Fachhochschule Nordwestschweiz
Hochschule für Technik

Reglerdimensionierung mit Hilfe der Schrittantwort

RDT One

Fachbericht

Projekt 2
Team 1

Windisch, 10.06.2015

Auftraggeber: Peter Niklaus

Betreuer: Pascal Buchschacher, Anita Gertiser

Experten: Peter Niklaus, Richard Gut

Team: Alexander Stocker
Claudius Jörg
Denis Stampfli
Martin Moser
Reto Freivogel
Yohannes Measho

Studiengang: Elektro- und Informationstechnik

Abstract

Regler werden heutzutage oft mit simplen Faustformeln dimensioniert. Für stabilere Regelungen kann die aufwändigere Phasengangmethode angewendet werden. Das Ziel dieser Arbeit ist es mit einem Programm die Phasengangmethode zu automatisieren. Diese Automatisierung wurde im Programm implementiert. Zusätzlich kann das Programm die Resultate mit verschiedenen Faustformeln vergleichen. Der Fachbericht setzt sich zusammen aus einem mathematischen und einem programmtechnischen Teil. Der mathematische Abschnitt behandelt die Streckenanalyse, die Phasengangmethode und ihr Ablauf, die Übertragungsfunktion der Regler, die Schrittantwort der Regelung und die Optimierungsmöglichkeiten. Er geht auch auf die Dimensionierung mit Faustformeln ein. Der programmtechnische Bericht erläutert die Software, die Benutzerschnittstelle und die Anwendung von Model-View-Controller.

Projekt P2 - Aufgabenstellung vom Auftraggeber (FS_2015)

Reglerdimensionierung mit Hilfe der Schrittantwort

1. Einleitung

In der Praxis werden die klassischen Regler (PI, PID, PD, …) oft mit sog. Faustformeln dimensioniert. Dazu benötigt man bestimmte Informationen der zu regelnden Strecke. Handelt es sich dabei um „langsame Strecken“ mit Zeitkonstanten im Bereich von Sekunden bis Minuten, so ist das Bestimmen und Ausmessen der Schrittantwort oft die einzige Möglichkeit zur Identifikation der Strecke. Typische Beispiele dafür sind Temperaturheizstrecken, welche meistens mit einem PTn-Verhalten modelliert werden können (Kaffeemaschine, Boiler, Raumheizungen, Lötkolben, Warmluftfön, usw.).

Die Schrittanwort wird mit Hilfe einer Wendetangente vermessen und die Kenngrössen Streckenbeiwert (K_s), Verzugszeit (T_u) und Anstiegszeit (T_g) werden bestimmt. Dies kann sowohl von Hand (grafisch) oder auch automatisiert durchgeführt werden, falls die Messdaten elektronisch vorliegen. Mit diesen drei Kenngrössen können mit Hilfe sog. Faustformeln PI- und PID-Regler dimensioniert werden (Ziegler/Nichols, Chien/Hrones/Reswick, Oppelt, Rosenberg). Die Faustformeln liefern zwar sehr schnell die Reglerdaten, aber die Schrittantworten der entspr. Regelungen sind teilweise weit vom "Optimum" entfernt und der Regelkreis kann sogar instabil werden. In der Praxis muss man diese "Startwerte" häufig nachoptimieren, damit die Schrittantwort der Regelung die Anforderungen erfüllt.

Die sog. "Phasengangmethode zur Reglerdimensionierung" wurde von Jakob Zellweger (FHNW) entwickelt und liefert Reglerdaten, welche näher am "Optimum" sind und für die Praxis direkt verwendet werden können. Dabei kann das Überschwingen der Schrittantwort vorgegeben werden (z.B. 20%, 10%, 2%, oder aperiodisch). Bei dieser Methode kann also das für viele Anwendungen wichtige Verhalten der Schrittantwort beeinflusst werden. Um die Phasengangmethode anwenden zu können, muss der Frequenzgang der Strecke bekannt sein (analytisch oder numerisch/gemessen). Mit Hilfe der Hudzovik-Approximation (oder anderer ähnlicher Verfahren) wird dieses Problem gelöst, in dem vorgängig aus den Kenngrössen der Schrittantwort (K_s, T_u,T_g) eine PTn-Approximation der Strecke erzeugt wird. Mit dem Frequenzgang der PTn-Approximation können dann die Regler dimensioniert werden (I, PI, PID). Die Phasengangmethode war ursprünglich eine grafische Methode, basierend auf dem Bodediagramm der Strecke. Aktuell soll die Methode direkt numerisch im Rechner durchgeführt werden.

In dieser Arbeit geht es um die Entwicklung und Realisierung eines Tools zur **Reglerdimensionierung mit der Phasengangmethode**. Ausgehend von der PTn-Schrittantwort der Strecke sollen "optimale Regler" (PI, PID-T1) dimensioniert werden, wobei das Überschwingen der Regelgrösse vorgegeben werden kann. Zum Vergleich sollen die Regler auch mit den üblichen Faustformeln dimensioniert werden. Wünschenswert wäre auch eine Simulation der Schrittantwort des geschlossenen Regelkreises, so dass die Dimensionierung kontrolliert und evtl. noch "verbessert" werden könnte.

2. Aufgaben/Anforderungen an Tool

Entwerfen und realisieren Sie ein benutzerfreundliches Tool/Programm/GUI/usw. mit welchem PI- und PID-Regler mit der Phasengangmethode dimensioniert werden können. Dabei sind folgende Anforderungen und Randbedingungen vorgegeben:

- Die zu regelnden Strecken sind PTn-Strecken, wobei entweder die Schrittantwort grafisch vorliegt oder die Kenngrössen K_s, T_u und T_g schon bekannt sind
- Die Bestimmung einer PTn-Approximation wird vom Auftraggeber zur Verfügung gestellt und muss entsprechend angepasst und eingebunden werden (Matlab zu Java)
- Das Überschwingen der Regelgrösse (Schrittantwort) soll gewählt werden können
- Zum Vergleich sind die Regler auch mit den üblichen Faustformeln zu dimensionieren.
- Das dynamische Verhalten des geschlossenen Regelkreises soll auch berechnet und visualisiert werden (Schrittantwort)

3. Bemerkungen

Die Software und das GUI sind in enger Absprache mit dem Auftraggeber zu entwickeln. Der Auftraggeber steht als Testbenutzer zu Verfügung und soll bei der Evaluation des GUI eingebunden werden. Alle verwendeten Formeln, Algorithmen und Berechnungen sind zu verifizieren, eine vorgängige oder parallele Programmierung in Matlab ist zu empfehlen. Zum Thema der Regelungstechnik und speziell zur Reglerdimensionierung mit der Phasengangmethode werden Fachinputs durchgeführt (Fachcoach).

Literatur

[1] J. Zellweger, *Regelkreise und Regelungen*, Vorlesungsskript.

[2] J. Zellweger, *Phasengang-Methode*, Kapitel aus Vorlesungsskript.

[3] H. Unbehauen, *Regelungstechnik I*, Vieweg Teubner, 2008.

[4] W. Schumacher, W. Leonhard, *Grundlagen der Regelungstechnik*, Vorlesungsskript, TU Braunschweig, 2003.

[5] B. Bate, *PID-Einstellregeln*, Projektbericht, FH Dortmund, 2009.

16.02.2015
Peter Niklaus

Inhaltsverzeichnis

1 Einleitung **6**

2 Theoretische Grundlagen **7**

2.1 Streckenanalyse 8

2.2 Phasengangmethode 9

2.2.1 Grundlagen 9

2.2.2 Ablauf der Phasengangmethode 12

2.3 Übertragungsfunktion der Regler 12

2.4 Dimensionierung mit Faustformeln 13

2.5 Schrittantwort der Regelung 14

2.5.1 Berechnung 14

2.5.2 Optimierungsmöglichkeiten 15

3 Software **16**

3.1 Packages und ihre Klassen 17

3.1.1 Klassen des Packages *controller* 17

3.1.2 Ausgewählte Klassen des Packages *model* 19

3.1.3 Ausgewählte Klassen des Packages *dimensionierung* 21

3.1.4 Klassen des Packages *view* 23

3.2 Benutzerinteraktion 25

4 Schlussfolgerung **26**

5 Ehrlichkeitserklärung **27**

6 Literatur **28**

7 Anhang **29**

1 Einleitung

Klassische Regler werden heutzutage oft über die gängigen Faustformeln dimensioniert. Leider erhält man durch die Anwendung von Faustformeln oft unbrauchbare Reglerparameter. Aus diesem Grund müssen Regler oft nachoptimiert werden. Eine andere Dimensionierungsmöglichkeit ist die Phasengangmethode von Jakob Zellweger. Diese Methode ist zwar aufwändiger, liefert dafür aber stabilere Regler. Wie der Name der Methode bereits verrät, werden die Reglerwerte über den Phasengang der Strecke ermittelt. Es handelt sich ursprünglich um eine Dimensionierung, die von Hand auf logarithmischen Papier erarbeitet wird, was aufwändig ist. Die Aufgabe dieses Projektes ist es, diese Methode mit Hilfe einer Software zu automatisieren.

Ziel ist es ein GUI (Graphical User Interface) zu erstellen, bei welchem die Streckenparameter eingegeben werden können. Die Software soll die Eingaben mit der Phasengangmethode weiterverarbeiten. Daraus resultieren die Reglerparameter. Ein weiteres Ziel ist die Schrittantwort der gesamten Regelung zu berechnen und grafisch darzustellen. Die Bedienung des GUI soll übersichtlich und einfach in der Handhabung sein.

Als Ausgangslage ist die Schrittantwort der Strecke gegeben, womit die Streckenparameter herausgelesen werden können. Diese Werte sind in die Software einzugeben. Resultierend daraus erhält der Anwender die Reglerparameter sowie die Schrittantwort davon. Dieser Prozess kann in sechs Bereiche unterteilt werden. Beginnend mit der Streckenidentifikation. Die Übertragungsfunktion der Strecke wird über die Sani-Approximation ermittelt. Als nächstes kann der Frequenzgang berechnet und mit der Phasengangmethode die Reglerwerte ermittelt werden. Somit sind die Reglerparameter bereits bestimmt und die Übertragungsfunktion des Reglers ist gegeben. Doch die Schrittantwort der Regelung fehlt zu diesem Zeitpunkt noch. Mit der Faltung werden die beiden Übertragungsfunktionen von Regler und Strecke zur Übertragungsfunktion der geschlossenen Regelung zusammengefügt. Für die Berechnung der Schrittantwort werden zuerst die Residuen der Übertragungsfunktion ermittelt. Die Residuen werden benötigt, um mittels der inversen Laplace Transformation die Impulsantwort zu bilden. Abschliessend muss die Impulsantwort integriert werden, sodass die Schrittantwort der Regelung entsteht.

Die Software ist im bekannten MVC-Pattern gehalten. Die View implementiert die graphische Oberfläche der Software. Sie fungiert als Benutzerschnittstelle und ermöglicht die Eingabe der Streckenparameter. Der Controller steuert die Berechnungen der Applikation. Er leitet die Aufgaben dem Model weiter. Im Model werden die Berechnungen der entsprechenden Regler getätigt. Unter Verwendung der Klasse Polynome, Transferfunction und Schrittantwort werden die Regler berechnet. Die Daten werden über die Klasse Observer dem View zurückgegeben und dort aktualisiert.

Der nachfolgende Bericht ist in zwei Abschnitte unterteilt. Der erste Teil beinhaltet die Grundlage der Phasengangmethode sowie deren wichtigste Formeln. Im zweiten Teil wird auf die Programmierung eingegangen, die auf dem MVC-Pattern beruht. Am Ende wird die Funktionalität der wichtigsten Methoden geklärt. Ein weiterer wichtiger Bestandteil ist das MVC-Pattern.

2 Theoretische Grundlagen

Eine Regelung (auch Regelkreis genannt) besteht, wie in Abbildung 1 zu sehen ist, aus einem Regler und einer Regelstrecke. Beispiele einer Regelung sind Raumheizungen, Lötkolben oder Geschwindigkeitsregelungen.

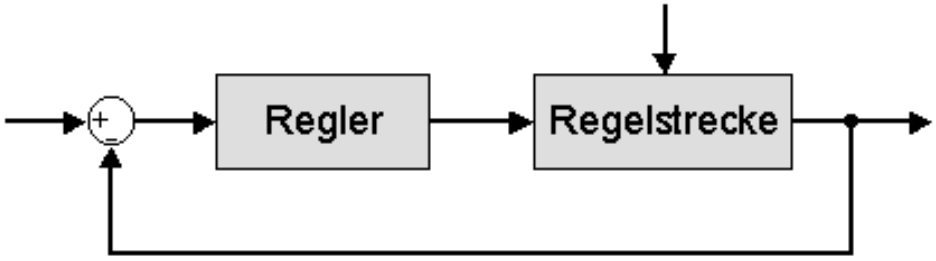


Abbildung 1: Regelung bestehend aus Regler und Regelstrecke [1]

Ist eine Regelstrecke gegeben, im Falle eines Lötkolbens wäre dies zum Beispiel die Distanz vom Heizelement bis zur Lötspitze, so muss ein dazu kompatibler Regler dimensioniert werden. Reglerdimensionierungen können über das Auswerten der Schrittantwort der Regelstrecke durchgeführt werden. Um die Schrittantwort zu erhalten wird das Verhalten der Ausgangsgrösse der Strecke aufgrund eines Schrittes der Eingangsgrösse gemessen. Für die Berechnungen werden die aus der Schrittantwort der Strecke ausgelesenen Kenngrössen Verzugszeit (T_u), Anstiegszeit (T_g) und Streckenbeiwert (K_s) verwendet. Zur Dimensionierung eines kompatiblen Reglers existieren verschiedenste Methoden. Im Folgenden werden nun zwei Möglichkeiten genauer erläutert, wobei das Hauptaugenmerk auf der ersten Methode, der Phasengangmethode, liegt und die zweite, die Dimensionierung mittels gängigen Faustformeln, lediglich zum Vergleich durchgeführt wird.

Die Kurve in Abbildung 2 stellt die Schrittantwort einer Regelstrecke dar. Daran angelegt ist die Wendetangente, die benötigt wird, um die Verzugs- und die Anstiegszeit messen zu können.

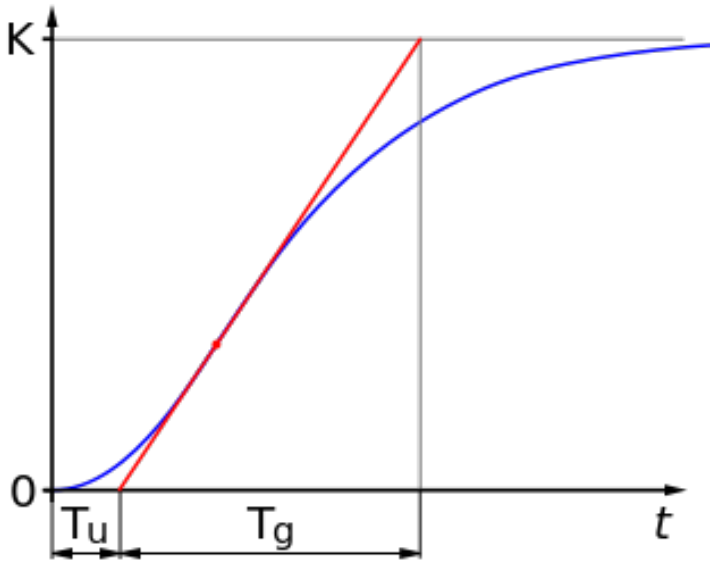


Abbildung 2: Schrittantwort einer Regelstrecke [2]

Ziel ist es, PI- sowie PID-Regler zu dimensionieren, das heisst deren Kennwerte zu berechnen. Der PI-Regler hat die Kennwerte Nachstellzeit (T_n) und Reglerverstärkung (K_R), beim PID-Regler kommt noch die Vorhaltezeit (T_v) und die parasitäre Zeitkonstante (T_p) hinzu.

2.1 Streckenanalyse

Aus den gegebenen Kennwerten der Schrittantwort soll nun die Übertragungsfunktion der Strecke berechnet werden, das heisst die Strecke muss identifiziert werden. Die Streckenidentifikation, wird mittels der Sani-Methode durchgeführt. Aus dem Verhältnis von T_u und T_g werden die Ordnung der Strecke (n) und die dazugehörigen Zeitkonstanten ($T_1, T_2, ..., T_n$) berechnet, die benötigt werden, um die Übertragungsfunktion der Regelstrecke darzustellen. Die Übertragungsfunktion der Strecke ist gemäss Formel 1 definiert: [2]

$$G_S(s) = \frac{K_S}{(1 + sT_1)(1 + sT_2)...(1 + sT_n)} \tag{1}$$

2.2 Phasengangmethode

Bei der Phasengangmethode handelt es sich eigentlich um eine grafische Dimensionierungsmethode, welche früher noch von Hand mit logarithmischem Papier durchgeführt wurde. Anstelle der grafischen Dimensionierung wird die Methode in diesem Fall komplett rechnerisch gelöst. Um die Methode zu verstehen wird ein theoretisches Grundwissen benötigt, welches in folgenden Unterkapiteln erklärt wird.

2.2.1 Grundlagen

Durch Ersetzen von s durch $j\omega$ in der Übertragungsfunktion der Strecke erhält man den Frequenzgang. Damit die Phasengangmethode angewendet werden kann, wird der Amplituden- sowie der Phasengang der Strecke benötigt (Formeln 2, 3 und 4).

$$A_S(\omega) = abs(G_S(j\omega)) \tag{2}$$

$$\varphi_S(\omega) = arg(G_S(j\omega)) \tag{3}$$

$$\varphi_S(\omega) = -arctan(\omega T_1) - arctan(\omega T_2)... - arctan(\omega T_n) \tag{4}$$

Die Zeitkonstanten des zu dimensionierenden Reglers stehen in direktem Zusammenhang mit dessen Amplitudengang. In den Abbildungen 3 und 4 sind die Amplitudengänge eines PI- und eines PID-Reglers dargestellt, wobei die x-Achsen logarithmisch skaliert sind. Wie die Kreisfrequenzen an den Knickstellen mit den Reglerparametern zusammenhängen sollen, ist in Tabelle 1 ersichtlich.

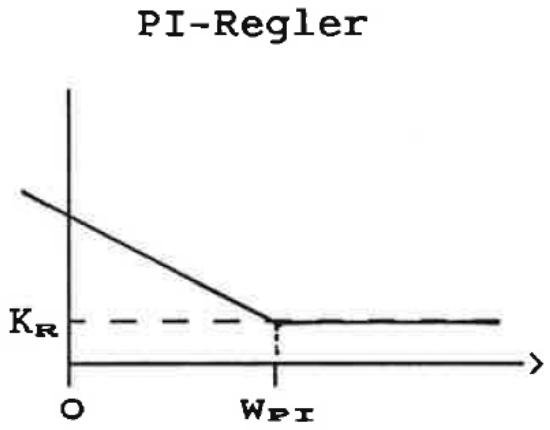


Abbildung 3: Amplitudengang eines PI-Reglers in Abhängigkeit der Kreisfrequenz ω (log. Darstellung) [3]

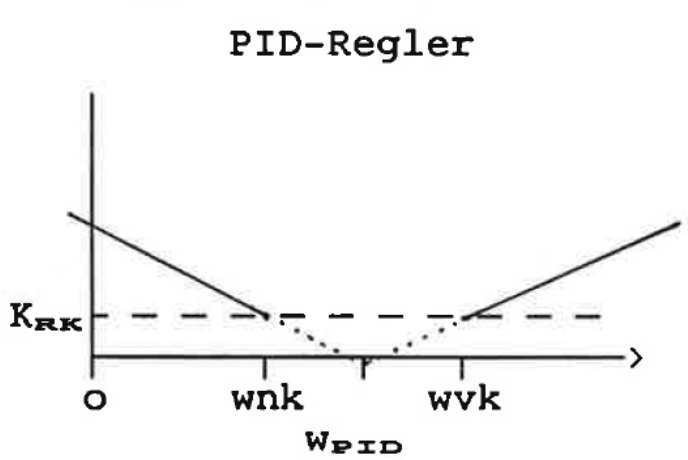


Abbildung 4: Amplitudengang eines PID-Reglers in Abhängigkeit der Kreisfrequenz ω (log. Darstellung) [3]

PI-Regler	**PID-Regler**	
$\omega_{PI} = \frac{1}{T_n}$	$\omega_{nk} = \frac{1}{T_{nk}} = \beta * \omega_{PID}$	$\omega_{vk} = \frac{1}{T_{vk}} = \frac{\omega_{PID}}{\beta}$

Tabelle 1: Amplitudengänge Regler und Zusammenhänge mit Knickkreisfrequenzen

Diese Knickkreisfrequenzen (ω_{PI} in Abb. 3 und ω_{PID} ind Abb. 4) werden mithilfe des Phasengangs der Strecke berechnet. Je nachdem welcher Reglertyp dimensioniert werden soll, müssen andere Punkte im Phasengang gesucht werden (Tabelle 2).

Regler	**Phasengang der Strecke φ_s**	**Knickkreisfrequenz**
PI	$-\frac{\pi}{2} = -90°$	$\omega_{PI} = \frac{1}{T_n}$
PID	$-\frac{3\pi}{4} = -135°$	$\omega_{PID} = \frac{1}{\sqrt{T_{nk} T_{vk}}}$

Tabelle 2: Gesuchte Punkte im Phasengang je nach Regler

Der PI-Regler kann anhand dieser Dimensionierungskriterien bereits dimensioniert werden. Beim PID-Regler muss allerdings noch ein Parameter mehr bestimmt werden. Es handelt sich um den Faktor β welcher benötigt wird um die Knickkreisfrequenzen des PID-Reglers berechnen zu können. Dieser Faktor β hängt mit der Tangentensteigung des Phasengangs der offenen Regelung im Punkt $-135°$ zusammen.
Ein Dimensionierungskriterium der Phasengangmethode lautet, dass die Steigung des Phasengangs der gesamten, offenen Regelung bei der Knickkreisfrequenz $\frac{-0.5}{\omega_{PID}}$ betragen soll. Dies hat damit zu tun, dass eine Steigung von -20dB/Dekade angestrebt wird. Die genaue Herleitung von β befindet sich im Anhang. Es gilt die folgende Beziehung:

$$\frac{2\beta}{1+\beta^2} + \omega_{PID}\frac{d\varphi_{Strecke(\omega_{PID})}}{d\omega} = -0.5 \tag{5}$$

Da der Phasengang der Strecke und somit dessen Ableitung gegeben ist, kann durch Auflösen nach β nun auch dieser Parameter ermittelt werden. Ergibt sich ein Wert der grösser als eins oder komplexwertig ist, so wird β als eins angenommen.

Die Verstärkung des Reglers wird mit Hilfe des Phasenrands bestimmt. Der Phasenrand ist die Differenz der Streckenphase zu $-180°$. Bei unterschiedlichem Phasenrand ergeben sich unterschiedliche Verstärkungsfaktoren, was auch auf das Überschwingen des Reglers Einfluss hat. Tabelle 3 zeigt den Zusammenhang zwischen dem gewählten Phasenrand und dem daraus resultierenden Überschwingen der Regelung.

Phasenrand φ_R	**Streckenphase** φ_S	**Überschwingen**
$45°$	$-135°$	23%
$51.8°$	$-128.5°$	16.3%
$65.5°$	$-114.6°$	4.6%
$76.3°$	$-103.7°$	0%

Tabelle 3: Zusammenhang Phasenrand und Überschwingen

Zur Berechnung der Verstärkung werden nun die Amplitudengänge der Strecke, sowie des Reglers benötigt. Dazu wird im Phasengang der offenen Regelung φ_S der Phasenrand abgetragen. Dies liefert die Kreisfrequenz ω_D. Die beiden Amplituden bei der Kreisfrequenz ω_D, werden miteinander multipliziert. Gemäss Phasengangmethode soll die Verstärkung an dieser Stelle eins sein. Dies führt zur Formel 6. $Go(\omega_D)$ entspricht dem Amplitudengang der offenen Regelung an der Stelle ω_D.

$$K_R = \frac{1}{|Go(\omega_D)|} \tag{6}$$

2.2.2 Ablauf der Phasengangmethode

Die Phasengangmethode der Strecke φ_S wird nach dem folgenden Ablauf durchgeführt. Die Punkte 2. und 3. werden nur für PID-Regler benötigt. Eine numerische Beispielrechnung für einen PI-Regler ist im Anhang angefügt.

1. Im Phasengang der Strecke φ_S werden die Kreisfrequenzen in bestimmten Punkten gesucht.
2. Der Faktor β wird bestimmt.
3. Mithilfe von β werden die Zeiten T_{nk} und T_{vk} ermittelt.
4. Der Phasengang der offenen Regelung φ_{go} wird berechnet (Phasengang Strecke + Phasengang Regler)
5. Die Kreisfrequenz ω_D beim gewählten Phasenrand wird berechnet.
6. Die Amplitudengänge der Strecke sowie des Reglers bei ω_D werden miteinander multipliziert. Damit wird die Reglerverstärkung K_R festgelegt.

Besondere Vorsicht ist bei den erhaltenen Kenngrössen des dimensionierten Reglers geboten. Es wird zwischen Regler- und Bodekonform unterschieden. Bodekonform bedeutet, dass die Parameter kaskadiert sind und die logarithmische Rechnung somit vereinfacht wird. Standardmässig wird in der Fachliteratur meist reglerkonform gerechnet. Die Parameter des PI-Reglers sind reglerkonform und können somit direkt weiterverwendet werden. Die Parameter des PID-Reglers, welche wir bei der Dimensionierung mittels Phasengangmethode erhalten, sind bodekonform. Sie können jedoch in die reglerkonforme Darstellung umgerechnet werden. Für die Umrechnung wurden die Formeln 7, 8 und 9 für PID-Regler hergeleitet (siehe Anhang): [3]

$$K_R = K_{rk}\left(1 + \frac{T_{vk}}{T_{nk}} - \frac{T_p}{T_{nk}}\right) \tag{7}$$

$$T_n = T_{nk} + T_{vk} - T_p \tag{8}$$

$$T_v = \frac{T_{nk}T_{vk}}{T_{nk} + T_{vk} - T_p} - T_p \tag{9}$$

Tp steht für die parasitäre Zeitkonstante. Diese wird benötigt, da ein idealer PID-Regler in der Praxis nicht umsetzbar ist. Der Index k in der bodekonformen Darstellung steht für Kaskadierung.

2.3 Übertragungsfunktion der Regler

Um schlussendlich die Schrittantwort der geschlossenen Regelung berechnen zu können, wird die Übertragungsfunktion des dimensionierten Reglers benötigt. Diese lässt sich mit den aus der Phasengangmethode erhaltenen Parametern aufstellen (Tabelle 4).

Beschreibung	**Darstellung**
PI-Regler bodekonform	$G_R(s) = K_R \frac{(1+sT_n)}{sT_n}$
PI-Regler reglerkonform	$G_R(s) = K_R \left(1 + \frac{1}{sT_n}\right)$
PID-Regler bodekonform	$G_R(s) = K_{rk} \frac{(1+sT_{nk})(1+sT_{vk})}{sT_{nk}(1+sT_p)}$
PID-Regler reglerkonform	$G_R(s) = K_R \left(1 + \frac{1}{sT_n} + \frac{sT_v}{1+sT_p}\right)$

Tabelle 4: Übertragungsfunktionen Regler [4]

2.4 Dimensionierung mit Faustformeln

Die Dimensionierung mittels Faustformeln wird durchgeführt, um die Dimensionierungsresultate der Phasengangmethode am Schluss zu vergleichen und auszuwerten. Es gibt etliche verschiedene Faustformeln von unterschiedlichen Personen. Die bekanntesten und gängigsten Dimensionierungsformeln sind in den folgenden Tabellen aufgelistet und werden im Programm implementiert.

Formeltyp	**Regler**	K_R	T_n	T_v
Ziegler/Nichols	**P**	$\frac{T_g}{K_s T_u}$		
	PI	$\frac{0.9*T_g}{K_s T_u}$	$3.3 * T_u$	
	PID	$\frac{0.9*T_g}{K_s T_u}$	$2 * T_u$	$0.42 * T_t$
Oppelt	**P**	$\frac{T_g}{K_s T_u}$		
	PI	$\frac{0.8*T_g}{K_s T_u}$	$3 * T_u$	
	PID	$\frac{1.2*T_g}{K_s T_u}$	$2 * T_u$	$0.42 * T_t$
Rosenberg	**P**	$\frac{T_g}{K_s T_u}$		
	PI	$\frac{0.91*T_g}{K_s T_u}$	$3.3 * T_u$	
	PID	$\frac{1.2*T_g}{K_s T_u}$	$2 * T_u$	$0.44 * T_t$

Tabelle 5: Fausformeln Ziegler/Nichols, Oppelt und Rosenberg [5]

Reglertyp	**aperiodischer Einschwingvorgang**		**Einschwingvorgang mit 20% Überschwingen**	
	Führung	Störung	Führung	Störung
P	$K_R = \frac{0.3*T_g}{K_s T_u}$	$K_R = \frac{0.3*T_g}{K_s T_u}$	$K_R = \frac{0.7*T_g}{K_s T_u}$	$K_R = \frac{0.7*T_g}{K_s T_u}$
PI	$K_R = \frac{0.35*T_g}{K_s T_u}$ $T_n = 1.2 * T_g$	$K_R = \frac{0.6*T_g}{K_s T_u}$ $T_n = 4 * T_u$	$K_R = \frac{0.6*T_g}{K_s T_u}$ $T_n = T_g$	$K_R = \frac{0.7*T_g}{K_s T_u}$ $T_n = 2.3 * T_u$
PID	$K_R = \frac{0.6*T_g}{K_s T_u}$ $T_n = T_g$ $T_v = 0.5 * T_u$	$K_R = \frac{0.95*T_g}{K_s T_u}$ $T_n = 2.4 * T_u$ $T_v = 0.42 * T_u$	$K_R = \frac{0.95*T_g}{K_s T_u}$ $T_n = 1.35 * T_g$ $T_v = 0.47 * T_u$	$K_R = \frac{1.2*T_g}{K_s T_u}$ $T_n = 2.3 * T_u$ $T_v = 0.42 * T_u$

Tabelle 6: Chien/Hrones und Reswick [5]

2.5 Schrittantwort der Regelung

2.5.1 Berechnung

Um die Schrittantwort der geschlossenen Regelung berechnen zu können, wird die Übertragungsfunktion der Regelung benötigt. Diese wird aus den Übertragungsfunktionen des Reglers und der Strecke gemässe Formel 10 berechnet.

$$G(s) = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)} \tag{10}$$

Die Berechnung der Schrittantwort wird mittels Residuenrechnung durchgeführt. Erstes Teilziel ist es die Übertragungsfunktion in die Form von Formel 11 zu bringen, wo $B(s)$ das Zählerpolynom der Übertragungsfunktion, $A(s)$ das Nennerpolynom, $R(n)$ die Residuen, $P(n)$ die Polstellen und K den Direktterm darstellen.

$$\frac{B(s)}{A(s)} = \frac{R(1)}{s - P(1)} + \frac{R(2)}{s - P(2)} + ... + K \tag{11}$$

Dafür werden als erstes die Polstellen der Übertragungsfunktion gesucht. Diese befinden sich bei den Nullstellen des Nennerpolynoms. Die Residuen erhält man durch die Partialbruchzerlegung mittels Residuenkalkül gemäss Formel 12, welche aber nur Gültigkeit hat, wenn ausschliesslich einfache Nullstellen vorkommen.

$$R(n) = \lim_{s \to P(n)} (s - P(n)) \frac{B(s)}{A(s)} \tag{12}$$

Durch die inverse Laplactransformation, unter Verwendung der Korrespondenztabellen erhält man die Impulsantwort $h(t)$ (Formel 13 mit $t \geq 0$).

$$h(t) = K\delta(t) + \sum_{i=1}^{n} R(i) * e^{P(i)t} \tag{13}$$

Aus der erhaltenen Implusantwort $h(t)$ muss nun noch die Schrittantwort berechnet werden, was mittels Integration gemacht wird (Formel 14).

$$s(t) = \int_0^t K\delta(\tau)d\tau + \sum_{i=1}^{n} \int_0^t R(i) * e^{P(i)\tau} d\tau \tag{14}$$

Die Schrittantwort wird zusätzlich noch analysiert. Folgende Werte werden ausgewertet:

- Anregelzeit T_{an}: Die Zeit, welche die Strecke benötigt bis der Sollwert das erste mal erreicht wird.
- Ausregelzeit T_{aus}: Die Zeit, welche die Strecke benötigt bis die Strecke in einem gewissen Toleranzbereich einmündet, und diesen nicht wieder verlässt.
- Max. Überschwingen Y_{max}: Der Wert an der Stelle, wo die Schrittantwort an meisten vom Sollwert abweicht.

2.5.2 Optimierungsmöglichkeiten

Falls die berechnete und dargestellte Schrittantwort der Regelung nicht der Vorstellung und den Ansprüchen des Nutzers entspricht, soll dieser die Möglichkeiten haben, die Schrittantwort nach seinen Anforderungen zu optimieren. Dazu gibt es verschiedene Möglichkeiten.

Einerseits kann der Regler durch seine Eigenschaften K_R, T_n, T_v und T_p modifiziert werden. Eine Schrittantwort durch die Veränderung dieser drei Eigenschaften zu optimieren ist äusserst schwierig. Dies erfordert grosses Wissen und viel Erfahrung in der Regelungstechnik, was einem weniger erfahrenen Nutzer die Optimierung nahezu verunmöglicht.

Es besteht jedoch die Möglichkeit die Optimierungen durch Verändern gewisser Parameter in der Phasengangmethode durchzuführen. So ist es aus der Theorie ersichtlich, dass der gewählte Phasenrand auf das Überschwingen Einfluss hat (siehe Kapitel 2.2.1, S. 11, Tabelle 3). Die Berechnung kann also mittels gewähltem Phasenrand solange optimiert werden, bis das Überschwingen der Schrittantwort den Anforderungen entspricht. Dies erlaubt eine iterative Berechnung, welche nach Eingabe des gewünschten Überschwingens in Prozent, solange den Phasenrand verändert bis die Schrittantwort den Anforderungen entspricht.

3 Software

Hinweise zur Benutzung
Der Fachbericht dient zur Erklärung des Aufbaus des Java Source Codes. Für eine Einführung in das Programm beachte man die Bedienungsanleitung im Anhang.
JavaKlassenNamen sind im Folgenden kursiv mit Grossbuchstaben geschrieben. Dagegen sind *packages* kursiv und klein geschrieben. Das gesamte Klassendiagramm befindet sich im Anhang. Fett gedruckte Nummern (**1**) beziehen sich auf Elemente des GUIs. Eine Abbildung des GUIs mit Nummerierung findet sich am Ende der Bedienungsanleitung im Anhang.

Ziele
Ziel der Applikation ist die automatische Dimensionierung eines Reglers mit der Phasengangmethode von Prof. J. Zellweger. Weiter soll die Schrittantwort des gesamten Regelkreises ermittelt und visualisiert werden. Die Regelstrecke ist durch die Parameter K_s, T_u und T_g gegeben. Neben der Phasengangmethode sind auch Faustregeln zur Reglerdimensionierung anwendbar. Die berechneten Werte sowie die markanten Werte der Schrittantwort sollen als Text ausgegeben werden. Die Software verfügt weiter über die nützlichen Befehle Rückgängig und Wiederholen. Die eingegebenen Werte können zudem gespeichert und geladen werden.

Aufbau und Packages
Die Applikation ist nach dem Model-View-Controller Architekturmuster aufgebaut. Wiedergespiegelt ist das in den drei Packages *model*, *view* und *controller*.
Das *model* soll eine Abstraktion der mathematischen Objekte bieten. Die meisten enthaltenen Klassen sind für die Vermeidung von Aliasing-Problemen (Zugriff zweier Klassen auf die gleiche Instanz) als unveränderlich gehalten.
Die Klassen zur Reglerdimensionierung sind für die bessere Kapselung im separaten Package *model.dimensionierung* enthalten. Alle Klassen dieses Packages sind als unveränderlich gehalten.
Das *view* erzeugt die graphische Benutzeroberfläche mit den Daten des *models*. Die Kommunikation zwischen *view* und *model* erfolgt mittels dem Observer Entwurfsmuster.
Der *controller* übernimmt die Steuerung der Applikation und die Ausführung der Befehle des Benutzers.

3.1 Packages und ihre Klassen

3.1.1 Klassen des Packages *controller*

Der Aufbau des Packages *controller* ist in Abbildung 5 ersichtlich.

Controller
Der *Controller* ist verantwortlich für die Steuerung der Applikation. Er besitzt für alle Befehle die der Benutzer im GUI tätigen kann eine Methode zur Modifizierung des *Models*. Meldet das *Model* beim Modifizieren einen Fehler informiert er den Benutzer mit der Methode displayError des *Views*. Vor jeder Änderung des *Models* speichert sich der *Controller* eine Kopie davon. Dadurch ist es möglich alte Zustände wiederherzustellen und die Befehle Rückgängig und Wiederholen zu unterstützen. Weiter ist der *Controller* für das Speichern und Laden zuständig. Dies funktioniert da das *Model* das *Serializable* Interface implementiert und der *Controller* so die Daten mit einem *ObjectOutputStream* in eine Datei schreiben bzw. mit einem *ObjectInputStream* aus einer Datei laden kann.

PIDRechner
Diese Klasse ist der Startpunkt (main) der Applikation. Sie initialisiert *Controller*, *View* und *Model*.

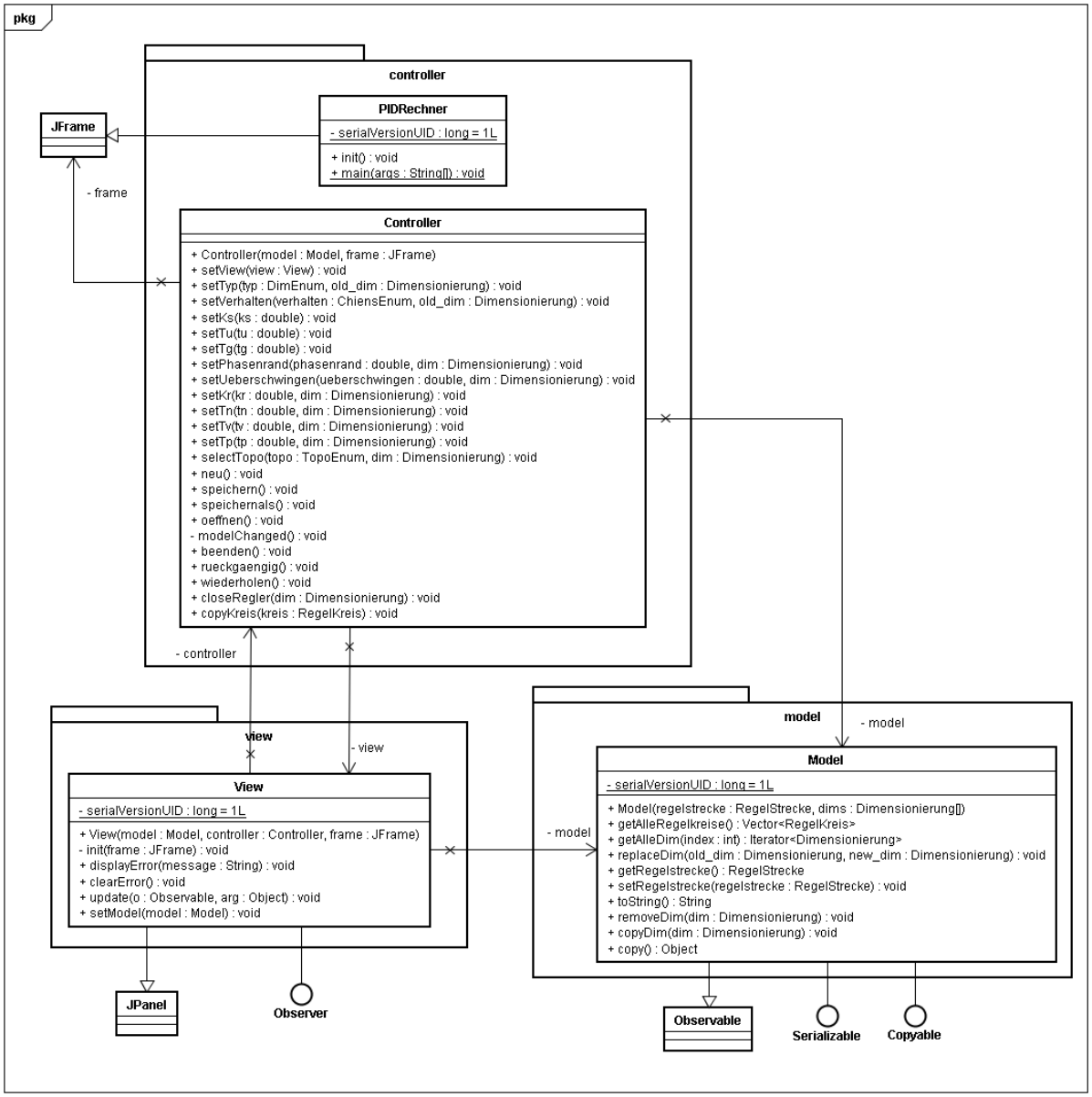


Abbildung 5: Package *controller*

3.1.2 Ausgewählte Klassen des Packages *model*

Der Aufbau des Packages *model* ist in Abbildung 6 ersichtlich.

Model
Das *Model* verwaltet die *Regelstecke* und die verschiedenen *Dimensionierungen*. Es bietet Zugriffsmethoden um die *Regelstrecke* und *Dimensionierungen* zu modifizieren. Auch kann es Dimensionierungsmethoden kopieren oder löschen.

Regelkreis
Die Klasse *Regelkreis* dient zur Umwandlung der Übertragungsfunktionen (*TransferFunction*) von Regelstrecke und Regler in die Übertragungsfunktion des ganzen Kreises.

Regelstrecke
Die Klasse *Regelstrecke* bietet eine Abstraktion einer PTn-Strecke mit den Attributen K_s, T_u und T_g. Eine *Regelstrecke* ist ein *Regelglied*, das heisst sie kann als eine Übertragungsfunktion angesehen werden. Die Übertragungsfunktion wird durch die Sani-Approximation (*SaniApprox*) gebildet.

Regler
Diese Klasse dient zur Abstraktion eines PID-T Reglers mit den Parametern K_R, T_n, T_v und T_p. Ein *Regler* wird normalerweise durch eine Dimensionierungsmethode (*Dimensionierung*) gebildet.

TransferFunction
Eine Übertragungsfunktion besteht aus Zähler- und Nenner-*Polynom*. Sie bietet Methoden zur Faltung und Berechnung der Residuen. Weiter kann aus jeder Übertragungsfunktion eine *Schrittantwort* gebildet werden.

Polynom
Polynom abstrahiert reelle Polynome beliebigen Grades. Es bietet Methoden zur Addition und Multiplikation mit anderen *Polynomen*. Weiter können die Wurzeln und die Residuen (mit einem angenommenen Zähler von 1) ausgelesen werden.

Schrittantwort
Die Klasse *Schrittantwort* stellt die Schrittantwort für ein lineares *Regelglied* ohne Durchgriff dar. Sie bietet zudem Methoden zur Analyse. So kann sie feststellen ob eine Schrittantwort ausschwingt und wie gross An- und Ausschwingzeit sind. Weiter sucht sie nach dem Maximalwert der Kurve und bietet die Möglichkeit dessen Zeitpunkt und Grösse auszulesen.

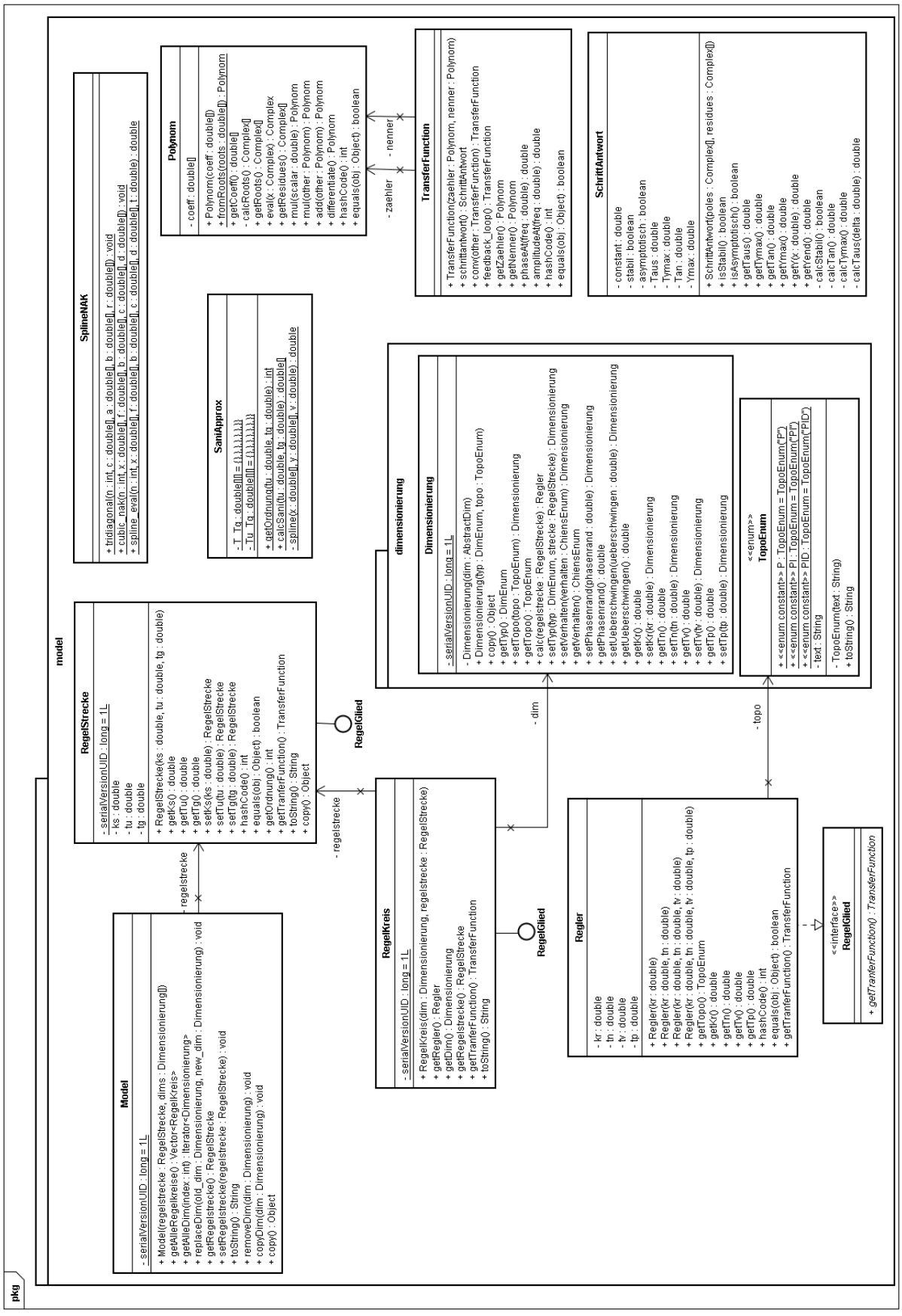


Abbildung 6: Package *model*

3.1.3 Ausgewählte Klassen des Packages *dimensionierung*

Der Aufbau des Packages *dimensionierung* ist in Abbildung 7 ersichtlich.

Dimensionierung
Diese Klasse ist eine Auswahl der anderen Dimensionierungsarten, wobei immer nur eine Dimensionierungsart aktiv ist. So bietet diese Klasse auch alle Methoden der anderen Dimensionierungsmethoden. Wird eine Methode bei einer falschen Variante verwendet, wirft die Klasse eine *ClassCastException*. Der Typ (*DimEnum*) kann mit der Methode setTyp gewechselt werden. Beim Wechseln zu der Manuellen Dimensionierungsmethode, werden die Werte des dimensionierten Reglers übernommen.

DimEnum
Dieser Enum dient zur Aufzählung aller Dimensionierungstypen. Er ordnet jedem Typ auch einen Text zu.

TopoEnum
Zur Unterscheidung von PI- und PID-Reglern wird dieser Enum verwendet.

AbstractDim
AbstractDim ist die abstrakte Grundklasse aller Dimensionierungsmethoden. Die Methode calc berechnet für eine gegebene Regelstrecke einen Regler. Durch get- und setTopo lässt sich die Reglertopologie (*TopoEnum*) modifizieren. Die Methode getTyp() gibt den Typ des konkreten *Reglers* zurück.

ZellwegerDim
Hier steckt die Phasengangmethode zur Reglerdimensionierung. Die Beschreibung der Methode findet sich im theoretischen Teil.

IterativDim
Diese Methode führt die Phasengangmethode (*ZellwegerDim*) mehrmals nacheinander aus und sucht mit einer binären Suchmethode nach dem gewünschten Überschwingen der Schrittantwort.

Die restlichen Klassen entsprechen den verschiedenen Faustregeln.

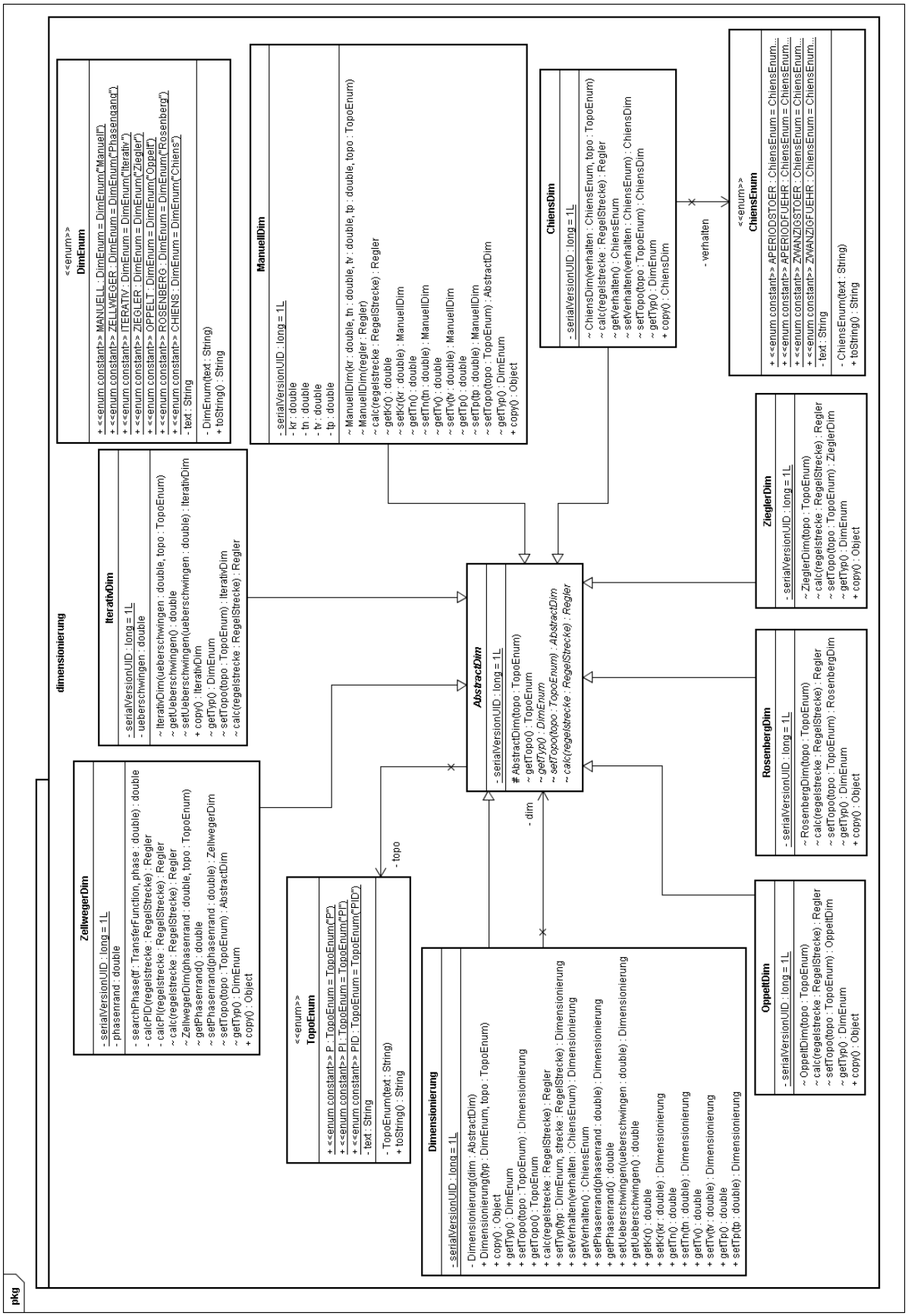


Abbildung 7: Package *dimensionierung*

3.1.4 Klassen des Packages *view*

Der Aufbau des Packages *view* ist in Abbildung 8 ersichtlich.

View
View ist die oberste Ebene im GUI (abgesehen vom *JFrame*). Es beheimatet die *Sidebar*, den *Graph* und die Statuszeile (**6**). Auch initialisiert es die Menübar (**5**) vom *JFrame*.

SidebarPanel
SidebarPanel ordnet die einzelnen Panels von *RegelstreckeView*, *ReglerView* und *AnalyseView* vertikal untereinander an. Es können mehrere Instanzen von *ReglerView* angezeigt werden.

RegelstreckeView (**1**)
RegelstreckeView zeigt die Parameter der *Regelstrecke* sowie die berechnete Ordnung und die Zeitkonstanten an.

ReglerView (**2**)
Dieses *JPanel* visualisiert den Typ der Dimensionierungsmethode, deren Parameter und die Werte des daraus resultierenden *Reglers*.

AnalyseView (**3**)
Listet die Eigenschaften Anstiegszeit (T_{an}), Ausschwingzeit (T_{aus}), Maximalwert (Y_{max}) und Zeitpunkt des Maximalwertes (T_{Ymax}) der *Schrittantwort* auf.

Graph (**4**)
Zeichnet alle *Schrittantworten* mit der Klasse *ChartPanel* von JFreeChart. Unter dem Graphen zeigt es weiter eine Legende. Die Zustände der *Checkboxen* werden nur vom Graph selbst verwendet und nicht dem *Controller* mitgeteilt.

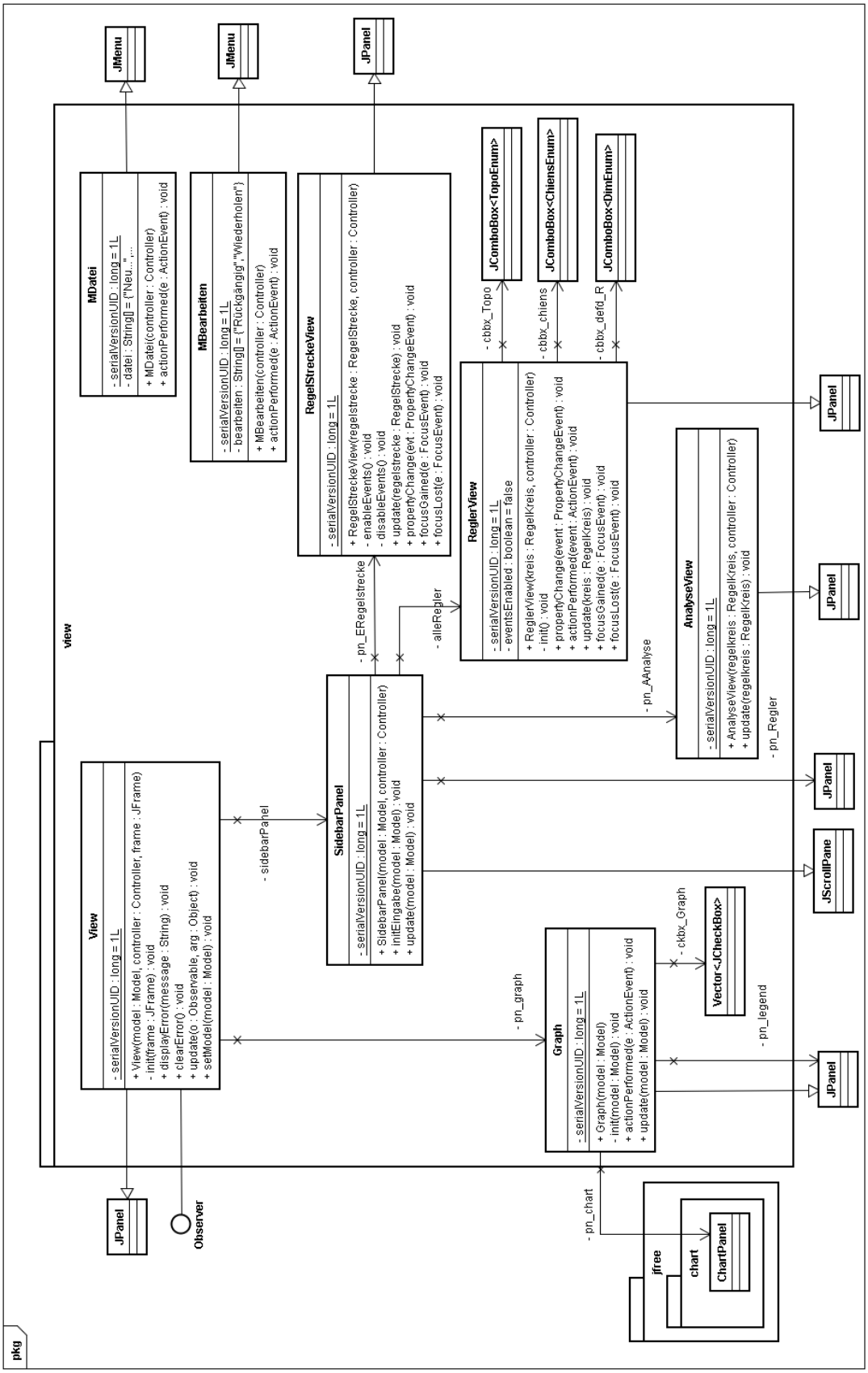


Abbildung 8: Package *view*

3.2 Benutzerinteraktion

Vorgaben: Das Programm wurde gestartet und im Eingabefeld T_u wurde der Wert 1.0 eingegeben. Weiter wurde als Dimensionierungsart Iterativ ausgewählt. Nun gibt man im Eingabefeld T_g den Wert 1.6 ein. Dadurch wird folgender Ablauf gestartet:

- In der Klasse *RegelStreckeView* wird durch das System die Methode propertyChange des *PropertyChangeListeners* aufgerufen. Es wird die Quelle des Events überprüft und festgestellt, dass es von dem Textfeld für T_g kommt. Danach wird der Zahlenwert des Textfeldes eingelesen und an die Methode setTg vom *Controller* weitergeleitet.

- Der *Controller* macht sich eine Kopie des *Models* und speichert sich diese auf einem Stack. Danach bildet er sich eine neue *Regelstrecke* mit der Methode setTg von der alten *Regelstrecke*. Nun setzt der *Controller* mit der Methode setRegelstrecke die Regelstrecke des *Models* auf die neue Instanz, dabei wird als Nebeneffekt die Methode update vom *View*, welches ein Observer ist, aufgerufen.

- Das *View* ruft nun die Methode update von den beiden Klassen *SideBarPanel* und *Graph* auf.

- Die update Methode vom *SideBarPanel* ruft update von *RegelStreckeView*, von allen *ReglerViews* und von *AnalyseView* auf.

- *RegelStreckeView* setzt die Werte seiner Textfelder auf die Werte im *Model*. Ist die Checkbox „zeige T“ angewählt, zeigt es zusätzlich die Textfelder für die Streckenzeiten der Sani-Approximation.

- Jedes *ReglerView* setzt nun in seiner update Routine die Werte seiner Textfelder auf die Werte der *Dimensionierung* bzw. auf die Werte des dimensionierten *Reglers*.

- Die update Methode von *AnalyseView* holt sich mit getTransferFunction() vom *Regelkreis* die *Übertragungsfunktion* und berechnet mittels schrittantwort() deren *Schrittantwort*. Danach setzt es die Werte seiner Textfelder auf die Eigenschaften der *Schrittantwort*.

- Der Kontrollfluss springt zurück ins update vom *View*. Dieses ruft nun update vom *Graphen* auf.

- Der *Graph* erstellt wie das *AnalyseView* die *Schrittantwort* des *Regelkreises* für jeden *Regler*. Aus der *Schrittantwort* werden 300 Punkte gelesen, die dann in den Datensatz vom JFreeChart eingefügt werden.

- Nun springt der Kontrollfluss zurück in den *Controller*. Hat sich bis jetzt kein Fehler ereignet, wird die Methode clearError des *Views* aufgerufen, ansonsten wird displayError mit der Fehlermeldung aufgerufen.

4 Schlussfolgerung

Das Ergebnis am Ende dieses Projektes ist eine Software welche auf dem MVC-Pattern aufgebaut ist. Die Streckenparameter können in der Software eingegeben werden und es resultieren die Parameter des Reglers mit der Schrittantwort der Regelungsstrecke. Die Schrittantwort der Regelung wird unmittelbar vermasst und die Vermassungswerte werden im Analyse Panel dargestellt. Damit verschiedene Regler verglichen werden können, besteht die Möglichkeit, mehrere Plots zu generieren. Das Ziel, dass die Software einfach zu bedienen ung somit benutzerfreundlich sein soll, haben wir mit einer übersichtlichen Darstellung und geeigneten Tastenkombinationen (Shortcuts) realisiert.
Im Grossen und Ganzen haben wir die gesetzten Ziele erreicht. Insbesondere die Soll-Ziele wurden alle erreicht.. Nicht erreicht wurde das Wunschziel der Monte-Carlo-Analyse. Die Idee war, durch zufällige Wahl der Eingabewerte in einem gewissen Prozentbereich, verschiedenste Regelungen zu berechnen und somit den maximalen Fehler optisch darstellen zu können. Aus zeitlichen Gründen konnte dieses Ziel jedoch nicht weiter verfolgt und realisiert werden.

5 Ehrlichkeitserklärung

Hiermit erklärt der Unterzeichnende (Projektleiter), dass die vorliegende Arbeit selbstständig, ohne Hilfe Dritter und nur unter Benutzung der angegebenen Quellen verfasst worden ist.

Alex Stocker (Projektleiter): _____

Datum, Ort: _____, _____

6 Literatur

[1] RN-Wissen. Regelungstechnik. http://rn-wissen.de/wiki/index.php/Datei:Regelkreis4.png (08.05.2015).
[2] D. Solenicki and Y. Bürgi, "Mathematisches Labor mlab - Systemidentifikation mit Hilfe der Schrittantwort," November 2013, FHNW.
[3] J. Zellweger, "Phasengang-Methode," unveröffent. Vorlesungsskript, ohne Jahr.
[4] ——, "Regelkreise und Regelungen," unveröffent. Vorlesungsskript, ohne Jahr.
[5] B. Bate, "Spezialgebiete der Steuer- und Regelungstechnik," 2009.

Abbildungsverzeichnis

1 Regelung bestehend aus Regler und Regelstrecke [1] 7
2 Schrittantwort einer Regelstrecke [2] 8
3 Amplitudengang eines PI-Reglers in Abhängigkeit der Kreisfrequenz ω (log. Darstellung) [3] 9
4 Amplitudengang eines PID-Reglers in Abhängigkeit der Kreisfrequenz ω (log. Darstellung) [3] 10
5 Package *controller* 18
6 Package *model* 20
7 Package *dimensionierung* 22
8 Package *view* 24

Tabellenverzeichnis

1 Amplitudengänge / Knickkreisfrequenzen 10
2 Wichtige Phasengangpunkte 10
3 Zusammenhang Phasenrand und Überschwingen 11
4 Übertragungsfunktionen Regler [4] 13
5 Fausformeln Ziegler/Nichols, Oppelt und Rosenberg [5] 13
6 Chien/Hrones und Reswick [5] 14

7 Anhang

- Herleitungen Formeln
- Beispielberechnung Phasengangmethode mithilfe von Matlab
- Bedienungsanleitung des Programmes
- Klassendiagramm
- CD-ROM

Herleitung β

$$\varphi_R(\omega) + \varphi_S(\omega) = \varphi_0(\omega)$$

$$\frac{d\varphi_R(\omega_{PID})}{d\omega} + \frac{d\varphi_S(\omega_{PID})}{d\omega} = \frac{d\varphi_0(\omega_{PID})}{d\omega} = \frac{-0,5}{\omega_{PID}}$$

$$\omega_{PID}\frac{d\varphi_R(\omega_{PID})}{d\omega} + \omega_{PID}\frac{d\varphi_S(\omega_{PID})}{d\omega} = \omega_{PID}\frac{d\varphi_0(\omega_{PID})}{d\omega} = -0,5$$

$$\omega_{PID}\frac{d\varphi_R(\omega_{PID})}{d\omega} = \frac{2\beta}{1+\beta^2}$$

$$\frac{2\beta}{1+\beta^2} + \omega_{PID}\frac{d\varphi_S(\omega_{PID})}{d\omega} = -0,5$$

$$\frac{2\beta}{1+\beta^2} = \underbrace{-0,5 - \omega_{PID}\frac{d\varphi_S(\omega_{PID})}{d\omega}}_{\text{Substitution} = z}$$

$$\frac{2\beta}{1+\beta^2} = z$$

$$2\beta = z + z\beta^2$$

$$0 = z\beta^2 - 2\beta + z$$

$$\beta_{1,2} = \frac{2 \pm \sqrt{4 - 4z^2}}{2z} = \frac{1 \pm \sqrt{1 - z^2}}{z}$$

$$\beta_{1,2} = \frac{1}{z} \pm \sqrt{\frac{1}{z^2} - 1}$$

Kommentare / Erklärungen:

$\varphi_R(\omega)$ Phasengang Regler
$\varphi_S(\omega)$ Phasengang Strecke
$\varphi_0(\omega)$ Phasengang offene Regelung

Steigung im Punkt ω_{PID} beträgt $-\frac{0,5}{\omega_{PID}}$

wird in obige Gleichung eingesetzt

Herleitung Umrechnung bodekonform → reglerkonform

Kommentare Erklärungen:

$$\frac{K_{RN}(1+sT_{nN})(1+sT_{VN})}{sT_{nN}(1+sT_p)} = K_R\left(1+\frac{1}{sT_n}+\frac{sT_V}{1+sT_p}\right)$$

Übertragungsfunktion bodekonform = Ü-funk reglerkonform

$$\frac{K_{RN}\left(s^2(T_{nN}T_{VN})+s(T_{nN}+T_{VN})+1\right)}{sT_{nN}(1+sT_p)} = \frac{K_R\left(s^2(T_nT_p+T_nT_V)+s(T_n+T_p)+1\right)}{sT_n(1+sT_p)}$$

kürzen $\frac{1}{s(1+sT_p)}$

$$\frac{K_{RN}}{T_{nN}}\left(s^2(T_{nN}T_{VN})+s(T_{nN}+T_{VN})+1\right) = \frac{K_R}{T_n}\left(s^2(T_nT_p+T_nT_V)+s(T_n+T_p)+1\right)$$

mit s gegen 0 $\Rightarrow$ $\frac{K_{RN}}{T_{nN}} = \frac{K_R}{T_n}$

kann auch gekürzt werden

$$s^2(T_{nN}T_{VN})+s(T_{nN}+T_{VN})+1 = s^2(T_nT_p+T_nT_V)+s(T_n+T_p)+1$$

Koeffizienten müssen gleich sein

$$T_{nN}T_{VN} = T_nT_p+T_nT_V \qquad T_{nN}+T_{VN} = T_n+T_p$$

$$T_n = T_{nN}+T_{VN}-T_p$$

$$T_V = \frac{T_{nN}T_{VN}}{T_n}-T_p$$

$$T_V = \frac{T_{nN}T_{VN}}{T_{nN}+T_{VN}-T_p}-T_p$$

$$\frac{K_R}{T_n} = \frac{K_{RN}}{T_{nN}} \Rightarrow K_R = T_n\cdot\frac{K_{RN}}{T_{nN}}$$

$$K_R = \frac{K_{RN}(T_{nN}+T_{VN}-T_p)}{T_{nN}}$$

$$K_R = K_{RN}\left(1+\frac{T_{VN}}{T_{nN}}-\frac{T_p}{T_{nN}}\right)$$

Dimensionierung PI-Regler mit Phasengangmethode

Die Schrittantwort ist mit den folgenden Werten gegeben:

$T_u = 1.11s \quad T_g = 8.62s \quad K_S = 1$

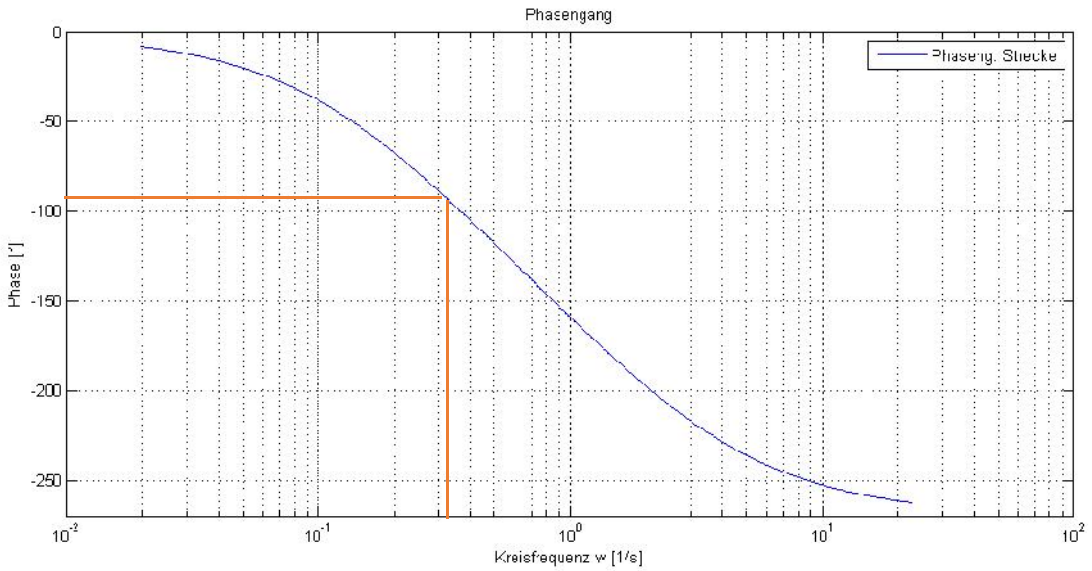
Die Sani-Methode ergibt:

$T_1 = 0.4396s \quad T_2 = 1.4967s \quad T_3 = 5.0952s$

Daraus folgt:

$$Gs\ s = K_S \frac{1}{1 + s * 0.4396 \quad 1 + s * 1.4967 \quad 1 + s * 5.0952}$$

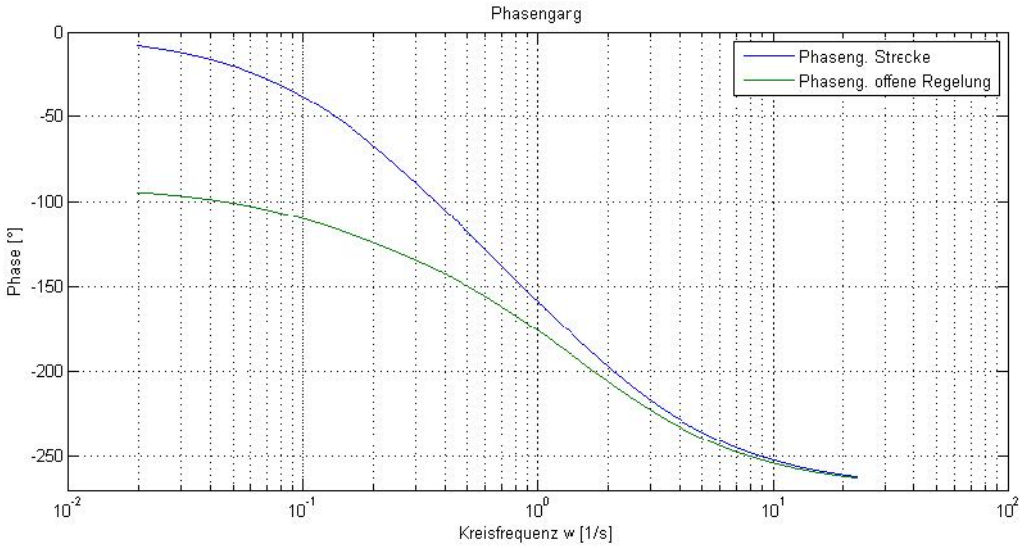
Im Phasengang der Strecke wird nun der -90°-Punkt gesucht.



Bei -90°: $\omega_{PI} = 0.308 \frac{1}{s} = \frac{1}{T_n} \quad T_n = 3.24s$

Durch Übertragungsfunktion des PI-Reglers kann man den Phasengang des Reglers bestimmen. Die beiden Phasengänge werden nun addiert.

$$G_R\ s\ =\ K_R\ \frac{1\ +\ sT_n}{sT_n}$$



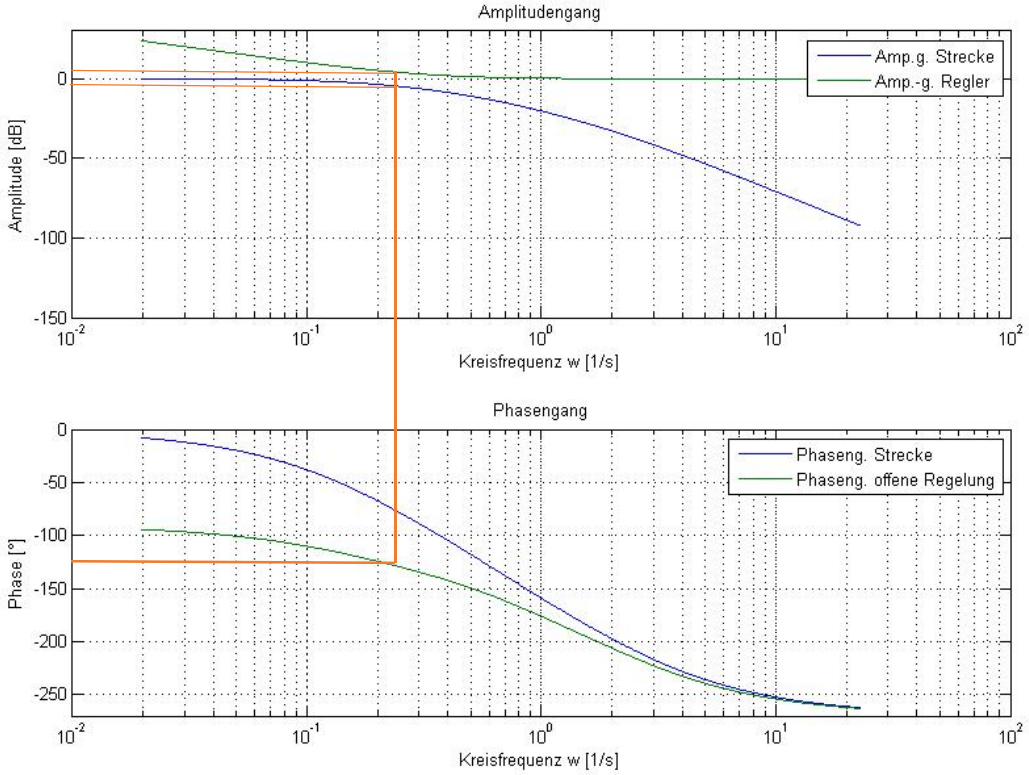
Die Amplitudengänge des Reglers und der Strecke werden nun benötigt um die Verstärkung des Reglers zu bestimmen. Dazu wird der Phasenrand (je nach gewünschtem Überschwingen) im Phasengang der offenen Regelung abgetragen. In diesem Beispiel beträgt er 51.8°, d.h. im Phasengang muss der -128.2-Punkt gesucht werden.

Bei -128.2°-Punkt: $\omega_D = 0.2374 \frac{1}{s}$

$Gs_D = 0.5871$ $Gr_D = 1.6390$ $Go_D = Gs_D * Gr_D = 0.9787$

$K_R = \frac{1}{|Go_D|} = 1.0218$

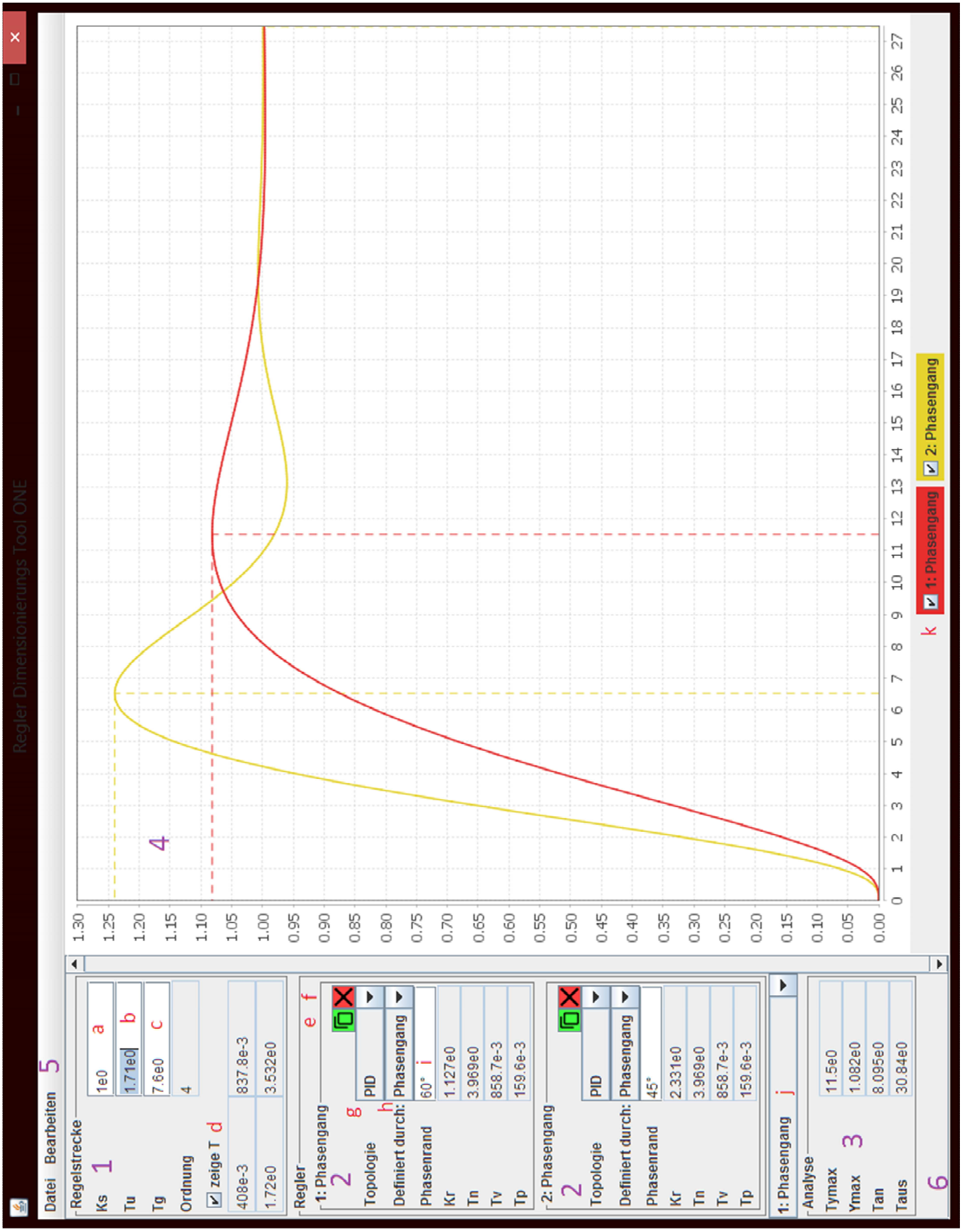
Achtung: Diese Werte sind linear, die Plots sind aber logarithmisch dargestellt!



Damit sind alle Parameter des PI-Reglers bestimmt und somit kann die vollständige Übertragungsfunktion weiterverwendet werden.

Bedienungsanleitung

1. Der Streckenbeiwert Ks kann bei **(a)** eingeben werden.
2. Die Verzugszeit Tu kann in **(b)** eingegeben werden.
3. Die Anstiegszeit Tg kann in **(c)** eingegeben werden.
   Diese Parameter können direkt aus der Schrittantwort der Strecke bestimmt werden, falls diese vorhanden ist.
   Die Ordnung der Regelstrecke wird mit dem Verhältnis Tu/Tg bestimmt.
4. Mit „zeige T“ **(d)** können die Streckenzeiten angezeigt werden
5. Mit der grünen Box **(e)** kann ein neuer Graph hinzugefügt werden. Mit der roten Box **(f)** kann ein Graph entfernt werden.
6. Die Topologie (PI, PID) des Reglers wird durch **(g)** ausgewählt.
7. Die Definition (Manuell, Phasengang, Ziegler, Oppelt, Rosenberg oder Chiens) des Reglers wird bei **(h)** bestimmt.
8. Falls die Definition Phasengang ausgewählt wird, muss der Phasenrand **(i)** vorgegeben werden. Der Phasenrand bestimmt die Verstärkung und somit das Überschwingen des Reglers.
9. Falls die Definition Chiens ausgewählt wird, muss das gewünschte Verhalten bei **(i)** noch bestimmt werden. Nun sind alle Eingaben getätigt und die Reglerwerte (Kr, Tn, Tv und Tp) werden Berechnet.
10. Bei **(j)** kann ausgewählt werden welche Schrittantwort analysiert werden, die Eigenschaften der Schrittantwort werden dann darunter angezeigt. Die Eigenschaften sind der Maximalwert (Ymax), der Zeitpunkt des Maximalwertes (Tymax), die Anschwingzeit(Tan) und die Ausschwingzeit (Taus).
11. Mit den Checkboxen **(k)** unter dem Graph lässt sich auswählen, welche Schrittantworten gezeichnet werden.
12. Wenn die Dimensionierung abgeschlossen ist kann die Datei unter Datei->Speichern (Ctrl-S) gespeichert werden.
13. Die Datei kann unter Datei->Öffnen (Ctrl-O) wieder geöffnet werden.
14. Eine allfällige Fehleingabe kann unter Bearbeiten->Rückgängig (Ctrl-Z) widerrufen werden. Mit Bearbeiten->Wiederholen (Ctrl-Y) kann die Änderung wiederholt werden.
15. Durch das Drücken der Maustaste im Graphen und ziehen nach rechts unten, kann gezoomt werden. Alternativ kann das Rechtsklick-Menü verwendet werden.


Regler Dimensionierungs Tool ONE
Datei Bearbeiten
5
Regelstrecke
Ks 1e0 a
Tu 1.71e0 b
Tg 7.6e0 c
Ordnung 4
zeige T d
408e-3 837.8e-3
1.72e0 3.532e0
Regler
1: Phasengang
2
e f
g
Topologie PID
h
Definiert durch: Phasengang
Phasenrand 60° i
Kr 1.127e0
Tn 3.969e0
Tv 858.7e-3
Tp 159.6e-3
2: Phasengang
2
Topologie PID
Definiert durch: Phasengang
Phasenrand 45°
Kr 2.331e0
Tn 3.969e0
Tv 858.7e-3
Tp 159.6e-3
1: Phasengang j
Analyse
Tymax 11.5e0
Ymax 3 1.082e0
Tan 8.095e0
Taus 30.84e0
6
4
k
1: Phasengang
2: Phasengang