



Diplomarbeit

Optimierung der Regelkreiseinstellungen bei Wasserturbinen

Markus Petschauer

Institut für Regelungs- und Automatisierungstechnik in Zusammenarbeit mit der
Firma Andritz AG

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 entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....

(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....

.....

date

(signature)

Danksagung

Diese Seite ist all jenen gewidmet, die mich während meiner langen Ausbildungs- bzw. Studienzeit unterstützt und begleitet haben.

Danke

Meinem Betreuer, Herrn Univ. Prof. Dipl.-Ing. Dr. techn. Anton Hofer, der für die Betreuung meiner Diplomarbeit seitens der TU-Graz zuständig war

Herrn Ing. Armin Martinz für die jahrelange gute Zusammenarbeit in der Andritz AG während meines gesamten Studiums, sowie für die Betreuung der Diplomarbeit seitens der Andritz AG

Meinen Eltern, Angelika und Franz, für die Unterstützung mit all ihren möglichen Mitteln während meiner gesamten Ausbildung

Meiner Frau Bettina, die aufgrund meiner langen Studienzeit sehr viel Geduld aufbringen mußte und mich auch immer aufs Neue motivieren konnte

Meinen Freunden, ob Privat, UNI oder Firma, die mich alle auf ihre eigene Art motivieren und vorantreiben konnten.

Markus Petschauer

Lannach, im Dezember 2009

Kurzfassung

Seit Beginn der 90iger Jahre werden digitale Turbinenregler in der Regelung von Wasserkraftwerken eingesetzt. Mehr als 90% dieser Regler besitzen eine PI- bzw. PID-Struktur. Im Rahmen der Diplomarbeit wurde ein Verfahren gesucht, das die Reglerparameter der PID-Regler des Turbinenreglers verbessert. Zur technischen Realisierung des Verfahrens wurde neben der Leitsystemsoftware APROL die Software Matlab (Simulink) verwendet. Für die Überprüfung der verschiedenen Optimierungsverfahren war es notwendig, ein bestehendes Simulationsmodell einer Wasserkraftanlage der Andritz AG in Matlab zu implementieren und mit den Daten einer realen Anlage zu parametrieren.

Nach einem Vergleich mehrerer möglicher Optimierungsverfahren, fiel die Wahl auf das Optimierungsverfahren *EXTREMUM SEEKING*. Dieses Verfahren bietet den Vorteil einer modellfreien Optimierung (Online-Optimierung).

Bei der Durchführung der Optimierung werden Sollwertsprünge im laufenden Betrieb des Regelkreises ausgeführt. Während des Einschwingvorganges wird der Wert einer Zielfunktion online ermittelt (z.B. über das Integral des quadratischen Fehlers - ISE). Ziel des Optimierungsverfahrens ist es, diesen Wert zu verkleinern bis ein gewünschtes Ergebnis vorliegt. Der Algorithmus benutzt den Wert der Zielfunktion mit den aktuell verwendeten Reglereinstellungen und liefert neue Reglerparameter, die bei dem nächsten Sollwertsprung verwendet werden.

Die notwendige Funktionsüberprüfung des Algorithmus erfolgte in Matlab Simulink. Abschließend wurde der Algorithmus inklusive einer Bedienoberfläche für den industriellen Einsatz in die Turbinenreglersoftware (APROL) implementiert.

Abstract

Since the beginning of the 90's, digital turbine controllers are used for the control in hydro power plants. More than 90% of these controllers have PI-or PID structure. The scope of the thesis was to search a method which improves the parameters of the PID controllers. For the technical realization the software Matlab (Simulink) was used beside the control system software APROL. To check different optimization methods it was necessary to implement an existing simulation model (Andritz AG model) in Matlab and to parameterize this model with data of a real hydro power plant.

After comparing several possible optimization methods, the optimization method *EXTREMUM SEEKING* was chosen. This method offers the advantage of an online, model-free optimization.

During the optimization step-response experiments are running online. At the end of each step-response experiment, a cost function (for example an integrated square error - ISE) is calculated. The aim of extremum seeking is to minimize the value of the cost function until a desired result is obtained. The ES-algorithm uses the result of the cost function and the parameters of the controller to generate new controller parameters. These new parameters are used for the next step-response experiment.

The necessary verification of the algorithm has been performed in Matlab Simulink. Finally the algorithm was implemented in an industrial application within the turbine controller software (APROL) together with a graphical user interface.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	5
1.2	Verwendete Werkzeuge	6
2	Modell	7
2.1	Aufbau des Turbinenmodells und Parametrierung	7
2.1.1	Ein-/ Ausgänge des Turbinenmodells	7
2.1.2	Modellabschnitt für Ablenker und Düsen	10
2.1.3	Modellabschnitt Durchflussberechnung	12
2.1.4	Modellabschnitt Momentenberechnung	14
2.1.5	Modellabschnitt Turbine und Generator	18
2.1.6	Modellabschnitt Stauraum	20
2.2	Vergleich Modell- und Anlagendaten	23
3	Regler	27
3.1	PID-Regler	28
3.1.1	Einführung	28
3.1.2	PID-Einstellregeln	31
3.1.3	Vergleich der ermittelten Reglerkoeffizienten	34
3.2	Extremum Seeking	36
3.2.1	Prinzipielle Funktionsweise	37
3.2.2	Der extremum seeking Algorithmus	40
3.2.3	Anwendung des ES-Algorithmus auf die Strecke 3.6	45
4	Implementierung	47
4.1	Implementierung in Simulink	47
4.1.1	PID - Implementierung in Simulink	47
4.1.2	ES - Implementierung in Simulink	48
4.1.3	Funktionsprüfung im Simulink	51
4.2	ES - Implementierung in APROL	54
4.2.1	Funktionsprüfung im APROL	63
A	Parametrierung des APROL Turbinenmodells	67
A.1	Parameter für Ablenker und Düsen	67
A.2	Parameter für die Durchflussberechnung (q)	70

A.3	Parameter für die Momentenberechnung (m)	71
A.4	Parameter für das Modell der Turbine und des Generators	73
A.5	Parameter für das Modell Pegel und Fallhöhe	74
B	Regler - Code und Einstellungen	77
B.1	Extremum Seeking realisiert im Matlab	77
B.1.1	Simulinkmodell der Strecke und des Reglers	77
B.1.2	Matlabcode Extremum seeking	78
C	Detailbeschreibungen zur Software	81
C.1	Bausteinbeschreibungen	81
C.1.1	PID-Regler im Simulink	81
C.1.2	PID-Regler im APROL	82
	Literaturverzeichnis	89

Kapitel 1

Einleitung

Wasserkraftwerke spielen in Österreich im Bereich der Energieerzeugung eine tragende Rolle. Da es sich bei Wasser um einen erneuerbaren Energieträger handelt, der zwar genutzt, aber nicht verbraucht wird, sind die Vorteile im Bereich Umweltverträglichkeit für die Zukunft offensichtlich.

Kraftwerkstypen:

Laufkraftwerke wandeln die Kraft des fließenden Wassers in elektrische Energie um. Die Fallhöhe, die für die Stromerzeugung eine bedeutende Rolle spielt, ist bei diesen Kraftwerkstypen im Gegensatz zu den Speicherkraftwerken relativ gering. Diese Art der Kraftwerke dient der Grundlastversorgung (siehe Abbildung 1.1).

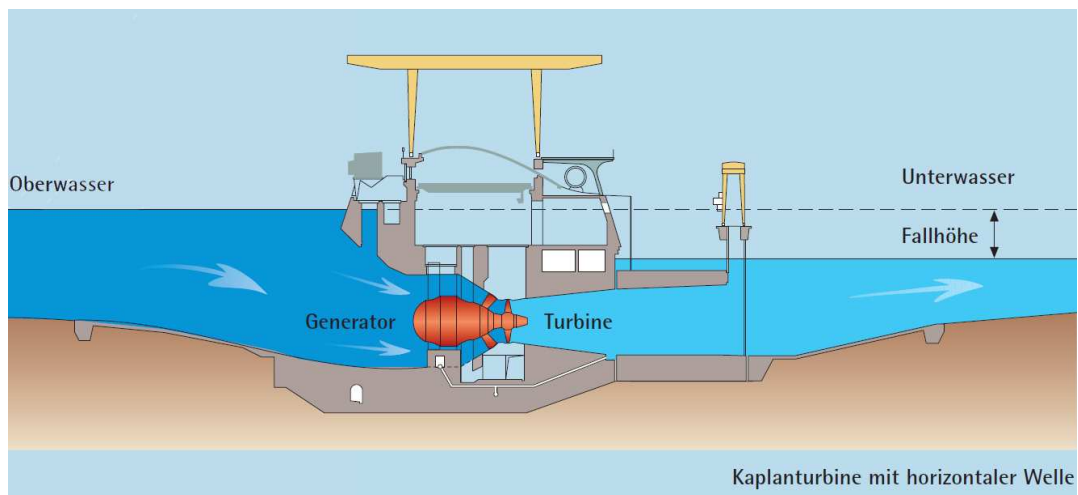


Abbildung 1.1: Übersicht eines Laufkraftwerks mit einer Kaplanmaschine (aus [1])

Speicherkraftwerke bestehen im Groben aus einem Stausee, einem Druckstollen (der die Verbindung zwischen Stausee und Kraftwerk herstellt) und dem Kraftwerk. Durch die große Fallhöhe können hohe Leistungen im Vergleich zu den Laufkraftwerken erzielt

werden. Diese Art von Kraftwerken wird zur Abdeckung von Spitzenlasten verwendet (siehe Abbildung 1.2).

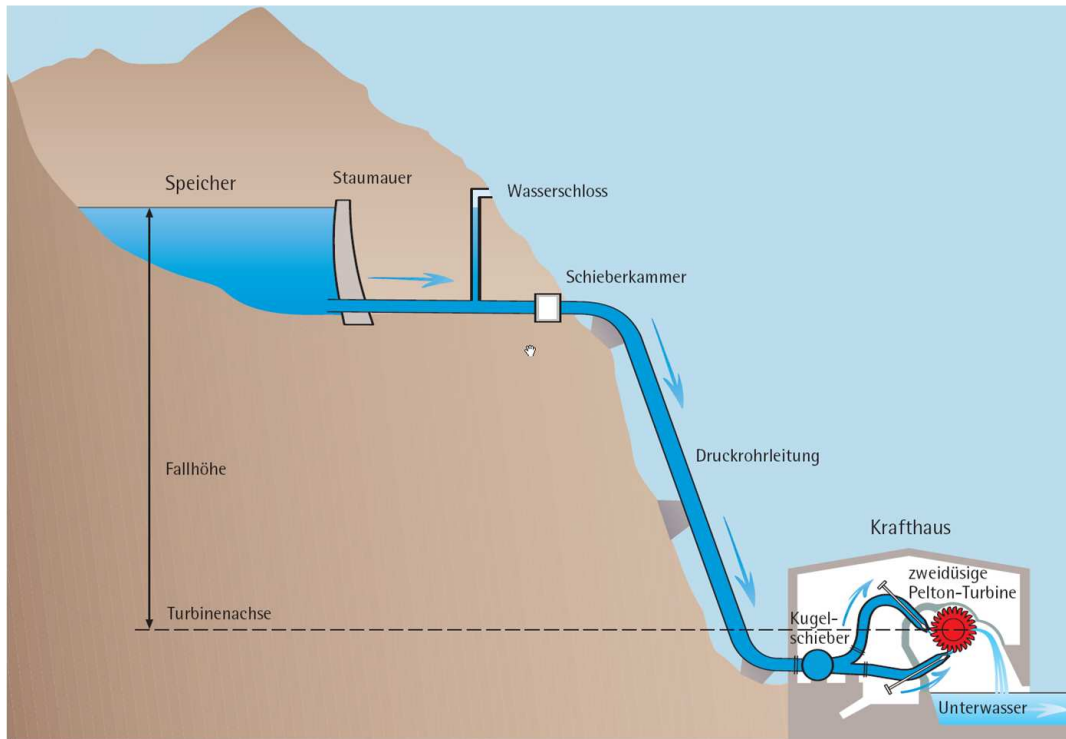


Abbildung 1.2: Übersicht eines Speicherkraftwerks mit einer Pelton-turbine (aus [1])

Turbinentypen:

Kaplan-Turbine besteht aus einem Leitapparat (reguliert die Wasserzufuhr zum Laufrad) und einem Laufrad (Ähnlichkeiten mit einer Schiffschraube). Beide Organe sind verstellbar und werden über den Turbinenregler geregelt. Der Einsatzbereich dieser Turbinen ist bei geringen Fallhöhen und großen Wassermengen (Laufkraftwerken) gegeben.

Francis-Turbine besteht ebenfalls aus einem Leitapparat und einem Laufrad. Im Gegensatz zur Kaplanturbine sind hier die Schaufeln des Laufrades nicht verstellbar. Das Wasser wird über einen spiralförmigen Einlauf durch die verstellbaren Schaufeln des Leitapparates auf die Schaufeln des Laufrades gedrückt. Der Einsatzbereich liegt bei mittleren Fallhöhen und konstanten Wassermengen (Speicherkraftwerken).

Pelton-Turbine besteht aus einer oder mehreren Hochdruckdüsen, die das Wasser auf ein Laufrad mit becherförmigen Schaufeln übertragen. Der Durchfluss wird über fein regulierbare Düsen bzw. über einen Ablenker (Abschneider) geregelt (Eine schematische Darstellung eines Kraftwerkes mit einer Pelton-turbine ist in Abbildung 1.2 angegeben). Der Einsatzbereich für Pelton-Turbinen ist bei großen Fallhöhen und einer geringen

Wassermenge (Speicherkraftwerke).

Unter einem **digitalen Turbinenregler** eines Wasserkraftwerkes versteht man die elektrische Komponente, die für die korrekte Abarbeitung der einzelnen mechanischen Komponenten pro Maschineneinheit benötigt wird. Zu einer Maschineneinheit zählen primär die Turbine, der Generator, die mechanischen und elektrischen Schutzfunktionen sowie die hydraulischen und elektrischen Regelorgane. Seit Beginn der 90iger Jahre werden digitale Turbinenregler anstatt der zuvor verwendeten Analogtechnik verwendet. In den meisten Fällen besteht der Softwarekern dieser digitalen Turbinenregler aus einer PI- bzw. PID Regelstruktur. Diese Struktur hat den Vorteil, dass sie für eine Vielzahl von Regelstrecken eingesetzt werden kann.

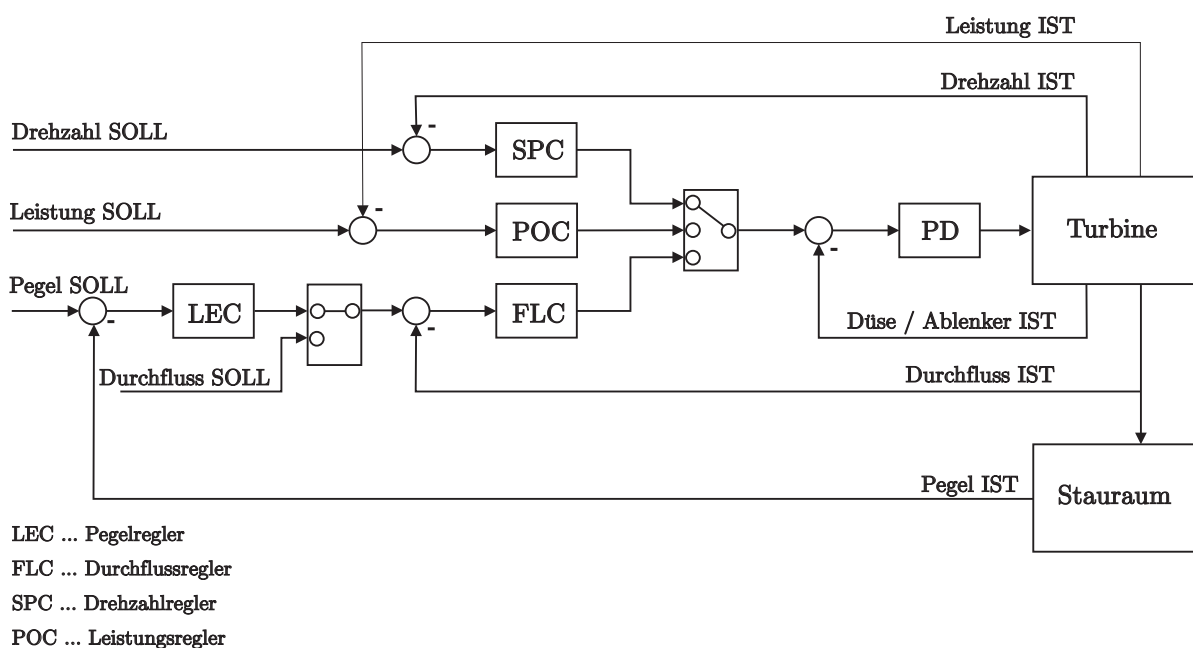


Abbildung 1.3: Turbinenreglerkonzept Andritz AG

Abbildung 1.3 zeigt eine typische Realisierung eines digitalen Turbinenreglers. Der Turbinenregler als Einheit unterteilt sich weiter in:

- SPC ... Drehzahlregler
- POC ... Leistungsregler
- FLC ... Durchflussregler
- LEC ... Pegelregler
- PD ... Proportional-Differential Regler → wird zur Positionierung der einzelnen Regelorgane verwendet

Regelorgane des Turbinenreglers: Im Fall einer Pelton-Turbine erfolgt die Regelung über die Düsenöffnung und die Position des (/der) Ablenker, bei einer Kaplan-Turbine über die Verstellung des Leitapparates und des Laufrades und bei einer Francisturbine mit dem Leitapparat.

Drehzahlregler: Wie der Name schon sagt, dient der Drehzahlregler zum Regeln der Drehzahl. Der Sollwert ist hier eine Drehzahl in %. Dieser Regler regelt den Hochlauf der Maschine bis zur Synchrondrehzahl (Drehzahl, bei der der Generator mit dem Netz synchronisiert werden kann). Weiters wird der Drehzahlregler bei Lastabwürfen (Leistungsschalter AUS während des Betriebes) verwendet, um die Maschine wieder auf Synchrondrehzahl zu stabilisieren.

Leistungsregler: Der Leistungsregler ist eine mögliche Variante der Regelung im stromerzeugendem Betrieb. Der Sollwert wird je nach Maschinengröße in kW bzw. MW an den Regler übergeben. Diese Betriebsart wird verwendet, wenn eine konstante Leistungsabgabe erwünscht ist. Voraussetzung für den Leistungsreglerbetrieb ist eine synchronisierte Maschine mit eingeschaltetem Leistungsschalter (LS = EIN).

Durchflussregler: Der Durchflussregler regelt, wie schon im Namen enthalten, den Durchfluss der Maschine. Der Sollwert wird in m^3/s an den Regler übergeben. Diese Variante der Regelung wird bei der Pegelregelung als untergeordneter Regler je Maschine verwendet.

Pegelregler: Der Pegelregler ist für das Halten eines gewünschten Niveaus (Pegel) im Stauraum verantwortlich. Die Sollwertvorgabe erfolgt in mM (Meter über Meer). Je nach Typ des Kraftwerkes und der Anzahl der regelbaren Organe werden vom Pegelregler Abflusssollwerte an die untergeordneten Durchflussregler (Abflussregler) weitergegeben.

1.1 Aufgabenstellung

Während der Inbetriebsetzungsphase eines Wasserkraftwerks werden nach den Signal- und Schutzüberprüfungen, Einstellungen und Optimierungen des vorhandenen Regelkreises durchgeführt. Die Inbetriebsetzungstätigkeiten müssen in Berichten dokumentiert werden. Diese Unterlagen dienen auch als Basis für spätere Vergleichsversuche im Zuge einer Revision eines Kraftwerkes.

Hardwarekonfiguration

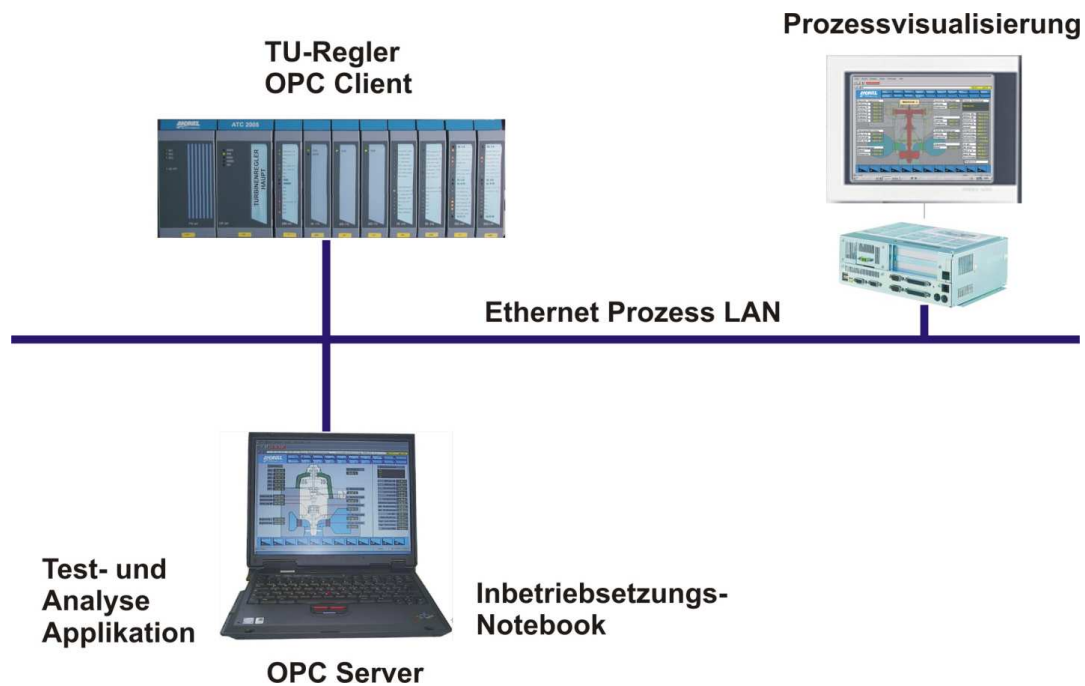


Abbildung 1.4: Hardwarekonzept

Abbildung 1.4 zeigt eine typische Hardwarekonfiguration eines Turbinenreglers. Diese besteht aus einer Steuerungseinheit (im Bild dargestellt durch den TU-Regler), einer Prozessvisualisierung und eines für die Inbetriebsetzung benötigten PC's (meist ein Notebook). Die Kommunikation zwischen den einzelnen Einheiten erfolgt über das sogenannte Prozess LAN.

Test- und Analyseprinzip

- Auswahl eines Regelkreises aus dem Turbinenregler Gesamtkonzept
- Vorgabe bzw. Ausführung eines Sollwertsprunges
- Prozessdatenerfassung
- Ermittlung der Reglerparameter für div. Verfahren für Regelkreisoptimierung

- Optimieren des Verhaltens der Regelkreise (Ein geeignetes Verfahren ist im Zuge der Diplomarbeit zu ermitteln)
- Verstellen der Parameter des Turbinenreglers
- Wiederholung des Sollwertsprunges zur Kontrolle

Grafische Darstellung diverser Prozessvorgänge für die AS-Build Dokumentation

- Trocken- und Nassindizierungen
- Abschaltversuche
- Inselversuche
- Schutzauslösungen

Werkzeuge

- APROL (Prozessleitsystem der Firma B&R)
- Matlab
- Simulink

1.2 Verwendete Werkzeuge

APROL [2]

APROL ist das Programmierwerkzeug der Firma Bernecker und Rainer (B&R), das zur Erstellung von Prozessleitsystemen verwendet wird. Beginnend mit der Hardwarekonfiguration einer Anlage wird dieses Werkzeug bis zur Fertigstellung der Software für die Controller, der Bedienebene sowie der Protokollierung verwendet.

Matlab [3]

Matlab ist eine Software zur Lösung von mathematischen Aufgabenstellungen mit der Möglichkeit einer grafischen Darstellung der Ergebnisse.

Simulink [4]

Simulink ist ein Zusatzprodukt von Matlab und dient der Modellierung und Simulation von Systemen.

Kapitel 2

Modell

Für alle Tests (sowohl auf der APROL-Ebene als auch auf der Matlab-Ebene) wurde eine leicht modifizierte Variante des Turbinenmodells der Firma Andritz AG verwendet [7]. Dieses hat sich in der Praxis über jahrelange Erfahrungswerte bei diversen Anlagen bewährt. Der einzige Nachteil an diesem Modell ist das Fehlen einer entsprechenden Dokumentation, wie eine richtige Parametrierung durch den Techniker zu erfolgen hat. Für einen Funktionstest der Software war dies bisher nicht notwendig. Sehr wohl macht eine korrekte Parametrierung des Modells Sinn im Bezug auf eine mögliche Zeitersparnis bei der Inbetriebnahme, da die Parameter für die einzelnen Regler entsprechend versorgt werden können und nur eine Feinabstimmung auf der Anlage von Nöten ist. Das Ermitteln der Parameter sowie eine Einführung in das Turbinenmodell wird in diesem Abschnitt behandelt.

2.1 Aufbau des Turbinenmodells und Parametrierung

Da die Anlage für die Diplomarbeit eine Pelton-turbine enthält, wird hier das Pelton-Modell behandelt.

Die Struktur des Pelton-Modells ist von der Düsenzahl abhängig. Bei der Turbine in der Anlage handelt es sich um eine 2-düsige Pelton-turbine und diese wird im folgenden beschrieben.

2.1.1 Ein-/ Ausgänge des Turbinenmodells

Für die Verwendung des Turbinenmodells in der Reglersoftware werden nur die unten angeführten Ein-/ Ausgänge benötigt.

Eingänge

- Signal Servoventil Ablenker (Abschneider) (SV_AB)
- Signal Servoventil Düse 1 (SV_N1)
- Signal Servoventil Düse 2 (SV_N2)

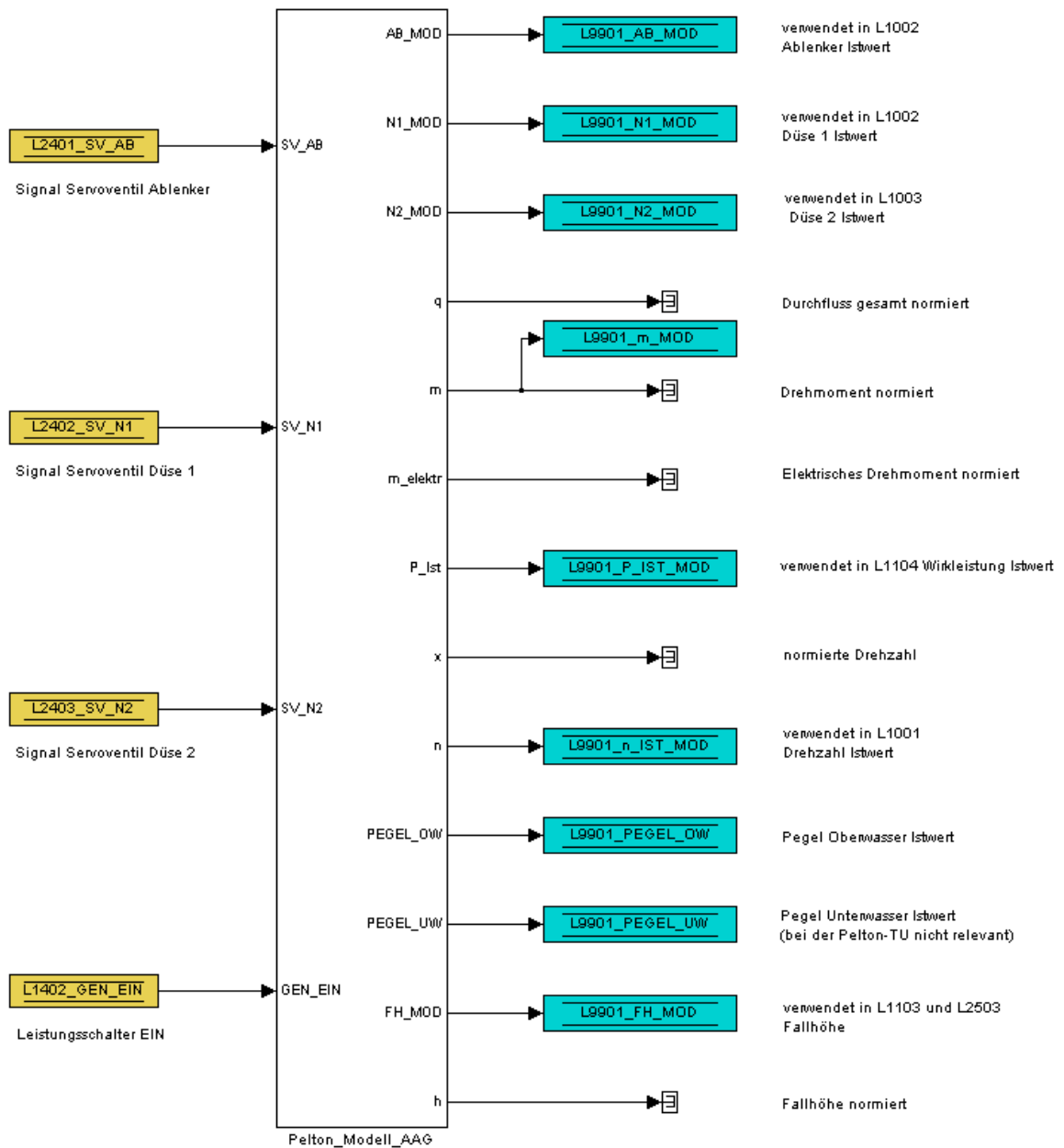


Abbildung 2.1: Pelton Modell der Andritz AG realisiert in Simulink

- Leistungsschalter EIN

Abbildung 2.1 zeigt das Pelton-Modell der Andritz AG realisiert in Simulink. Je nach Bedarf können die abgebildeten Ausgänge entsprechend in der Turbinenreglersoftware verwendet werden.

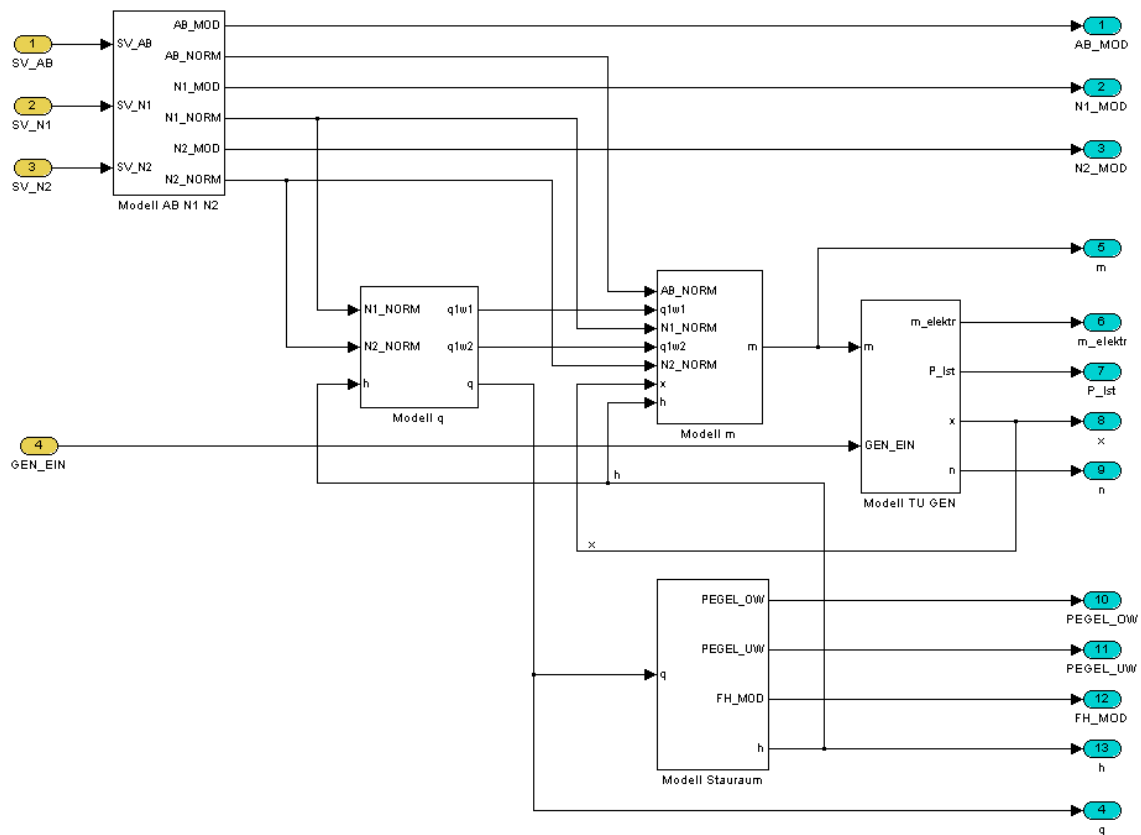


Abbildung 2.2: Pelton Modell der Andritz AG

Abbildung 2.2 zeigt die Hauptfunktionen (Submodelle), die für die korrekte Funktionsweise des Modells benötigt werden.

Diese sind:

- Modell für Ablenker und Düsen
- Modell Durchflussberechnung
- Modell Momentenberechnung
- Modell Turbine und Generator
- Modell Stauraum

und werden in den folgenden Abschnitten genauer beschrieben.

2.1.2 Modellabschnitt für Ablenker und Düsen

Mit den oben definierten Eingängen für den Ablenker und den beiden Düsen werden die Istwerte der Organe über die Gleichung 2.1 ermittelt.

Die aktuelle Ablenker- bzw. Düsenposition ergibt sich zu:

$$N_{x_MOD} = \int_{t_0=0}^t S_{v_Nx}(\tau) d\tau, \quad (2.1)$$

wobei für S_{v_Nx} gilt: $t_{Nx_zu} < S_{v_Nx} < t_{Nx_auf}$. t_{Nx_zu} ist die Schließzeit der Düse (des Ablenkers), t_{Nx_auf} die Öffnungszeit. Für N_{x_MOD} gilt: $0 < N_{x_MOD} < 100$.

Für die weitere Verwendung der Ablenker- und Düsenpositionen im Turbinenmodell werden diese, wie in Gleichung 2.2 gezeigt, normiert.

$$N_{x_NORM} = 0,01N_{x_MOD} + Offset \quad (2.2)$$

Der Offset dient einer nachträglichen Korrekturmöglichkeit.

Abbildung 2.3 zeigt eine Umsetzung der oben dargestellten Gleichungen im Simulink.

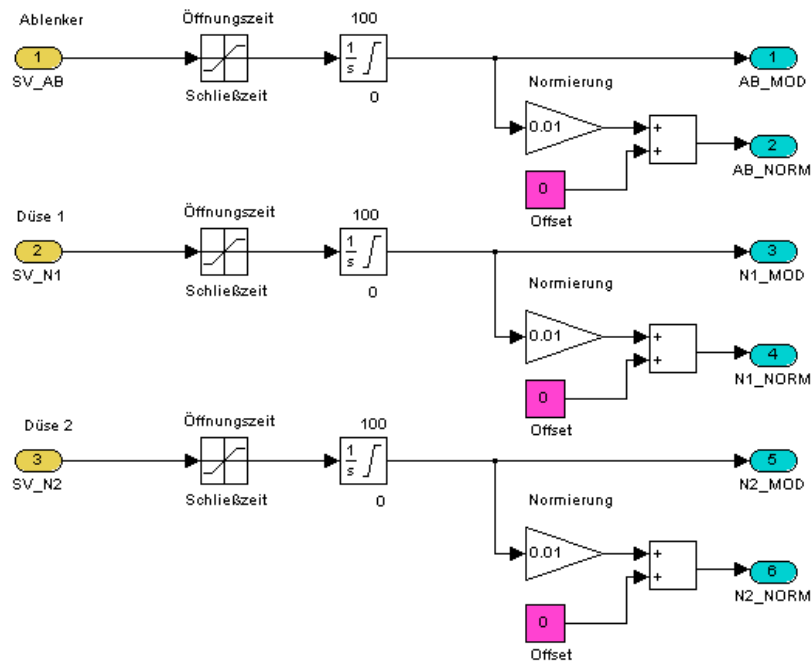


Abbildung 2.3: Modell für Ablenker und Düsenistwerte

Die Ausgänge aus Abbildung 2.3 AB_MOD , $N1_MOD$ und $N2_MOD$ werden in der Turbinenreglersoftware als IST-Werte der einzelnen Organe weiterverwendet. Die Werte AB_NORM , $N1_NORM$ sowie $N2_NORM$ dienen der Berechnung des Durchflusses sowie des Drehmoments der Turbine.

Parameter für Ablenker und Düsen

Die Parameter für das Turbinenmodell werden aus den Anlagenkenndaten bzw. aus den Inbetriebnahmekurven entsprechend ermittelt. Die Ablenkerzeiten im vorliegendem Fall ergeben sich zu:

$$t_{AB_Auf} = 3,9s$$

$$t_{AB_Zu} = 1,2s$$

Diese Verstellzeiten werden auf $\%/s$ normiert, was in weiterer Folge die entsprechenden Parameterwerte für das Modell (sowohl Simulink als auch APROL) liefert. Die Umrechnung erfolgt mit Hilfe der Formel

$$t_{\%/s} = \frac{100}{t_{AB_Auf}} \quad (2.3)$$

Damit ergibt sich der Parameterwert für

$$t_{AB_Auf} = 25,64$$

und der Wert für die Schließzeit des Ablenkers

$$t_{AB_Zu} = -83,33.$$

Für die Berechnung der Düsenzeiten gilt die gleiche Vorgehensweise. Da die beiden Düsen ähnliches Verhalten aufweisen, wurden für diese die gleichen Parameter verwendet.

Die Düsenzeiten ergeben sich zu:

$$t_{Nx_Auf} = 138s$$

$$t_{Nx_Zu} = 36s$$

Aus den Öffnungs- bzw. Schließzeiten der Düsen erhält man mit Hilfe der Formel 2.3 folgende Parameter:

$$t_{Nx_Auf} = 0,725$$

$$t_{N_x_zu} = 2,778$$

Die oben ermittelten Parameter entsprechen den Öffnungs- bzw. Schließzeiten der Begrenzungsblöcke aus Abbildung 2.3.

2.1.3 Modellabschnitt Durchflussberechnung

Mit Hilfe der oben ermittelten Werte für die Positionen der Düsen werden die normierten Durchflussbeiwerte der einzelnen Düsen sowie der Gesamtabfluss anhand den Gleichungen 2.4 und 2.5 bestimmt. Unter dem Durchflussbeiwert pro Düse versteht man die Durchflussmenge, die je Düse abgegeben wird.

$$q_{1wx} = (K_1 - K_2 N_{x_NORM}) N_{x_NORM} \quad (2.4)$$

Der Gesamtabfluss ergibt sich aus den beiden Beiwerten der Düsen (q_{1w1} und q_{1w2}) gewichtet mit dem Faktor $\frac{q_{1s}}{z_0}$ (wobei $q_{1s} = 0.5$ einem Korrekturwert aus der Erfahrung zahlreicher Anwendungen und z_0 der Düsenanzahl entspricht) und unter Berücksichtigung des Einflusses der Fallhöhe.

$$q = (q_{1w1} + q_{1w2}) \frac{q_{1s}}{z_0} \sqrt{h} \quad (2.5)$$

Abbildung 2.3 zeigt eine Umsetzung der oben dargestellten Gleichungen im Simulink.

Parameter für die Durchflussberechnung (q)

Für die Berechnung des Durchflusses wird in diesem Fall eine Polynomfunktion 2. Grades verwendet. Diese stellt den Zusammenhang von Durchfluss zu Düsenöffnung dar.

$$q_{1w1} = (K_1 - K_2 N_{1_NORM}) N_{1_NORM} \quad (2.6)$$

Aus Abbildung 2.5 ergeben sich folgende Werte für die Parameter K_1 und K_2 :

$$K_1 = 1,496$$

$$K_2 = 0,5$$

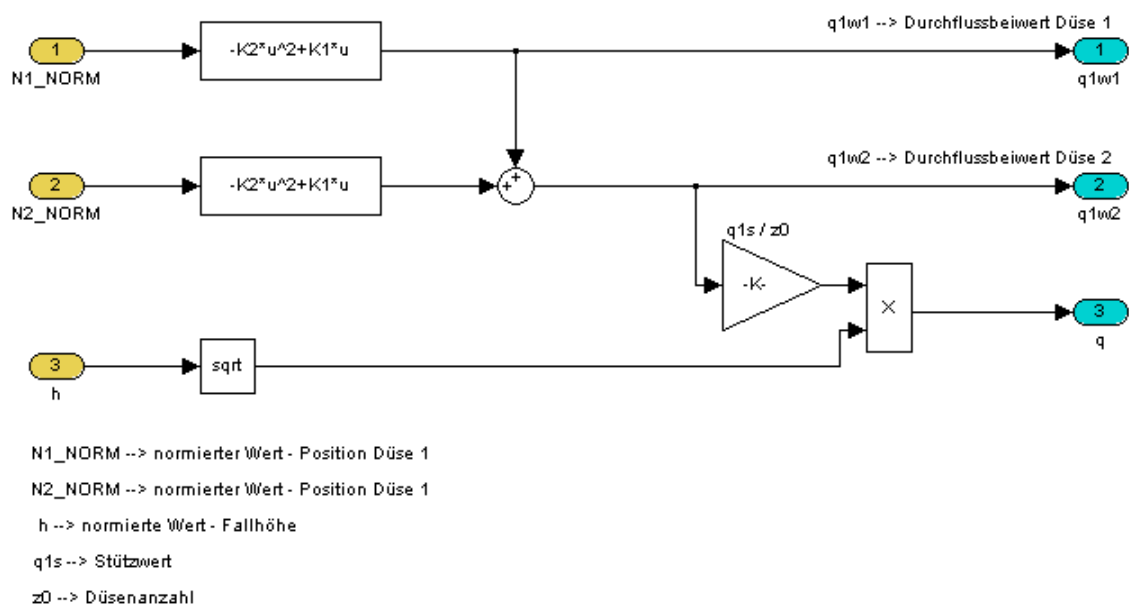


Abbildung 2.4: Modell für die Durchflussberechnung

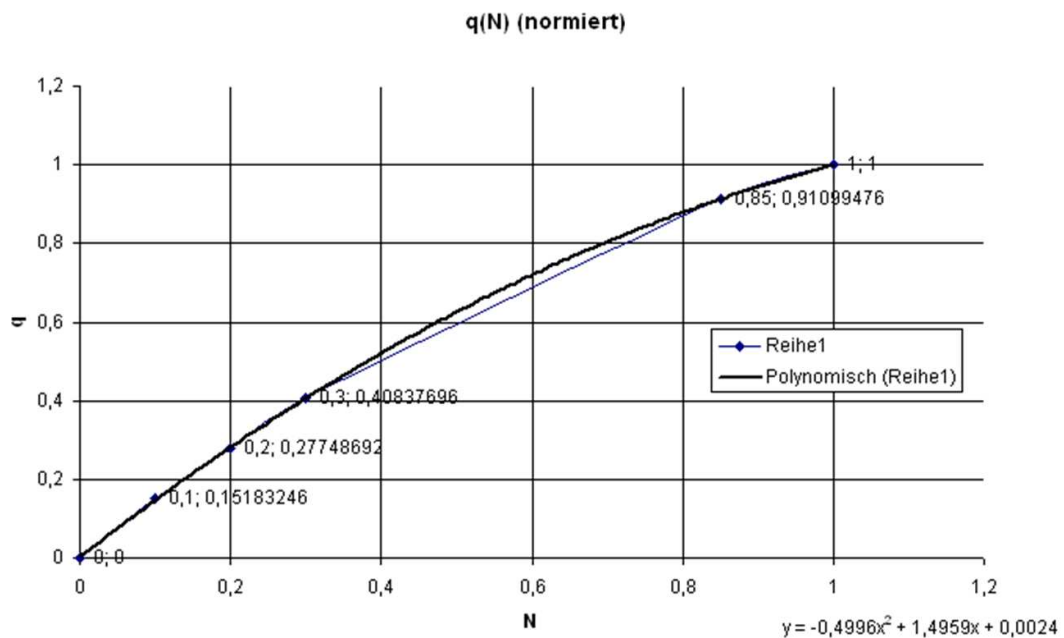


Abbildung 2.5: q (Düse) in Abhängigkeit der Düsenposition

2.1.4 Modellabschnitt Momentenberechnung

In die Berechnung des Drehmomentes fließen die oben ermittelten Größen des Ablenkers, der Düsen sowie die Durchflussbeiwerte, die relative Fallhöhe und die relative Drehzahl ein. Die mathematischen Zusammenhänge sind in den Gleichungen 2.7 bis 2.14 ersichtlich.

$$m_{1w1} = (K_1 - K_2 N_{1_NORM}) N_{1_NORM} \quad (2.7)$$

m_{1w1} ist der theoretische Einfluss je Düse auf das Drehmoment (ohne Einfluss der Drehzahl, des Ablenkers und des hydraulischen Bremsmoments). Als nächstes wird die normierte Position des Ablenkers e_s im Strahl ermittelt. Zur Ermittlung dienen die Parameter $a_{e0} = 0,85$, $a_{e1} = 0,056$ und $e_0 = 1,47$.

$$e_s = e_0 - (q_{1w1} (a_{e1} + a_{e0})) \left(e_0 - \frac{A_{B_NORM}}{\sqrt{q_{1w1}}} \right) \quad (2.8)$$

Für e_s gilt: $0,4 < e_s < 0,94$. Der Ausdruck $\sqrt{q_{1w1}}$ wird mit einer unteren Grenze von $0,02 < \sqrt{q_{1w1}}$ begrenzt. Über den Zusammenhang $f_m(e_s)$ (Abbildung 2.6 - Polynom 5. Grades) erhält man den Gewichtungsfaktor f_m für den jeweiligen Strahl (siehe Gleichung 2.9).

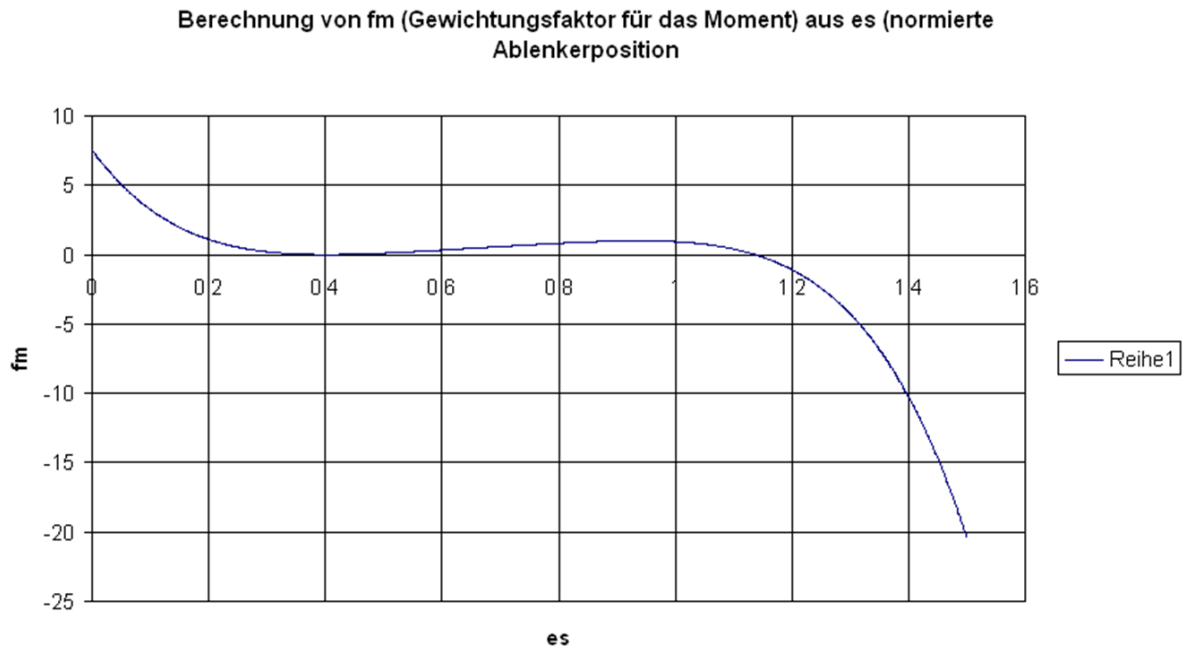


Abbildung 2.6: Zusammenhang $f_m(e_s)$

$$f_m = -51,151 e_s^5 + 172,32 e_s^4 - 237,86 e_s^3 + 167,69 e_s^2 - 57,619 e_s + 7,5526 \quad (2.9)$$

Mit Hilfe der Gleichung 2.10 erhält man den Beitrag / Düse für das Drehmoment (Momentenbeiwert).

$$m_{1kw1} = m_{1w1} f_m \quad (2.10)$$

Für die Berechnung des Momentenbeiwertes m_{1kw2} der zweiten Düse gilt die gleiche Vorgehensweise wie in Gleichung 2.10.

Die relative Fallhöhe h bzw. die relative Drehzahl x werden in Gleichung 2.11 berücksichtigt.

$$z_w = x \sqrt{h} \quad (2.11)$$

Den Drehzahleinflussfaktor m_z erhält man aus einer Polynomfunktion 5. Grades (siehe Abbildung 2.7 bzw. Gleichung 2.12).

$$m_z = 0,161 z_w^5 - 0,435 z_w^4 + 0,098 z_w^3 + 0,013 z_w^2 - 0,275 z_w + 1,4 \quad (2.12)$$

Die Summe der Momentenbeiträge der einzelnen Düsen wird mit dem Verhältnis aus $\frac{m_{1s}}{z_0}$ skaliert (wobei m_{1s} = Korrekturfaktor (Wahl bei 1) und z_0 die Düsenzahl ($z_0 = 2$) wiedergibt) (siehe Abbildung 2.10). Wahl von

$$\frac{m_{1s}}{z_0} = 0.5$$

Für das Planschmoment (es ist ein Bremsmoment, das aufgrund des Wasserstrahles gegen das Antriebsmoment der Turbinenwelle wirkt) hat sich folgende Faustformel als praktisch erwiesen:

$$m_{p1} = 0,03 z_w^2 \quad (2.13)$$

Daraus ergibt sich für das Drehmoment m :

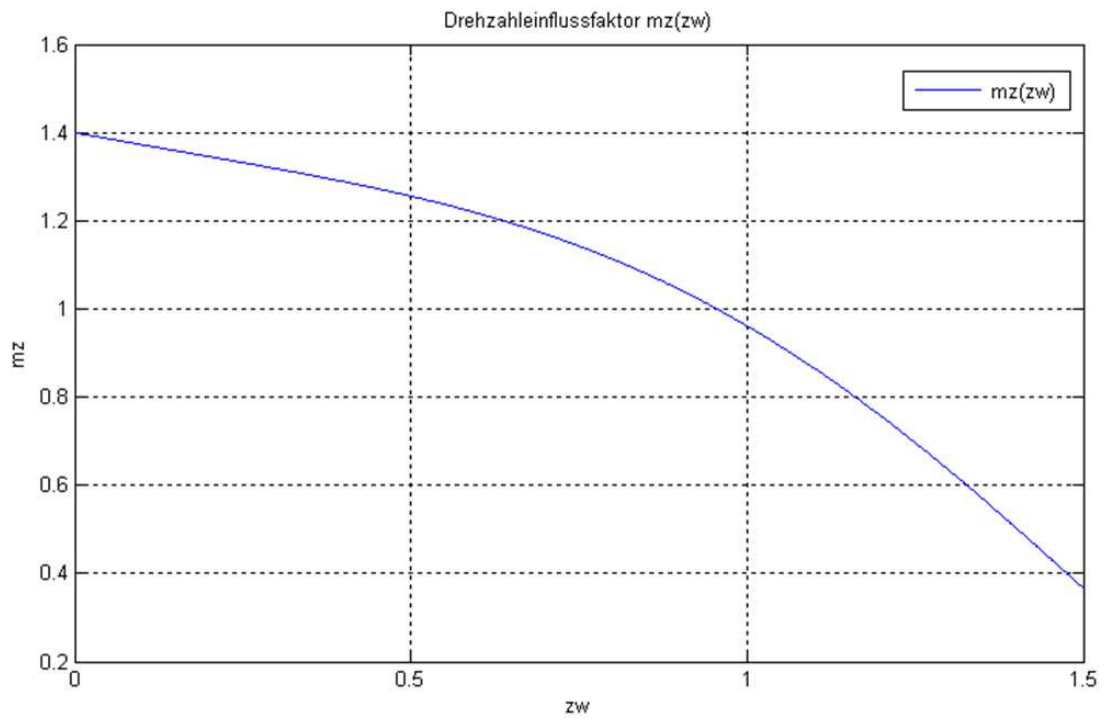


Abbildung 2.7: Drehzahleinflussfaktor $m_z(z_w)$

$$m = \left((m_{k1w1} + m_{k1w2}) \frac{m_{1s}}{z_0} m_z - m_{p1} \right) h. \quad (2.14)$$

Die Abbildungen 2.8, 2.9 sowie 2.10 zeigen die Umsetzung in Simulink.

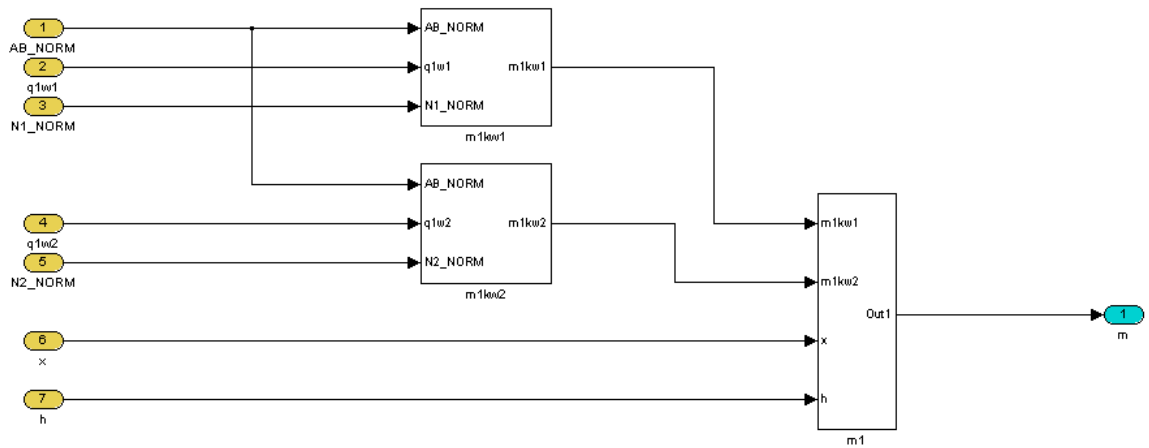


Abbildung 2.8: Modell für die Momentenberechnung

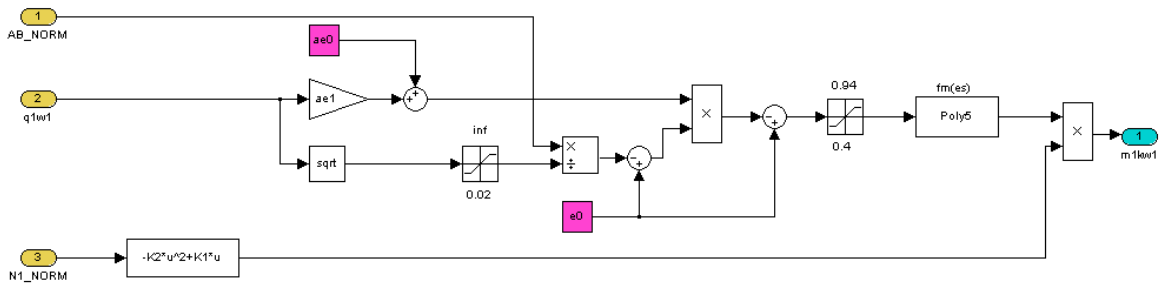


Abbildung 2.9: m_{1kw1} - Momentenbeiwert Düse 1 (gleiches für Düse 2)

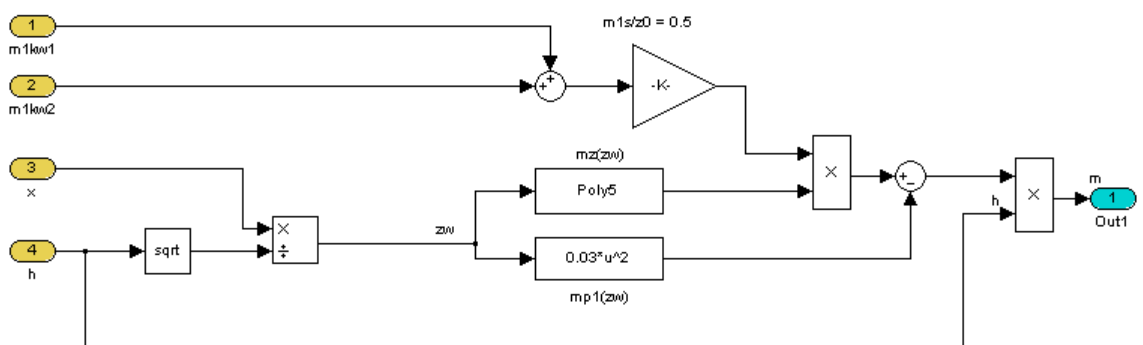


Abbildung 2.10: m_1 - Zusammenführung der beiden Momentenbeiwerte + Gewichtung

Parameter für die Momentenberechnung (m)

Für die Parametrierung des Modelles zur Momentenberechnung werden die Parameter der Polynomfunktion 2. Grades (Drehmoment m in Abhängigkeit der Düsenposition) des theoretischen Momenteneinfluss / Düse benötigt. Diese können über die Abbildung 2.11 ermittelt werden.

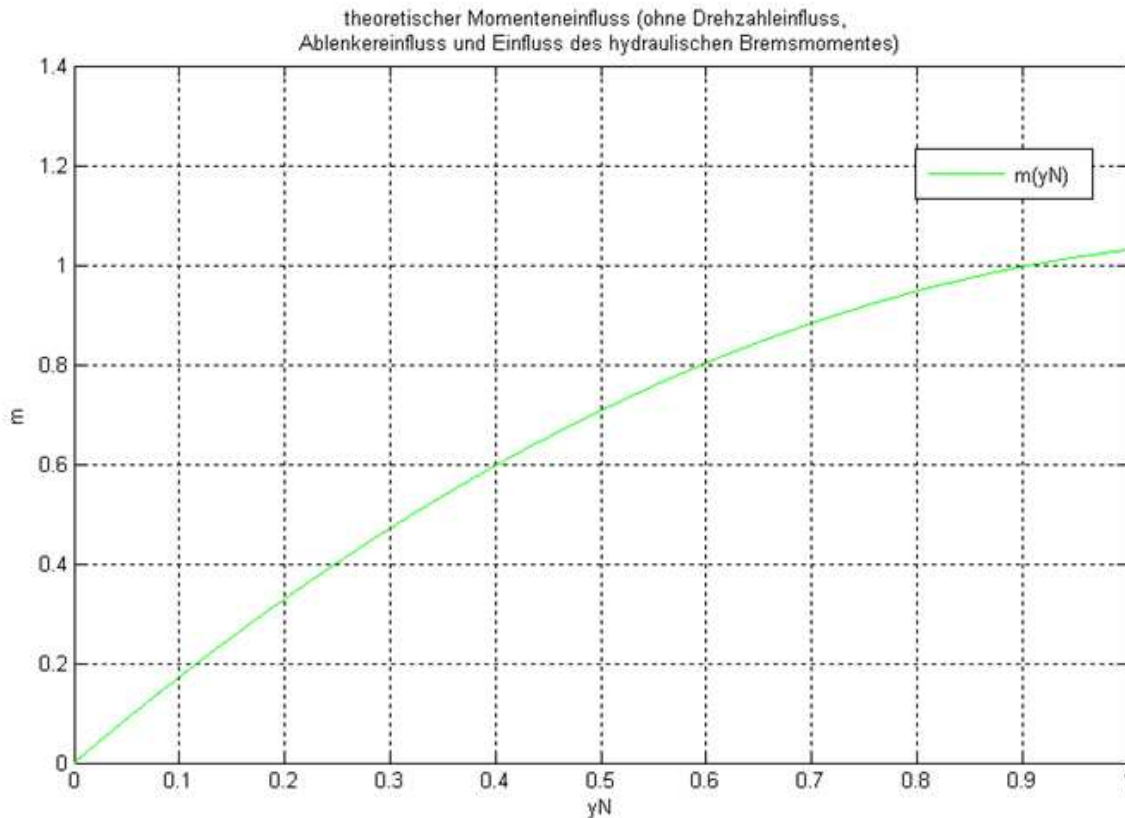


Abbildung 2.11: theoretischer Momenteneinfluss (ohne Drehzahleinfluss, Ablenkereinfluss und Einfluss des hydraulischen Bremsmomentes)

$$m_{1w1} = (K_1 - K_2 N_{1_NORM}) N_{1_NORM} \quad (2.15)$$

Für die Koeffizienten K_1 und K_2 werden folgende Größen verwendet:

$$K_1 = 1,8$$

$$K_2 = 0,77$$

2.1.5 Modellabschnitt Turbine und Generator

Die Drehzahl n wird mit Hilfe der Gleichungen 2.16 bis 2.18 berechnet. Die Zeitkonstante T_A kann als Anlaufzeitkonstante der Maschine interpretiert werden (berücksichtigt die

Schwungmasse der Maschine) und wird über Kenndaten bzw. Versuche auf der Anlage ermittelt.

Wichtig: T_A muß auf die verwendete Maschine eingestellt werden, da es ansonsten zu Fehlinterpretationen der einzelnen Betriebszustände kommen kann.

$$x = \frac{1}{T_A} \int_{t_0=0}^t m(\tau) d\tau \quad (2.16)$$

x entspricht der relativen Drehzahl in den Grenzen $0 < x < 2$.

$$x_P = \begin{cases} x & \text{wenn } GEN_{EIN} == 0 \\ 1 & \text{sonst} \end{cases} \quad (2.17)$$

Die relative Drehzahl x_P ist abhängig von der Stellung des Generatorschalters (Leistungsschalters). Je nachdem, ob der Schalter eingelegt ist (Generator ist synchronisiert), wird die die relative Drehzahl auf $x_P = 1$ begrenzt.

$$n = 100 x_P \quad (2.18)$$

Für die Ermittlung der Leistung P_{Ist} gelten folgende Zusammenhänge:

$$m_x = \begin{cases} 0 & \text{wenn } GEN_{EIN} == 0 \\ x_P & \text{sonst} \end{cases} \quad (2.19)$$

$$P_{Ist} = P_{nom} m_x \quad (2.20)$$

Die Nennleistung des Generators wird mit dem Parameter P_{nom} berücksichtigt.

Für den Fall, dass in der Reglersoftware das elektrische Drehmoment benötigt wird, kann dieses wie folgt berechnet werden:

$$m_{elektr} = \frac{P_{Insel}}{P_{nom}} / (x_P + 0.00001) \quad (2.21)$$

P_{INSEL} entspricht der Leistung im Inselbetrieb (d.h. keine Synchronisierung mit dem Verbundnetz). Der Faktor 0.00001 dient als Offset, um eine Division durch 0 zu vermeiden.

Abbildung 2.12 zeigt die Umsetzung in Simulink.

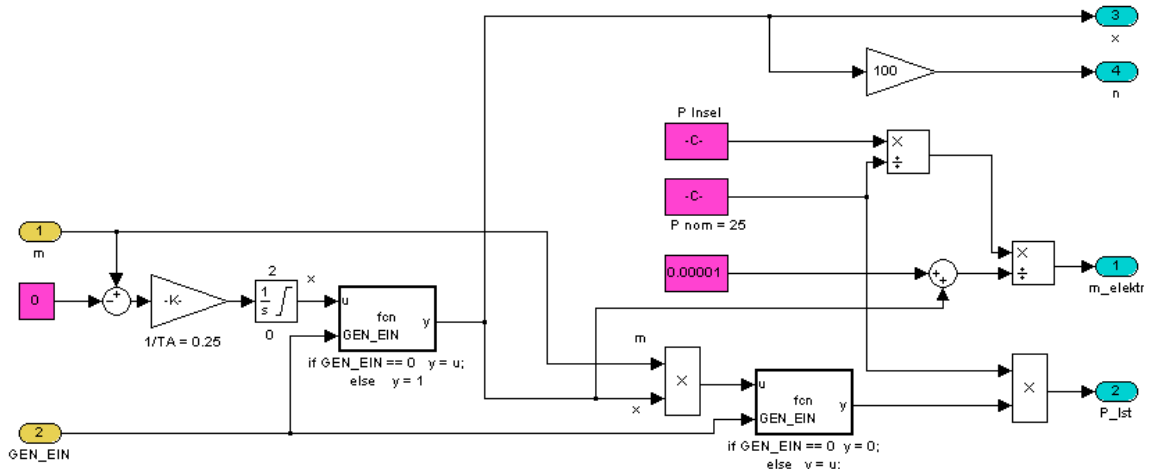


Abbildung 2.12: Modell für die Ermittlung von Turbinen- bzw. Generatoreigenschaften

2.1.6 Modellabschnitt Stauraum

Dieses Modell simuliert die Wasserführung im Stauraum. Es beinhaltet die Berechnung der aktuellen Fallhöhe, sowie den aktuellen Pegelstand im Stauraum.

$$Q_{Stau} = Q_{IN} - Q_{nom} q \quad (2.22)$$

Q_{nom} entspricht dem Nenndurchfluss der Turbine, Q_{IN} dem Zufluss des Stauraums.

$$L_{Pegel_OW} = \frac{1}{T_I} \int_{t_0=0}^t Q_{Stau}(\tau) d\tau \quad (2.23)$$

Mit L_{Pegel_OW} wird der Pegel im Stauraum (Oberwasserpegel) bezeichnet. Für L_{Pegel_OW} gilt : $L_{Pegel_OW_L} < L_{Pegel_OW} < L_{Pegel_OW_H}$. $L_{Pegel_OW_H}$ entspricht dem höchst zulässigen Pegelstand im Stauraum, $L_{Pegel_OW_L}$ dem niedrigsten. Der Parameter T_I dient als Zeitkonstante für den Integrator zur Auffüllung des Staubeckens.

Die Fallhöhe F_{H_MOD} ergibt sich nun bei der Pelton-Turbine aus der Differenz zwischen dem Oberwasserpegel und der Maschinenkote (Achsenmitte des Laufrades bezogen auf den Meeresspiegel). Da dieses Modell jedoch auch für andere Turbinentypen verwendet werden kann, gibt es die Möglichkeit, den Unterwasserpegel in Abhängigkeit des Maschinendurchflusses zu berechnen (siehe Gleichung 2.24).

$$L_{Pegel_UW} = (Q_{nom} q q_{prop_TW}) + L_{Pegel_UW_Offset} \quad (2.24)$$

$$F_{H_MOD} = L_{Pegel_OW} - L_{Pegel_UW} \quad (2.25)$$

Wie schon erwähnt wird im Fall einer Pelton-Turbine für den Parameter $L_{Pegel_UW_Offset}$ die Maschinenkote eingesetzt. Der Parameter q_{prop_TW} dient als Proportionalfaktor für den Durchfluss zur Bestimmung des Unterwasserpegelstand. Im Fall einer Pelton-Turbine ist $q_{prop_TW} = 0$.

Die relative Fallhöhe h ergibt sich aus den Gleichungen 2.26 bis 2.27.

$$h_1 = \frac{F_{H_MOD}}{H_{nom}} \quad (2.26)$$

Für den Parameter H_{nom} wird die Nennfallhöhe eingesetzt.

Mit Hilfe der Übertragungsfunktion

$$G(s) = \frac{h_w(s)}{q(s)} = \frac{T_D s}{K_{TD} T_D s + 1} \quad (2.27)$$

wird die Anlaufzeit der Wassermassen im Stauraum berücksichtigt. Die Wahl für K_{TD} und T_D liegt bei $K_{TD} = 1$ und $T_D = 1.53$ (Standardwerte des Andritzmodells).

Daraus ergibt sich für die relative Fallhöhe:

$$h = h_1 - h_w. \quad (2.28)$$

Abbildung 2.13 zeigt die Umsetzung in Simulink.

Parameter für die Stauraumberechnung

Q_{IN} ist der Zufluss in m^3/s . Dieser wurde für die Versuche entsprechend hoch gewählt ($Q_{IN} = 10$), sodass sich das simulierte Staubecken nicht während der Simulation entleert.

Der Parameter $Q_{nom} = 7,6 m^3/s$ ergibt sich aus Abbildung 2.14 unter Berücksichtigung der Düsenanzahl (Wert bei 100% Öffnung wird sinngemäß mit 2 multipliziert (2-düsige Peltonmaschine)).

Die Parameter OW_{LIH} ($L_{Pegel_OW_H}$) und OW_{LIL} ($L_{Pegel_OW_L}$) sind die Begrenzungen des Oberwasserpegels (Eingabe in m_M).

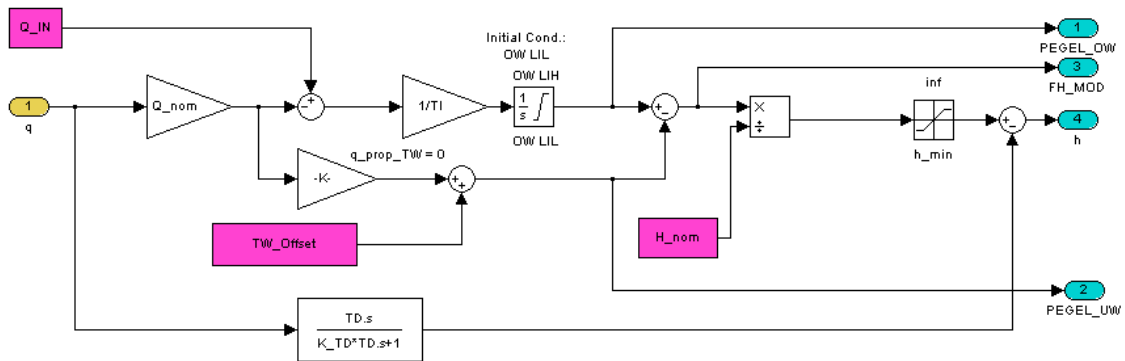


Abbildung 2.13: Modell des Stauraums

Für den Parameter TW_{Offset} ($L_{Pegel_UW_Offset}$) wird die Kote der Maschinenachse verwendet (bei anderen Turbinentypen der Unterwasserpegel). Die Fallhöhe wird in H_{nom} berücksichtigt und mit der normierten Minimalfallhöhe h_{MIN} nach unten begrenzt.

Mit den Parametern T_D und K_{TD} wird die Anlaufzeit der Wassermassen berücksichtigt. (Wahl von $T_D = 1,5$ und $K_{TD} = 1$).
Achtung: T_D richtig wählen (klein, aber aufgrund von Stabilitätsproblemen nicht zu klein).

Die Parametrierung des APROL-Modells wird im Anhang A genauer beschrieben.

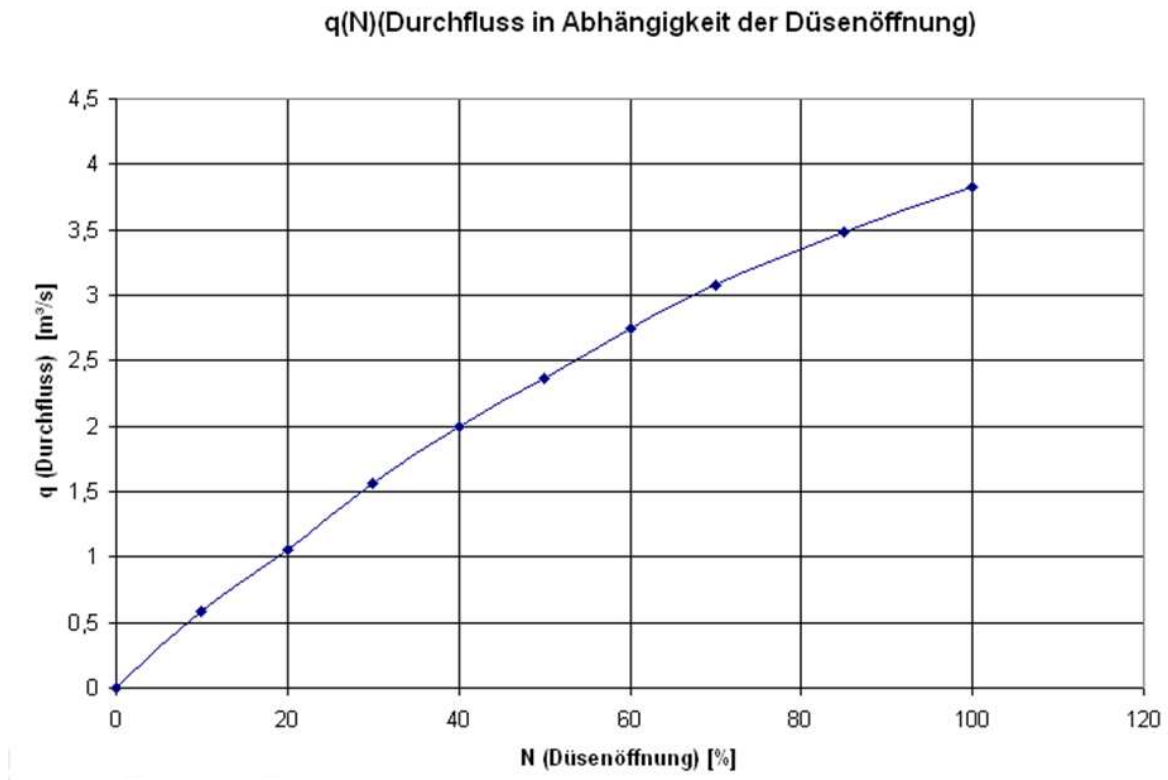


Abbildung 2.14: Durchfluss in Abhängigkeit der Düsenöffnung

2.2 Vergleich Modell- und Anlagendaten

Wie aus Abbildung 2.15 ersichtlich, bietet das Turbinenmodell eine für die Simulation ausreichend genaue Rekonstruktion des Ablenkers. Lediglich die Druckstöße beim Sprung von 0% auf 100% werden nicht berücksichtigt. Für das Modell der Düsen gilt eine ähnliche Übereinstimmung.

Abbildung 2.16 zeigt das Startverhalten der Turbine im Modellbetrieb (strichliert) bzw. zum Vergleich das Verhalten der Anlage. Auch hier ist ersichtlich, dass das Modell eine ausreichende Genauigkeit aufweist.

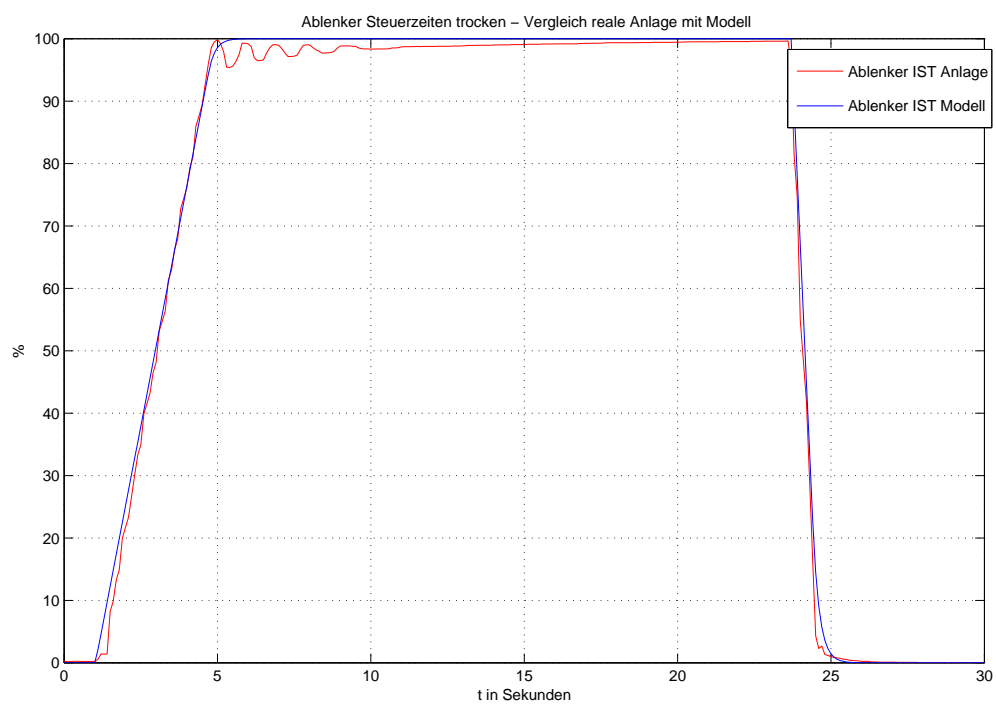


Abbildung 2.15: Vergleich Ablenker Steuerzeiten des Modells bzw. der Anlage

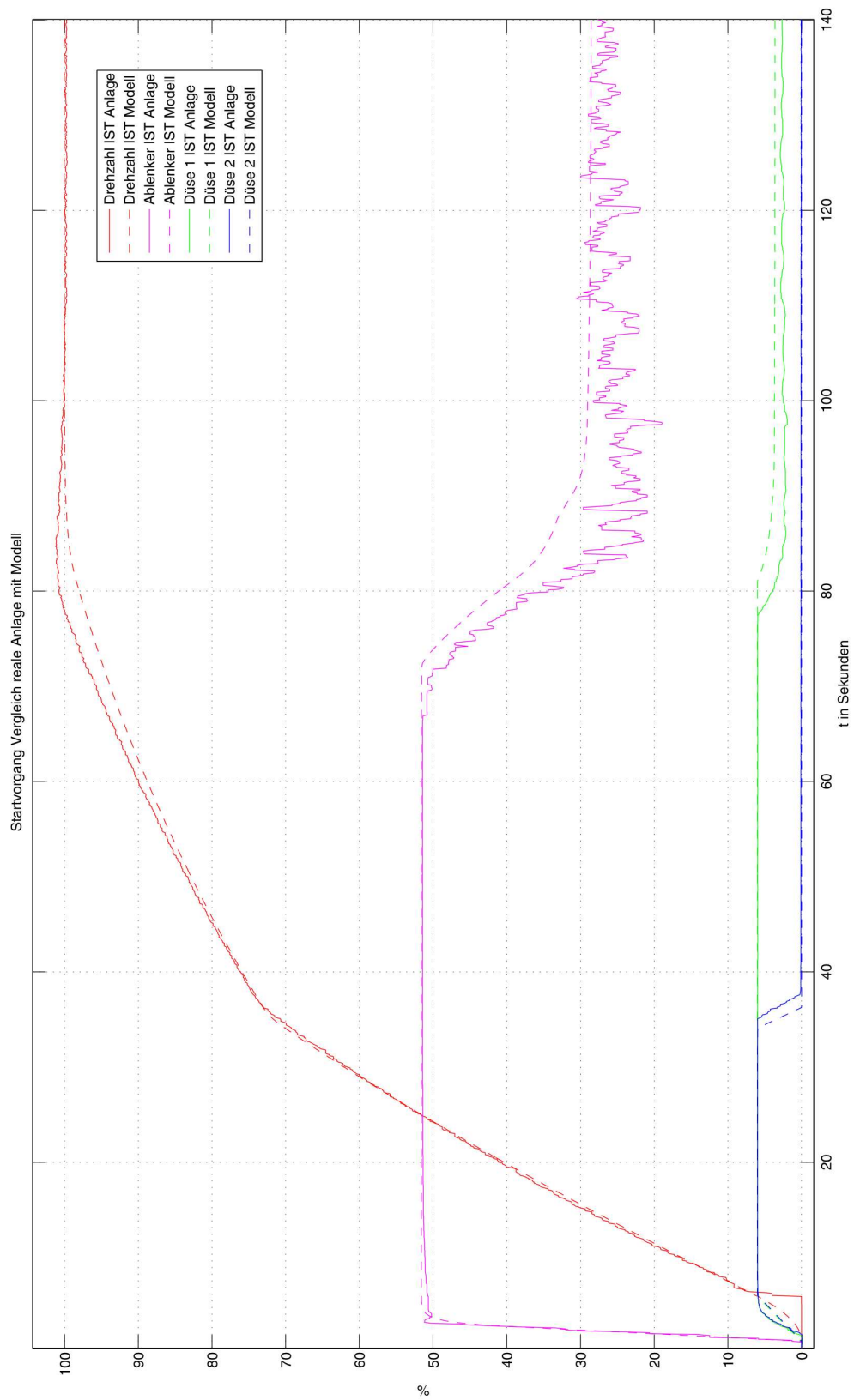


Abbildung 2.16: Vergleich Startverhalten des Modells bzw. der Anlage

Kapitel 3

Regler

Als Basis für den Turbinenregler wurde die Reglerstruktur der Firma Andritz verwendet.

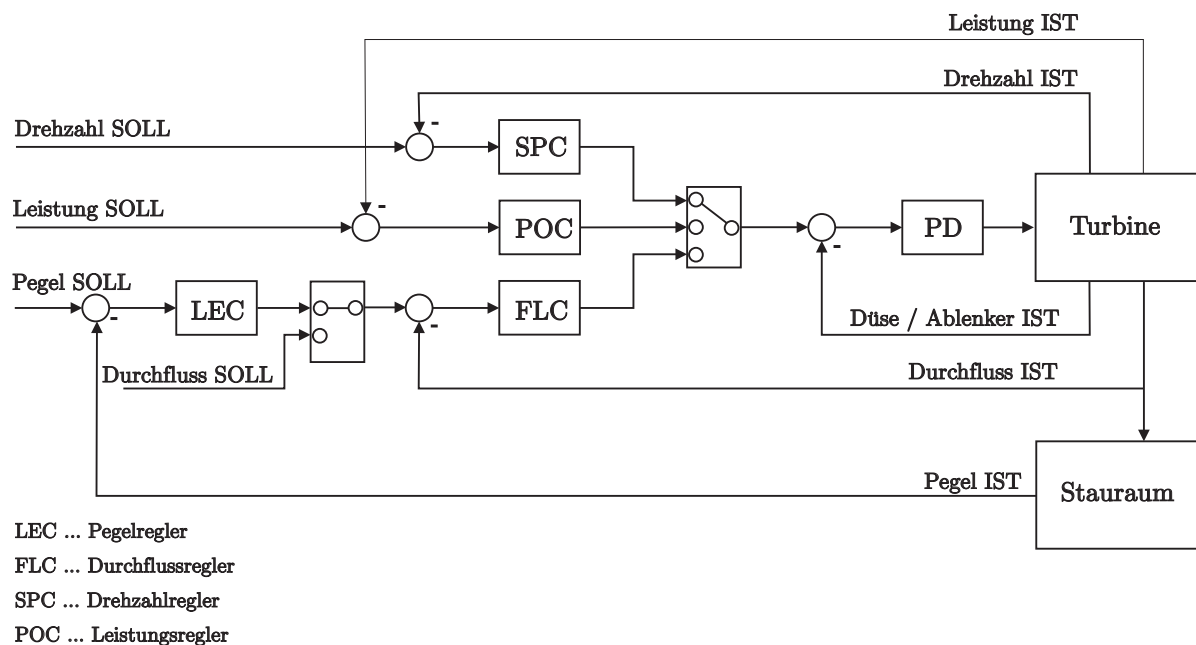


Abbildung 3.1: Turbinenreglerkonzept Andritz AG

Wie aus Abbildung 3.1 ersichtlich ist, gibt es mehrere Regelkreise in der Turbinenreglersoftware. Diese unterteilen sich in:

- SPC ... Drehzahlregler
- POC ... Leistungsregler
- FLC ... Durchflussregler
- LEC ... Pegelregler

Den oben erwähnten Reglern dient als Grundlage eine PI- bzw. PID Reglerstruktur. Abhängig von den Erfordernissen des Reglers erfolgt die Reglerparametrierung über fixe bzw. über variable Reglerparameter.

Der folgende Abschnitt liefert eine kurze Einleitung in die Reglerstruktur und in die Eigenschaften des PID Reglers. Des weiteren wird ein modellfreies Tuningverfahren für PID Regler vorgestellt.

3.1 PID-Regler

3.1.1 Einführung

Ausgehend von einem Standardregelkreis (siehe Abbildung 3.2) mit der Führungsgröße r , der Stellgröße u , der Regelgröße y und dem Regelfehler e , kann der PID-Regler (Proportional-Integral-Differential) mit Hilfe der folgenden Gleichung beschrieben werden:

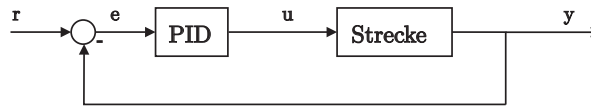


Abbildung 3.2: PID-Regler im Standardregelkreis

$$u(t) = \underbrace{K_1 e(t)}_{P\text{-Anteil}} + \underbrace{K_2 \int_0^t e(\tau) d\tau}_{I\text{-Anteil}} + \underbrace{K_3 \frac{de(t)}{dt}}_{D\text{-Anteil}} \quad (3.1)$$

Wie man in Gleichung 3.1 erkennen kann, setzt sich die Struktur aus drei Teilen zusammen. Der P-Anteil ist dem augenblicklichen Wert des Regelfehlers e proportional. Der I-Anteil ist dem Integral über den Regelfehler proportional \rightarrow Einfluss des Verlaufes von e in u . Der D-Anteil ist der zeitlichen Ableitung des Regelfehlers proportional und berücksichtigt somit die momentane Änderungstendenz des Regelfehlers.

Mit Hilfe der Parameter,

$$K_1 := K_P; \quad K_2 := \frac{K_P}{T_I}; \quad K_3 := K_P T_D$$

erhält man die in der Literatur übliche Darstellung des PID-Reglers

$$u(t) = K_P \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right]. \quad (3.2)$$

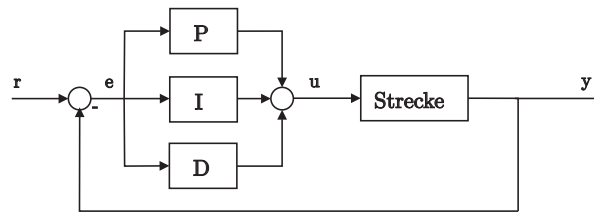


Abbildung 3.3: Regelkreis mit klassischem PID-Regler

Dabei sind folgende Bezeichnungen für die Parameter üblich:

- K_P ... Proportionalbeiwert
- T_I ... Nachstellzeit
- T_D ... Vorhaltezeit

Abhängig von der Wahl der Koeffizienten werden folgende Fälle unterschieden:

T_I	T_D	Bezeichnung
$\rightarrow \infty$	$= 0$	P-Regler
endlich	$= 0$	PI-Regler
$\rightarrow \infty$	$\neq 0$	PD-Regler
endlich	$\neq 0$	PID-Regler

Wird die Gleichung 3.2 mit Hilfe der Laplace-Transformation in den Bildbereich transformiert, erhält man, bei verschwindendem Anfangswert $e(0)$, die Übertragungsfunktion des idealen PID-Reglers

$$R(s) = \frac{u(s)}{e(s)} = K_P \left(1 + \frac{1}{s T_I} + s T_D \right). \quad (3.3)$$

Dieser Regler $R(s)$ ist aufgrund des Differenziergliedes $s T_D$ nicht realisierbar. Für eine praktische Realisierung des D-Anteiles wird dieser durch ein DT_1 -Glieð (Differenzierer mit Verzögerungsverhalten) ersetzt:

$$s T_D \rightarrow \frac{s T_D}{1 + s T} \quad (3.4)$$

T kann auch weiters noch als $T = A T_D$ angeschrieben werden. Der Faktor T bzw. A ist so zu wählen, dass sich das DT_1 -Glieð im interessierenden Frequenzbereich annähernd wie

das entsprechende D-Glied verhält. Die Übertragungsfunktion des realen PID-Reglers (eigentlich ein PIDT₁-Regler) ergibt sich somit zu:

$$R(s) = \frac{u(s)}{e(s)} = K_P \left(1 + \frac{1}{s T_I} + \frac{s T_D}{1 + s A T_D} \right). \quad (3.5)$$

Eine technische Umsetzung eines PI bzw. DT₁ Gliedes ist im Anhang C.2 genauer beschrieben.

Die Sprungantworten zu den einzelnen Reglertypen sind in Abbildung 3.4 ersichtlich.

Für weitere Erläuterungen zum Thema wird auf [9], [5] bzw. [6] verwiesen.

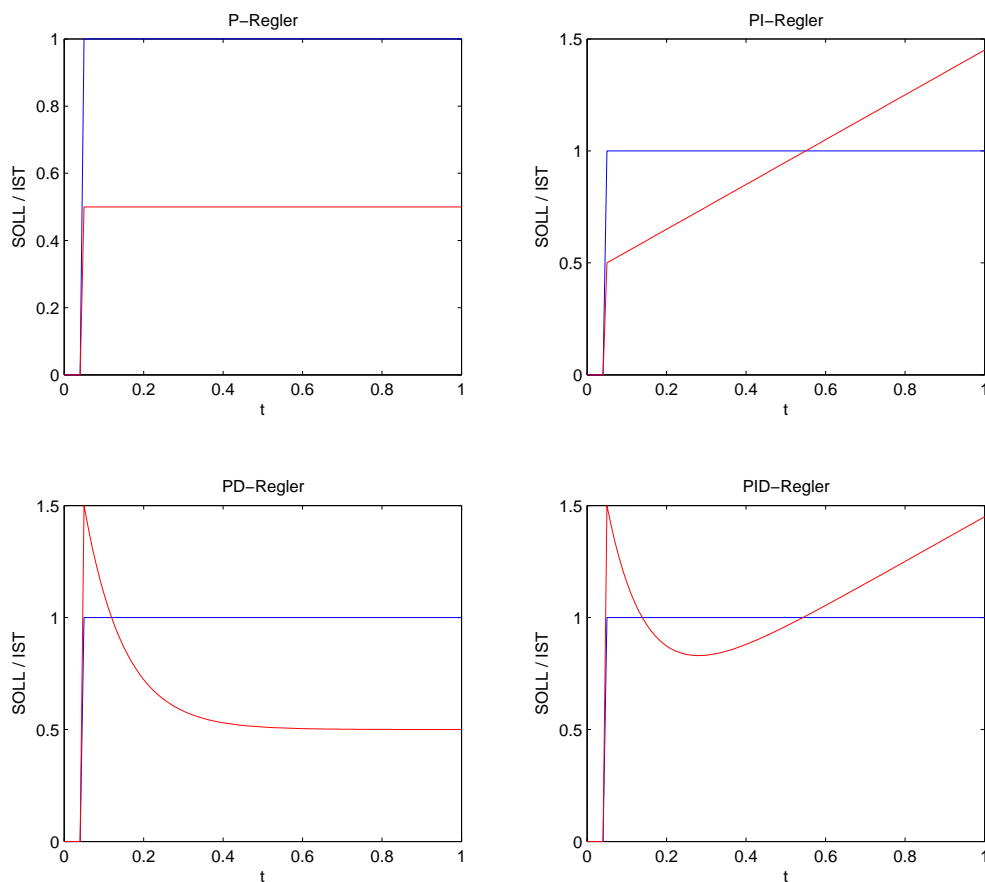


Abbildung 3.4: Sprungantworten der einzelnen Regler

3.1.2 PID-Einstellregeln

Für die meisten Optimierungsverfahren ist es wichtig, dass sich der Regelkreis vor der Optimierung in einem stabilen Zustand befindet. Dies kann vielfach mit Hilfe von Standardeinstellregeln bewerkstelligt werden.

Einstellregeln nach Ziegler-Nichols

Bei den Einstellregeln nach Ziegler-Nichols gibt es unterschiedliche Ansätze.

open loop method

Bei dieser Methode wird die Sprungantwort der Strecke aufgezeichnet. Durch den Wendepunkt wird eine Tangente (Wendetangente) gezeichnet (siehe Abbildung 3.5). Der Abstand a entspricht dem Wert zwischen dem Startpunkt der Sprungantwort in y-Richtung und dem Schnittpunkt der Wendetangente mit der y-Achse. Die Ermittlung der Zeiten T_v (Verzugszeit) und T_g (Ausgleichszeit) sind ebenfalls in Abbildung 3.5 ersichtlich.

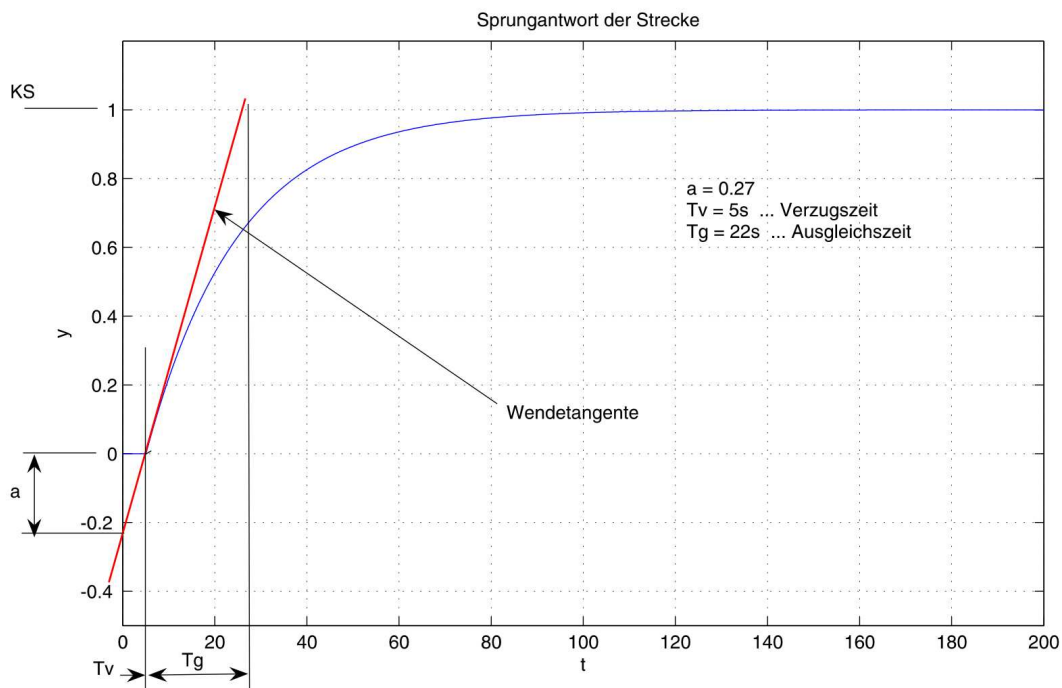


Abbildung 3.5: Sprungantwort der Strecke

Als Testmodell wurde folgende Streckenübertragungsfunktion gewählt:

$$G_1(s) = \frac{1}{20s + 1} e^{-5s} \quad (3.6)$$

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 3.1: Ziegler-Nichols Faustformeln 'open loop method'

Dieses Modell wird in weiterer Folge als Referenzmodell für die kommenden Einstellregeln bzw. für das hier verwendete Optimierungsverfahren verwendet.

Mit den aus Abbildung 3.5 ermittelten Werten für a und für T_v ($a = 0.27$; $T_v = 5s$) können nun über die Tabelle 3.1 direkt die Reglerparameter abgelesen werden.

Im vorliegenden Beispiel ergeben sich diese zu:

$$K_P = 4.44; T_I = 10; T_D = 2.5$$

closed loop method

Für diese Methode ist kein Eingriff in den offenen Regelkreis notwendig. Der PID-Regler wird vorerst nur als P-Regler konfiguriert (d.h. $T_I = \infty$; $T_D = 0$). K_P wird nun schrittweise erhöht, bis sich bei einer sprunghaften Änderung am Eingang, eine ungedämpfte Schwingung am Steckenausgang einstellt (siehe Abbildung 3.6). Diese Schwingung wird wiederum zur Ermittlung der Reglerkoeffizienten verwendet.

Der Nachteil an diesem Verfahren ist, dass dieses natürlich nur bei Strecken verwendet werden sollte, bei denen ein Erreichen der Stabilitätsgrenze keine Gefahren mit sich bringt.

Reglertyp	K_P	T_I	T_D
P	$0.5 K_u$	-	-
PI	$0.4 K_u$	$0.8 T_u$	-
PID	$0.6 K_u$	$0.5 T_u$	$0.12 T_u$

Tabelle 3.2: Ziegler-Nichols Faustformeln 'closed loop method'

Der Versuch mit der Strecke aus Gleichung 3.6 (siehe Abbildung 3.6) liefert das folgende Ergebnis:

$$K_u = 6.93; T_u = 18,4s$$

Durch Einsetzen dieser Größen in die Tabelle 3.2 ergeben sich die Reglerkoeffizienten zu:

$$K_P = 4.158; T_I = 9,2; T_D = 2.208$$

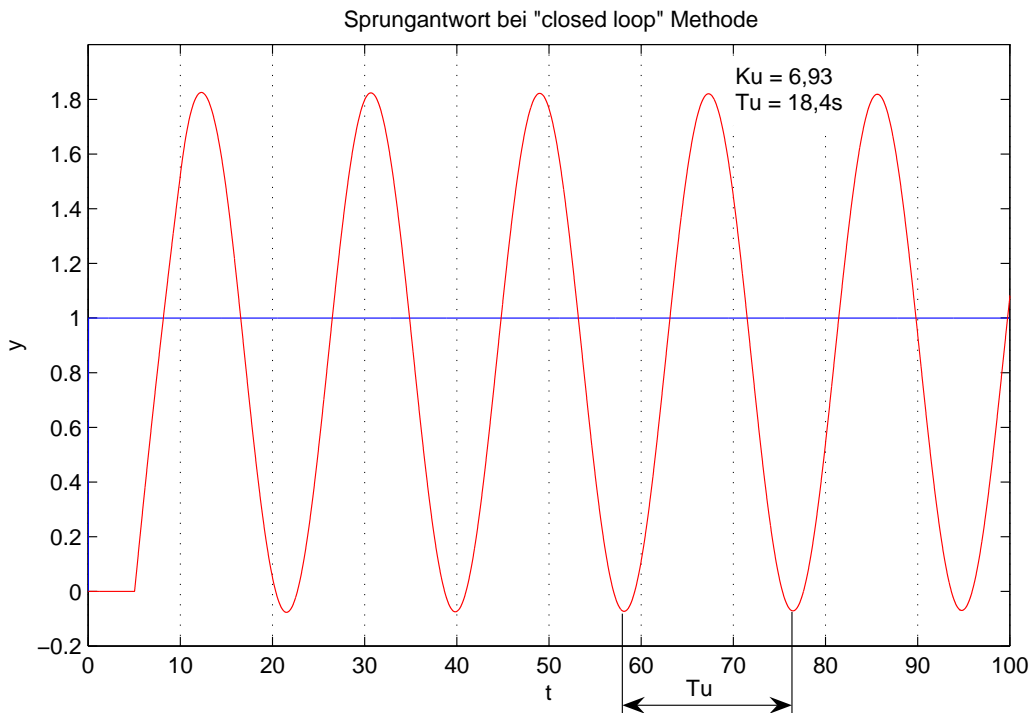


Abbildung 3.6: Sprungantwort der Strecke mit P-Regler (closed loop method)

Einstellregeln nach der T-Summenregel

Bei diesem Verfahren wird wie bei der 'open loop method' ein Sprung am Stelleingang der Strecke durchgeführt. Hat die Sprungantwort der Strecke einen s-förmigen Charakter, so kann die T-Summenregel zur Ermittlung der Parameter angewendet werden.

Die Idee besteht nun darin, über die aufgezeichnete Sprungantwort, die Summenzeitkonstante zu ermitteln. Wie in Abbildung 3.7 ersichtlich, wird die s-förmige Sprungantwort so 'geteilt', dass sich zwei gleich große Flächen zwischen der Teilgeraden und der Kurve ergeben. Über den Schnittpunkt der Teilgeraden mit der x-Achse erhält man die Summenzeitkonstante T_{Σ} . Der Wert für die Streckenverstärkung ergibt sich aus:

$$K_S = \frac{\Delta y}{\Delta u} = \frac{y_2 - y_1}{u_2 - u_1}, \quad (3.7)$$

wobei y die Ausgangsgröße der Strecke und u die Stellgröße darstellt.

Wie aus Abbildung 3.7 ersichtlich, ergeben sich die Werte von K_S und T_{Σ} zu:

$$K_S = 1; \quad T_{\Sigma} = 27s$$

Durch Einsetzen von K_S und T_{Σ} in die Tabelle 3.3 ergeben sich die Reglerkoeffizienten zu:

$$K_P = 1; \quad T_I = 17.82; \quad T_D = 4.59$$

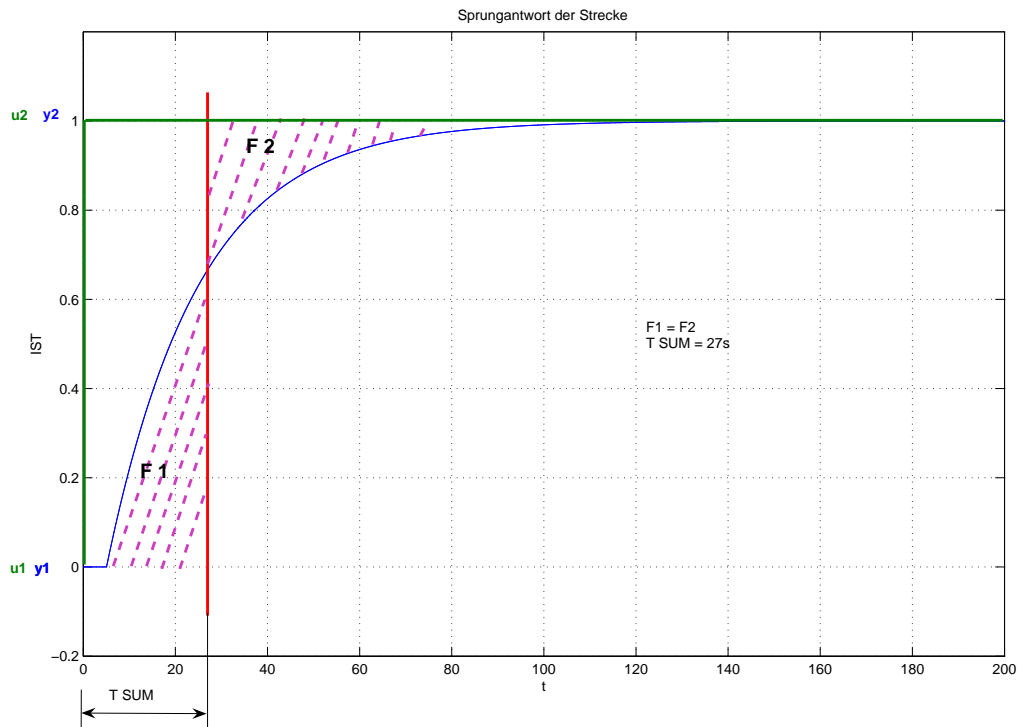


Abbildung 3.7: Sprungantwort der Strecke - T-Summe

Reglertyp	K_P	T_I	T_D
P	K_S^{-1}	-	-
PI	$0.5 K_S^{-1}$	$0.5 T_\Sigma$	-
PID	K_S^{-1}	$0.66 T_\Sigma$	$0.17 T_\Sigma$

Tabelle 3.3: Faustformeln für T-Summenregel

Ist ein schnellerer Regelverlauf gewünscht, kann die Tabelle 3.4 zur Ermittlung der Reglerkoeffizienten verwendet werden.

Für den 2. Fall → schneller Regelverlauf gewünscht, ergeben sich die Koeffizienten zu:

$$K_P = 2; \quad T_I = 21.6; \quad T_D = 5.238$$

3.1.3 Vergleich der ermittelten Reglerkoeffizienten

Ein Vergleich der in Kapitel 3.1.2 ermittelten Reglerparameter wird in Abbildung 3.8 dargestellt. Je nachdem, welche Eigenschaften die Regelung aufweisen soll, kann hier eine entsprechende Wahl der Parameter getroffen werden. Die beiden Ziegler-Nichols Methoden reagieren zwar schneller auf Sollwertänderungen, weisen jedoch ein erhöhtes Überschwingen auf.

Reglertyp	K_P	T_I	T_D
P	K_S^{-1}	-	-
PI	K_S^{-1}	$0.7 T_\Sigma$	-
PID	$2 K_S^{-1}$	$0.8 T_\Sigma$	$0.194 T_\Sigma$

Tabelle 3.4: Faustformeln für T-Summenregel (schneller Regelverlauf)

Einstellregel	K_P	T_I	T_D
ZN - open loop	4.44	10	2.5
ZN - closed loop	4.158	9.2	2.208
T-Summe	1	17.82	4.59
T-Summe (schneller Regelverlauf)	2	21.6	5.238

Tabelle 3.5: Vergleich der Einstellregeln

Die oben erwähnten Einstellregeln dienen somit nur einer Voreinstellung der Regler. Wie aus Abbildung 3.8 ersichtlich, ist eine weitere Optimierung dieser Regelparameter wünschenswert.

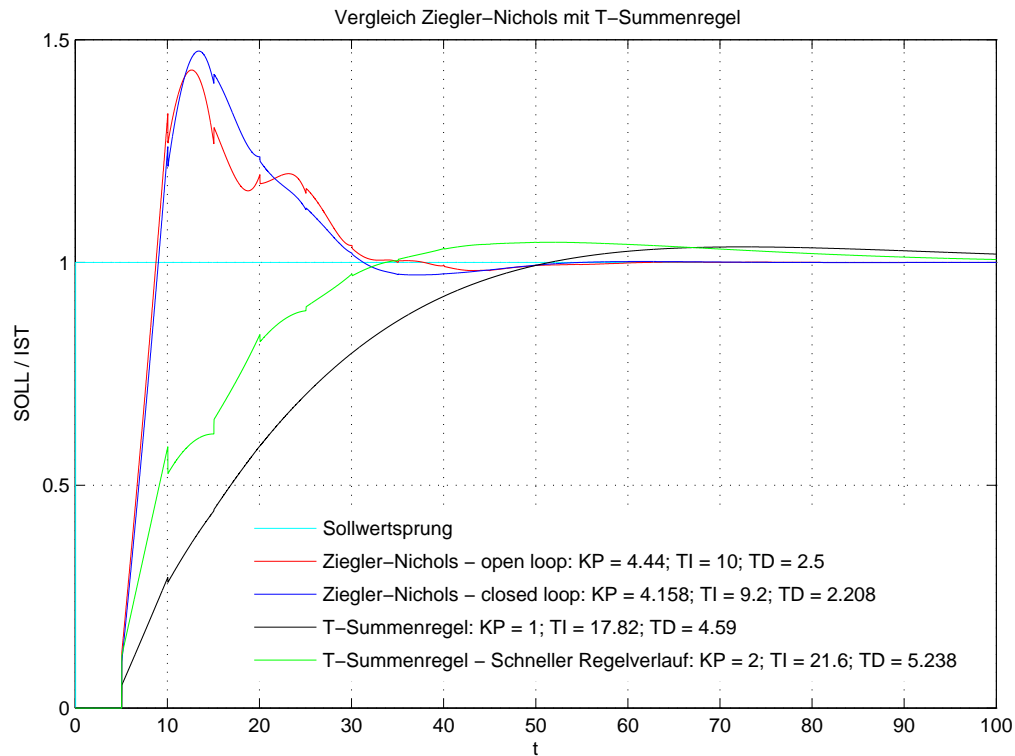


Abbildung 3.8: Vergleich der Einstellregeln

3.2 Extremum Seeking

Ein Ziel der Diplomarbeit war die Angabe eines Optimierungsverfahrens, das in einem Turbinenregler implementiert und angewendet werden kann. Im Zuge dieser Ermittlung wurden mehrere Methoden miteinander verglichen und ein konkretes Anforderungsprofil erstellt.

Anforderungen an das Optimierungsverfahren:

- Der PID-Regler als Regler soll nicht durch eine andere Regelstruktur abgelöst werden.
- Das Optimierungsverfahren muß ohne Streckenmodell funktionieren \rightarrow modellfrei.
- Nach wenigen Iterationen soll sich ein gutes Ergebnis einstellen.
- Da die Software letztendlich auf einer SPS implementiert wird, ist das Ziel, ein Verfahren zu verwenden, das möglichst wenig Speicherbedarf bzw. Rechenaufwand benötigt.

Aus den oben erwähnten Gründen wurde die Suche nach einem passenden Optimierungsverfahren auf modellfreie Verfahren eingeschränkt. In der engeren Auswahl standen die

Verfahren *iterative feedback tuning (IFT)* [8] und *extremum seeking (ES)* [10]. Beide Verfahren versuchen, nach einem Sollwertsprung, den Wert einer Zielfunktion zu minimieren. Der wesentliche Vorteil von ES gegenüber IFT liegt in der weit aus niedrigeren Anzahl an benötigten Iterationen zur Findung der verbesserten Reglerparameter.

3.2.1 Prinzipielle Funktionsweise

Wie schon kurz erwähnt, handelt es sich beim *extremum seeking (ES)* [10] um ein modellfreies, im Betrieb verwendbares (onlinefähiges) Optimierungsverfahren. Der Algorithmus ist bereits seit 1922 bekannt und wurde erstmalig bei elektrischen Eisenbahnsystemen verwendet.

Die Gleichung 3.8 zeigt die Berechnungsvorschrift für den Wert der Kostenfunktion (integrated square error (ISE)). Dieser Wert wird nach Abschluss eines Sollwertsprunges im eingeregelter Zustand ermittelt. Der Fehler wird über das Zeitintervall $[t_0, T]$ berücksichtigt, wobei für die Zeit $t_0 = T_{MAX}$ und für die Zeit T der Zeitpunkt des eingeschwungenen Zustandes eingesetzt wird. T_{MAX} ist der Zeitpunkt, bei dem das erste Maximum (das erste Überschwingen) der Sprungantwort auftritt. Somit wird in der Berechnung von J der Zeitraum zwischen dem ersten Maximum und dem eingeregelter Zustand berücksichtigt (siehe Abbildung 3.9). Die Zeit T_{MAX} wird nach dem ersten Sollwertsprung ermittelt und bleibt für jeden weiteren Versuch konstant. Die Zeit T ist vor Beginn der Optimierung mittels Sollwertsprung vom Benutzer abzulesen.

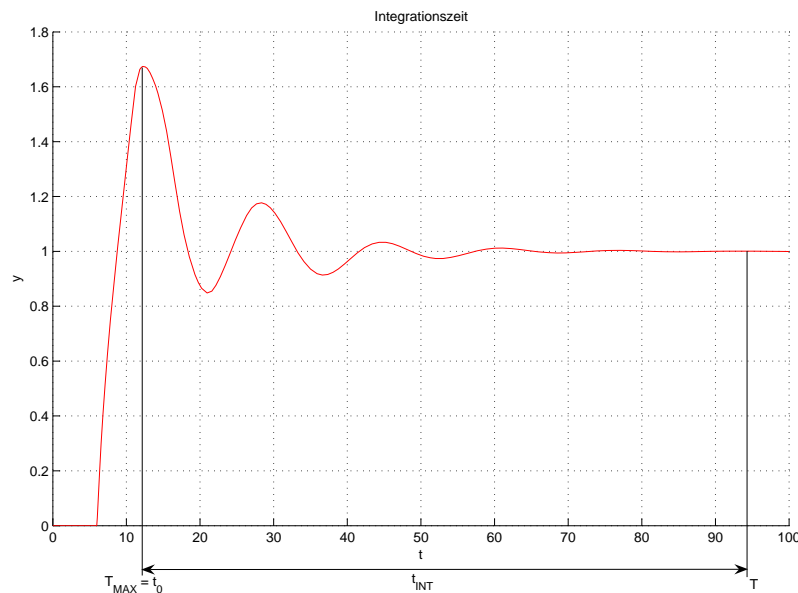


Abbildung 3.9: Berechnungszeitraum für J

$$J(\boldsymbol{\theta}) \triangleq \frac{1}{T - t_0} \int_{t_0}^T e^2(t, \boldsymbol{\theta}) dt \quad (3.8)$$

Der Fehler wird mittels der Gleichung 3.9 ermittelt. Im Parametervektor $\boldsymbol{\theta}$ werden die Reglerparameter des PID-Reglers zusammengefasst (siehe Gleichung 3.10).

$$e(t, \boldsymbol{\theta}) \triangleq r(t) - y(t, \boldsymbol{\theta}) \quad (3.9)$$

$$\boldsymbol{\theta} \triangleq [K_P, T_I, T_D]^T \quad (3.10)$$

Die in diesem Verfahren verwendete PID-Reglerstruktur unterscheidet sich zur klassischen Struktur aus Abbildung 3.3 in der Rückführung des Differentialanteiles (siehe Abbildung 3.10). Der Differentialanteil wird nur durch die Ausgangsgröße y bestimmt.

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau - T_D \dot{y}(t) \right) \quad (3.11)$$

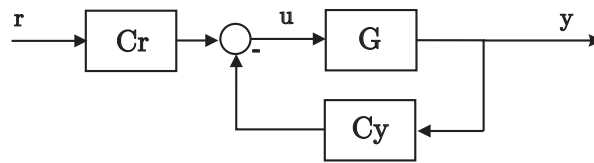


Abbildung 3.10: PID-Struktur für ES

Für die Blöcke C_r bzw. C_y aus Abbildung 3.10 gelten die folgenden Zusammenhänge:

$$C_r(s) = K_P \left(1 + \frac{1}{T_I s} \right) \quad (3.12)$$

$$C_y(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (3.13)$$

Eine PID-Struktur in der Art wie Abbildung 3.10 hat den Vorteil, dass bei einer sprunghaften Änderung des Eingangssignals der D-Anteil nicht direkt auf die Stellgröße wirkt und dadurch eine große Stellgrößenänderung vermieden wird (siehe Abbildung 3.11). Mit G wird die Regelstrecke beschrieben.

Abbildung 3.12 zeigt das generelle Optimierungsschema beim Extremum Seeking. Nach Durchführung eines Sollwertsprunges wird der Wert für J ermittelt. Ziel ist es, diesen

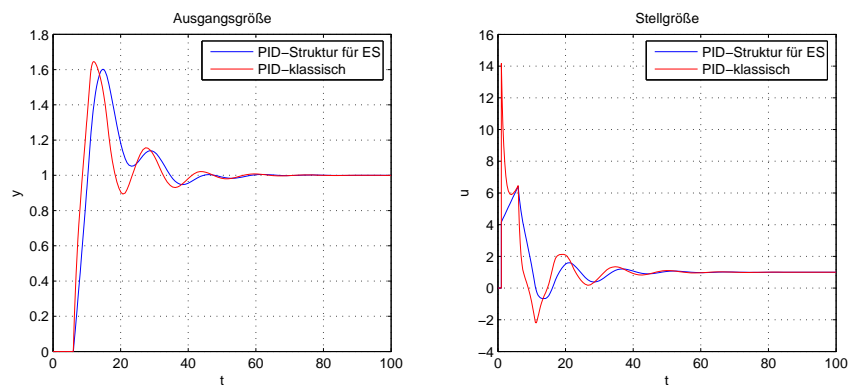


Abbildung 3.11: Vergleich PID-Struktur für ES mit PID klassisch

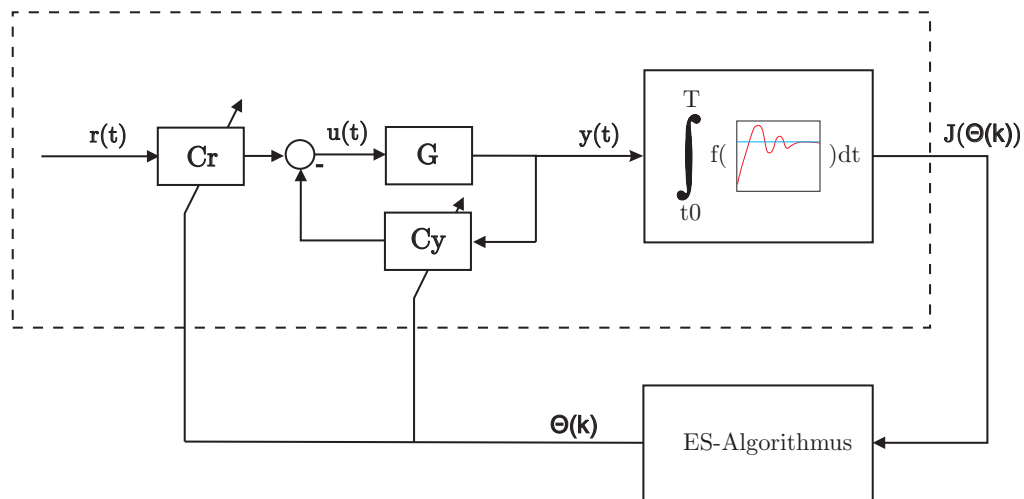


Abbildung 3.12: ES Optimierungsschema

Wert von J zu minimieren. Der Wert von J wird mit den bekannten Reglerparametern im ES-Algorithmus ausgewertet. Durch ein relativ einfaches Rechenverfahren werden daraus die neuen Reglerkoeffizienten für den nächsten Versuch bestimmt.

Zusammenfassend kann man die Schritte wie folgt unterteilen:

- Durchführung eines Sollwertsprunges
- Berechnen von J
- Ermitteln der neuen Reglerkoeffizienten über den ES-Algorithmus
- Wiederholung des Versuches, bis ein gewünschtes Abbruchkriterium erfüllt ist (z.B. keine Verbesserung mehr möglich; max. Anzahl an Iterationen erreicht ...)

3.2.2 Der extremum seeking Algorithmus

Der Einfachheit halber wird in dieser kurzen Einführung der Fall für einen einzigen Parameter erläutert. Dieser wird hier mit θ bezeichnet.

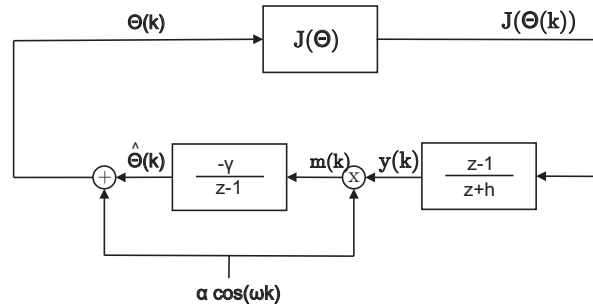


Abbildung 3.13: Zeitdiskretes ES Optimierungsschema

In Abbildung 3.13 bezeichnet k den Iterationsindex.

Die Zielfunktion wird wie folgt angegeben:

$$J(\theta) = J^* + \frac{J''}{2} (\theta - \theta^*)^2, \quad (3.14)$$

Jede zweifach stetig differenzierbare Funktion kann durch Gleichung 3.14 lokal angenähert werden. Die Gleichung 3.14 entspricht einer Taylorreihe der Zielfunktion entwickelt um θ^* , die nach der zweiten Ableitung abgebrochen wurde.

J = Wert der Zielfunktion, die minimiert werden soll, θ = der aktuell verwendete Parameter, J^* = Minimum der Zielfunktion, J'' = zweite Ableitung, θ^* = unbekannter optimaler Parameter.

Der Schätzfehler $\tilde{\theta}$ ergibt sich zu

$$\tilde{\theta} = \theta^* - \hat{\theta}, \quad (3.15)$$

wobei $\hat{\theta}$ dem geschätzten Parameter entspricht.

Das Ziel von ES ist nun, $J(\theta)$ so weit zu minimieren, dass für den geschätzten Wert von $\hat{\theta}$ gilt: $\hat{\theta} = \theta^*$ sowie $J(\hat{\theta}) = J^*$.

Durch Substitution von

$$\theta - \theta^* = \tilde{\theta} - \alpha \cos(\omega k), \quad (3.16)$$

in die Gleichung 3.14, erhält man für $J(\theta)$:

$$J(\theta) = J^* + \frac{J''}{2} \left(\tilde{\theta} - \alpha \cos(\omega k) \right)^2, \quad (3.17)$$

$$J(\theta) = J^* + \frac{J''}{2} \tilde{\theta}^2 - \alpha \tilde{\theta} J'' \cos(\omega k) + \frac{\alpha^2 J''}{2} \cos^2(\omega k) \quad (3.18)$$

α entspricht dem Amplitudenwert des Testsignales aus Abbildung 3.13, ω der Modulationsfrequenz und k entspricht dem Index der aktuellen Iteration.

Die Trigonometrische Identität für $2\cos^2(\omega k) = 1 + \cos(2\omega k)$ angewandt auf Gleichung 3.18 liefert:

$$\begin{aligned} J(\theta) &= J^* + \frac{J''}{2} \tilde{\theta}^2 - \alpha \tilde{\theta} J'' \cos(\omega k) + \frac{\alpha^2 J''}{4} + \frac{\alpha^2 J''}{4} \cos(2\omega k), \quad \xrightarrow{\text{umgeformt}} \\ J(\theta) &= J^* + \frac{\alpha^2 J''}{4} + \frac{\alpha^2 J''}{4} \cos(2\omega k) - \left(\alpha J'' \cos(\omega k) \right) \tilde{\theta} + \frac{J''}{2} \tilde{\theta}^2 \end{aligned} \quad (3.19)$$

Der Term $\frac{J''}{2} \tilde{\theta}^2$ wird aufgrund des quadratischen Einflusses in $\tilde{\theta}$ und dem Fokus auf eine lokale Analyse vernachlässigt. Der Hochpassfilter aus Abbildung 3.13 $\frac{z-1}{z+h}$ dient der Filterung der Gleichanteile. Die Größen J^* und $\frac{\alpha^2 J''}{4}$ werden in weiterer Folge ebenfalls vernachlässigt.

Anmerkung: In der weiteren Abhandlung wird mit z der Verschiebeoperator und nicht wie sonst üblich die komplexe Variable der z -Transformation bezeichnet. Das heißt, es gilt:

$$z y(k) = y(k+1) \quad (3.20)$$

Daraus ergibt sich für die Ausgangsgröße des Hochpassfilters

$$y(k) = \frac{z-1}{z+h} J(\theta) \approx \frac{\alpha^2 J''}{4} \cos(2\omega k) - \left(\alpha J'' \cos(\omega k) \right) \tilde{\theta}. \quad (3.21)$$

Nach 'Demodulation' mit dem Signal $\alpha \cos(\omega k)$ erhält man

$$m(k) = \alpha \cos(\omega k) \frac{z-1}{z+h} J(\theta) \approx \frac{\alpha^3 J''}{4} \cos(\omega k) \cos(2\omega k) - \left(\alpha^2 J'' \cos^2(\omega k) \right) \tilde{\theta}. \quad (3.22)$$

Durch Anwendung der trigonometrischen Identitäten für

$\cos(x)\cos(y) = \frac{1}{2}(\cos(x-y) + \cos(x+y)) \rightarrow \cos(2x)\cos(x) = \frac{1}{2}(\cos(\omega k) + \cos(3\omega k))$
und $2\cos^2(\omega k) = 1 + \cos(2\omega k)$ ergibt sich die Gleichung 3.22 zu

$$m(k) = \alpha \cos(\omega k) \frac{z-1}{z+h} J(\theta) \approx \frac{\alpha^3 J''}{8} (\cos(\omega k) + \cos(3\omega k)) - \frac{\alpha^2 J''}{2} \tilde{\theta} - \frac{\alpha^2 J''}{2} \tilde{\theta} \cos(2\omega k). \quad (3.23)$$

Durch den Integratoranteil $\frac{\gamma}{z-1}$, der wie ein Tiefpassfilter wirkt, werden die hochfrequenten Anteile im Signal $m(k)$ geschwächt und in weiterer Folge vernachlässigt. Daraus ergibt sich für den Schätzwert

$$\hat{\theta} \approx \frac{\gamma}{z-1} \left[\frac{\alpha^2 J''}{2} \tilde{\theta} \right] \quad (3.24)$$

Durch Verwendung der Gleichung 3.14 und der Randbedingung, dass θ^* konstant ist, folgt:

$$(z-1)[\tilde{\theta} - \theta^*] \approx -\gamma \left[\frac{\alpha^2 J''}{2} \tilde{\theta} \right] \quad (3.25)$$

$$\tilde{\theta}(k+1) - \tilde{\theta}(k) - \theta^* + \theta^* \approx -\gamma \left[\frac{\alpha^2 J''}{2} \tilde{\theta}(k) \right] \quad (3.26)$$

daraus folgt:

$$\tilde{\theta}(k+1) \approx \left(1 - \frac{\gamma \alpha^2 J''}{2} \right) \tilde{\theta}(k) \quad (3.27)$$

Wie aus Gleichung 3.27 ersichtlich, wird der Schätzfehler pro Iteration kleiner ($\tilde{\theta}(k+1) < \tilde{\theta}(k)$), wenn der Verstärkungsfaktor γ und der Amplitudenwert des Testsignals α so gewählt werden, dass gilt: $0 < \frac{\gamma \alpha^2 J''}{2} < 2$.

Der Multiparameter ES Algorithmus

Für die Ermittlung der Berechnungsvorschrift zur Reglerparametrierung wird nun das aus Abbildung 3.13 dargestellte Optimierungsschema herangezogen. Im Vektor $\boldsymbol{\theta}(k)$ werden die aktuellen Reglerparameter zusammengefasst.

$J(\boldsymbol{\theta}(k))$ erhält man aus den jeweiligen Sprungversuchen, wobei k für die Zahl der Iteration steht. Der Filter $\frac{z-1}{z+h}$ wird für die weitere Ermittlung in $\frac{1}{z+h}$ und $(z-1)$ zerlegt.

Daraus ergibt sich nun:

$$\xi(k)(z+h) = J(\boldsymbol{\theta}(k)) \text{ bzw.} \quad (3.28)$$

$$\xi(k+1) = -\xi(k)h + J(\boldsymbol{\theta}(k)) \quad (3.29)$$

$y(k)$ ergibt sich zu:

$$y(k) = (z-1)\xi(k) \rightarrow y(k) = \xi(k+1) - \xi(k) \quad (3.30)$$

Gleichung 3.29 eingesetzt in 3.30 ergibt für $y(k)$:

$$y(k) = -\xi(k)h + J(\boldsymbol{\theta}(k)) - \xi(k) \rightarrow y(k) = -(1+h)\xi(k) + J(\boldsymbol{\theta}(k)) \quad (3.31)$$

Der Einfachheit halber wird eine zusätzliche Größe $m(k)$ eingeführt.

$$m(k) = \alpha \cos(\omega k) (-(1+h)\xi(k) + J(\boldsymbol{\theta}(k))) \quad (3.32)$$

$\hat{\boldsymbol{\theta}}(k)$ ergibt sich nun zu:

$$\hat{\boldsymbol{\theta}}(k) = \frac{-\gamma}{z-1} m(k) \rightarrow \hat{\boldsymbol{\theta}}(k)(z-1) = -m(k)\gamma$$

$$\hat{\boldsymbol{\theta}}(k+1) = \hat{\boldsymbol{\theta}}(k) - \gamma m(k) \quad (3.33)$$

Gleichung 3.32 eingesetzt in 3.33 ergibt für $\hat{\boldsymbol{\theta}}(k)$:

$$\hat{\boldsymbol{\theta}}(k+1) = \hat{\boldsymbol{\theta}}(k) - \gamma \alpha \cos(\omega k) (-(1+h)\xi(k) + J(\boldsymbol{\theta}(k))) \quad (3.34)$$

Zuletzt erhält man noch für $\boldsymbol{\theta}(k+1)$:

$$\boldsymbol{\theta}(k+1) = \hat{\boldsymbol{\theta}}(k+1) + \alpha \cos(\omega(k+1)) \quad (3.35)$$

Aus den Gleichungen 3.29, 3.34 und 3.35 können nun die drei für die Implementierung benötigten Gleichungen angeschrieben werden:

$$\xi(k) = -h\xi(k-1) + J(\theta(k-1)) \quad (3.36)$$

$$\hat{\theta}_i(k+1) = \hat{\theta}_i(k) - \gamma_i \alpha_i \cos(\omega_i k) [J(\theta(k)) - (1+h)\xi(k)] \quad (3.37)$$

$$\theta_i(k+1) = \hat{\theta}_i(k+1) + \alpha_i \cos(\omega_i(k+1)) \quad (3.38)$$

Der Index i steht für den jeweiligen Parameter aus $\boldsymbol{\theta}$.

Die Anfangsbedingung für ξ_0 kann aus Gleichung 3.31 ermittelt werden. Da $y(k)$ zum Zeitpunkt $t=0$ gleich $y(k) = 0$ ist, ergibt sich für ξ_0 folgender Zusammenhang:

$$\xi_0 = \frac{1}{1+h} + J(\theta(0)) \quad (3.39)$$

Der Wert von $J(\theta)$ zum Zeitpunkt 0 entspricht dem Wert von $J(\theta)$, der sich mit den Startparametern von θ nach dem ersten Sprungversuch einstellt. Für alle weiteren Schritte gilt die Beziehung aus Gleichung 3.36.

Die Werte für die Schätzparameter entsprechen zum Startzeitpunkt den Reglerparametern, d.h.:

$$\hat{\theta}_i(0) = \theta_i(0) \quad (3.40)$$

Weiters ist zu erwähnen, dass bei einem iterativen Ansatz (wie es auch hier der Fall ist), der interne Startwert von k bei 2 liegt, obwohl es sich eigentlich um den ersten Versuch handelt. Dies kann dadurch erklärt werden, dass für die Berechnung einerseits der verzögerte Wert von ξ und weiters der neue Wert von θ_i einfließt. Diese Bedingung muß bei einer Auswertung der Daten entsprechend berücksichtigt werden (näheres ist im Codebeispiel B ersichtlich).

$\xi(k)$ ist ein skalarer Wert, der zwar nicht explizit in Abbildung 3.13 ersichtlich ist, jedoch eine nötige Zwischenlösung für den Updatealgorithmus liefert. Mit i wird der Eintrag im Vektor bezeichnet (beim PID-Regler sind es die drei Optimierungsparameter). γ_i ist wie oben beschrieben der Verstärkungsfaktor pro Parameter, α_i der Amplitudenwert

des Testsignals wählbar pro Parameter. *Vorsicht bei der Wahl von α_i und γ_i , da diese beiden Faktoren einen Einfluss auf die Stabilität und Konvergenz haben.* Die Modulationsfrequenz ω_i wird über $\omega_i = a^i \pi$ berechnet, wobei a zwischen $0 < a < 1$ liegt. Es ist wichtig, dass alle ω_i 's voneinander verschieden sind, da der Effekt jedes dieser Signale vom Algorithmus unterschieden werden muß. Der Faktor h des Hochpassfilters $\frac{z-1}{z+h}$ wird zwischen $0 < h < 1$ gewählt.

3.2.3 Anwendung des ES-Algorithmus auf die Strecke 3.6

Der Übersicht halber wird die Strecke 3.6 nochmals angeführt:

$$G_1(s) = \frac{1}{20s + 1} e^{-5s}$$

Wie aus den Versuchen für die Parameterbestimmung aus Abschnitt 3.1.3 ersichtlich, ist eine Betrachtung für den zu bewertenden Teil der Sprungantwort zwischen $t_0 = 10$ bzw. $T = 100$ sinnvoll. Die weiteren Parameter wurden zu: $\alpha = [0.1, 1, 0.1]^T$, $\gamma = [200, 1200, 200]^T$ und $\omega_i = a^i \pi$ gewählt. Generell haben sich die Werte für $a = 0.8$ und $h = 0.5$ als praktisch erwiesen (näheres dazu kann unter [10] bzw. [11] nachgelesen werden).

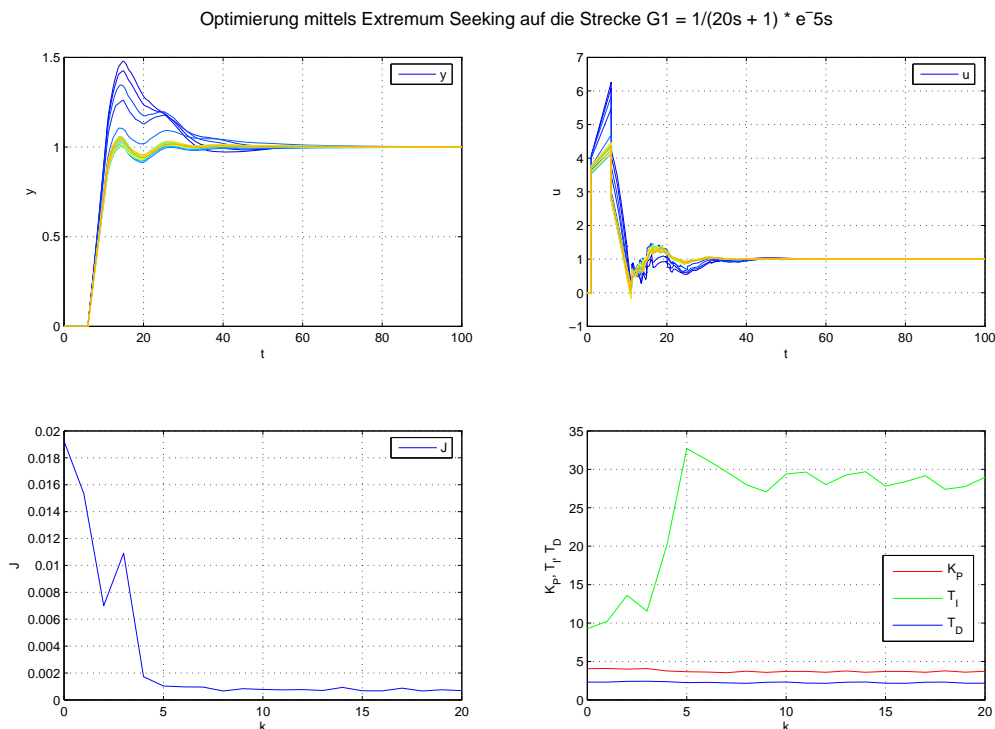


Abbildung 3.14: Optimierung mittels Extremum Seeking auf die Strecke G_1

Als Startvariablen für das ES wurden die Ziegler-Nichols Parameter aus Abschnitt 3.1.2 **closed loop** gewählt.

Wie aus Abbildung 3.14 ersichtlich, pendelt sich das Optimierungsverfahren nach 5 Versuchen auf einen gegenüber den Startparametern stark verbesserten Wert ein (siehe Verlauf von $J(k)$). Die folgende Tabelle zeigt den Wert der Zielfunktion und die Reglerparameter aus ES verglichen mit den Ziegler-Nichols - closed loop Parametern.

Einstellregel / Verfahren	K_P	T_I	T_D	J
ZN - closed loop	4.158	9.2	2.208	0.0208
Extremum seeking	3.7683	28.7965	2.2233	0.0007

Tabelle 3.6: Vergleich der Einstellregeln

Kapitel 4

Implementierung

Um den Code tatsächlich in der Steuerungssoftware (APROL) implementieren zu können, sind einige Anpassungen an dem in Kapitel 3 Abschnitt 3.2 vorgestellten Verfahren durchzuführen.

4.1 Implementierung in Simulink

Für eine korrekte Simulation der einzelnen Regler des Turbinenreglers inklusive der diversen Umschaltpunkte und Sicherheitsverriegelungen wurde die Turbinenreglersoftware mit den benötigten Bibliotheksbausteinen in Matlab-Simulink umgesetzt. Dadurch ist eine äquivalente Simulation zwischen Simulink und APROL gewährleistet.

4.1.1 PID - Implementierung in Simulink

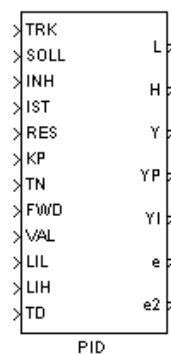


Abbildung 4.1: PID-Regler realisiert in Simulink

Neben der Erstellung eines Programmteiles für das ES wurden Anpassungen am bestehenden PI-Regler durchgeführt. Bis jetzt wurde der D-Anteil des PID-Reglers, falls er

benötigt wurde, separat zum bestehenden PI-Regler hinzugefügt. Der neu erstellte Regler beinhaltet den D-Anteil (aus PI wurde ein PID) in der Struktur, wie sie in Abbildung 3.10 dargestellt ist. Die Ein-/ Ausgangsbelegung wurde um den Eingang T_D erweitert, was jedoch keinen Einfluss für bestehende Pläne mit sich bringt, da der Standardwert von T_D im nicht beschriebenen Zustand bei 0 liegt (Abbildung 4.1 zeigt den Block des PID-Reglers). Eine detaillierte Beschreibung des in Simulink realisierten PID-Reglers ist im Anhang C.1.1 zu finden.

4.1.2 ES - Implementierung in Simulink

In Abbildung 4.2 sind die Ein-/ Ausgangsgrößen, die in Simulink zur Durchführung der Simulationen benötigt werden, ersichtlich und sie werden der Übersicht halber nochmals angeschrieben:

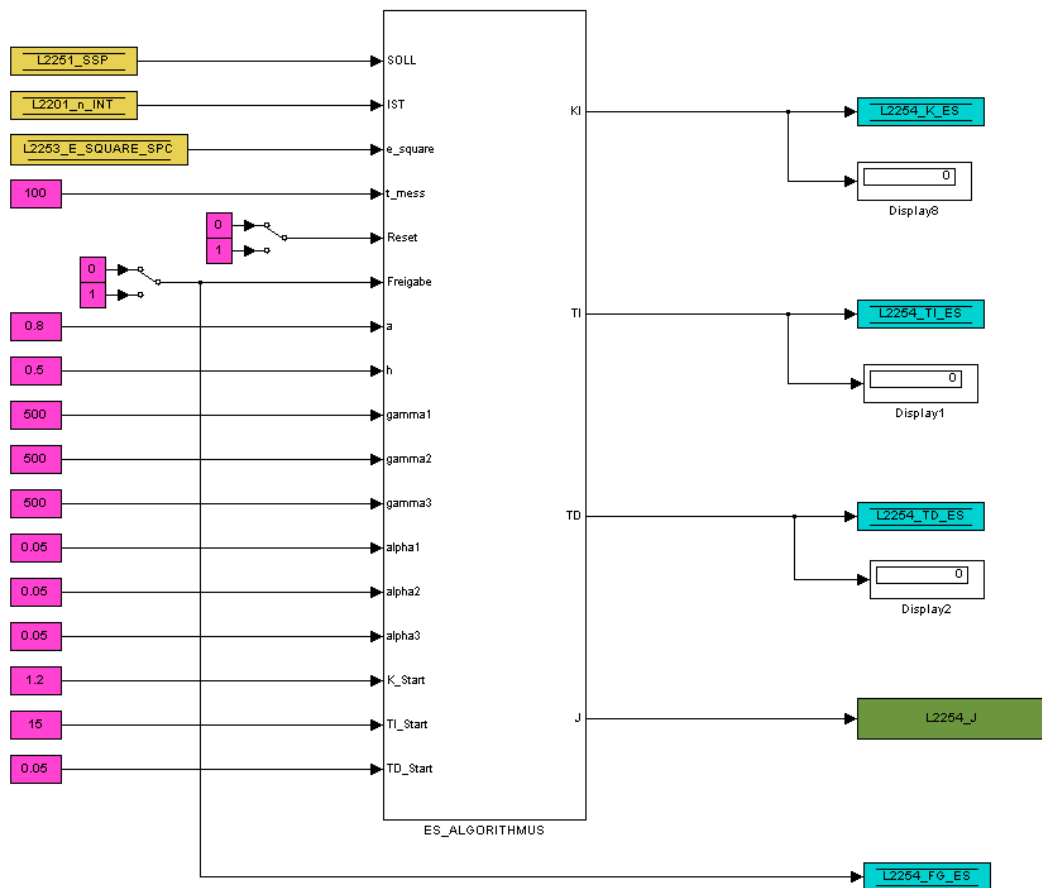


Abbildung 4.2: ES Simulink Block

Eingänge

- *SOLL* → Sollwert
- *IST* → Istwert
- *e_square* → quadratischer Fehler
- *t_mess* → Messzeit
- *Reset* → Rücksetzen aller Werte auf den Urzustand
- *Freigabe* → Optimierungsfreigabe

Parameter

- *a* → dient zur Berechnung der Modulationsfrequenz $\omega_i = a^i\pi$, wobei ($0 < a < 1$)
- *h* → Faktor für den Hochpassfilter ($0 < h < 1$)
- *gamma_1*, *gamma_2*, *gamma_3* → Verstärkungsfaktoren je Reglerparameter
- *alpha_1*, *alpha_2*, *alpha_3* → Amplitudenwert des Testsignals je Reglerparameter
- *K_Start* → Startwert für K_P
- *TI_Start* → Startwert für T_I
- *TD_Start* → Startwert für T_D

Ausgänge

- *K* → neu ermittelter Wert für K_P
- *TI* → neu ermittelter Wert für T_I
- *TD* → neu ermittelter Wert für T_D
- *J* → aktueller Wert der Kostenfunktion J

Die Abbildung 4.3 zeigt die erste Unterebene des ES-Programmblocks.

ES-Unterprogramme

- Ermittlung des Meßstartzeitpunktes und Berechnung der Zielfunktion
- Ausführen des ES-Algorithmus
- Parameterupdate

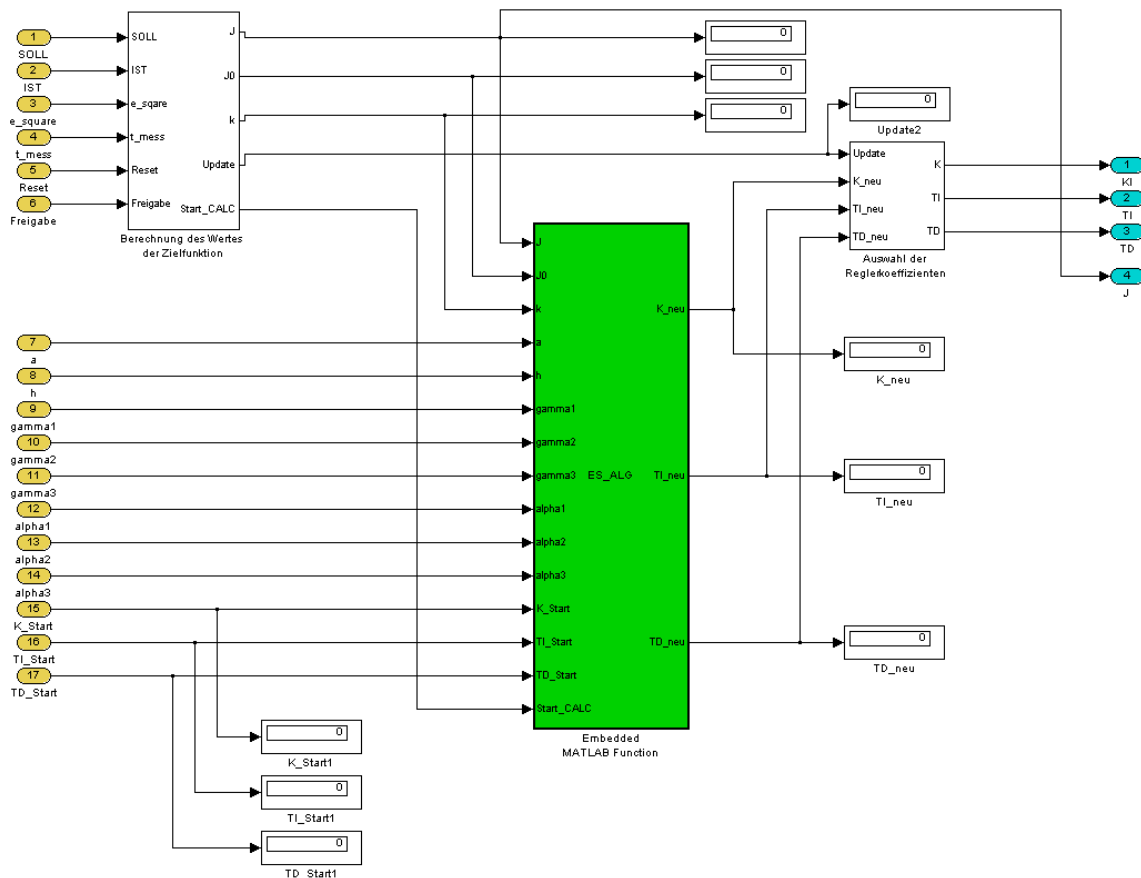


Abbildung 4.3: ES Simulink Block Detailansicht

Ermittlung des Meßstartzeitpunktes und Berechnung der Zielfunktion

Vorraussetzung für einen erfolgreichen Start der Optimierung ist ein stabiler Regelkreis, der ein Überschwingen aufweist. Dies kann z.B. mit einer der in 3.1.2 präsentierten Regeln bewerkstelligt werden. Für das Verfahren sind, wie in 3.2 schon näher beschrieben, die relevanten Zeiten $t_0 = T_{MAX}$ und T zu ermitteln. Die Differenz aus $T - T_{MAX}$ dient als Meßzeit für die Zielfunktion. Das Überschwingen des Regelkreises wird zur Erkennung des Startzeitpunktes der Integration für die Zielfunktion benötigt. In die Ermittlung der Zielfunktion fließt nun jener Anteil ein, der sich zwischen T_{MAX} und T befindet. Ist der Wert der Zielfunktion ermittelt, wird dieser an den Block `ES_ALG` in Kombination mit einer Berechnungsfreigabe (`START_CALC`) übergeben.

Ausführen des ES-Algorithmus

Sind die Werte für J berechnet und die Berechnungsfreigabe (`START_CALC`) vorhanden, wird der Block `ES_ALG` ausgeführt. Wichtig hierbei ist, dass die Startwerte für K , T_I und T_D den Werten entsprechen, die zuvor für den ersten Sollwertsprung am

Regler verwendet wurden. Die Parameter an der Eingangsseite des ES_ALG Blockes wurden oben schon näher beschrieben. Nach erfolgreicher Berechnung liefert der Block die neuen Parameter für K , T_I und T_D .

Parameterupdate

Der Block 'Auswahl der Reglerkoeffizienten' übermittelt bei $Update = 1$ die neuen Parameter an den Regler. Dies erfolgt eine gewisse Zeit vor dem nächsten Sollwertsprung, da mögliche Ausgleichsbewegungen des Regelkreises in die Zeit miteingerechnet werden müssen.

Zum Abbruch des Algorithmus können, wie schon in 3.2.1 erwähnt, mehrere Kriterien verwendet werden, wie z.B. eine maximale Anzahl an Iterationen (k); eine Mindestanforderung an J wurde bereits unterschritten \rightarrow keine weitere Optimierung notwendig; Fehler bei der Optimierung ist aufgetreten (Regelkreis wird instabil (kann bei einer falschen Wahl der Parameter α und γ nicht ausgeschlossen werden)).

4.1.3 Funktionsprüfung im Simulink

Versuche am Drehzahlregler

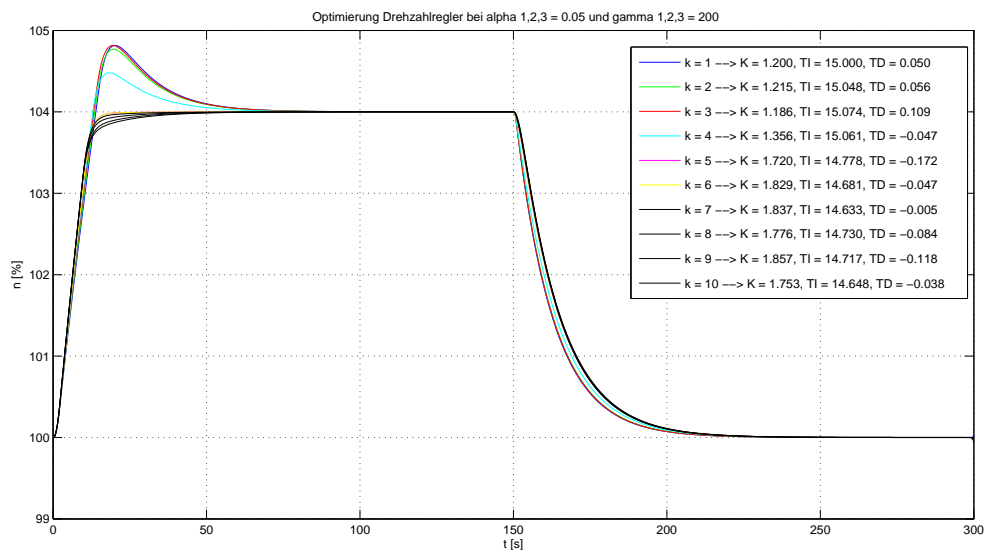


Abbildung 4.4: Optimierung des Drehzahlreglers im Simulink

Abbildung 4.4 zeigt eine Optimierung am Drehzahlregler, wobei die Werte für $\alpha_1, \alpha_2, \alpha_3 = 0.05$ und $\gamma_1, \gamma_2, \gamma_3 = 200$ gewählt wurden. Die Werte für a und h wurden nicht verändert (siehe 3.2.3).

Wie aus dem Verlauf von $J(k)$ ersichtlich, findet der Algorithmus innerhalb weniger Iterationen ein zufriedenstellendes Ergebnis (im vorliegenden Fall nach dem fünften

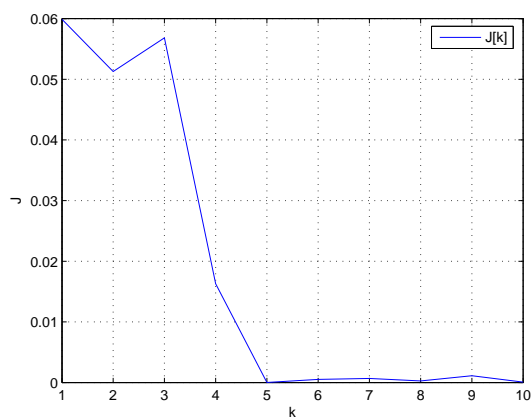


Abbildung 4.5: Verlauf der Kostenfunktion bei Drehzahlregleroptimierung

Versuch).

Versuche am Leistungsregler

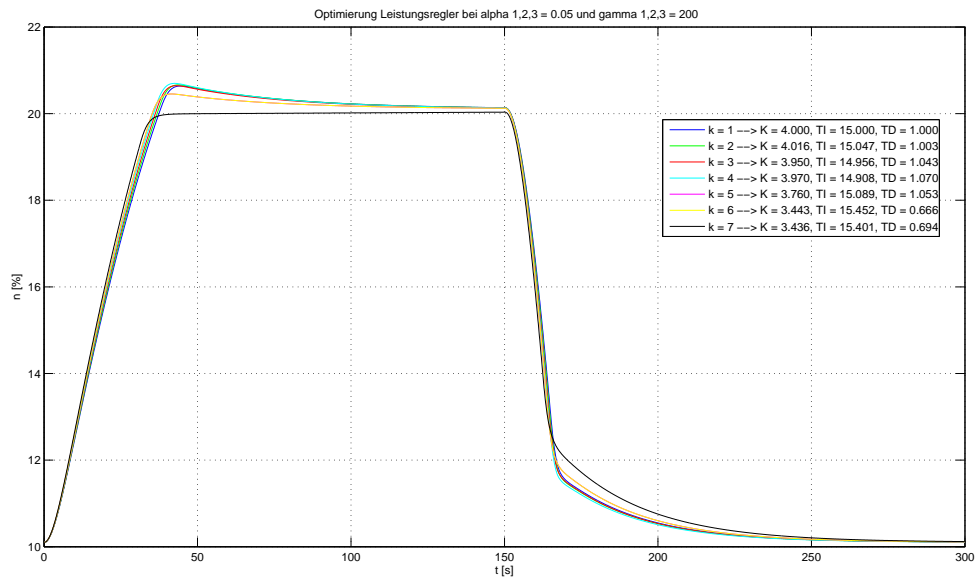


Abbildung 4.6: Optimierung des Leistungsreglers im Simulink

Abbildung 4.6 zeigt eine Optimierung am Leistungsregler. Auch hier wird innerhalb weniger Iterationen eine verbesserte Lösung gefunden.

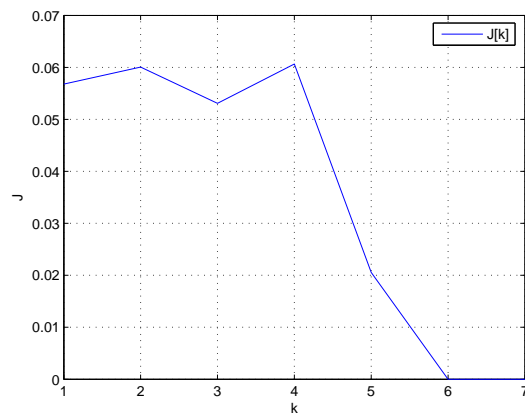


Abbildung 4.7: Verlauf der Kostenfunktion bei Leistungsregleroptimierung

4.2 ES - Implementierung in APROL

Ähnlich wie die Implementierung in Simulink, erfolgt die Implementierung in APROL. Die Subsysteme in Simulink entsprechen sogenannten Hypermakros in APROL.

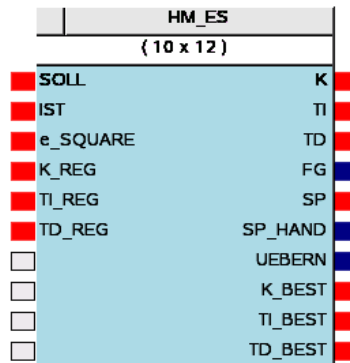


Abbildung 4.8: APROL Hypermakro (beinhaltet die grafische Darstellung inkl. der logischen Auswertung)

Abbildung 4.8 zeigt das fertige Hypermakro, das bei der Softwareerstellung des Turbinenreglers verwendet werden kann. Wie man aufgrund der Ein-/ Ausgänge des Bausteins erkennt, werden für den Betrieb nur die Informationen *SOLL*, *IST*, e^2 , sowie die aktuellen Reglerparameter K_{REG} , T_{I-REG} und T_{D-REG} benötigt. Der Block *HM_ES* beinhaltet eine grafische Benutzeroberfläche, über die das Optimierungsverfahren konfiguriert und gestartet werden kann (siehe Abbildung 4.14), den Algorithmus und diverse Blöcke zur Auswertung der Nebenbedingungen.

Abbildung 4.9 zeigt das 'Herzstück' des Hypermakros *HM_ES*. Im Block *ES_ALG* ist der ES-Algorithmus mittels C-Code implementiert und wird wie schon oben unter 4.1.2 erwähnt mit dem Eingang *START_CALC* gestartet.

In der Turbinenreglersoftware ist der Standard PI Regler durch das Hypermakro *HM_PID* zu ersetzen (siehe Abbildung 4.10). Dieser Block besitzt einen zusätzlichen Eingang für den Parameter T_D und einen Ausgang für das Fehlerquadrat $e_{sq} = e^2$.

Abbildung 4.11 zeigt eine fertige Implementierung des ES-Algorithmus (ohne Adaption des Reglers). Die Variablen auf der linken Seite entsprechen den Eingangsvariablen (werden in APROL als Konnektoren bezeichnet), die rechte Seite entspricht den Ausgangsvariablen.

Die Abbildungen 4.12 und 4.13 zeigen die Realisierung eines Drehzahlreglers mit einem PID-Kern.

Abbildung 4.14 zeigt die Bedienoberfläche für das Extremum Seeking. Generell sind alle schwarzen Analoganzeigen als 'Eingabefelder' bzw. alle grauen Analoganzeigen als 'An-

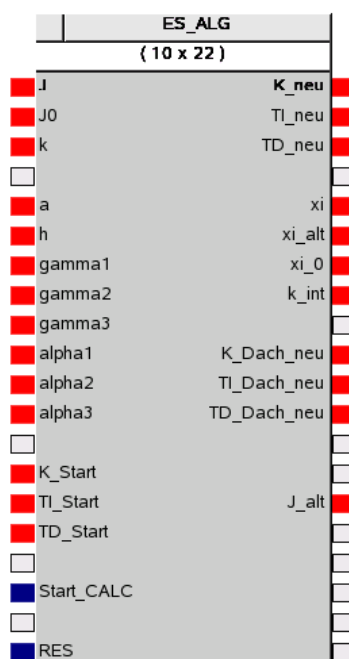


Abbildung 4.9: ES Algorithmus realisiert in APROL

zeigewerte' zu interpretieren.

Abbildung 4.15 zeigt die Bedienleiste des Tools. Diese unterteilt sich in den Bereich der Bedienung (Buttons) und der Statusanzeige.

Buttons

- *Opt. EIN* bzw. *Opt. AUS* → Start- bzw. Stopp der Optimierung
- *Reset* → Rücksetzen aller berechneten Parameter
- *Opt.neg.Fl.* bzw. *Opt.pos.Fl.* → Es kann eine Optimierung bei einem positiven bzw. einem negativen Sollwertsprung gewählt werden. Dies ist vor allem bei Strecken mit unterschiedlicher Dynamik bei pos. bzw. neg. Änderung von Vorteil.
- *SP_HAND* bzw. *SP_AUTO* → Für die Ermittlung der Startparameter ist die Bedienung im Handbetrieb vorteilhaft. In diesem Fall wird das Feld *SOLL_HAND* schwarz hinterlegt und es können div. Sollwertsprünge durchgeführt werden. Im Automatikfall wird das Feld *SOLL_HAND* grau und die Felder *SOLL_POS*, *SOLL_NEG*, *t_ON_POS*, *t_ON_NEG* und *t_EINSCHWING* schwarz hinterlegt. Der Wert *SOLL_POS* gibt den positiven Sollwertsprung an, der Wert *SOLL_NEG* den negativen Sollwertsprung. Über die Parameter *t_ON_POS* bzw. *t_ON_NEG* wird die Dauer des jeweiligen Sprunges definiert. Die Zeit *t_EINSCHWING* dient

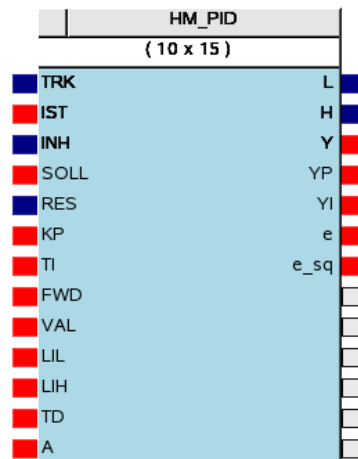


Abbildung 4.10: PID Regler realisiert in APROL

dem Einschwingvorgang nach der Parameteraktualisierung. Dieser Wert wird abhängig von der gewählten Betriebsart (Optimierung positiv oder negativ) von den Zeiten t_{ON_POS} bzw. t_{ON_NEG} abgezogen.

- *Übernahme* → Sind die Parameter ermittelt, können diese mit dem Button Übernahme in die Turbinenreglersoftware als Fixparameter übernommen werden.

Die jeweilige Auswahl der oben angeführten Button's ist über das Anzeigefeld ersichtlich.

Statusanzeigen

- *Optimierung aktiv* bzw. *Optimierung nicht freigegeben*
- *Optimierung für positive Flanke* bzw. *Optimierung für negative Flanke*
- *Sollwertsprung automatisch über ES* bzw. *Sollwertsprung von HAND*

Die Messzeit (Erklärung siehe 4.1.2) muß im Feld *MESSZEIT* eingetragen werden (siehe Abbildung 4.17. Das Feld *ANZAHL DER VERSUCHE - k* ist selbsterklärend (nur im Automatikfall relevant).

Abbildung 4.18 zeigt die einzelnen Ergebnisse der Zielfunktionen. J_0 entspricht dem Wert der Zielfunktion zum Startzeitpunkt; J_{AKT} entspricht dem aktuellen Wert mit den am Regler verwendeten Parametern; J_{BEST} entspricht dem besten Wert der Optimierung.

Abbildung 4.19 zeigt den Bereich für die Eingabe der Optimierungsparameter.

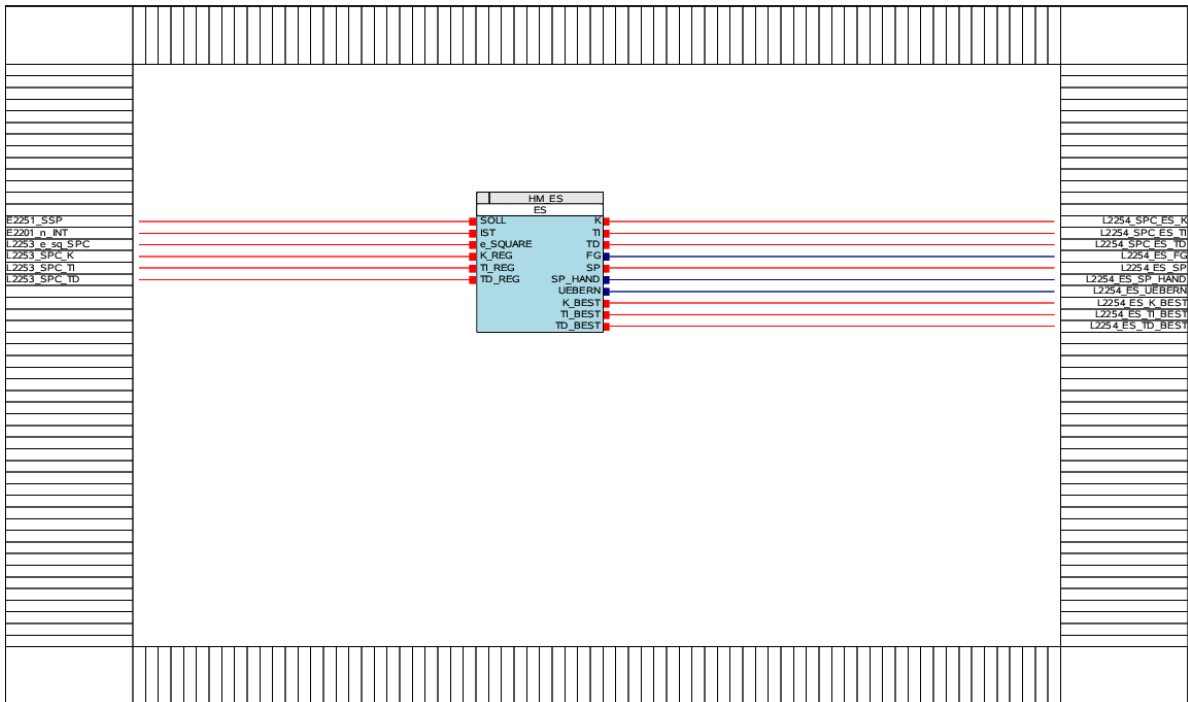


Abbildung 4.11: ES Hypermakro logisch in APROL eingebunden

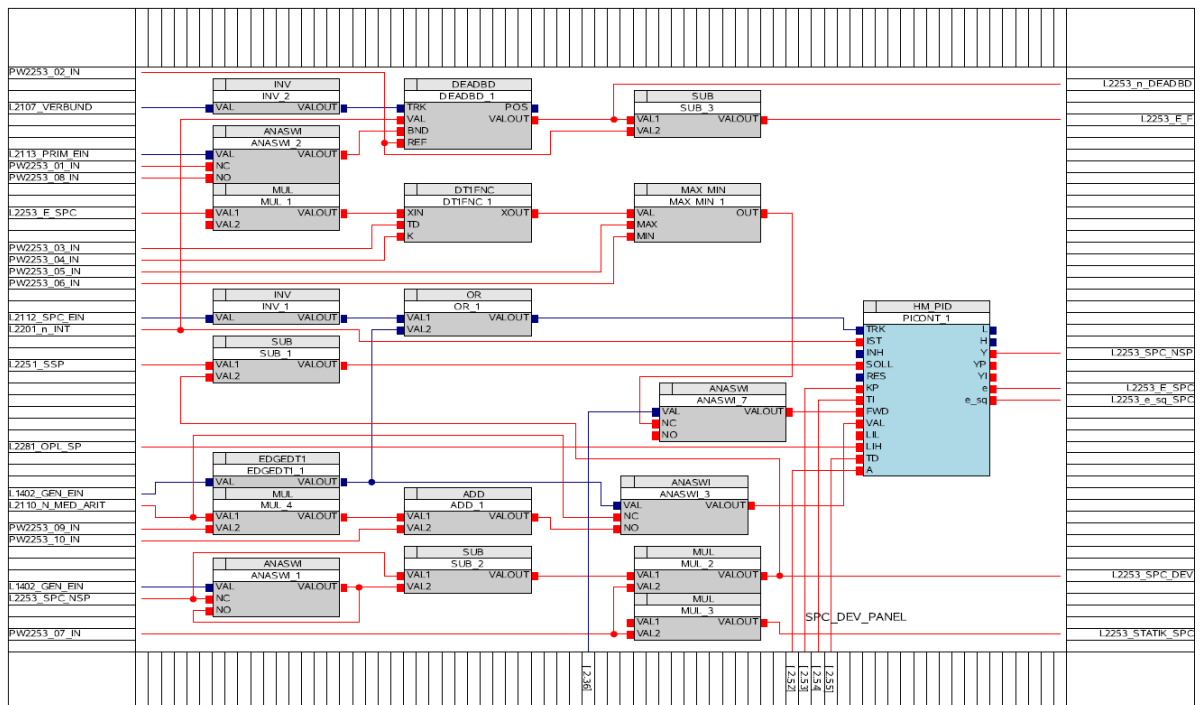


Abbildung 4.12: Drehzahlregler realisiert in APROL (Teil 1)

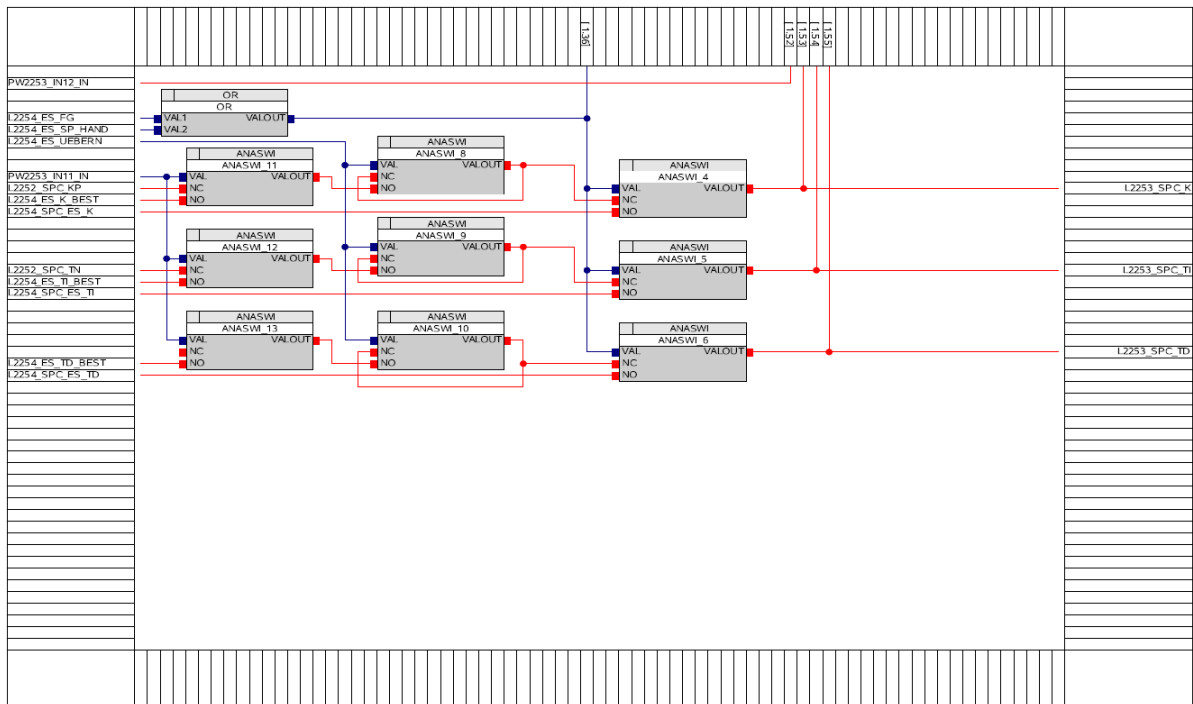


Abbildung 4.13: Drehzahlregler realisiert in APROL (Teil 2)

Abbildung 4.14: ES APROL Bedienung und Visualisierung



Abbildung 4.15: ES APROL Bedienung

SOLL_POS	104.0	t_ON_POS	140.0 s
SOLL_NEG	100.0	t_ON_NEG	80.0 s
		t_EINSCHW	30.0 s
SOLL HAND	100.0	SOLL	104.000
		IST	104.000

Abbildung 4.16: ES APROL Parametrierung der Sollwertsprünge (HAND oder AUTO)

MESSZEIT	100.0 s
ANZAHL DER VERSUCHE - k	15

Abbildung 4.17: ES APROL Messzeit und Anzahl der Versuche

J0	0.1285	k_IST	4
J_AKT	0.0013		
J_BEST	0.1285		

Abbildung 4.18: ES APROL Anzeigen der Werte der Zielfunktionen

OPTIMIERUNGSPARAMETER					
		ALPHA		GAMMA	
a	0.8	1	0.10	1	200.0
h	0.5	2	0.10	2	200.0
		3	0.10	3	200.0

Abbildung 4.19: ES APROL Optimierungsparameter

PID - PARAMETER			
START	BERECHNET	AKTUELL AM REGL.	BESTES ERGEBNIS
K 1.0000	K 2.1494	K 2.1494	K 1.0000
TI 5.0000	TI 4.6366	TI 4.6366	TI 5.0000
TD 0.0500	TD -1.5385	TD -1.5385	TD 0.0500

Abbildung 4.20: ES APROL Optimierungsparameter

Abbildung 4.20 zeigt die einzelnen Reglerparameter. Dargestellt sind:

- Startparameter
- Berechnete Parameter aus ES
- aktuell verwendete Parameter am Regler
- Parametersatz mit dem besten Ergebnis

Die Abbildung 4.21 zeigt die Darstellung der Trends im TrendViewer. Dieser Viewer dient der grafischen Analyse der einzelnen Analogwerte und wird auch zu Dokumentationszwecken verwendet. Über eine gerasterte Abfrage können die aufgezeichneten Daten über *csv* exportiert und mit MS-Excel oder Matlab weiterverarbeitet werden.

Eine zweite Möglichkeit der Dokumentation zeigt Abbildung 4.22.

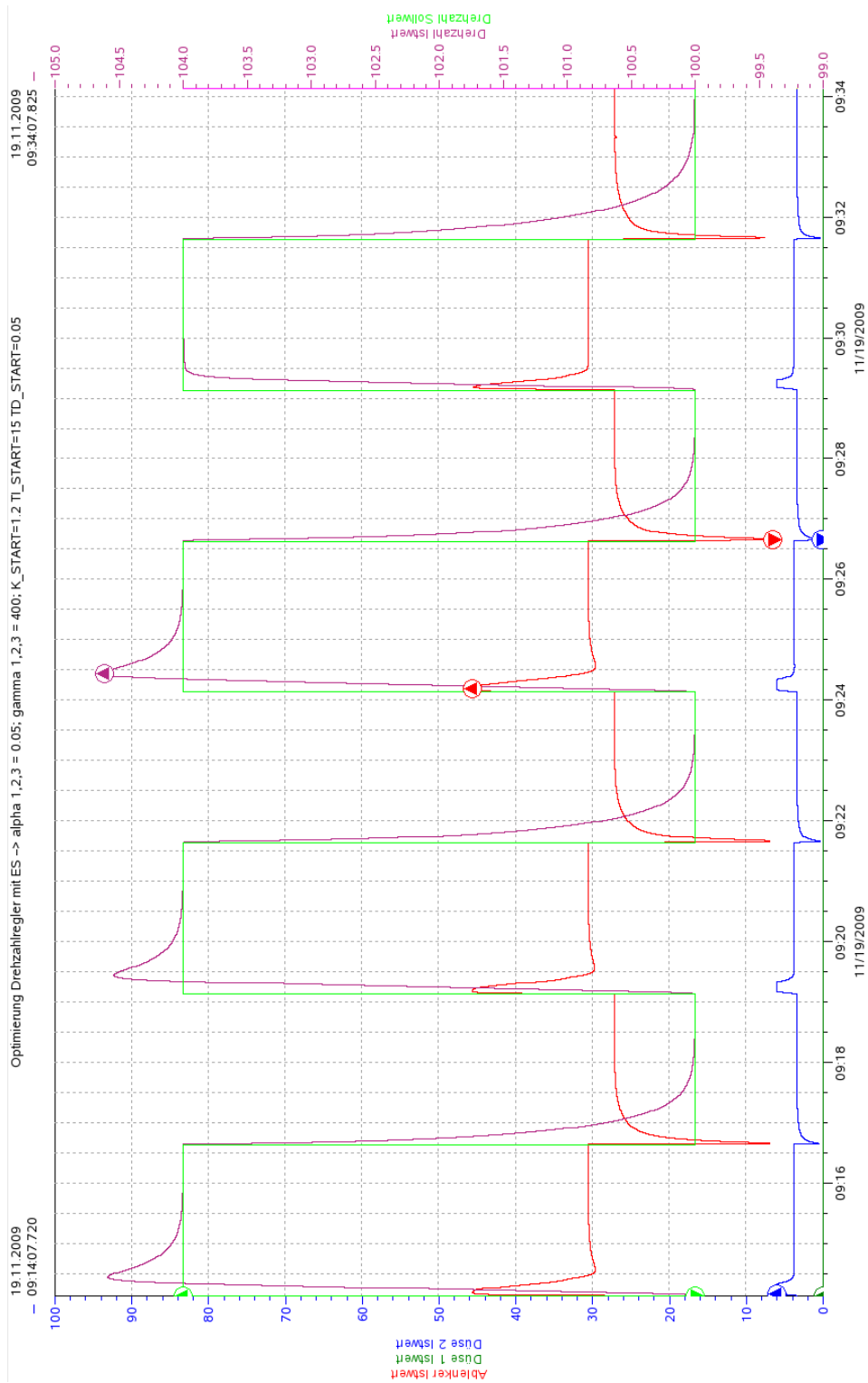


Abbildung 4.21: APROL Trendviewer

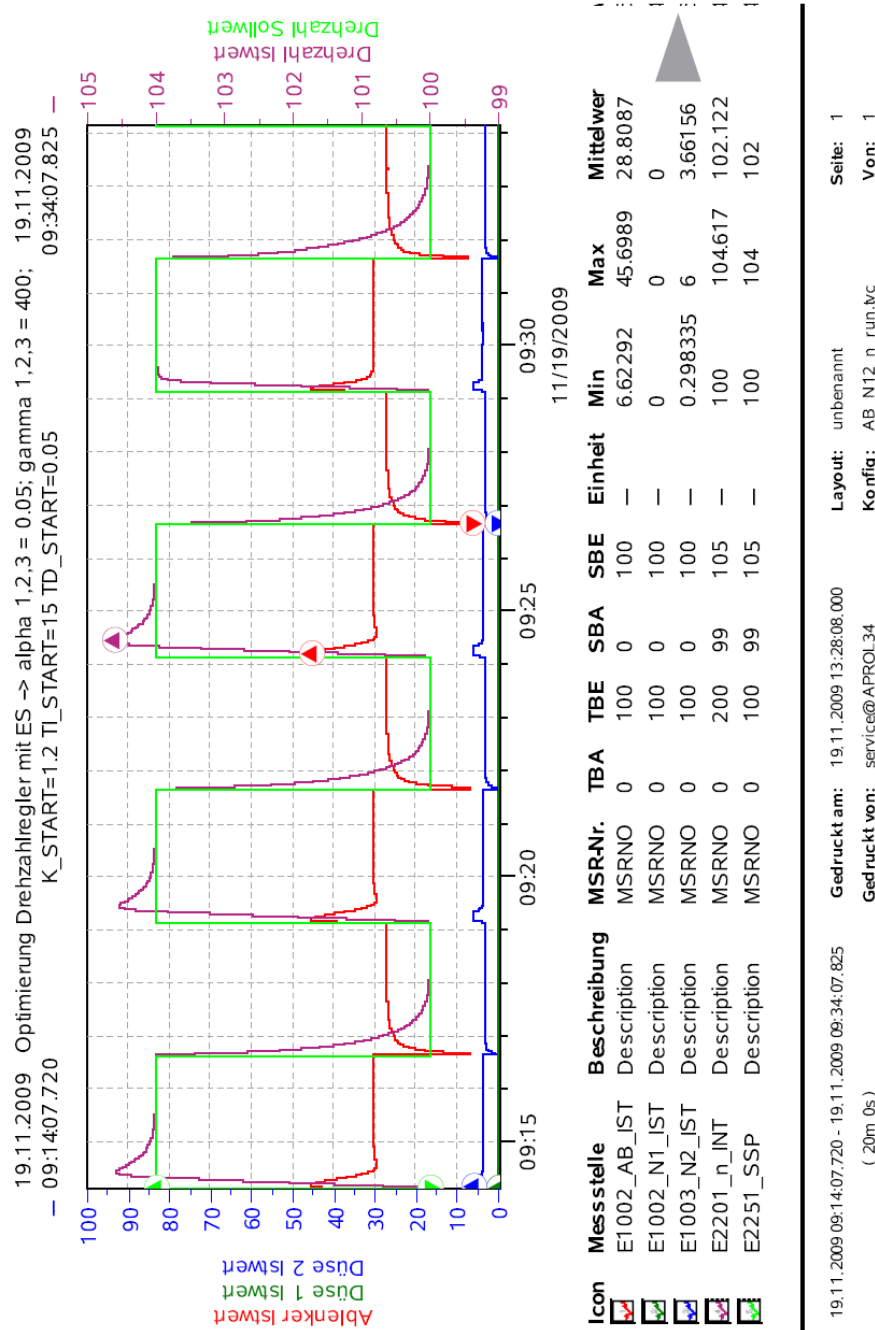


Abbildung 4.22: APROL Trendviewer (Möglichkeit der Dokumentation)

4.2.1 Funktionsprüfung im APROL

Versuche am Drehzahlregler

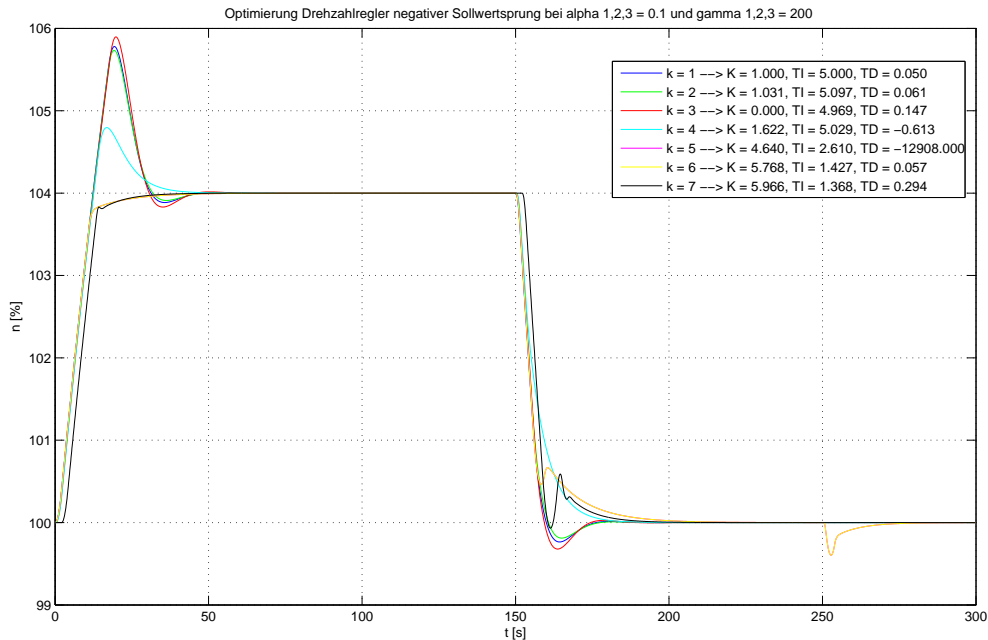


Abbildung 4.23: Optimierung des Drehzahlreglers im APROL

Abbildung 4.23 zeigt eine Optimierung am Drehzahlregler (bei positivem Sollwertsprung), wobei die Werte für $\alpha_1, \alpha_2, \alpha_3 = 0.1$ und $\gamma_1, \gamma_2, \gamma_3 = 200$ gewählt wurden. Die Startwerte von K , T_I und T_D wurden zu $K = 1$, $T_I = 5$ und $T_D = 0.05$ gewählt. Die Werte für a und h wurden, wie auch bei der Simulation in Simulink, nicht verändert (siehe 3.2.3).

Wie aus dem Verlauf von $J(k)$ (siehe Abbildung 4.24) ersichtlich, findet der Algorithmus auch in APROL innerhalb weniger Iterationen ein zufriedenstellendes Ergebnis.

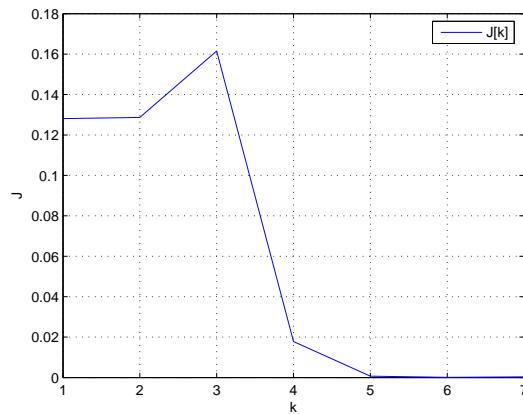


Abbildung 4.24: Verlauf der Zielfunktion bei Drehzahlregleroptimierung in APROL (pos. Sollwertsprung)

Abbildung 4.25 zeigt eine Optimierung am Drehzahlregler (bei negativem Sollwertsprung). Die Startwerte von K , T_I und T_D wurden zu $K = 1$, $T_I = 5$ und $T_D = 0.05$ gewählt.

Bei einem Vergleich der Ergebnisse für den positiven und den negativen Sollwertsprung ist ersichtlich, dass eine mögliche Umschaltung der Reglerparameter für jeden Fall (positiver und negativer Sprung) von Vorteil wäre. Ansonsten muss ein Mittelweg für die Reglerparameter gefunden werden.

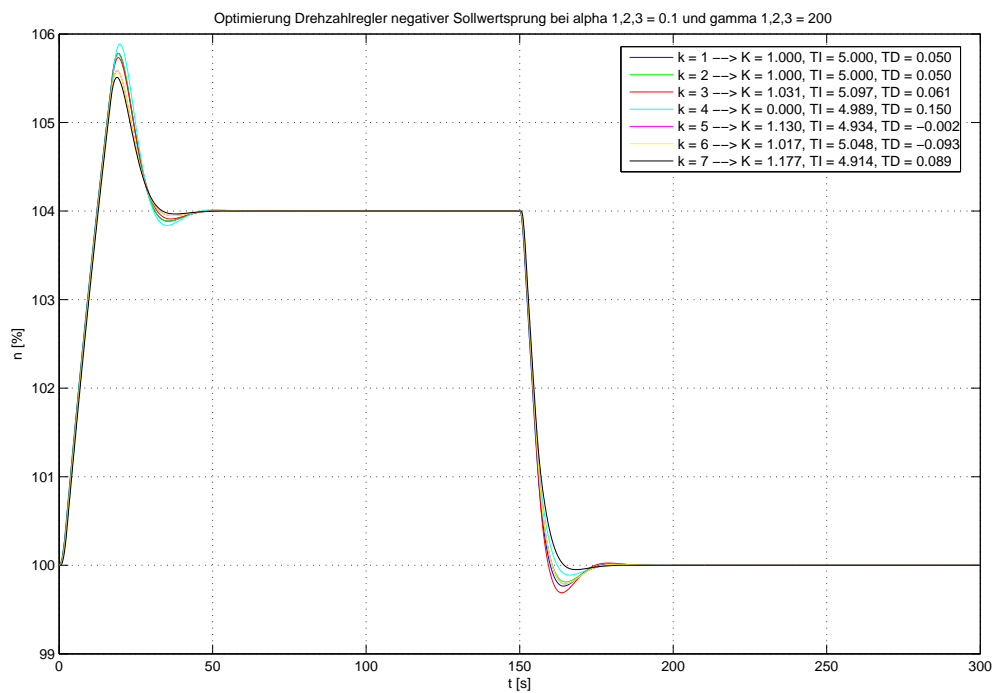


Abbildung 4.25: Optimierung des Drehzahlreglers im APROL

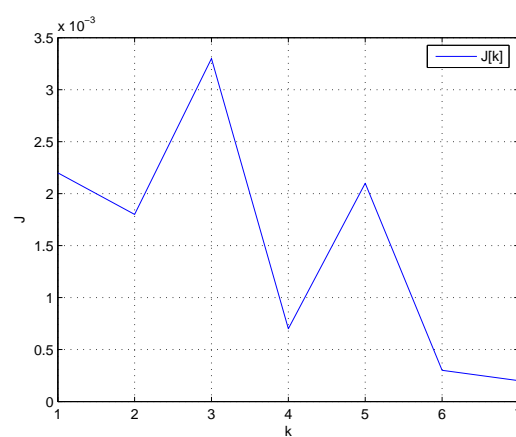


Abbildung 4.26: Verlauf der Zielfunktion bei Drehzahlregleroptimierung in APROL (neg. Sollwertsprung)

Anhang A

Parametrierung des APROL Turbinenmodells

A.1 Parameter für Ablenker und Düsen

Die Parameter für das Turbinenmodell wurden aus den Anlagenkenndaten bzw. aus den Inbetriebnahmekurven entsprechend ermittelt. Die Ablenkerzeiten ergeben sich zu:

$$t_{AB_Auf} = 3,9s$$

$$t_{AB_Zu} = 1,2s$$

Diese Verstellzeiten werden auf %/s normiert, was in weiterer Folge die entsprechenden Parameterwerte für das Modell (sowohl Simulink als auch APROL) liefert. Die Umrechnung erfolgt mit Hilfe der Formel 2.3

Damit ergibt sich der Parameterwert für

$$t_{AB_Auf} \rightarrow PW2901_01 = 25,64$$

und der Wert für die Schließzeit des Ablenkers

$$t_{AB_Zu} \rightarrow PW2901_02 = -83,33.$$

Für die Berechnung der Düsenzeiten gilt die gleiche Vorgehensweise. Da die beiden Düsen ähnliches Verhalten besitzen, wurden für diese die gleichen Parameter verwendet.

Die Düsenzeiten ergeben sich zu:

$$t_{Nx_Auf} = 138s$$

$$t_{Nx_Zu} = 36s$$

Aus den Öffnungs- bzw. Schließzeiten der Düsen erhält man mit Hilfe der Formel 2.3 folgende Parameter:

$$t_{Nx_Auf} \rightarrow PW2901_05 = 0,725$$

$$t_{Nx_Zu} \rightarrow PW2901_06 = 2,778$$

Die Konnektoren $L24xx$, die mit dem Block MIN_MAX auf der linken Seite der Abbildungen A.1 und A.2 begrenzt sind, kommen aus den jeweiligen Positionierkreisen. Die Parameter $PW2901_04_IN$, $PW2901_07_IN$ und $PW2901_08_IN$ dienen als mögliche Offsets für die Düsen bzw. dem Ablenker. Der Parameter $PW2901_03_IN$ wird als Integrationszeitkonstante (TI) bei allen Integratoren verwendet. Der Wert hierfür ist vorzugsweise = 1.

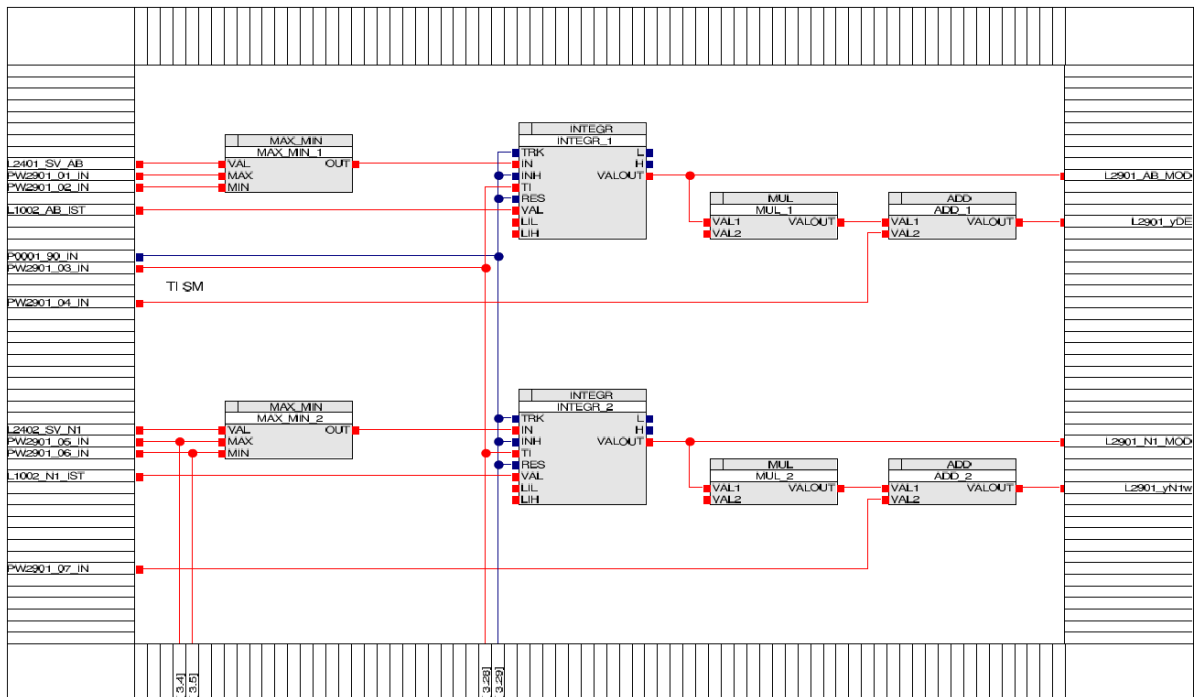


Abbildung A.1: Modell für Ablenker und Düse realisiert in APROL (L2901_SM)

Die oben beschriebenen Modelle entsprechen dem für die Verstellung der einzelnen Organe verwendeten Servomotoren.

Eine Zuordnung zwischen dem Servohub des Ablenkers und der tatsächlichen Position von diesem kann mit einer Polynomfunktion 5. Ordnung (siehe Abbildung A.3) berücksichtigt werden. Im vorliegenden Fall gilt für $y = x$ ($a_1 = 1$ der Rest 0).

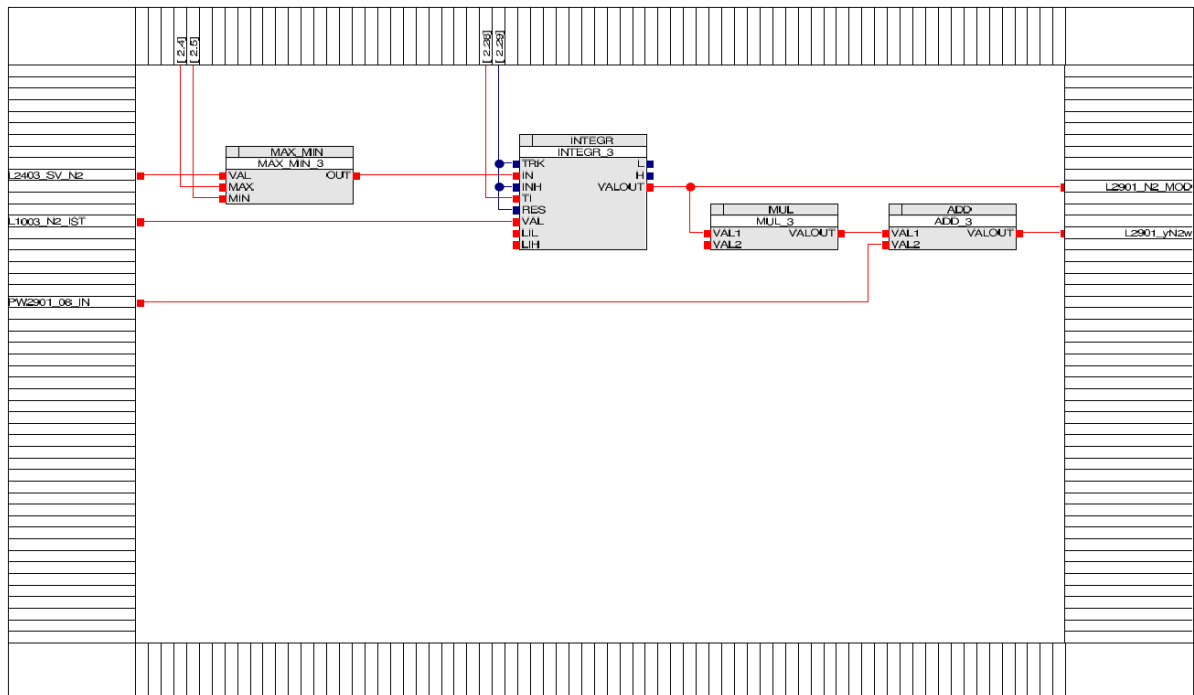


Abbildung A.2: Modell für Ablenker und Düse realisiert in APROL (L2901_SM)

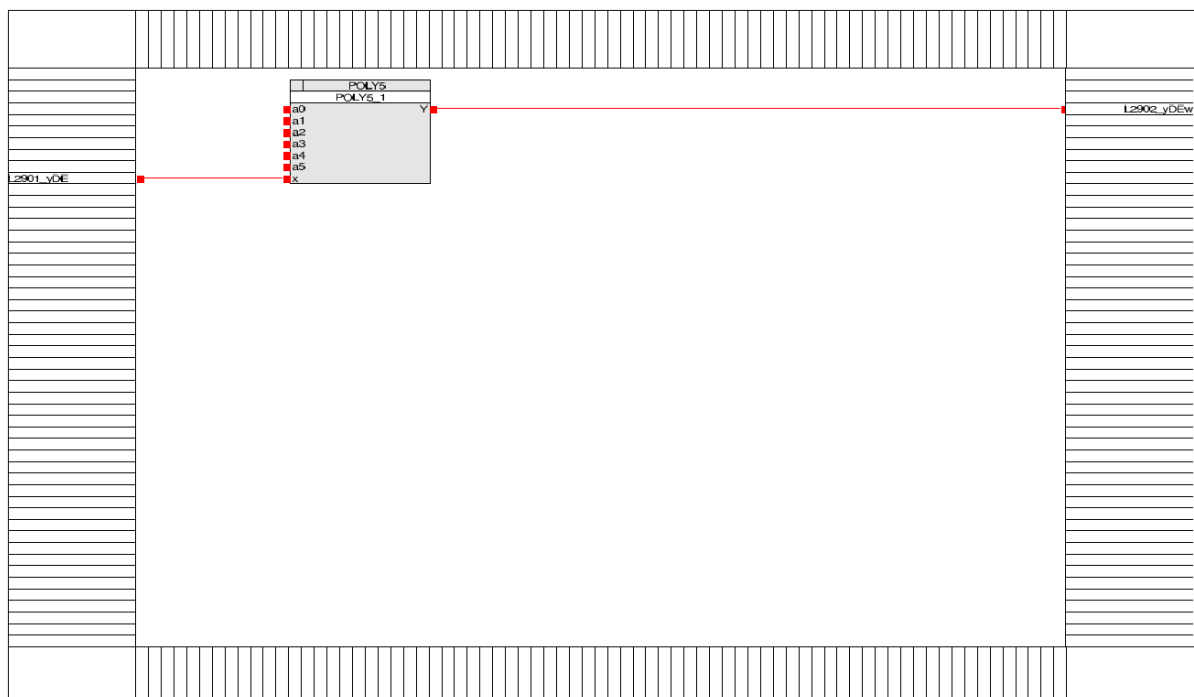


Abbildung A.3: Normierung Ablenkerposition realisiert in APROL (L2902_NORM)

A.2 Parameter für die Durchflussberechnung (q)

Für die Berechnung des Durchflusses wird in unserem Fall eine Polynomfunktion 2. Ordnung verwendet. Diese stellt den Zusammenhang von Durchfluss zu Düsenöffnung dar.

Aus Abbildung 2.5 ergeben sich folgende Werte für die Parameter K_1 und K_2 :

$$K_1 = 1,496$$

$$K_2 = 0,5$$

Für die Berechnung des Durchflusses der einzelnen Düsen werden noch zwei weitere Parameter benötigt. $PW2903_01_IN$ dient als Stützwert (zs) für die Drehzahl. Dieser wird als zusätzliche Korrekturmöglichkeit für den Wert zw verwendet, der in Folge für die Ermittlung des Planschmoments bzw. des Drehzahleinflussfaktors benötigt wird. zs ist in der Regel 1.

$$PW2903_01_IN = 1$$

$PW2903_02_IN$ bildet das Verhältnis aus $frac{q_{1s} z_0$, wobei q_{1s} = Stützwert und z_0 die Düsenzahl wiedergibt. Wahl von

$$PW2903_02_IN = 0.25$$

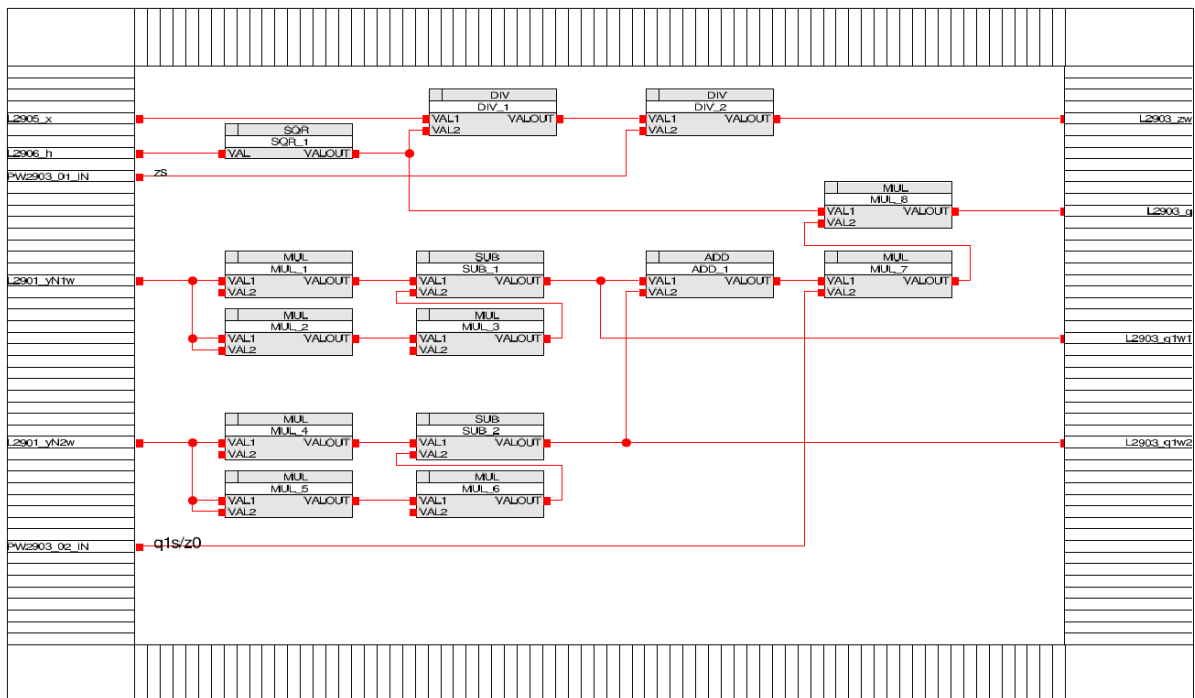


Abbildung A.4: Berechnung des Durchflusses realisiert in APROL (L2903_q_CALC)

A.3 Parameter für die Momentenberechnung (m)

Verwendete Parameter:

$$\begin{aligned}
 PW2904_01_IN &= a_{e0} \\
 PW2904_02_IN &= a_{e1} \\
 PW2904_03_IN &= e_0 \\
 PW2904_04_IN &= m_{1s}/z_0 \\
 Poly5 : f_m &: f_m(e_s) \\
 Poly5 : m_{P1} &: m_{P1}(z_w) \\
 Poly5 : m_z &: m_z(z_w)
 \end{aligned}$$

Bei der Berechnung des Drehmoments werden mehrere Faktoren berücksichtigt. Über eine Polynomfunktion 2. Ordnung (Drehmoment m in Abhängigkeit der Düsenposition) wird der theoretische Momenteneinfluss / Düse (ohne Drehzahleinfluss, Ablenkereinfluss und Einfluss des hydraulischen Bremsmoments) ermittelt.

Die Koeffizienten $K_1 = 1,8$ und $K_2 = 0,77$ sind als fixe Größen in den Multiplikatoren (siehe Abbildung A.5) eingetragen. Als nächstes wird die Position (normierte) des Ablenkers es im Strahl ermittelt. Zur Ermittlung dienen die Parameter $a_{e0} = 0,85$, $a_{e1} = 0,056$ und $e_0 = 1,47$.

Über den Zusammenhang $f_m(e_s)$ (Abbildung 2.6)(Polynom 5. Ordnung) erhält man den Gewichtungsfaktor f_m für den jeweiligen Strahl. Dieser multipliziert mit dem theoretischen Einfluss / Düse ergibt den Momentenbeitrag / Düse (wobei e_s zwischen 0,4 und 0,94 begrenzt ist).

$$f_m = -51,151 e_s^5 + 172,32 e_s^4 - 237,86 e_s^3 + 167,69 e_s^2 - 57,619 e_s + 7,5526 \quad (\text{A.1})$$

Die Summe der Momentenbeiträge der einzelnen Düsen wird mit dem Parameter $PW2904_04_IN$ (bildet das Verhältnis aus $frac{m_{1s}}{z_0}$, wobei m_{1s} = Stützwert und z_0 die Düsenzahl wiedergibt) skaliert. Wahl von

$$PW2904_04_IN = 0.5$$

Dieser Wert wird mit dem Drehzahleinflussfaktor m_z gewichtet und das Planschmoment vom Ergebnis abgezogen. m_z erhält man wiederum aus einer Polynomfunktion 5. Ordnung aus Abbildung 2.7.

$$m_z = 0,161 z_w^5 - 0,435 z_w^4 + 0,098 z_w^3 + 0,013 z_w^2 - 0,275 z_w + 1,4 \quad (\text{A.2})$$

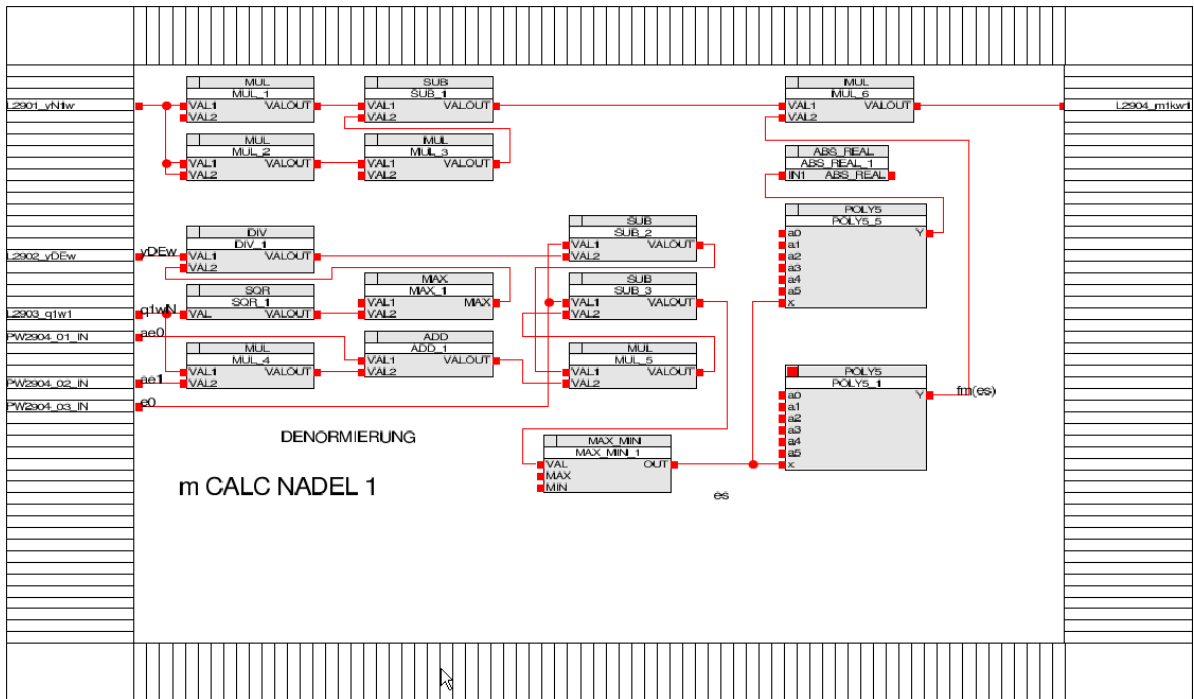


Abbildung A.5: Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)

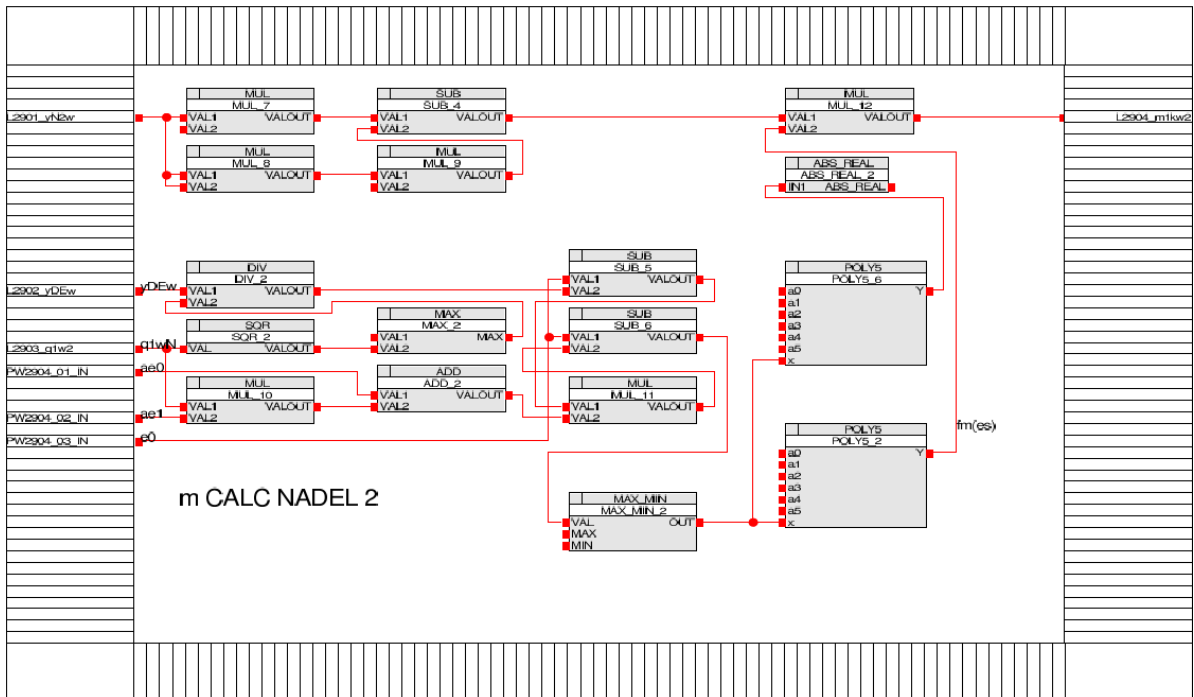


Abbildung A.6: Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)

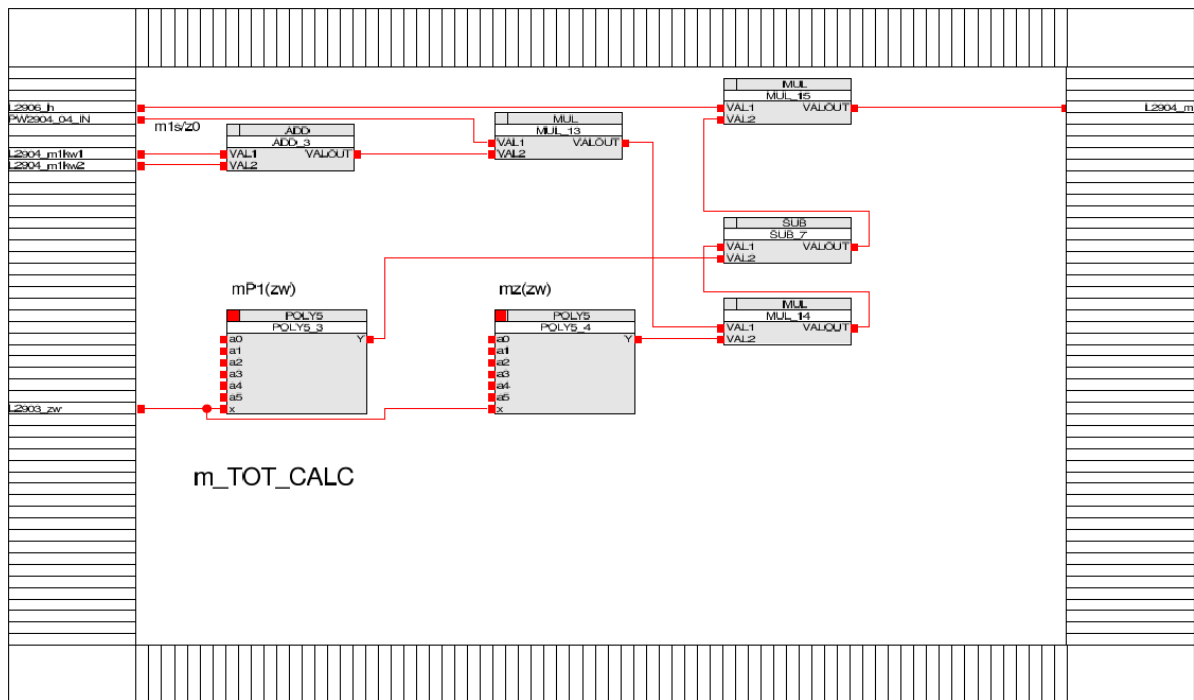


Abbildung A.7: Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)

Für das Planschmoment hat sich folgende Faustformel als praktisch erwiesen:

$$m_{p1} = 0,03 z_w^2 \quad (\text{A.3})$$

In der Software wurde dies mit einer Polynomfunktion 5. Ordnung realisiert. Die Koeffizienten außer $a2$ sind dementsprechend 0.

A.4 Parameter für das Modell der Turbine und des Generators

Verwendete Parameter:

$$PW2905_01_IN = \text{Gegenmoment}$$

$$PW2905_02_IN = T_A$$

$$PW2905_03_IN = P_{Insel}$$

$$PW2905_04_IN = P_{nom}$$

$$PW2905_05_IN = C_{0eg}$$

$$PW2905_06_IN = C_{01eg}$$

$$PW2905_07_IN = C_{1eg}$$

$$PW2905_08_IN = \text{Bremsmoment bei Bremse EIN oder } n < 50\%$$

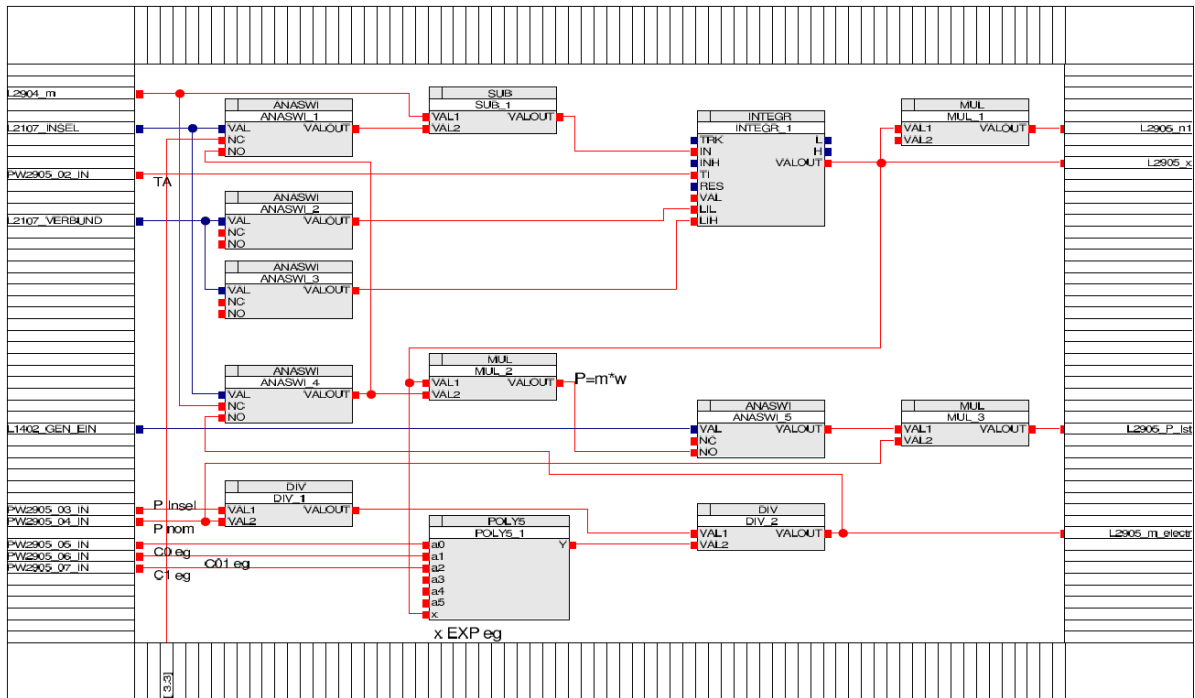


Abbildung A.8: Modell für Turbine und Gen. realisiert in APROL (L2905_MOD_TU_GEN)

Der Parameter $PW2905_01_IN$ dient als Korrekturwert des berechneten Moments (in unserem Fall 0). Mit dem Parameter $PW2905_08_IN$ kann das Bremsmoment berücksichtigt werden.

Die Drehzahl wird über den in Abbildung A.8 dargestellten Integratorblock berechnet. Der Eingang des Integratorblocks ist das in Punkt 2.1.4 berechnete Moment, abzüglich dem elektrischem Moment, dass im Leerlauf = 0 ist. Dieser Integrator hat die Zeitkonstante T_A ($PW2905_02_IN$), die die Schwungmasse der Maschine widerspiegelt. T_A muß auf die verwendete Maschine eingestellt werden, da es ansonsten zu Fehlinterpretationen der einzelnen Betriebszustände kommen kann.

A.5 Parameter für das Modell Pegel und Fallhöhe

Verwendete Parameter:

$$\begin{aligned}
 PW2906_01_IN &= Q_{IN} \\
 PW2906_02_IN &= GRD_Q \\
 PW2906_03_IN &= Q_{nom} \\
 PW2906_04_IN &= q_{prop_TW} \\
 PW2906_05_IN &= T_{W_Offset} \\
 PW2906_06_IN &= h_{nom} \\
 PW2906_07_IN &= h_{min} \\
 PW2906_08_IN &= T_D
 \end{aligned}$$

$$PW2906_09_IN = K_{TD}$$

$$PW2906_11_IN = T_{I_HW}$$

$$PW2906_12_IN = L_{IL_HW}$$

$$PW2906_13_IN = L_{IH_HW}$$

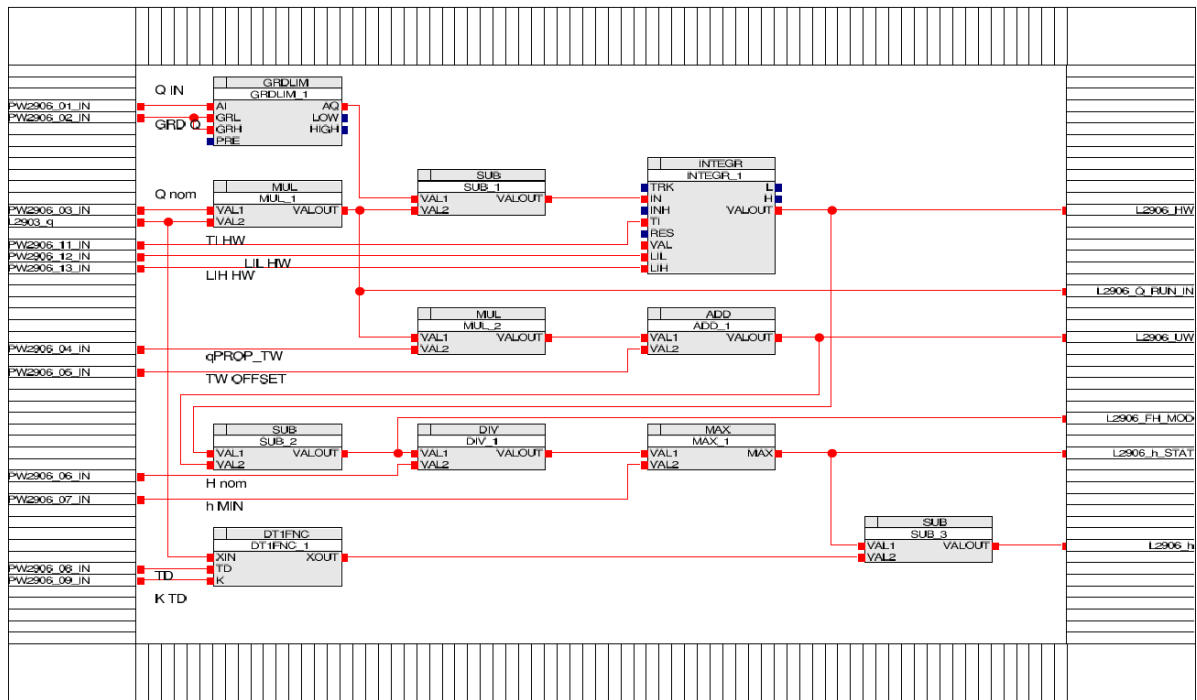


Abbildung A.9: Modell für Pegel und Fallhöhe realisiert in APROL (L2906_LEVEL)

Dieses Modell simuliert die Wasserführung im Staubecken. $PW2906_01_IN$ (Q_{IN}) ist der Zufluss in m^3/s . Dieser wurde für die Versuche entsprechend hoch gewählt ($PW2906_01_IN = 10$), sodass sich das simulierte Staubecken nicht während der Simulation entleert. $PW2906_02_IN = 1$ gibt die Änderungsrate des Zulaufes an.

Der Parameter $PW2906_03_IN(Q_{nom}) = 7,6 m^3/s$ ergibt sich aus Abbildung 2.14. Der Wert bei 100% Öffnung wird sinngemäß mit 2 multipliziert (2 düsige Peltonmaschine).

Der Parameter $PW2906_11_IN(T_{IHW})$ steht für die Oberfläche des Staubeckens und dient als Zeitkonstante für den Integrator zur Auffüllung des Staubeckens. $PW2906_11_IN = 1000 \rightarrow$ gewählt.

Die Parameter $PW2906_12_IN(L_{ILHW})$ und $PW2906_13_IN(L_{IHWW})$ sind die Begrenzungen des Oberwasserpegels (Eingabe in müM).

Für den Parameter $PW2906_05_IN(T_{WOffset})$ wird der Unterwasserpegel (Kote der Maschinenachse) verwendet. Die Normfallhöhe wird in $PW2906_06_IN(H_{nom})$ berücksichtigt und auf die normierte Minimalfallhöhe $h_{MIN}(PW2906_07_IN)$ nach unten begrenzt.

Da in unserem Fall die Normfallhöhe bekannt ist, ($PW2906_06_IN = 587,5$) wurden die Werte für $PW2906_12_IN = 1044,5$, $PW2906_13_IN = 1072$ und $PW2906_05_IN = 474,5$ gewählt.

Mit den Parametern $PW2906_08_IN(T_D)$ und $PW2906_09_IN(K_{TD})$ wird die Anlaufzeit der Wassermassen berücksichtigt. (Wahl von $T_D = 1,5$ und $K = 1$).

Achtung: T_D richtig wählen (klein aber aufgrund von Stabilitätsproblemen nicht zu klein).

Anhang B

Regler - Code und Einstellungen

B.1 Extremum Seeking realisiert im Matlab

B.1.1 Simulinkmodell der Strecke und des Reglers

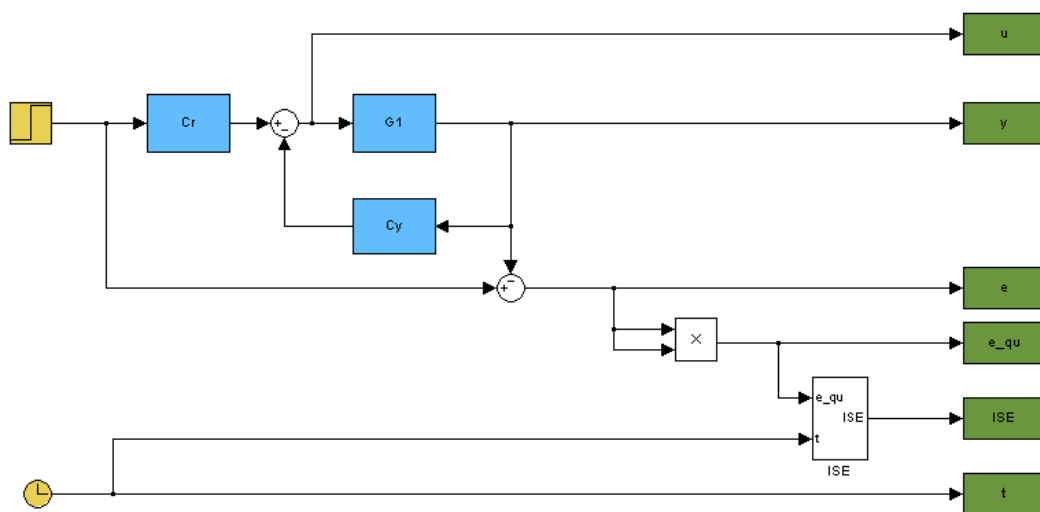


Abbildung B.1: Simulinkmodell zur Berechnung von ES

B.1.2 Matlabcode Extremum seeking

code: ES_Algorithmus_20091110_PID.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1
% 2
% Extremum Seeking Algorithmus für PID laut Killingsworth 3
% 4
% Umgesetzt durch Markus Petschauer 5
% 6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 7
% 8
% Initvariablen 9
% 10
clear all; 11
close all; 12
clc; 13
% 14
J_old= 0; 15
J = 0; 16
gsi_old = 0; 17
k = 0; 18
c = colormap; 19
% 20
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 21
% 22
% Parametersatz für den Algorithmus 23
% 24
h = 0.5; 25
a = 0.8; 26
gamma1 = 200; 27
alpha1 = 0.1; 28
gamma2 = 1200; 29
alpha2 = 1; 30
gamma3 = 200; 31
alpha3 = 0.1; 32
omega1 = a^1 * pi; 33
omega2 = a^2 * pi; 34
omega3 = a^3 * pi; 35
% 36
% Reglerparameter für G1 (ermittelt aus dem Ziegler–Nichols (closed loop) 37
K = 4.158; 38
TI = 9.2; 39
TD = 2.208; 40
% 41
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 42
% 43
% Berechnen der Zielfunktion für k=0 --> Startwerte für den Algorithmus 44
sim('ES_Testmodell_20091108'); 45
% 46
ISE_MESS = find( ISE); % Ermitteln der Werte ungleich 0 aus dem quadratischen Fehler 47
ISE_START = ISE_MESS(1); % Ermitteln des Startzeitpunktes der Berechnung des ISE 48
t_START = t(ISE_START); 49
% 50
J = ISE(end)/(t(end) - t_START); 51
J_0 = J; 52
xi_0 = 1/(1+h)*J; 53
J(1) = J; 54
J(2) = J; 55
% 56
xi(k+1) = -(h * xi_0) + J_0; % Berechnung von xi_1 57
% 58
%-----% 59
% Ermitteln von K_1 60
Kv_Dach(1) = K; 61
% 62
% 63

```



```

% grafische Ausgabe
x=0;
x = 0:1:k-2;
subplot(2,2,3)
plot(x,J(1,2:1:k));
grid on
xlabel('k');
ylabel('J');
legend('J');

subplot(2,2,4)
plot(x,Kv(1,2:1:k),'r')
hold on
grid on
plot(x,Tlv(1,2:1:k),'g')
plot(x,TDv(1,2:1:k))
xlabel('k');
ylabel('K_P,T_I,T_D');
legend('K_P','T_I','T_D');

```

132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

Anhang C

Detailbeschreibungen zur Software

C.1 Bausteinbeschreibungen

C.1.1 PID-Regler im Simulink

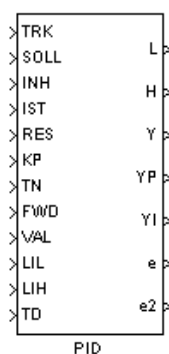


Abbildung C.1: PID-Regler realisiert in Simulink

Der PID-Regler aus Abbildung C.1 verwendet im Standardregelfall die Eingänge *SOLL* bzw. *IST* zur Berechnung des Regelfehlers. Ist ein Stoppen des Integrators aufgrund einer logischen Verknüpfung erwünscht, kann dies mit dem Eingang *INH* realisiert werden (Integratorstopp bei $INH = 1$). Die Reglerkoeffizienten werden über die Parameter K_p , T_N bzw. T_D vorgegeben. Die Eingänge *LIH* und *LIL* dienen der Begrenzung des Reglerausganges. Befindet sich der Regler im Nachlauf, wird der Eingang *VAL* über den Eingang *TRK* an den Ausgang weitergegeben. Eine zusätzliche Störgrößenaufschaltung ist über den Eingang *FWD* möglich (wurde früher auch zur Anbindung des D-Anteiles verwendet). Der Eingang *RES* dient zum Rücksetzen des Integratorblocks.

Die Ausgänge *L* bzw. *H* wechseln in den logischen Zustand 1, wenn das jeweilige Limit des Reglerausganges ($L \rightarrow$ unteres Limit bzw. $H \rightarrow$ oberes Limit) erreicht ist. Der Ausgang *Y* stellt den Reglerausgang dar. Die Ausgang *YP4* zeigt den aktuellen Propor-

tionalanteil, Y_I zeigt den aktuellen Integralanteil des Reglerausganges. Mit e wird der Regelfehler, mit e_2 der quadratische Regelfehler bezeichnet.

C.1.2 PID-Regler im APROL

Der Aufbau des PID-Reglerblocks entspricht dem Simulinkblock mit einem zusätzlichen nach außen geführten Eingang für den Verstärkungsfaktor A des Totzeitgliedes (DT_1). Der Defaultwert liegt bei $A = 0.5$ und liefert zufriedenstellende Ergebnisse. Abbildung zeigt die Realisierung in APROL.

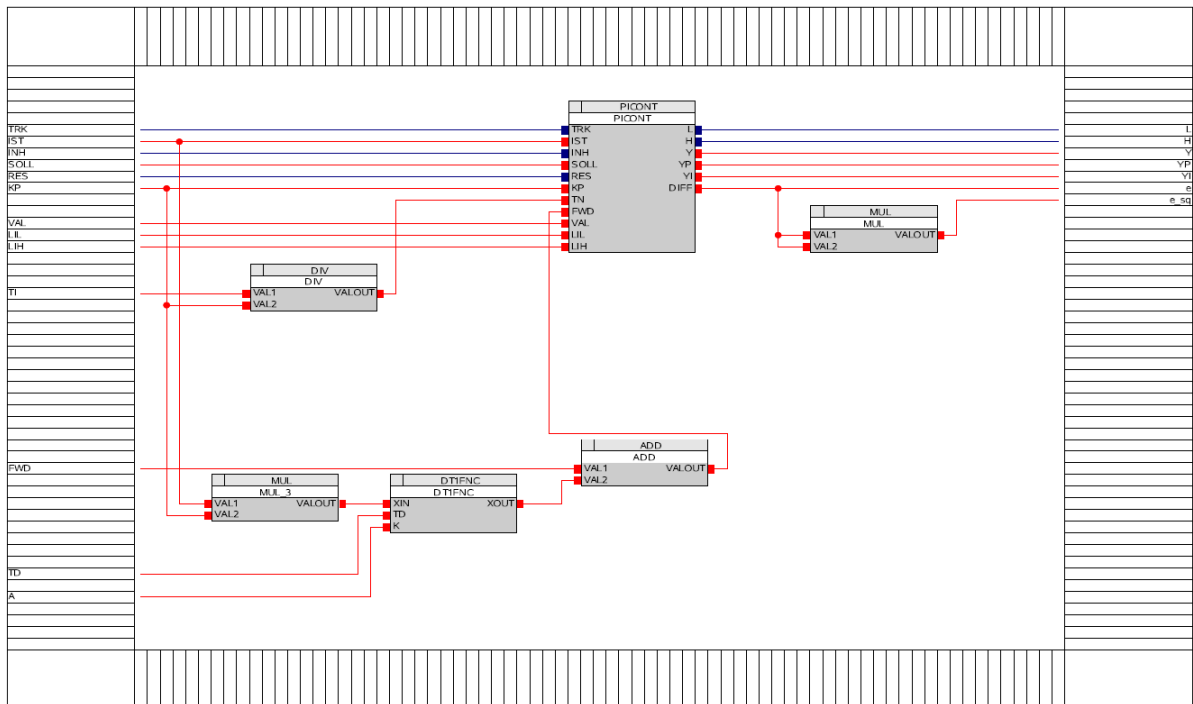


Abbildung C.2: PID-Regler realisiert in APROL

Der Baustein *ADD* bildet die Summe aus den beiden Eingängen und liefert das Ergebnis an den Ausgang. Der Baustein *MUL* multipliziert die beiden Eingänge, *DIV* führt eine Division durch. Eine Beschreibung des Codes ist hier nicht erforderlich. Da im *PICONT* für die Nachstellzeit des I-Anteiles $\frac{1}{T_N}$ gilt, wird der Parameter T_I außerhalb des Reglercodes mit K_P entsprechend $\frac{T_I}{K_P}$ modifiziert.

Die Ein-/ Ausgänge des PID-Reglers entsprechen den Funktionen wie in C.1.1 beschrieben. Der implementierte 'c-Code' für den PI-Regler ist im folgenden ersichtlich:
Block *PICONT*:

code: *PICONT*

*/**** PI Realisierung *****/*

```

TN_I = TN * 1000;
3
4
5
***** Initialisierungs Routine *****
6
if (!init)
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

```

```

    L = 1;
    /**** if ((YP + FWD) < LIL) ****/
    /**** { ****/
    /*** YI = 0; ****/
    /*** } ****/
    /**** else ****/
    /**** { ****/
    YI = LIL - (YP + FWD);
    /**** } ****/
    Y = LIL;
}
else
{
    L = 0;
}

if (Y > LIH)
{
    H = 1;
    L = 0;
    /**** if((YP+FWD) > LIH) ****/
    /**** { ****/
    /**** YI = 0; ****/
    /**** } ****/
    /**** else ****/
    /**** { ****/
    YI = LIH - (YP + FWD);
    /**** } ****/
    Y = LIH;
}
else
{
    H = 0;
}
}

/***** Regler im Handbetrieb *****/
/***** Regler im Handbetrieb *****/
else
{
    YI = (VAL - (DIFF * KP) - FWD);
    YP = 0;
    Y = VAL;

    /***** Stellgrößen Begrenzung *****/
    if (Y > LIH)
    {
        H = 1;
        L = 0;
    }
    else if (Y < LIL)
    {
        H = 0;
        L = 1;
    }
    else
    {
        H = 0;
        L = 0;
    }
}

/***** Regler im Handbetrieb *****/;

```

Um eine PID-Struktur zu erhalten wurde der PI-Regler mit einem DT1-Glied entsprechend Abbildung C.2 erweitert. Die Realisierung des DT1-Gliedes ist im folgenden er-

sichtlich:

Block DT1FNC:

code: DT1FNC

```
1  /***** DT1FNC: *****/
2
3  Zykluszeit = APROL_ATC_Library_TimeDelta;
4
5
6  if ((TD * 1000) < Zykluszeit)
7  {
8      TDI = (Zykluszeit / 1000);
9  }
10 else
11 {
12     TDI = TD;
13 }
14
15 if ((TD <= 0) | (K <= 0))
16 {
17     YT = 0 ;
18     XOUT = 0;
19 }
20 else
21 {
22     YT = YT + (Zykluszeit / (1000 * TDI) ) * (XIN - YT);
23     XOUT = K * (XIN - YT);
24 };
```


Abbildungsverzeichnis

1.1	Übersicht eines Laufkraftwerks mit einer Kaplan­turbine (aus [1])	1
1.2	Übersicht eines Speicherkraftwerks mit einer Pelton­­turbine (aus [1])	2
1.3	Turbinenreglerkonzept Andritz AG	3
1.4	Hardwarekonzept	5
2.1	Pelton Modell der Andritz AG realisiert in Simulink	8
2.2	Pelton Modell der Andritz AG	9
2.3	Modell für Ablenker und Düsenistwerte	10
2.4	Modell für die Durchflussberechnung	13
2.5	q (Düse) in Abhängigkeit der Düsenposition	13
2.6	Zusammenhang $f_m(e_s)$	14
2.7	Drehzahleinflussfaktor $m_z(z_w)$	16
2.8	Modell für die Momentenberechnung	17
2.9	m_{1kw1} - Momentenbeiwert Düse 1 (gleiches für Düse 2)	17
2.10	m_1 - Zusammenführung der beiden Momentenbeiwerte + Gewichtung	17
2.11	theoretischer Momenteneinfluss (ohne Drehzahleinfluss, Ablenkereinfluss und Einfluss des hydraulischen Bremsmomentes)	18
2.12	Modell für die Ermittlung von Turbinen- bzw. Generatoreigenschaften	20
2.13	Modell des Stauraums	22
2.14	Durchfluss in Abhängigkeit der Düsenöffnung	23
2.15	Vergleich Ablenker Steuerzeiten des Modells bzw. der Anlage	24
2.16	Vergleich Startverhalten des Modells bzw. der Anlage	25
3.1	Turbinenreglerkonzept Andritz AG	27
3.2	PID-Regler im Standardregelkreis	28
3.3	Regelkreis mit klassischem PID-Regler	29
3.4	Sprungantworten der einzelnen Regler	30
3.5	Sprungantwort der Strecke	31
3.6	Sprungantwort der Strecke mit P-Regler (closed loop method)	33
3.7	Sprungantwort der Strecke - T-Summe	34
3.8	Vergleich der Einstellregeln	36
3.9	Berechnungszeitraum für J	37
3.10	PID-Struktur für ES	38
3.11	Vergleich PID-Struktur für ES mit PID klassisch	39
3.12	ES Optimierungsschema	39

3.13	Zeitdiskretes ES Optimierungsschema	40
3.14	Optimierung mittels Extremum Seeking auf die Strecke G1	45
4.1	PID-Regler realisiert in Simulink	47
4.2	ES Simulink Block	48
4.3	ES Simulink Block Detailansicht	50
4.4	Optimierung des Drehzahlreglers im Simulink	51
4.5	Verlauf der Kostenfunktion bei Drehzahlregleroptimierung	52
4.6	Optimierung des Leistungsreglers im Simulink	53
4.7	Verlauf der Kostenfunktion bei Leistungsregleroptimierung	53
4.8	APROL Hypermakro (beinhaltet die grafische Darstellung inkl. der logischen Auswertung)	54
4.9	ES Algorithmus realisiert in APROL	55
4.10	PID Regler realisiert in APROL	56
4.11	ES Hypermakro logisch in APROL eingebunden	57
4.12	Drehzahlregler realisiert in APROL (Teil 1)	57
4.13	Drehzahlregler realisiert in APROL (Teil 2)	58
4.14	ES APROL Bedienung und Visualisierung	58
4.15	ES APROL Bedienung	59
4.16	ES APROL Parametrierung der Sollwertsprünge (HAND oder AUTO)	59
4.17	ES APROL Messzeit und Anzahl der Versuche	59
4.18	ES APROL Anzeigen der Werte der Zielfunktionen	59
4.19	ES APROL Optimierungsparameter	59
4.20	ES APROL Optimierungsparameter	60
4.21	APROL Trendviewer	61
4.22	APROL Trendviewer (Möglichkeit der Dokumentation)	62
4.23	Optimierung des Drehzahlreglers im APROL	63
4.24	Verlauf der Zielfunktion bei Drehzahlregleroptimierung in APROL (pos. Sollwertsprung)	64
4.25	Optimierung des Drehzahlreglers im APROL	65
4.26	Verlauf der Zielfunktion bei Drehzahlregleroptimierung in APROL (neg. Sollwertsprung)	65
A.1	Modell für Ablenker und Düse realisiert in APROL (L2901_SM)	68
A.2	Modell für Ablenker und Düse realisiert in APROL (L2901_SM)	69
A.3	Normierung Ablenkerposition realisiert in APROL (L2902_NORM)	69
A.4	Berechnung des Durchflusses realisiert in APROL (L2903_q_CALC)	70
A.5	Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)	72
A.6	Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)	72
A.7	Berechnung des Drehmoments (m) realisiert in APROL (L2904_m_CALC)	73
A.8	Modell für Turbine und Gen. realisiert in APROL (L2905_MOD_TU_GEN)	74
A.9	Modell für Pegel und Fallhöhe realisiert in APROL (L2906_LEVEL)	75
B.1	Simulinkmodell zur Berechnung von ES	77
C.1	PID-Regler realisiert in Simulink	81

C.2 PID-Regler realisiert in APROL 82

Literaturverzeichnis

- [1] Strom aus Wasserkraft - VERBUND-Austrian Hydro Power AG. In: *VERBUND-Austrian Hydro Power AG* (2007)
- [2] *B&R - Perfection in Automation: Prozessleittechnik APROL*. <http://www.aprol.com>. Version: 12 2009
- [3] *The MathWorks: MATLAB - The Language Of Technical Computing*. <http://www.mathworks.com/products/matlab/>. Version: 12 2009
- [4] *The MathWorks: Simulink - Simulation and Model-Based Design*. <http://www.mathworks.com/products/simulink/>. Version: 12 2009
- [5] ASTRÖM, Karl J. ; HÄGGLUND, Tore: *Automatic Tuning of PID controllers*. Instrument Society of America, 1988
- [6] ASTRÖM, Karl J. ; HÄGGLUND, Tore: *PID controllers: theory, design, and tuning (2nd ed.)*. Instrument Society of America, 1995
- [7] DÖRFLER, P.: *Simulationsmodelle für Wasserturbinen / Technischer Bericht - Sulzer Hydro*. 1996. – Forschungsbericht
- [8] HJALMARSSON, H. ; GEVERS, M. ; GUNNARSSON, S. ; LEQUIN, O.: *Iterative feedback tuning: theory and applications*. In: *IEEE Control System Magazine* 18 (1998), Aug., Nr. 4, S. 26–41. <http://dx.doi.org/10.1109/37.710876>. – DOI 10.1109/37.710876
- [9] HORN, Martin ; DOURDOUMAS, Nicolaos: *Regelungstechnik - Rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise*. Pearson Studium, 2004
- [10] KILLINGSWORTH, N. J. ; KRSTIC, M.: *PID tuning using extremum seeking: online, model-free performance optimization*. In: *IEEE Control System Magazine* 26 (2006), Feb., Nr. 1, S. 70–79. <http://dx.doi.org/10.1109/MCS.2006.1580155>. – DOI 10.1109/MCS.2006.1580155
- [11] KILLINGSWORTH, Nicholas J.: *HCCI Engine Control and Optimization*, UNIVERSITY OF CALIFORNIA, SAN DIEGO, Diplomarbeit, 2007