Regelstrecke und somit können einfache Einstellregeln angegeben werden.

Bei integrierenden Strecken wird als Parameter $k_{i150} = \frac{\omega_{150G} |G(j\omega_{150G})|}{\lim_{\omega \rightarrow 0} \omega G(j\omega)}$ verwendet. Da nicht jede Strecke, die bei einem Reglerentwurf auftritt, eine Phasendrehung bis -150° aufweist, ist dieses Verfahren nur bedingt einsetzbar.

PID-Einstellregeln für stabile nichtoszillierende Strecken:

$$\zeta = 0.75$$

$$\frac{1}{\tau} = \omega_{150G} (0.44 + 0.86 k_{150})$$

$$k_\infty = \frac{1}{G(0)} \min \left(3 + \frac{2}{k_{150}}, 25 \right)$$

$$k_i = \frac{\omega_{150G}}{G(0)} \left(\frac{0.45}{k_{150} + 0.07} - 0.1 \right)$$

PI-Einstellregeln für Strecken mit einer Ordnung ≥ 2 :

$$\frac{1}{\tau} = \omega_{150G} (0.06 + 1.6 k_{150} - 0.06 k_{150}^2)$$

$$k_i = \frac{\omega_{150G}}{G(0)} \left(0.2 + \frac{0.075}{k_{150} + 0.05} \right)$$

$$\beta = 1 \text{ und } \zeta = 1$$

PID-Einstellregeln für integrierende Strecken

$$\text{für } k_{i150} \geq 0.7$$

$$\beta = 20, \quad \zeta = 1/(3.5 - 3 k_{i150})$$

$$\frac{1}{\tau} = 0.6 \omega_{150G}, \quad k_i = \frac{\omega_{150G}^2}{\lim_{\omega \rightarrow 0} \omega G(j\omega)} e^{6.4 - 9 k_{i150}}$$

$$\text{für } k_{i150} < 0.7$$

$$k_\infty = \frac{20 \omega_{150G}}{\lim_{\omega \rightarrow 0} \omega G(j\omega)}, \quad \zeta = \frac{1}{3.5 - 3 k_{i150}}$$

$$\frac{1}{\tau} = 0.68 \omega_{150G}, \quad k_i = \frac{\omega_{150G}^2}{\lim_{\omega \rightarrow 0} \omega G(j\omega)} e^{4.4 - 6.5 k_{i150}}$$

Mittels einfacher Umrechnungsvorschriften, erhält man wieder die traditionellen Reglerparameter K_P , T_I , T_D , T_f .

$$T_f = \frac{\tau}{\beta}$$

$$T_I = \frac{\tau}{\beta} (2\zeta\beta - 1)$$

$$T_D = \frac{\tau}{\beta} \left(\frac{\beta^2}{2\zeta\beta - 1} - 1 \right)$$

$$K_P = k_i \frac{\tau}{\beta} (2\zeta\beta - 1)$$

Die Berechnung der Reglerparameter erfolgt unter MATLAB mit der Funktion:

`function [Kp, Ti, Td, Tf, typ, method] = k150(P, w, typ_R)`

Eingabe-Parameter:

- **P:** Verwendete Regelstrecke
- **w:** Frequenzvektor für das Bode-Diagramm
- **typ_R:** Auswahl des zu entwerfenden Reglers: P, PI, PID

3.3.1 Beispiel zum Entwurf mit k150

Methode:	k150	
PID	K_P	1.5802
	T_I	1.4772
	T_D	0.8435
	T_f	0.1972
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 7: Reglerparameter Testbeispiel k150

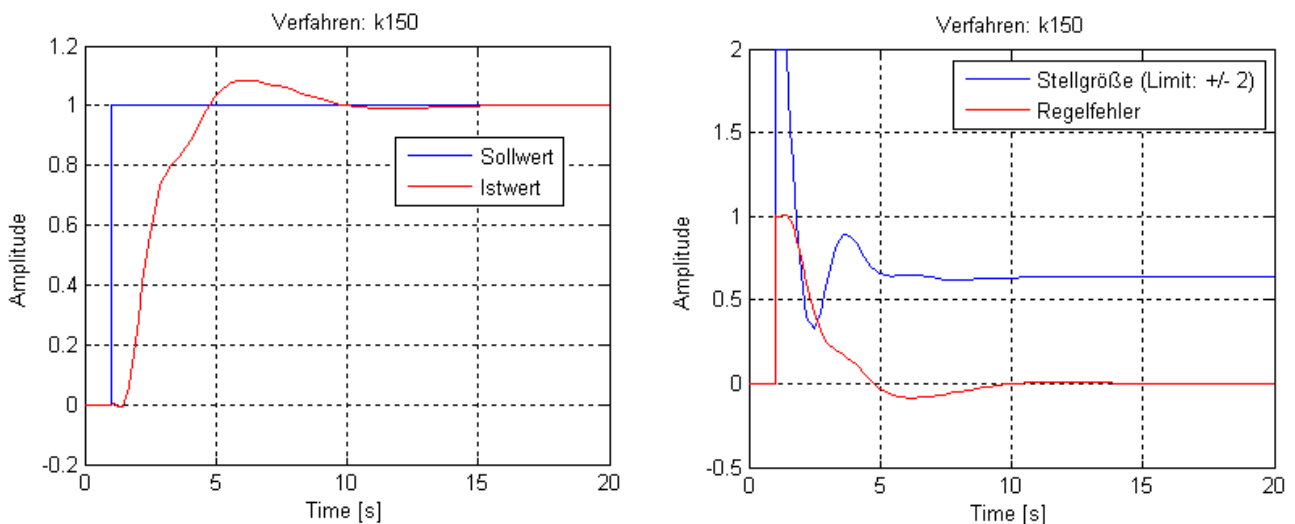


Abbildung 3.10: Simulationsergebnis mit dem k150-Verfahren

Aus den Simulationsergebnissen des geschlossenen Regelkreises mit dem k150-Verfahren ist ersichtlich, dass der erhaltene Regler bisher das geringste Überschwingen aufweist, jedoch im Vergleich zum Ziegler Nichols „closed loop“ Regler etwas länger zum Ausregeln benötigt.

3.4 gKYP-Lemma

Das folgende Entwurfsverfahren wurde in [2] erstmals angegeben und beruht auf dem sogenannten generalisierten Kalman-Yakubovich-Popov Lemma. Dabei werden zur Lösung semidefinite Programme (SDP) verwendet.

Für lineare zeitinvariante Systeme können viele Analyseprobleme als Frequenzbereichsungleichung (FDI)

$$G(j\omega)^* \Pi G(j\omega) < 0 \quad \forall \omega \in \mathbb{R}$$

formuliert werden. Wobei Π eine reelle symmetrische Matrix ist und $G(j\omega) \rightarrow G(s) = C(sE - A)^{-1}B + D$.¹⁰

Eine Möglichkeit die FDI zu überprüfen, ist die Frequenzachse zu rastern und zu kontrollieren, ob die Ungleichung $G(j\omega_k)^* \Pi G(j\omega_k) < 0$ für alle Frequenzpunkte ω_k erfüllt ist.

Um Approximationen, verursacht durch die Rasterung des Frequenzbereiches, zu vermeiden, wird beim KYP-Lemma die Frequenzvariable ω eliminiert und durch eine Variable („multiplier“ Matrix P) ersetzt. Das KYP-Lemma besagt, dass die FDI gilt, wenn eine symmetrische Matrix P existiert, welche die lineare Matrizenungleichung (LMI)

$$\begin{bmatrix} PA + A^T P & PB \\ B^T P & 0 \end{bmatrix} + \begin{bmatrix} C^T \\ D^T \end{bmatrix} \Pi \begin{bmatrix} C & D \end{bmatrix} < 0$$

erfüllt und die Matrix A keine Eigenwerte auf der imaginären Achse besitzt. Dadurch kann für das Analyseproblem die Überprüfung der FDI, in eine Suche nach einer Matrix P umgewandelt werden, welche die LMI erfüllt. Lineare Matrizenungleichungen können numerisch effizienter gelöst werden und falls eine Lösung für das LMI gefunden wurde, ist auch die FDI erfüllt. Bedauerlicherweise gibt es eine erhebliche Einschränkung bei der Verwendung des KYP-Lemmas für praktische Entwurfsprobleme. Das Problem liegt darin, dass das KYP-Lemma nur für den gesamten Frequenzbereich ($0 \leq \omega \leq \infty$) und nicht für einen bestimmten Bereich, welcher für die Analyse von besonderem Interesse ist, angewendet werden kann.

Wie aus dem folgenden praktischen Entwurfsproblem für einen PID-Regler ersichtlich, sind mehrere FDI's für verschiedene Frequenzbereiche angegeben.

minimiere t_1

unter

$$|L(j\omega)| \leq t_h \quad \text{für } \omega_h \leq \omega \leq \infty$$

$$a \operatorname{Re}\{L(j\omega)\} + b \operatorname{Im}\{L(j\omega)\} + c \leq 0 \quad \text{für } 0 \leq \omega \leq \infty$$

$$\operatorname{Im}\{L(j\omega)\} \leq -t_1 \quad 0 \leq \omega \leq \omega_l$$

¹⁰ A, B, C und D sind Matrizen der Zustandsraumdarstellung eines mathematischen Modells

Die erste Bedingung für ein kleines t_h garantiert eine gute Robustheit gegenüber Modelunsicherheiten, welche typischerweise bei hohen Frequenzen existieren. Die zweite Bedingung sichert eine robuste Stabilität des geschlossenen Regelkreises. Die dritte Bedingung, stellt eine hohe Verstärkung bei niedrigen Frequenzen für eine geringe Regelabweichung und eine gute Unterdrückung von Störgrößen sicher.

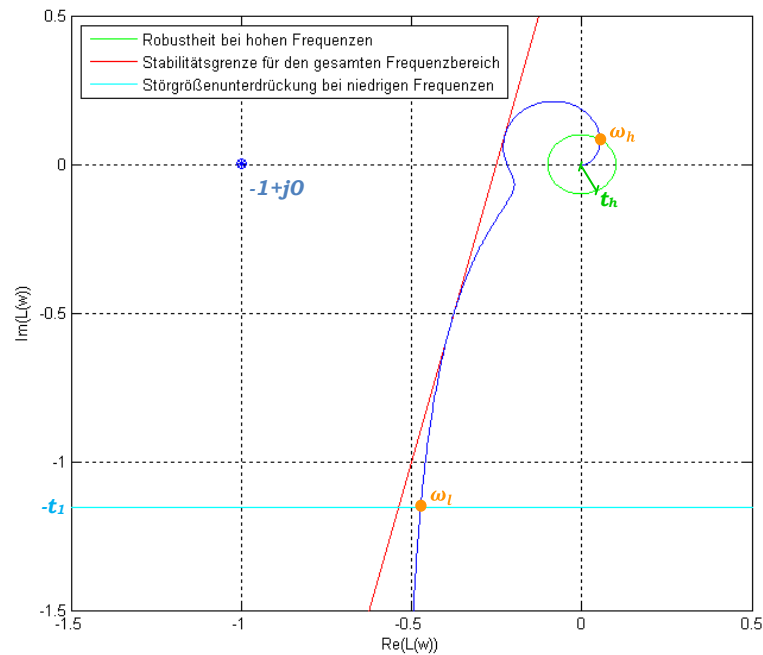


Abbildung 3.11: Nyquist-Diagramm des offenen Kreises $L(j\omega)$ mit den Entwurfsspezifikationen für „loop-shaping“

Diese Bedingungen der Übertragungsfunktion des offenen Regelkreises beruhen auf dem Entwurfsverfahren „loop-shaping“¹¹. Hierbei werden Forderungen an den offenen Regelkreis gestellt und mit Hilfe eines Reglers versucht, diese zu erfüllen. Der Vorteil der Verwendung der Übertragungsfunktion des offenen Kreises liegt darin, dass diese linear von den Übertragungsfunktionen R und P abhängt, während die Übertragungsfunktionen des geschlossenen Kreises über einen nichtlinearen Zusammenhang aus R und P gebildet werden.

Um nun das zuvor gezeigte Entwurfsproblem mit mehreren unterschiedlichen Frequenzbereichen mittels linearer Matrizenungleichungen zu lösen, wird die generalisierte Form des KYP-Lemmas benötigt. Ein Frequenzbereich kann im zeitkontinuierlichen Fall als Intervall auf der imaginären Achse ($\lambda = j\omega$) angegeben werden. Somit kann dieser als quadratische Gleichung und Ungleichung der Form

$$\begin{bmatrix} \lambda \\ 1 \end{bmatrix}^H \phi \begin{bmatrix} \lambda \\ 1 \end{bmatrix} = 0, \quad \begin{bmatrix} \lambda \\ 1 \end{bmatrix}^H \psi \begin{bmatrix} \lambda \\ 1 \end{bmatrix} \geq 0$$

charakterisiert werden.

¹¹ Nähere Informationen sind in [13] zu finden.

Das gKYP-Lemma besagt nun, dass für alle $\lambda \in \mathbb{C}$ die FDI gilt, wenn die Matrizen $P, Q \in \mathbf{H}^n$ existieren, welche die LMI's

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^H (\phi \otimes P + \psi \otimes Q) \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + M \preceq 0$$

$$Q \succeq 0$$

erfüllen.

Somit weist das SDP für die weitere Verwendung folgende Struktur auf [14]:

$$\text{minimiere} \quad w^T x$$

$$\begin{bmatrix} A_i & B_i \\ I & 0 \end{bmatrix}^T (\phi_i \otimes P_i + \psi_i \otimes Q_i) \begin{bmatrix} A_i & B_i \\ I & 0 \end{bmatrix} + M_i(x) + N_i \preceq 0$$

unter

$$Q_i \succeq 0, \quad i = 1, 2, \dots, L$$

wobei die Problemdata $w \in \mathbf{R}^p$, $A_i \in \mathbf{R}^{n_i \times n_i}$, $B_i \in \mathbf{R}^{n_i \times m_i}$, $\phi_i \in H^2$, $\psi_i \in H^2$, $N_i \in \mathbf{H}^{n_i \times m_i}$ und die Optimierungsvariablen $x \in \mathbf{R}^p$, $P_i \in \mathbf{H}_i^n$ und $Q_i \in H^{n_i}$ sind. \otimes ist das Kronecker Produkt.

Für den weiteren Verlauf des Entwurfs, wird die Reglerübertragungsfunktion der Form $R(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{(1+T_f)}$ in der Regelungsnormalform¹²

$$R(s) = C_R(sE - A_R)^{-1}B_R + D_R = \left[\begin{array}{c|c} A_R & B_R \\ \hline C_R & D_R \end{array} \right] = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & -\frac{1}{T_f} & 1 \\ \hline \frac{K_I}{s} & K_I - \frac{K_D}{T_f^2} & K_P + \frac{K_D}{T_f} \end{array} \right]$$

angeschrieben, wobei T_f die Zeitkonstante des Realisierungsterms für den Differenzierer ist.

Die Übertragungsfunktion des offenen Kreises $L(s) = P(s)R(s)$ lautet nun in Zustandsraumdarstellung:

$$L(s) = \left[\begin{array}{cc|c} A_P & 0 & B_P \\ B_R C_P & A_R & B_R D_P \\ \hline D_R C_P & C_R & D_R D_P \end{array} \right]$$

Für den Entwurf eines zeitkontinuierlichen Reglers ergibt sich die Matrix ϕ_i zu $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

¹² Auch Steuerbarkeitsnormalform bzw. 1 Normalform genannt. $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$ ist eine Kurzschreibweise für die Übertragungsfunktion $G(s) = C(sE - A)^{-1}B + D$.

Der Frequenzbereich der Entwurfsbedingungen wird mit den Matrizen ψ_i festgelegt.

$$\omega_h \leq \omega \leq \infty \rightarrow \psi_1 = \begin{bmatrix} 0 & j \\ -j & 2\omega_h \end{bmatrix}$$

$$0 \leq \omega \leq \infty \rightarrow \psi_2 = \begin{bmatrix} 0 & j \\ -j & 0 \end{bmatrix}$$

$$0 \leq \omega \leq \omega_l \rightarrow \psi_3 = \begin{bmatrix} -1 & j\frac{\omega_l}{2} \\ -j\frac{\omega_l}{2} & 0 \end{bmatrix}$$

Die Entwurfs-Spezifikationen werden in den Matrizen M_i und N_i angegeben.

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ und } N_1 = \begin{bmatrix} 0 & 0 \\ 0 & -t_h^2 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & a + jb \\ a - jb & 0 \end{bmatrix} \text{ und } N_2 = \begin{bmatrix} 0 & 0 \\ 0 & 2c \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & j \\ -j & 0 \end{bmatrix} \text{ und } N_3 = \begin{bmatrix} 0 & 0 \\ 0 & t_l \end{bmatrix}$$

Die erhaltenen Spezifikationen können nun dem Algorithmus übergeben und der Reglerentwurf gestartet werden. Als Standardwerte für den gKYP-Entwurf werden folgende Parameter verwendet:

$$\omega_l = 0.005, \quad \omega_h = 4, \quad t_h = 0.001, \quad a = -4, \quad b = 1, \quad c = -1, \quad T_f = 0.01.$$

Je nach dem dynamischen Verhalten der Strecke können einige Werte angepasst werden, um ein besseres Ergebnis zu erhalten. Der Frequenzbereich in dem der Amplitudengang des offenen Regelkreises optimiert werden soll, kann mit den Parametern ω_l und ω_h definiert werden. Die obere Grenze vom Betrag des offenen Kreises $|L(j\omega)|$, für den Frequenzbereich über ω_h , kann mit t_h angepasst werden. Wie bereits erwähnt, gewährleistet ein kleiner Wert für den Parameter t_h eine größere Robustheit im hohen Frequenzbereich. Weiters kann die Zeitkonstante T_f des Realisierungsterms für den D-Anteil des Reglers variiert werden.

Die Gewichtungen für den $Re\{L(j\omega)\}$ und den $Im\{L(j\omega)\}$, sowie der Parameter c , sind mit diesen Werten vorgegeben und können notfalls nur im m-File des Entwurfsverfahrens angepasst werden. Für nähere Erklärung dazu wird auf [2] verwiesen.

Zum Lösen des zuvor beschriebenen Problems, wird die gKYPSPD-Toolbox [14] benötigt. Sie ist eine freie, speziell für MATLAB erstellte Toolbox und muss vor der Benützung installiert werden. Nähere Informationen zur Installation siehe Kapitel 5.1.

Die Berechnung der Reglerparameter erfolgt unter MATLAB mit der Funktion:

```
function [Kp, Ti, Td, Tf, typ, method] = PID_gKYP(P, wl, wh, th, Tf)
```

Eingabe-Parameter:

- P: Verwendete Regelstrecke
 - wl:
 - wh:
 - th:
 - Tf:
- } Entwurfsparameter für die Ermittlung des Reglers

3.4.1 Beispiel zum Entwurf mit dem gKYP-Lemma

Methode:	gKYP	
PID	K_P	1.1753
	T_I	1.8429
	T_D	0.6561
	T_f	0.05
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 8: Reglerparameter Testbeispiel gKYP-Lemma

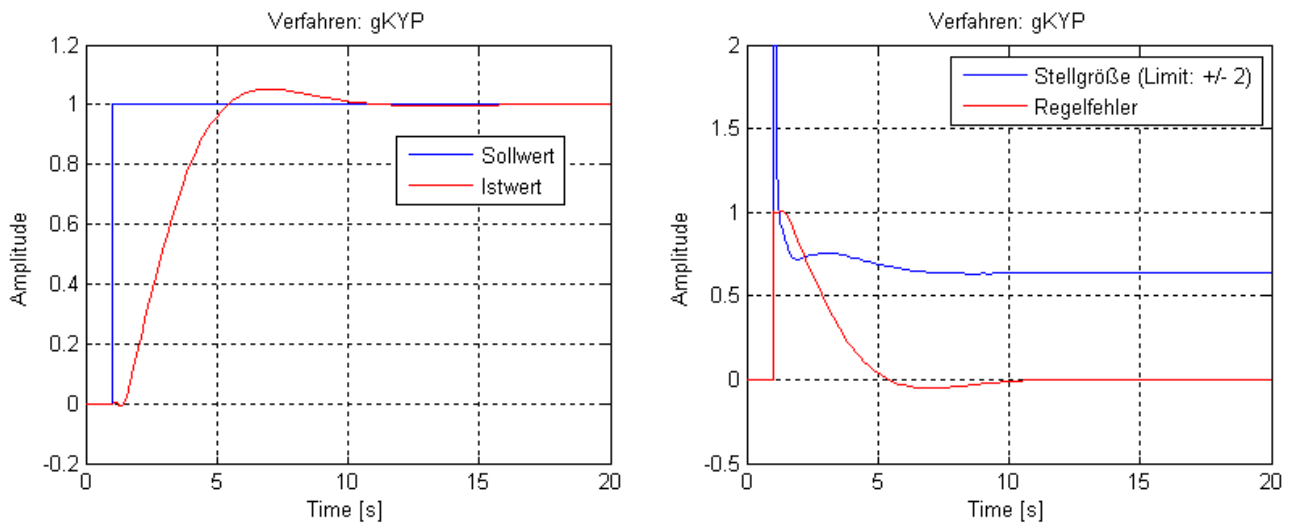


Abbildung 3.12: Simulationsergebnis mit dem gKYP-Verfahren

Das Verfahren mit dem gKYP-Lemma weist bis jetzt das beste Führungsverhalten auf. Die kurze Ausregelzeit und das geringe Überschwingen des geschlossenen Regelkreises, liefern im Vergleich, zu den bisherigen durchgeführten Verfahren, das beste Ergebnis. Der Verlauf der Stellgröße verweilt nur kurze Zeit an der oberen Grenze.

3.5 Minimierung von Integralkriterien

Ein weiteres Verfahren für den Reglerentwurf kann durch die Bewertung des Regelfehlers hergeleitet werden. Die Qualität des dynamischen Verhaltens eines Regelkreises kann mittels des Regelfehlers $e = r - y$ beurteilt werden. Man ist üblicherweise daran interessiert, den Regelfehler bei gegebenen Stör- bzw. Führungssignal möglichst klein zu machen. Hierfür wird eine Beschreibung für die Größe des Regelfehlers benötigt.

Integral über den Betrag des Regelfehlers (IAE)

Das Integral über den Regelfehlerbetrag (integral of absolute error)

$$\int_0^{\infty} |e(t)| dt$$

ist mathematisch schwierig auszuwerten, da die Betragsbildung eine nichtlineare Operation darstellt.

Integral über das Regelfehlerquadrat (ISE)

Das Gütekriterium (integral of square error)

$$\int_0^{\infty} e^2(t) dt$$

gewichtet einen großen Regelfehler besonders stark.

Der Nachteil der beiden zuvor erwähnten Kriterien besteht darin, dass die mit fortlaufender Zeit kleiner werdenden Amplituden den Integralwert kaum mehr beeinflussen. Durch Multiplikation des Regelfehler $e(t)$ mit der Zeit t können die kleineren Amplituden stärker berücksichtigt werden.

Integral über den zeitgewichteten Regelfehlerbetrag (ITAE)

Bei diesem Kriterium (integral of time multiplied by absolute error) wird eine zeitliche Gewichtung eingeführt, d.h.

$$\int_0^{\infty} t |e(t)| dt$$

Der Regelfehler wird also für kleine Zeitwerte weniger gewichtet als für große Werte von t . Man beeinflusst damit das stationäre Verhalten.

Integral über das zeitgewichteten Regelfehlerquadrat (ITSE)

Das Integral über das Quadrat des Regelfehlers mit einer zeitlichen Gewichtung (integral of time multiplied by square error)

$$\int_0^{\infty} t e^2(t) dt$$

Der Regelfehler wird für große Zeit- und Regelfehlerwerte besonders stark gewichtet. Hierbei wird wieder sehr stark das stationäre Verhalten beeinflusst.¹³

Die zuvor erläuterten Kriterien wurden in MATLAB mit der Funktion „iecost“ implementiert, welche von der Hauptfunktion in jeder Iteration aufgerufen wird und einen neuen Kostenfunktionswert berechnet. Der Optimierungsvektor x beinhaltet den Zähler der Übertragungsfunktion des Reglers.

z.B.: PID-Regler:

$$R = \frac{K_D s^2 + K_P s + K_I}{s} \rightarrow x = [K_D \ K_P \ K_I]$$

Er wird nun mittels der MATLAB-Funktion „fminunc“ so variiert, damit die Kostenfunktion minimiert wird.

`function J = icrost(x, P, den, t, dt ,idx)`

Eingabe-Parameter:

- x : Optimierungsvektor: Zähler des momentanen Reglers
- P : Regelstrecke
- den : Nenner des momentanen Reglers
- t : Zeitvektor der Sprungantwort
- dt : Zeitintervall
- idx : Auswahl des Kriteriums

Nach erfolgreicher Beendigung, können aus dem Vektor x die optimalen Reglerparameter für die gewählte Regelstrecke ermittelt werden.

Der Ablauf für den Reglerentwurf kann aus Abbildung 3.11 entnommen werden. Als Startwerte für die Optimierung stehen hierbei drei Auswahlmöglichkeiten zur Verfügung:

- 1.) gKYP-Lemma
- 2.) Ziegler Nichols
- 3.) Chien Hrones Reswick

¹³ Weiter Informationen sind in [4] zu finden.

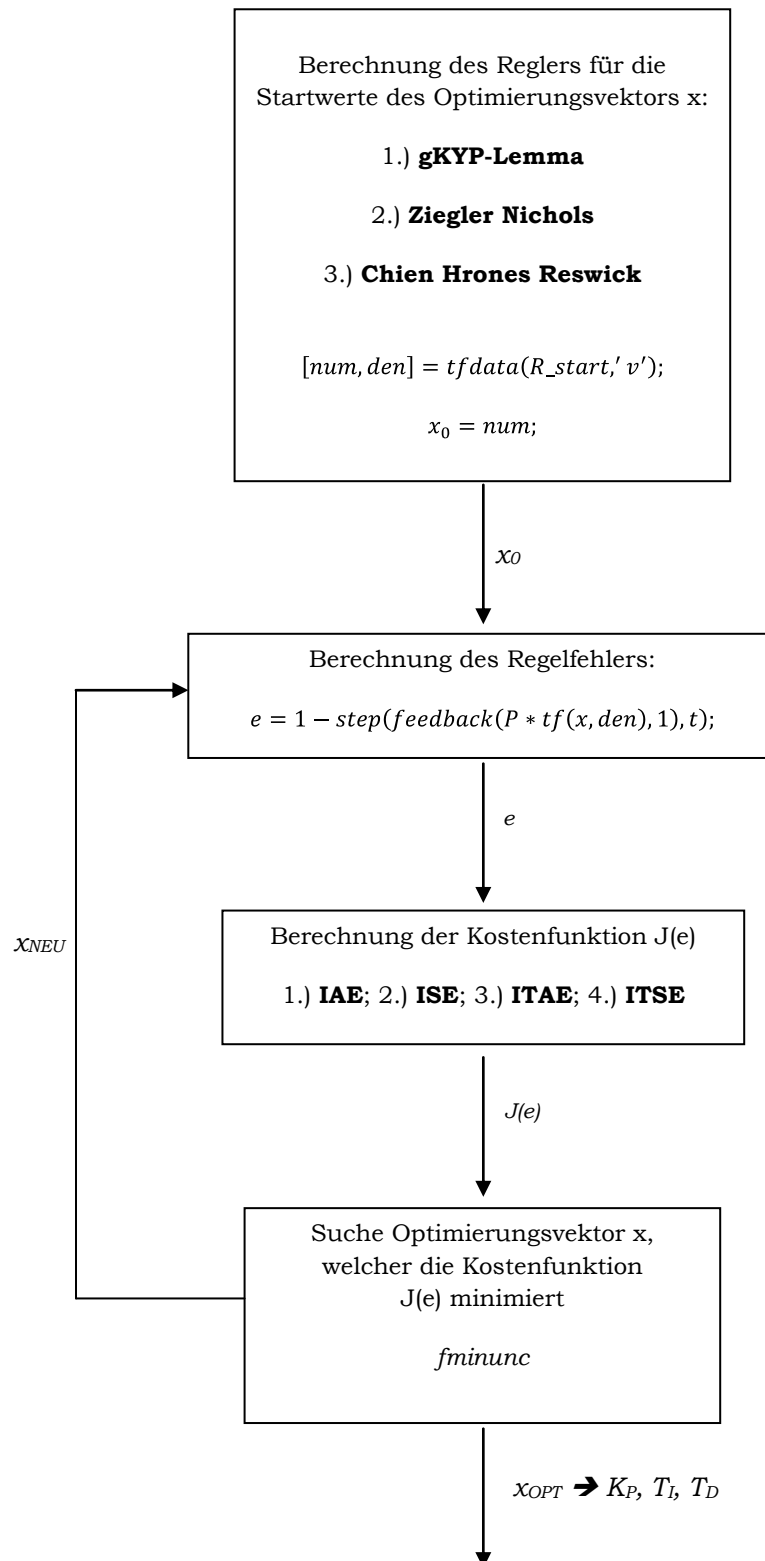


Abbildung 3.13: Ablauf Minimierung von Integralkriterien

Durch die Verwendung der MATLAB-Funktion „*fminunc*“ kann dieses Verfahren zu längeren Rechenzeiten führen. Weiters kann nicht für jede Regelstrecke eine Lösung garantiert werden, da dieses Verfahren sehr stark von der Wahl der Startwerte und des verwendeten Integralkriteriums abhängig ist.

Der Aufruf unter MATLAB erfolgt mit der Funktion:

```
function [Kp, Ti, Td, Tf, typ, method] = pioPID(P, w, typ_R, idx, x0)
```

Eingabe-Parameter:

- P: Verwendete Regelstrecke
- w: Frequenzvektor für das Bode-Diagramm
- typ_R: Auswahl des zu entwerfenden Reglers: P, PI, PID
- idx: 1.) IAE
 2.) ISE
 3.) ITAE
 4.) ITSE.
- x0: 1.) gKYP-Lemma
 2.) Ziegler Nichols
 3.) Chien Hrones Reswick

3.5.1 Beispiel zum Entwurf mit Minimierung von Integralkriterien

Methode:	Minimierung von Integralkriterien	
PID	K_P	2.1302
	T_I	2.5313
	T_D	0.6467
	T_f	0.0323
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 9: Reglerparameter Testbeispiel Minimierung von Integralkriterien

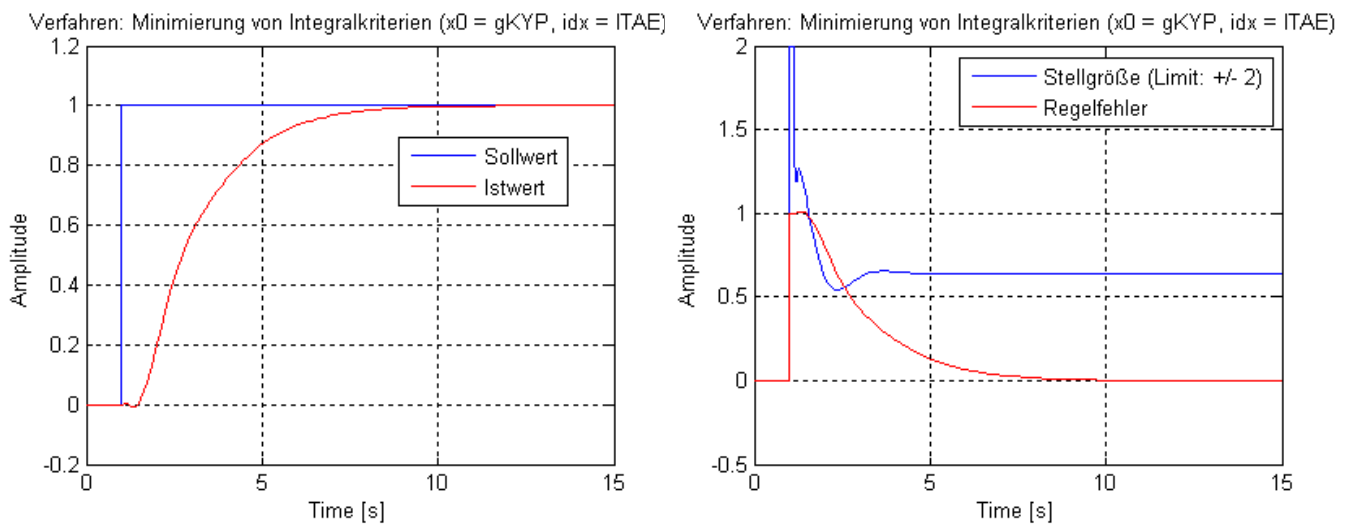


Abbildung 3.14: Simulationsergebnis mit Minimierung der Integralkriterien

Die Entwurfsmethode mit der Minimierung der Integralkriterien weist einen annähernd aperiodischen Einschwingvorgang auf. Es tritt kein Überschwingen auf und die Ausgangsgröße ist bereits nach etwa 12 Sekunden ausgeregelt. Der erhaltene Regler bietet, falls kein Überschwingen erwünscht ist, eine gute Alternative zu jenem aus dem gKYP-Verfahren.

3.6 Improved Performance

Dieses Verfahren basiert auf einer Approximation der Regelstrecke durch ein System zweiter Ordnung mit Totzeit und einer Strategie zur Platzierung der Pole des geschlossenen Regelkreises. Es beruht auf den Untersuchungen in [3].

Das Ziel dieses Entwurfsverfahren besteht darin, eine gute Performance für den Regelkreis, unabhängig von der Dynamik, der Totzeit, des Einschwingverhaltens und der Ordnung der Strecke, zu erhalten. Unter einer guten Performance wird, bei diesem Verfahren, eine schnelle Annäherung der Ausgangsgröße, ohne großes Überspringen, an den Sollwert des Regelkreises verstanden. Mit der Approximation der Regelstrecke¹⁴ wird versucht, den PID-Regler so zu entwerfen, dass Pole der Strecke gekürzt werden und ein einfacher Kreis, bestehend aus einem Integrator mit Totzeit und einem Verstärkungsfaktor, übrigbleibt.

Für den Entwurf wird ein PID-Regler der Form $R_{PID} = K_P + \frac{K_I}{s} + K_D s$ verwendet. Das Ziel des Verfahrens ist es, die Parameter K_P , K_I und K_D so zu bestimmen, dass der Regler eine hohe und gleichbleibende Performance aufweist.

Eine übliche Methode ist es, Strecken mit einer hohen Ordnung durch mehrere Systeme erster Ordnung mit Totzeit zu approximieren. Jedoch haben diese Modelle nur reelle Polstellen und können deshalb keine Resonanzstellen im Frequenzgang von oszillierenden Prozessen generieren. Deshalb verwendet dieses Verfahren ein Streckenmodell 2.Ordnung mit Totzeit und folgender Struktur

$$\tilde{P}(s) = \frac{e^{-s T_t}}{a s^2 + b s + c}$$

wobei die Parameter a , b , c und T_t die zu bestimmenden Unbekannten sind. Abhängig von den Parametern a , b und c weist die approximierte Strecke reelle oder komplexe Polstellen auf und somit kann diese Struktur auch zum Nachbilden von schwingungsfähigen Strecken verwendet werden.

Um die vier Unbekannten zu berechnen, werden vier Gleichungen benötigt, welche aus zwei Punkten der Übertragungsfunktion $G(s)$ gebildet werden können, sodass die Gleichungen $\tilde{P}(j\omega_c) = P(j\omega_c)$ und $\tilde{P}(j\omega_b) = P(j\omega_b)$ erfüllt sind.

Dabei werden die Frequenzen ω_c und ω_b so gewählt, dass

$$\arg(P(j\omega_c)) = -\pi$$

$$\arg(P(j\omega_b)) = -\pi/2$$

gilt.

¹⁴ Strecke 2. Ordnung mit Totzeit

Mit den Gleichungen

$$c - a \omega_c^2 + j b \omega_c = \frac{\cos(\omega_c T_t) - j \sin(\omega_c T_t)}{-|P(j\omega_c)|}$$

$$c - a \omega_b^2 + j b \omega_b = \frac{\cos(\omega_b T_t) - j \sin(\omega_b T_t)}{-|P(j\omega_b)|}$$

werden die Parameter a , b , c und T_t berechnet.

Dieses Verfahren zielt darauf ab, dass die ursprüngliche und approximierte Strecke im Bode-Diagramm hinreichend genau bei den Frequenzen ω_c und ω_b übereinstimmen. Dieses Ziel wird angestrebt, da dieser Frequenzbereich wichtig für den Reglerentwurf ist und somit die Approximation in diesem Bereich sehr gut übereinstimmen sollte. Die Lösungen der beiden zuvor gezeigten Gleichungen geben eine Berechnungsvorschrift für die vier unbekannten Parameter an.

$$\frac{\sin(\omega_c T_t)}{\cos(\omega_b T_t)} = \frac{(\omega_c |P(j\omega_c)|)}{(\omega_b |P(j\omega_b)|)} := \theta$$

nach Approximation mit:

$$\sin(x) = p x^2 + q x$$

$$\cos(x) = p x^2 + r x + 1$$

$$p = 8/\pi^2(1 - \sqrt{2})$$

$$q = \left(\frac{2}{\pi}\right)(2\sqrt{2} - 1)$$

$$r = \left(\frac{2}{\pi}\right)(2\sqrt{2} - 3)$$

$$\theta = (\omega_c |P(j\omega_c)|)/(\omega_b |P(j\omega_b)|)$$

erhält man die Gleichung für die unbekannten Parameter

$$p(\omega_c^2 - \theta \omega_b^2)T_t^2 + (q\omega_c - \theta r \omega_b)T_t - \theta = 0$$

$$a = \frac{1}{\omega_c^2 - \omega_b^2} \left[\frac{\sin(\omega_b T_t)}{|P(j\omega_b)|} + \frac{\cos(\omega_c T_t)}{|P(j\omega_c)|} \right]$$

$$b = \frac{\sin(\omega_c T_t)}{\omega_c |P(j\omega_c)|}$$

$$c = \frac{1}{\omega_c^2 - \omega_b^2} \left[\frac{\omega_c^2 \sin(\omega_b T_t)}{|P(j\omega_b)|} + \frac{\omega_b^2 \cos(\omega_c T_t)}{|P(j\omega_c)|} \right]$$

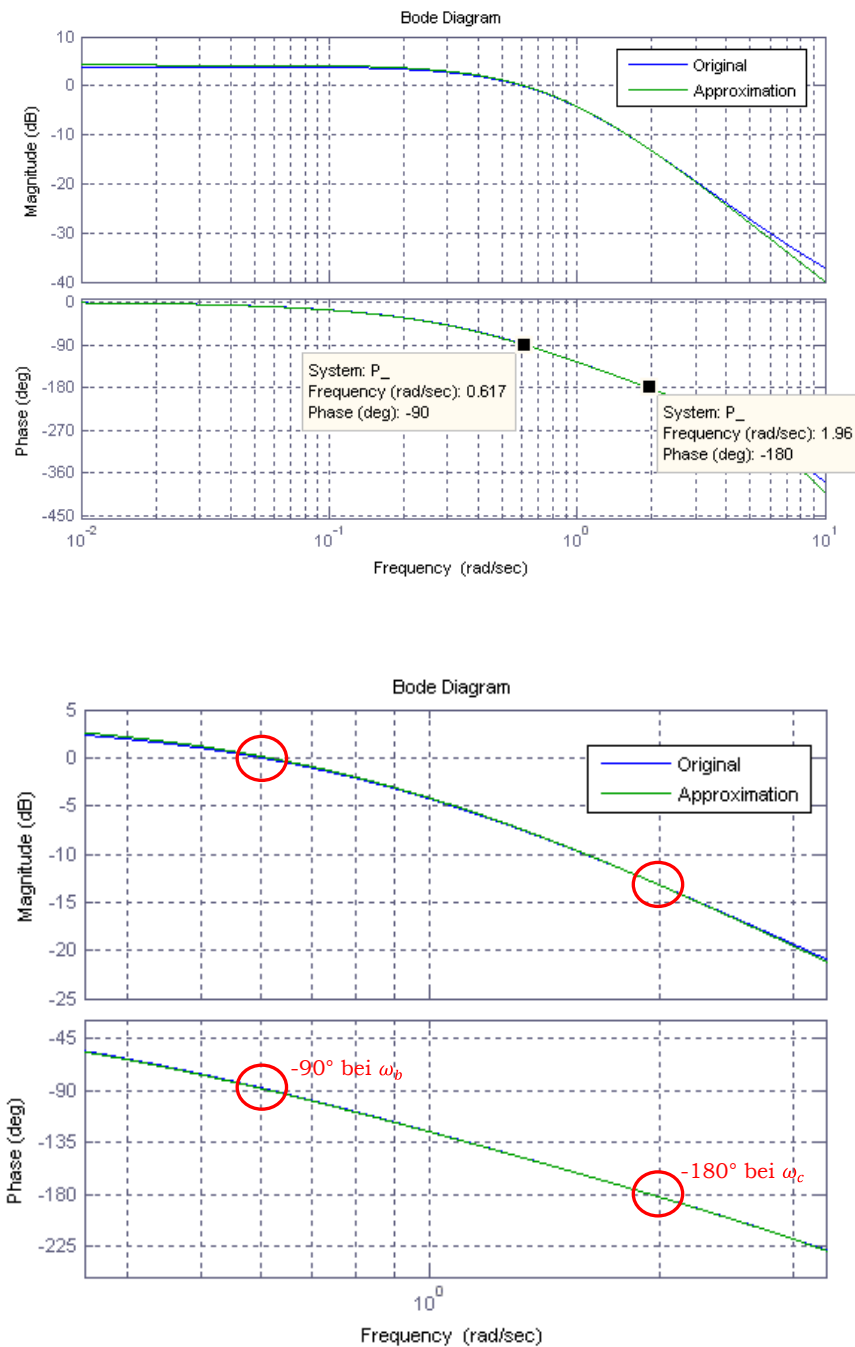


Abbildung 3.15: Vergleich der Bode-Diagramme der gegebenen und approximierten Streckenübertragungsfunktion

Wie bereits zuvor erwähnt, zielt dieses Verfahren auf eine Übereinstimmung des Frequenzganges der gegebenen und der approximierten Strecke bei den Punkten ω_c und ω_b ab. Wie aus der Abbildung 3.13 ersichtlich, liefert das Verfahren, für die Approximation der Regelstrecke durch ein System 2. Ordnung mit Totzeit, ein sehr gutes Ergebnis und kann für den Reglerentwurf verwendet werden.

Prozedur des PID-Reglerentwurfs:

Mit der erhaltenen Strecke kann nun der Entwurf des Reglers begonnen werden. Aus den folgenden Berechnungsvorschriften wird der Parameter k für einen Regler der Form

$$R(s) = \frac{k(a s^2 + b s + a)}{s}$$

berechnet. Wie bereits erwähnt, kürzt dieser Regler in weiterer Folge die Pole der approximierten Strecke, damit für den offenen Kreis $L(s)$ lediglich ein Verstärkungsfaktor mit einem Integrator und einer Totzeit übrigbleibt.

$$L(s) = R(s)\tilde{P}(s) = \frac{k(a s^2 + b s + a)}{s} \frac{e^{-s T_t}}{a s^2 + b s + c} = \frac{k e^{-s T_t}}{s}$$

Berechnungsvorschriften:

$$\frac{1}{\tau_0} = \begin{cases} \frac{c}{\sqrt{b^2 - 4ac}} & \text{für } b^2 - 4ac < 0 \\ 1 & \text{für } b^2 - 4ac > 0 \end{cases}$$

$$\zeta_0 = \begin{cases} \frac{b}{2\sqrt{ac}} & \text{für } b^2 - 4ac < 0 \\ 1 & \text{für } b^2 - 4ac > 0 \end{cases}$$

$$k = \begin{cases} \min \left\{ \frac{1}{\tau_0} e^{-\left(\frac{T_t}{\tau_0}\right)}, -s e^{s T_t} \Big|_{s=-\left(\frac{1}{T_t}\right)} = \frac{1}{e T_t} \right\} & \text{für } \zeta_0 \leq 0.7071 \text{ oder } 0.15 \leq \frac{T_t}{\tau_0} \leq 1 \\ \text{sonst } \frac{0.5}{T_t} & \end{cases}$$

Aus den vier erhaltenen Unbekannten können nun die Parameter für den traditionellen PID-Regler einfach bestimmt werden:

$$K_P = b k$$

$$K_I = c k \rightarrow T_i = K_P / K_I$$

$$K_D = a k \rightarrow T_D = K_D / K_P$$

Der Aufruf unter MATLAB erfolgt mit der Funktion:

`function [Kp, Ti, Td, Tf, typ, method] = impPID(P, w, typ_R)`

Eingabe-Parameter:

- **P:** Verwendete Regelstrecke
- **w:** Frequenzvektor für das Bode-Diagramm
- **typ_R:** Auswahl des zu entwerfenden Reglers: PID

3.6.1 Beispiel zum Entwurf mit Improved Performance

Methode:	Improved Performance	
PID	K_P	0.6899
	T_I	2.5571
	T_D	0.6201
	T_f	0.0310
Anti-Wind-up	k_{AW}	1
	+ Limit	2
	- Limit	-2

Tabelle 10: Reglerparameter Testbeispiel Improved Performance

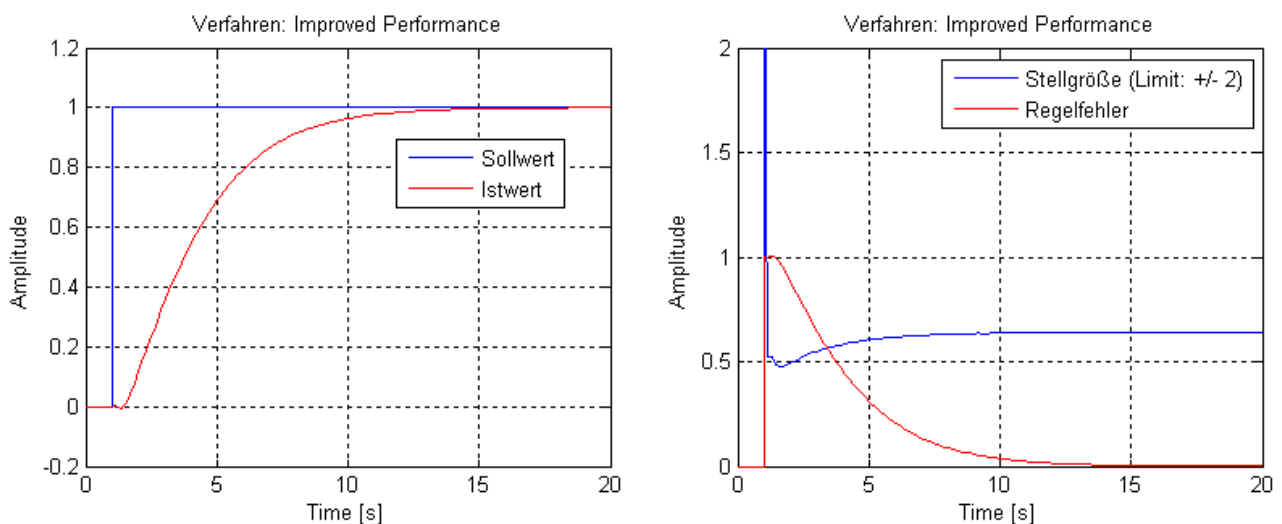


Abbildung 3.16: Simulationsergebnis mit Improved Performance

Der erhaltene Regler aus dem Verfahren Improved Performance, hat ein ähnliches Verhalten wie jener aus der Minimierung der Integralkriterien. Lediglich die Ausregelzeit des geschlossenen Regelkreises ist etwas länger.

4 Aufbau der Software

Im Zuge dieser Arbeit wurde laut Vorgabe der Firma TG GmbH eine Softwareumgebung mit dem Namen „PIDexpert“ entwickelt. Diese Software stellt dem Anwender Entwurfsverfahren zur Entwicklung und die Möglichkeit zur Simulation eines PID-Reglers für das erhaltene MISO-Modell aus der Identifikationsumgebung „Parameterident“ zur Verfügung.

4.1 Vorgaben der Firma TG GmbH

Folgende Punkte sollen bei der Entwicklung der Softwareumgebung beachtet werden:

- Programmiersprache MATLAB/Simulink

Das Programm ist als Teil einer Software zu entwickeln und muss kompatibel mit der Identifikationsumgebung „Parameterident“ sein. Gespeicherte Projekte sollen von beiden Programmen geöffnet und aktualisiert werden können. Die Programmierung erfolgt unter MATLAB 2007b.

- Einfache Bedienung durch den Anwender

Das Programm soll so einfach wie möglich gehalten werden und auch von nicht speziell geschultem Regelungstechnikpersonal verwendet werden können.

- Prozessdaten und Modelle

Jene Prozessdaten, welche zur Identifikation verwendet wurden, sollen als Eingangsgrößen für die Störgrößensimulation vorhanden sein, um eine möglichst praxisnahe Simulation zu gewährleisten. Die erhaltenen Regler sollen mit den verschiedenen mathematischen Modellen, welche durch unterschiedliche Identifikationsmethoden im Programm Parameterident ermittelt wurden, getestet werden können. Werden neue Daten bzw. Modelle in Parameterident hinzugefügt bzw. erstellt, so sollen diese auch im Projekt des Programms PIDexpert hinzugefügt werden können.

- Bereitstellen einer Programmdokumentation

Eine ausreichende Dokumentation über die Benützung des Programms ist anzufertigen.

- Regelstrukturen und Abtastzeiten

Beim Entwurfsvorgang, soll die Struktur des Reglers (P, PI, PID, PD), soweit möglich, von vornherein festlegbar sein. Verschiedene PID-Strukturen (PID, I-PD, PI-D) sollen simulierbar sein.

Der Regler soll zeitkontinuierlich und zeitdiskret, mit festgelegten Abtastzeiten, aufgebaut werden können. Die Einstellung der Anti-Wind-up Maßnahme für Regler mit Integral-Teil soll übersichtlich und bei Bedarf konfigurierbar sein.

- Anpassen der Reglerparameter

Werden bei einem entworfenen Regler einzelne Parameter verändert, so soll diese Änderung nach erneuter Simulation sofort dem Benutzer in Form eines Plots ersichtlich und wieder speicherbar sein.

- Reglerverwaltung

Nicht jeder Reglerentwurf führt zu einer zufriedenstellenden Lösung für die Anforderungen des Benutzers. Deshalb soll die Möglichkeit der Speicherung einzelner Regler bestehen, um eine Auswahl des endgültigen Reglers für das gewählte mathematische Modell treffen zu können.

- Visualisierung des Projekts

Bei der Visualisierung der Simulationsdaten soll die Möglichkeit bestehen, zwei fertig entwickelte Regler anhand ihres Stellgrößenverlaufes, Soll- und Istwertverlaufs zu vergleichen, um eine erste schnelle Beurteilung zu ermöglichen.

- Exportieren des Projekts

Das Programm soll die Möglichkeit beinhalten, fertige Projekte in den MATLAB Workspace zu exportieren, um weitere manuelle Simulationen durchführen zu können. In weiterer Folge sollen die wichtigsten Informationen des Projekts in ein druckbares Format (html, pdf) exportiert werden können, um einen Überblick über den Entwurf, auch ohne MATLAB, zu bekommen. Dieser Export sollte die gesamte Reglerkonfiguration, die Störgrößenkonfiguration und die Plots der einzelnen Simulationen beinhalten. Weiters soll die Möglichkeit vorhanden sein, jeden Regler einzeln zu exportieren, um eine bessere Übersicht bei der Inbetriebnahme in der realen Anlage zu gewährleisten.

4.2 Struktur der Entwicklungsumgebung

Mit den Vorgaben der Firma TG GmbH und dem Entwurfsablauf aus Kapitel 1, wurde eine Struktur für die Programmierung der Entwicklungsumgebung PIDexpert erstellt, welcher sich in vier Abschnitte unterteilen lässt.

Der erste Abschnitt beschäftigt sich mit der Auswahl der Regelstrecke. Danach wird ein Entwurfsverfahren ausgewählt, konfiguriert und der Vorgang zur Ermittlung des Reglers gestartet. Im dritten Abschnitt kann der erhaltene Regler simuliert werden und an das gewünschte Verhalten angepasst werden. Zum Abschluss kann der Regler mit Störgrößen simuliert werden, um eine praxisnahe Beurteilung des Verhaltens des Regelkreises zu ermöglichen.

4.2.1 Auswahl der Regelstrecke

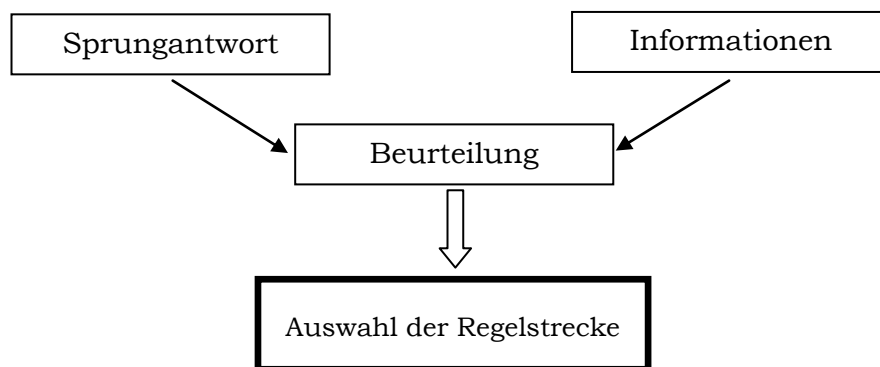


Abbildung 4.1: Struktur bei Auswahl der Regelstrecke

Dieser Abschnitt enthält alle Funktionen, die zur Auswahl einer Regelstrecke für den Entwurf benötigt werden. Es stehen alle Übertragungsfunktionen des MISO-Systems aus der Identifikationssoftware Parameterident zur Verfügung. Die Namen der einzelnen Strecken des Modells werden in einem List-Fenster zusammengefasst, um einen ersten Überblick über das gegebene MISO-Modell zu erhalten. Durch Auswahl einer vorhandenen Strecke können verschiedene Informationen, sowie die Sprungantwort der Strecke, abgerufen werden, welche als Hilfestellung zur Auswahl eines geeigneten Entwurfsverfahrens anzusehen sind.

Informationen zur Strecke:

- Eingang der Strecke
- Ausgang des MISO-Modells
- z-Übertragungsfunktion der Strecke mit der jeweiligen Abtastzeit
- Name des MISO-Modells
- Zusatzinformationen der Identifikation aus Parameterident

4.2.2 Auswahl und Konfiguration des Entwurfsverfahrens

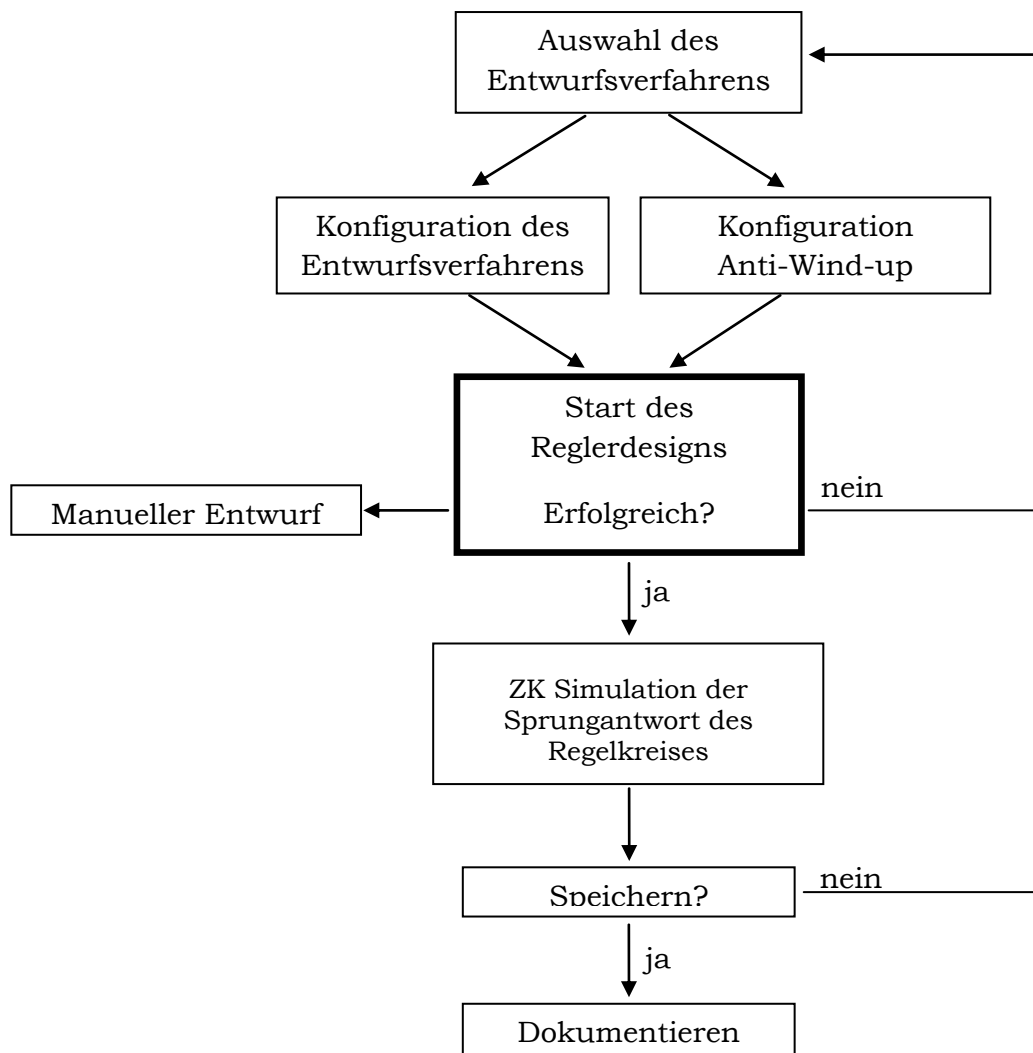


Abbildung 4.2: Ablauf bei Auswahl und Konfiguration des Entwurfsverfahrens

Dieser Abschnitt beschäftigt sich mit der Verwaltung der vorhandenen Entwurfsverfahren und deren Konfiguration. Für jedes Verfahren werden eigenständige Einstellungsfenster programmiert, mit welchen man die gewünschte Reglerstruktur, sowie die einzelnen spezifischen Parameter der Verfahren einstellen kann. Weiters besteht die Möglichkeit, bei jedem Einstellungsfenster die Stellgrößenbeschränkung und den Verstärkungsfaktor für die Rückkopplung der „Anti-Wind-up“ Maßnahme zu konfigurieren. Diese Parameter können natürlich nach einem erfolgreichen Entwurf des Reglers angepasst werden. Nach einer vollständigen Konfiguration des gewünschten Entwurfsverfahrens wird die Berechnung freigeschaltet und der Entwurf kann gestartet werden.

Wenn das Entwurfsverfahren erfolgreich beendet wurde, wird mit den erhaltenen Reglerparametern eine Simulation der Sprungantwort mit einem zeitkontinuierlichen PID-Regler durchgeführt. Der Anwender soll danach

selbstständig entscheiden können, ob der erhaltene Regler seinen Kriterien genügt und die Parameter abgespeichert werden sollen. Den Namen des Reglers kann der Anwender selbst wählen und gegebenenfalls an die Entwurfsmethode anpassen. Falls kein zufriedenstellendes Ergebnis mit der gewählten Konfiguration des Entwurfsverfahrens erreicht wurde, kann der Anwender diese anpassen oder ein anderes Verfahren wählen. Für den Fall, dass kein automatischer Entwurf zu einer Lösung geführt hat, kann auch durch eine manuelle Variation der Parameter ein Regler gesucht werden. Um immer wieder Informationen über die Parameter und das verwendete Entwurfsverfahren des gespeicherten Reglers zu erhalten, wird ein eigenes Info-Fenster erstellt, bei dem auch die Möglichkeit der Namensänderung des Reglers besteht.

Informationen über den gespeicherten Regler:

- zeitkontinuierlich/zeitdiskret
- Reglertyp: P, PI, PID, PD
- Übertragungsfunktion des Reglers
- Name des MISO-Modells und der gewählten Strecke
- Reglerparameter: K_P , T_I , T_D , T_F
- Parameter der Anti-Wind-up Konfiguration
- Verwendete Regelstruktur: PID, PI-D, I-PD
- Verwendete Entwurfsmethode

Falls bereits ein Regler entworfen wurde, kann dieser mit der gleichen bzw. einer alternativen Regelstrecke eines anderen Modells getestet werden, um zu überprüfen ob der erhaltene Regler empfindlich auf Parameterschwankungen reagiert bzw. auch für eine andere Regelstrecke funktioniert.

4.2.3 Simulation des erhaltenen Reglers

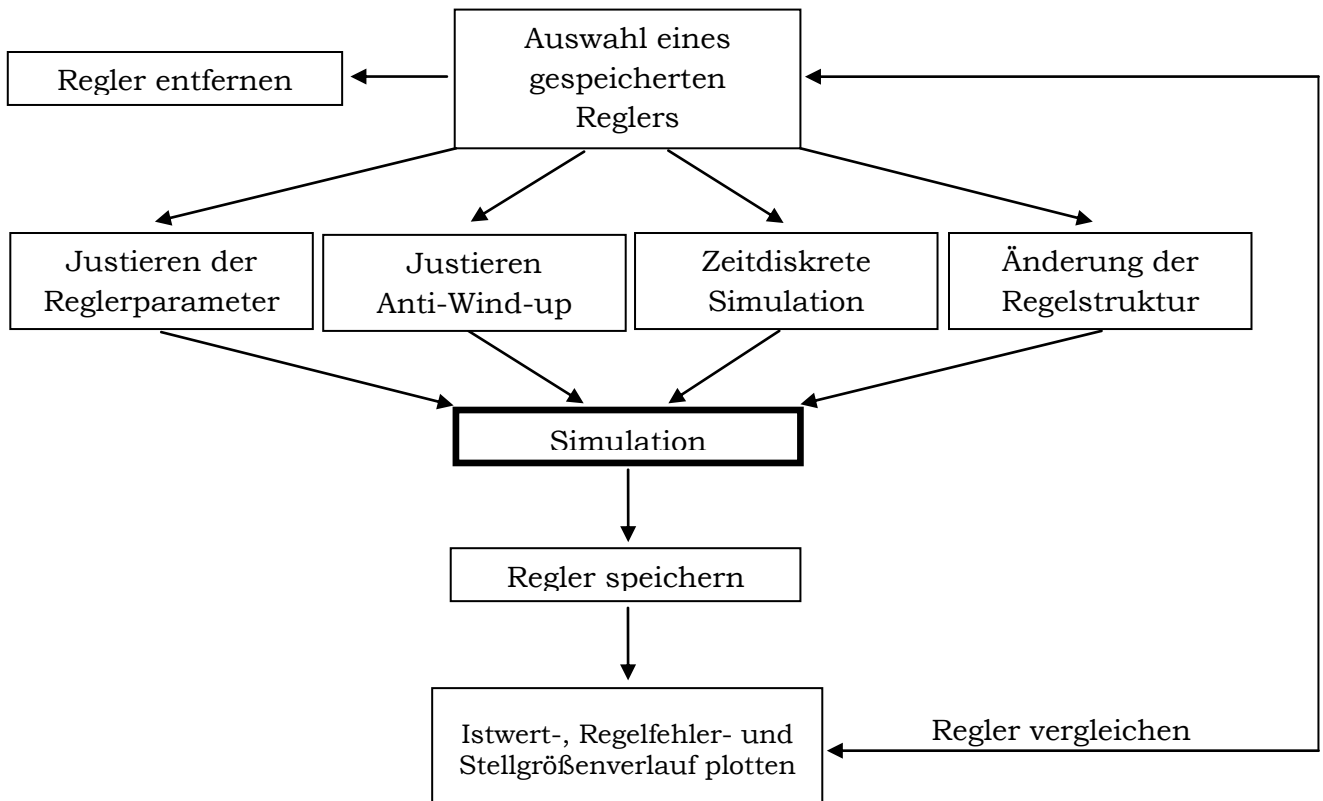


Abbildung 4.3: Struktur bei der Simulation des Reglers

Bei der Simulation des Reglers ist eine genauere Beurteilung des durch den Entwurf erhaltenen Reglers möglich. Falls eine Nachjustierung der Reglerparameter bzw. der Anti-Wind-up Parameter durch den Anwender gewünscht wird, soll das Programm dies ermöglichen und bei Änderungen sofort eine Simulation durchführen. Das Ergebnis der Simulation wird dem Anwender in Form des Istwertverlaufes im Regelkreis angezeigt und es wird die Möglichkeit geboten, den neu erhaltenen Regler zu speichern.

In weiterer Folge ist es möglich, die erhaltenen Regler aus dem automatisierten Entwurf miteinander zu vergleichen, indem man sich die einzelnen Plots des Istwertverlaufes, des Regelfehlers und der Stellgröße anzeigen lässt. Außerdem kann man eine Änderung der Regelstruktur von PID auf PI-D bzw. I-PD durchführen, um den Einfluss des D-Anteils beim Stellgrößenverlauf besser beurteilen zu können. Eine Simulation mit einem zeitdiskreten Aufbau steht hierbei auch zur Verfügung, um eine Beurteilung der erhaltenen Reglerparameter bei bestimmten Abtastzeiten zu erhalten und diese an die Praxis anzupassen.

All diese möglichen Änderungen an den bereits gespeicherten Reglern können in einem neuen Reglerstruct abgespeichert und nicht mehr benötigte gelöscht werden.

4.2.4 Simulation mit Störgrößen

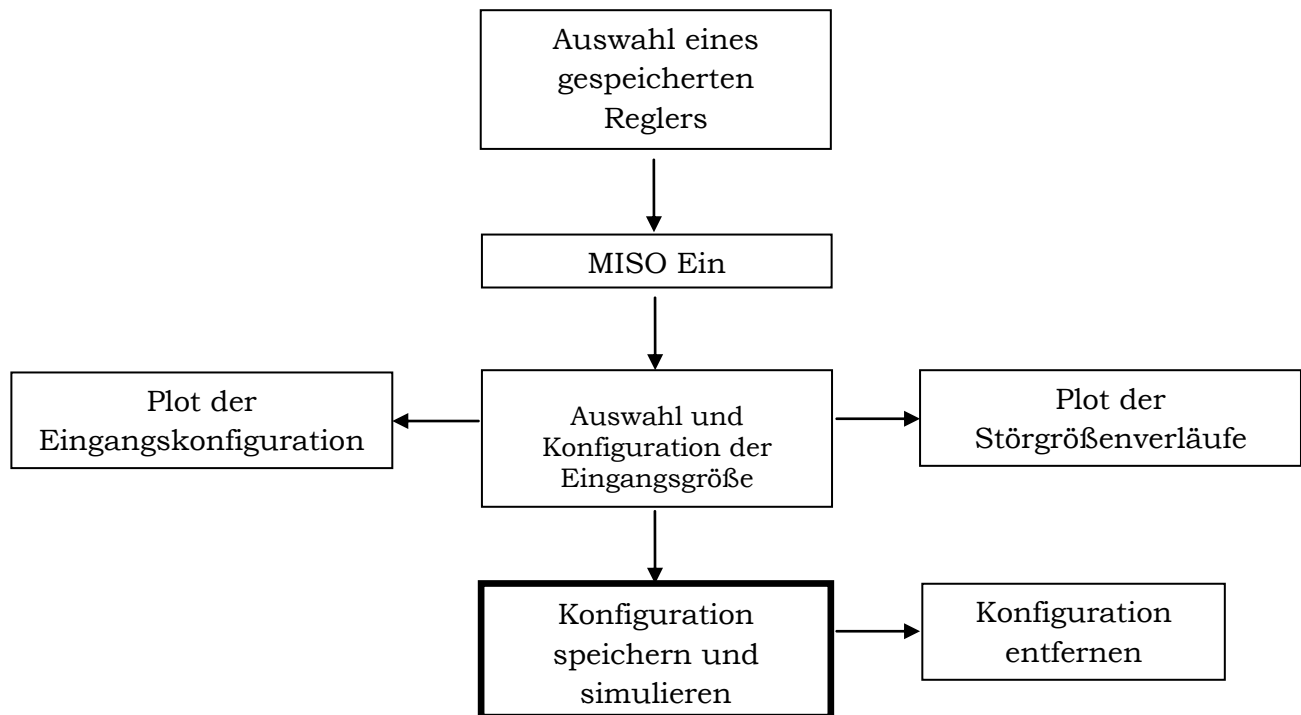


Abbildung 4.4: Struktur bei Simulationen mit Störgrößen

Alle Varianten für die Störgrößensimulation werden in diesem Abschnitt zusammengefasst. Für jede Störgröße wird hier die Möglichkeit bestehen, eine Eingangsgröße zu wählen und zu konfigurieren. Für jeden gespeicherten Regler können hier mehrere Konfigurationen der Störgrößen gespeichert und verwaltet werden. Die Verläufe der einzelnen Störgrößen können auf Knopfdruck angezeigt und verglichen werden.

Eingangsgrößen der Störgrößen:

- Sprungfunktion
- Rampenfunktion
- Prozessdaten der Identifikation
- Treppenfunktion mit Zufallsamplitude

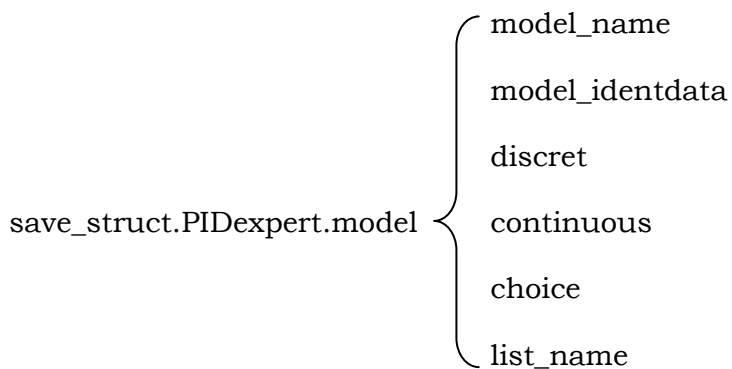
In weiterer Folge kann man auch für jeden einzelnen gespeicherten Regler oder für das gesamte Projekt eine Zusammenfassung als HTML-File exportieren. Dies bietet die Möglichkeit, auch ohne MATLAB einen Überblick über die Ergebnisse zu erhalten. Ein Export in den Workspace von MATLAB ist auch implementiert, um die erhaltenen Regler auch für andere Simulationen verwenden zu können.

4.3 Datenstruktur

Die Modell- und Reglerdaten werden als Anwendungsdaten im GUI der Software PIDexpert abgespeichert. Weiters wurden die Identifikationsdaten für die Simulation aus Parameterident mit der gleichen Struktur¹⁵ übernommen. Der Zugriff auf die verschiedenen Daten erfolgt bei der GUI-Programmierung unter MATLAB mit Hilfe sogenannter „handles“. Nähere Information zur Programmierung von GUI's und zur Verwendung von handles kann man der MATLAB-Hilfe entnehmen.

4.3.1 Modelldaten

Die Daten der einzelnen aus Parameterident erhaltenen MISO-Modelle werden in folgender Struktur verwaltet und abgespeichert.

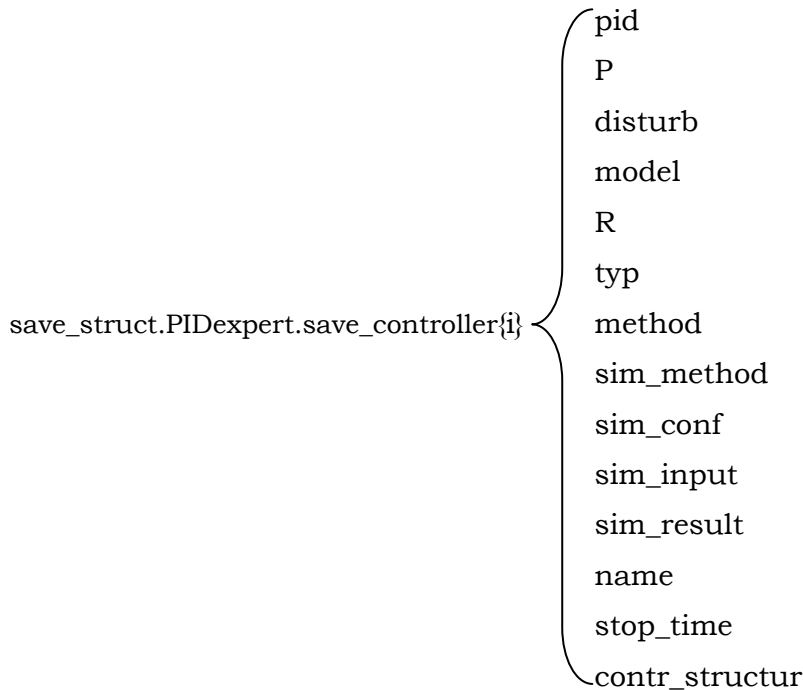


model_name:	Die Namen aller zur Verfügung stehender MISO-Modelle
model_idntdata:	Name des Datensatzes zur Identifikation des Modells
discret:	z-Übertragungsfunktionen des MISO-Modells
continuous:	Mittels „c2d“ erhaltene s-Übertragungsfunktionen mit Totzeit und mit einer Pade-Approximation
choice:	Gewähltes MISO-Modell
list_name:	Namen der einzelnen Strecken des MISO-Modells

¹⁵ Näheres siehe [5]

4.3.2 Reglerdaten

Für die Verwaltung der erhaltenen Reglerparameter wurde in PIDexpert eine spezielle Struktur entwickelt. Die entworfenen Regler werden in Cell-Arrays gespeichert und beinhalten alle wichtigen Informationen für die weitere Simulation.



pid:	Enthält alle Parameter des entworfenen Reglers:
	<ul style="list-style-type: none"> • K_P Verstärkung • T_I Integrierzeit (Nachstellzeit T_N) • T_D Differenzierzeit (Vorhaltezeit T_V) • T_f Zeitkonstante des Realisierungsterms • T_S Abtastzeit bei zeitdiskretem Aufbau • k_{AW} Verstärkung der Rückkopplung der Anti-Wind-up Maßnahme • upperlimit Stellgrößenbeschränkung obere Schranke • lowerlimit Stellgrößenbeschränkung untere Schranke • filter Differenzierer mit Realisierungsterm
P:	Beinhaltet die zeitkontinuierliche und zeitdiskrete Übertragungsfunktion der gewählten Regelstrecke für den Entwurf.
disturb:	Strecken des MISO-Systems (ohne Regelstrecke), welche als Störgrößen auf den Regelkreis aufgeschaltet werden können.
model:	Name des gewählten MISO-Modells, welches aus Parameterident zur Verfügung steht.

R: Übertragungsfunktion des Reglers. Je nach Simulationsmethode, zeitkontinuierlich bzw. zeitdiskret aufgebaut

typ: Reglerart des entworfenen Reglers: P, PI, PID, PD

method: Name des verwendeten Verfahrens zum Entwurf des Reglers

sim_method: Zeitdiskrete oder zeitkontinuierliche Simulationsmethode

sim_conf: Konfiguration der Eingangsgrößen für die Störgrößensimulation

sim_input: Verläufe der Eingangsgrößen bei der Störgrößensimulation

sim_result: Verläufe der Simulationsergebnisse mit und ohne Störgrößen

name: Name des entworfenen Reglers

stop_time: Simulationszeit des Regelkreises

contr_structur: Struktur des Reglers. PID, PI-D und I-PD

Diese Datenstruktur wird beim Export in den Workspace übergeben. Auf die einzelnen Regler kann mittels

controller{i} i = 1... Anzahl der gespeicherten Regler

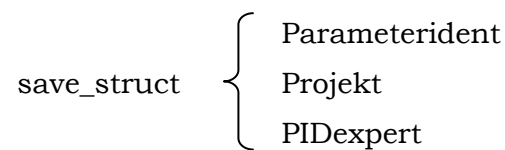
zugegriffen werden. Bei der Generierung des Reports werden aus den vorhandenen Daten die wichtigsten Informationen für die Inbetriebnahme herausgenommen und die Simulationsergebnisse geplottet.

Die Konstruktion der zuvor erläuterten Datenstruktur wird von folgender Funktion übernommen:

```
function controller = reglerstruct(Kp, Ti, Td, Tf, Ts, P, disturb, P_discret,
                                   disturb_discret, kaw, lim_u, lim_l, stop_time,
                                   typ, method, sim_method, model, sim_conf,
                                   sim_input, sim_result, name);
```

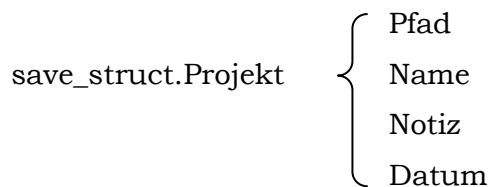

4.3.3 Kompatibilität vom Parameterident und PIDexpert

Ein weiterer wichtiger Punkt bei den Vorgaben der Firma TG GmbH war die Kompatibilität zwischen der Identifikationsumgebung Parameterident und der zu entwerfenden Entwicklungsumgebung PIDexpert. Um diese Bedingung zu erfüllen, wurde eine gemeinsame Struktur zur Speicherung der jeweiligen Projektdaten entworfen. Die Projektdaten werden über ein „mat-File“ den einzelnen Programmen übergeben. Das verwendete „mat-File“ gliedert sich in drei Datenverbunde, welche auch „structs“ genannt werden. Die Elemente der Projektdaten besitzen folgende Struktur:



In der Instanz Parameterident befinden sich die gesamten Projektdaten der Identifikationssoftware Parameterident. Für eine genaue Erläuterung der einzelnen Daten wird auf [5] verwiesen.

Projekt beinhaltet verschiedene Informationen über das „mat-File“.



Pfad: Enthält den Pfad der „mat-File“ bei der letzten Speicherung des Projekts.

Name: Name des „mat-Files“

Notiz: Verschiedene Informationen über die enthaltenen Projekte von Parameterident und PIDexpert

Datum: Enthält Datum und Uhrzeit der letzten Speicherung des Projekts

Die Instanz PIDexpert beinhaltet die schon zuvor erläuterten Regler- und Modelldaten, sowie die Prozessdaten für die Simulation aus Parameterident und ein Arbeitsverzeichnis in der sich Einstellungen für die Benutzeroberfläche der Software befinden.

Die Kommunikation zwischen den beiden Softwarepaketen übernimmt die Entwicklungsumgebung PIDexpert. Werden in Parameterident neue Modelle identifiziert bzw. neue Prozessdaten dem Projekt hinzugefügt, überprüft PIDexpert diese beim Laden des Projekts und dem Anwender wird mitgeteilt, dass neue Modelle bzw. Prozessdaten zur Verfügung stehen. Der Anwender kann dann selbständig entscheiden, ob er die neuen Daten dem Projekt in PIDexpert hinzufügen bzw. alte Daten entfernen will.

Die Bearbeitung der Projektdaten von PIDexpert erfolgt über ein eigens dafür programmiertes Fenster:

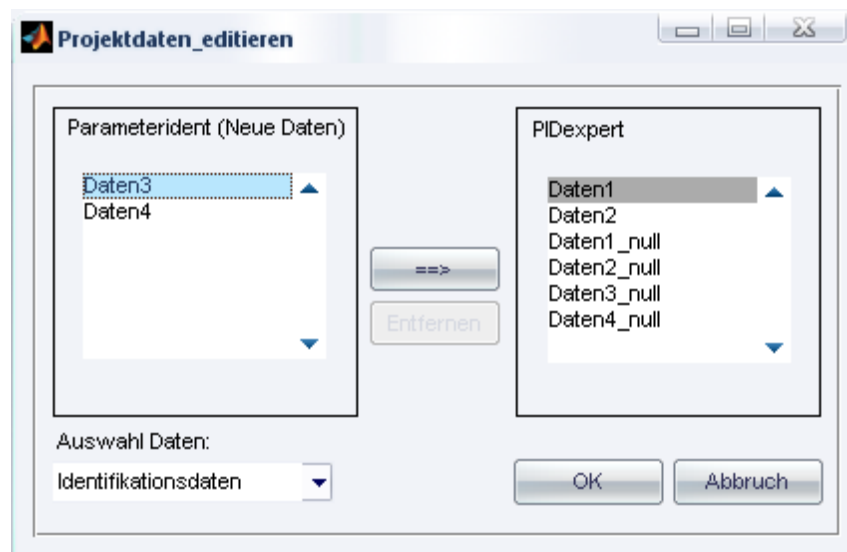


Abbildung 4.5: Fenster zur Bearbeitung der Projektdaten

In der linken Listbox sind die neuen Daten für den Anwender ersichtlich. Er kann falls erwünscht diese dem vorhandenen Datensatz hinzufügen bzw. nicht mehr benötigte Daten aus dem aktuellen Projekt im rechten Fenster entfernen. Mit der Listbox „Auswahl Daten“ kann der Anwender zwischen Prozessdaten und vorhandenen Modellen umschalten. Werden Daten zum Entfernen ausgewählt, welche bereits im Projekt zum Reglerentwurf verwendet werden, erhält der Anwender eine Fehlermeldung.



Abbildung 4.6: Fehlermeldung bei Bearbeitung der Projektdaten

Auch die letzten Daten in einem Datensatz können nicht gelöscht werden um ein weiterarbeiten mit PIDexpert zu gewährleisten.

4.4 Aufbau in MATLAB/Simulink

Für die Simulation des Regelkreises wurden zwei Simulink-Modelle erstellt. Ein Modell wird für die zeitkontinuierliche Simulation und das andere wird für zeitdiskrete Simulation verwendet. Für den zu verwendenden Simulationsalgorithmus wird für die zeitkontinuierliche Simulation der Algorithmus „ode45“¹⁶ und für die zeitdiskrete Simulation der Algorithmus „ode3“¹⁷ verwendet.

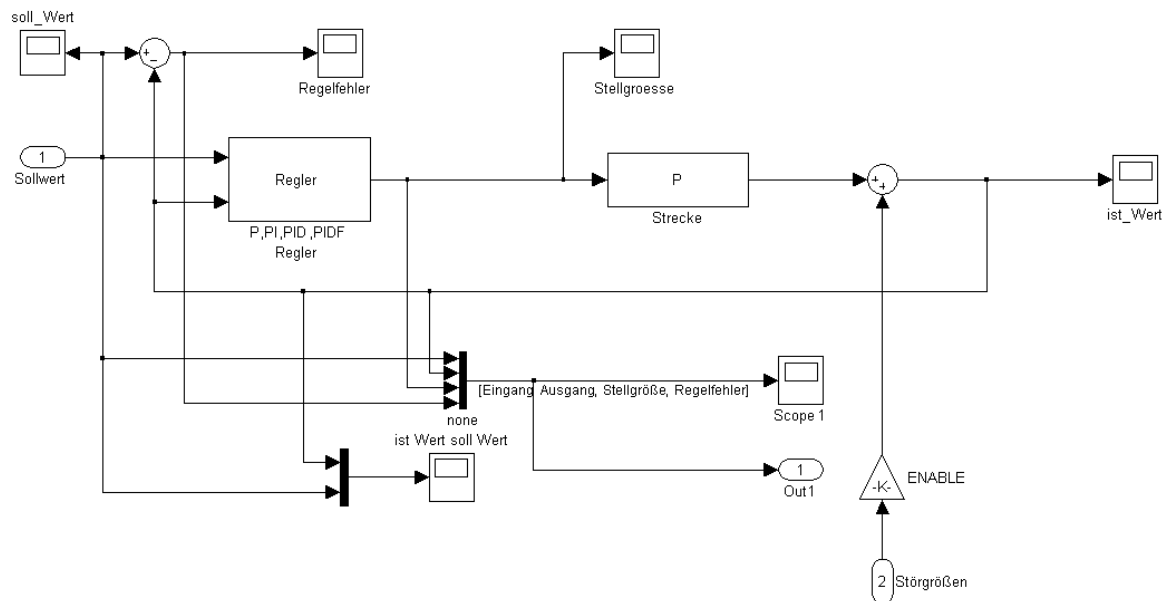


Abbildung 4.7: Koppelplan des Regelkreises unter Simulink (ZK)

Der Sollwert wird in einem m-File anhand der Konfiguration des Anwenders generiert und dem Simulink-Modell übergeben. Die Störgrößen werden im m-File mit ihren vom Anwender konfigurierten Eingangsgrößen simuliert und zur Ausgangsgröße des Regelkreises addiert.

Die Störgrößensimulation wurde unter MATLAB mittels der Funktion „lsim“ durchgeführt:

```
y = lsim(disturb_system, u, t);
```

disturb_system: Die ausgewählten Störgrößen-Übertragungsfunktionen

u: Vektor mit den erzeugten Eingangsgrößen

t: Zeitvektor für die Simulation

Die Verläufe des Sollwerts, des Istwerts, des Regelfehlers und der Stellgröße werden dem Workspace von MATLAB übergeben und können dort gespeichert und weiterverarbeitet werden.

¹⁶ Runge-Kutta 4. Ordnung

¹⁷ Bogacki-Shampine

4.4.1 PID-Regler

Die Kommunikation zwischen dem GUI und dem Regler im Simulink-Modell erfolgt über ein „Struct“ mit allen notwendigen Parametern zur Simulation. Der Regler wurde als maskiertes Submodell aufgebaut und kann als eigener Bibliotheks-Baustein betrachtet werden. Somit wird eine weitere Verwendung des PID-Reglers für Simulationen bei anderen Aufgabenstellungen erleichtert.

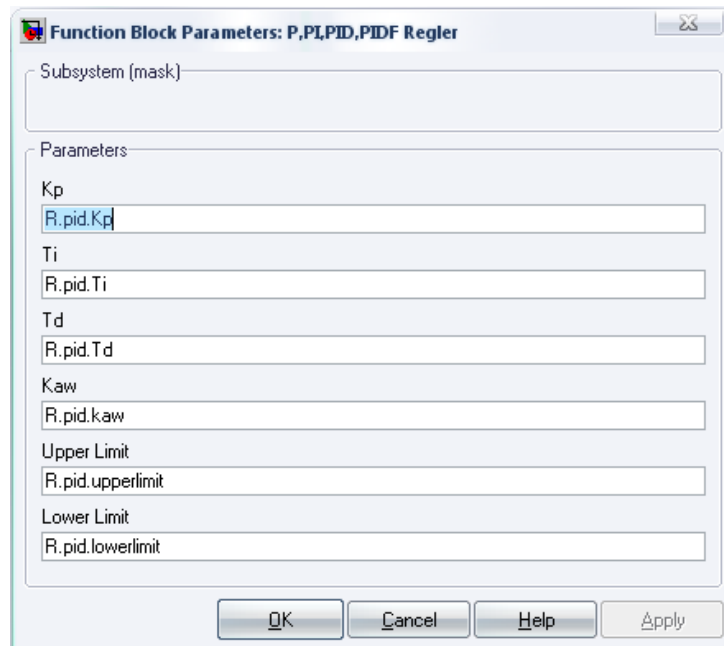


Abbildung 4.8: Maske des PID-Reglers

Wie bereits erwähnt, ist der Regler als ein Subsystem aufgebaut und beinhaltet die verschiedenen Regelstrukturen¹⁸, welche der Anwender für die Simulation auswählen kann. Je nach Konfiguration der Simulationsparameter wird zwischen den einzelnen Regelstrukturen umgeschaltet bzw. das jeweilige Simulink-Modell für die Simulation aufgerufen.

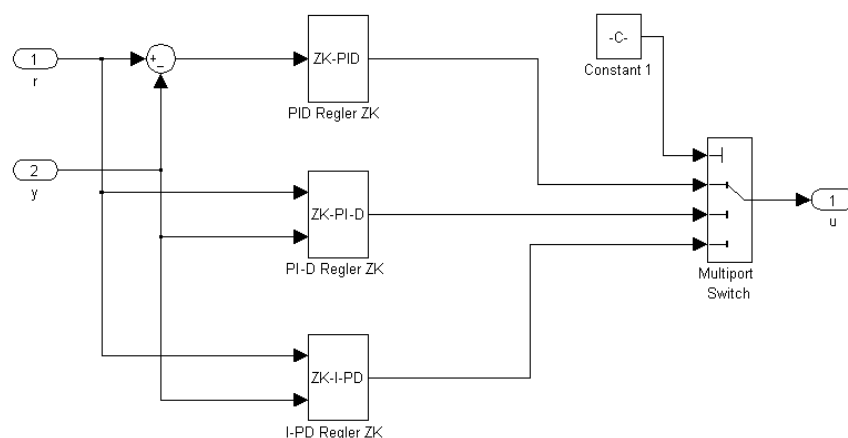


Abbildung 4.9: Koppelplan für die Umschaltung der Regelstrukturen

¹⁸ PID, PI-D und I-PD

Der Aufbau eines Reglers im Subsystem wird anhand eines zeitkontinuierlichen PID-Reglers erläutert. Die weiteren Regelstrukturen, wie PI-D und I-PD, sind wie in Kapitel 2.3 beschrieben aufgebaut und werden deshalb in weiterer Folge nicht näher betrachtet.

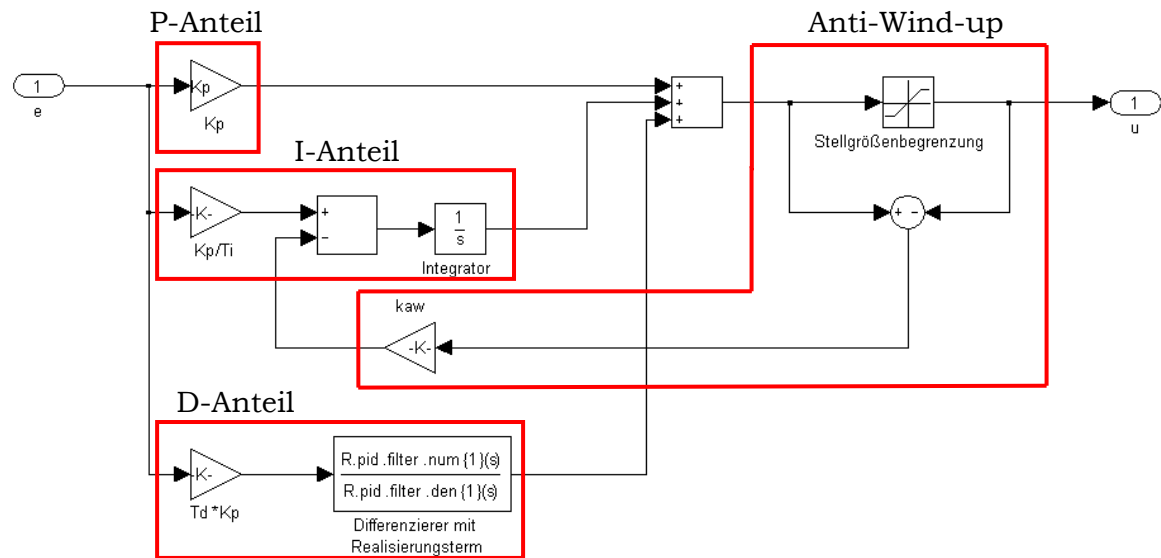


Abbildung 4.10: Koppelplan des zeitkontinuierlichen PID-Reglers

Der in der zuvor gezeigten Abbildung ersichtliche Regler besteht aus einem PID-Regler mit einer Stellgrößenbeschränkung. Wie bereits in Kapitel 2.4 erwähnt, ist bei einem Regler mit einem Integral-Teil und einer Begrenzung der Stellgröße eine Anti-Wind-up Maßnahme vorzunehmen. Der D-Anteil des Reglers wurde nicht mit dem d/dt-Block aus Simulink mit einem Realisierungsterm sondern als Filter aufgebaut um bessere numerische Ergebnisse zu erhalten.

Beim zeitdiskreten Aufbau ist für den D-Anteil kein Realisierungsterm für die praktische Implementierung notwendig, jedoch wurde in dieser Arbeit der zeitkontinuierliche Realisierungsterm in einen zeitdiskreten konvertiert um bei hohen Frequenzen als Tiefpass zu wirken und somit bei kleinen Abtastzeiten ein kritisches Verhalten zu vermeiden. Der zeitkontinuierliche und zeitdiskrete Aufbau des PID-Reglers unterscheidet sich lediglich beim I-Teil, D-Teil und der Einstellung der Abtastzeit.

$$I - TEIL: \quad \frac{1}{s} \rightarrow T_s \frac{z^{-1}}{1 - z^{-1}} \quad \text{"Forward Euler"}$$

$$D - TEIL: \quad s \rightarrow \frac{1}{T_s} \frac{1 - z^{-1}}{1}$$

5 PIDexpert

Dieses Kapitel beschäftigt sich mit der graphischen Benutzeroberfläche der Entwicklungsumgebung PIDexpert. Es wurde versucht, die Bedienung der Software so einfach und intuitiv wie möglich zu gestalten, um auch eine Benutzung durch nicht speziell geschultes Personal zu ermöglichen. Die Benutzeroberfläche lässt sich in vier Abschnitte unterteilen, welche in der Abfolge des Entwicklungsvorgangs gegliedert sind.

- 1.) Auswahl der Regelstrecke
- 2.) Auswahl und Konfiguration des Entwurfsverfahrens
- 3.) Simulation des erhaltenen Reglers
- 4.) Simulation mit Störgrößen

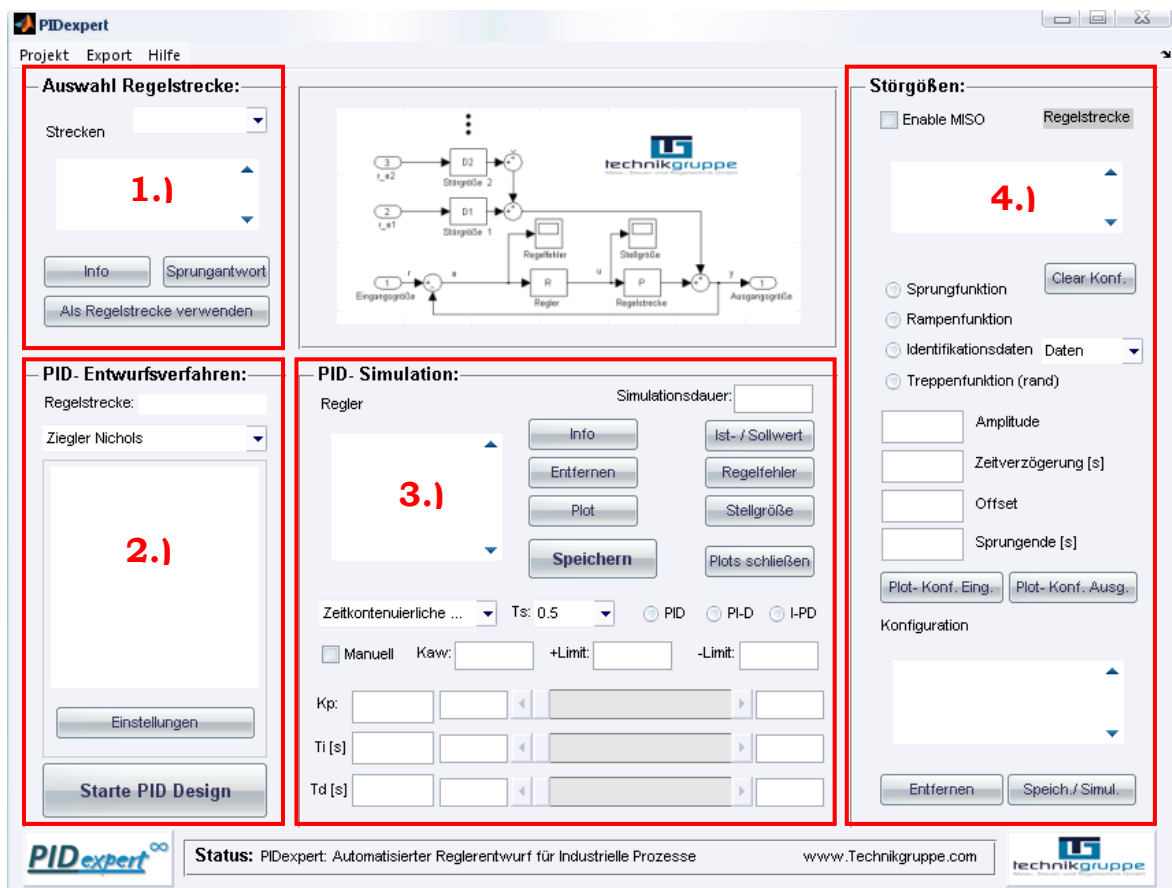


Abbildung 5.1: Benutzeroberfläche PIDexpert

Das GUI (Graphical User Interface) wurde für MATLAB/Simulink entwickelt und ist ab der Version 2007a lauffähig. Je nach Darstellungseinstellung des Betriebssystems, kann die Optik der Benutzeroberfläche etwas variieren. Die hier gezeigten Screenshots des Programms wurden auf einem Windows-Vista-Rechner generiert.

5.1 Erste Schritte

5.1.1 Installation der Toolbox

Zunächst ist für die Benutzung von PIDexpert die Installation der Toolbox für die Entwicklung eines Reglers mit dem gKYP-Lemma notwendig. Hierfür wurde bereits eine „Installation Toolbox.bat“-Datei geschrieben, um den Vorgang zu erleichtern. Sie befindet sich im Ordner „gkypsdp“ und muss auch aus diesem Verzeichnis ausgeführt werden. Alternativ kann die Toolbox auch manuell dem MATLAB-Path hinzugefügt oder die Datei „INSTALLATION_TOOLBOX.m“ unter MATLAB ausgeführt werden.

Zur Kontrolle der erfolgreichen Installation, kann dies im MATLAB-Hauptfenster unter „File/Set Path“ kontrolliert werden.

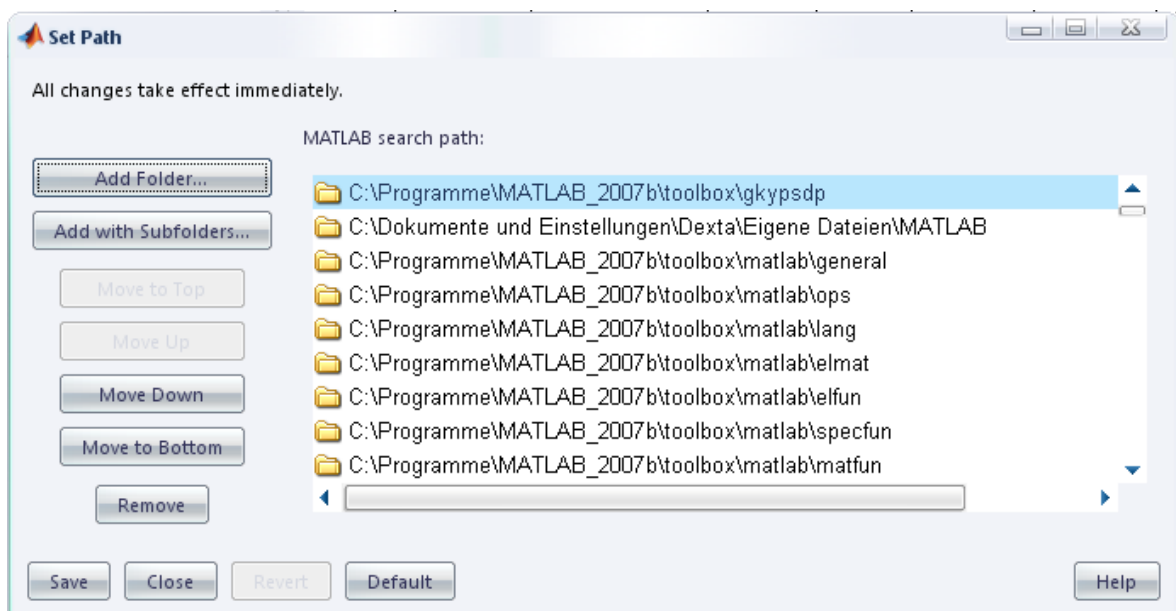


Abbildung 5.2: Hinzufügen eines MATLAB Datenpfades

5.1.2 Starten der Entwicklungsumgebung

Zum Starten von PIDexpert stehen auch mehrere Varianten zur Verfügung.

- 1.) „PIDexpert.bat“ ausführen.
- 2.) In MATLAB den Pfad des Ordners PIDexpert einstellen und in der Konsole PIDexpert eintippen.
- 3.) Die Datei „PIDexpert.m“ unter MATLAB öffnen und ausführen.

5.2 Auswahl der Regelstrecke

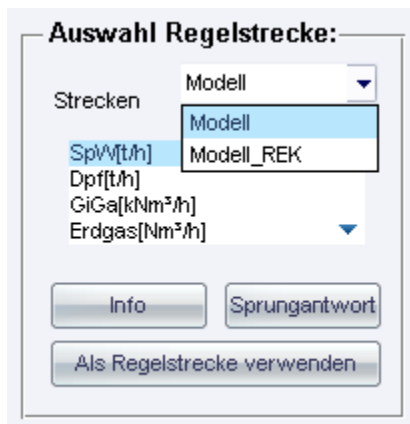


Abbildung 5.3: Auswahl Regelstrecke

Dieser Abschnitt der Benutzeroberfläche beinhaltet alle notwendigen Funktionen zur Auswahl der Regelstrecke. Wird ein Projekt neu angelegt bzw. geladen, stehen hier alle MISO-Modelle aus der Identifikationsumgebung Parameterident zur Verfügung. Die Namen der einzelnen Übertragungsfunktionen des ausgewählten Modells werden in einer List-Box zusammengefasst. Nach Auswahl einer der vorhandenen Strecken, können verschiedene Informationen bzw. die Sprungantwort der Strecke angezeigt werden. Diese Informationen können als Hilfestellung für die Auswahl einer geeigneten Entwurfsmethode verwendet werden.

Sprungantwort:

Bei dieser Funktion wird auf die ausgewählte Strecke ein Einheitssprung als Eingangsgröße gelegt. Das Ergebnis wird in einem Plot dargestellt und beinhaltet die Sprungantwort der zeitdiskreten- und der zeitkontinuierlichen Übertragungsfunktion, welche in weiterer Folge für den Entwurf verwendet wird.

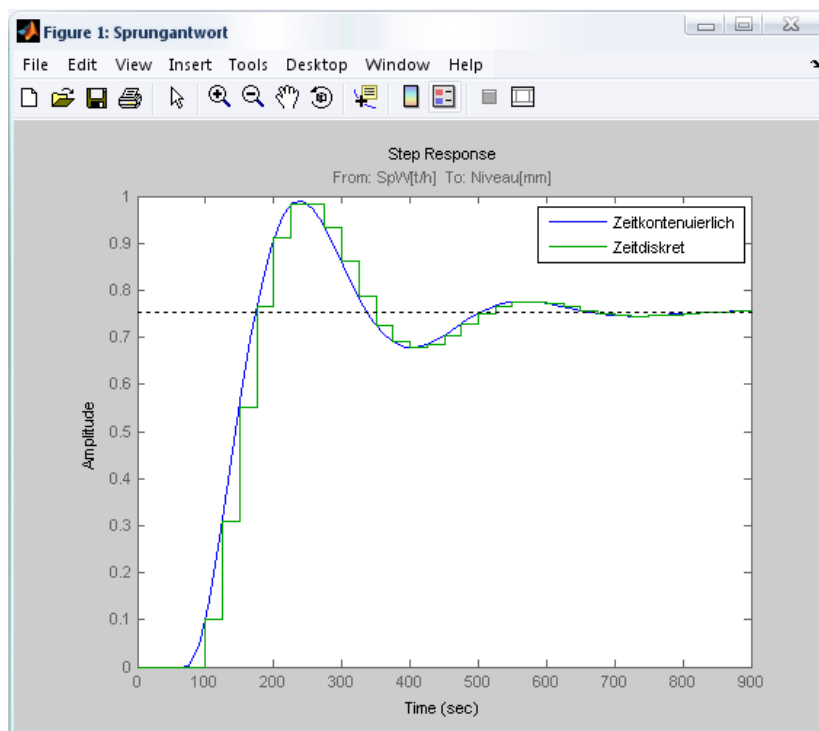


Abbildung 5.4: Sprungantwort der markierten Strecke

Strecken/Info:

Mittels dieser Funktion werden Informationen über die ausgewählte Übertragungsfunktion in einem Fenster angezeigt. Es beinhaltet die wichtigsten Daten aus der Identifikationsumgebung Parameterident und bietet so die Möglichkeit sich einen Überblick über die gewählte Strecke des identifizierten MISO-Modells zu schaffen.

- Eingang der Strecke
- Ausgang des MISO-Modells
- z-Übertragungsfunktion der Strecke mit der jeweiligen Abtastzeit
- Name des MISO-Modells
- Zusatzinformationen der Identifikation aus Parameterident

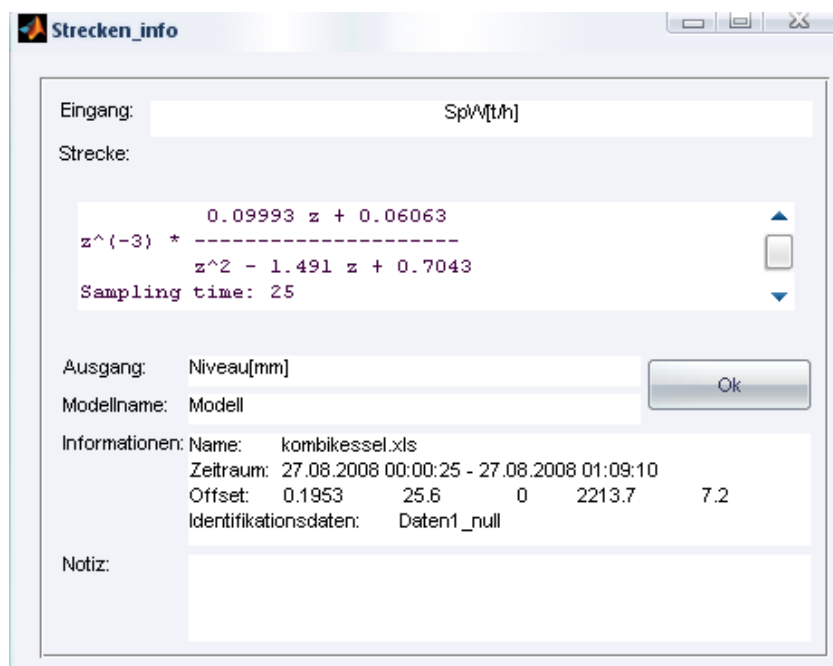


Abbildung 5.5: Informationen über die markierte Strecke

Als Regelstrecke verwenden:

Mit diesem Button wird die ausgewählte Übertragungsfunktion als Regelstrecke für den weiteren Entwurf übernommen und der nächste Abschnitt für den Entwurf wird frei geschaltet. Die Regelstrecke kann jederzeit während des Projektes neu gewählt werden, um weitere Regler für eine andere Übertragungsfunktion des gleichen bzw. eines anderen MISO-Modells zu entwickeln.

Ist bereits ein Regler entworfen und gespeichert worden, wird bei erneuter Auswahl einer Regelstrecke, diese mit dem markierten gespeicherten Regler simuliert. Hierbei kann der Anwender überprüfen, ob der Regler auch für die gleiche Strecke eines anderen MISO-Modells bzw. für eine beliebig andere Strecke funktioniert. Er kann in weiterer Folge entscheiden, ob dieser Regler mit der neuen Regelstrecke gespeichert oder ein neuer Entwurf durchgeführt werden soll.

5.3 Auswahl und Konfiguration des Entwurfsverfahrens

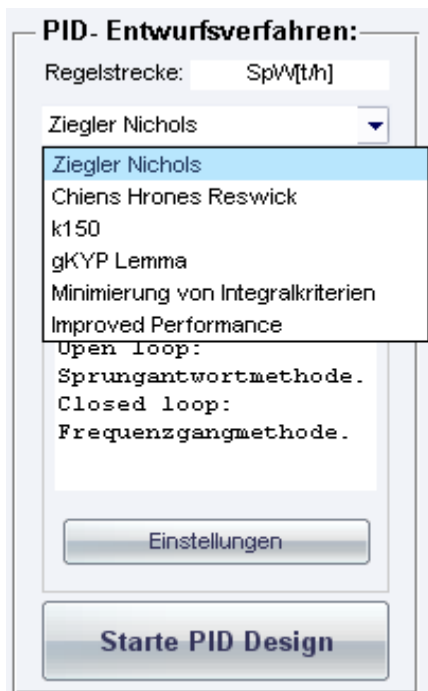


Abbildung 5.6: Auswahl und Konfiguration des Entwurfsverfahrens

Nach Festlegung einer Regelstrecke stehen in dieser Version der Software PIDexpert sechs verschiedene Entwurfsverfahren zur Verfügung. Für jedes Verfahren werden in einem Textfenster Informationen angezeigt, welche als Unterstützung bei der Auswahl dienen sollen. Für jedes Verfahren sind vor dem Entwurf Einstellungen vorzunehmen. Diese beinhalten die gewünschte Struktur des Reglers, Anti-Wind-up Konfiguration und spezifische Einstellungen für das jeweilige Verfahren.

Nach erfolgreicher Einstellung der Parameter für den Reglerentwurf, wird der Button für das PID-Design frei geschaltet und die Berechnung kann gestartet werden. Die Simulationsdauer wird in einer Statusleiste angezeigt.

Wichtig: Während des Entwurfes und der Simulation sollten keine weiteren Konfigurationen durchgeführt werden, um Dateninkonsistenz zu vermeiden!

Wurde ein Regler mit dem gewählten Verfahren erfolgreich ermittelt, wird das Entwurfsergebnis in Form einer Sprungantwort und in einem eigens dafür programmierten Fenster angezeigt. Der User hat nun die Möglichkeit, dem erhaltenen Regler einen Namen zu geben und ihn zu speichern oder den Regler zu verwerfen.

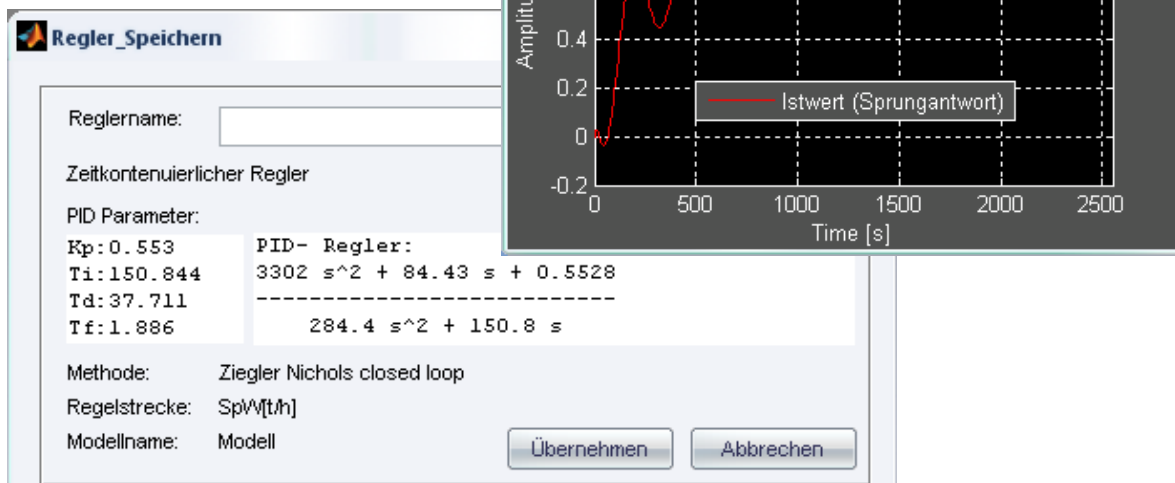


Abbildung 5.7: Speichern des erhaltenen Reglers

5.3.1 Einstellungen

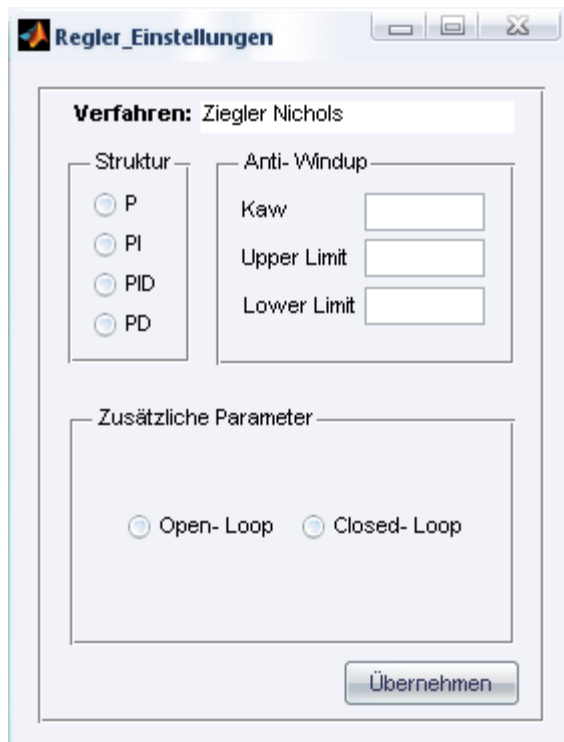


Abbildung 5.8: Einstellung des Entwurfsverfahrens

Für jedes Entwurfsverfahren wurde ein eigenes Konfigurationsfenster erstellt. Die Einstellung der Parameter für die Struktur des gewünschten Reglers und die Anti-Wind-up Maßnahme sind bei allen Verfahren gleich.

Die einzelnen Konfigurationsfenster unterscheiden sich lediglich bei den zusätzlichen Parametern.

- Reglerstruktur: P, PI, PID, PD
- Anti-Wind-up: k_{AW} , obere und untere Stellgrößenbeschränkung.
- Spezifische Einstellungen für das jeweilige Entwurfsverfahren.

Ziegler Nichols: P, PI, PID, PD

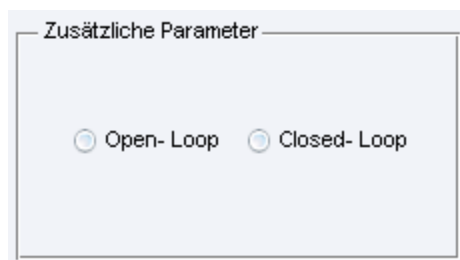


Abbildung 5.9: Spezifische Einstellungen bei Ziegler Nichols

Das Ziegler Nichols Verfahren ist für stark verzögernde Prozesse vorgesehen, welche einen s-förmigen Verlauf der Sprungantwort besitzen.

Open-Loop:

Aus der Sprungantwort werden durch Anlegen der Wendetangente die Kenngrößen a und T_v ermittelt. Die Reglerparameter für die jeweilige Reglerstruktur können mit diesen ermittelten Parametern berechnet werden.

Closed-Loop:

Aus dem Bode-Diagramm der Regelstrecke werden die Kenngrößen K_u (*ultimate gain*) und T_u (*ultimate period*) ermittelt und mit einer Berechnungsvorschrift werden die jeweiligen Reglerparameter der gewünschten Reglerstruktur festgelegt.

Chien Hrones Reswick: P , PI , PID

Abbildung 5.10: Spezifische Einstellungen bei Chien Hrones Reswick

Dieses Verfahren eignet sich, ähnlich wie bei Ziegler-Nichols, für Strecken mit einem s-förmigen Verlauf der Sprungantwort. Die einzelnen Parameter werden auch hier durch Anlegen der Wendetangente an die Sprungantwort erhalten. Hierbei kann jedoch der Wunsch eines aperiodischen bzw. schwingenden Einschwingverhaltens geäußert werden.

- Aperiodisch nach Aeström und Hägglund
- Aperiodisch nach Chien Hrones Reswick
- 20% Überschwingen nach Chien Hrones Reswick

k150-Verfahren: P , PI , PID

Kann nur für Strecken mit einer Phasendrehung $\geq -150^\circ$ eingesetzt werden. Dieses Verfahren eignet sich besonders gut für Totzeiten und kann vorteilhaft auf integrierende Strecken angewandt werden. Der Realisierungsterm für den D-Anteil des Reglers wird intern optimiert. Für dieses Verfahren sind keine zusätzlichen Parameter zu konfigurieren.

gKYP-Lemma: PID

Abbildung 5.11: Spezifische Einstellungen bei gKYP Lemma

Optimiert mittels Loop-Shaping einen PID-Regler. Wird beim Entwurf kein passender Regler erhalten, können die Standard-Designparameter für das Loop-Shaping Verfahren angepasst werden. Als Struktur für den zu entwerfenden Regler, kann bei diesem Verfahren nur PID ausgewählt werden. Falls eine andere Struktur (P , PI , PD) erwünscht ist, kann diese durch manuelle Anpassung der erhaltenen Parameter einfach erzeugt werden.

Die Zeitkonstante für den Realisierungsterm T_f kann in den Designparametern eingestellt werden.

omega low: Untere Grenze des Frequenzbereiches in der $L(j\omega)$ liegen soll.

omega high: Obere Grenze des Frequenzbereiches in der $L(j\omega)$ liegen soll.

th: $|L(j\omega)| \leq t_h$. Obere Grenze für den Betrag von $L(j\omega)$ bei hohen Frequenzen.

Tf: Realisierungsterm für den D-Teil des Reglers.

Minimierung von Integralkriterien: *P, PI, PID*

Abbildung 5.12: Spezifische Einstellungen bei Minimierung von Integralkriterien

Dieses Verfahren entwirft einen Regler durch Minimierung der Gütefunktionen ISE, IAE, ITSE und ITAE. Bei Auswahl eines PI-Reglers wird aus dynamischen Gründen ITAE gefordert. Der Entwurf mittels der Minimierung von Integralkriterien, kann zu einer langen Berechnungsdauer durch den Optimierungsalgorithmus führen. Die Startwerte für die Optimierung können aus den Ergebnissen von drei unterschiedlichen

Entwurfsverfahren ausgewählt werden:

- gKYP
- Ziegler Nichols
- Chien Hrones Reswick

Improved Performance: *PID*

Dieses Verfahren basiert auf einer Approximation der Regelstrecke durch ein System zweiter Ordnung mit Totzeit und einer Strategie zur Platzierung der Pole des geschlossenen Regelkreises. Der Entwurf zielt auf eine schnelle Annäherung der Ausgangsgröße ohne großes Überschwingen an den Sollwert des Regelkreises ab. Für dieses Verfahren sind keine zusätzlichen Parameter zu konfigurieren und es entwirft lediglich PID-Regler. Falls eine andere Struktur (P, PI, PD) erwünscht ist, kann diese durch manuelle Anpassung der erhaltenen Parameter einfach erzeugt werden.

5.4 Simulation des erhaltenen Reglers

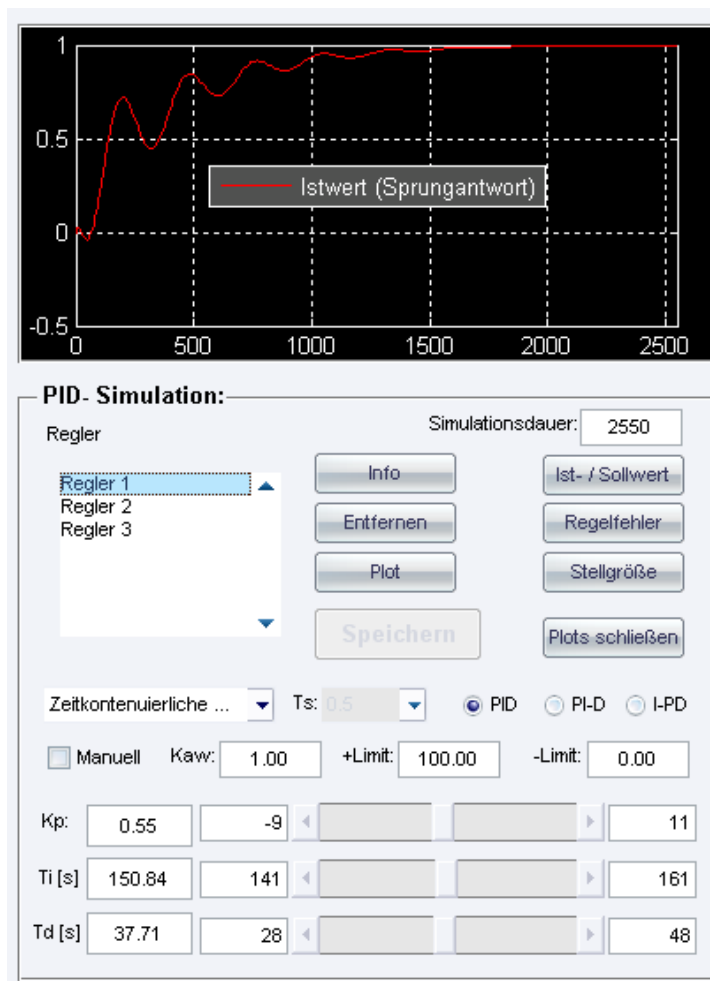


Abbildung 5.13: Simulation des erhaltenen Reglers

In diesem Bereich werden die gespeicherten Regler in einer List-box zusammengefasst und können bei Bedarf weiter variiert bzw. optimiert werden. Weiters besteht die Möglichkeit Informationen über den markierten Regler zu erhalten oder ihn aus dem Projekt zu entfernen. Besteht der Wunsch zwei Regler miteinander zu vergleichen, können die jeweiligen Verläufe der Simulationsergebnisse abgerufen werden. Um eine übersichtliche Darstellung der einzelnen Verläufe zu gewährleisten, wurde die Anzahl der Fenster in dieser Version auf sechs begrenzt. Die Fenster werden geordnet dargestellt und bei Aufruf eines siebenten Fensters wird das älteste geschlossen und durch das neue ersetzt.

Ist-/Sollwert, Regelfehler, Stellgröße:

Öffnet den jeweiligen Verlauf des markierten Reglers. Somit können die Verläufe von zwei verschiedenen Reglern miteinander verglichen werden, um einen besseren Eindruck des Regelverhaltens zu bekommen. Falls die Checkbox „Enable MISO“ aktiviert und eine Störgrößenkonfiguration vorhanden ist, werden die Verläufe der Störgrößensimulation dargestellt.

Plot:

Mit diesem Button werden die Verläufe des Ist-/Sollwerts, des Regelfehlers und der Stellgröße des markierten Reglers jeweils in einem eigenen Fenster angezeigt.

Plots schließen:

Diese Funktion ermöglicht das Schließen aller geöffneten Verläufe auf einen Klick.

Regler/Info:

Mit diesem Button werden nochmals alle Informationen über den markierten gespeicherten Regler angezeigt. Diese beinhalten zum Beispiel die Simulationmethode, das heißt, ob der Regelkreis zeitkontinuierlich bzw. zeitdiskret aufgebaut und simuliert wurde.

Weitere Informationen:

- Modellname: Name des gewählten MISO-Modells für den Entwurf
- Reglerparameter: K_P , T_I , T_D , T_f des gespeicherten Reglers
- Üfkt. des Reglers: Je nach Simulationmethode zeitkontinuierlich bzw. zeitdiskret
- Anti-Wind-up: k_{AW} , obere und untere Stellgrößenbeschränkung
- Methode: Name der verwendeten Methode für den Entwurf
- Regelstrecke: Name der gewählten Regelstrecke
- Regelstruktur: verwendete Regelstruktur für die Simulation (PID, PI-D, I-PD)

Außerdem besteht hier nochmals die Möglichkeit den Namen des Reglers gegebenenfalls anzupassen.

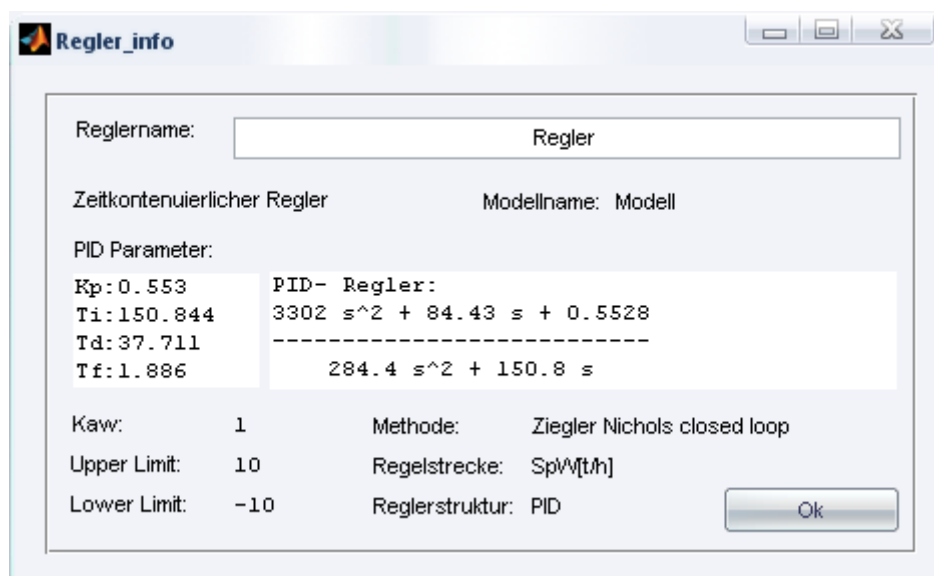


Abbildung 5.14: Informationen über gespeicherte Regler

Regler/Entfernen:

Markierte gespeicherte Regler, welche im laufenden Projekt von PIDexpert nicht mehr benötigt werden, können durch Betätigen des Buttons „Entfernen“ gelöscht werden.

Manuell:

Kann keines der Entwurfsverfahren einen zufriedenstellenden Regler entwickeln, wird mit der Checkbox „Manuell“ die Möglichkeit geboten einen Regler selbstständig zu entwickeln. Nach der Konfiguration aller notwendigen Parameter wird die Simulation des Regelkreises automatisch gestartet.

Speichern:

Mit dieser Funktion können Änderungen an den verschiedenen Parametern bzw. an den Einstellungen des Reglers für die Simulation übernommen werden. Das Fenster zur Speicherung wird geöffnet und es kann ein Name für den neuen Regler eingegeben werden. Dieser wird in weiterer Folge der List-Box der gespeicherten Regler hinzugefügt.

Simulationsdauer:

Die Simulationsdauer wird beim Reglerentwurf anhand der Sprungantwort jedes einzelnen geschlossenen Regelkreises geschätzt und kann bei Bedarf manuell angepasst werden. Bei einer Änderung wird dieser Parameter automatisch für den gewählten Regler gespeichert und alle Simulationen werden wiederholt, um immer auf die aktuellen Simulationsergebnisse zugreifen zu können. Die Dauer wird immer auf ein ganzzahliges Vielfaches der Abtastzeit des MISO-Modells gerundet.

5.4.1 Reglerparameter variieren

Die Änderung der Reglerparameter bzw. der Einstellungen für die Simulation sind in der Software PIDexpert in dem abgebildeten Bereich zusammengefasst.

Abbildung 5.15: Variation der Reglerparameter

Bei Auswahl eines gespeicherten Reglers aus der List-Box, werden seine Parameter bzw. seine Konfiguration dargestellt und können bei Bedarf variiert werden. Nach jeder Modifikation der einzelnen Werte wird automatisch eine neue Simulation gestartet und der Verlauf des Istwerts im Hauptfenster der Benutzeroberfläche dargestellt.

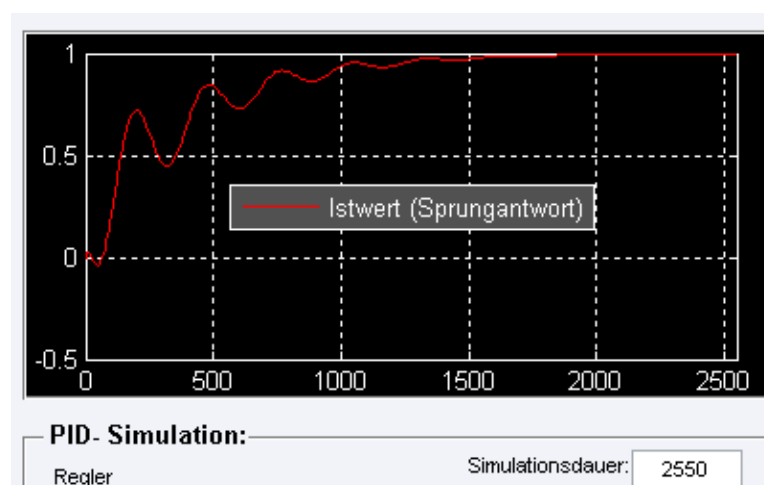


Abbildung 5.16: Simulationsergebnis nach Variation der Parameter

Wird ein Parameter während der Simulation geändert, so kommt dieser Wert in die Warteschlange und wird nach Beendigung der laufenden Simulation abgearbeitet. Bei Auswahl der zeitdiskreten Simulation kann die Abtastzeit im Bereich von 0.05 bis 5 Sekunden variiert werden, um den erhaltenen Regler bereits für mögliche praktische Implementierungen zu testen. Die Konfiguration der Anti-Wind-up Maßnahme kann hier ebenfalls angepasst werden.

Zusätzlich besteht hier die Möglichkeit die Regelstruktur von PID auf PI-D bzw. I-PD zu verändern und die Auswirkungen auf den Regelkreis, insbesondere auf die Stellgröße, zu untersuchen.

Die einzelnen Reglerparameter (K_P , T_I , T_D) können zusätzlich zur direkten Eingabe des Wertes mit einem „Slider“ schrittweise simuliert werden.

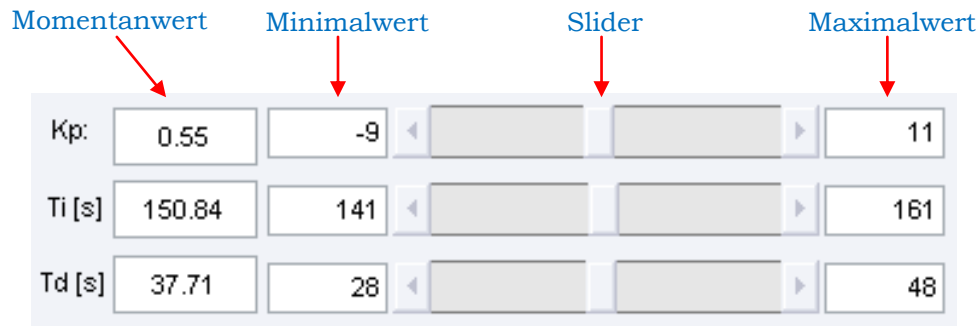


Abbildung 5.17: Variation der Parameter mittels Slider

Die Schrittweite berechnet sich aus dem Minimal- und Maximalwert des Sliders und beträgt ein Hundertstel der Differenz. Daraus folgt, dass für eine kleinere Schrittweite des Sliders der Maximal- und Minimalwert angepasst werden muss. Wird bei der direkten Eingabe ein Wert größer bzw. kleiner der gegebenen Slidergrenzen eingegeben, so wird der Wert automatisch auf den Maximal- bzw. Minimalwert gesetzt. Möchte man einen Betrag außerhalb der Grenzen eingeben, so müssen diese wiederum angepasst werden, um dies zu ermöglichen.

Anmerkung: Wird bei der Konfiguration der Parameter T_D variiert, ändert sich der Realisierungsterm für den D-Teil auf $T_f = |T_D|/20$ unabhängig vom Ergebnis aus dem Entwurf.

5.5 Simulation mit Störgrößen

Abbildung 5.18: Simulation mit Störgrößen

Dieser Abschnitt beschäftigt sich mit der Konfiguration der Eingangsgrößen der Regelstrecke (Sollwert) bzw. für die jeweilige Störgröße und kann mittels der Checkbox „Enable MISO“ frei geschaltet werden. Durch die Simulation mit den einzelnen Störgrößen ist es möglich, eine bessere Beurteilung des Reglers zu erhalten und zu prüfen, ob der gewählte Regler in der Praxis zum Einsatz kommen kann.

Nach einem erfolgreichen Entwurf eines Reglers stehen hier die einzelnen Teilsysteme des MISO-Modells in einer List-Box zusammengefasst, beginnend mit der Regelstrecke, zur Verfügung. Für die Simulation können mehrere Eingangsfunktionen ausgewählt werden:

- Sprungfunktion
- Rampenfunktion
- Identifikationsdaten
- Treppenfunktion

Die Einstellungen können gespeichert und jederzeit wieder abgerufen werden. Weiters wird die Möglichkeit geboten, sich Plots der einzelnen Verläufe der Eingangsgrößen bzw. der Ausgangsgrößen nach der jeweiligen Strecke darstellen zu lassen. Wird eine der

Konfigurationen nicht mehr benötigt, besteht natürlich wieder die Möglichkeit diese zu entfernen.

Anmerkung: Wird keine Eingangsgröße für die Regelstrecke definiert, wird als Sollwert null vorgegeben. Bei einer Simulation mit den Identifikationsdaten, wird um eine Praxisnähe zu gewährleisten, für alle Eingangsgrößen derselbe Datensatz verwendet (z.B.: Strecke1 → Daten 1 für Strecke1; Strecke2 → Daten 1 für Strecke2). Wird eine Simulationsdauer eingestellt, die größer ist als die Zeitspanne der aufgenommenen Prozessdaten, werden diese gespiegelt und dem Datensatz angehängt, um eine sprunghafte Eingangsgröße zu erhalten und somit längere Simulationen durchführen zu können.

Clear Konf.:

Diese Funktion initialisiert die Konfiguration der Eingangsfunktionen und löscht alle zuvor getätigten Einstellungen.

Plot-Konf. Eing./Plot-Konf. Ausg.:

Öffnet die Verläufe der Eingangsgrößen bzw. der Ausgangsgrößen nach der jeweiligen Strecke und dient als Kontrolle für die jeweilige Konfiguration.

Speich./Simul.:

Dient zur Speicherung der Konfiguration für die Eingangsgrößen und startet automatisch die Simulation des Regelkreises.

Konfiguration/Entfernen:

Markierte Konfigurationen, welche im laufenden Projekt von PIDexpert nicht mehr benötigt werden, können durch Betätigen des Buttons „Entfernen“ gelöscht werden.

5.6 Menüleiste

Im Menü befinden sich Funktionen zur Verwaltung und zum Export des Projekts. Weiters kann hier auf die vorhandene Programmdokumentation zugegriffen werden.

Neu/Speichern/Laden

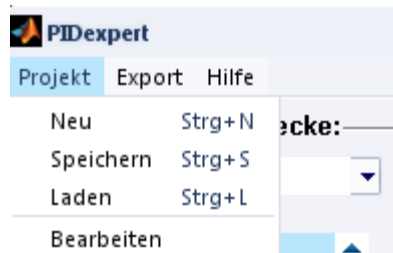


Abbildung 5.19:
Neu/Speichern/Laden PIDexpert

Mittels des Menüpunkts „Neu“ wird die Möglichkeit geboten, ein neues Projekt für PIDexpert anzulegen. Nach Auswahl eines fertigen Projektfiles aus der Identifikationsumgebung Parameterident, stehen die verschiedenen beinhalteten MISO-Modelle zur Verfügung.

Mit den Menüpunkten „Speichern“ können alle erhaltenen Regler, Simulationsergebnisse und Einstellungen in einem „*.mat“-File abgespeichert werden. Beim Laden dieser Datei kann der gespeicherte Zustand wieder hergestellt werden.

Bearbeiten

Mit dieser Funktion können nachträglich hinzugefügte Modelle bzw. Prozessdaten unter Parameterident im aktuellen PIDexpert Projekt hinzugefügt werden. Weiters kann hiermit das laufende Projekt verwaltet werden und gegebenenfalls Modelle bzw. Daten entfernt werden.

5.6.1 Export



Abbildung 5.20: Projektexport
PIDexpert

Der Menüpunkt Export ermöglicht dem Anwender die gespeicherten Regler des laufenden Projekts zu exportieren. Dabei stehen zwei verschiedenen Varianten zur Verfügung.

1. Generierung eines HTML-Reports
2. Export des Projekts in den MATLAB-Workspace

Report:

Bei dieser Variante hat der Anwender die Möglichkeit, einen bzw. mehrere gespeicherte Regler auszuwählen und einen HTML-Report zu generieren. Der Report beinhaltet die gesamte Reglerkonfiguration, sowie alle Simulationsergebnisse des ausgewählten Reglers.

Der Vorteil des HTML-Reports besteht darin, auch ohne MATLAB ein Überblick über die Ergebnisse des laufenden Projektes zu erhalten.

Wichtig: Im Speicherpfad der Reportdatei dürfen keine Umlaute enthalten sein!

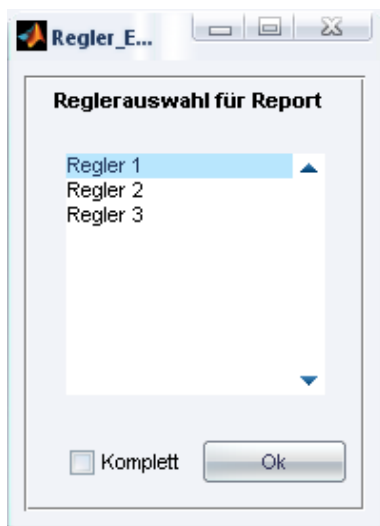


Abbildung 5.21: Auswahl der vorhandenen Regler für den Export

Aufbau des HTML-Reports:¹⁹

1. Name des Reglers
 - 1.1. Verwendete Regelstrecke
 - 1.2. Entwurfsverfahren
 - 1.3. Simulation
 - 1.3.1. Konfiguration 1
 - 1.3.2. Konfiguration 2 ...
2. Name des Reglers 2
 - 2.1. Verwendete Regelstrecke ...

Workspace:

Diese Option gestattet dem Anwender, alle gespeicherten Regler in den MATLAB-Workspace zu exportieren und somit die erhaltenen Ergebnisse auch für weitere Simulationen zu verwenden. Auf die einzelnen Regler kann in der Konsole mit „controller{i}“ (i = 1... Anzahl der Regler) zugegriffen werden.

¹⁹ Beispiel für Report siehe Anhang.

6 Beispiel aus der Praxis

Als praktisches Beispiel für die nähere Erklärung der einzelnen Möglichkeiten der Entwicklungsumgebung PIDexpert wurde das Modell eines Dampfkessels verwendet, welcher ein Teil einer Energieversorgungsanlage in einem Stahlwerk ist. Das dazugehörige mathematische Modell wurde im Zuge der Diplomarbeit „Entwicklung einer Software-Umgebung für die Parameteridentifikation industrieller Prozesse“ ermittelt und während Tests an der realen Anlage verfeinert.

6.1 Beschreibung der Anlage

Die in der Produktion von Stahl anfallende Menge an Gichtgas wird in der Kesselanlage zur Erzeugung von Heißwasserdampf verbrannt. Der entstehende Dampf wird für den Betrieb der Dampfturbine verwendet, welche den Generator für die betriebsinterne Energieversorgung antreibt. Die anfallende Menge von Gichtgas ist stark produktionsabhängig und kann bei Engpässen durch zusätzliches Erdgas kompensiert werden, um eine konstante Energieversorgung zu gewährleisten. Das Prinzip einer Dampfkraftanlage kann in drei Abschnitte eingeteilt werden:

- 1.) Erwärmung des Wassers auf Siedetemperatur
- 2.) Verdampfung
- 3.) Überhitzung

Das benötigte Wasser wird in einem Speisewasservorwärmer durch das Rauchgas auf Verdampfungsstempertur gebracht und in den Kessel gepumpt. Hier wird unter konstantem Druck weiter Wärme zugeführt und der entstehende Dampf in seinen Sättigungszustand gebracht. Der entstehende Dampf, welcher kein Wasser in flüssiger Form mehr erhält, wird Sattedampf genannt. Der Überhitzer erwärmt den Sattedampf weiter und erzeugt somit den sogenannten Heißdampf. Dieser wird in Druckleitungen der Turbine zugeführt, in der sich der Dampf wieder entspannt und sich im Kondensator wieder verflüssigt. Das erhaltene Wasser wird nun wieder mittels der Speisewasserpumpe in den Kessel zurückgepumpt. Das Prinzip ist in Abbildung 6.1 nochmals dargestellt.

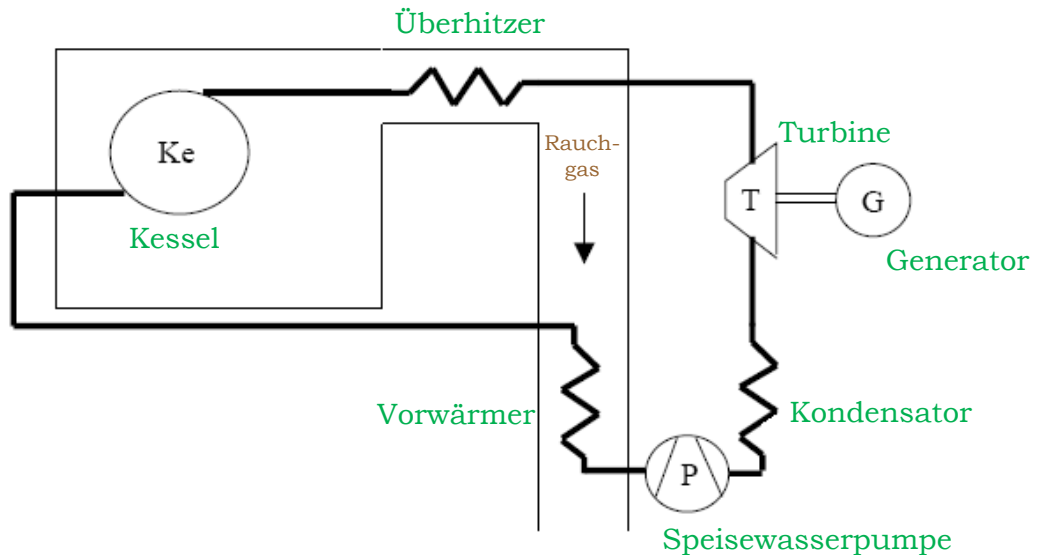


Abbildung 6.1: Prinzip einer Dampfkraftanlage aus [5]

Der Ausgangspunkt für die Wahl der Ein- und Ausgangsgrößen zur Modellbildung ist im Normalfall die zu regelnde Strecke. In diesem Beispiel ist das Ziel für die zu entwerfende Regelung, den Wasserstand im Kessel auf einem konstanten Niveau zu halten. Um dies zu ermöglichen bedarf es einer Regelung des Speisewasserzuflusses. Das Niveau hängt jedoch auch noch von weiteren Einflussgrößen ab, welche als Störgrößen auf die Regelstrecke berücksichtigt werden müssen. Näheres zu den verschiedenen Einflussgrößen siehe [5].

Das Modell mit dem Kesselniveau [mm] als Ausgangsgröße benötigt somit folgende Eingangsgrößen:

Eingang:

- Speisewasserzufluss [t/h]
- Verbrauchter Heißdampf [t/h]
- Verbrannte Gichtgasmenge [kNm³/h]
- Verbrannte Erdgasmenge [Nm³/h]

Für die vier Eingangsgrößen und die Ausgangsgröße wurden Messdaten aufgenommen und mittels der Identifikationsumgebung Parameterident mehrere mathematische Modelle ermittelt. Eines dieser erhaltenen Modelle wird nun in weiterer Folge für den Entwurf eines PID-Reglers zur Regelung des Kesselniveaus verwendet. Der erhaltene Regler wird mit einem anderen Modell auf seine Robustheit gegenüber Parameterschwankungen bei der Identifizierung getestet und mittels verschiedener weiterer Störgrößensimulationen verifiziert.

6.2 Reglerentwurf

Das nachfolgende mathematische Modell des Prozesses wurde mittels des Verfahrens „Minimierung der Fehlerquadrate“ identifiziert und wird in weiterer Folge für den Entwurf verwendet.

MODELL:

$$G_{SPW} = z^{-3} \frac{0.09993 z + 0.06063}{z^2 - 1.491 z + 0.7043} \quad G_{Dampf} = \frac{2.841 z - 3.075}{z^2 - 1.491 z + 0.7043}$$

$$G_{GiGa} = \frac{-0.1026 z + 0.1657}{z^2 - 1.491 z + 0.7043} \quad G_{Erdgas} = \frac{-0.01162 z + 0.01142}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Die Übertragungsstrecke G_{SPW} wird als Regelstrecke gewählt und in eine zeitkontinuierliche Übertragungsfunktion umgewandelt. Die vorhandene Totzeit wird mittels einer Pade-Approximation 2. Ordnung angenähert und man erhält somit folgende Strecke:

$$G_{SPW} = \frac{0.0007393 s^3 + 0.0002521 s^2 - 2.332e - 005 s + 6.639e - 007}{s^4 + 0.09402 s^3 + 0.003669 s^2 + 6.3e - 005 s + 8.821e - 007}$$

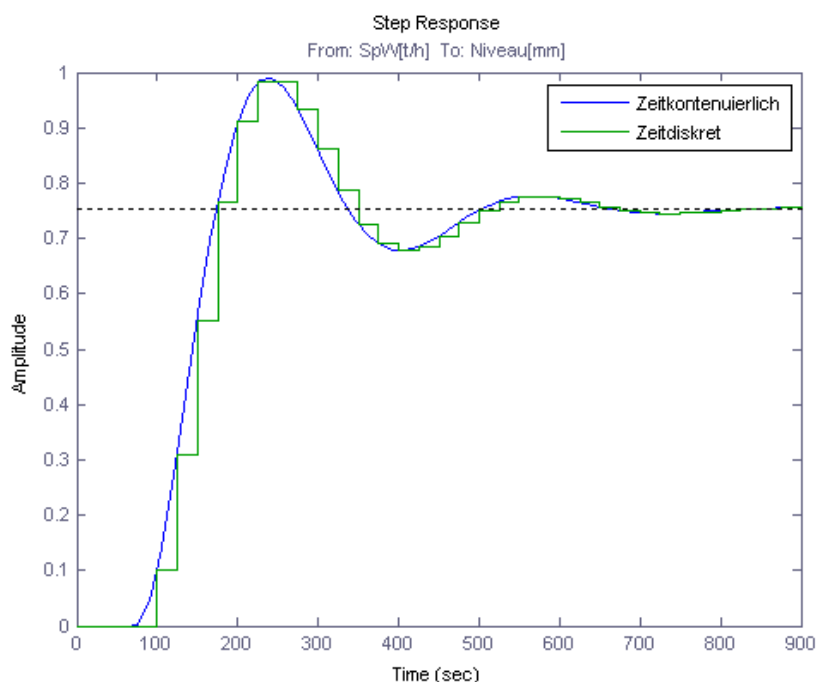


Abbildung 6.2: Sprungantwort der Regelstrecke

Als Stellgrößenbegrenzung wird der Messbereich der Messung für den Speisewasserzufluss 0 bis 100 t/h definiert und der Rückkopplungsfaktor für die Anti-Wind-up Maßnahme k_{AW} wird für die ersten Entwürfe auf Eins gesetzt. Die einzelnen Parameter können natürlich nach dem Entwurf entsprechend optimiert bzw. angepasst werden.

6.2.1 Ergebnisse der Entwurfsverfahren

Annähernd alle Entwurfsverfahren fanden eine Lösung für die gewählte Regelstrecke. Nur das Verfahren nach Chien Hrones Reswick und die open loop-Methode bei Ziegler Nichols konnten keine Ergebnisse für einen PID-Regler liefern. Bei der Minimierung von Integralkriterien wurde auf einen weiteren Entwurf mit anderen Startwerten verzichtet, da sich die erhaltenen Ergebnisse nur minimal von jenen mit dem Ziegler Nichols Startwert unterschied.

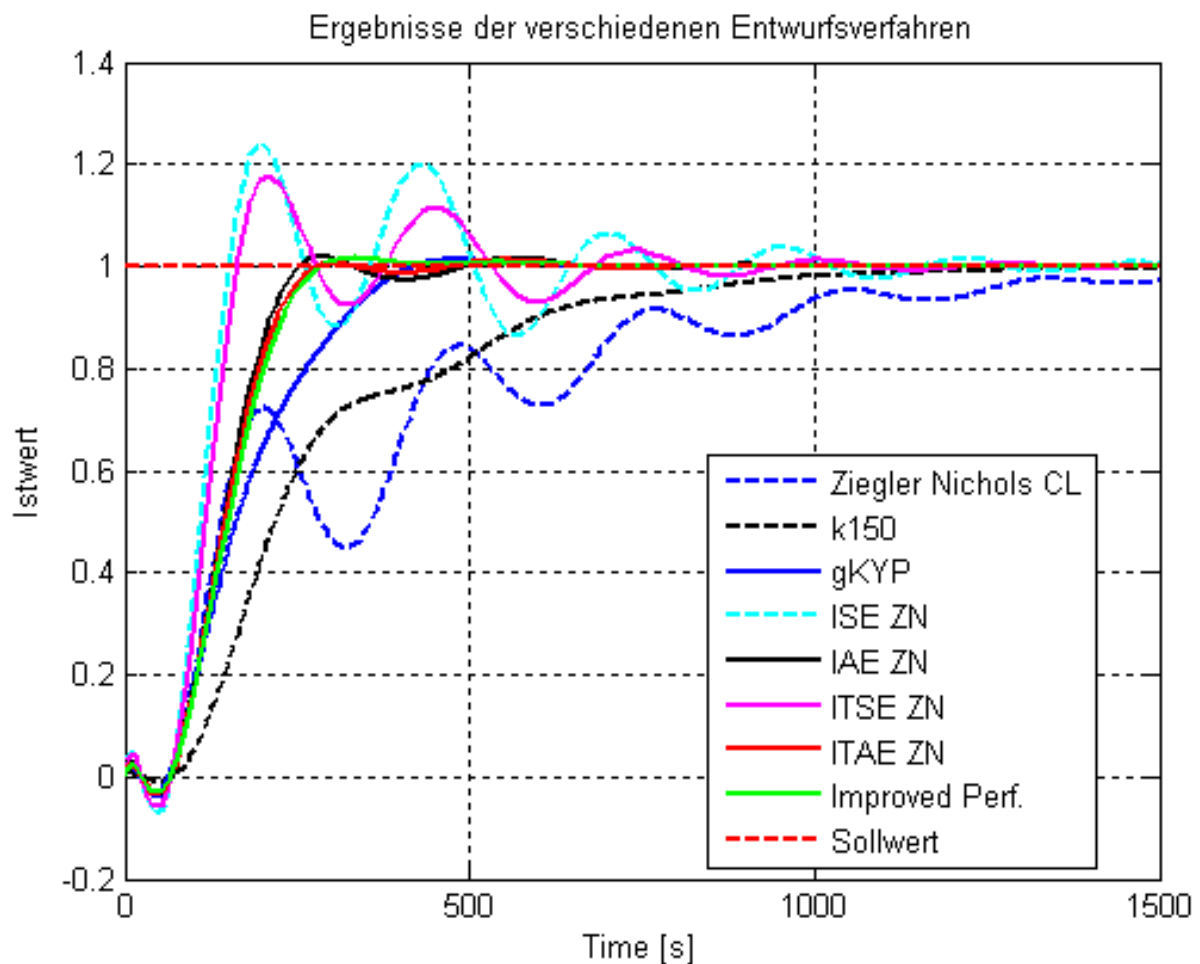


Abbildung 6.3: Verläufe der Entwurfsergebnisse des Praxisbeispiels

MISO-Modell:	Entwurfsverfahren	K_P	T_I	T_N	T_F
	<i>Ziegler Nichols CL</i>	0.5528	150.8443	37.7111	1.8856
Modell	<i>k150</i>	0.1993	45.5673	20.9058	0.7295
	<i>gKYP</i>	0.2341	32.1066	87.5372	0.5000
Anti-Wind-up	<i>ISE ZN</i>	0.4861	33.3901	101.9638	5.0982
	<i>IAE ZN</i>	0.3554	39.5042	58.6656	2.9333
k_{AW} : 1	<i>ITSE ZN</i>	0.4909	37.6126	83.3012	4.1651
upperlimit: 100	<i>ITAE ZN</i>	0.3325	37.6616	62.6962	3.1348
lowerlimit: 0	<i>Improved Perf.</i>	0.3013	34.4807	67.4183	3.3709

Tabelle 11: Reglerparameter aus den Entwurfsverfahren für das Praxisbeispiel

Aus den Istwertverläufen der einzelnen Regler in Abb. 6.3 und der Tabelle mit den Reglerparametern kann man deutlich erkennen, dass die Entwurfsverfahren unterschiedliche Ergebnisse liefern. Die Verfahren nach Ziegler Nichols bzw. ITSE weisen ein sehr träges bzw. ein sehr stark schwingendes Regelverhalten auf. Für die weitere Simulation sollte jedoch ein Regler mit einer guten Dynamik und nur geringem Überschwingen ausgewählt werden.

6.2.2 Auswahl des PID-Reglers

Der erhaltene Regler aus dem Entwurfsverfahren Improved Performance erfüllt für die weitere Vorgehensweise am besten die gewünschten Bedingungen und wird deshalb näher untersucht.

Methode:	Improved Performance	
PID	K_P	0.3013
	T_I	34.4807
	T_D	67.4183
	T_f	3.3709
Anti-Wind-up	k_{AW}	1
	+ Limit	100
	- Limit	0

Tabelle 12: Reglerparameter des gewählten Reglers

Für die Inbetriebnahme in der realen Anlage wird ein zeitdiskreter PID-Regler mit einer Abtastzeit von 0.5 Sekunden benötigt. Da die einzelnen Teilsysteme des verwendeten mathematischen Modells ein sehr träges Verhalten aufweisen, ist eine Umsetzung des zeitkontinuierlichen in einen zeitdiskreten Regler mit der gewählten Abtastzeit problemlos möglich und bedarf keiner Anpassung der Reglerparameter. Alle weiteren Simulationen werden ab jetzt mit dem zeitdiskreten Regler durchgeführt.

Wie bereits in Kapitel 2.3 erwähnt, kann die klassische Parallelstruktur des PID-Reglers zu unerwünschten Nebeneffekten führen. Eine sprunghafte Änderung der Führungsgröße führt, aufgrund des P- und des D-Anteils, zu betragsmäßig großen Stellgrößenwerten. Aus diesem Grund wird als nächstes eine Simulation mit den alternativ zur Verfügung stehenden Regelstrukturen PI-D und I-PD durchgeführt.

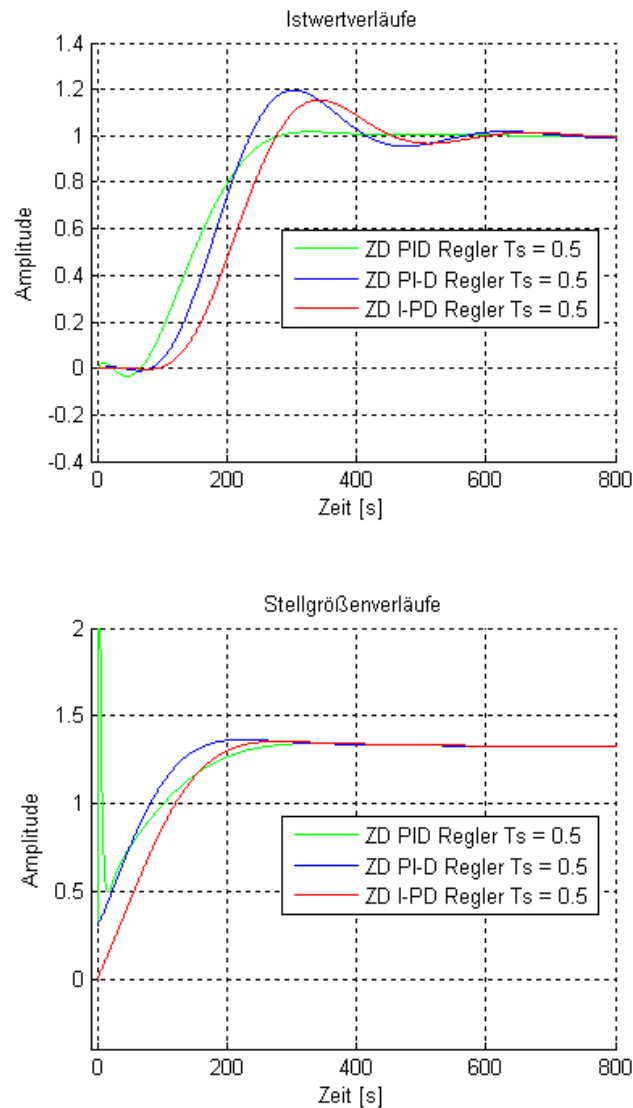


Abbildung 6.4: Istwert- und Stellgrößenverläufe mit den verschiedenen Regelstrukturen

Aus den zuvor abgebildeten Simulationsergebnissen ist ersichtlich, dass sich der Istwertverlauf des geschlossenen Regelkreises mit den verschiedenen Regelstrukturen etwas verschlechtert. Jedoch ist eine eindeutige Verbesserung des Stellgrößenverlaufs erkennbar, da keine sprunghaften Änderungen auftreten und sich ein betragsmäßig kleinerer Verlauf einstellt. Dieses Verhalten wird in der Praxis meist gewünscht und deshalb muss in weiterer Folge eine Wahl zwischen den beiden alternativen Regelstrukturen getroffen werden.

Da die I-PD Realisierung ein trägeres Regelverhalten aufweist, trifft die Wahl trotz eines geringeren Überschwingens auf die PI-D Struktur. Der gewählte Regler mit der PI-D Struktur wird nun mit der Regelstrecke aus einem anderen zur Verfügung stehenden Modell getestet.

Das nachfolgende mathematische Modell wurde in der Identifikationsumgebung Parameterident mit der Methode „Rekonstruktion der Störgrößen“ ermittelt.

MODELL_REK:

$$G_{SPW} = z^{-3} \frac{0.09891 z + 0.0578}{z^2 - 1.616 z + 0.8035} \quad G_{Dampf} = \frac{2.759 z - 3.088}{z^2 - 1.616 z + 0.8035}$$

$$G_{GiGa} = \frac{-0.09635 z + 0.1819}{z^2 - 1.616 z + 0.8035} \quad G_{Erdgas} = \frac{-0.01101 z + 0.01022}{z^2 - 1.616 z + 0.8035}$$

Sampling time: 25

Für den Test des ausgewählten Reglers wird nun in der Entwicklungsumgebung PIDexpert das zuvor gezeigte Modell ausgewählt und wiederum die Übertragungsfunktion G_{SPW} als Regelstrecke ausgewählt. Die Simulation wird automatisch gestartet und das Ergebnis nach erfolgreicher Beendigung angezeigt.

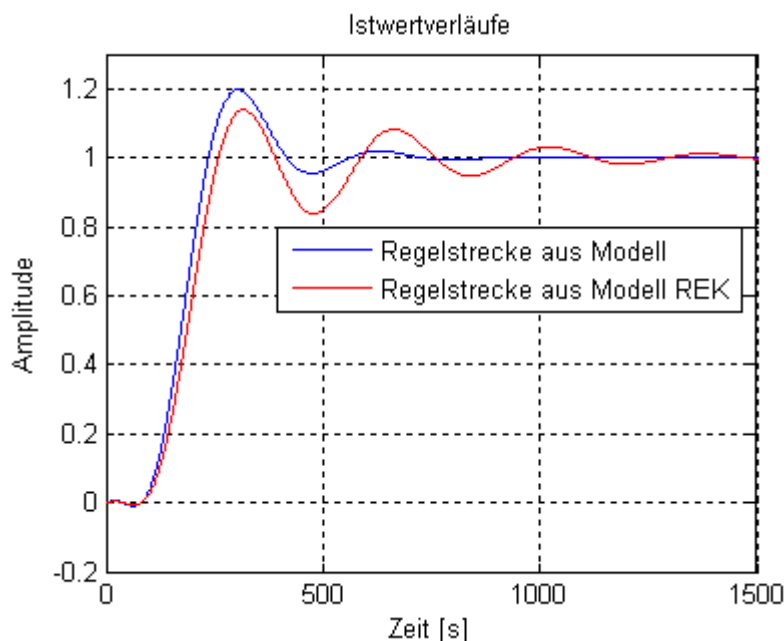


Abbildung 6.5: Vergleich mit der Regelstrecke aus Modell_REK

Aus der Abbildung 6.5 ist ersichtlich, dass der gewählte Regler auch für die Regelstrecke des zweiten Modells verwendet werden kann. Der Regler weist hier ein etwas schlechteres Regelverhalten auf, welches auf die etwas höhere Dynamik des Modells zurückzuführen ist. Es kann zwar eine Anpassung der Reglerparameter durchgeführt werden, um für beide Modelle ein annähernd gleiches Ergebnis zu erhalten, jedoch ist dieser Vorgang höchstwahrscheinlich ohnehin bei der Inbetriebnahme an der realen Anlage notwendig. Aus diesem Grund wird auf eine weitere Anpassung verzichtet und die Störgrößensimulation mit den bisherigen Parametern durchgeführt.

6.2.3 Störgrößensimulation

Bei dieser Simulation besteht nun die Möglichkeit, den erhaltenen Regelkreis mit den restlichen Teilsystemen des mathematischen Modells, welche als Störgrößen wirken, zu testen. Hierbei geht es darum, eine Beurteilung zu erhalten, ob der ermittelte Regler das Kesselniveau einen vorgegebenen Sollwert nachführen kann. Hierfür werden mehrere Versuche durchgeführt:

- 1.) Simulation mit selbstdefinierten Eingangsgrößen
- 2.) Simulation mit verschiedenen Prozessdaten und Vergleich mit dem Verlauf des Niveaus aus Parameterident

6.2.3.1 Störgrößensimulation mit selbstdefinierten Eingangsgrößen

Für den ersten Versuch wird ein Sollwertsprung für das Kesselniveau auf 1 mm nach 200 Sekunden durchgeführt. Die Heißdampfentnahme wird mittels einer Rampe mit einer Steigung von 0.001 t/h nach 1500 Sekunden um 5 t/h erhöht. Es wird angenommen, dass in der Produktion wieder mehr Gichtgas zur Verfügung steht und deshalb wird dieses mittels einer Rampe mit einer Steigung von 0.001 nach 1500 Sekunden auf 2 kNm³/h erhöht. Zur gleichen Zeit kann der teure Erdgasverbrauch ebenfalls mit einer Rampe mit Steigung 1 um 1000 Nm³/h verringert werden.

KONFIGURATION		PARAMETER				
		TYP	Amplitude	Zeitverzögerung	Offset	Sprungende
SPW	[t/h]	Sprung	1	200	0	inf
		TYP	Steigung	Zeitverzögerung	Offset	Rampen Limit
Dampf	[t/h]	Rampe	0.001	1500	0	5
Gichtgas	[kNm ³ /h]		0.001	1500	0	2
Erdgas	[Nm ³ /h]		-1	1500	0	1000

Tabelle 13: Störgrößensimulation für die selbstdefinierten Eingangsgrößen

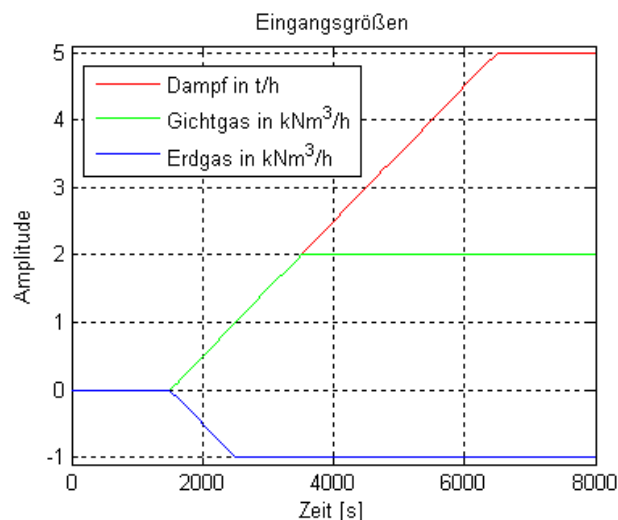


Abbildung 6.6: Verlauf der selbstdefinierten Eingangsgrößen

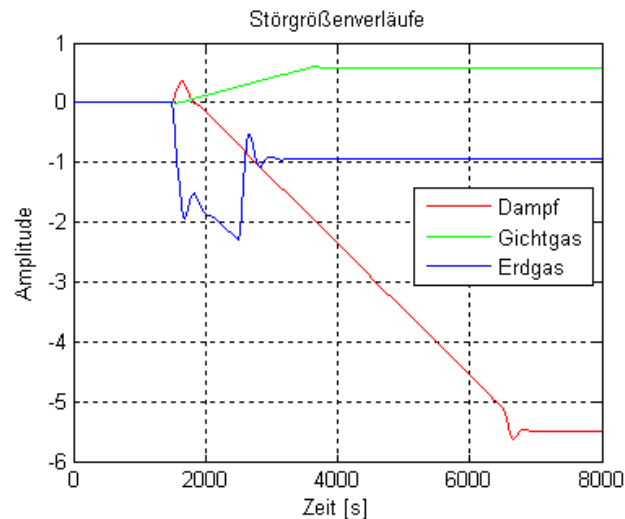


Abbildung 6.7: Verläufe der einzelnen Störgrößen

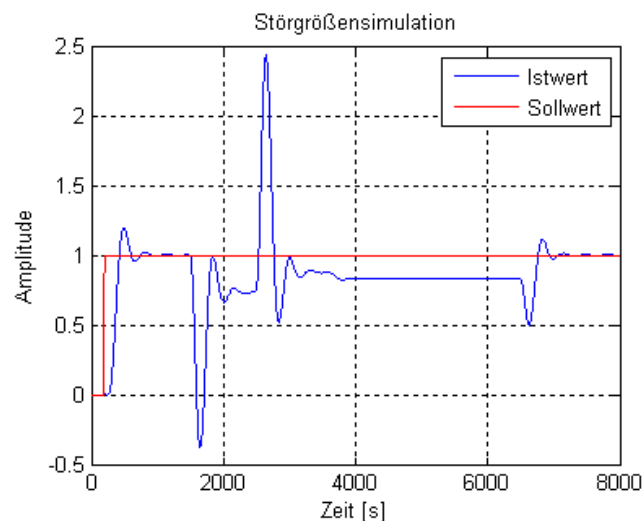


Abbildung 6.8: Verlauf des Soll- und Istwertes des Niveaus des Kesselwasserstandes

Aus den Ergebnissen der Simulation ist ersichtlich, dass sich der Istwert des geschlossenen Regelkreises nach Beendigung der rampenförmigen Aufschaltung der Störgrößen wieder auf den Sollwert einschwingt. Die Simulation lässt die Vermutung zu, dass der ermittelte Regler für die Inbetriebnahme an der realen Anlage geeignet ist. Dies wird jedoch noch durch weitere Simulationen mit den vorhandenen Prozessdaten verifiziert.

6.2.3.2 Störgrößensimulation mit Prozessdaten

Beim zweiten Versuch kommen als Eingangsgrößen für die Störgrößenübertragungsfunktionen die vorhandenen Prozessdaten der Identifikation aus Parameterident zum Einsatz. Für die Eingangsgrößen werden jene Daten verwendet, bei der der Arbeitspunkt bereits berücksichtigt²⁰ wurde und es werden für jede Übertragungsfunktion die vorhandenen spezifischen Daten verwendet.

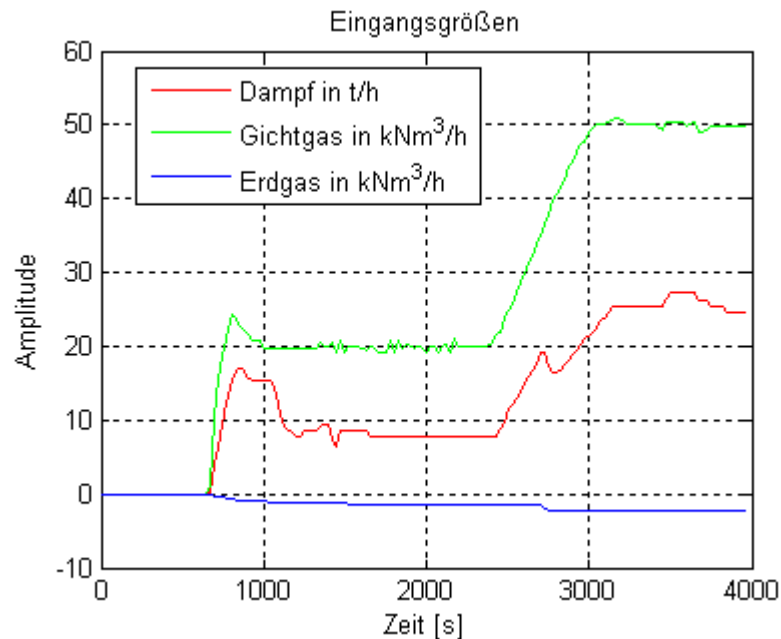


Abbildung 6.9: Beispiel für Verläufe der Prozessdaten mit berücksichtigten Arbeitspunkt

Die nachfolgenden Simulationen werden mit drei verschiedenen Datensätzen durchgeführt. Das Ergebnis wird zum einen als Ist- und Sollwertplot und zum anderen ein Vergleich mit den simulierten Daten des Niveauverlaufs aus Parameterident und mit dem Ergebnis des geregelten Niveaus der Simulation des geschlossenen Regelkreises in PIDexpert dargestellt. Weiters werden die Verläufe der Ausgänge nach den einzelnen Störgrößenübertragungsfunktionen angezeigt, um einen Überblick über die, auf den Regelkreis wirkenden, Störgrößen zu erhalten. Da die gemessenen Eingangsdaten bereits aus einem geregelten Prozess aufgenommen wurden, bietet der Vergleich mit dem Istwert der Niveauregelung einen guten Überblick über das Regelverhalten des entworfenen Reglers in der Praxis. Für den Sollwert des Niveaus wird für die Simulation ein konstanter Wert vorgegeben. Dies begründet sich dadurch, dass der Sollwert in der realen Anlage noch immer per Hand eingestellt wird und während der Messung der Prozessdaten mit ziemlicher Sicherheit nicht verändert wurde. Da bei den verwendeten Daten der Arbeitspunkt bereits berücksichtigt wurde und deshalb alle Eingangsgrößen bei null beginnen, wird der konstante Sollwert ebenfalls mit null vorgegeben.

²⁰ Näheres siehe [5]

Simulation mit „Daten1 null“:

KONFIGURATION		PARAMETER				
		TYP	Amplitude	Zeitverzög.	Offset	Sprungende
SPW	[t/h]	Sprung	0	0	0	inf
		TYP	Verstärkung	Zeitverzög.	Offset	Daten
Dampf	[t/h]	Identifikationsdaten	1	0	0	Daten1_null
Gichtgas	[kNm ³ /h]		1	0	0	
Erdgas	[Nm ³ /h]		1	0	0	

Tabelle 14: Störgrößenkonfiguration für Daten1_null

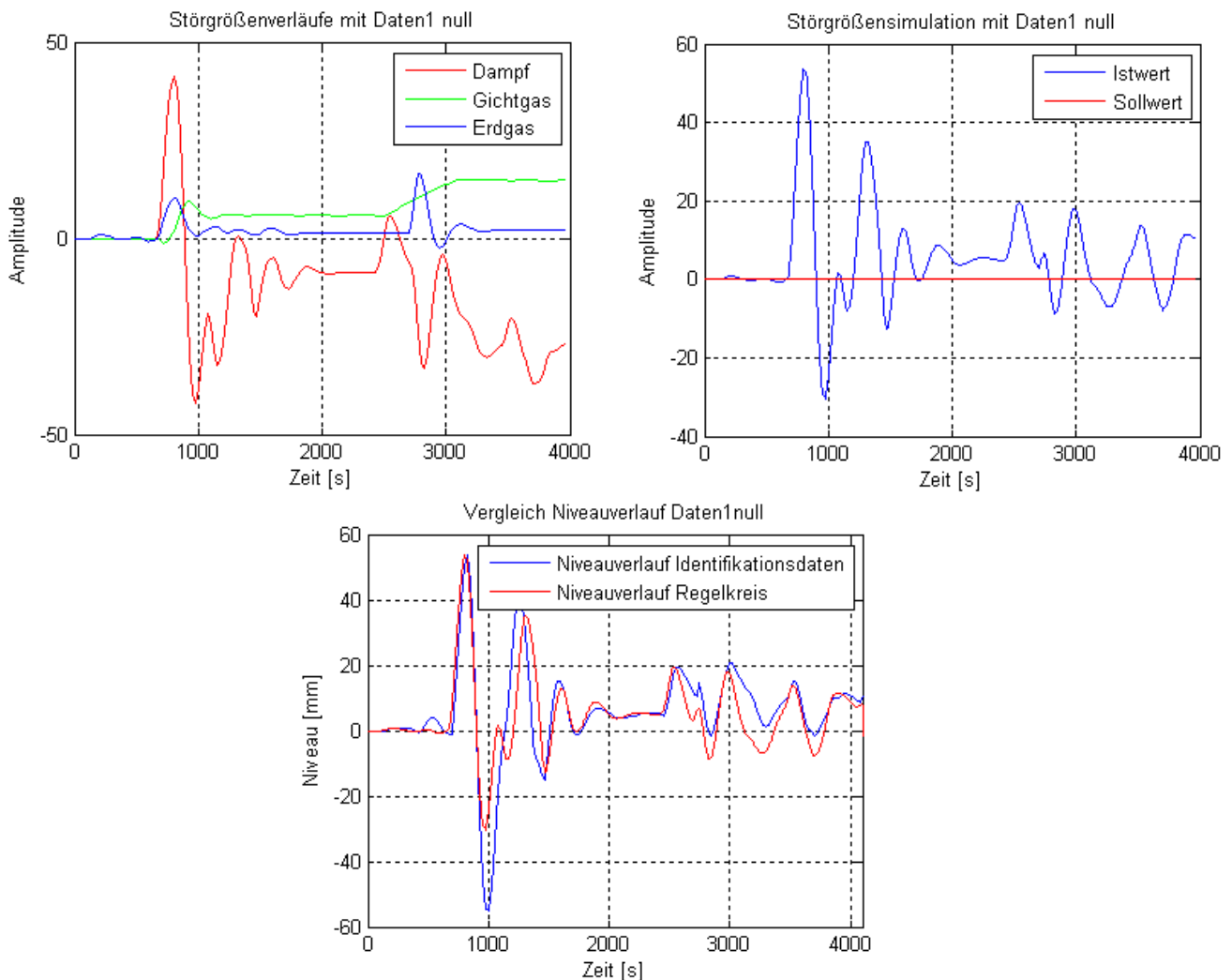


Abbildung 6.10: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten1_null

Aus den Ergebnissen der Störgrößensimulation für die Eingangsgrößen aus den „Daten1_null“ ist ersichtlich, dass der entworfene Regler ein etwas besseres Regelverhalten als der derzeit implementierte Regler aufweist. Er besitzt eine bessere Dynamik und ein geringeres Über- bzw. Unterschwingen. Aus den erhaltenen Verläufen kann die Schlussfolgerung gezogen werden, dass mit den erhaltenen Reglerparametern eine Verbesserung des vorhandenen Reglers durchgeführt werden kann.

Simulation mit „Daten2 null“:

KONFIGURATION		PARAMETER				
		TYP	Amplitude	Zeitverzög.	Offset	Sprungende
SPW	[t/h]	Sprung	0	0	0	inf
		TYP	Verstärkung	Zeitverzög.	Offset	Daten
Dampf	[t/h]	Identifikations- daten	1	0	0	Daten2_null
Gichtgas	[kNm ³ /h]		1	0	0	
Erdgas	[Nm ³ /h]		1	0	0	

Tabelle 15: Störgrößenkonfiguration für Daten2_null

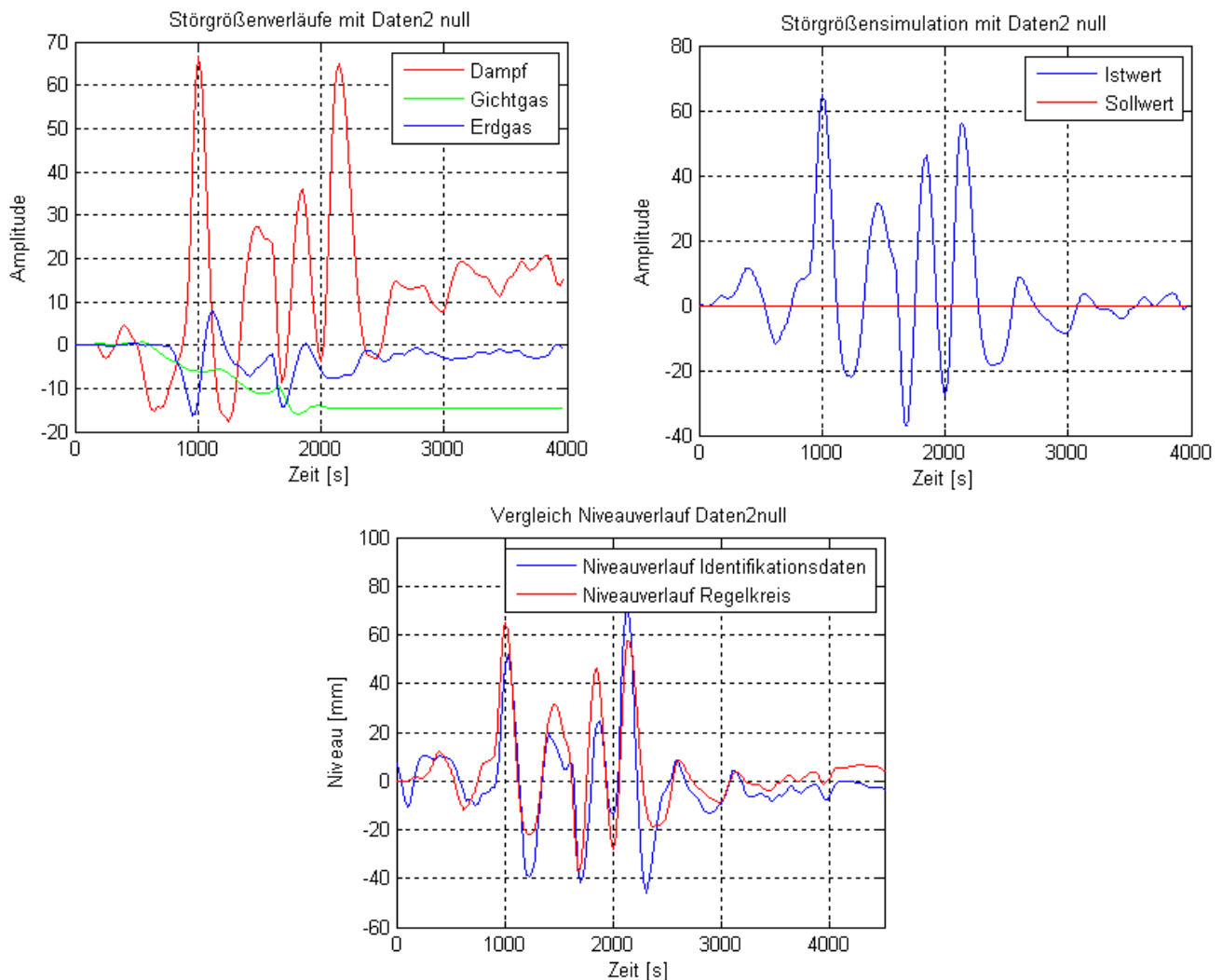


Abbildung 6.11: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten2_null

Simulation mit „Daten3 null“:

KONFIGURATION		PARAMETER				
		TYP	Amplitude	Zeitverzög.	Offset	Sprungende
SPW	[t/h]	Sprung	0	0	0	inf
		TYP	Verstärkung	Zeitverzög.	Offset	Daten
Dampf	[t/h]	Identifikationsdaten	1	0	0	Daten3_null
Gichtgas	[kNm ³ /h]		1	0	0	
Erdgas	[Nm ³ /h]		1	0	0	

Tabelle 16: Störgrößenkonfiguration für Daten3_null

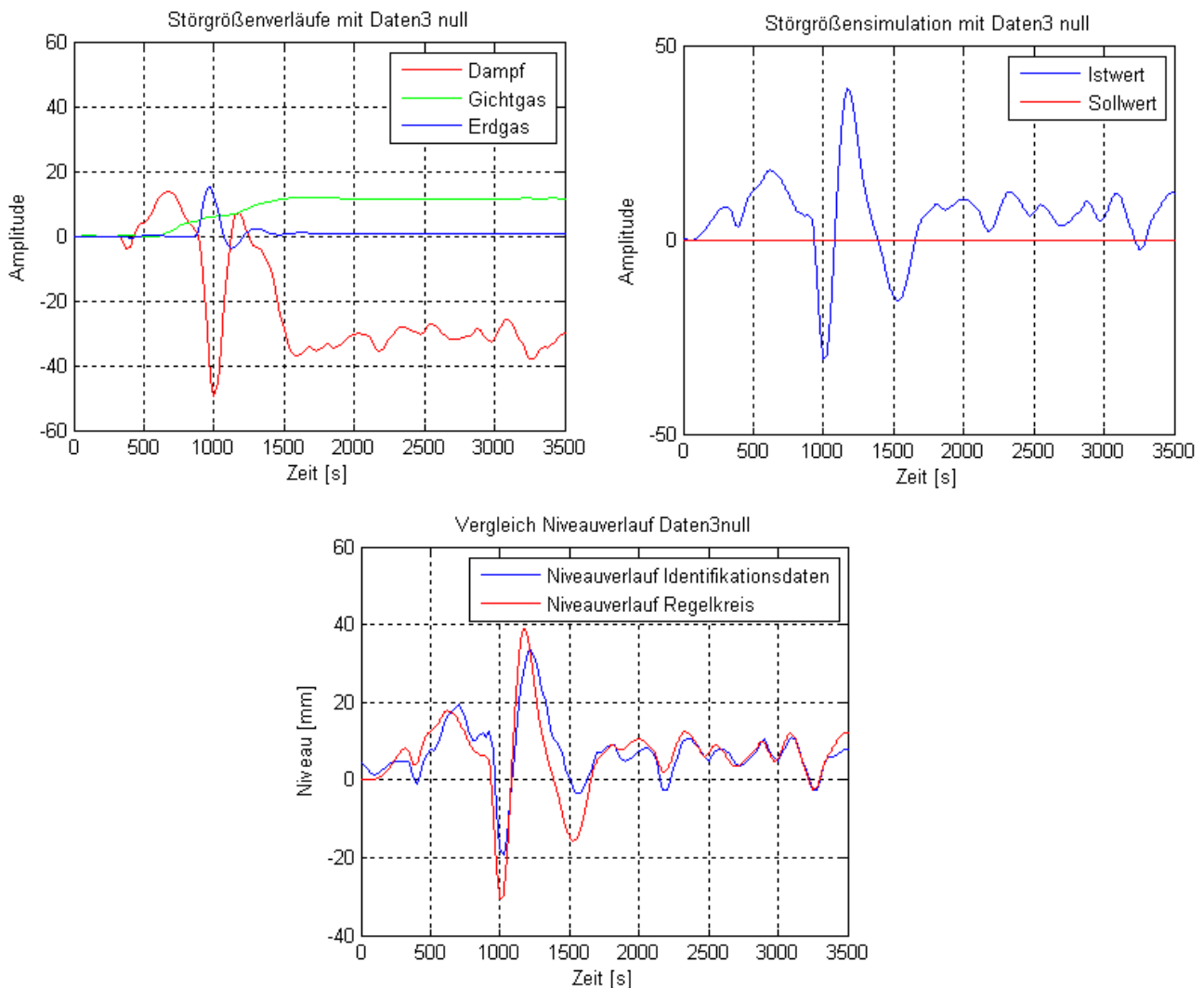


Abbildung 6.12: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten3_null

Die Simulationen mit den weiteren Prozessdaten „Daten2_null“ und „Daten3_null“ weisen ebenfalls ein annähernd gleich gutes Regelverhalten wie der in der Anlage implementierte Regler auf. Der Grund, dass aus diesen Verläufen die Verbesserung nicht eindeutig ersichtlich ist, kann darauf zurückgeführt werden, dass das Modell aus den Prozessdaten „Daten1_null“ identifiziert wurde.

7 Zusammenfassung und Ausblick

Die in dieser Arbeit entwickelte Software *PIDexpert* komplettiert, mit der Identifikationsumgebung *Parameterident*, die Softwareumgebung der Firma TG GmbH und kann als Basis für zukünftige Optimierungsanwendungen in bestehenden Industrieanlagen verwendet werden. Hiermit kann die Optimierung vorhandener Regler einfacher und effizienter durchgeführt werden, wodurch sich Vorteile für das technische Personal bei der Inbetriebnahme als auch für den Kunden ergeben. Mittels der Softwareumgebung können Stillstandszeiten der Anlage verringert werden und durch Anpassung der Reglerparameter ein wirtschaftlicherer und umweltfreundlicherer Betrieb ermöglicht werden.

Um den Entwurf eines Reglers mit einer gewünschten Struktur zu erleichtern und die Benutzerfreundlichkeit zu erhöhen, wurde eine GUI unter MATLAB erstellt. Aus der großen Anzahl von Entwurfsverfahren zur Ermittlung der Reglerparameter für PID-Regler, wurden einige ausgewählt und implementiert. Diese liefern ein gutes Paket, um den großen Bereich der möglichen Regelstrecken abzudecken, da nicht jedes Verfahren für beliebige Strecken einsetzbar ist. Falls keines der Verfahren eine Lösung findet, wird die Möglichkeit geboten, einen Regler durch manuelle Variation der Parameter zu ermitteln. Mit einem erhaltenen Regler können nun weitere Simulationen, wie z.B. alternative Regelstrukturen, eine zeitdiskrete Realisierung mit verschiedenen Abtastzeiten und eine Variation der Anti-Wind-up Konfiguration durchgeführt werden. Die einzelnen Verläufe des geschlossenen Regelkreises der verschiedenen Regler können miteinander verglichen werden, um eine erste Beurteilung des Regelverhaltens zu erhalten und gegebenenfalls eine Anpassung der Reglerparameter durchzuführen. Nach Auswahl eines Reglers, kann eine Simulation mit den vorhandenen Störgrößen durchgeführt werden. Hierbei besteht die Möglichkeit verschiedene Eingangsgrößen für die Störgrößenübertragungsfunktionen zu konfigurieren und die Simulationsergebnisse zu vergleichen. Das fertige Projekt kann in ein HTML-File bzw. in den MATLAB Workspace exportiert werden, um auch ohne MATLAB einen Überblick über die Ergebnisse zu erhalten bzw. weitere Simulationen durchzuführen.

Ideen für eine Verbesserung der Benutzerfreundlichkeit bzw. für eine Erweiterung der Möglichkeiten der entwickelten Software, werden sich sicherlich nach einigen Verwendungen in der Praxis ergeben. Wobei dies ohne großen Aufwand mit der MATLAB/Simulink Software durchgeführt werden kann.

Zum Abschluss dieser Arbeit ist nochmals darauf hinzuweisen, dass nur ein gutes mathematisches Modell, welches die Dynamik des Prozesses treffend beschreibt und somit annähernd der Realität entspricht, gute Voraussetzungen für einen erfolgreichen Reglerentwurf mit *PIDexpert* liefert.

8 Anhang

8.1 HTML-Report

Auszug aus einer Reportdatei

PIDexpert Report

Benutzer z.B.: Draxler Martin

Datum Uhrzeit z.B.: 17-Nov-2009 15:05:48

Inhaltsverzeichnis

- 1. IMP ZD PI-D 0.5
 - 1.1. Verwendete Regelstrecke
 - 1.2. Entwurfsverfahren
 - 1.3. Simulation
 - 1.3.1. Konfiguration 1
 - 1.3.2. Konfiguration 2

Kapitel 1. IMP ZD PI-D 0.5

Inhaltsverzeichnis

- 1.1. Verwendete Regelstrecke
- 1.2. Entwurfsverfahren
- 1.3. Simulation
 - 1.3.1. Konfiguration 1
 - 1.3.2. Konfiguration 2

1.1. Verwendete Regelstrecke

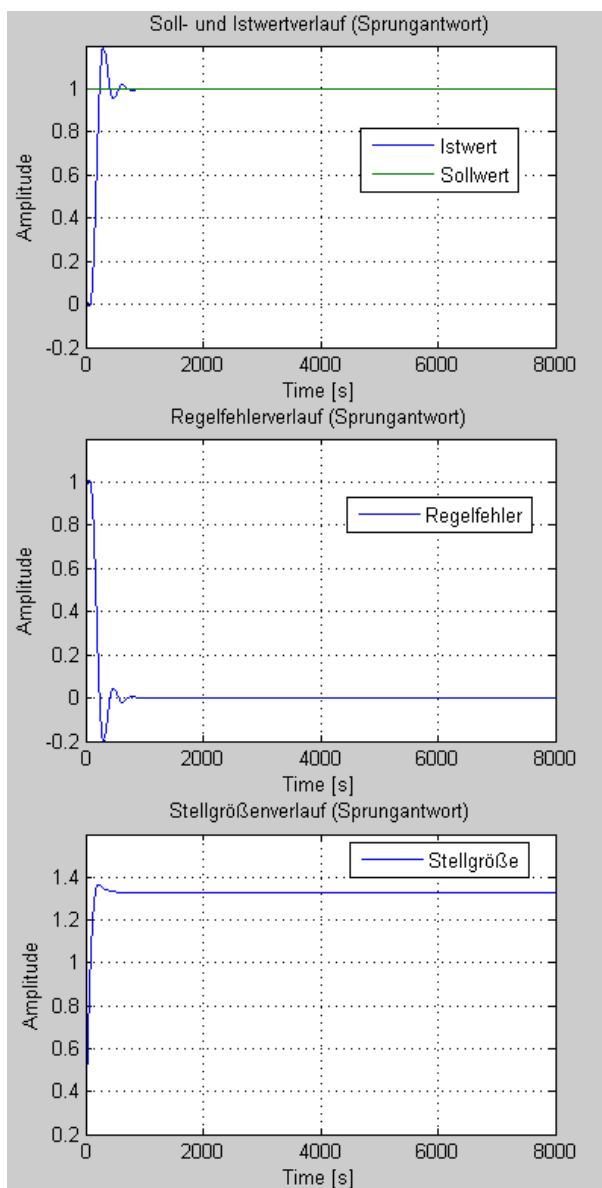
Transfer function from input "SpW[t/h]" to output "Niveau[mm]":
$$z^{(-3)} * \frac{0.09993 z + 0.06063}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

1.2. Entwurfsverfahren

Verwendetes Modell: Modell
Verwendete Methode: Improved Performance Methode
Typ: PID
Simulations Methode: ZD
Abtastzeit: 0.5000 Sekunden
Regler Struktur: PI-D

$K_p = 0.3013$
 $T_i = 34.4807$
 $T_d = 67.4183$
 $T_f = 3.3709$ (Zeitkonstante für Realisierungsterm)
 $k_{aw} = 1$
upperlimit = 100
lowerlimit = 0



1.3. Simulation

1.3.1. Konfiguration 1

1.3.1.1. Regelstrecke SpW[t/h]

Transfer function from input "SpW[t/h]" to output "Niveau[mm]":

$$z^{-3} * \frac{0.09993 z + 0.06063}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Regelstrecke:

Verwendeter Typ: Step

Amplitude: 1
Zeitverzögerung [s]: 200
Offset: 0
Sprungende [s]: Inf

1.3.1.2. Störgröße Dpf[t/h]

Transfer function from input "Dpf[t/h]" to output "Niveau[mm]":

$$\frac{2.841 z - 3.075}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:

Verwendeter Typ: Ramp

Steigung Rampe: 1.0000e-003
Zeitverzögerung [s]: 1500
Offset: 0
Rampen Limit [Betrag]: 5

1.3.1.3. Störgröße GiGa[kNm³/h]

Transfer function from input "GiGa[kNm³/h]" to output "Niveau[mm]":

$$\frac{-0.1026 z + 0.1657}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:

Verwendeter Typ: Ramp

Steigung Rampe: 1.0000e-003
Zeitverzögerung [s]: 1500
Offset: 0
Rampen Limit [Betrag]: 2

1.3.1.4. Störgröße Erdgas[Nm³/h]

Transfer function from input "Erdgas[Nm³/h]" to output "Niveau[mm]":

$$-0.01162 z + 0.01142$$

$$z^2 - 1.491 z + 0.7043$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:

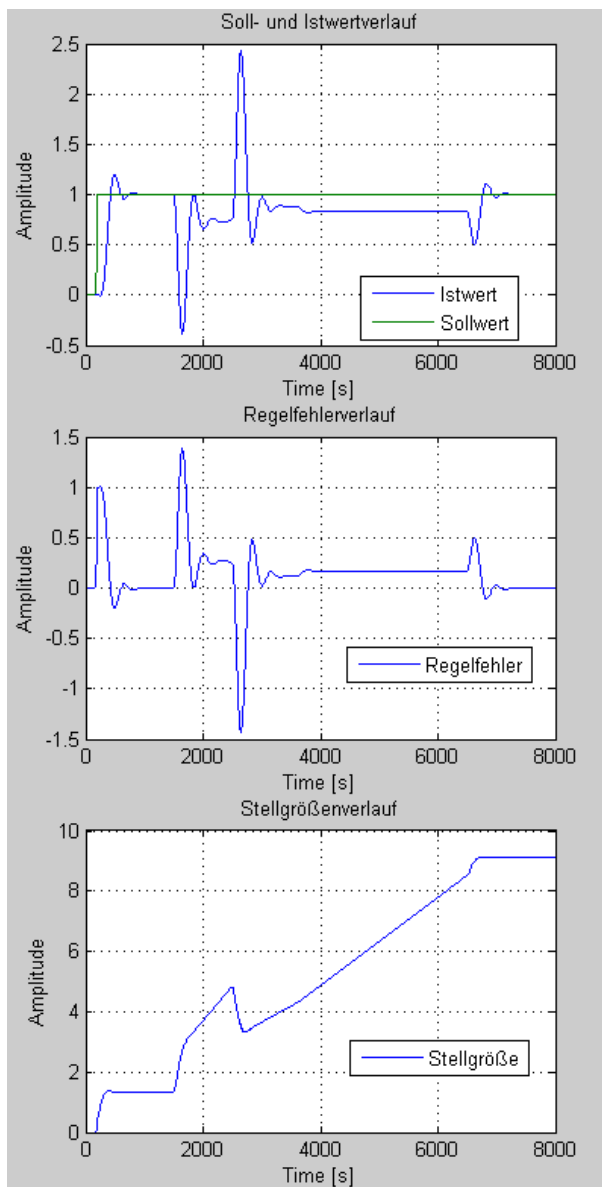
Verwendeter Typ: Ramp

Steigung Rampe: -1

Zeitverzögerung [s]: 1500

Offset: 0

Rampen Limit [Betrag]: 1000



1.3.2. Konfiguration 2

1.3.2.1. Regelstrecke SpW[t/h]

Transfer function from input "SpW[t/h]" to output "Niveau[mm]":

$$z^{(-3)} * \frac{0.09993 z + 0.06063}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Regelstrecke:
Verwendeter Typ: Step

Amplitude: 5
Zeitverzögerung [s]: 1000
Offset: 0
Sprungende [s]: Inf

1.3.2.2. Störgröße Dpf[t/h]

Transfer function from input "Dpf[t/h]" to output "Niveau[mm]":

$$\frac{2.841 z - 3.075}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:
Verwendeter Typ: Ident_Data

Verwendeter Datensatz: Daten1_null
Verstärkung: 1
Zeitverzögerung [s]: 0
Offset: 0

1.3.2.3. Störgröße GiGa[kNm³/h]

Transfer function from input "GiGa[kNm³/h]" to output "Niveau[mm]":

$$\frac{-0.1026 z + 0.1657}{z^2 - 1.491 z + 0.7043}$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:
Verwendeter Typ: Ident_Data

Verwendeter Datensatz: Daten1_null
Verstärkung: 1
Zeitverzögerung [s]: 0
Offset: 0

1.3.2.4. Störgröße Erdgas[Nm³/h]

Transfer function from input "Erdgas[Nm³/h]" to output "Niveau[mm]":

$$-0.01162 z + 0.01142$$

$$z^2 - 1.491 z + 0.7043$$

Sampling time: 25

Konfiguration Eingangssignal Störgröße:

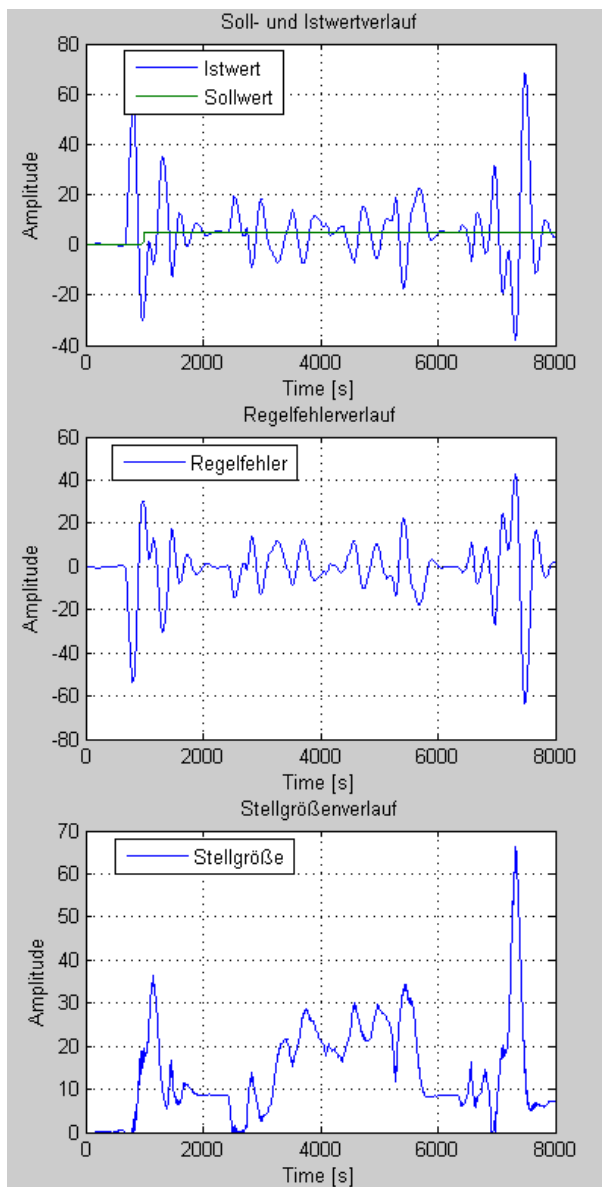
Verwendeter Typ: Ident_Data

Verwendeter Datensatz: Daten1_null

Verstärkung: 1

Zeitverzögerung [s]: 0

Offset: 0



8.2 Abbildungsverzeichnis

Abbildung 1.1: Ablauf der Entwurfsaufgabe mit PIDexpert	9
Abbildung 2.1: Benutzeroberfläche Parameterident aus [5]	11
Abbildung 2.2: Eingangsgrößen für die Identifikation	12
Abbildung 2.3: Ausgangsgröße für die Identifikation.....	13
Abbildung 2.4: Ablauf einer Identifikationsaufgabe entnommen aus [5]	14
Abbildung 2.5: MISO-System aus [5].....	15
Abbildung 2.6: Prinzipieller Aufbau des Regelkreises	15
Abbildung 2.7: Darstellungsformen eines PID-Reglers aus [9]	16
Abbildung 2.8: Sprungantwort eines PID-Reglers aus [9]	17
Abbildung 2.9: Idealer und realer Differenzierer aus [9]	17
Abbildung 2.10: Problematik der traditionellen PID-Struktur.....	19
Abbildung 2.11: PI-D Struktur	20
Abbildung 2.12: Verläufe der PI-D Struktur.....	20
Abbildung 2.13: I-PD Struktur	21
Abbildung 2.14: Verläufe bei I-PD Struktur	21
Abbildung 2.15: Stellgrößenbeschränkung bei Regelkreisen aus [9]	22
Abbildung 2.16: Verschlechterung der Regelgüte durch Wind-up Effekt.....	23
Abbildung 2.17: Strukturbild der Anti-Wind-up Maßnahme	24
Abbildung 2.18: Verbesserung der Regelgüte durch Anti-Wind-up	24
Abbildung 3.1: Bestimmung der Kenngrößen für die „open loop“ Methode	26
Abbildung 3.2: Regelkreis „closed loop“ Methode	27
Abbildung 3.3: Dauerschwingen bei der „closed loop“ Methode	27
Abbildung 3.4: Ermittlung der Kenngrößen „closed loop“ Methode aus dem Bode-Diagramm	28
Abbildung 3.5: Simulationsergebnis mit Ziegler Nichols „open loop“	30
Abbildung 3.6: Simulationsergebnis mit Ziegler Nichols „closed loop“	31
Abbildung 3.7: Ermittlung der Kenngrößen für Chien Hrones Reswick	32
Abbildung 3.8: Simulationsergebnis mit Chien Hrones Reswick.....	33
Abbildung 3.9: Darstellung des Stabilitätsrandes für $L(j\omega)$ entnommen aus [1].	35
Abbildung 3.10: Simulationsergebnis mit dem k150-Verfahren.....	38
Abbildung 3.11: Nyquist-Diagramm des offenen Kreises $L(j\omega)$ mit den Entwurfsspezifikationen für „loop-shaping“	40
Abbildung 3.12: Simulationsergebnis mit dem gKYP-Verfahren	44
Abbildung 3.13: Ablauf Minimierung von Integralkriterien.....	47
Abbildung 3.14: Simulationsergebnis mit Minimierung der Integralkriterien	49
Abbildung 3.15: Vergleich der Bode-Diagramme der gegebenen und approximierten Streckenübertragungsfunktion.....	52
Abbildung 3.16: Simulationsergebnis mit Improved Performance.....	54
Abbildung 4.1: Struktur bei Auswahl der Regelstrecke	57
Abbildung 4.2: Ablauf bei Auswahl und Konfiguration des Entwurfsverfahrens	58
Abbildung 4.3: Struktur bei der Simulation des Reglers	60
Abbildung 4.4: Struktur bei Simulationen mit Störgrößen	61
Abbildung 4.5: Fenster zur Bearbeitung der Projektdaten.....	66
Abbildung 4.6: Fehlermeldung bei Bearbeitung der Projektdaten	66
Abbildung 4.7: Koppelplan des Regelkreises unter Simulink (ZK).....	67

Abbildung 4.8: Maske des PID-Reglers	68
Abbildung 4.9: Koppelplan für die Umschaltung der Regelstrukturen	68
Abbildung 4.10: Koppelplan des zeitkontinuierlichen PID-Reglers.....	69
Abbildung 5.1: Benutzeroberfläche PIDexpert.....	70
Abbildung 5.2: Hinzufügen eines MATLAB Datenpfades	71
Abbildung 5.3: Auswahl Regelstrecke	72
Abbildung 5.4: Sprungantwort der markierten Strecke	72
Abbildung 5.5: Informationen über die markierte Strecke	73
Abbildung 5.6: Auswahl und Konfiguration des Entwurfsverfahrens	74
Abbildung 5.7: Speichern des erhaltenen Reglers	74
Abbildung 5.8: Einstellung des Entwurfsverfahrens	75
Abbildung 5.9: Spezifische Einstellungen bei Ziegler Nichols	75
Abbildung 5.10: Spezifische Einstellungen bei Chien Hrones Reswick.....	76
Abbildung 5.11: Spezifische Einstellungen bei gKYP Lemma.....	76
Abbildung 5.12: Spezifische Einstellungen bei Minimierung von Integralkriterien	77
Abbildung 5.13: Simulation des erhaltenen Reglers	78
Abbildung 5.14: Informationen über gespeicherte Regler	79
Abbildung 5.15: Variation der Reglerparameter	81
Abbildung 5.16: Simulationsergebnis nach Variation der Parameter	81
Abbildung 5.17: Variation der Parameter mittels Slider	82
Abbildung 5.18: Simulation mit Störgrößen.....	83
Abbildung 5.19: Neu/Speichern/Laden PIDexpert	85
Abbildung 5.20: Projektexport PIDexpert	86
Abbildung 5.21: Auswahl der vorhandenen Regler für den Export.....	86
Abbildung 6.1: Prinzip einer Dampfkraftanlage aus [5]	88
Abbildung 6.2: Sprungantwort der Regelstrecke	89
Abbildung 6.3: Verläufe der Entwurfsergebnisse des Praxisbeispiels.....	90
Abbildung 6.4: Istwert- und Stellgrößenverläufe mit den verschiedenen Regelstrukturen	92
Abbildung 6.5: Vergleich mit der Regelstrecke aus Modell_REK	93
Abbildung 6.6: Verlauf der selbstdefinierten Eingangsgrößen	94
Abbildung 6.7: Verläufe der einzelnen Störgrößen	95
Abbildung 6.8: Verlauf des Soll- und Istwertes des Niveaus des Kesselwasserstandes.....	95
Abbildung 6.9: Beispiel für Verläufe der Prozessdaten mit berücksichtigten Arbeitspunkt.....	96
Abbildung 6.10: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten1_null	97
Abbildung 6.11: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten2_null.....	98
Abbildung 6.12: Verläufe der Simulationsergebnisse mit Eingangsgrößen Daten3_null.....	99

8.3 Tabellenverzeichnis

Tabelle 1: Einstellregeln Ziegler Nichols „open loop“.....	26
Tabelle 2: Einstellregeln Ziegler Nichols „closed loop“.....	28
Tabelle 3: Reglerparameter Testbeispiel Ziegler Nichols „open loop“	30
Tabelle 4: Reglerparameter Testbeispiel Ziegler Nichols „closed loop“	31
Tabelle 5: Einstellregeln Chien Hrones Reswick.....	32
Tabelle 6: Reglerparameter Testbeispiel Chien Hrones Reswick.....	33
Tabelle 7: Reglerparameter Testbeispiel k150	38
Tabelle 8: Reglerparameter Testbeispiel gKYP-Lemma.....	44
Tabelle 9: Reglerparameter Testbeispiel Minimierung von Integralkriterien	49
Tabelle 10: Reglerparameter Testbeispiel Improved Performance	54
Tabelle 11: Reglerparameter aus den Entwurfsverfahren für das Praxisbeispiel	91
Tabelle 12: Reglerparameter des gewählten Reglers	91
Tabelle 13: Störgrößenkonfiguration für die selbstdefinierten Eingangsgrößen..	94
Tabelle 14: Störgrößenkonfiguration für Daten1_null	97
Tabelle 15: Störgrößenkonfiguration für Daten2_null	98
Tabelle 16: Störgrößenkonfiguration für Daten3_null	99

8.4 Literaturverzeichnis

- [1] BRIGITTE KRISTIANSSON und BENGT LENNARTSON, „Robust Tuning of PI und PID Controllers“, IEEE Control Systems Magazine 2006, S. 55-69.
- [2] SHINJI HARA, TETSUYA IWASAKI und DAISUKE SHIOKATA, „Robust PID Control Using Generalized KYP Synthesis“, IEEE Control System Magazine 2006, S. 80-91.
- [3] QING-GUO WANG, TONG-HENG LEE, HO-WANG FUNG, QIANG BI und YU ZHANG, „PID Tuning for Improved Performance“, IEEE Transactions on Control Systems Technology 1999, S. 457-456.
- [4] MARTIN HORN und NICOLAOS DOURDOUMAS, „Regelungstechnik, Rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise“, Pearson 2003.
- [5] MANFRED MARCHL: „Entwicklung einer Software-Umgebung für die Parameteridentifikation industrieller Prozesse“, Diplomarbeit, Institut für Regelungstechnik, Universität Graz 2003.
- [6] MARKUS KRANABETTER: „Ermittlung von Parametern komplexer Regelstrecken“, Diplomarbeit, Fachhochschule Technikum Wien 2008.
- [7] DATTA ANIRUDDHA, HO MING-TZU und BHATTACHARYYA SHANKAR, „Structure and synthesis of PID controllers“ Springer Verlag 2000.
- [8] SERGE ZACHER und MANFRED REUTER, „Regelungstechnik für Ingenieure“, Vieweg Verlag 2002.
- [9] JAN LUNZE, „Regelungstechnik 1 und 2“, Springer Verlag 2008.
- [10] OLIVER NELLES, „Mess- und Regelungstechnik 1 und 2“, Universität Siegen 2008.
- [11] ANTON BRAUN, „Kontinuierliche und diskrete Systeme“, Carl Hanser Verlag 2005.
- [12] JOHN DOYLE, BRUCE FRANCIS und ALLEN TANNENBAUM, „Feedback Control Theory“, Macmillan Verlag 1990.
- [13] KARL JOHAN AESTRÖM und TORE HÄGGLUND, „PID Controllers: Theory, Design and Tuning“, ISA 1994.
- [14] ZHANG LIU und LIEVEN VANDENBERGHE, „gKYPSPDP User Guide“, University of California 2004.
- [15] ADOLF HERMANN GLATTFELDER und WERNER SCHAUFELBERGER, „Control Systems with Input and Output Constraints“, Springer Verlag 2003.