

# Entwicklung eines Verfahrens zum Reglertuning für Anlagen in der Lebensmittelindustrie

Diplomarbeit angefertigt an der  
Universität Lüneburg,  
Fakultät III – Umwelt & Technik,  
Bereich Automatisierungstechnik,  
Studiengang Angewandte Automatisierungstechnik  
zur Erlangung des akademischen Grades eines  
Dipl.-Ing. (FH)

Vorgelegt von  
Axel Kern  
aus  
Lüneburg

<b>Erstprüfer:</b>	Prof. Dr.-Ing. Philipp Odensass
<b>Zweitprüfer:</b>	Prof. Dr.-Ing. Klaus-Dieter Hübner
<b>Betreuer im Unternehmen:</b>	Dipl.-Ing. Jürgen Dechow
<b>Fachlicher Betreuer im Unternehmen:</b>	Dipl.-Ing. Tino Horn
<b>Tag des Kolloquiums:</b>	28.09.2006

## **Vorwort**

Die vorliegende Diplomarbeit habe ich im Rahmen meiner praktischen Tätigkeit bei der Firma GEA Tuchenhagen Dairy Systems GmbH (GEA TDS) in Büchen angefertigt. In der heutigen Zeit sind in vielen Unternehmen Prozesse ohne Regler undenkbar geworden, gerade wenn es darum geht Prozesse in der Industrie, insbesondere auch in der Produktion, automatisch ablaufen zu lassen und eine gewisse Konstanz zu halten. Besonders in der Lebensmittelindustrie ist es umso wichtiger Regelungsprozesse für z.B. Temperatur und Niveau prozessgerecht arbeiten zu lassen, um ein hohes Maß an Qualität zu gewährleisten. Deshalb liegt es der GEA TDS bei der Inbetriebnahme ihrer Molkereianlagen am Herzen, die Regelgüte ihrer Regler für die Prozesse zu optimieren, um so mit einem ruhigen Gewissen von der Baustelle zu gehen.

An dieser Stelle möchte ich denjenigen danken, die mir dieses Studium überhaupt ermöglicht haben. Als erstes sehr großen Dank an meine Mutter und meinen Bruder, die mir in meinen Sturm- und Drangphasen oft aus schwierigen Situationen herausgeholfen haben und mich immer wieder ermutigt haben, mich weiter zu bilden und nicht aufzugeben. Ein großes Dankeschön habe ich auch an meine Freunde, meine Ex-Freundin Emilia und an meine heutige Lebensgefährtin Anke zu richten, die meine Launen und meinen Zeitmangel während des Studiums, in den Klausurzeiten und in der Diplomzeit ertragen mussten und immer vollstes Verständnis dafür hatten. Als letztes gilt mein Dank auch noch meinen Arbeitskollegen bei der Firma GEA Tuchenhagen Dairy Systems, die mir bei Verständnisproblemen immer mit Rat und Tat zur Seite standen.

## **Kurzzusammenfassung**

In dieser Diplomarbeit ist die Entwicklung eines Software -Tools zur Optimierung von PID-Reglern beschrieben. Das Programm ist, durch die Vorgabe der GEA TDS, in Visual BASIC 6 programmiert worden und ist speziell auf die Optimierung der Standard-PID-Regler in der Siemens S7-Steuerung ausgelegt. Die Verbindung zu der Steuerung findet über einen OPC-Server statt. Der Anwender kann von seinem Laptop aus auf den OPC-Server zugreifen, auf dem die Access-Datenbank, in der die PID-Regler konfiguriert sind, abgelegt ist.

Die entwickelte Anwendung soll in den von GEA TDS gebauten Anlagen eingesetzt werden, um den Anwendern des Programms das Einstellen der PID-Regler zu vereinfachen und eine gute Reglergüte beim Kunden zu hinterlassen.

## **Abstract**

In this diploma thesis the development of a software tool of the optimisation is described by PID-controller. The program has been programmed, by the demand of the GEA TDS, in Visual BASIC 6 and is laid-out specially on the optimisation of the standard-PID-controllers in the Siemens S7 control. The connection with the control takes place about an OPC server. The user can access from his laptop the OPC-server on which the Access-database where the PID-controllers are configured is filed.

The developed application should be used in the arrangements built by GEA TDS to simplify the adjusting of the closed loop controllers to the users of the program and to leave a good PID-controllers goodness with the customer.

## Inhaltsverzeichnis

Vorwort.....	I
Kurzzusammenfassung / Abstract.....	II
1 Einleitung .....	1
2 Grundlagen .....	6
2.1 Ausgangssituation .....	6
2.2 Abgrenzung der Steuerung zur Regelung .....	7
2.3 Grundlegendes zur Regelungstechnik.....	9
2.4 Grundlegendes zur Steuerungstechnik .....	17
2.5 Programmrelevante Verbindungen.....	21
2.5.1 OPC-Verbindung.....	22
2.5.2 ODBC.....	22
3 Standard PID Control V5.0.1 [4].....	24
3.1 Das Konzept des Standard PID-Reglers .....	24
3.2 Übersicht über die Grundfunktionen des Standard PID-Reglers .....	25
3.3 Erstellen der Regelung .....	26
3.4 Einsatzumgebung und Anwendungsbereich .....	26
3.5 Die Abtastzeit CYCLE .....	30
3.6 Entwerfen von digitalen Regelungen .....	31
3.7 Struktur und Arbeitsweise des Standard PID Control.....	31
4 Verfahren zur Regleroptimierung .....	36
4.1 Das Tangentenverfahren.....	36
4.2 Optimierung nach Ziegler und Nichols [3].....	38
4.3 Optimierung nach Takahashi [3].....	41
5 Softwareentwicklung .....	43
5.1 Die Startform frmVerbindung.....	45
5.1.1 Die Verbindung zur Datenbank.....	46
5.1.2 Beispiel einer Datenbankverbindung .....	47
5.1.3 Lesen der Daten aus einem Recordset.....	48
5.2 Die Form frmReglerauswahl.....	50
5.2.1 Das DataGrid .....	51
5.2.2 Die Objekte im Frame1 .....	52

---

5.2.2.1 Die Verbindung zur Steuerung .....	53
5.2.2.2 Lesen der Daten aus der Steuerung.....	54
5.2.3 Die Objekte im Frame2 .....	57
5.2.3.1 Schreiben der Daten in die Steuerung.....	58
5.2.4 Starten der Aufzeichnung der Sprungantwort in der PictureBox.....	59
5.3 Die Form frmReglerbearbeitung .....	64
5.3.1 Einstellen der Min- und Maxwerte der Stellwertgröße.....	65
5.3.2 Starten der Aufzeichnung der Sprungantwort der Strecke.....	66
5.4 Vergleich der Sprungantworten auf einen Sollwertsprung.....	75
6 Praktischer Test an der Anlage .....	77
7 Zusammenfassung.....	79
8 Ausblick.....	80
Literaturverzeichnis .....	81
Abkürzungsverzeichnis .....	83
Formelzeichenverzeichnis.....	85
Inhalt der CD-ROM .....	86
Anhang Programm Quellcode .....	87
Erklärung zur Diplomarbeit.....	127

# 1 Einleitung

Die GEA Tuchenhagen Dairy Systems GmbH ist ein international tätiges Unternehmen für technisch und wirtschaftlich optimierte Prozesstechnologie zur Verarbeitung von Milch, Nahrungsmitteln und Saft. Die Gesellschaft hat circa 270 Mitarbeiter und erzielt jährlich einen Umsatz von 80 Millionen Euro. Sie entstand aus einem Zusammenschluss folgender Gesellschaften:

- Tuchenhagen, Bereich Nahrungsmittel und Milch (gegründet 1931, Erfinder des Doppelsitzventils, Mitglied der GEA Gruppe seit 1995, Sitz Büchen, Vertretungen weltweit)
- GEA Ahlborn (gegründet 1856, Pionier in der Milchindustrie und Erfinder des Plattenwärmetauscher, Mitglied der GEA Gruppe seit 1979, Sitz Sarstedt)
- GEA Finnah (gegründet 1975, führend in der aseptischen Prozesstechnologie, Röhrenwärmetauschern, UHT-Anlagen, Mitglied der GEA Gruppe seit 1989, Sitz Ahaus)

Inzwischen wurden diese drei erfolgreichen Gesellschaften unter dem Namen Tuchenhagen Dairy Systems zusammengeführt und sind heute wie folgt organisiert:



## **Sarstedt**

Milch  
Fruchtsaft  
Montage



## **Büchen**

Nahrungsmittel  
Export  
Automation



## **Ahaus**

Aseptik

**Abbildung 1.1:** Die GEA Tuchenhagen Dairy Systems GmbH [GEA TDS]

Ein Beispiel für die Prozesstechnik bei der GEA TDS ist die nachfolgend abgebildete Anlage. Die Prozesslinien stammen aus einer Hand und sind eine wirkungsvolle Verbindung von Hygiene und Produktivität.



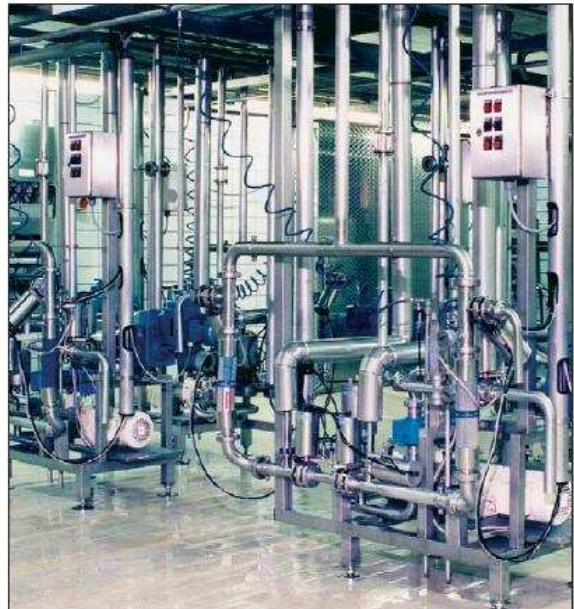
Milch-Kurzzeiterhitzung



Ventilknoten



Milch-Annahme



Fruchtmischstation

**Abbildung 1.2:** Molkereianlage der GEA Tuchenhagen Dairy Systems [GEA TDS]

Damit ein Unternehmen in der heutigen Zeit international konkurrenzfähig sein kann ist es notwendig, die Produktionskosten so weit wie möglich zu senken. Dieses wird meistens durch einen hohen Automatisierungsgrad erreicht, womit die zu steuernden Anlagen immer größer und komplexer werden können. Dadurch kann auch die Anzahl der Produkte, die in einem Arbeitsdurchgang erstellt werden, viel größer werden, als noch vor einigen Jahrzehnten. Der Nachteil an den immer größer werdenden Mengen und Produktionsanlagen ist jedoch, dass kleine Fehler große Folgen haben können. Die Produkte können fehlerhaft sein durch:

- ein minderwertiges oder belastetes Produkt vom Erzeuger
- einen Bedienungsfehler in der Produktionsanlage
- einen Hard- oder Softwarefehler in der Produktionsanlage

Durch die großen Produktionsmengen und die internationalen Konzerne erreicht auch ein fehlerhaftes Produkt nicht mehr nur eine Person, Familie oder die Käufer eines Supermarktes, sondern auch die Käufer einer ganzen Supermarktkette - vielleicht sogar in mehreren Ländern. Beispiele hierfür waren in den letzten Jahren immer wieder in den Medien zu hören und zu sehen.

In Israel starben mindestens zwei Säuglinge an Vitamin B1 Mangel, weil in der Firma Humana, in der die Babymilch für Säuglinge hergestellt wurde, einer Mitarbeiterin ein Rechenfehler bei der Rezepturerstellung unterlief.

Bekannt dürften dem Leser die vielen Rückrufaktionen in der Automobilindustrie sein und auch die Software des weltgrößten Softwareherstellers Microsoft musste nach den Auslieferungen noch etliche Male nachgebessert werden.

Anhand dieser Beispiele muss man leider feststellen, dass Fehler menschlich sind und immer wieder passieren. Die Folge ist dann natürlich aber auch eine Unsicherheit beim Verbraucher. Natürlich muss versucht werden, die Fehlerrate immer weiter zu senken und wenn möglich Fehler ganz auszuschließen. Um Produkte in den von GEA TDS gefertigten Anlagen rückverfolgen zu können und somit Fehler schnell zu erkennen oder vermeiden zu können, wurde im Jahr 2005 durch eine Diplomarbeit ein grafisches Frontend zur Chargenverfolgung entwickelt, welches den Namen Track & Tracing hat.



Auf die große Besorgnis der Verbraucher reagieren die Gesetzgeber. In den letzten Jahren wurden immer mehr gesetzliche Richtlinien beschlossen, wie es aus der folgenden Abbildung zu entnehmen ist.



**Abbildung 1.3:** Verordnung zum Verbraucherschutz von Lebensmitteln [GEA TDS]

Eine große Aufgabe für die Lebensmittelindustrie ist es, im Falle eines Fehlers den Schaden für die Menschen sowie für die Firma so gering wie möglich zu halten. Dazu ist es vorteilhaft, den Fehler möglichst früh zu erkennen. Tritt ein Defekt in der Produktionsanlage auf, so wird in gut programmierten Anlagen sofort eine Warnung auf den Bildschirmen der Visualisierung im Kontrollraum angezeigt. Der Bediener kann in diesem Fall sofort reagieren und nötige Schritte einleiten, sofern es die Steuerung nicht schon von sich aus getan hat.

Die Motivation bezogen auf den Einsatz technischer Regler und deren Verbesserung bezüglich ihrer Regelgüte ist:

- Geschwindigkeit
- Präzision
- Sicherheit
- Robustheit
- Automatisierung, Wiederholbarkeit

Die GEA TDS baut weltweit Anlagen zur Erzeugung von Milchprodukten. Bei der Inbetriebnahme der Anlagen ist es wichtig die Regler so einzustellen, dass stabile Prozesse entstehen, um Ausfälle, Schäden oder Produktausschuss zu vermeiden. Deswegen ist es für die Firma von Bedeutung, am Ende ein gutes Produkt an den Kunden zu übergeben, um einen Imageschaden zu vermeiden und somit weltweit konkurrenzfähig zu bleiben.

Die Automation ist ein wesentlicher Bestandteil der Firma, die Ihren Platz in Büchen hat. Hier wird das komplette Software-Engineering abgewickelt, welches unter anderem aus der Programmierung von Steuerungen, Programmen und Datenbanken sowie der Visualisierung von Prozessen besteht.

Diese Diplomarbeit entstand aus dem Wunsch der GEA TDS heraus, eine für die Firma zugeschnittene Software zu besitzen, mit der die Standard-PID-Regler der S7-Steuerung vorzugsweise für Prozesse wie Temperatur, Niveau und Durchfluss einfach und schnell nach bekannten Einstellregeln wie z.B. Ziegler und Nichols optimieren werden können. Der praktische Nutzen dieses Programms soll nach Fertigstellung an der laufenden Anlage bewiesen werden.

Ziel dieser Arbeit ist es eine grafische Oberfläche in Visual BASIC 6 zur Regleroptimierung nach bekannten Einstellalgorithmen zu entwickeln, welche auf die von der GEA TDS definierten Datenbanken und Steuerungen zugreifen kann. Dabei soll der GEA TDS-Programmierstandard hinsichtlich der Programmierung und Benutzeroberfläche eingehalten werden.

Das erste Kapitel dieser Diplomarbeit beinhaltet eine Einführung in die Thematik und definiert das Ziel dieser Diplomarbeit. Kapitel zwei beschreibt die Grundlagen der benötigten Hard- und Software, welche für diese Diplomarbeit von Bedeutung sind. Im dritten Kapitel wird der Standard-PID-Regler beschrieben. Das vierte Kapitel erläutert die verwendeten Optimierungsverfahren. Die Software-Entwicklungsumgebung und Programmierung des Tools ist Inhalt des fünften Kapitels. Das sechste Kapitel beschreibt dann den praktischen Test an einer Klimaanlage in einer Molkerei in Uelzen. Die Zusammenfassung ist Teil des siebten Kapitels und der Ausblick ist Inhalt des achten und letzten Kapitels.

## 2 Grundlagen

### 2.1 Ausgangssituation

Bei der Firma GEA Tuchenhagen Dairy Systems GmbH wird das komplette Software-Engineering für die Prozesse von Molkereianlagen realisiert. Vorwiegend werden die Programme in Visual BASIC geschrieben, um einen bestimmten Standard für alle Ingenieure zu haben, mit dem jeder umgehen kann.

Es werden hier nur Software-Regler eingesetzt, die überwiegend in der S7-Steuerung implementiert sind. Sofern der Kunde eine andere Steuerung (z.B. von Rockwell) in seiner Anlage wünscht, werden auch diese von der Firma eingesetzt.

Die Access-Datenbank, die auf dem OPC-Server gespeichert ist, wurde für die Konfiguration der Regler programmiert. Hier werden alle Daten des Reglers im Voraus konfiguriert. Diese Daten werden dann über einen Download-Button in die Steuerung geschrieben.

Bisher werden die Softwareregler empirisch eingestellt. Eine Software zum Optimieren von PID-Reglern existiert zwar in der Firma, der Siemens PID-Selftuner, wird aber aufgrund der mangelnden Zeit der Ingenieure und der Komplexität seiner Beschreibung und Anwendung nicht eingesetzt.

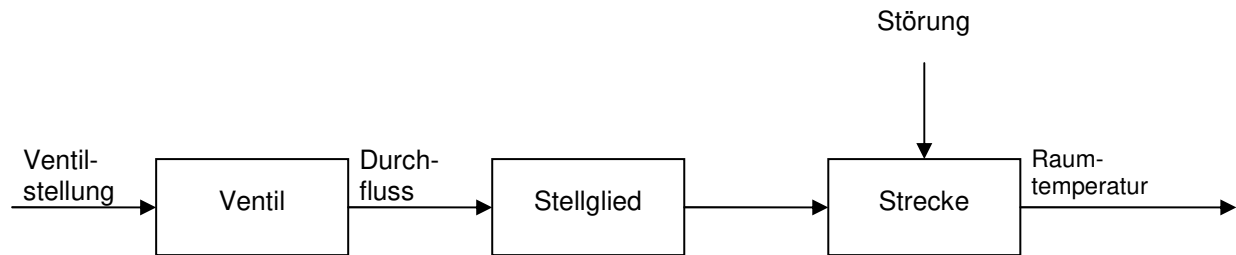
## 2.2 Abgrenzung der Steuerung zur Regelung

Der Begriff Regelung ist zu unterscheiden von dem im allgemeinen Sprachgebrauch oft synonym gebrauchten Begriff der Steuerung, bei welcher jedoch die fortlaufende Rückkopplung und deren Bearbeitung fehlt. So können z.B. bei einer SPS durchaus Rückkopplungen vorgesehen sein, bspw. mittels Sensorsignalen. Die hierdurch erfasste Größe wirkt jedoch über die Steuerung nicht fortlaufend auf sich selbst ein. Typischerweise stellen SPS die Abarbeitung von schrittweise ablaufenden Prozessen sicher oder realisieren eine Schaltalgebra. Es gibt bestimmte Bereiche, bei z.B. einer Positionsregelung von  $\pm 10$  V Servoantrieben, die nicht mit einer Steuerung realisierbar sind, sondern in jedem Fall eine Regelung benötigen. Merkmale und Eigenschaften von Steuerungen und Regelungen sind in der folgenden Tabelle zusammengefasst.

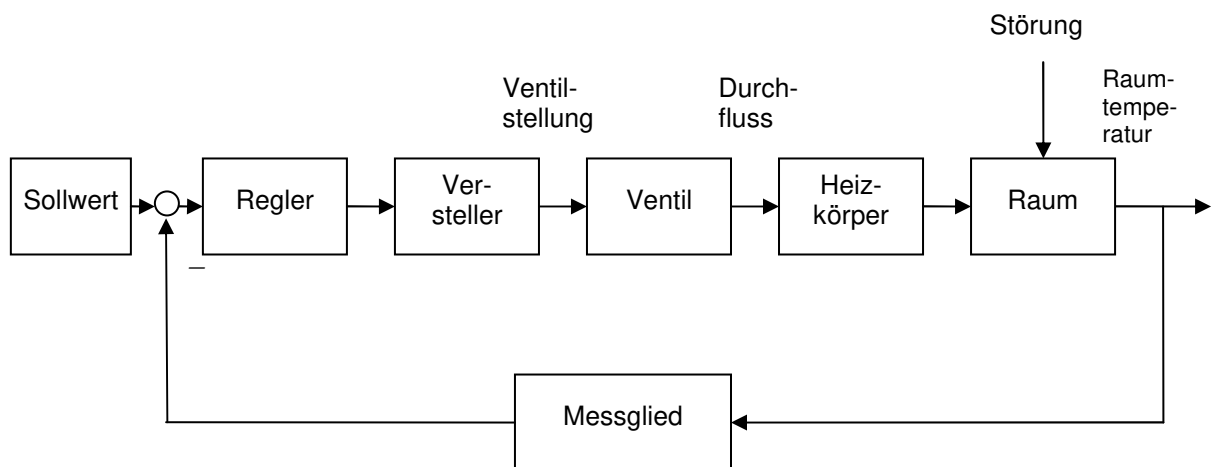
Kennzeichen	Regelung	Steuerung
<b>Messung und Vergleich der einzustellenden Größe</b>	Zu regelnde Größe wird gemessen und verglichen.	Zu steuernde Größe wird nicht gemessen und verglichen.
<b>Reaktion auf Störungen allgemein</b>	Wirkt allen Störgrößen entgegen, die an dem zu regelnden System angreifen.	Reagiert nur auf Störgrößen die gemessen und in der Steuerung verarbeitet werden.
<b>Reaktion auf Störungen zeitlich</b>	Reagiert erst dann, wenn die Differenz von Soll- und Istwert sich ändert.	Reagiert schnell, da die Störung direkt gemessen wird.
<b>Technischer Aufwand</b>	Geringer Aufwand: Messung der zu regelnden Größe, Soll- Istwert-Vergleich, Leistungsverstärkung.	Hoher Aufwand, wenn viele Störungen berücksichtigt werden müssen, geringer Aufwand, wenn keine Störungen auftreten.
<b>Verhalten bei instabilen Systemen</b>	Bei instabilen Systemen müssen Regelungen eingesetzt werden.	Steuerungen sind bei instabilen Systemen unbrauchbar.

**Tabelle 2.1:** Regelung und Steuerung im Vergleich [3]

Die beiden folgenden Beispiele sollen anhand einer Heizung verdeutlichen, wie sich die Steuerung von der Regelung unterscheidet.



**Abbildung 2.1:** Steuerung [12]



**Abbildung 2.2:** Regelung [12]

## 2.3 Grundlegendes zur Regelungstechnik

Zum besseren Verständnis werden in diesem Kapitel vorab einige Begriffe der Regelungs- und Steuerungstechnik erklärt, welche ein Bestandteil dieser Diplomarbeit sind. Allerdings werden hier nicht alle Grundlagen der Regelungstechnik erläutert, sondern nur diejenigen, die für diese Diplomarbeit wichtig sind. Da in der Firma ausschließlich digitale PID-Regler mit oder ohne D-Anteil benutzt werden und nur Regelstrecken 1.Ordnung bei den Anlagen vorkommen, befasst sich diese Diplomarbeit auch nur mit diesen Themen. Allerdings erfolgt in den meisten Fällen keine Nutzung des D-Anteils der Regler.

### **Regelung allgemein:**

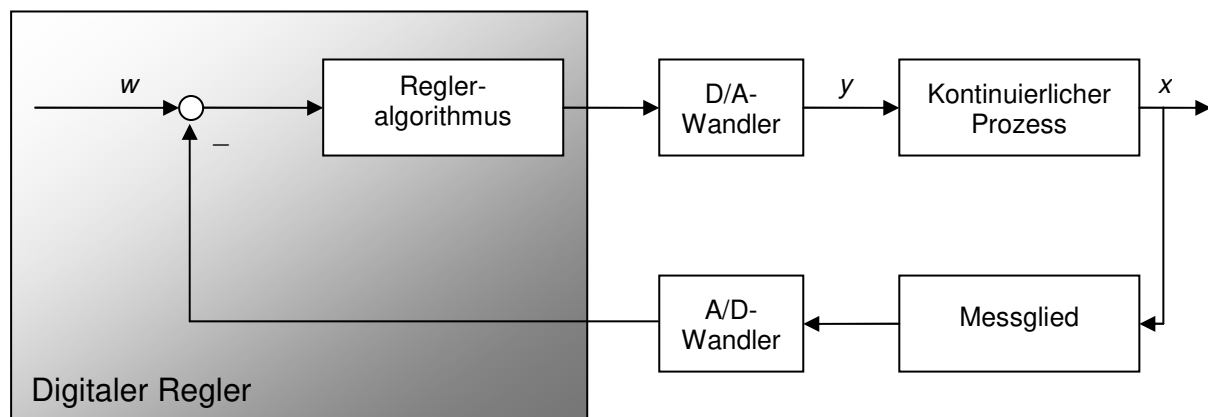
Die allgemeine Aufgabe der Regelung ist, eine bestimmte Zustandsgröße oder auch mehrere Zustandsgrößen auf einen vorgegebenen Sollwert zu überwachen und Abweichungen oder Störungen selbständig zu beseitigen.

Als Beispiel wird hier ein Fliehkraftregler näher erläutert. Der Fliehkraftregler ist ein Maschinenelement zur Regelung der Drehzahl einer Maschine und wurde von James Watt in den allgemeinen Maschinenbau eingeführt. James Watt benutzte den Fliehkraftregler, um die Arbeitsgeschwindigkeit der von ihm verbesserten Dampfmaschine konstant zu halten. Zuvor war dieses Maschinenelement bei den damaligen Mühlen für den gleichen Zweck verwendet worden.

Durch die Drehung des Fliehkraftreglers in der folgenden Abbildung werden die beiden kugelförmigen Gewichte aus Gusseisen durch die Fliehkraft immer weiter gegen die Schwerkraft nach außen gezogen. Über einen Gelenk- und Hebelmechanismus wird in der Dampfleitung der Maschine eine Drosselklappe betätigt, welche die weitere Dampfzufuhr zur Maschine verkleinert. Die Maschine läuft daraufhin langsamer, so dass sich die Drosselklappe wieder öffnet. Diese Anordnung ist ein Musterbeispiel für einen Regelkreis mit negativer Rückkopplung. Fliehkraftregler gibt es in vielerlei verschiedenen Arten, Bauformen und Ausführungen, aber das Prinzip ist immer gleich.



Die Verknüpfung des digitalen Reglers mit einem kontinuierlichen Prozess erfordert die Berücksichtigung der Signalwandlung am Eingang (A/D) und Ausgang (D/A) des Reglers. Während das Stellglied meist ebenfalls kontinuierlich arbeitet (Ausnahme z.B. Schrittmotoren) kann die Diskretisierung der Messsignale entweder im Sensor selbst oder in einem nachgeschalteten Wandler erfolgen. Der digitale Regelkreis hat die in der folgenden Abbildung gezeigte Grobstruktur.



**Abbildung 2.4:** Abtastregelkreis

$w$  = Führungsgröße = Sollwert

$y$  = Reglerausgangsgröße = Stellgröße

$x$  = Regelgröße = Istwert

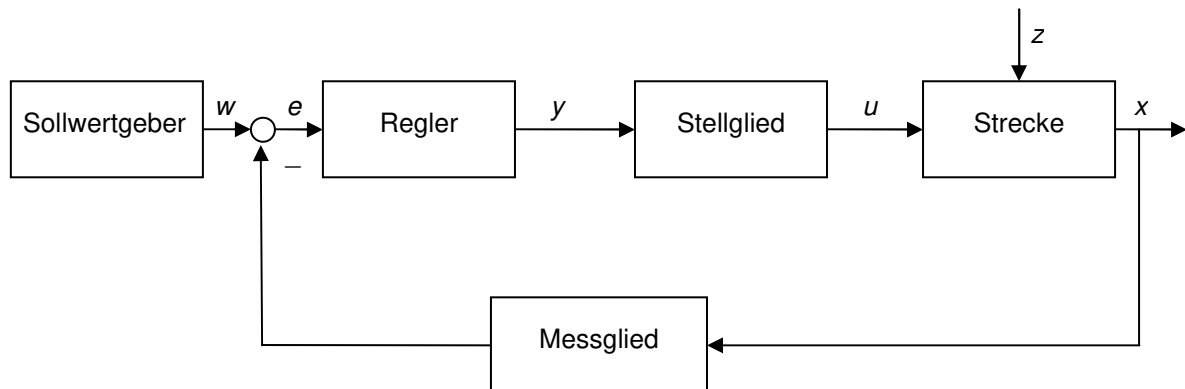
Für die Berechnung des Regelkreises muss der kontinuierliche Prozess einschließlich Stellglied ebenfalls diskretisiert werden und der gesamte Regelkreis ist dann als Abtastsystem zu behandeln. Um die Grundbegriffe der digitalen Regelungstechnik kennen zu lernen, wird im Folgenden zunächst die Signalwandlung in einem Regelkreis mit diskretem Regler eingehender behandelt. Es wird darauf hingewiesen, dass nur die Signalumformung betrachtet wird.

Wenn der digitale Regler circa zehn mal schneller ist als die Regelstrecke, kann das System als ein kontinuierliches System betrachten werden. [Vorlesung Werkzeugmaschinen 1]



**Der Regelkreis:**

Ein Regler kann nur in einem geschlossenen Regelkreis seine Aufgabe erfüllen. Der Messwertaufnehmer erfasst die Prozesstemperatur und gibt sie auf den Eingang des Reglers. Der Regler vergleicht diesen Istwert mit dem eingestellten Sollwert und berechnet mit Hilfe der eingestellten PID-Parameter die Stellgröße. Diese Stellgröße beeinflusst über das Stellglied die Leistungszuführung zum Prozess. Die Prozessreaktion wird durch den Messwertaufnehmer erfasst, womit der Regelkreis geschlossen ist. In der folgenden Abbildung ist der Aufbau eines Regelkreises deutlich gemacht.



**Abbildung 2.5:** Elemente eines Regelkreises

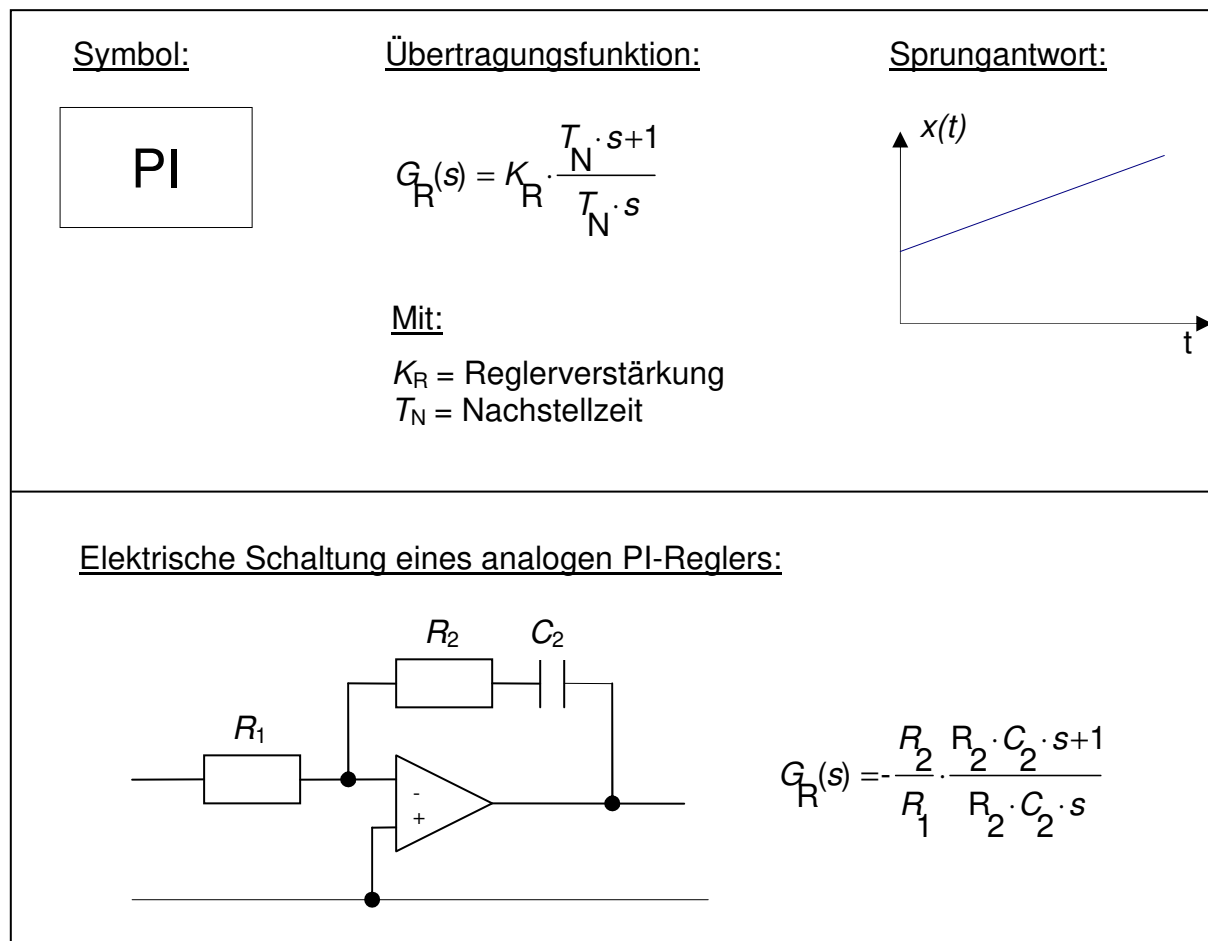
- $w$  = Führungsgröße = Sollwert
- $e$  = Regeldifferenz = Regelabweichung
- $y$  = Reglerausgangsgröße = Stellgröße
- $u$  = Ausgangsvariable
- $z$  = Störgröße
- $x$  = Regelgröße = Istwert

**Der Regler allgemein:**

Ein Regler ist ein Gerät, welches ein Eingangssignal (Istwert) mit einer vorgegebenen Einstellung (Sollwert) vergleicht und nach Berechnung (PID-Algorithmus) den Regelausgang (Stellgröße) einstellt. Die Wahl eines bestimmten Reglertyps richtet sich nach dem geforderten Zeitverhalten und der geforderten Regelgenauigkeit der Regelstrecke.

**Der PI-Regler:**

Der PI-Regler ist die Kombination aus P- und I-Regler und vereint den Vorteil des P-Reglers, nämlich schnelle Reaktion, mit dem Vorteil des I-Reglers, der exakten Ausregelung. Der PI-geregelte Kreis ist somit genau und mittelschnell. Der Proportional- und Integralanteil wird im folgenden Abschnitt des PID-Reglers erläutert.



**Abbildung 2.6:** PI-Regler [Skript Regelungstechnik]

**Der PID-Regler:**

Bei Industrieregeln und Prozessreglern kommt fast ausschließlich der PID-Algorithmus zu tragen. Selten werden Zweipunktalgorithmen eingesetzt. Der PID-Regler wirkt in erster Linie durch seine drei Funktionsanteile mit nachfolgenden Bezeichnungen:

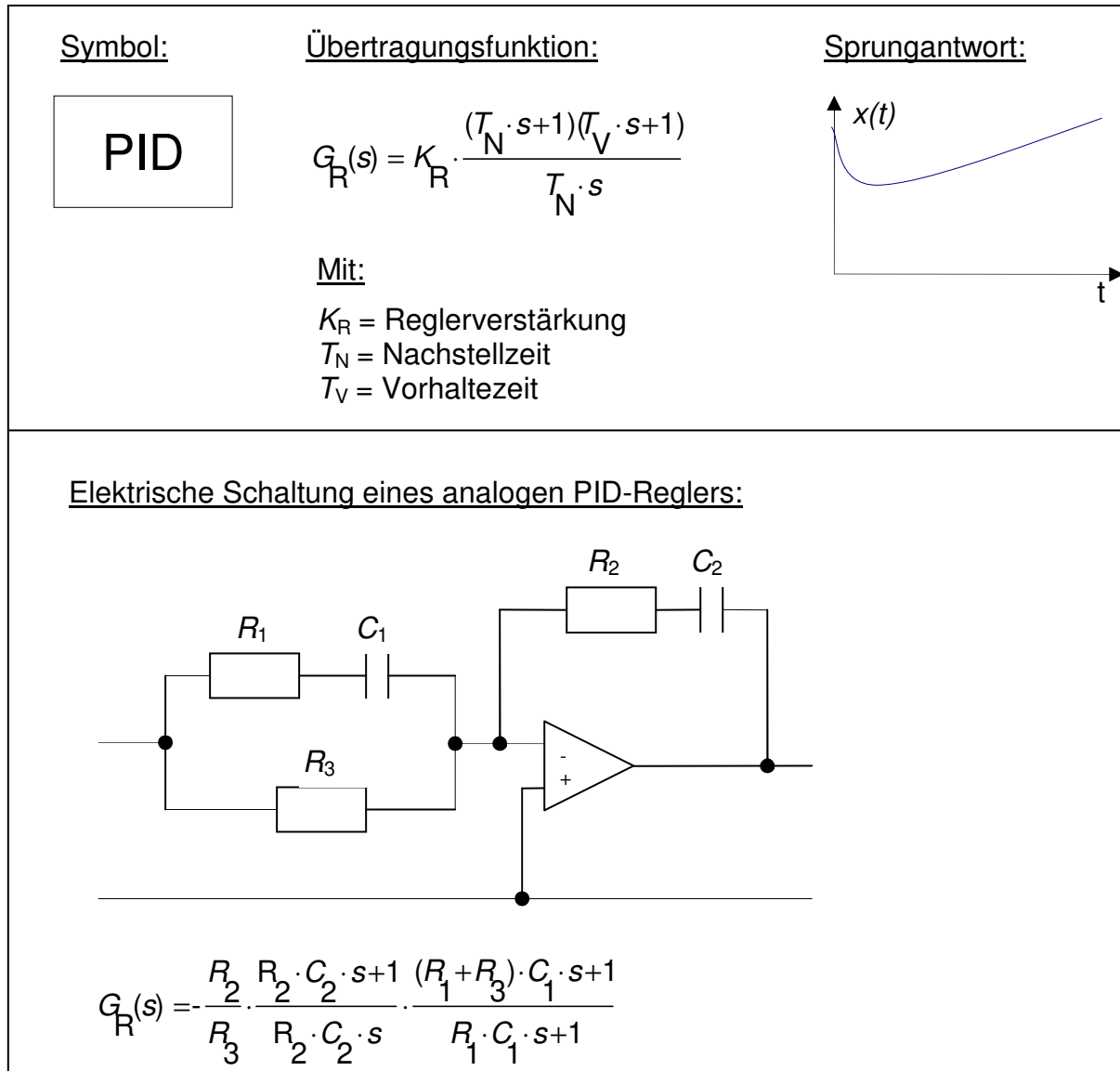
- **P** für Proportionalanteil
- **I** für Integralanteil
- **D** für Differentialanteil

**P** – Proportionalanteil ( $K_R$ ): Der Proportionalanteil liefert einem Regler Ausgangsänderungen (Stellgröße), welche direkt proportional zu der Größe der Regelabweichung sind, so dass unmittelbar der Störung oder Änderung entgegen gewirkt wird. Innerhalb des Proportionalbandes wird die Ausgangs-Stellgröße vom minimalen Wert bis zum maximalen Wert verstellt.

**I** – Integralanteil ( $T_i$ ): Der Integralanteil, auch Nachstellzeit genannt, gleicht über eine lineare Steigungsfunktion die Ausgangsstellgröße in Richtung weniger oder mehr so an, dass die vorhandene Regelabweichung restlos ausgeglichen wird. Die Steigung der Integralfunktion ist von der Amplitude und der Zeitdauer des Regelabweichungssignals abhängig. Die Nachstellzeit ist die Zeit, die der Integrator benötigt, um die gleich große Stellgröße aufzubauen, wie diese bei gleicher Regelabweichung durch den Proportional sprung erzeugt würde. Ein Nachteil des Integralanteils besteht darin, dass die Regelabweichung sozusagen aufsummiert und im Extremfall auch bei Erreichen des Sollwertes eine Stellgröße generiert wird.

**D** – Differentialanteil ( $T_D$ ): Der Differentialanteil, auch Vorhaltezeit genannt, bewirkt eine zusätzliche Stellgrößenänderung proportional zu der Änderungsgeschwindigkeit des Regelabweichungssignals. Damit wirkt der D-Anteil durch seine voreilende Wirkung schnell auf den Stellausgang und durch seine hohe Dynamik sorgt dieser für eine schnellere Ausregelung. Über- bzw. Unterschwingungen, die durch plötzliche große Regelabweichungen und Störgrößen erzeugt würden, werden dadurch minimiert.

Der Reglerstellgrößen Ausgang ( $y$ ) ergibt sich aus der Summe der drei Anteile  $K_R + T_i + T_D = y$ . Die Berechnung der jeweiligen Stellgröße ist dabei eine Funktion aus der Größe und dem zeitlichen Verlauf der Regelabweichung sowie dem Verlauf der Prozessgröße (Temperatur).



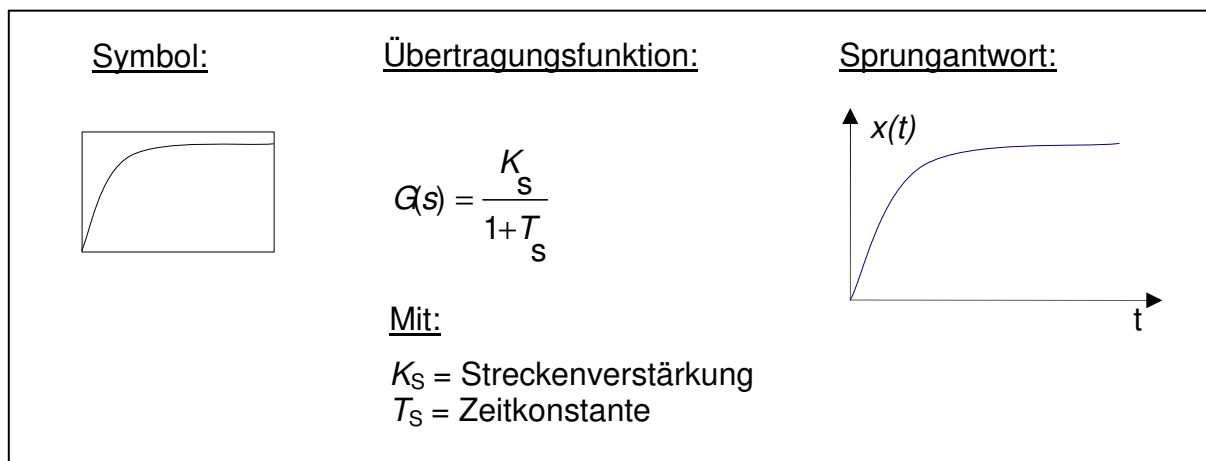
**Abbildung 2.7:** PID-Regler [Skript Regelungstechnik]

**Die Regelstrecke:**

Die Regelstrecke stellt den zu regelnden Teil bzw. den zu regelnden Prozess dar und umfasst normalerweise eine Reihe von einzelnen Gliedern. Die Glieder werden entsprechend ihrem Zeitverhalten charakterisiert. Um das Zeitverhalten herauszufinden, legt man an den Eingang ein Testsignal an und zeichnet die Antwort auf. Im einfachsten Fall wird der Eingang mit einer sprunghaften Änderung beaufschlagt. Die Antwort auf die sprunghafte Änderung der Eingangsgröße wird Sprungantwort genannt und gibt Aufschluss über die Art und das Verhalten der Regelstrecke an und kann eventuell bereits genutzt werden, um die Parameter der Regelstrecke zu bestimmen.

Für den Aufbau eines gut funktionierenden Regelkreises und die Auslegung von Reglern ist es Voraussetzung zu wissen, wie die Regelstrecke reagiert. Ohne ein genaues Wissen um das dynamische Verhalten der Regelstrecke ist es nicht möglich geeignete Regler auszuwählen und diese zu parametrieren.

Das wichtigste Glied dieser Diplomarbeit ist das PT1-Glied, welches ein proportionales Übertragungsverhalten mit Verzögerung 1. Ordnung aufweist.



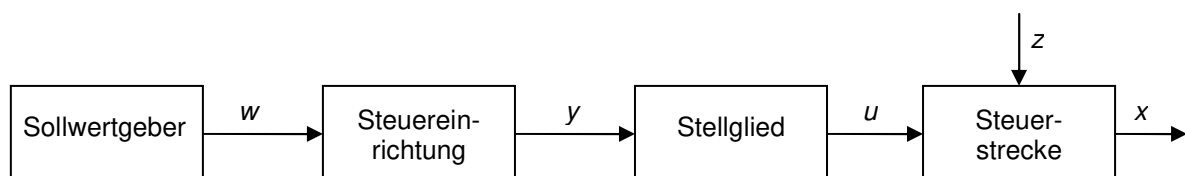
**Abbildung 2.8:** Verzögerungsglied 1. Ordnung (PT1-Glied)

## 2.4 Grundlegendes zur Steuerungstechnik

Der Begriff Steuerung (engl.: *control*) bezeichnet im allgemeinen den Vorgang einer Veränderung, welche nicht zufällig, sondern kontrolliert verläuft. Dahinter steht die Annahme einer willkürlichen Kraft, die das zu steuernde Objekt meist direkt bewegt oder kontrolliert.

Eine Steuerung zeichnet sich dadurch aus, dass eine oder mehrere Eingangsgrößen eines Systems, dessen Ausgangsgrößen, entsprechend der im System herrschenden Gesetzmäßigkeiten, beeinflussen. Nur wenn alle Einflussgrößen eines Prozesses sowie ihre Wirkung auf die Ausgangsgröße hinreichend bekannt und in der Steuerungseinrichtung als Kennlinie hinterlegt sind, erhält man bei einer Steuerung den gewünschten Verlauf der Ausgangsgröße.

Bei der Steuerung findet kein Vergleich der Ausgangsgröße mit der Eingangsgröße statt. Es liegt somit ein offener Wirkungsablauf vor, bei dem keine Rückkopplung vorhanden ist. Das Prinzipschema in Abbildung 2.9 zeigt die charakteristische Kette einer Steuerung.



**Abbildung 2.9:** Elemente einer Steuerkette

$w$  = Sollwert

$y$  = Stellgröße

$u$  = Ausgangsvariable

$z$  = Störgröße

$x$  = Ausgangsgröße = Istwert

### **Die Speicherprogrammierbare Steuerung:**

Eine Speicherprogrammierbare Steuerung (SPS, engl.: PLC = programmable logic controller) ist eine einem Computer ähnliche elektronische Baugruppe, die für Steuerungs- und Regelungsaufgaben in der Automatisierungstechnik eingesetzt wird.

Die SPS besitzt eine feste interne Verdrahtung, unabhängig von der jeweiligen Steuerungsaufgabe. Die Anpassung an die zu steuernde Maschine oder Anlage erfolgt durch ein Programm, welches den gewünschten Ablauf festlegt. Im Gegensatz hierzu gibt es verbindungsprogrammierte Steuerungen, die jedoch bei jeder Änderung im Steuerungsablauf Hardwareseitig umgebaut werden müssen.

Viele moderne Steuerungen sind allerdings Computersysteme (Microcontroller, PC-basiert) mit einer Basissoftware. Die Basissoftware besteht aus einem Echtzeitbetriebssystem und SPS-spezifischen "Bausteinen", die SPS-Funktionen wie Addierer und sonstiger Verknüpfungsglieder nachbilden. Üblicherweise gehört auch noch Software zur Kommunikation mit einem Programmiergerät dazu.

Wie jeder andere Computer arbeitet die SPS nach dem EVA-Prinzip, sie besitzt also einen **E**ingabe-, einen **V**erarbeitungs- und einen **A**usgabeteil. Die E/A-Geräte (die an die Eingänge/Ausgänge, engl.: Input/Output, angeschlossenen Geräte) sind mit der SPS verdrahtet.

Im Unterschied zu herkömmlichen Computern arbeitet die SPS zyklisch. Sie liest die Werte aller Eingänge am Beginn eines Zyklus ein, führt daraufhin die gespeicherten Programme (auch als Bausteine oder Netzwerke bezeichnet) aus und setzt am Ende die Ausgänge. Im Anschluss startet der Zyklus erneut; ein Programmende gibt es nicht.

Zustandsänderungen, die an den Eingängen angeschlossenen Sensoren, die sich während des Zyklusdurchlaufs ereignen, werden nicht erkannt. Abhängig von deren Werten werden die an ihren Ausgängen angeschlossenen Aktoren gemäß eingebautem Programm angesteuert. Dies geschieht einmal am Ende des Zyklus.

Eingänge können digitale oder analoge Signale sein, aber auch komplexere Vorgänge wie z.B. eine serielle Schnittstelle. Beispiele für Sensoren sind: Drucksensoren, Temperaturfühler, Endschalter, Grenztaster, Drehzahlgeber etc..

Ausgänge steuern z.B. Motoren, Relais, Pneumatik-, Hydraulikzylinder, werden aber auch zum digitalen Signalaustausch mit anderen Steuerungen genutzt. Viele weitere Schnittstellen zu diversen technischen Einrichtungen existieren.

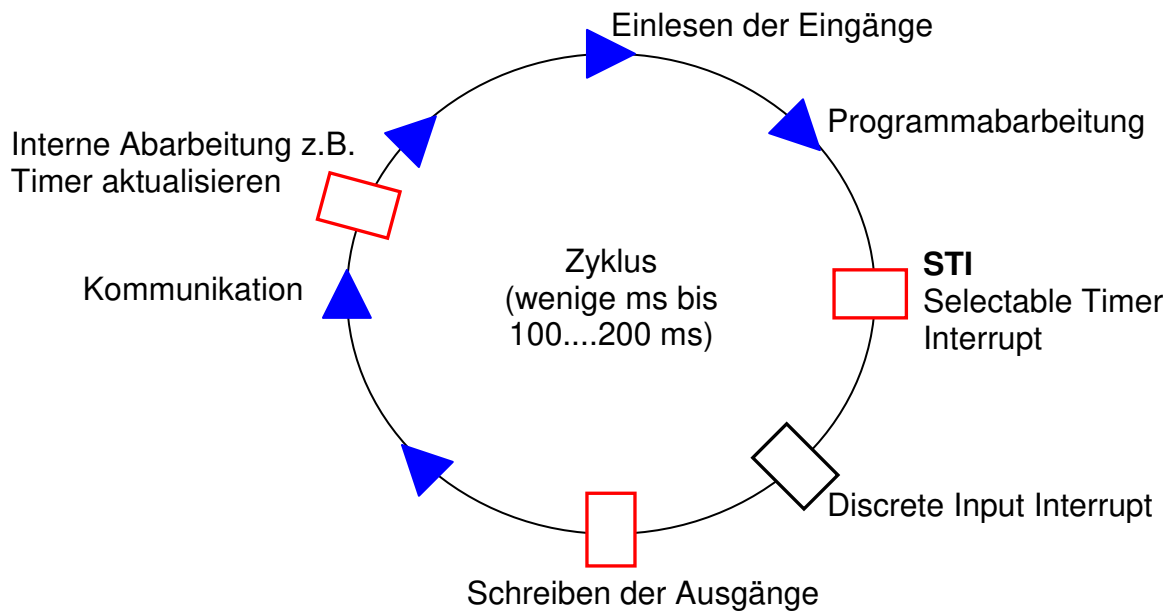
Die Steuerung verfügt mindestens über Und/Oder-Funktionen, Zähler, Merker und Timer. Weitere Möglichkeiten hängen von der verwendeten Sprache ab. Die Programmierung von Timern für die Steuerung zeitlicher Abläufe ist eine häufige Tätigkeit in der SPS-Programmierung.

Moderne Steuerungen wie z.B. die Rockwell Automation Control Logix-Familie, verfügen auch über Multitasking-Möglichkeiten. Die besondere Schwierigkeit hierbei ist jedoch, dass das Zeitverhalten reproduzierbar sein muss (Echtzeitverhalten). Das heißt, dasselbe Ereignis an den Eingängen muss zu einem vorhersagbaren Zeitpunkt zum entsprechenden Zustand an den Ausgängen führen. Dieses ist auch der Grund, warum sich ein normaler PC mit einem Standard-Betriebssystem nicht besonders gut für Steuerungsaufgaben eignet; das Betriebssystem reagiert auf verschiedenste Ereignisse (z.B. Mausbewegung, Tastatureingabe) und verzögert dadurch ein eventuell parallel ablaufendes Steuerungsprogramm. Es gibt inzwischen Ansätze, auch PC's für Steuerungsaufgaben tauglich zu machen. Die Idee hierbei ist, einem besonderen Programm (Kernel) die Zeitsteuerung zu übertragen. Dieses wiederum teilt dem Betriebssystem und dem SPS-Betriebssystem Rechenzeit zu. Es ist aber auch jederzeit in der Lage dem Betriebssystem die Kontrolle über den PC zu entziehen. Diese Aufgabe ist nicht ganz trivial. Da ein PC außerdem selten über die für Steuerungsaufgaben erforderlichen Schnittstellen verfügt, müssen Vor- und Nachteile des Einsatzes eines PC's für Steuerungsaufgaben gut überlegt werden.

Der Einsatz speicherprogrammierbarer Steuerungen (Begriff im industriellen Sinn verwendet) bedeutet nicht zwingend, dass regelungstechnisch gesehen nur gesteuert wird, sie können durchaus Regelungsfunktionen übernehmen, d.h. Teile von Rückkopplungskreisen sein.

Das preemptive Multitasking ist ein Mehrprozessbetrieb. Das bedeutet die quasi gleichzeitige (verzahnte) Abarbeitung von mehreren Programmen, von denen jedes in einem eigenen Adressraum liegt.



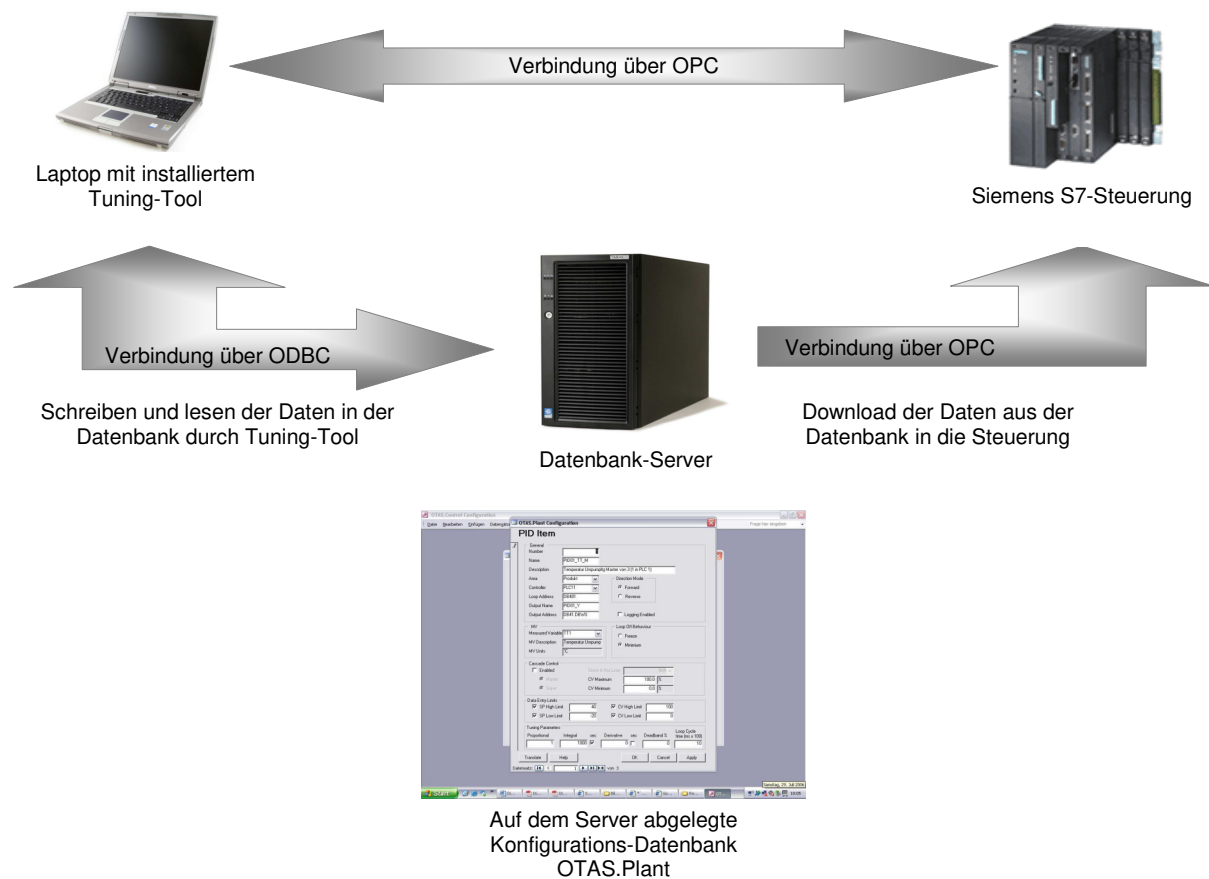


**Abbildung 2.10:** Abarbeitung in der SPS [Vorlesung Werkzeugmaschinen 1]

## 2.5 Programmrelevante Verbindungen

Die folgenden Verbindungsarten sind in diesem Teil des Kapitels näher beschrieben, da sie zum Lesen und Schreiben von Daten benötigt werden. Es erfolgt die Unterscheidung von zwei Arten der Verbindungen; die OPC-Verbindung zum Zugriff auf Daten in der Steuerung und die ODBC-Verbindung zum Zugriff auf Daten in der Access-Datenbank.

Die folgende Abbildung verdeutlicht die Anwendung der Verbindungsarten.



**Abbildung 2.11:** Weg des Datentransfers

### 2.5.1 OPC-Verbindung

OPC steht für **O**penness, **P**roductivity, **C**ollaboration (vormals: OLE for Process Control). Es ist eine standardisierte Software-Schnittstelle, die es Anwendungen unterschiedlichster Hersteller ermöglicht, Daten auszutauschen.

Mit dem Ziel einer Standardisierung und Kostenminimierung wurde im Jahr 1995 von verschiedenen Firmen unter Beteiligung der Siemens AG die OPC Task Force gegründet. Das Ziel ihres Zusammenschlusses war es, einen Standard für den Zugriff aus Echtzeitdaten unter Windows Betriebssystemen zu erarbeiten. Dieser Standard sollte auf OLE/DCOM Technologie von Microsoft basieren. So entstand OLE for Process Control, kurz OPC. Damit ist Plug&Play in der Automatisierungstechnik Wirklichkeit geworden.

In dieser Diplomarbeit wird die Verbindung für den Austausch von Daten zwischen Steuerung und Software-Tool und zum Download von Daten aus der Konfigurations-Datenbank zur Steuerung genutzt.

### 2.5.2 ODBC

ODBC (engl.: **O**pen **D**atabase **C**onnectivity) ist die Bezeichnung einer standardisierten Datenbankanschnittstelle der Datenbankabfragesprache SQL (**S**tructured **Q**uery **L**anguage). Sie wurde im Hause Microsoft entwickelt und basiert zum großen Teil auf dem Call Level Interface (CLI), das von der SQL Access Group entwickelt wurde. ODBC wurde erstmals im September 1992 der Öffentlichkeit für Windows-Anwendungen vorgestellt.

ODBC stellt eine objektorientierte C++ Klassenbibliothek zur Verfügung. Es lassen sich aber auch andere Programmiersprachen wie z.B. Fortran6 oder Visual BASIC benutzen. Durch die Benutzung dieses Basic-Dialektes ist auch der Zugriff auf Datenbanken von üblichen Office-Programmen wie Textverarbeitung oder Tabellenkalkulation möglich.

Der Vorteil von ODBC ist, dass die Anwendungssoftware eine einzige Datenbankschnittstelle unterstützen muss, nämlich ODBC, egal welche physische Datenbank hinter dem ODBC-Treiber existiert.

Eine ODBC Verbindung kann von mehreren Anwendungen gleichzeitig genutzt werden. Man braucht nur einmal die Verbindungsdaten eingeben, und zwar bei der

Konfiguration des DSN (Data Source Name). Alle Anwendungen greifen dann über den DSN-Namen auf die Datenbank zu.

Wenn jetzt der Ort der Datenbankdatei geändert wird, muss nicht in jedem Programm die Änderung vorgenommen werden, sondern nur in der DSN-Konfiguration.

Ein Nachteil von ODBC-Verbindungen ist jedoch die schlechtere Performance gegenüber einem direkten Datenbankzugriff.

Auf Grund der überwiegenden Vorteile ist die ODBC-Datenbankverbindung schon längere Zeit bei GEA TDS im Einsatz und wird auch in diesem Projekt verwendet werden.

## **3 Standard PID Control V5.0.1 [4]**

### **3.1 Das Konzept des Standard PID-Reglers**

Das Softwareprodukt "Standard PID Control" besteht im wesentlichen aus zwei Funktionsbausteinen (FB's), in denen die Algorithmen zur Bildung der Regel- und Signalverarbeitungs-Funktionen für kontinuierliche bzw. für Schrittreger enthalten sind. Es handelt sich also um eine reine Softwareregung, bei der ein Standard-Funktionsbaustein die komplette Funktionalität des Reglers verkörpert.

Das Verhalten des Reglers selbst und die Eigenschaften der Funktionen im Mess- und Stellkanal werden durch die numerischen Algorithmen des Funktionsbausteins realisiert bzw. nachgebildet. Die für diese zyklischen Berechnungen benötigten Daten sind in regelkreisspezifischen Datenbausteinen hinterlegt. Zur Erzeugung mehrerer Regler wird ein FB nur einmal benötigt.

Jeder Regler wird durch einen Instanz-DB repräsentiert, welcher applikationsabhängig zu erstellen ist. Bei Nutzung des Werkzeugs "Standard PID Control Tool" erfolgt diese DB-Erstellung implizit. Das bedeutet, die Auslegung eines spezifischen Reglers beschränkt sich auf das Festlegen der Struktur- und Wertparameter in den Bearbeitungsfenstern der Bedienoberfläche. Der Instanz-DB wird von dem Konfigurationswerkzeug erstellt.

Die Berechnung der Algorithmen für einen bestimmten Regler erfolgt im Prozessor des S7-Automatisierungssystems, und zwar in den eingestellten Zeitintervallen (Abtastzeiten). Die Berechnungsergebnisse und damit die aktualisierten Werte von Ein- und Ausgangsvariablen (Mess- und Stellgrößen) sowie Zustandsmeldungen (Grenzwerte) werden im dazugehörigen Instanz-DB hinterlegt bzw. an die Prozessperipherie übergeben.

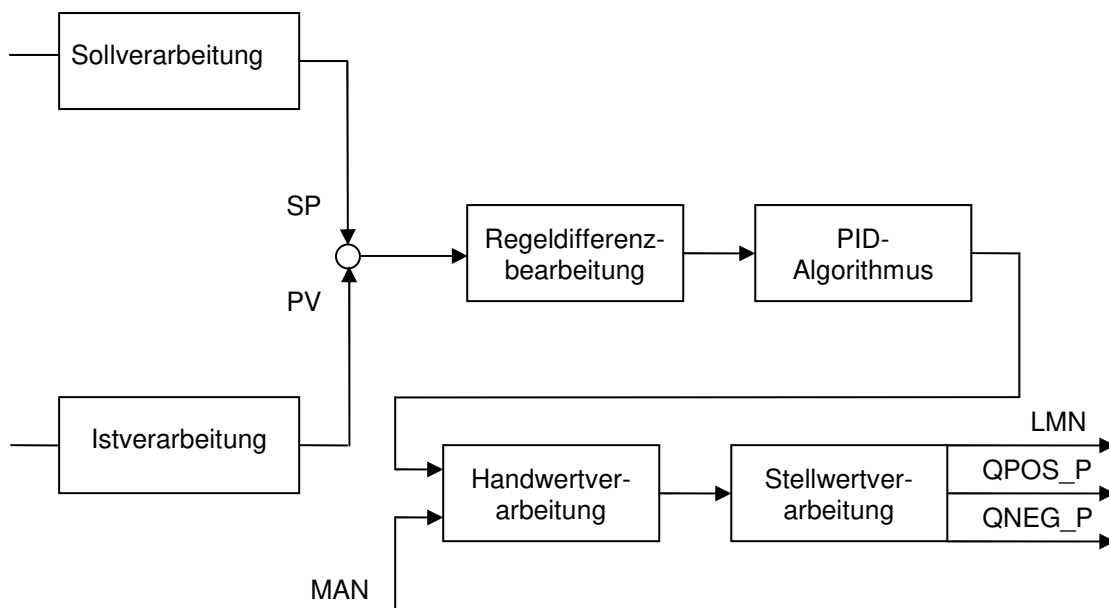
Bei Bearbeitung von vielen Regelkreisen, die je nach Trägheit der jeweiligen Regelstrecke unterschiedlich oft - dabei aber zeitlich äquidistant - bearbeitet werden sollen, steht ein Regleraufrufverteiler (Loop Scheduler = LP\_SCHED) zur Verfügung, mit dessen Hilfe die Konfiguration umfangreicher Anlagenregelungen sehr übersichtlich und deshalb einfacher wird. Außerdem ist so eine gleichmäßige Auslastung der CPU gewährleistet.

### 3.2 Übersicht über die Grundfunktionen des Standard PID-Reglers

Bei vielen Regelungsaufgaben steht nicht allein der klassische PID-Regler als prozessbeeinflussendes Element im Vordergrund, sondern es werden auch hohe Anforderungen an die Signalverarbeitung gestellt.

Ein mit Hilfe des Softwarepakets "Standard PID Control" gebildeter Regler setzt sich deshalb aus einer Reihe von Teilfunktionen zusammen, die von separat parametrierbar werden können. Zusätzlich zum eigentlichen Regler mit dem PID-Algorithmus sind auch Funktionen zur Aufbereitung des Soll- und Istwertes sowie zur Nachbearbeitung der errechneten Stellgröße integriert.

Hinzu kommen Anzeige- und Überwachungsfunktionen, welche im folgenden Übersichtsschema nicht dargestellt sind:



**Abbildung 3.1:** Funktionsübersicht des Softwarebausteins [4]

SP = Führungsgröße

PV = Regelgröße

LMN = Stellgröße

MAN = Handbetrieb

QPOS\_P = Impulsausgang (Positiv)

QNEG\_P = Impulsausgang (Negativ)

### 3.3 Erstellen der Regelung

Mit dem Softwarepaket "Standard PID Control" lässt sich ein Regler für eine bestimmte Regelungsaufgabe konfigurieren. Sein Funktionsvorrat kann eingeschränkt projiziert werden. Mittels so genannter Strukturschalter können Teilfunktionen zu- bzw. abgeschaltet oder ganze Zweige unwirksam gemacht werden. Zu parametrieren sind anschließend nur noch die in der reduzierten Struktur verbliebenen Funktionsteile.

Die Erstellung einer Regelung von der Strukturierung über die Parametrierung bis hin zum zeitgerechten Aufruf durch das Systemprogramm ist weitgehend ohne Programmierung möglich. STEP 7-Kenntnisse sind hierfür allerdings erforderlich.

Jedem Struktur- bzw. Wert-Parameter ist ein Datum, d.h. eine Zeile vorbehalten. Durch Editieren der entsprechenden Eintragungen lassen sich sowohl Struktur als auch die gewünschten Eigenschaften der Regelung festlegen.

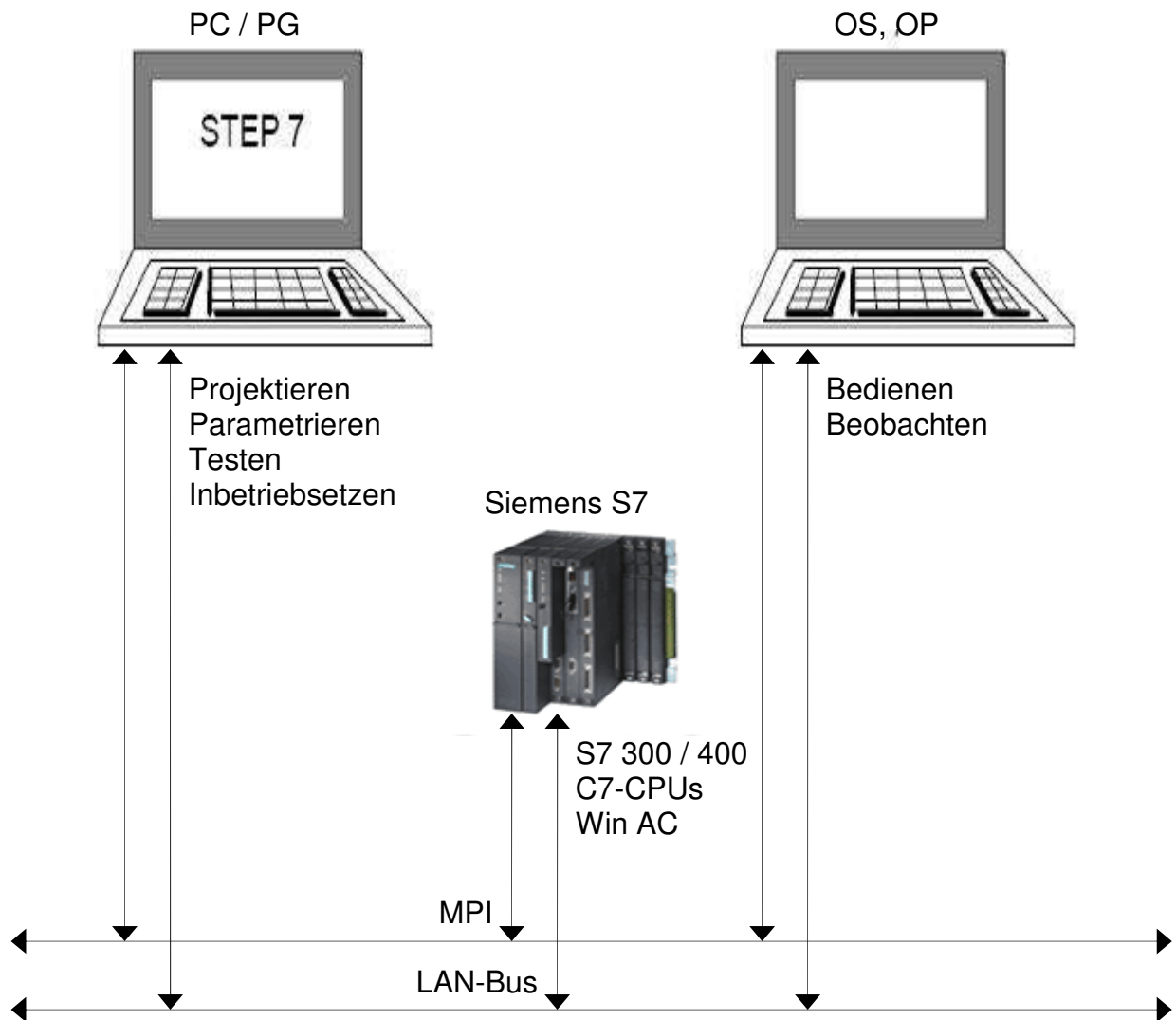
Diese Vorgehensweise wird jedoch aufgrund ihrer Unübersichtlichkeit nicht empfohlen. Das speziell für Standard PID Control konzipierte Konfigurationswerkzeug vereinfacht diese Aufgabe erheblich.

### 3.4 Einsatzumgebung und Anwendungsbereich

#### Hardwareumgebung:

Der "Standard PID Control" ist auf den folgenden Zielsystemen ablauffähig:

- S7-300- und S7-400 (CPU mit Gleitpunkt und Weckalarm)
- C7-CPU's
- Win AC



**Abbildung 3.2:** Einsatzumgebung des Softwarebausteins [4]

### Softwareumgebung:

Das Softwarepaket "Standard PID Control" ist für den Einsatz in der Programmgruppe STEP 7 konzipiert.

Die Erstellungssoftware für Standard-Regelungen kann lokal auf einem PC / PG oder aber in einem Netz auf einem zentralen Netzlaufwerk installiert werden.



**Der Systemrahmen:**

Da es sich bei der digitalen Realisierung von Reglerfunktionen immer um rechenaufwändige Operationen handelt (Wortverarbeitung), ist es wichtig, sich rechtzeitig ein Bild von der Belastung der zur Verfügung stehenden CPU zu machen. Hierfür gibt es folgende Anhaltspunkte:

- Codeumfang eines Funktionsbausteins (PID\_CP bzw. PID\_ES):  $\leq 8$  KByte
- Daten pro Regler  $\leq 0,5$  KByte
- Eckdaten für minimale Laufzeiten (Bearbeitungszeiten) eines PID-Reglers auf unterschiedlichen Automatisierungssystemen
- Speicherbedarf für einen L-Stack besteht nicht
- Alarme werden durch die Bearbeitung des Regelungs-FB nicht verzögert

**Regleraufrufverteiler:**

Müssen viele Regler oder Regler mit großen Abtastzeiten aufgerufen werden, reicht der Umfang des Prioritätsklassenmodells in Bezug auf Weckalarm-OBs nicht aus. Mit dem Regleraufrufverteiler LP\_SCHED (Loop Scheduler) können in einem Weckalarm-OB mehrere Regler mit verschiedenen Abtastzeiten äquidistant aufgerufen werden.

Die Aufgaben des Aufrufverteilers sind:

- Steuerung des zeitlichen Aufrufs der einzelnen Regler innerhalb einer (Weckalarm-) Prioritätsklasse.
- Aufruf der installierten Standard-Regler beim Anlauf der CPU.

**Einsatzmöglichkeiten und -grenzen von Standard PID Control:**

Eine durch die Abarbeitung eines Funktionsbausteins verkörperte Regelfunktion ist in Bezug auf ihre möglichen Einsatzfelder grundsätzlich neutral. Die Regelleistung und damit die Schnelligkeit der Bearbeitung von konkreten Regelkreisen hängt ausschließlich von der Leistung der verwendeten CPU ab.

Bei einer gegebenen CPU muss zwischen der Anzahl der Regler und der Häufigkeit, mit der die einzelnen Regler bearbeitet werden müssen, ein Kompromiss getroffen

werden. Je schneller die aufgeschalteten Regelkreise sind, d.h. je öfter die Stellgrößen pro Zeiteinheit errechnet werden müssen, umso geringer ist die Anzahl der installierbaren Regler.

Die Standard-Funktionsbausteine PID\_CP und PID\_ES ermöglichen es, Software-regler zu generieren und zu betreiben, welche auf dem konventionellen PID-Algorithmus von Standard PID Control basieren. Sonderfunktionen bezüglich der Verarbeitung der Prozesssignale im Regler sind nicht vorgesehen. Einschränkungen bezüglich der Art der regelbaren Prozesse bestehen nicht. Es können sowohl träge Strecken (z.B. Temperaturen, Füllstände usw.) als auch sehr schnelle Strecken (Durchflüsse, Drehzahlen usw.) geregelt werden.

Für den Einsatz von "Standard PID Control" existieren folgende Formen:

- Festwertregelung mit P-, PI-, PD-, PID-Schrittregler
- Festwertregelung mit kontinuierlichem P-, PI-, PD-, PID-Regler
- Festwertregelung mit Störgrößenaufschaltung
- Kaskadenregelung (Schrittregler nur im Folgekreis)
- Verhältnisregelung (zweischleifig)
- Mischungsregelung

### **Funktionsspektrum von Standard PID Control:**

Durch entsprechendes Projektieren der im Produkt "Standard PID Control" enthaltenen Funktionen lassen sich Regler mit nachstehenden Eigenschaften und Betriebsweisen generieren:

- Verstellung des Sollwertes durch Zeitplangeber (Zeitplanregelung)
- Begrenzung der Änderungsgeschwindigkeit der Führungsgröße und (bei Reglern mit kontinuierlichem Ausgang) der Stellgröße
- Begrenzung der Absolutwerte der Führungsgröße und (bei Reglern mit kontinuierlichem Ausgang) der Stellgröße
- Unterdrückung des Rauschens im Ist- oder Sollwertzweig durch Filterung des Regeldifferenzsignals
- Unterdrückung von höherfrequenten Schwingungen im Istwertzweig durch Glättung (Verzögerung) des Istwertsignals

- Linearisierung von quadratischen Funktionen der Regelgröße (Durchflussregelung mit Differenzdruckgebern)
- Möglichkeit des Aufrufs von "eigenen Funktionen" im Sollwert-, Istwert- und/oder Stellwert-Zweig
- Handbetrieb (Steuerung der Stellgröße von einem PG oder OP/OS aus)
- Überwachung der Regelgröße und/oder der Regeldifferenz auf Überschreitung von jeweils zwei oberen und zwei unteren Grenzwerten
- Überwachung der Änderungsgeschwindigkeit der Regelgröße
- Wahlweise P- und D-Wirkung in der Rückführung des Reglers

### 3.5 Die Abtastzeit CYCLE

Die Abtastzeit ist die Basis-Kenngröße für das dynamische Verhalten von Standard PID Control. Sie bestimmt, ob der betreffende Regler schnell genug auf Prozessänderungen reagiert und den Regler in allen Betriebszuständen beherrschen kann. Die zeitgebundenen Parameter von Standard PID Control werden jeweils durch den Wert der Abtastzeit nach unten begrenzt.

Die Wahl der Abtastzeit erfordert immer einen Kompromiss zwischen mehreren teils gegenläufigen Forderungen. Deshalb können hierfür nur Richtwerte angegeben werden.

- Die CPU-abhängige Laufzeit für die Bearbeitung des Regelprogramms, d.h. für die Abarbeitung des Funktionsbausteins stellt die unterste Grenze für die Abtastzeit CYCLE<sub>min</sub> dar.
- Die tolerierbare Obergrenze für die Abtastzeit wird im Allgemeinen durch die Prozeßdynamik festgelegt. Dabei ist die Prozeßdynamik durch den Typ und die Kennwerte der Regelstrecke charakterisiert.

## 3.6 Entwerfen von digitalen Regelungen

### Prozesseigenschaften und Regler:

Das statische Verhalten (Verstärkung) und die dynamischen Eigenschaften (Verzögerung, Totzeit, Integrationskonstante usw.) des zu regelnden Prozesses haben entscheidenden Einfluss auf die Art und das Zeitverhalten der Signalbearbeitung im Regler, welche die jeweiligen Prozesszustände stabil halten bzw. nach vorgegebenem zeitlichen Verlauf ändern soll.

Unter den Gliedern eines Regelkreises nehmen die Regelstrecken eine Sonderstellung ein. Ihre Eigenschaften sind durch verfahrens-/maschinentechnische Gegebenheiten festgelegt und kaum zu beeinflussen. Ein gutes Reglergebnis kann also nur durch Auswahl eines für den Streckentyp am besten geeigneten Reglertyps sowie dessen Anpassung an das Zeitverhalten der Strecke erreicht werden.

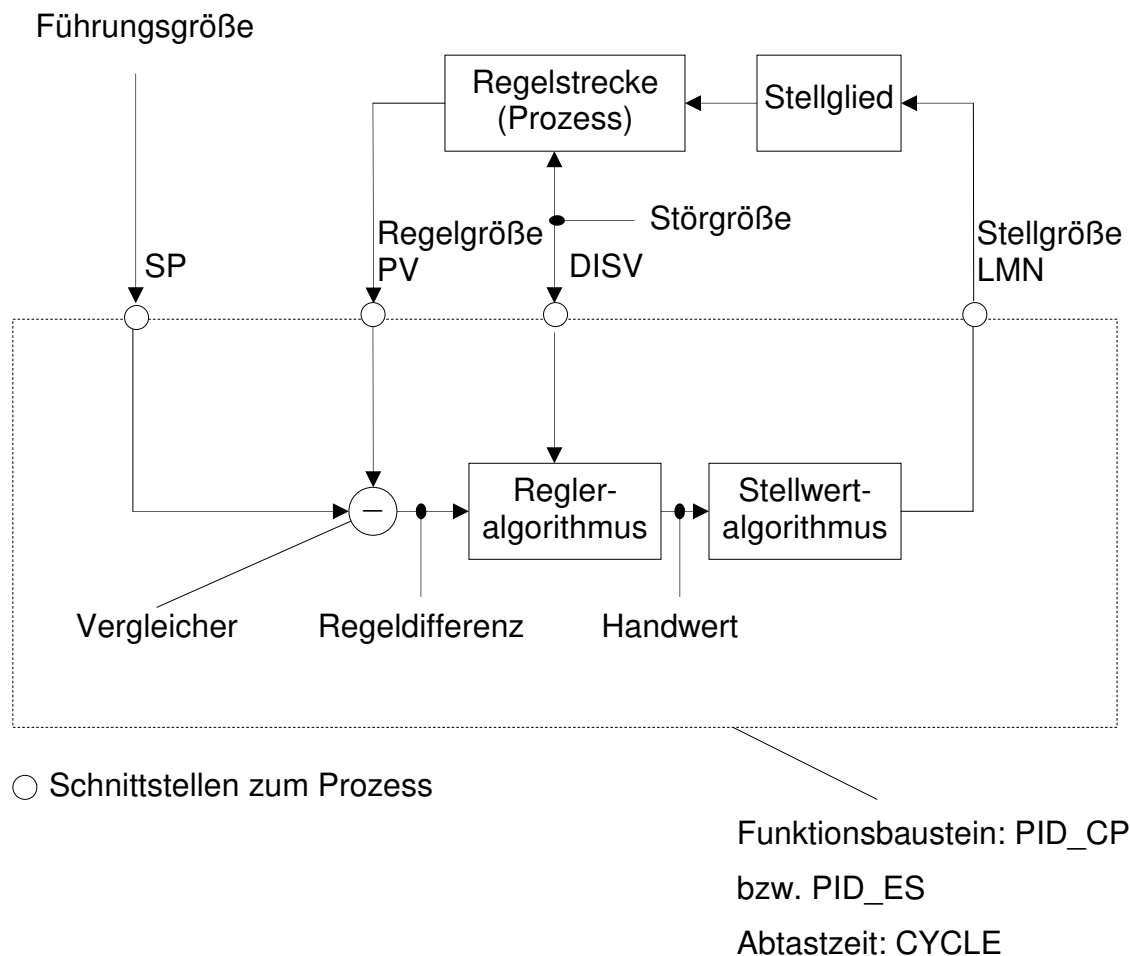
Genauere Kenntnisse des Typs und der Kenndaten der Regelstrecke sind deshalb für die Auslegung bzw. den Entwurf des Reglers und der Dimensionierung seiner statischen (P-Einfluss) und dynamischen Parameter (I- und D-Einfluss) unerlässlich.

## 3.7 Struktur und Arbeitsweise des Standard PID Control

### Die Abtastregelung:

Die mit "Standard PID Control" realisierbaren Regler sind grundsätzlich digitale Abtastregler (DDC-Regler, = Direct Digital Control). Abtastregler arbeiten zeitgesteuert, d.h. sie werden in immer gleichen Zeitabständen (der Abtastzeit CYCLE) bearbeitet. Die Abtastzeit bzw. die Häufigkeit, mit der ein bestimmter Regler bearbeitet wird, kann eingestellt werden.

Im folgenden Bild ist der einfache Regelkreis mit dem Standard-Regler dargestellt. Anhand des Bildes sollen die Bezeichnungen der wichtigsten Kreisgrößen sowie deren in diesem Handbuch verwendeten Parameterbezeichnungen (Abkürzungen) noch einmal in Erinnerung gerufen werden.



**Abbildung 3.4:** Abtastregler von Standard PID Control im Regelkreis [4]

Bei den Regelfunktionen, die in den Funktionsbausteinen PID\_CP und PID\_ES realisiert werden, handelt es sich um reine Software-Regelungen. Deren Ein- und Ausgangswerte werden mit Hilfe von digitalen Algorithmen auf einer CPU bearbeitet. Da die Bearbeitung der Reglerbausteine im Prozessor der CPU seriell erfolgt, können Eingangswerte nur zu diskreten Zeitpunkten erfasst und ebenso Ausgangswerte nur zu definierten Zeitpunkten ausgegeben werden. Dieses Verhalten ist das Charakteristikum der Abtastregelung.

**Regel-Algorithmus und konventionelle Regelung:**

Der Regel-Algorithmus im Prozessor bildet den Regler unter Echtzeitbedingungen nach. Zwischen den Abtastzeitpunkten reagiert die Regelung auf Änderungen der Prozessgröße PV nicht und die Stellgröße LMN bleibt unverändert.

Vorausgesetzt jedoch, die Abtastung geschieht in so kurzen Abständen, dass die Folge der Abtastwerte den kontinuierlichen Messgrößenverlauf annähernd getreu wiedergibt, lässt sich die digitale Regelung als eine quasikontinuierliche betrachten. Es können auch im Falle von "Standard PID Control" die üblichen Methoden zur Ermittlung von Struktur und Kennwerteinstellung wie bei kontinuierlichen Reglern angewendet werden.

Diese Voraussetzung für die Erstellung und Dimensionierung von Regelungen mit dem Regelungspaket "Standard PID Control" ist erfüllt, wenn die Abtastzeit kürzer ist als 20% der Gesamtstreckenzeitkonstante.

Wird diese Bedingung eingehalten, lassen sich die Funktionen von "Standard PID Control" wie die von konventionellen Reglern beschreiben. Der gleiche Funktionsvorrat steht mit den gleichen Möglichkeiten für die Überwachung von Regelkreisgrößen und für das Tuning des Reglers zur Verfügung.

**Die Funktionalität von "Standard PID Control":**

Die nachfolgende Abbildung zeigt die vorkonfektionierten Reglerstruktur von Standard PID Control als Blockschema. In der Abbildung 3.5 ist der kontinuierliche Regler mit den Signalverarbeitungszweigen für Ist- und Sollwert, dem Regler und dem Stellwertzweig dargestellt. Hier ist zu erkennen, welche Funktionen nach der Signalumformung am Eingang zu realisieren sind und welche nicht.

Das Funktionsinventar von "Standard PID Control" ist zwar starr, kann jedoch in allen Signalverarbeitungszweigen um jeweils eine eigene Funktion (FC) erweitert werden. Im Bedarfsfall kann der Anwender die FC einfügen, die Ergänzungsfunktionen enthält, d.h. vom Anwender geschriebenen FC's können in den Sollwert-, den Istwert- und / oder den Stellwertzweig eingefügt werden. Die folgenden Adressen müssen dann auf TRUE gesetzt werden.

- SPFC (SPFC\_ON = TRUE)
- PVFC (PVFC\_ON = TRUE)
- LMNFC (LMNFC\_ON = TRUE)

Der Regler führt einen Aufruf der Anwender-FC durch. Vorhandene Ein-/Ausgangsparameter der Anwender-FC werden dabei nicht versorgt.

Durch Einfügen einer anwenderspezifischen Funktion in den Sollwertzweig ist es möglich, eine extern vorgegebene Führungsgröße vor Aufschaltung auf den Regler einer Signalbehandlung (z.B. einer Signalverzögerung oder Linearisierung) zu unterziehen.

Durch Einfügen einer anwenderspezifischen Funktion in den Istwertzweig kann die Regelgröße vor der Weiterverarbeitung im Regler einer Signalbehandlung, z.B. einer Signalverzögerung oder Linearisierung, unterzogen werden.

Wenn eine anwenderspezifische Funktion in den Stellgrößenzweig eingefügt wird, kann die im Regler gebildete Stellgröße vor Aufschaltung auf den Ausgang des Reglers einer Signalbehandlung (z.B. einer Signalverzögerung) unterzogen werden.

Die Abzweigungen in der folgenden Abbildung die hier mit gestrichelten Linien dargestellt sind haben den Zweck, Blöcke wie z.B. die Anwenderfunktion oder Glättung zu übergehen.

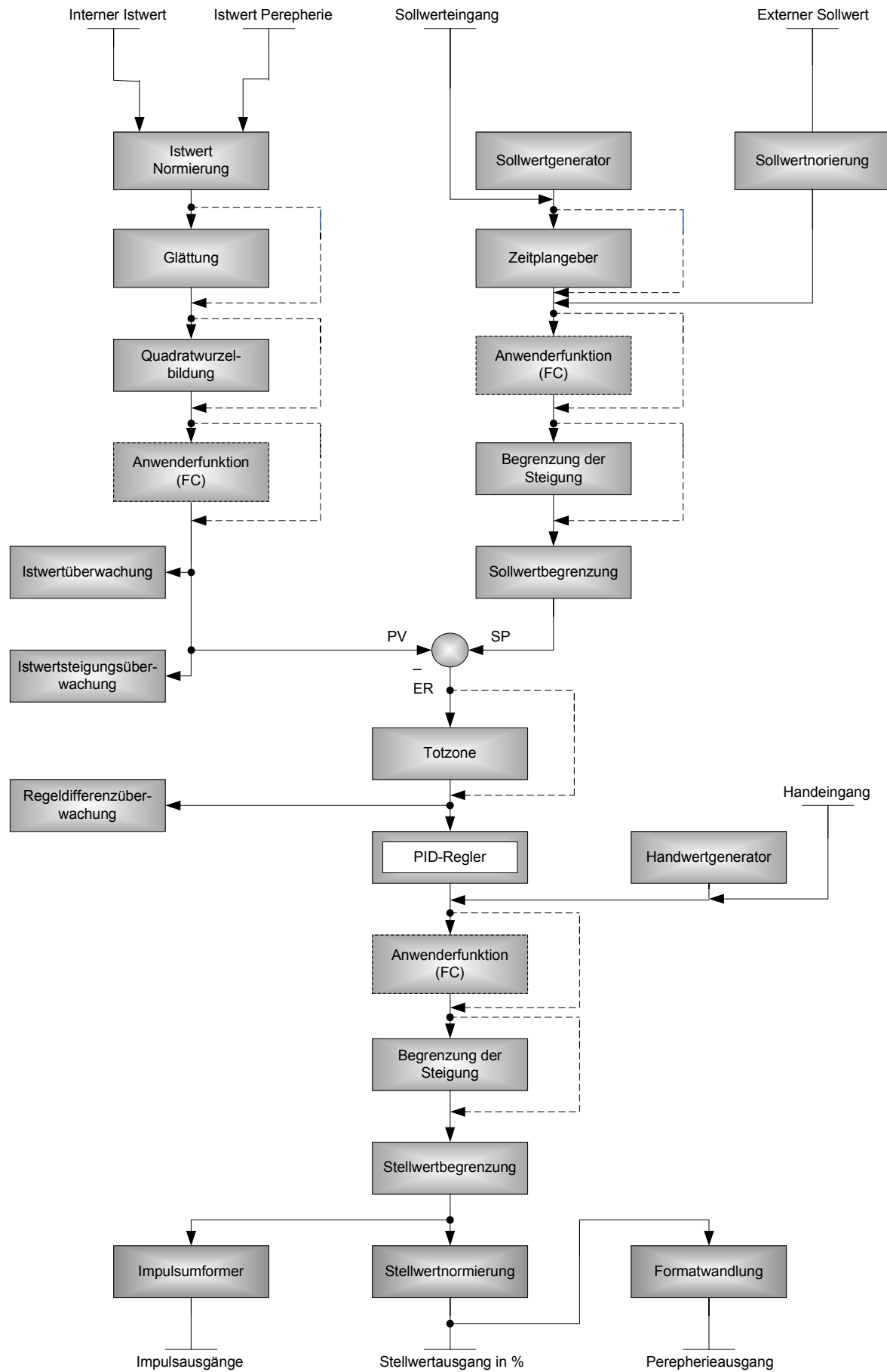


Abbildung 3.5: Funktionskette von Standard PID Control (kontinuierlicher Regler) [4]



## 4 Verfahren zur Regleroptimierung

Unter Reglereinstellung versteht man die Anpassung eines Reglers an die gegebene Regelstrecke. Wird die Güte des geregelten Gesamtsystems nach einem Gütekriterium, z.B. minimales Überschwingen oder minimaler quadratischer Regelfehler, durch Anpassung des Reglers eingestellt, so spricht man von Optimierung. Für das Optimieren von Reglern in der verfahrenstechnischen Praxis, haben sich empirische Einstellregeln bewährt. Hierfür müssen die Regelparameter ( $K_R$ ,  $T_N$ ,  $T_V$ ) so gewählt werden, dass bei den gegebenen Betriebsverhältnissen ein möglichst günstiges Verhalten des Regelkreises erzielt wird.

Der wichtigste Teil dieser Diplomarbeit beschäftigt sich mit den für diese Arbeit verwendeten Optimierungs- bzw. Einstellverfahren. Diese werden im Folgenden bezüglich ihrer Anwendung näher erläutert.

### 4.1 Das Tangentenverfahren

Vorraussetzung bei der Optimierung nach Ziegler und Nichols und Takahashi ist im Vorwege das Tangentenverfahren mit dem die Verzugszeit ( $T_U$ ) und die Ausgleichzeit ( $T_g$ ) bestimmt wird. Bei der PT1-Strecke wird allerdings die Totzeit ( $T_t$ ), die Zeitkonstante ( $T_S$ ) und der Übertragungsbeiwert ( $K_S$ ) bestimmt. Der Übertragungsbeiwert ist die Streckenverstärkung. Als Totzeit wird in der Regelungstechnik die Zeitspanne zwischen Änderung am Systemeingang und der Antwort am Systemausgang einer Regelstrecke bezeichnet. Ein System mit Totzeit wird auch Totzeitglied genannt.

In diesem Verfahren wird auf die Strecke ein Sprung gegeben und die Sprungantwort der Strecke aufgezeichnet. In der folgenden Abbildung ist dieses noch einmal verdeutlicht.

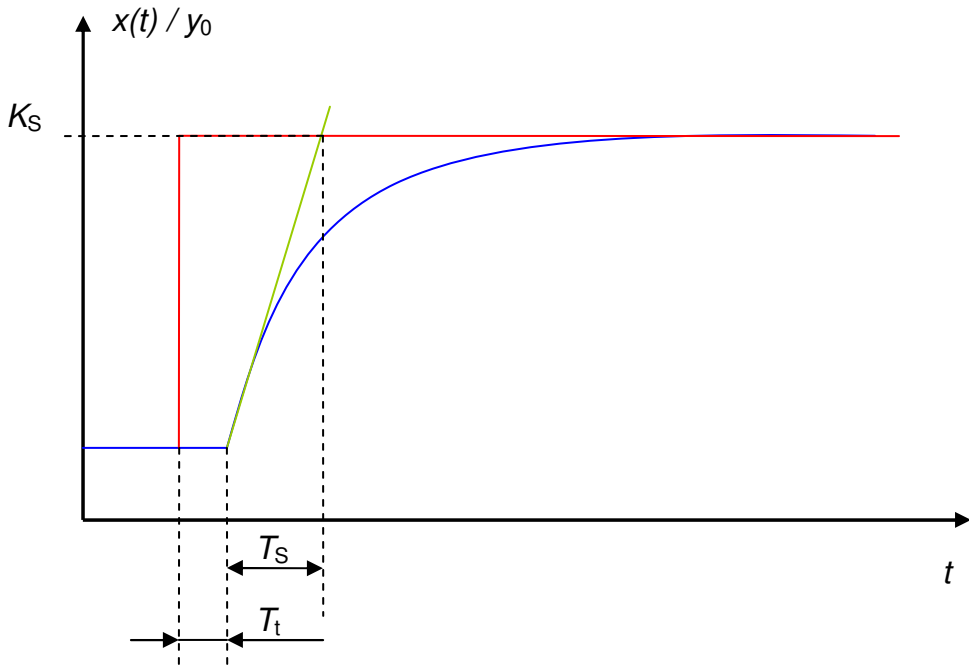


Abbildung 4.1: Verhalten des PT1-Gliedes mit Totzeit

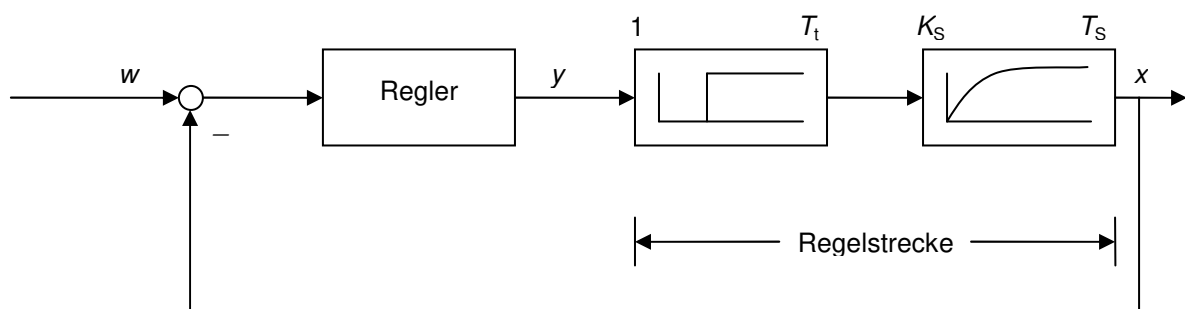
## 4.2 Optimierung nach Ziegler und Nichols [3]

Viele Regelstrecken der Verfahrenstechnik lassen sich durch ein Totzeitelement mit der Totzeit  $T_t$ , einem Verzögerungselement 1.Ordnung mit Streckenverstärkung  $K_S$  und einer Verzögerungszeitkonstante  $T_S$  angenähert darstellen.

Die Streckenübertragungsfunktion lautet hier:

$$G_S(s) = K_S \cdot \frac{e^{-T_t \cdot s}}{1 + T_S \cdot s}$$

Der Regelkreis lässt sich folgendermaßen beschreiben:



**Abbildung 4.2:** Regelkreis mit Totzeit und Verzögerung [3]

Transportvorgänge von Materie und Energie können näherungsweise durch Totzeitelemente modelliert werden. Das Verzögerungselement beschreibt näherungsweise das Verhalten von Energie- oder Materiespeicher. Sofern die Werte der Regelstrecke bekannt sind, ergeben sich die Einstellwerte gemäß folgender Tabelle:

Regler	$K_R$	$T_i = T_N$	$T_D = T_V$
P-Regler	$\frac{T_s}{K_s \cdot T_t}$	————	————
PI-Regler	$0,9 \cdot \frac{T_s}{K_s \cdot T_t}$	$3,33 \cdot T_t$	————
PID-Regler (additive Form)	$1,2 \cdot \frac{T_s}{K_s \cdot T_t}$	$2,0 \cdot T_t$	$0,5 \cdot T_t$

**Tabelle 4.1:** Optimierung nach Ziegler und Nichols mit bekannten Werten [3]

Liegen die Daten der Strecke nicht vor, so wird die optimale Reglereinstellung wie folgt bestimmt:

Die Regelstrecke wird zunächst mit einem Proportionalregler betrieben. Die Verstärkung  $K_R$  wird so lange erhöht, bis bei

$$K_R = K_{R\text{krit}}$$

der Regelkreis die Stabilitätsgrenze erreicht. Die Periodendauer  $T_{R\text{krit}}$  der entstehenden Schwingung wird gemessen. Für die verschiedenen Reglerarten wird dann die folgende Reglereinstellung verwendet:

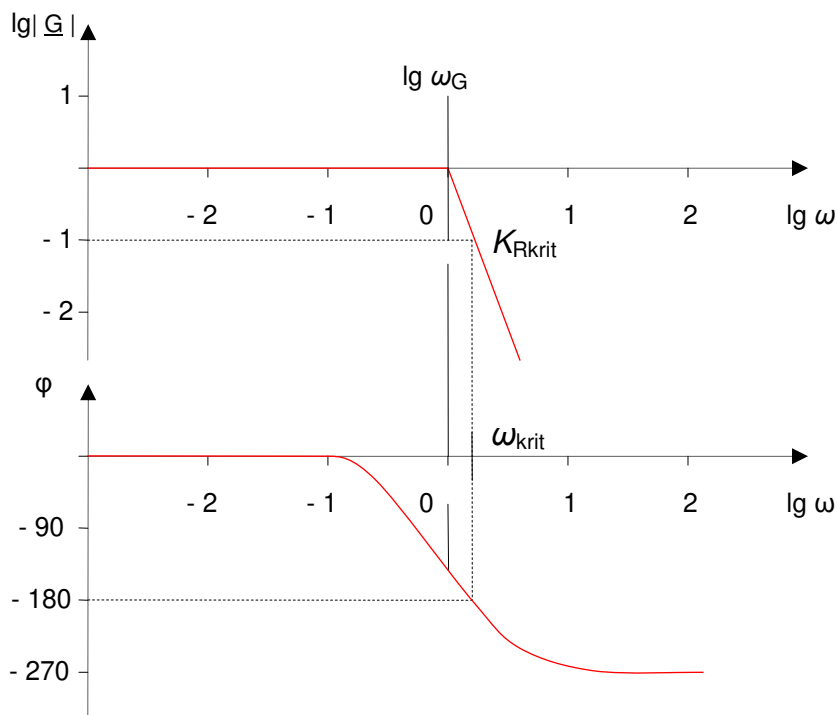
Regler	$K_R$	$T_i = T_N$	$T_D = T_V$
P-Regler	$0,5 \cdot K_{R\text{krit}}$	————	————
PI-Regler	$0,45 \cdot K_{R\text{krit}}$	$0,83 \cdot T_{\text{krit}}$	————
PID-Regler (additive Form)	$0,6 \cdot K_{R\text{krit}}$	$0,5 \cdot T_{\text{krit}}$	$0,125 \cdot T_{\text{krit}}$

**Tabelle 4.2:** Optimierung nach Ziegler und Nichols mit unbekanntem Werten [3]

Wenn die Regelstrecke nicht an der Stabilitätsgrenze betrieben werden darf, kann mit dem BODE-Diagramm die Periodendauer der Schwingung mit

$$T_{\text{krit}} = \frac{2\pi}{\omega_{\text{krit}}}$$

berechnet werden.



**Abbildung 4.3:** BODE-Diagramm zur Ermittlung der Periodendauer [12]

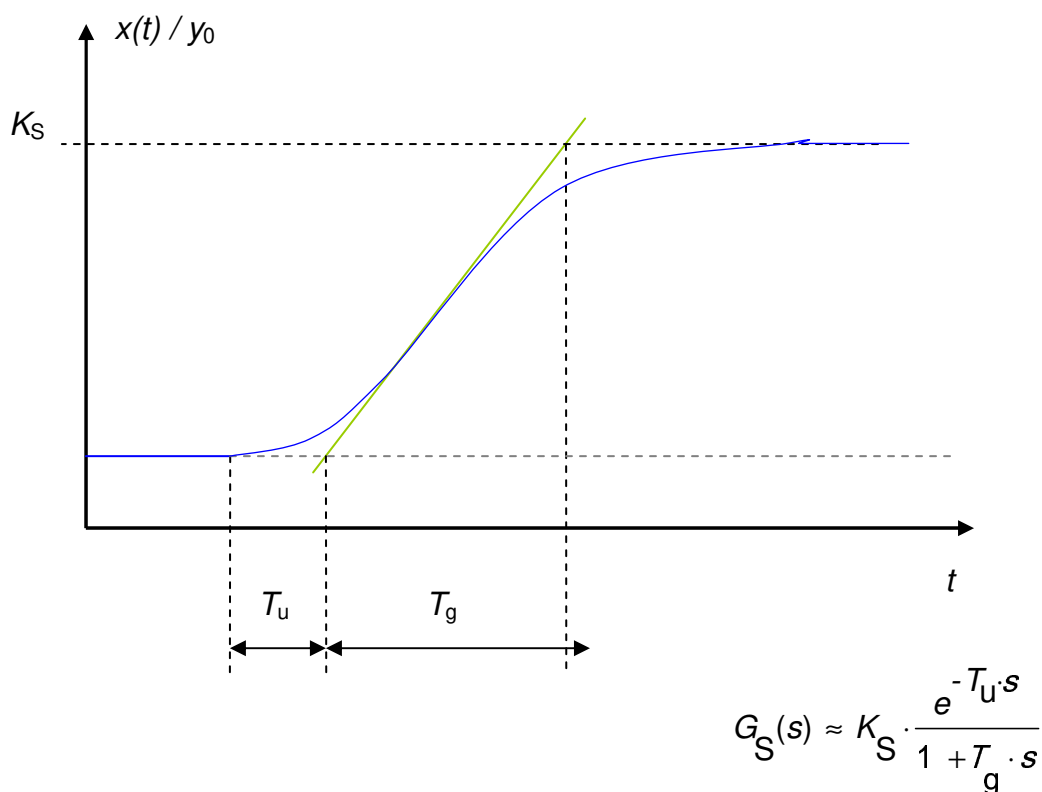
Die Optimierung gilt für sprungartige Störungen am Eingang der Strecke. Regelkreise mit PT2-Verhalten haben bei der Einstellung nach Ziegler und Nichols eine Dämpfung von  $D \approx 0,3$ .

Da in diesem entwickelten Programm die zur Optimierung benötigten Parameter durch das Tangentenverfahren ermittelt werden, kommen hier die Berechnungen aus der Tabelle 4.1 zur Anwendung.

### 4.3 Optimierung nach Takahashi [3]

Die Einstellregeln nach Takahashi wurden auf der Grundlage der Optimierung nach Ziegler und Nichols entwickelt. Die Einstellregeln berücksichtigen die Abtastzeit  $T$  und die Speicherung der Stellgröße in digitalen Regelungssystemen. Das Optimierungsverfahren nach Ziegler und Nichols liegt in zwei Formen vor. Die Auswertung der Sprungantwort verwendet die Kenngrößen der Regelstrecke:

Verstärkung  $K_S$ , Totzeit  $T_t$  und Zeitkonstante  $T_S$ . Anstelle der Totzeit  $T_t$  und der Zeitkonstante  $T_S$  werden auch die Kenngrößen der Sprungantwort Verzugszeit (Ersatztotzeit)  $T_u$  und Ausgleichzeit  $T_g$  verwendet.



**Abbildung 4.4:** Auswertung der Sprungantwort [3]

Die Einstellregeln nach Takahashi bei Vorgabe der Streckenwerte sind in der Tabelle 4.3 zusammengefasst. Sie gelten entsprechend für  $T_t$  und  $T_s$ . Für Abtastzeiten  $T \rightarrow 0$  ergeben sich wieder die Einstellregeln nach Ziegler und Nichols.

Die Einstellregeln nach Takahashi sind für  $T \leq 2 \cdot T_u$  gültig.

Regler	$K_R$	$T_i = T_N$	$T_D = T_V$
P-Regler	$\frac{T_g}{K_S \cdot (T_t + T)}$	————	————
PI-Regler	$0,9 \cdot \frac{T_g}{K_S \cdot \left(T_u + \frac{T}{2}\right)}$	$3,33 \cdot \left(T_u + \frac{T}{2}\right)$	————
PID-Regler (additive Form)	$1,2 \cdot \frac{T_g}{K_S \cdot (T_u + T)}$	$2,0 \cdot \frac{\left(T_u + \frac{T}{2}\right)^2}{T_u + T}$	$0,5 \cdot (T_u + T)$

**Tabelle 4.3:** Optimierung nach Takahashi mit bekannten Werten [3]

Liegen die Daten der Strecke nicht vor, dann gelten die Einstellregeln nach Ziegler und Nichols wie in der Tabelle 4.2.

## 5 Softwareentwicklung

Vorab wird hier eine Übersicht gegeben, welche Module und Klassen vorhanden sind und wofür sie im Programm genutzt werden. Die Klasse in diesem Programm ist ein von der Firma vorgefertigtes "Objekt", das mit der Programmierung des eigentlichen Programms nichts zu tun hat, sondern nur zum Lesen und Schreiben von Daten in der Steuerung vorhanden ist. Aus diesem Grund wird an dieser Stelle nur kurz auf die Klasse eingegangen. Die Module sind zwar auch vorgefertigt, mussten aber programmspezifisch geändert werden.

### Klasse:

- clsOPCGroup

Diese Klasse dient zum Lesen und Schreiben von Daten in der Steuerung. Hier hat jede OPC-Gruppe seinen eigenen OPC Server, oder jeder OPC-Server hat nur eine Gruppe. OPC-Server bedeutet eine Verbindung zu OPC oder SimaticNet. Das Schreiben muss in einem Block erfolgen. Es dürfen keine Lücken zwischen den schriftlichen Adressen sein, sonst wird das Schreiben in mehr als einem Block durchgeführt.

### Module:

- mdlMain

Dieses Modul dient mehreren Zwecken. Zum einen enthält dieses Modul die Deklaration der Verbindung zur Datenbank, die als erstes im Programm genutzt wird. Zum anderen sind hier Prozeduren festgelegt, die bestimmte Festlegungen von z.B. der Anzahl der Adressen, die gelesen werden sollen und den Namen der OPC-Gruppe enthalten. Diese Festlegungen werden dann wiederum an die Klasse übergeben.

- mdlError

In diesem Modul werden Daten aus den Prozeduren im Programm wie z.B. die Ursache eines Fehlers und dessen Fehlernummer übergeben und in einer Text-Datei gespeichert. Dazu wird in bestimmten Prozeduren im Programm, in denen ein Fehler verursacht werden kann, ein Aufruf für dieses Modul



erzeugt. Im Aufruf stehen dann die Fehlernummer, die Fehlerbeschreibung und der Name der Prozedur, in welcher der Fehler verursacht wurde.

- mdILogFileGeneration

Hier werden Daten wie auch im vorherigen Modul in Text-Dateien gespeichert. Wenn der Anwender das Programm nutzt, werden in bestimmten Prozeduren Daten in eine Textdatei geschrieben, die automatisch angelegt wird. Wenn der Anwender z.B. einen Regler aus dem DataGrid auswählt, erscheinen in der Text-Datei der Name des Reglers und dessen Anfangswerte. So kann aus der Textdatei sehen, wann was mit welchem Regler gemacht wurde.

## 5.1 Die Startform frmVerbindung



**Abbildung 5.1:** Übersicht der Startform frmVerbindung

Diese Form erscheint dem Anwender, wenn er das Programm startet. Sie hat den Zweck dem Anwender die Möglichkeit zu geben, eine Datenbank auszuwählen. Dieses ist deswegen von Nutzen, da bei der späteren Optimierung der Regler mehr als eine Datenbank vorhanden sein kann.

Bevor der Anwender die Datenbank im Programm auswählen kann, muss diese erstmal in Windows unter Systemsteuerung / Verwaltung / Datenquellen(ODBC) angelegt werden.

Wenn das Programm nach dem Anlegen der Datenbank gestartet wird, kann der Anwender die angelegte Datenbank in der ComboBox auswählen, wie nachfolgend abgebildet.



**Abbildung 5.2:** Auswahl der Datenbank

Nachdem der Anwender die Datenbank ausgewählt hat, muss er den Benutzernamen und das Passwort in die beiden TextBoxen eintragen. Erst dann kann er auf den Verbinden-Button klicken. Die Daten werden dann an das Modul mdlMain übergeben, in dem diese Daten für die Verbindung zur Datenbank geschrieben werden. Somit ist die Verbindung für die Datenbank angelegt und geöffnet.

### 5.1.1 Die Verbindung zur Datenbank

Um die oben beschriebenen Daten der Regler auslesen zu können, muss das Programm, wie zuvor erwähnt, eine Verbindung zur Datenbank herstellen. In Visual BASIC wird diese Verbindung über das Connection-Objekt hergestellt. Dieses greift direkt oder über eine ODBC Verbindung, die über einen Data Source Name (DSN) angesprochen werden kann, auf die Datenbank zu. Das Connection-Objekt benötigt zur Verbindung den Connection String. In dieser Zeichenkette sind die Informationen über den Provider, die Datenbank oder den DSN-Namen, den Benutzernamen und das Passwort enthalten. In der Regel muss diese Verbindung nur einmal geöffnet werden und lässt sich dann an vielen Stellen im Programm benutzen. In diesem Projekt wird diese Verbindung nur einmal verwendet, nämlich dann, wenn das Programm gestartet wird.

### 5.1.2 Beispiel einer Datenbankverbindung

Nachdem die Connection definiert wurde, muss noch ein Recordset definiert werden. Das Recordset dient zum Datenaustausch mit der Datenbank-, bzw. über die ODBC-Verbindung.

```
Public Cn As New Connection, Rs As New Recordset
```

Die Verbindung wird hier am Anfang des Programms geöffnet.

```
Cn.ConnectionString = "Provider=MSDataShape.1;DSN=OTASODBC;User ID=Max;Password=123"  
Cn.Open
```

Um eine Datenbankabfrage stellen zu können, muss man dem Recordset den String mit der SQL-Anweisung

```
Rs.Source = strsql
```

und die Datenbankverbindung zuweisen.

```
Set Rs.ActiveConnection = Cn
```

Wenn man das Recordset jetzt öffnet, wird die SQL-Anweisung über den Provider und die ODBC-Verbindung an die Datenbank geschickt und beantwortet. Das Ergebnis steht dann im Recordset, welches wie eine Tabelle aufgebaut ist.

```
Rs.Open
```

Vor Beendigung des Programms müssen die Verbindung und das Recordset unbedingt geschlossen werden, da ansonsten die Ressourcen, die für eine Verbindung vom Betriebssystem reserviert werden, nicht wieder freigegeben werden.

```
Set Cn = Nothing
```

```
Set Rs = Nothing
```

### 5.1.3 Lesen der Daten aus einem Recordset

Um Daten aus einem Recordset darzustellen oder sie weiter verarbeiten zu können gibt es zwei Möglichkeiten:

Die erste Möglichkeit ist das Anzeigen mit einem Grid. Ein Grid ist ein Tabellen-Steuerungselement. Einigen Grids kann man den Inhalt des Recordsets ganz einfach übergeben (DataGrid1 ist der Name des Grids):

```
Set DataGrid1.DataSource = Rs
```

Die Daten werden in diesem Fall selbständig ausgelesen und angezeigt.

Die zweite Möglichkeit ist die manuelle Übergabe der Daten des Recordsets an ein Grid, oder aber auch das Weiterverarbeiten von Werten. Für diese Anwendungsgebiete bietet das Recordset einige Methoden und Funktionen an. Wie schon oben erwähnt, kann man sich ein Recordset wie eine Tabelle vorstellen. Es gibt einen Cursor, den man durch die Tabelle bewegen kann, und an dessen Position man den Wert auslesen kann.

Die für dieses Projekt wichtigen Methoden und Funktionen sollen hier kurz vorgestellt werden.

Open                   startet die SQL-Abfrage mit dem vorher zugewiesenen SQL-String

Close                   gibt den reservierten Speicherplatz wieder frei

MoveNext               bewegt den Cursor des Recordsets eine Zeile nach unten

MovePrevious           bewegt den Cursor eine Zeile nach oben

MoveFirst              bewegt den Cursor in die erste Zeile

MoveLast               bewegt den Cursor in die letzte Zeile

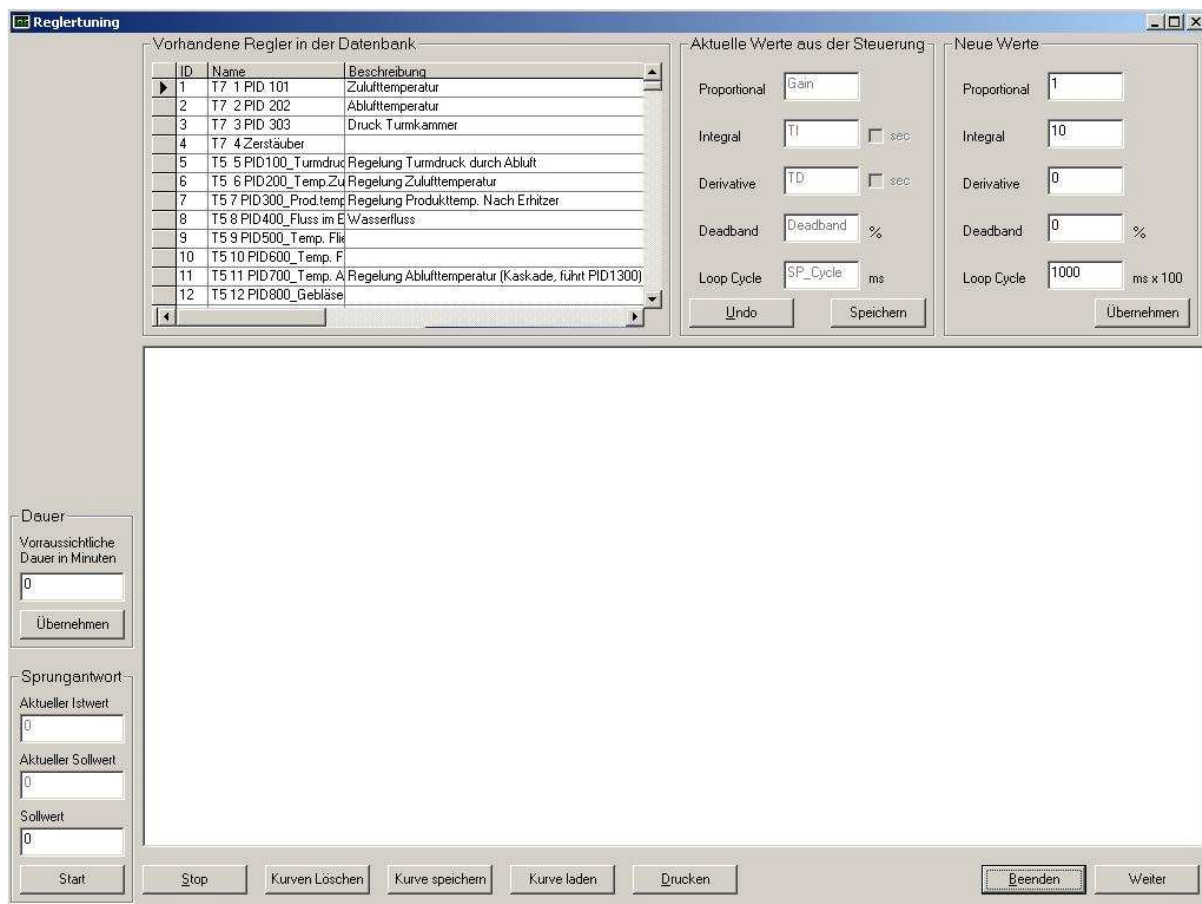
EOF (Funktion) bekommt den Rückgabewert „True“ wenn der Cursor nach unten über die Tabelle hinausgelaufen ist (**E**nd **O**f **F**ile).

BOF (Funktion) bekommt den Rückgabewert „True“ wenn der Cursor nach oben über die Tabelle hinausgelaufen ist (**B**eginning **O**f **F**ile).

Fields (Funktion, Spalte) gibt den Wert des Feldes der aktuellen Zeile und der in „Spalte“ angegebenen Spalte an.

Im Rahmen dieser Diplomarbeit wurden nur die Funktionen Fields, EOF und die Methoden Open, Close und MoveNext verwendet. Es werden keine Daten über das Recordset in die Datenbank zurück geschrieben, sondern nur einmal gelesen und direkt an ein Grid gegeben. Diese fünf Funktionen und Methoden reichen aus, um das gesamte Recordset in einer Schleife auszulesen.

## 5.2 Die Form frmReglerauswahl



**Abbildung 5.3:** Übersicht der Form frmReglerauswahl

Die oben abgebildete Form erscheint, wenn der Anwender eine Datenbankverbindung aus der Startform hergestellt hat. Hier werden wesentliche Parameter und Texte der Regler aufgelistet, um den Regler richtig identifizieren zu können. Werte aus der Steuerung können hier eingelesen und auch wieder in die Steuerung geschrieben werden. Der Anwender kann hier einen Sollwertsprung auf den Regler geben und die Systemantwort aufzeichnen. Bevor die Aufzeichnung gestartet werden kann, muss der Anwender eine voraussichtliche Dauer vorgeben, um den x-Bereich in der PictureBox festzulegen. Diese kann dann abgespeichert und zum Vergleich mit dem optimierten Regler wieder in die PictureBox geladen werden. Die Startform besteht aus Objekten, deren Funktionen im weiteren Verlauf dieser Arbeit erklärt werden.

### 5.2.1 Das DataGrid

Hier erscheinen die Regler-ID, der Reglername und die Reglerbeschreibung aus der vorher ausgewählten Datenbank. Das DataGrid ist ein tabellenartiges Steuerelement in Visual BASIC, welches die Möglichkeit bietet Daten aus der Datenbank auszulesen und zu konfigurieren. Hier wird es zur Auflistung von Reglerinformationen und zur Auswahl eines Reglers eingesetzt. Der Anwender bekommt einen Überblick über die Anzahl der vorhandenen Regler und kann den richtigen Regler anhand der Informationen auswählen. Wenn man hier einen Regler auswählt, wird in der Steuerung überprüft, ob der Regler gerade aktiv ist. Wenn der ausgewählte Regler gerade aktiv ist, wird der Anwender darüber informiert und die weitere Nutzung des Reglers wird unterbunden. So verhindert man, dass der Regler seinen aktiven Prozess abbricht. Wenn der Regler zu diesem Zeitpunkt nicht aktiv ist, werden die Werte aus der Steuerung ausgelesen und angezeigt.

Im DataGrid befinden sich noch zwei weitere Spalten, die aber nicht sichtbar sind. Sie beinhalten die Min- und Maxwerte des Sollwertes, die später im Programm benötigt werden.

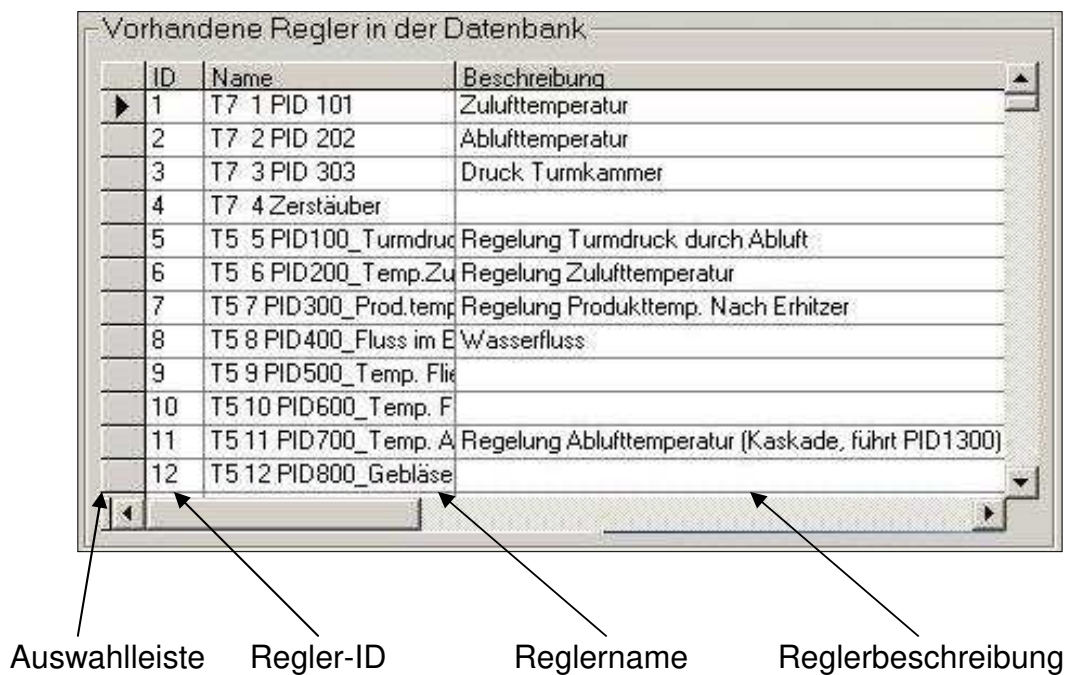


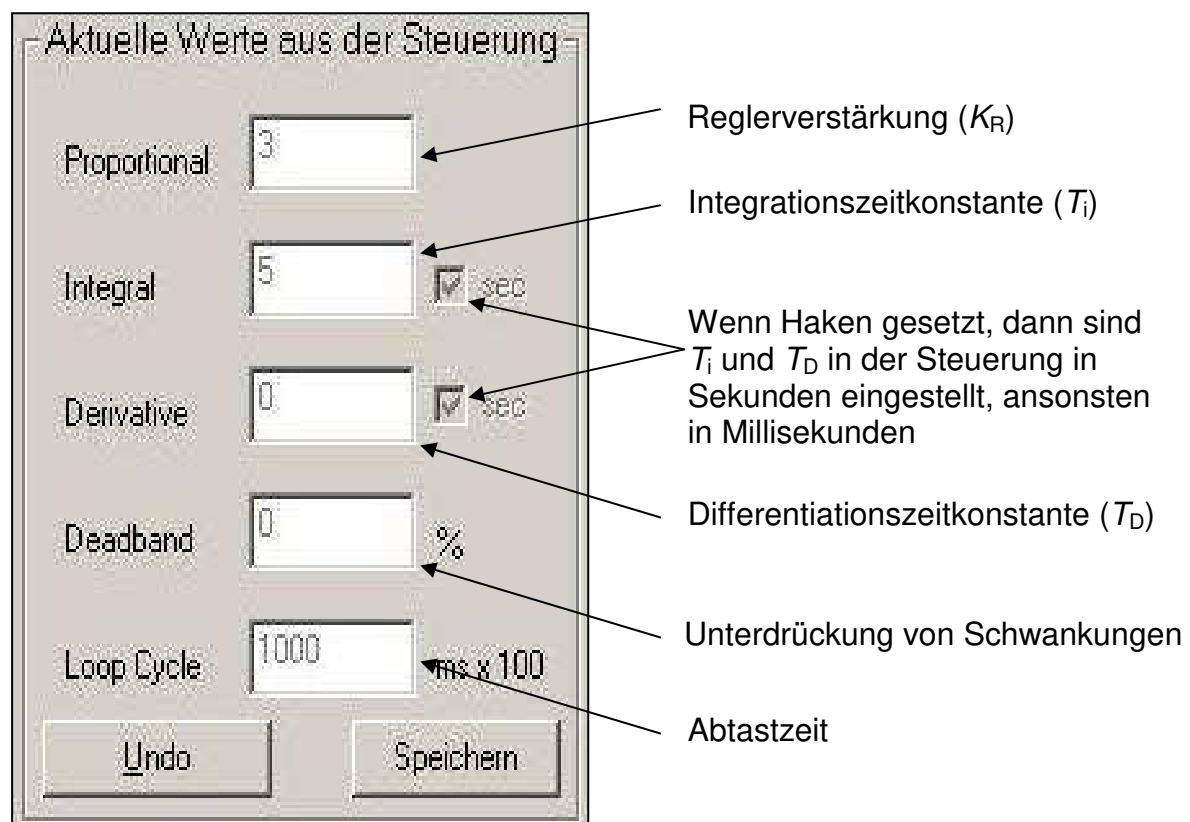
Abbildung 5.4: Das DataGrid



### 5.2.2 Die Objekte im Frame1

In diesem Frame (Rahmen) sind insgesamt fünf TextBoxen und zwei CheckBoxen implementiert. Im gesamten Programm haben die Frames keine weiteren Funktionen. Sie dienen lediglich der besseren Optik und dazu, die in ihnen integrierten Objekte beim verschieben immer auf der gleichen Position zu halten.

Die Objekte in diesem Frame sind dazu bestimmt, die Werte aus der Steuerung anzuzeigen und durch einen Timer jede Sekunde zu aktualisieren, um sicher zu gehen, dass nicht schon andere Werte in der Steuerung stehen.



**Abbildung 5.5:** Objekte im Frame1

Die Werte, die gerade in diesen TextBoxen stehen, können durch den Speichern-Button in einer CSV-Datei gespeichert werden. Der Anwender hat somit die Möglichkeit diese gespeicherten Werte wieder aus der CSV-Datei auszulesen, um somit zu alten Werten zurückzukehren.

### 5.2.2.1 Die Verbindung zur Steuerung

Nachdem der Regler aus dem DataGrid ausgewählt wurde, wird die OPC-Verbindung geöffnet. Ein Programm in Visual BASIC kann mehrere OPC-Gruppen enthalten, um mit der Steuerung zu kommunizieren. In diesem Programm wird nur eine OPC-Gruppe angelegt, da die Gruppe nur einmal beim Anlegen der relevanten Adressen aus der Steuerung benötigt wird. Die für dieses Programm notwendige OPC-Gruppe wird hier mOPCGroup(1) genannt.

Bei der Anlegung der OPC-Gruppe müssen vorher im Modul mdlMain des Programms die dafür nötigen OPC-Parameter der Verbindung deklariert werden. Dazu gehören die folgenden Bestandteile:

- mOPCGroup(1).OPCServerName = "OPC.SimaticNet"
- mOPCGroup(1).OPCGroupName = "AxelTest1"
- mOPCGroup(1).OPCUpdateRate = 1000
- mOPCGroup(1).OPCItemsCount = 17

Wenn die Daten bei der Auswahl des Reglers benötigt werden, wird ein Aufruf gestartet.

`call OPC_Connect`

In dieser Form stehen alle Strings für die Adressen, die in der Steuerung angesprochen werden sollen. Die Strings werden dann an die Prozedur im Modul mdlMain übergeben und in Arrays gespeichert.

`Public Sub OPC_Connect`



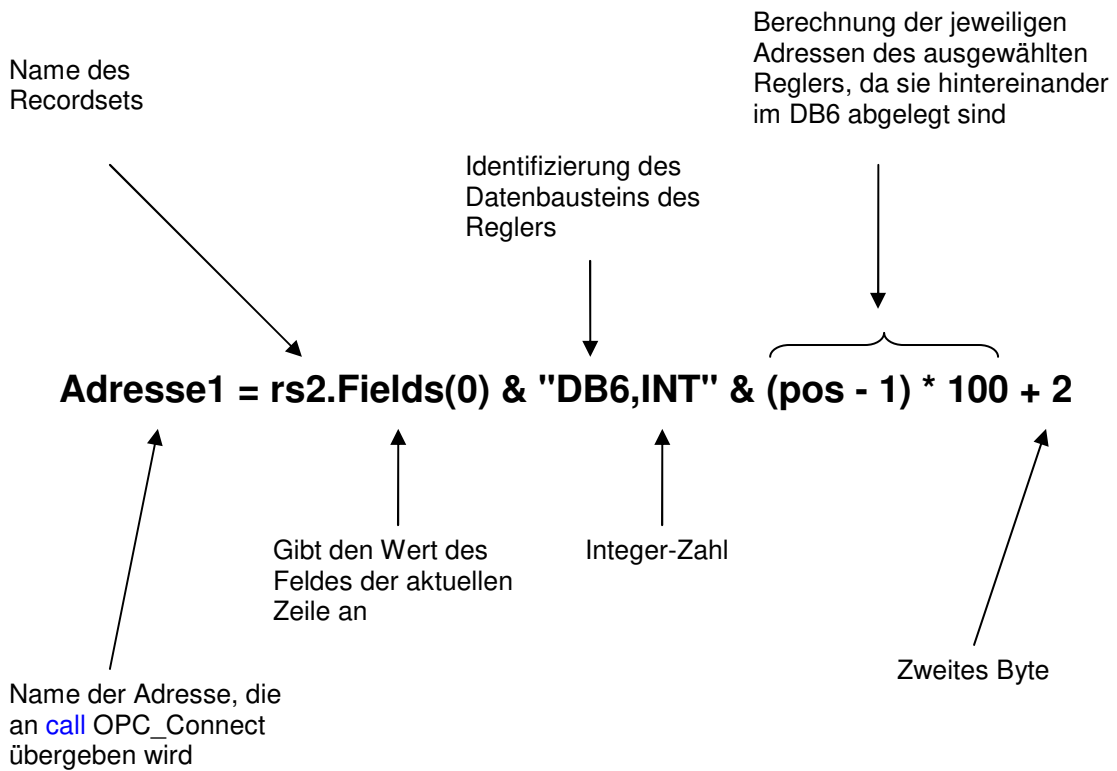
im Modul mdlMain liest daraufhin die Daten aus der Steuerung aus. Hier wird geprüft, ob Fehler entstanden sind. Wenn hier keine Fehler aufgetreten sind, werden die Werte, die in den dazugehörigen Adressen stehen, ausgelesen und in weiteren Arrays gespeichert. Die Werte aus den Arrays werden dann an die Text- und CheckBoxen übergeben und in der Form frmReglerauswahl angezeigt. Gleichzeitig wird in der Steuerung überprüft, ob die Zeiten für die Integrations- und Differentiationszeit in Sekunden oder Millisekunden angezeigt werden. Wenn in der Adresse einer der Zeiten ein "True" steht wird die Zeit in Sekunden angezeigt.

Der Speichern-Button dient dazu, die Werte aus diesen TextBoxen in einer CSV-Datei zu speichern. Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfacher strukturierter Daten. Das Kürzel CSV steht dabei für **C**haracter **S**eparated **V**alues oder Comma Separated Values, weil die einzelnen Werte durch ein spezielles Trennzeichen, bspw. das Komma, getrennt werden. Ein allgemeiner Standard für das Dateiformat existiert nicht. Als Trennzeichen sind neben Komma auch Semikolon, Doppelpunkt, Tabulator und andere Zeichen üblich.

Hier werden die Daten durch den Split-Befehl im Programm mit einem Semikolon getrennt.

Der Undo-Button ist dafür da, die Werte, die über den Speichern-Button in der CSV-Datei gespeichert wurden wieder auszulesen. So hat der Anwender die Möglichkeit zu älteren Werten zurückzukehren, sofern die aktuelle Einstellung nicht optimal ist. Beim Auslesen werden die Werten nicht gleich in die Steuerung geschrieben, sondern erscheinen in den TextBoxen des Frame2. Diese Lösung ist vorteilhafter, da bei der Auswahl der falschen CSV-Datei nicht gleich die Werte für den Regler in die Steuerung geschrieben werden. Der Anwender kann die Werte somit ersteinmal sehen und dann entscheiden, ob er sie durch den Übernehmen-Button im anderen Frame in die Steuerung schreiben möchte.

Um Adressen in der Steuerung ansprechen und deren Werte auslesen zu können, muss die Adresse in Visual BASIC in Strings zusammengestellt werden. Die folgende Abbildung zeigt, wie ein String für eine Adresse in Visual BASIC aussehen muss.



**Abbildung 5.6:** Aufbau eines Strings zum Auslesen der Werte in den Adressen

### 5.2.3 Die Objekte im Frame2

Dieser Frame dient zum Schreiben in die Steuerung, um den ausgewählten Regler vorher so zu konfigurieren, dass er gut regelt. Wenn hier Werte geändert werden, wird das, was in den TextBoxen steht, beim Klick auf den Übernehmen-Button in die Steuerung geschrieben. Nach kurzer Zeit erscheinen dann die Werte in den TextBoxen des Frame1, da diese jede Sekunde aktualisiert werden.

Die CheckBoxen sind nur im Frame1 vorhanden, weil diese nicht geändert werden dürfen. Bei der Inbetriebnahme einer Anlage werden die Zeiten in Sekunden oder Millisekunden nur einmal eingestellt und sind dann fix.

Wenn der Anwender hier alle Parameter richtig eingestellt hat, wird beim Klick auf den Übernehmen-Button für alle fünf Adressen die Prozedur writeValuesOPC aufgerufen und die Werte aus den TextBoxen werden nacheinander in die Steuerung geschrieben.

Neue Werte	
Proportional	1
Integral	10
Derivative	0
Deadband	0 %
Loop Cycle	1000 ms x 100

Übernehmen

Alle Werte können hier eingestellt werden. Die Werte, die jetzt in diesen Objekten stehen, haben noch nichts mit der eigentlichen OPC-Verbindung zu tun. Sie sind nur voreingestellt und werden beim Start des Programms automatisch in diese Objekte geschrieben.

**Abbildung 5.7:** Objekte im Frame2

### 5.2.3.1 Schreiben der Daten in die Steuerung

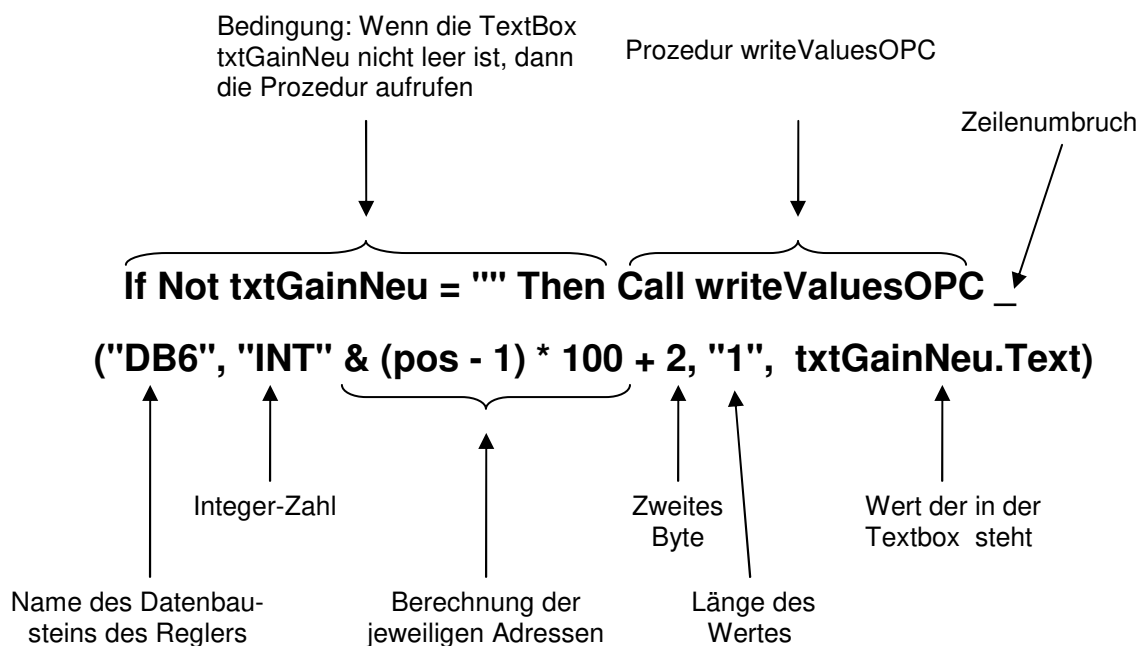
Das Schreiben der Daten in die Steuerung ist in diesem Programm nicht mehr so kompliziert wie beim Lesen der Daten, da die OPC-Verbindung bereits definiert wurde.

Ein gutes Beispiel ist z.B. die Verstärkung (Gain) der Regler. Sie ist von der Reihenfolge her die erste Adresse, die in der Steuerung angesprochen wird und dessen Wert als erstes ausgelesen und in eine Textbox geschrieben wird. Wenn der vorgegebene Wert in die Steuerung geschrieben werden soll, wird durch den Aufruf

`call writeValuesOPC`

die Prozedur zum Schreiben im Modul mdlMain aufgerufen. Diese Prozedur wird nicht für alle zu schreibenden Werte einmal aufgerufen, wie es z.B. beim Lesen ist. Beim Schreiben wird dieser Aufruf für jede Variable erzeugt, da noch eine Bedingung an die Textboxen vorgegeben werden müssen.

Wie aus in der folgenden Abbildung zu entnehmen ist, ist der String zum Schreiben dem String zum Lesen sehr ähnlich.



**Abbildung 5.8:** Aufbau eines Strings zum Schreiben der Werte in die Steuerung

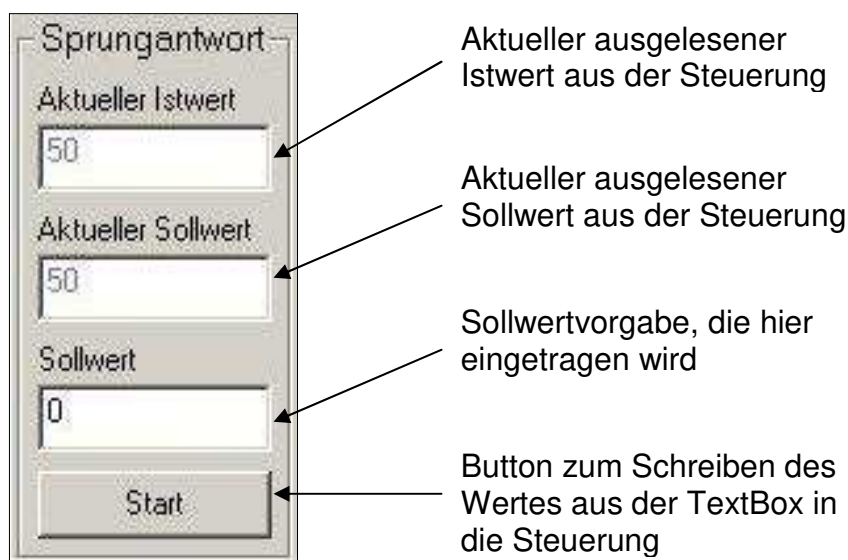
### 5.2.4 Starten der Aufzeichnung der Sprungantwort in der PictureBox

Hier kann die Sprungantwort auf einen Sollwertsprung aufgezeichnet werden. Somit bekommt man einen Überblick über das Gesamtsystem und kann nachweisen, ob der Regler nach der Optimierung ein besseres Verhalten aufweist.

Im ersten Schritt muss der Anwender nach der Auswahl des Reglers einen Sollwert in die dafür vorgesehene TextBox schreiben. In dieser Form ist auf der linken Seite der Frame3 mit drei TextBoxen für den aktuellen Soll- und Istwert, der Sollwertvorgabe und dem Start-Button.

Die TextBoxen für den aktuellen Ist- und Sollwert sind so eingestellt, dass der Anwender keine Eintragungen vornehmen kann. Die TextBox für die Sollwertvorgabe ist dagegen offen zum Ändern von Werten. Der Anwender kann hier allerdings nur Werte eingeben, die im Bereich zwischen Min- und Maxwert liegen. Diese stehen in den nicht sichtbaren Spalten des DataGrids. Sollte der Wert nicht in diesem Bereich liegen wird eine MessageBox geöffnet, die dem Anwender den zulässigen Bereich mitteilt.

Welche TextBoxen nicht zum Schreiben freigegeben sind kann man daran erkennen, dass der Inhalt in Grau dargestellt wird.



**Abbildung 5.9:** Der Frame3 zum Ändern des Sollwertes



Die Prozedur beim Klick-Ereignis des Start-Buttons umfasst mehrere Call-Aufrufe die zur Aufzeichnung benötigt werden. Diese stehen allerdings nicht in der Prozedur selbst, sondern in einem Timer-Ereignis. In der Prozedur wird nur der Timer aufgerufen und das Intervall festgelegt.

Die Aufrufe heißen hier:

- `call ReadFromPLC`
  
- `call csv_WriteSollwert`
  
- `call TrendView_show`

Bevor die Prozeduren aufgerufen werden, erfolgt erst die Einstellung des Timers auf ein Intervall von 1000 Millisekunden und dessen Aktivierung. Dadurch werden die oben genannten Call-Aufrufe jede Sekunde ausgeführt.

Der Aufruf `ReadFromPLC` wurde schon erklärt und muss an dieser Stelle nicht mehr behandelt werden.

Bei dem Aufruf `csv_WriteSollwert` wird ein neuer Unterordner im Ordner des Programms angelegt. In diesem Unterordner legt sich dann eine erzeugte CSV-Datei ab, die hier `Test.csv` heißt. Durch den Aufruf `ReadFromPLC` werden die Werte vom Soll- und Istwert jede Sekunde aus der Steuerung ausgelesen. Durch den Aufruf `csv_WriteSollwert` lassen sich dann diese Werte und der jeweilige Schritt des Timers in die schon benannte CSV-Datei schreiben. Die CSV-Datei besteht dann aus drei Spalten, deren gerade aktuellen Werte dann nach jeder Sekunde in die nächste Zeile geschrieben werden.

Die Werte könnten auch ohne den Umweg über die CSV-Datei im Programm verarbeitet werden. Der Vorteil durch die Erzeugung der CSV-Datei ist jedoch, dass sich diese Werte in einer anderen CSV-Datei speichern lassen und somit durch späteres Auslesen der gespeicherten Datei, mehrere Kurven miteinander verglichen werden können. So kann der Anwender erkennen, ob z.B. der gerade optimierte Regler eine bessere Güte aufweist, als das zuvor gespeicherte Reglerverhalten. Auf diesen Teil der Diplomarbeit wird erst bei der Beschreibung der weiteren Funktionen der Buttons eingegangen.

Bevor man allerdings mit der Aufzeichnung beginnen kann, muss der ausgewählte Regler auf Automatik-Betrieb gestellt werden.

Der Aufbau der Datei ist in der folgenden Abbildung verdeutlicht:

Spalte "A" für den Timer-Schritt. Für die Aufzeichnung sind dies die x-Werte für beide Linien

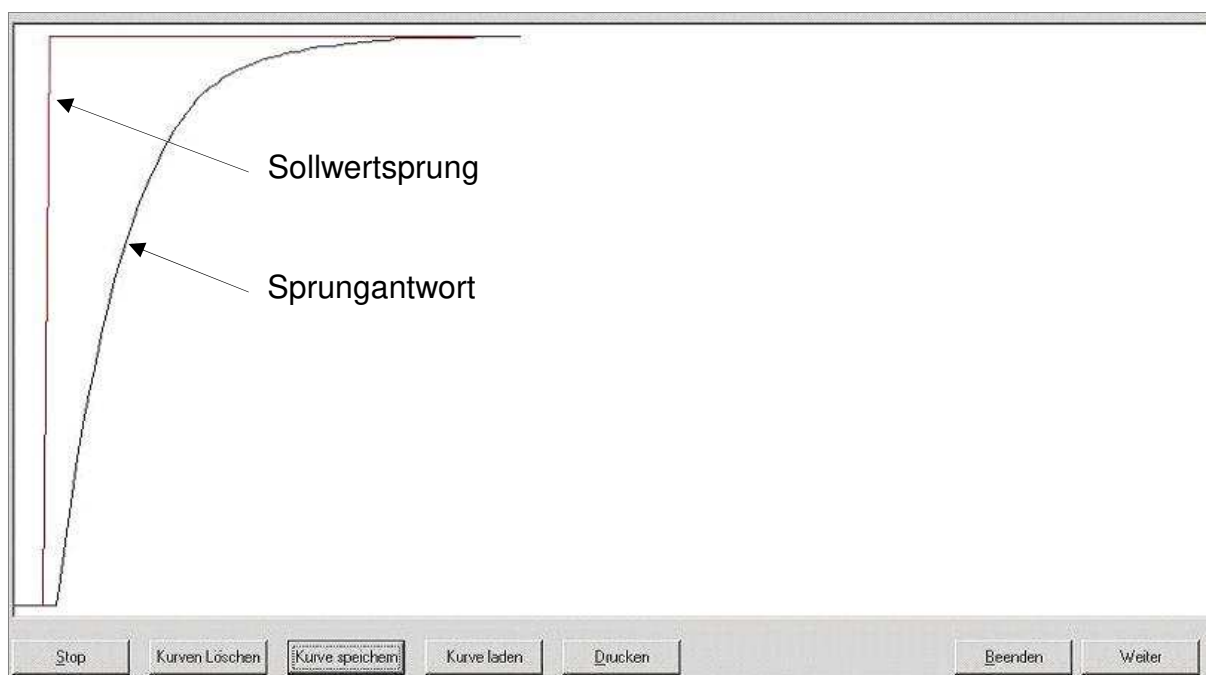
Spalte "B" für die Sprungantwort. Dies sind die Werte in y-Richtung

Spalte "C" für die Sollwertsprung. Dies sind ebenfalls die Werte in y-Richtung

	A	B	C
1	1	0	0
2	2	0	0
3	3	0	0
4	4	0	0
5	5	0	0
6	6	0	0
7	7	0	0
8	8	0	0
9	9	0	0
10	10	0	0
11	11	0	0
12	12	0	0
13	13	0	0
14	14	0	200
15	15	0	200
16	16	35	200
17	17	50	200
18	18	87	200
19	19	107	200
20	20	136	200
21	21	147	200
22	22	157	200
23	23	164	200
24	24	175	200
25	25	180	200
26	26	183	200
27	27	186	200
28	28	191	200
29	29	192	200
30	30	194	200
31	31	195	200
32	32	196	200
33	33	197	200
34	34	198	200
35	35	198	200
36	36	198	200
37	37	199	200
38	38	199	200
39	39	199	200
40	40	199	200
41	41	200	200

Abbildung 5.10: Aufbau der CSV-Datei

Der Aufruf TrendView\_show nutzt diese Daten aus der Datei Test.csv. In dieser Prozedur sind der Pfad und der Name der Datei angegeben, die zum Aufzeichnen der Linien benötigt werden. Der Split-Befehl im Programm liest die einzelnen Zeilen für alle Spalten jede Sekunde wieder aus. Die Spalte "B" wird der Sprungantwort und die Spalte "C" dann dem Sollwertsprung zum Aufzeichnen der beiden Linien zugeordnet. In der PictureBox werden Punkte eingezeichnet, die anschließend mit vielen kleinen Linien verbunden sind. Das Ergebnis ist in der folgenden Abbildung dargestellt:



**Abbildung 5.11:** Aufzeichnung von Sollwertsprung und Sprungantwort

In der Abbildung ist zu erkennen, wie das System auf eine Änderung des Sollwertes reagiert. Der Regler ist hier noch nicht optimiert.

Der Anwender hat die Möglichkeit die Aufzeichnung jederzeit durch den Stop-Button zu beenden. In dieser Prozedur wird dann der Timer angehalten und das weitere Schreiben in die CSV-Datei verhindert. Sofern der Anwender danach wieder den Start-Button klickt, startet der Timer wieder und die neuen Daten werden an die vorherigen in der Datei Test.csv angehängt.

Wenn der Anwender die Prozedur stoppt, kann die Aufzeichnung in der PictureBox gelöscht werden. Die CSV-Datei wird beim Klick-Ereignis des Kurven-Löschen-Buttons gelöscht und die PictureBox wieder auf den Anfangszustand zurückgesetzt bzw. gesäubert.

Bevor man die Aufzeichnung löscht, besteht auch die Möglichkeit die Aufzeichnung durch den Kurve-Speichern-Button zu speichern. In diesem Fall öffnet sich ein Dialog, der automatisch den Namen des Reglers und das aktuelle Datum als Vorschlag für den Dateinamen enthält. Wenn der Anwender die CSV-Datei unter dem vorgeschlagenen Namen speichert, wird sie in diesem Ordner abgelegt und die Datei Test.csv gelöscht.

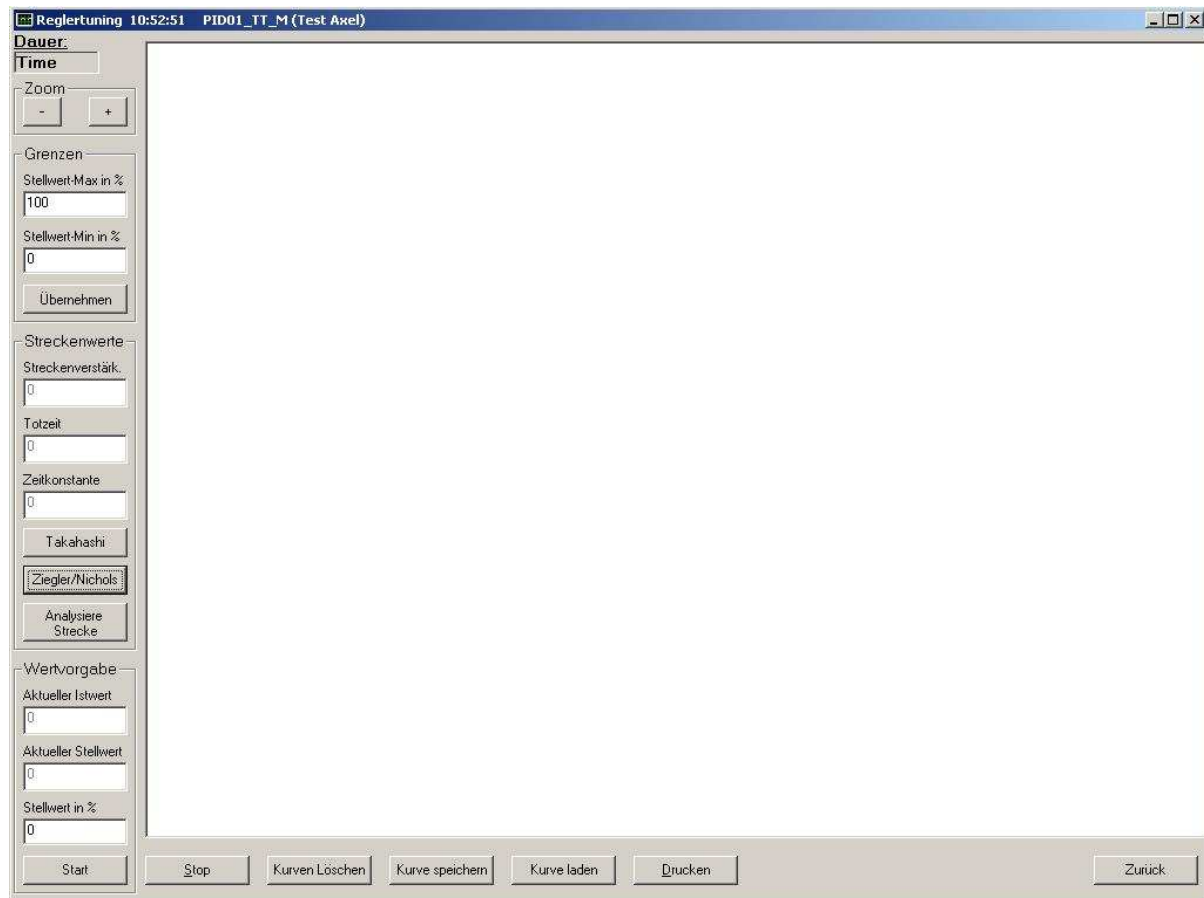
Durch den Kurve-Laden-Button kann der Anwender die Aufzeichnung wieder auslesen und sie in der PictureBox anzeigen lassen. Hier öffnet sich ebenfalls ein Dialog, aus welchem die zu öffnende Kurve ausgewählt werden kann. Dieses dient dem Vergleich mehrerer Aufzeichnungen eines Reglers.

Der Drucken-Button gibt dem Anwender die Möglichkeit, die aufgezeichneten Kurven auf DIN A4-Papier zu drucken. Der Dialog der sich dann öffnet bietet die Möglichkeit, einen eingerichteten Drucker auszuwählen.

Der Beenden-Button ist dafür vorgesehen, alle Formen des Programms zu schließen und somit das Programm zu beenden. Der Anwender wird über eine MessageBox gefragt, ob er das Programm wirklich beenden möchte, welches er dann Bestätigung muss oder auch nicht. Sofern er die Abfrage mit ja bestätigt, werden hier Parameter die bei der Auswahl eines Reglers im DataGridView aus der Steuerung ausgelesen wurden wieder zurückgesetzt, um den gerade aktiven Regler wieder so einzustellen, wie er vor dem Start des Programms gewesen ist. So ist er dann wieder für den Prozess aktiv.

Der Weiter-Button bringt den Anwender auf die zweite Form des Programms, welche hier frmReglerbearbeitung heißt. Sofern der Anwender noch keinen Regler aus dem DataGridView ausgewählt hat, wird er durch eine MessageBox darüber informiert und gleichzeitig aufgefordert einen Regler auszuwählen. Solange kein Regler gewählt wird, gelangt der Anwender auch nicht auf die nächste Form.

## 5.3 Die Form frmReglerbearbeitung



**Abbildung 5.12:** Übersicht der Form frmReglerbearbeitung

Diese Form ist für die Optimierung des ausgewählten Reglers konzipiert. Die Voreinstellungen beziehen sich hierbei nur auf die Stellgrößenbegrenzung, die der Benutzer an dieser Stelle vorgeben kann. Wenn diese Form gestartet wird, erscheint in der oberen Leiste der Form der Name des ausgewählten Reglers, damit der Benutzer immer weiß, welchen Regler er gerade bearbeitet. Weiterhin erscheinen hier gleichzeitig der aktuelle Istwert und Stellwert des Reglers, die durch den selben Timer aus der Form frmReglerauswahl jede Sekunde aktualisiert werden.

Hier muss der Regler auf Hand-Betrieb gestellt werden, um somit den Regler vom System abzukoppeln. Dieses ist Voraussetzung, da nur so ein Stellwertsprung durchgeführt werden kann. Bei der Optimierung wird nur das Streckenverhalten analysiert, nicht das gesamte System.

### 5.3.1 Einstellen der Min- und Maxwerte der Stellwertgröße

Diese Parameter werden von einem anderen Timer nur alle zwanzig Sekunden aktualisiert, um dem Anwender die Zeit zu geben diese Werte in die dafür vorgesehenen TextBoxen zu schreiben und in die Steuerung zu übertragen.

Diese Werte sind für das eigentliche Ziel dieser Arbeit nicht von Bedeutung. Die Ingenieure der Firma finden es jedoch komfortabler, die Grenzen von z.B. 0 – 100% der Stellgröße von der Software aus ändern zu können. Die Programmierung dafür ist nicht sonderlich komplex, da die Verbindung zur Steuerung vorher schon definiert wurde. Hierzu mussten lediglich zwei weitere Strings für die Adressen zusammengestellt und an die OPC-Verbindung übergeben werden. Der Anwender kann diese Werte aus den TextBoxen wieder über den Klick auf einen anderen dafür vorgesehenen Übernehmen-Button in die Steuerung schreiben.

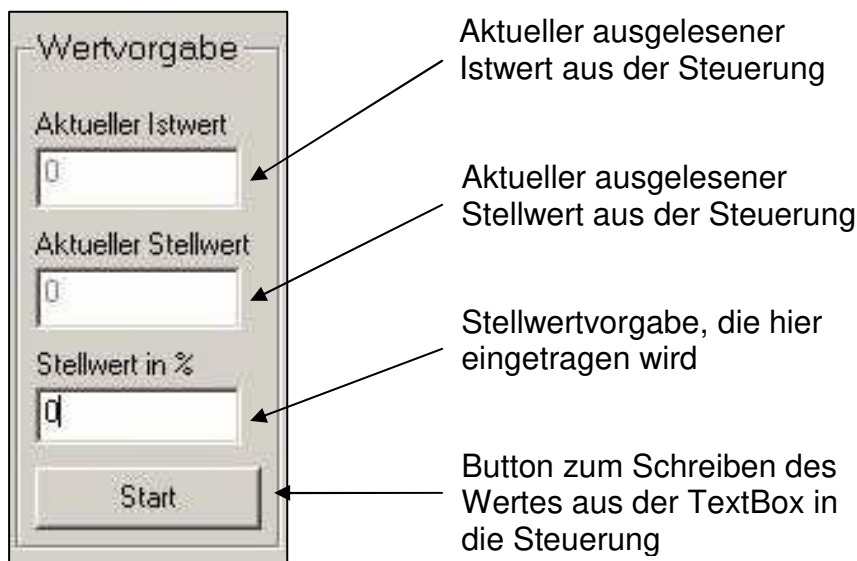


**Abbildung 5.13:** Der Frame4 zur Stellwertbegrenzung

### 5.3.2 Starten der Aufzeichnung der Sprungantwort der Strecke

Nachdem alle Konfigurationen durchgeführt wurden, kann der Anwender mit der Aufzeichnung des Stellwertsprungs und der Sprungantwort der Strecke des ausgewählten Reglers beginnen.

Auf dieser Form ist unten links der Frame5 mit drei TextBoxen für den aktuellen Ist-, Stellwert und der Stellwertvorgabe sowie dem Start-Button.

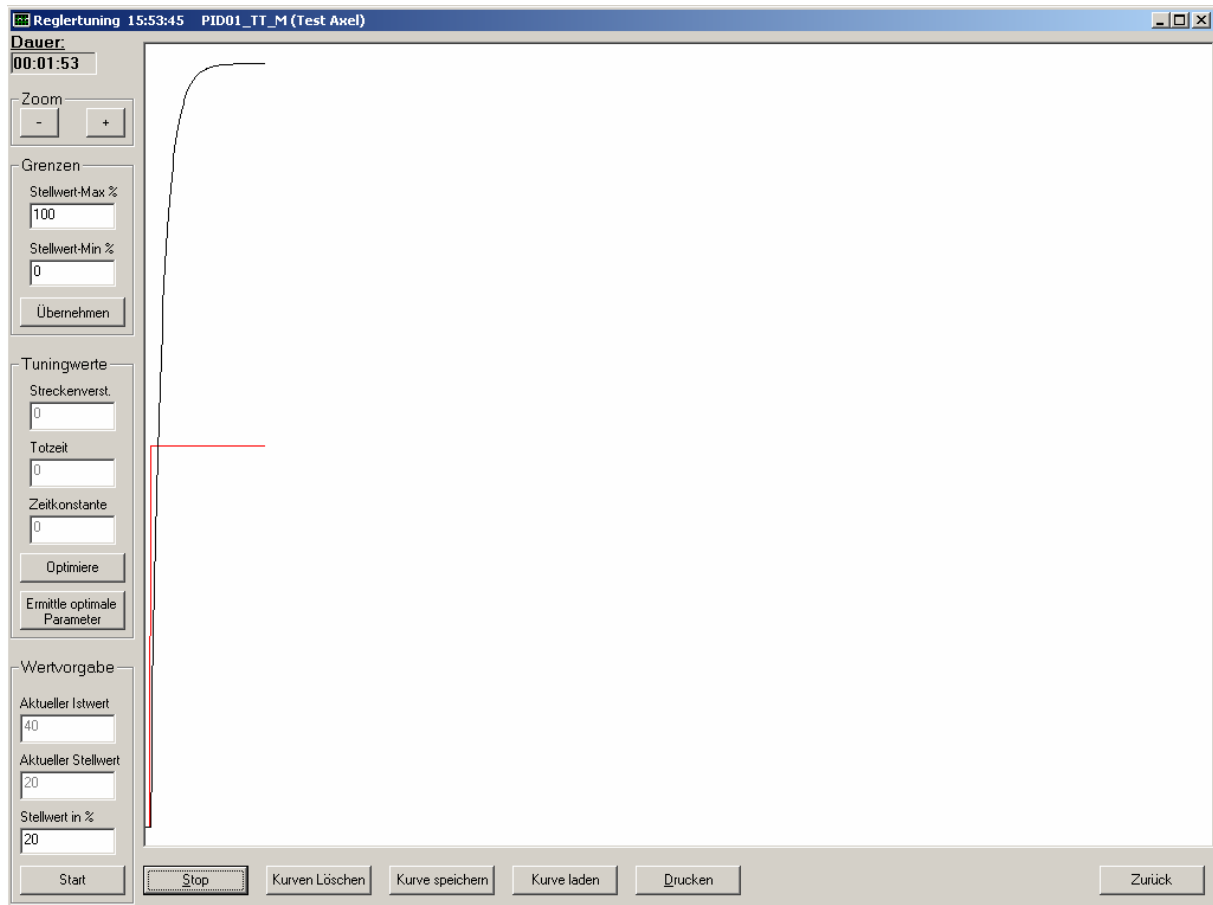


**Abbildung 5.14:** Der Frame5 zum Ändern des Stellwertes

Bevor der Anwender auf den Start-Button klickt, muss er einen Stellwert in der TextBox vorgeben. Dazu trägt der Anwender einen Stellwert von z.B. 20 % in die TextBox des Stellwertes ein und klickt dann auf den Start-Button. In diesem Augenblick wird der Wert aus den TextBoxen in die dafür vorgesehene Adresse in die Steuerung geschrieben.

Sobald dieser Stellwert in der Steuerung steht, werden die aktuellen Ist- und Stellwerte gleich wieder aus der Steuerung ausgelesen. Nach kurzer Zeit, je nachdem wann der Istwert des Prozesses beginnt sich zu ändern, sieht der Anwender eine Änderung des Istwertes in der TextBox und den dazu analogen Anstieg oder Abstieg der Sprungantwort. Die Aufzeichnung funktioniert genauso wie bei der Aufzeichnung der Sprungantwort auf einen Sollwertsprung in der Form frmReglerauswahl und wird deshalb nicht mehr weiter erläutert.

Der Unterschied besteht nur darin, dass hier der Regler vom System abgekoppelt und statt des Sollwertes der Stellwert vorgegeben wird.



**Abbildung 5.15:** Aufzeichnung der Sprungantwort der Regelstrecke

Die Aufzeichnung kann hier ebenfalls durch das Anhalten des Timers gestoppt werden. Dieses sollte aber erst vorgenommen werden, wenn der aufgezeichnete Istwert einige Zeit auf einem Wert bleibt. Erst dann kann man davon ausgehen, dass der Endwert erreicht wurde, um so die Streckenverstärkung zu bestimmen.



Wenn der maximale Istwert jetzt erreicht wird, kann der Anwender die Aufzeichnung durch den Stop-Button beenden. Dann können die Streckenparameter durch den Analysiere-Strecke-Button im Frame6 ermittelt werden. Es erfolgt dann die Ermittlung und die Berechnung der Streckenverstärkung, der Totzeit und der Zeitkonstante aus der CSV-Datei.

Im Programm werden zur Ermittlung ersteinmal die minimalen und maximalen Werte des Ist- und Stellwertes ermittelt. Damit ergibt sich die Streckenverstärkung:

$$K_S = \frac{\Delta x}{\Delta y} = \frac{\Delta \text{Istwert}}{\Delta \text{Stellwert}}$$

Die Totzeit ist die Differenz vom Beginn der Änderung des Stellwertes und vom Beginn der Änderung des Istwertes. Diese werden ermittelt und die Differenz wird dann aus der Spalte "A" der CSV-Datei ermittelt.

Die Zeitkonstante ist weniger einfach zu ermitteln. Sie ergibt sich bei dieser Strecke aus der Zeit vom Beginn der Änderung des Istwertes, bis zum Erreichen von 63% ihrer Sättigung bzw. ihres Maximalwertes, da es sich hierbei um eine Sättigungsfunktion handelt. Hier muss aber noch berücksichtigt werden, dass der Istwert zu Beginn der Aufzeichnung eine Temperatur von z.B. 20 °C haben kann. Das hat zu Folge, dass dieser Wert bei der Berechnung mit berücksichtigt werden muss. Wenn man also von einer Anfangstemperatur von z.B. 20 °C ausgeht und eine Temperatur bei einer Stellgrößenänderung von 60 °C erreicht, dann ergibt sich die Verzugszeit bei einem Wert von 63% der Differenz (60 °C – 20 °C) plus dem Anfangswert von 20 °C.

Es ist aber in den wenigsten Fällen davon auszugehen, dass genau dieser Wert in der CSV-Datei steht. Die Bedingung im Programm sieht vor, dass die Zeit einfach ausgelesen wird, wenn genau dieser Wert auch in der Spalte "B" der CSV-Datei steht. Wenn dieses nicht der Fall ist, dann wird in einer Schleife der nächste Wert ausgelesen, welcher größer ist als der benötigte. Nachdem dieser ausgelesen ist, wird auch der Wert ausgelesen, der in der Zeile davor steht. Die Zeiten dieser beiden Werte in der Spalte "A" werden auch ausgelesen, um dann die benötigte Zeit durch eine Linearisierung zu bestimmen. Daraufhin erscheinen die ermittelten Werte in den TextBoxen des Frame6.

Beispiel zur Ermittlung der Zeitkonstante  $T_S$ :

Gegeben:

$y(11)$  ist der Wert in Zeile 11, der größer ist als der benötigte Wert

$y(10)$  ist der Wert in Zeile 10, der kleiner ist als der benötigte Wert

$x(11)$  ist die Zeit für den Wert, der größer ist als der benötigte Wert

$x(10)$  ist die Zeit für den Wert, der größer ist als der benötigte Wert

$y(63\%)$  ist der Wert, an dem 63% der Differenz plus Anfangswert erreicht sind

Gesucht:

$x(63\%)$  ist die Zeit, an dem 63% der Differenz plus Anfangswert erreicht sind

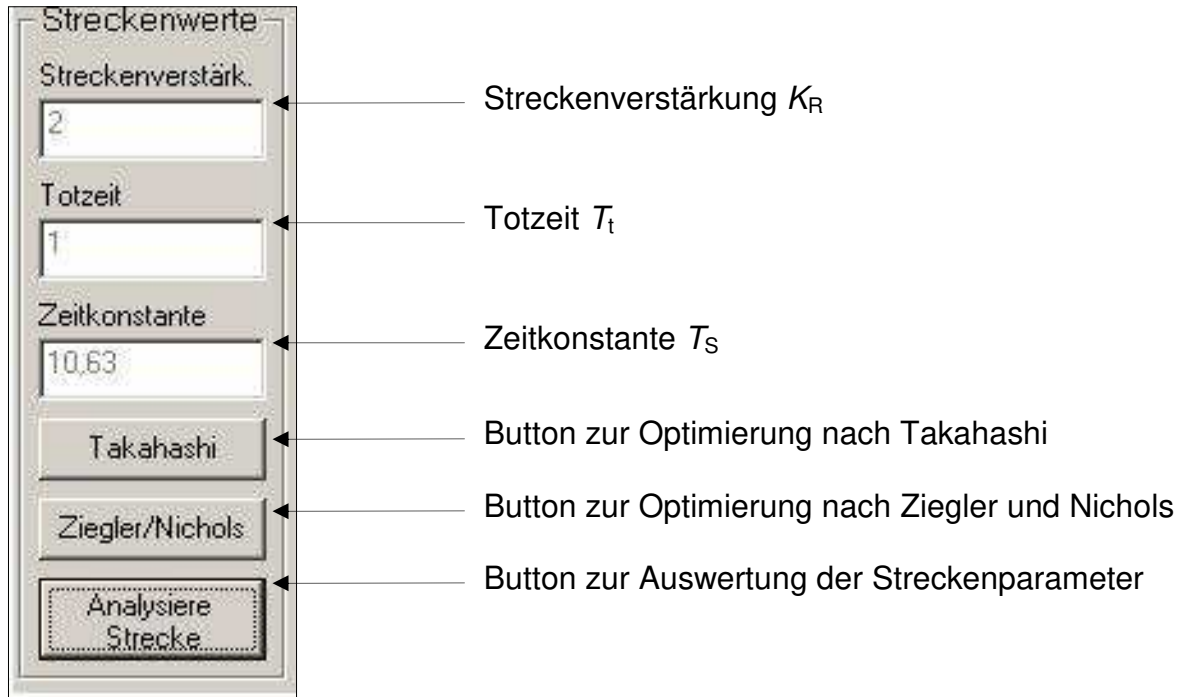
Berechnung:

Gradengleichung:  $y = mx + b$  mit  $b = 0$

$$m = \frac{y(11) - y(10)}{x(11) - x(10)}$$

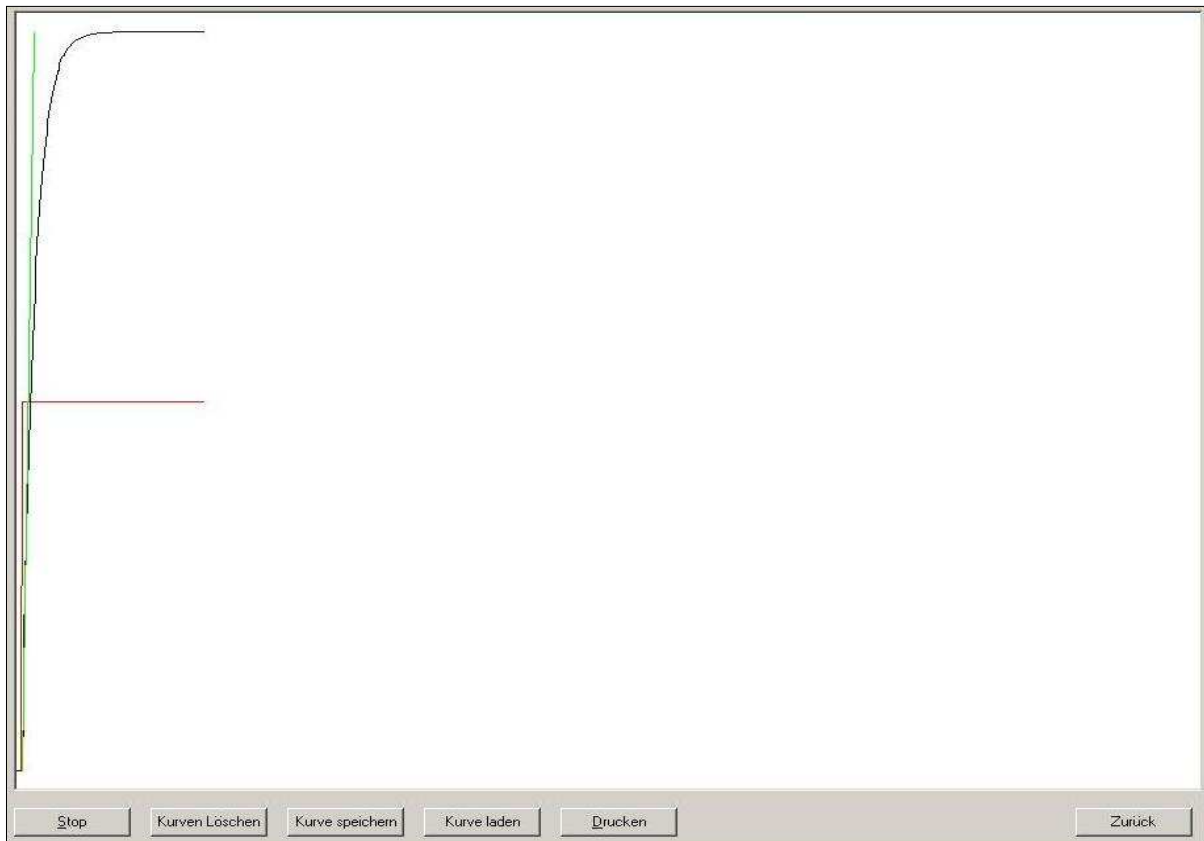
$$b = y(10) - m \cdot x(10)$$

$$x(63\%) = \frac{y(63\%) - b}{m}$$



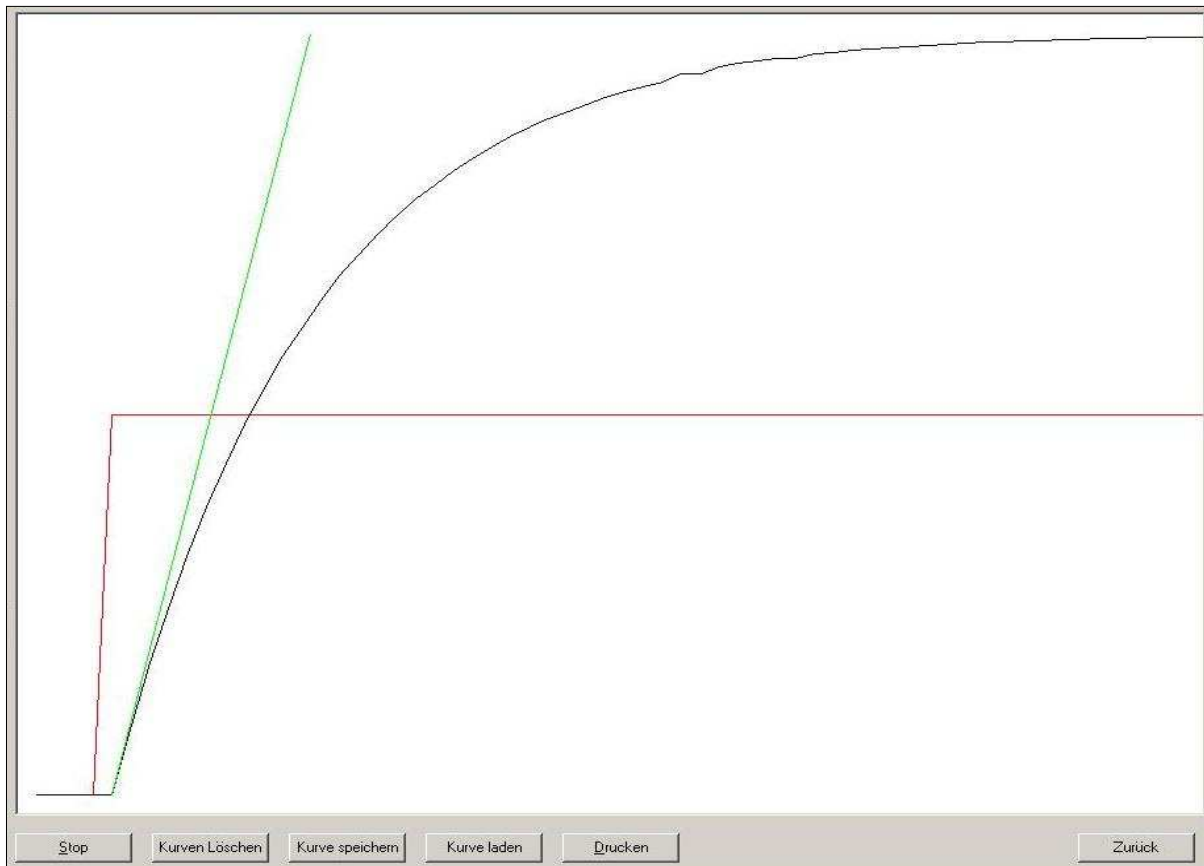
**Abbildung 5.16:** Der Frame6 zur Optimierung des Reglers

Gleichzeitig wird durch die ermittelten Parameter die Tangente an der Sprungantwort der Strecke angelegt, damit der Anwender auch sicher gehen kann, dass die Ermittlung der Parameter richtig gelaufen ist.



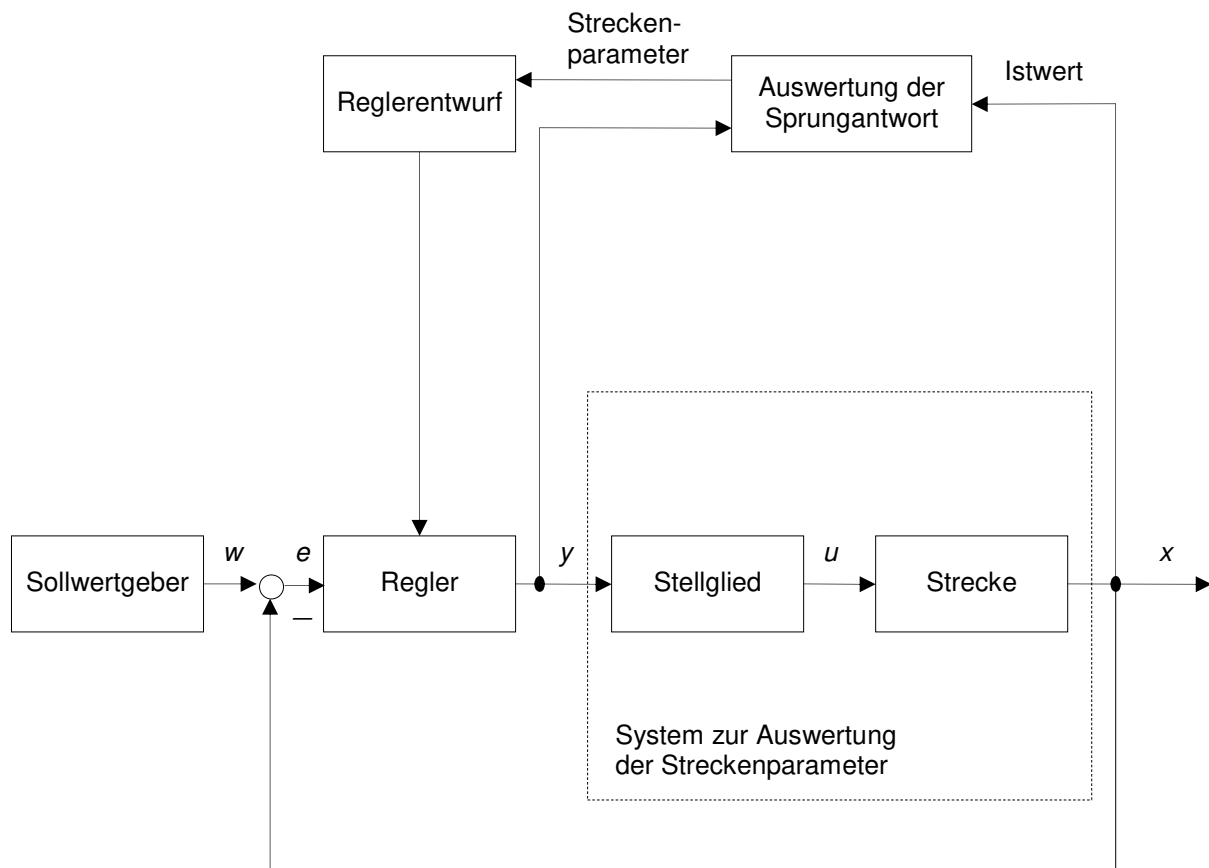
**Abbildung 5.17:** Anlegen der Tangente an der Sprungantwort

Da der zeitliche Bereich in x-Richtung ziemlich groß ist, da die Prozesse von sehr langer Dauer sein können, hat man hier die Möglichkeit die Aufzeichnung in x-Richtung zu zoomen. Die Plus- und Minus-Button sind oben links auf der Form zu erkennen und verdoppeln oder halbieren den Bereich, je nachdem, ob man das Bild vergrößern oder verkleinern möchte. Der Anwender kann somit die Aufzeichnung besser erkennen.



**Abbildung 5.18:** Zoom-Funktion in der PictureBox

Die ermittelten Werte können jetzt dazu genutzt werden, den Regler zu optimieren. Dazu hat der Anwender die Möglichkeit, das Verfahren nach Takahashi oder Ziegler und Nichols auszuwählen. Dazu klickt der Anwender auf einen der Buttons und die ermittelten Werte werden für die optimalen Parameter des Reglers, wie schon im Kapitel 4 erwähnt, berechnet und für den ausgewählten Regler in die Steuerung geschrieben. Der Ablauf zum Optimieren des Reglers ist in der folgenden Abbildung zu sehen.



**Abbildung 5.19:** Ablauf zur Regleroptimierung

In der Prozedur des Klick-Ereignisses wird allerdings noch eine Bedingung gestellt. Da bei der GEA TDS meistens kein D-Anteil des PID-Reglers genutzt wird, muss die Berechnung der optimalen Parameter den Regler wie einen PI-Regler behandeln. Dazu wird einfach eine Abfrage gestartet, die in der TextBox des D-Anteils in der Form frmReglerauswahl prüft, ob eine Null enthalten ist. Wenn eine Null in der TextBox steht, führt die Prozedur die Berechnung für einen PI-Regler durch, ansonsten für einen PID-Regler.

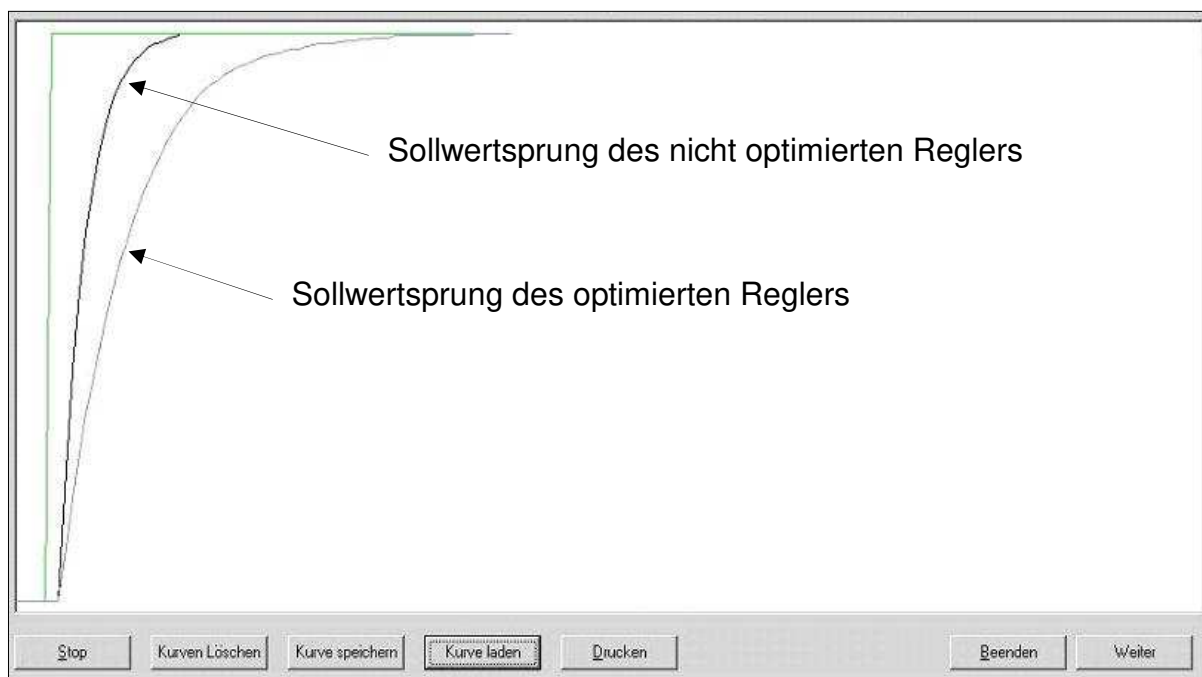
Nach dem Schreiben der optimalen Werte in die Steuerung erscheinen die Werte im Frame1 der Form frmReglerauswahl.

Die weiteren Buttons unter der PictureBox auf dieser Form werden nicht mehr behandelt, da sie die gleichen Funktionen haben wie die Buttons auf der Form frmReglerauswahl. Einzige Ausnahme ist der Zurück-Button, der den Anwender auf die vorherige Form bringt und den Regler wieder auf Automatik-Betrieb stellt.

## 5.4 Vergleich der Sprungantworten auf einen Sollwertsprung

Wenn der Anwender die optimalen Werte in die Steuerung übertragen hat, kann er die Sprungantworten des optimierten und nicht optimierten Reglers vergleichen. Voraussetzung ist, dass die Aufzeichnung des nicht optimierten Reglers vorher aufgezeichnet und abgespeichert wurde.

Dazu beginnt der Anwender, die Sprungantwort in der Form frmReglerauswahl mit den optimierten Werten aufzuzeichnen. Wenn der Sollwert erreicht ist, wird die Aufzeichnung gestoppt. Durch den Kurve-Laden-Button kann der Anwender die gespeicherte Kurve wieder in die PictureBox laden. Man sieht dann beide Aufzeichnungen in der PictureBox und kann diese vergleichen und auswerten.

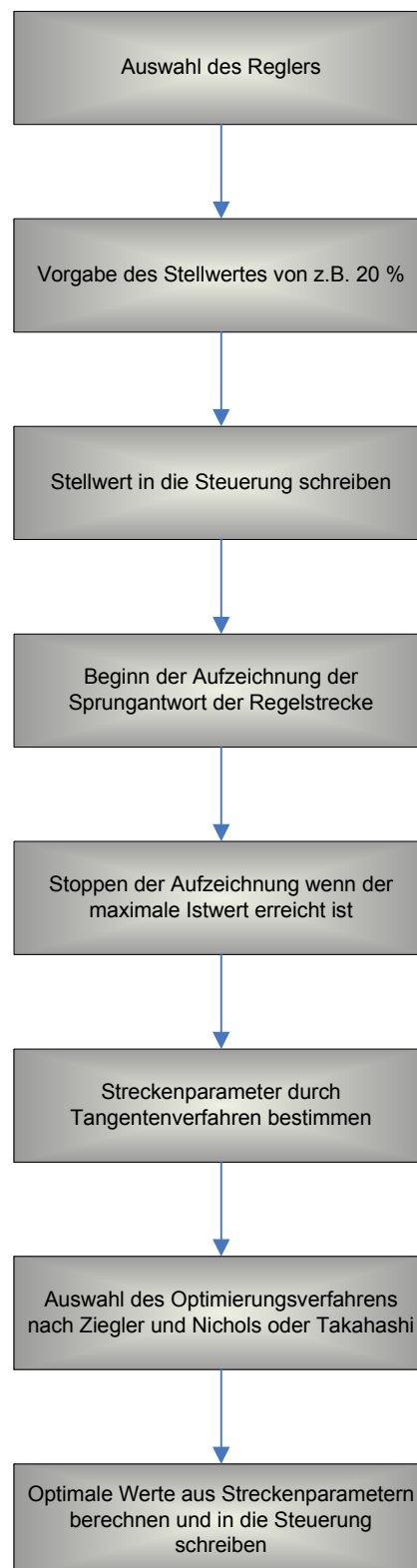


**Abbildung 5.20:** Vergleich der Sprungantworten

Es ist hier deutlich zu erkennen, dass der optimierte Regler ein wesentlich schnelleres Verhalten aufweist, als der nicht optimierte. Der Anwender sollte hier bei der Aufzeichnung immer den gleichen Sollwert vorgeben, da so der optische Vergleich der Sprungantworten besser ist.



Der Ablauf der Optimierung eines Reglers in der Steuerung lässt sich wie folgt beschreiben.



**Abbildung 5.21:** Ablaufdiagramm zur Regleroptimierung

## 6 Praktischer Test an der Anlage

Im vorherigen Kapitel wurde die Software zwar schon erläutert, doch bisher ist nur der Lead\_Lag-Baustein von Siemens, der eine PT1-Strecke simuliert, verwendet worden. Die Verwendung dieses Bausteins ist mit der Praxis aber nicht zu vergleichen, da die Strecke sehr schnell ist und eine Totzeit nur durch eine Verzögerung des Timers im Programm erzeugt wurde.

In diesem Kapitel ist der praktische Test eines Reglers für die Klimaanlage einer Molkerei in Uelzen beschrieben. Zum Vergleich werden hier beide Optimierungsverfahren angewendet, um einen Vergleich zu bekommen, welches Verfahren für dieses System am besten geeignet ist. Durch die Streckenanalyse ergaben sich folgende Werte:

$$K_S = 2,28$$

$$T_t = 7 \text{ Sekunden}$$

$$T_s = 50,91 \text{ Sekunden}$$

In der folgenden Abbildung sind die Aufzeichnungen mit optimalen Werten und mit der Anfangseinstellung zu sehen. Die Anfangseinstellung mit  $K_R = 0,5$  und  $T_i = 15$  Sekunden weist ein sehr langsames Verhalten auf. Durch die Ermittlung der Streckenparameter ergaben sich die folgenden optimalen Regelparameter:

Nach Ziegler und Nichols:

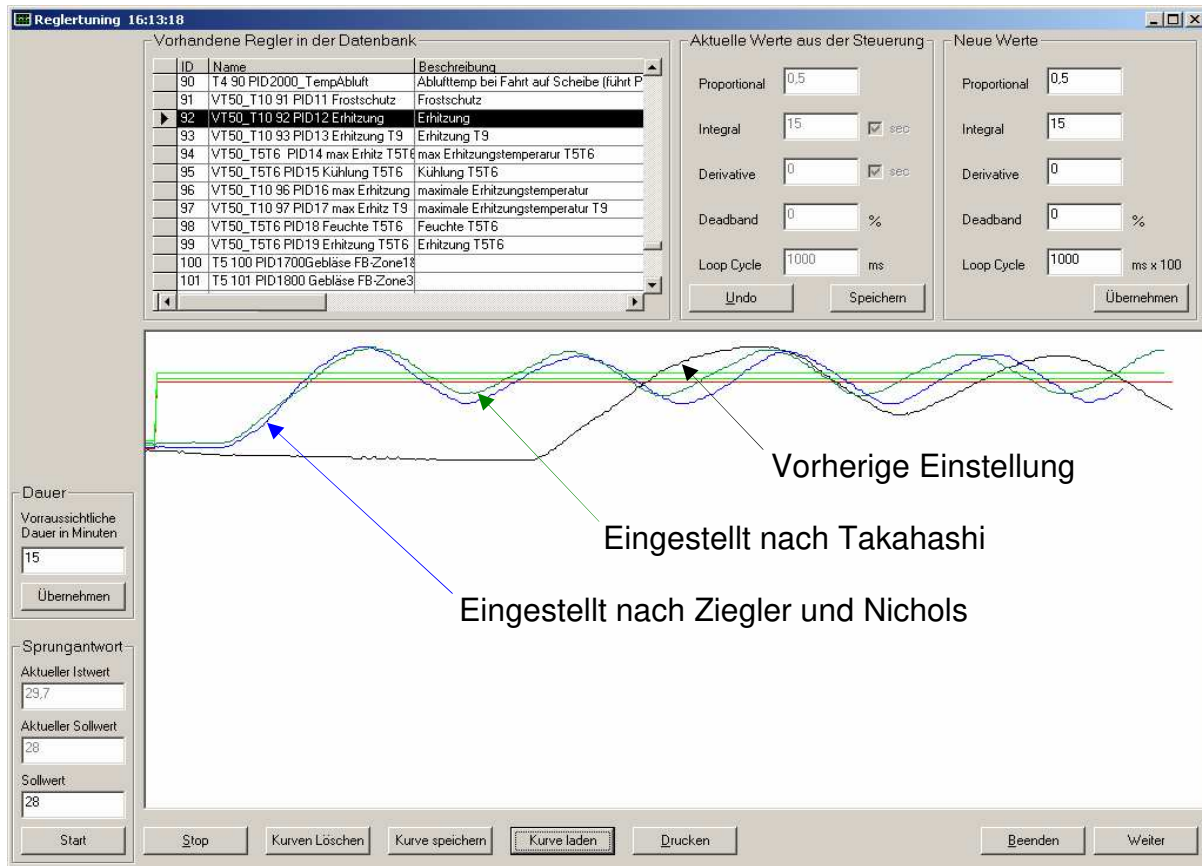
$$K_R = 2,9$$

$$T_i = 23 \text{ Sekunden}$$

Nach Takahashi:

$$K_R = 2,5$$

$$T_i = 25 \text{ Sekunden}$$



**Abbildung 6.1:** Vergleich der Sprungantworten an der Anlage

An der Abbildung lässt sich erkennen, dass die beiden Optimierungsverfahren den Regler zwar schneller machen, aber ebenfalls große Schwingungen im System verursachen. Der Versuch an anderen Reglern hat weiterhin ergeben, dass sich diese Optimierungsverfahren schlecht oder gar nicht an allen Prozessen, wie z.B. bei einem CIP-Prozess (Cleaning in place), der GEA TDS anwenden lassen.

## 7 Zusammenfassung

Ziel dieser Diplomarbeit war es, eine Software in Visual BASIC zu entwickeln, mit dem die PID-Regler in der Siemens S7 Steuerung nach bekannten Einstellregeln optimiert werden können. Folgende Unterziele sind Teil dieser Arbeit:

- Einsatz vorzugsweise an Durchfluss-, Niveau- und Temperaturreglern
- Log-File Generierung zur Nachverfolgung der Aktivitäten (Protokollierung der Parametersätze mit Historie)
- Undo-Funktionalitäten (Rückkehr zur alten Parametrierung)

Diese Ziele wurden mit der vorliegenden Arbeit erreicht. Die Ingenieure haben mit dieser Software die Möglichkeit, diese Regler einfach und schnell zu optimieren und dieses anhand von Aufzeichnungen der Sprungantworten auf einen Sollwertsprung auch zu erkennen.

## 8 Ausblick

Einige Vorschläge zur Erweiterung und Verbesserung sollen hier kurz erläutert werden:

- Da aus zeitlichen Gründen ausreichende Tests an der Anlage nicht möglich waren, sollte das Programm weitgehend getestet werden, um so den Nutzen der Software bei der GEA TDS sicher zu stellen
- Vorschläge zur weiteren Verbesserung der Regelkreise in typischen Einsatzsituationen bei GEA TDS (Auswertung zu Stellgliedern, Sensoren, Regelstrecke)
- Entwicklung einer Oberfläche (Front - End) zur Einbettung in eine Prozessvisualisierung (WinCC, Intouch, IFix) um Regleroptimierungen an der laufenden Anlage durchführen zu können
- Erweiterung des Programms für die Ermittlung von Streckenparametern bei Strecken > 1.Ordnung

## Literaturverzeichnis

- [1] **Kofler, Michael:** Visual Basic 6. Programmieretechniken, Datenbanken, Internet. Addison-Wesley-Longman, Bonn: 1989
- [2] **Monadjemi, Peter:** Visual Basic 6. Programmieren mit Windows 9x, NT und 2000. Markt + Technik Verlag, München: 2002
- [3] **Wendt, Lutz:** Taschenbuch der Regelungstechnik. Verlag Harry Deutsch, Frankfurt am Main: 2. überarbeitete und erweiterte Auflage von 1998
- [4] **Siemens:** SIMATIC Standard PID Control, Handbuch. Siemens AG: 2003
- [5] **Stähr, Matthias:** Entwicklung eines graphischen Frontends mit Visual BASIC 6 zur Chargenverfolgung in der Lebensmittelindustrie. Diplomarbeit Hochschule für Angewandte Wissenschaften, Hamburg: 2005.
- [6] **Unitec:** Integrated Supply Chain Management in Outsourcing  
<http://www.unitec.it/de/servizi/glossar18.php>  
Abruf am 04.08.2006
- [7] **Know-Library:** Knowledge Library - der deutschsprachigen Wissensbibliothek,  
[http://speicherprogrammierbare\\_steuerung.know-library.net/](http://speicherprogrammierbare_steuerung.know-library.net/)  
<http://steuerungstechnik.know-library.net/>  
Abruf am 04.08.2006

- [8] Wikipedia:** Wikipedia – die freie Enzyklopädie  
[http://de.wikipedia.org/wiki/OLE\\_for\\_Process\\_Control](http://de.wikipedia.org/wiki/OLE_for_Process_Control)  
Abruf am 04.08.2006  
<http://de.wikipedia.org/wiki/CSV-Datei>  
Abruf am 19.08.2006  
<http://de.wikipedia.org/wiki/PID-Regler>  
Abruf am 24.08.2006  
<http://de.wikipedia.org/wiki/Fliehkraftregler>  
Abruf am 06.09.2006
- [9] Lycos:** Lycos – meet you there  
<http://webmaster.lycos.de/glossary/S/>  
Abruf am 04.08.2006
- [10] RoboterNetz:** RoboterNetz – Roboter. Elektronik. Microcontroller  
[http://www.roboternetz.de/wissen/index.php/Regelungstechnik#Verz.C3.B6gerungsglied\\_1.Ordnung\\_.28PT1-Glied.29](http://www.roboternetz.de/wissen/index.php/Regelungstechnik#Verz.C3.B6gerungsglied_1.Ordnung_.28PT1-Glied.29)  
Abruf am 14.08.2006
- [11] Eurotherm:** Eurotherm – The Flexible Solutions Group  
[http://www.eurotherm.de/index.php?navi=support&site=support&navi2=faq&site2=faq\\_reglerpara#33](http://www.eurotherm.de/index.php?navi=support&site=support&navi2=faq&site2=faq_reglerpara#33)  
Abruf am 21.08.06
- [12] Haderer, Ralf:** Skript Regelungstechnik FH NON  
Universität Lüneburg, Fakultät III - Umwelt & Technik,  
Bereich Automatisierungstechnik, Lüneburg: 2003.

## Abkürzungsverzeichnis

A/D	Analog / Digital
BOF	Beginning Of File
CLI	Call Level Interface
CIP	Cleaning in place
CPU	Central Processing Unit
CSV	Character Separated Values oder Comma Separated Values
D/A	Digital / Analog
DB	Datenbaustein
DCOM	Distributed Component Object Model
DDC	Direct Digital Control
DISV	Disturbance Variable = Störgröße
DSN	Data Source Name
E/A	Eingänge / Ausgänge
EOF	End Of File
EVA	Eingabe-, Verarbeitungs- und Ausgabeteil
FB	Funktionsbaustein
FC	Funktion
ID	Identifikation
LAN	Local area network
LMN	Manipulated Value = Stellwert
LMNFC	FC-Aufruf im Stellwertzweig
LP_SCHED	Loop Scheduler
MAN	Manuel Value
MPI	Multiple protocol interface
OB	Operationsbaustein
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OP	Operating panel
OPC	Openness, Productivity, Collaboration
OS	Operating system
PG	Programmiergerät



PI	Proportional, Integral
PID	Proportional, Integral, Differenzial
PID_CP	Kontinuierlicher Regler
PID_ES	Schrittregler
PLC	programmable logic controller
PT1	Proportional-Element mit Verzögerung I. Ordnung
PV	Process Variable = Istwert
PVFC	FC-Aufruf im Istwertzweig
QNEG_P	Impulsausgang (Negativ)
QPOS_P	Impulsausgang (Positiv)
SP	Setpoint = Sollwert
SPFC	FC-Aufruf im Sollwertzweig
SPS	Speicherprogrammierbare Steuerung
SQL	Structured Query Language
STI	Selectable Timer Interrupt
UHT	Ultra-High-Temperature

## Formelzeichenverzeichnis

$e$	Regeldifferenz = Regelabweichung
$K_R$	Reglerverstärkung
$K_S$	Proportionalfaktor = Übertragungsbeiwert
$T$	Abtastzeit
$T_g$	Ausgleichszeit
$T_N$	Nachstellzeit
$T_S$	Zeitkonstante
$T_t$	Totzeit
$T_u$	Verzugszeit
$T_V$	Vorhaltezeit
$u$	Ausgangsvariable
$w$	Führungsgröße = Sollwert
$x$	Regelgröße = Istwert
$y$	Reglerausgangsgröße = Stellgröße
$z$	Störgröße

## **Inhalt der CD-ROM**

- Diplomarbeit
- Handbuch SIMATIC Standard PID Control

**Anhang****Programmquellcode****Quellcode in frmVerbindung**

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
```

```
-----
' Source Code Administration Information
```

```
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
```

```
-----
' File description
```

```
' PROGRAM: Reglertuning
```

```
' FILENAME: frmVerbindung.frm
```

```
' FUNCTIONS: - Datenbankverbindung öffnen
```

```
' COMMENTS:
```

```
' MODIFICATIONS:
```

```
-----Beim Klick die Datenbankverbindung öffnen
```

```
Private Sub cmdVerbinden_Click()
```

```
    On Error GoTo ErrorHandler
```

```
    If cboDSNList.Text = "" Then
```

```
        MsgBox "Bitte eine Verbindung auswählen!", vbCritical, "Verbindungsfehler"
```

```
        Exit Sub
```

```
    End If
```

```
    Me.Hide
```

```
Exit Sub
```

```
ErrorHandler:
```

```
MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"
```

```
ErrorLog Err.Number, Err.Description, "FrmVerbindung: cmdVerbinden_Click"
```

```
Resume Next
```

```
End Sub
```

```
-----
Private Sub Form_Load()
```

```
    On Error GoTo ErrorHandler
```

```
    GetDSNsAndDrivers
```

```
    cboDSNList.ListIndex = 0
```

```
Exit Sub
```

```
ErrorHandler:
```

```
ErrorLog Err.Number, Err.Description, "FrmVerbindung: Form_Load"
```

```
Resume Next
```

---

End Sub

---

```
Public Sub GetDSNsAndDrivers()
Dim i As Integer
Dim sDSNItem As String * 1024
Dim sDRVItem As String * 1024
Dim sDSN As String
Dim sDRV As String
Dim iDSNLen As Integer
Dim iDRVLen As Integer
Dim IHenv As Long           'Zugriffsnummer zur Umgebung

On Error Resume Next
cboDSNList.AddItem "(None)"

'DSNs abrufen
If SQLAllocEnv(IHenv) <> -1 Then
Do Until i <> SQL_SUCCESS
sDSNItem = Space(1024)
sDRVItem = Space(1024)
' Aufruf der API Funktion zur Ausgabe der nächsten ODBC Verbindung
i = SQLDataSources(IHenv, SQL_FETCH_NEXT, sDSNItem, 1024, iDSNLen, sDRVItem, 1024, iDRVLen)
sDSN = Left(sDSNItem, iDSNLen)
sDRV = Left(sDRVItem, iDRVLen)

If sDSN <> Space(iDSNLen) Then
cboDSNList.AddItem sDSN
End If
Loop
End If
End Sub
```

## Quellcode in frmReglerbearbeitung

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
'
'-----
' Source Code Administration Information
'
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
'
'-----
' File description
'
' PROGRAM:  Reglertuning
'
' FILENAME: frmReglerauswahl.frm
'
' FUNCTIONS: - Datenbankinhalte anzeigen
'            - Regler auswählen
'            - Daten des ausgewählten Reglers aus der Steuerung lesen und anzeigen
'            - Regler Pparametrieren durch schreiben von Werten in die Steuerung
'            - Anfangswerte speicher und wieder lesen
'            - Sollwertsprung mit Sprungantwort aufzeichnen
'            - Aufzeichnung stoppen
'            - Aufzeichnung speichern und laden
'            - Aufzeichnung drucken
'            - Programm beenden
'
' COMMENTS:
'
' MODIFICATIONS:
'
'-----
Option Explicit

Dim ExtSPOn As Boolean
Dim FixedOutputOn As Boolean
Dim LoopOn As Boolean

'Deklaration der Adressen für die Werte in der Steuerung
Dim Adresse1 As String, Adresse2 As String, Adresse3 As String, _
    Adresse4 As String, Adresse5 As String, Adresse6 As String, _
    Adresse7 As String, Adresse8 As String, Adresse9 As String, _
    Adresse10 As String, Adresse11 As String, Adresse12 As String, _
    Adresse13 As String, Adresse14 As String, Adresse15 As String, _
    Adresse16 As String, Adresse17 As String

'-----Für TextBox nur Zahlen zulassen

Private Sub txtGainNeu_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case 188, 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtGainNeu = "0" Then txtGainNeu = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtGainNeu = Left(txtGainNeu, Len(txtGainNeu) - 1)
    End Select
End Sub
```

---

'-----Für TextBox nur Zahlen zulassen

```
Private Sub txtTINeu_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtTINeu = "0" Then txtTINeu = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtTINeu = Left(txtTINeu, Len(txtTINeu) - 1)
    End Select
End Sub
```

---

'-----Für TextBox nur Zahlen zulassen

```
Private Sub txtTDNeu_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtTDNeu = "0" Then txtTDNeu = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtTDNeu = Left(txtTDNeu, Len(txtTDNeu) - 1)
    End Select
End Sub
```

---

'-----Für TextBox nur Zahlen zulassen

```
Private Sub txtDeadbandNeu_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtDeadbandNeu = "0" Then txtDeadbandNeu = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtDeadbandNeu = Left(txtDeadbandNeu, Len(txtDeadbandNeu) - 1)
    End Select
End Sub
```

---

'-----Für TextBox nur Zahlen zulassen

```
Private Sub txtLoopCycleNeu_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtLoopCycleNeu = "0" Then txtLoopCycleNeu = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtLoopCycleNeu = Left(txtLoopCycleNeu, Len(txtLoopCycleNeu) - 1)
    End Select
End Sub
```

---

---

```
'-----Für TextBox nur Zahlen zulassen
```

```
Private Sub txtSollwert_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 188, 189, 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeySubtract, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8,
            vbKeyNumpad9
            If txtSollwert = "0" Then txtSollwert = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 And vbKeySubtract And vbKeyDecimal Then txtSollwert = Left(txtSollwert,
                Len(txtSollwert) - 1)
            End Select
    End Sub
```

---

```
'-----Beim Laden der Form die Daten aus Datenbank holen
```

```
Public Sub Form_Load()
    Dim strPath$, strDataFile$

    On Error GoTo ErrorHandler

    Picture1.AutoRedraw = True
    Picture1.ScaleMode = 3
    frmReglerbearbeitung.Picture1.AutoRedraw = True
    frmReglerbearbeitung.Picture1.ScaleMode = 3

    'CSV-Datei Test.csv löschen
    strPath = App.Path & "\CSV-Dateien\"
    strDataFile = strPath & "Test.csv"

    On Error Resume Next
    fso.DeleteFile strDataFile
    On Error GoTo 0

    SelChangeTrueFalse = False

    ReDim Field(3, 1)

    strPath = App.Path
    If Right$(strPath, 1) <> "\" Then strPath = strPath & "\" 'Anwendungspfad ermitteln

    With rs
        .ActiveConnection = cn 'Aktive Verbindung benennen
        .CursorLocation = adUseClient
        .Source = "SELECT lid AS ID,sname_1 AS Name,sdescription_1 AS Beschreibung," & _
            "dSPLowLimit, dSPHighLimit FROM tblPIDItem"
        .Open 'Recordset laden
    End With

    Set DataGrid1.DataSource = rs 'DataGrid1 als Quelle den Recordset zuweisen

    With DataGrid1
        .Columns("ID").Width = 400 'DataGrid formatieren
        .Columns("Beschreibung").Width = 4000 'Spalte verkleinern
        .Columns("dSPLowLimit").Visible = False 'Spalte vergrößern
        .Columns("dSPHighLimit").Visible = False
    End With

    Exit Sub
ErrorHandler:

    MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
        vbCritical, "Fehler"

    ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: Form_Load"
    Resume Next

End Sub
```

---



'-----Werte aus Steuerung für den ausgewählten Regler anzeigen

```

Public Sub DataGrid1_SelChange(cancel As Integer)
Dim i As Integer
Dim SQL1 As String
Dim SQL2 As String
Dim SQL3 As String
Dim SQL4 As String
Dim Result As Long

    On Error GoTo ErrorHandler

    StartTrueFalse = False
    StartTrueFalse = False
    Start2TrueFalse = False
    SelChangeTrueFalse = True
    StopTrueFalse = True

    On Error Resume Next
    'Regler auf Anfangswerte zurücksetzen
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".2", "1", ExtSPOn)
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", FixedOutputOn)
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".4", "1", LoopOn)
    On Error GoTo 0

    Set rs2 = New ADODB.Recordset

    SQL1 = "SELECT IControllerID FROM tblPIDItem WHERE IID=" & DataGrid1.SelBookmarks.Item(0)
    SQL2 = "SELECT IID FROM tblPIDItem WHERE IcontrollerID=(" & SQL1 & ") ORDER BY IID"
    SQL3 = "SELECT sOPCAccessPath FROM tblControllers," & _
        "tblPIDItem WHERE tblControllers.IID=tblPIDItem.IControllerID AND tblPIDItem.IID=" &
        DataGrid1.SelBookmarks.Item(0)

    Set rs3 = New ADODB.Recordset

    SQL4 = "SELECT tblAnalogItem.IPrecision FROM tblAnalogItem,tblPIDItem WHERE" &
        "tblAnalogItem.IID=tblPIDItem.IMeasuredVariableID AND tblPIDItem.IID =" & DataGrid1.SelBookmarks.Item(0)

    Timer1.Enabled = False

    With rs2
        .ActiveConnection = cn
        .CursorLocation = adUseClient
        .Source = SQL2
        .Open
    End With

    With rs3
        .ActiveConnection = cn
        .CursorLocation = adUseClient
        .Source = SQL4
        .Open
    End With

    Precision = rs3.Fields(0)

    rs3.Close

    pos = -1
    i = 1

    Do
        If rs2.Fields(0) = DataGrid1.SelBookmarks.Item(0) Then pos = i
        i = i + 1
        rs2.MoveNext
    Loop Until rs2.EOF Or pos <> -1

    rs2.Close
    rs2.Open SQL3, cn, adOpenStatic, adLockReadOnly
    Adresse1 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 2
    Adresse2 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 4
    Adresse3 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 6
    Adresse4 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 8
    Adresse5 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 10
    Adresse6 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 12

```

'Sollwert Intern

```

Adresse7 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 18 'Stellwertgröße Min
Adresse8 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 20 'Stellwertgröße Max
Adresse9 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 22
Adresse10 = rs2.Fields(0) & "DB6,TIME" & (pos - 1) * 100 + 24
Adresse11 = rs2.Fields(0) & "DB6," & (pos - 1) * 100 + 32 & ".2"
Adresse12 = rs2.Fields(0) & "DB6," & (pos - 1) * 100 + 32 & ".3" 'Festwert
Adresse13 = rs2.Fields(0) & "DB6," & (pos - 1) * 100 + 32 & ".4" 'Wenn Regler ein, dann 1
Adresse14 = rs2.Fields(0) & "DB6," & (pos - 1) * 100 + 33 & ".6"
Adresse15 = rs2.Fields(0) & "DB6," & (pos - 1) * 100 + 33 & ".7"
Adresse16 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 14
Adresse17 = rs2.Fields(0) & "DB6,INT" & (pos - 1) * 100 + 16

mstrOPCConnection = Mid(rs2.Fields(0), 5, Len(rs2.Fields(0)) - 5)
rs2.Close

Call OPC_Connect(Adresse1, Adresse2, Adresse3, Adresse4, Adresse5, Adresse6, Adresse7, Adresse8, _
Adresse9, Adresse10, Adresse11, Adresse12, Adresse13, Adresse14, Adresse15, Adresse16, _
Adresse17)

On Error Resume Next

SollLowLimit = DataGrid1.Columns.Item(3).Value
SollHighLimit = DataGrid1.Columns.Item(4).Value
'Speichern der Anfangswerte
Call ReadFromPLC1
ExtSPOn = TempArray2(11)
FixedOutputOn = TempArray2(12)
LoopOn = TempArray2(13)

On Error GoTo 0

Call ReadFromPLC1
If TempArray2(14) = True Then
    MsgBox "Tuning nicht möglich, da dieser Regler gerade läuft! Bearbeiten Sie einen anderen Regler oder versuchen Sie _
    es später.", vbInformation, "Tuning nicht möglich"
Else

Timer1.Interval = 1000
Timer1_Timer
Timer1.Enabled = True

Call LogFileGeneration("Datum und Uhrzeit: " & vbTab & Now() & vbCrLf _
& "=====") _
& vbCrLf & "Reglername: " & vbTab & vbTab & DataGrid1.Columns.Item(1).Value _
& vbCrLf & "Reglerbeschreibung: " & vbTab & DataGrid1.Columns.Item(2).Value _
& vbCrLf & vbCrLf & "Anfangswerte: " & vbTab & vbTab & "KR = " & txtGain.Text _
& vbCrLf & vbTab & vbTab & vbTab & "TI = " & txtTI.Text & vbCrLf & vbTab & vbTab _
& vbTab & "TD = " & txtTD.Text & vbCrLf & vbTab & vbTab & vbTab & "Deadband = " _
& txtDeadband.Text & vbCrLf & vbTab & vbTab & vbTab & "Cycle_Time = " _
& txtLoopCycle.Text & " ms * 100")

End If

Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: DataGrid1_SelChange"
Resume Next

End Sub

```

'-----Bei Klick auf Start-Button den Timer2 aufrufen und starten

```

Private Sub cmdStart_Click()
Dim Result As Long

On Error GoTo ErrorHandler

strDataFile = App.Path & "CSV-Dateien" & "Test.csv"

If txtSollwert.Text < SollLowLimit Or txtSollwert.Text > SollHighLimit Then
    Result = MsgBoxT(Me, "Der Bereich darf nur zwischen " & Chr$(34) & SollLowLimit & _
    Chr$(34) & " und " & Chr$(34) & SollHighLimit & Chr$(34) & " liegen!", vbInformation, "Wert nicht im gültigen Bereich")

```

```

Else
    If SelChangeTrueFalse = True Then
        Start2TrueFalse = True

        txtSollwertAktuell.Text = TempArray2(5) / 10

        'Verzögerung der Aufzeichnung
        Timer_Start.Interval = 3000
        Timer_Start.Enabled = True

        'Regler auf Automatikbetrieb stellen
        Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".2", "1", False)
        Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", False)
        Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".4", "1", True)

        Timer3.Interval = 1000
        Timer3.Enabled = True
        Timer3_Timer

    Else
        Result = MsgBoxT(Me, "Sie müssen erst einen Regler aus dem DataGrid auswählen!", _
            vbInformation, "Regler auswählen")

    End If

    Call LogFileGeneration("Beginn der Aufzeichnung! " & Time & vbCrLf _
        & "Sollwertsprung" & vbCrLf & " = " & txtSollwert.Text)

    End If

    StopTrueFalse = False

Exit Sub
ErrorHandler:
MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdStart_Click"
Resume Next

End Sub

```

---

```

'-----Verzögerungstimer, um Aufzeichnung nach 3 Sekunden zu beginnen
Private Sub Timer_Start_Timer()
    Timer_Start.Enabled = False
    If Not txtSollwert = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 _
        + 12, "1", txtSollwert.Text * 10)

End Sub

```

---

```

'-----Wenn Timer3 kommt ReadFromPLC, Trendview_show und csv_WriteSollwert aufrufen
Private Sub Timer3_Timer()
    Call ReadFromPLC1
    Call csv_WriteSollwert
    Call frmReglerbearbeitung.TrendView_show

End Sub

```

---

```

'-----Reglerdaten in CSV-Datei schreiben
Public Sub csv_WriteSollwert()
    Dim i As Long
    Dim fno%, strPath$, strDataFile$, strDataText$

    On Error GoTo ErrorHandler

```

```

X_Zaehler = X_Zaehler + 1

If Not Field(1, 1) = "" Then
    i = UBound(Field(), 2)
    ReDim Preserve Field(3, i + 1)
End If

Field(1, UBound(Field(), 2)) = X_Zaehler
Field(2, UBound(Field(), 2)) = TempArray2(5) / 10
Field(3, UBound(Field(), 2)) = TempArray2(6) / 10

strPath = App.Path & "\CSV-Dateien\"
strDataFile = strPath & "Test.csv"
strDataText = X_Zaehler & ";" & TempArray2(5) / 10 & ";" & TempArray2(6) / 10

fno = FreeFile
Open strDataFile For Append As #fno
Print #fno, strDataText
Close #fno

Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: csv_WriteSollwert"
Resume Next

End Sub

```

---

'-----Bei Klick auf Stop-Button den Timer anhalten

```

Public Sub cmdStop1_Click()
Dim Result As Long

    On Error GoTo ErrorHandler

    StopTrueFalse = True
    Result = MsgBoxT(Me, "Möchten Sie die Aufzeichnung wirklich stoppen?", _
        vbYesNo + vbQuestion + vbCritical, "Aufzeichnung stoppen")

    If Result = vbYes Then
        Call LogFileGeneration("Aufzeichnung Sollwertsprung gestoppt: " & Now _
            & vbCrLf & "Istwertsprung" & vbTab & " = " & txtIstwert.Text)
        Timer3 = False
    Else

    End If

Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdStop1_Click"
Resume Next

End Sub

```

---

'-----CSV-Datei durch CommonDialog speichern

```

Public Sub cmdKurveSpeichern_Click()
Dim Reglername As String, i As Integer
Static j&
Dim iFile%, strDataText$, Result&

    j = j + 1
    Reglername = "Sollwertsprung, " & frmReglerauswahl.DataGrid1.Columns(2)

    CommonDialog1.CancelError = True
    CommonDialog1.DialogTitle = "Datei speichern"
    CommonDialog1.InitDir = App.Path & "\CSV-Dateien\"
    CommonDialog1.Filter = "(*.csv)|*.csv"
    CommonDialog1.FileName = Reglername

    On Error Resume Next

```

```

CommonDialog1.ShowSave

If Err = 0 Then

    iFile = FreeFile
    Open CommonDialog1.FileName For Append As #iFile

    If FileLen(CommonDialog1.FileName) > 0 Then

        Result = MsgBox("Die Datei " & Chr$(34) & Reglername & ".csv" & Chr$(34) & " existiert bereits!" & _
            " Möchten Sie die existierende Datei ersetzen?", _
            vbYesNo, "Ersetzen von Dateien bestätigen")

        If Result = vbYes Then

            Close #iFile
            Open CommonDialog1.FileName For Output As #iFile
            fso.DeleteFile strDataFile           'Datei Test.csv löschen

        Else
            Exit Sub
        End If
    End If

    For i = 1 To UBound(Field(), 2)
        Print #iFile, Field(1, i) & ";" & Field(2, i) _
            & ";" & Field(3, i)
    Next i
    Close #iFile

End If

End Sub

```

---

#### '-----CSV-Datei durch CommonDialog in PictureBox laden

```

Public Sub cmdKurveLaden_Click()
    Dim i As Integer
    Dim iFile%, strDataText$
    Dim L() As String
    Dim X() As Double
    Dim Y() As Double
    Dim y2() As Double
    Dim v() As String
    Dim fno, s As String
    Dim xmin As Double, xmax As Double, ymin As Double, ymax As Double

    On Error GoTo ErrorHandler

    CommonDialog1.CancelError = True
    CommonDialog1.DialogTitle = "Datei laden"
    CommonDialog1.InitDir = App.Path & "\CSV-Dateien\"
    CommonDialog1.Filter = "(*.csv)*.csv"

    On Error Resume Next
    CommonDialog1.ShowOpen

    strDataFile = CommonDialog1.FileName

    fno = FreeFile
    Open CommonDialog1.FileName For Binary As #fno
    s = Space(LOF(fno))
    Get #fno, , s
    Close #fno

    L() = Split(s, vbCrLf)
    ReDim X(1 To UBound(L))
    ReDim Y(1 To UBound(L))
    ReDim y2(1 To UBound(L))

    For i = 0 To UBound(L)

        v() = Split(L(i), ";")
        If UBound(v) = 2 Then
            X(i + 1) = v(0)
            Y(i + 1) = v(1)

```

```

        y2(i + 1) = v(2)
    End If
Next i

For i = 1 To UBound(X)

    If X(i) > xmax Then
        xmax = X(i)
    End If

    If X(i) < xmin Then
        xmin = X(i)
    End If

    If Y(i) > ymax Then
        ymax = Y(i)
    End If

    If Y(i) < ymin Then
        ymin = Y(i)
    End If

Next i

CommonDialog1.ShowColor                'Farbauswahl für Graphen anzeigen

Picture1.Scale (1, ymax + 1)-(1000, ymin - 1)    'Start und Ende der Skala festlegen
Picture1.ForeColor = CommonDialog1.Color

Picture1.PSet (X(1), y2(1))
For i = 2 To UBound(y2) - 1
Picture1.Line -(X(i), y2(i)), vbGreen
Next i

Picture1.PSet (X(1), Y(1))
For i = 2 To UBound(Y) - 1
Picture1.Line -(X(i), Y(i))
Next i

Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdKurveLaden_Click"
Resume Next

End Sub

```

---

'-----Anzeige in Picture1 löschen und Timer3 stoppen

```

Private Sub cmdKurvenLöschen_Click()
Dim Result As Long
Dim strPath$, strDataFile$

    On Error GoTo ErrorHandler

    strPath = App.Path & "\CSV-Dateien\"
    strDataFile = strPath & "Test.csv"

    'Messagebox öffnen
    Result = MsgBoxT(Me, "Möchten Sie die Kurven wirklich löschen?", _
        vbYesNo + vbQuestion + vbCritical, "Kurven löschen")

    'Wenn yes gewählt wird, Timer ausschalten und PictureBox säubern
    If Result = vbYes Then

        Timer3 = False
        Picture1.Cls
        On Error Resume Next
        fso.DeleteFile strDataFile

    'ansonsten nichts machen und Messagebox schließen
    Else

    End If

X_Zaehler = 0

```

```
Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdKurvenLöschen_Click"
Resume Next

End Sub
```

---

```
'-----Graphen in Picture1 drucken
```

```
Public Sub cmdPrint_Click()
Dim BeginPage, EndPage, NumCopies, Orientation
Dim i As Integer

    On Error GoTo ErrorHandler

'Dialogfeld "Drucken" anzeigen
CommonDialog2.CancelError = True           'Cancel auf True setzen
On Error GoTo ErrorHandler

cmdWeiter.SetFocus
CommonDialog2.ShowPrinter                   'Dialogfeld "Drucken" anzeigen

'Benutzerdefinierte Werte aus dem Dialogfeld abrufen
BeginPage = CommonDialog2.FromPage
EndPage = CommonDialog2.ToPage
NumCopies = CommonDialog2.Copies
Orientation = CommonDialog2.Orientation = cdlPortrait

For i = 1 To NumCopies
    Printer.PaintPicture Picture1.Image, 0, 0    'Drucken
    Printer.EndDoc
Next
Exit Sub
```

```
Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: csv_WriteSollwert"
Resume Next

End Sub
```

---

```
'-----Wenn Timer1 kommt csv_WriteAll aufrufen und Werte aus der Steuerung auslesen
```

```
Private Sub Timer1_Timer()

    On Error Resume Next

    Call ReadFromPLC1

    txtGain.Text = TempArray2(1) / 10
    txtTI.Text = TempArray2(2)
    txtTD.Text = TempArray2(3)
    txtDeadband.Text = TempArray2(4)
    txtLoopCycle.Text = TempArray2(10)

    If TempArray2(14) = True Then
        Check1 = 1
    Else
        Check1 = False
    End If

    If TempArray2(15) = True Then
        Check2 = 1
    Else
        Check2 = False
    End If

    frmReglerbearbeitung.txtIstwert = TempArray2(5) / 10 ^ Precision
    frmReglerbearbeitung.txtStellwertAktuell = TempArray2(9) / 10
```

```

txtSollwertAktuell = TempArray2(6) / 10
txtIstwert = TempArray2(5) / 10

If Time <> frmReglerauswahl.Caption Then
    frmReglerauswahl.Caption = "Reglertuning" & " " & Time
    frmReglerbearbeitung.Caption = "Reglertuning" & " " & Time & " " & _
    DataGrid1.Columns.Item(1).Value & " (" & DataGrid1.Columns.Item(2).Value & ")"
End If

On Error GoTo 0

```

```
End Sub
```

---

'-----TextBoxen alle 10 Sekunden aktualisieren

```

Private Sub Timer2_Timer()

    Call ReadFromPLC1
    frmReglerbearbeitung.txtStellwertMin.Text = TempArray2(7) / 10
    frmReglerbearbeitung.txtStellwertMax.Text = TempArray2(8) / 10

```

```
End Sub
```

---

'-----Neue Parameter übernehmen und in die Steuerung schreiben

```

Private Sub cmdÜbernehmen_Click()
Dim Result As Long

    On Error GoTo ErrorHandler

    If txtGainNeu.Text < -32768 Or txtGainNeu.Text > 32768 Or _
    txtTINeu.Text < -32768 Or txtTINeu.Text > 32768 Or _
    txtTDNeu.Text < -32768 Or txtTDNeu.Text > 32768 Or _
    txtDeadbandNeu.Text < -32768 Or txtDeadbandNeu.Text > 32768 Or _
    txtLoopCycleNeu.Text < -32768 Or txtLoopCycleNeu.Text > 32768 Then
        Result = MsgBoxT(Me, "Der Bereich der Verstärkung darf nur zwischen -32768 und +32767 liegen!", _
        vbInformation, "Wert zu hoch")
    Else

        If SelChangeTrueFalse = True Then
            If Not txtGainNeu = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", txtGainNeu.Text * 10)
            If Not txtTINeu = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 4, "1", txtTINeu.Text)
            If Not txtTDNeu = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 6, "1", txtTDNeu.Text)
            If Not txtDeadbandNeu = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 8, "1", txtDeadbandNeu.Text)
            If Not txtLoopCycleNeu = "" Then Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 24, "1", txtLoopCycleNeu.Text)

            Call LogFileGeneration("Übernommene Werte: " & vbTab & "KR = " & txtGainNeu.Text & vbCrLf & vbTab & _
            & vbTab & vbTab & "TI = " & txtTINeu.Text & vbCrLf & vbTab & vbTab & vbTab & vbTab & _
            & "TD = " & txtTDNeu.Text & vbCrLf & vbTab & vbTab & vbTab & vbTab & "Deadband = " & _
            & txtDeadbandNeu.Text & vbCrLf & vbTab & vbTab & vbTab & vbTab & "Cycle_Time = " & _
            & txtLoopCycleNeu.Text & " ms * 100")

        Else
            Result = MsgBoxT(Me, "Sie müssen erst einen Regler aus dem DataGrid auswählen!", _
            vbInformation, "Regler auswählen")
        End If

    End If

End Sub

Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdÜbernehmen_Click"
Resume Next

End Sub

```

---



---

```
'-----Anfangswerte aus den fünf TextBoxen in einer CSV-Datei speichern
```

```
Private Sub cmdSpeichern_Click()
Dim i As Integer
Dim Reglername As String
Dim Result As Long
Dim iFile%, strPath$, strDataFile$, strDataText$

On Error GoTo ErrorHandler

If SelChangeTrueFalse = True Then
Reglername = "Anfangswerte, " & frmReglerauswahl.DataGrid1.Columns(2)
CommonDialog3.CancelError = True
CommonDialog3.DialogTitle = "Datei speichern"
CommonDialog3.InitDir = App.Path & "\Undo-Dateien\"
CommonDialog3.Filter = "(*.csv)*.csv"
CommonDialog3.FileName = "UNDO" & Reglername & Date

On Error Resume Next
CommonDialog3.ShowSave

strPath = App.Path & "\Undo-Dateien\"
strDataText = txtGain.Text & ";" & txtTI.Text & ";" & txtTD.Text & ";" & txtDeadband.Text & ";" & txtLoopCycle.Text

If Err = 0 Then
iFile = FreeFile
Open CommonDialog3.FileName For Append As #iFile

If FileLen(CommonDialog3.FileName) > 0 Then
Result = MsgBox("Die Datei " & Chr$(34) & Reglername & ", " & Date & ".csv" & Chr$(34) & " existiert bereits!" & _
" Möchten Sie die existierende Datei ersetzen?", _
vbYesNo, "Ersetzen von Dateien bestätigen")

If Result = vbYes Then
Close #iFile
Open CommonDialog3.FileName For Output As #iFile
Else
Exit Sub
End If
End If

Print #iFile, strDataText
Close #iFile

End If

Else

Result = MsgBoxT(Me, "Sie müssen erst einen Regler aus dem DataGrid auswählen!", _
vbInformation, "Regler auswählen")

End If

Exit Sub
ErrorHandler:
ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdSpeichern_Click"
Resume Next

End Sub
```

---

```
'-----Anfangswerte aus den fünf TextBoxen wieder auslesen und übernehmen
```

```
Private Sub cmdUndo_Click()
Dim strPath$, strDataFile$, strDataText$
Dim v() As String, s As String
Dim L() As String
Dim X1() As Double
Dim X2() As Double
Dim X3() As Double
Dim X4() As Double
Dim X5() As Double
Dim fno
Dim i As Integer

CommonDialog3.CancelError = True
```

```

CommonDialog3.DialogTitle = "Zu alten Werten zurückkehren"
CommonDialog3.InitDir = App.Path & "\Undo-Dateien\"
CommonDialog3.Filter = "(*.csv)*.csv"

```

```

On Error Resume Next
CommonDialog3.ShowOpen

```

```

fno = FreeFile
Open CommonDialog3.FileName For Binary As #fno
s = Space(LOF(fno))
Get #fno, , s
Close #fno

```

```

s = Replace(s, vbCrLf, "")
v() = Split(s, ";")
If UBound(v) = 4 Then
    txtGainNeu.Text = v(0)
    txtTINeu.Text = v(1)
    txtTDNeu.Text = v(2)
    txtDeadbandNeu.Text = v(3)
    txtLoopCycleNeu.Text = v(4)
End If

```

```
Close #fno
```

```
End Sub
```

---

'-----Auf nächste Form gehen

```

Private Sub cmdWeiter_Click()
Dim Result As Long

```

```
On Error GoTo ErrorHandler
```

```

Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".2", "1", False)
Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", True) ' Festwert auf True
Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".4", "1", True) ' Regler einschalten

```

```
If SelChangeTrueFalse = True Then
```

```

    If StopTrueFalse = False Then
        Result = MsgBoxT(Me, "Sie müssen die Aufzeichnung erst stoppen!", _
            vbInformation, "Aufzeichnung stoppen")
    Else

```

```

        'Regler auf Handbetrieb stellen
        Start2TrueFalse = False
        Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", True)

```

```

        Timer2.Enabled = True
        Timer2.Interval = 50000
        Timer2_Timer

```

```

        frmReglerbearbeitung.Show
        On Error Resume Next
        fso.DeleteFile strDataFile
        X_Zaehler = 0
        On Error GoTo 0
        End If
    Else

```

```

        Result = MsgBoxT(Me, "Sie müssen erst einen Regler aus dem DataGrid auswählen!", _
            vbInformation, "Regler auswählen")
        End If

```

```
Exit Sub
```

```
ErrorHandler:
```

```

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"

```

```

ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdWeiter_Click"
Resume Next

```

```
End Sub
```

---

## '-----Beenden-Button

```
Private Sub cmdBeenden_Click()
```

```
    On Error GoTo ErrorHandler
```

## 'Aktiven Regler auf Anfangswerte zurücksetzen

```
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".2", "1", ExtSPon)
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", FixedOutputon)
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".4", "1", Loopon)
```

```
    Unload Me
    Unload frmReglerbearbeitung
    Unload frmVerbindung
```

```
Exit Sub
```

```
ErrorHandler:
```

```
MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"
```

```
ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: cmdBeenden_Click"
Resume Next
```

```
End Sub
```

---

## '-----Beenden-Abfrage

```
Private Sub Form_QueryUnload(cancel As Integer, UnloadMode As Integer)
```

```
Dim strPath$, strDataFile$
```

```
Dim Result As Long
```

```
    On Error GoTo ErrorHandler
```

```
    strPath = App.Path & "\CSV-Dateien\"
    strDataFile = strPath & "Test.csv"
```

```
    Result = MsgBoxT(Me, "Möchten Sie das Programm wirklich beenden?", _
        vbYesNo + vbQuestion, "Programm beenden")
```

```
    If Result = vbYes Then
```

```
        Timer1 = False
        Timer2 = False
        frmReglerbearbeitung.Timer3 = False
        On Error Resume Next
        fso.DeleteFile strDataFile 'Datei Test.csv löschen
        On Error GoTo 0
```

```
    Else
        cancel = True
    End If
```

```
Exit Sub
```

```
ErrorHandler:
```

```
ErrorLog Err.Number, Err.Description, "FrmReglerauswahl: Form_QueryUnload"
Resume Next
```

```
End Sub
```

---

## '-----Terminieren

```
Private Sub Form_Terminate()
```

## 'Aufräumen...

```
    On Error Resume Next
```

## 'Objecte schließen und Verweise freigeben'

```
    rs.Close
    Set rs = Nothing
```

```
    rs2.Close
    Set rs2 = Nothing
```

```
rs3.Close  
Set rs3 = Nothing  
  
cn.Close  
Set cn = Nothing  
  
On Error GoTo 0  
End Sub
```

## Quellcode in frmReglerbearbeitung

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
'
'-----
'
' Source Code Administration Information
'
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
'
'-----
'
' File description
'
' PROGRAM:  Reglertuning
'
' FILENAME: frmReglerbearbeitungl.frm
'
' FUNCTIONS: - Min- und Maxwerte für den Stellwert vorgeben
'            - Stellwertsprung mit Sprungantwort aufzeichnen
'            - Aufzeichnung stoppen
'            - Aufzeichnung speichern und laden
'            - Aufzeichnung drucken
'            - Tangente anlegen und Streckenparameter ermitteln
'            - Regler nach Ziegler und Nichols optimieren
'            - Regler nach Takahashi optimieren
'
' COMMENTS:
'
' MODIFICATIONS:
'
'-----
```

### Option Explicit

```
Dim Start As Date
Dim T63 As Double
Dim T0 As Double
Dim yTMax As Double
Dim yTKMax As Double
Public ySAMax As Double
Public ySAKMax As Double
```

### '-----Für TextBox nur Zahlen zulassen

```
Private Sub txtStellwertvorgabe_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case 188, 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtStellwertvorgabe = "0" Then txtStellwertvorgabe = ""
        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtStellwertvorgabe = Left(txtStellwertvorgabe, Len(txtStellwertvorgabe) - 1)
    End Select
End Sub
```

### '-----Für TextBox nur Zahlen zulassen

```
Private Sub txtStellwertMin_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case 188, 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8, _
            vbKeyNumpad9
            If txtStellwertMin = "0" Then txtStellwertMin = ""
    End Select
End Sub
```

```

    Case Else
        MsgBox "Bitte nur Zahlen eingeben!", vbInformation
        If KeyCode < 91 And KeyCode > 65 Then txtStellwertMin = Left(txtStellwertMin, Len(txtStellwertMin) - 1)
    End Select
End Sub

```

---

'-----Für TextBox nur Zahlen zulassen

```

Private Sub txtStellwertMax_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode

        Case 188, 8, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, vbKeyDelete, vbKeyInsert, vbKeyLeft, _
            vbKeyUp, vbKeyRight, vbKeyDown, vbKeyClear, vbKeyNumpad0, vbKeyNumpad1, vbKeyNumpad2, _
            vbKeyNumpad3, vbKeyNumpad4, vbKeyNumpad5, vbKeyNumpad6, vbKeyNumpad7, vbKeyNumpad8,
            vbKeyNumpad9
            If txtStellwertMax = "0" Then txtStellwertMax = ""

        Case Else
            MsgBox "Bitte nur Zahlen eingeben!", vbInformation
            If KeyCode < 91 And KeyCode > 65 Then txtStellwertMax = Left(txtStellwertMax, Len(txtStellwertMax) - 1)
    End Select
End Sub

```

---

'-----Beim laden der Form Vorgaben fürs Label 5 vergeben

```

Private Sub Form_Load()

    On Error GoTo ErrorHandler

    Label5.Font.Bold = True

Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "FrmReglerbearbeitung: Form_Load"
Resume Next

End Sub

```

---

'-----Min- und Maxwert der Stellgröße in die Steuerung schreiben

```

Private Sub cmdÜbernehmenGrenzen_Click()

    If Not txtStellwertMax = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 20, "1", txtStellwertMax * 10)
    If Not txtStellwertMin = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 18, "1", txtStellwertMin * 10)

End Sub

```

---

'-----Bei Klick auf Start-Button den Timer2 aufrufen und starten

```

Private Sub cmdStart_Click()
    Dim Result As Long

    On Error GoTo ErrorHandler

    strDataFile = App.Path & "\CSV-Dateien\" & "Test.csv"

    Start = Time
    StartTrueFalse = True

    Timer_Start.Interval = 1000
    Timer_Start.Enabled = True

    Timer3.Interval = 1000
    Timer3.Enabled = True
    Timer3_Timer

    Call LogFileGeneration("Beginn der Aufzeichnung! " & Time & vbCrLf _
        & "Stellwertsprung" & vbCrLf & " = " & txtStellwertvorgabe.Text)

```

```
Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerbearbeitung: cmdStart_Click"
Resume Next

End Sub
```

---

#### '-----Reglerdaten in CSV-Datei schreiben

```
Public Sub csv_WriteStellwert()
Dim i As Long
Dim fno%, strPath$, strDataFile$, strDataText$

    On Error GoTo ErrorHandler

    X_Zaehler = X_Zaehler + 1

    If Not Field(1, 1) = "" Then
        i = UBound(Field(), 2)
        ReDim Preserve Field(3, i + 1)
    End If

    Field(1, UBound(Field(), 2)) = X_Zaehler
    Field(2, UBound(Field(), 2)) = TempArray2(5)
    Field(3, UBound(Field(), 2)) = TempArray2(17)

    strPath = App.Path & "\CSV-Dateien\"
    strDataFile = strPath & "Test.csv"
    strDataText = X_Zaehler & "," & TempArray2(5) / 10 ^ Precision & ";" & TempArray2(17) / 10

    fno = FreeFile
    Open strDataFile For Append As #fno
    Print #fno, strDataText
    Close #fno

Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerbearbeitung: csv_WriteStellwert"
Resume Next

End Sub
```

---

#### '-----Reglerverhalten in der PictureBox anzeigen

```
Public Sub TrendView_show()
Dim L() As String
Dim X() As Double
Dim Y() As Double
Dim y2() As Double
Dim v() As String
Dim fno, s As String, i As Integer
Dim xmin As Double, xmax As Double, ymin As Double, ymax As Double

    On Error GoTo ErrorHandler

    fno = FreeFile
    Open strDataFile For Binary As #fno
    s = Space(LOF(fno))
    Get #fno, , s
    Close #fno

    L() = Split(s, vbCrLf)
    ReDim X(1 To UBound(L) + 1)
    ReDim Y(1 To UBound(L) + 1)
    ReDim y2(1 To UBound(L) + 1)
```

```
For i = 0 To UBound(L)

    v() = Split(L(i), ";")
    If UBound(v) = 2 Then
        X(i + 1) = v(0)
        Y(i + 1) = v(1)
        y2(i + 1) = v(2)
    End If

Next i

For i = 1 To UBound(X)

    If X(i) > xmax Then
        xmax = X(i)
    End If

    If X(i) < xmin Then
        xmin = X(i)
    End If

    If Y(i) > ymax Then
        ymax = Y(i)
    End If

    If Y(i) < ymin Then
        ymin = Y(i)
    End If

Next i

If Start2TrueFalse = False Then

    Picture1.Scale (1, ymax + 1)-(1000, ymin - 1) 'Start und Ende der Skala festlegen
    Picture1.Cls

    Picture1.PSet (X(1), y2(1))
    For i = 2 To UBound(y2) - 1
        Picture1.Line -(X(i), y2(i)), vbRed
    Next i

    Picture1.PSet (X(1), Y(1))
    For i = 2 To UBound(Y) - 1
        Picture1.Line -(X(i), Y(i)), vbBlack
    Next i

Else

    frmReglerauswahl.Picture1.Scale (1, ymax + 1)-(1000, ymin - 1) 'Start und Ende der Skala festlegen
    frmReglerauswahl.Picture1.Cls

    frmReglerauswahl.Picture1.PSet (X(1), y2(1))
    For i = 2 To UBound(y2) - 1
        frmReglerauswahl.Picture1.Line -(X(i), y2(i)), vbRed
    Next i

    frmReglerauswahl.Picture1.PSet (X(1), Y(1))
    For i = 2 To UBound(Y) - 1
        frmReglerauswahl.Picture1.Line -(X(i), Y(i)), vbBlack
    Next i

End If

Exit Sub
ErrorHandler:

ErrorLog Err.Number, Err.Description, "FrmReglerbearbeitung: TrendView_show"
Resume Next

End Sub
```

---



'-----Bei Klick auf Stop-Button den Timer anhalten

```
Public Sub cmdStop_Click()
Dim Result As Long

StopTrueFalse = True

Result = MsgBoxT(Me, "Möchten Sie die Aufzeichnung wirklich stoppen?", _
vbYesNo + vbQuestion + vbCritical, "Aufzeichnung stoppen")

If Result = vbYes Then

Call LogFileGeneration("Aufzeichnung gestoppt: " & vbTab & Now & vbCrLf & vbCrLf _
& "Istwert: " & vbTab & txtIstwert.Text & vbCrLf _
& "Dauer der Aufzeichnung: " & Label5.Caption)

Timer3 = False

Else

End If

End Sub
```

'-----Streckenparameter bestimmen und Tangente am Graphen anlegen

```
Private Sub cmdAnalysiereStrecke_Click()
Dim Result As Long
Dim mG As Double
Dim bG As Double
Dim k As Integer
Dim j As Integer
Dim m As Integer
Dim z As Integer
Dim i As Integer
Dim Stellwertsprung_Max As Double
Dim ySAJMax As Double
Dim Sprungantwort_Max As Double
Dim Sprungantwort_Min As Double
Dim T_Stellwertsprung As Integer
Dim T_Sprungantwort As Integer
Dim Differenz_Sprungantwort As Double
Dim KS As Double
Dim Tt As Double
Dim Ts As Double
Dim TTan As Double
Dim y63P As Double
Dim y63PMin As Double
Dim y63PMax As Double
Dim X63PMax As Double
Dim X63PMin As Double
Dim fno$, s$, L() As String, v() As String, X() As Double, Y() As Double, y2() As Double

If StopTrueFalse = True Then

fno = FreeFile
Open strDataFile For Binary As #fno
s = Space(LOF(fno))
Get #fno, , s
Close #fno

L() = Split(s, vbCrLf)
ReDim X(1 To UBound(L))
ReDim Y(1 To UBound(L))
ReDim y2(1 To UBound(L))

For i = 0 To UBound(L)
v() = Split(L(i), ";")
If UBound(v) = 2 Then
X(i + 1) = v(0)
Y(i + 1) = v(1)
y2(i + 1) = v(2)
End If
Next i

'Min- und Maximalwert der Sprungantwort ermitteln aus der Spalte der CSV auslesen
Sprungantwort_Max = Y(1)
```

```

For j = 1 To UBound(Y)
  If Y(j) > Sprungantwort_Max Then Sprungantwort_Max = Y(j)
Next j

Sprungantwort_Min = Y(1)

For z = 1 To UBound(Y)
  If Y(z) < Sprungantwort_Min Then Sprungantwort_Min = Y(z)
Next z

If Hoch(Y, Sprungantwort_Max, Sprungantwort_Min) Then

  'Maximalwert des Sprungs aus der Spalte der CSV auslesen
  Stellwertsprung_Max = 0

  For j = 1 To UBound(y2)
    If y2(j) > Stellwertsprung_Max Then Stellwertsprung_Max = y2(j)
  Next j

  'Zeit (j) von Anfang des Sprungs berechnen
  ySAJMax = y2(1)

  For j = 1 To UBound(y2)
    If y2(j) > ySAJMax + 1 Or y2(j) < ySAJMax - 1 Then
      ySAJMax = y2(j)
      Exit For
    End If
  Next j

  T_Stellwertsprung = j - 1

  Differenz_Sprungantwort = Sprungantwort_Max - Sprungantwort_Min

  'Zeit (k) von Anfang der Sprungantwort berechnen
  ySAKMax = Y(1)

  For k = 1 To UBound(Y)
    If Y(k) > ySAKMax Then
      ySAKMax = Y(k - 1)
      Exit For
    End If
  Next k

  T_Sprungantwort = k - 1

  y63PMax = 0

  For m = 1 To UBound(Y)
    If Y(m) = 0.63 * Differenz_Sprungantwort + Y(1) Then
      y63PMax = Y(m)

      Tt = T_Sprungantwort - T_Stellwertsprung
      Ts = m - T_Sprungantwort

      KS = (Sprungantwort_Max - Y(1)) / (Stellwertsprung_Max - y2(1))
      txtKS.Text = KS
      txtTt.Text = Tt
      txtTs.Text = Ts

      T63 = TTan
      T0 = T_Sprungantwort
      ySAMax = Sprungantwort_Max

      Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen
      Exit For

    Else
      If Y(m) > 0.63 * Differenz_Sprungantwort + Y(1) Then
        y63PMax = Y(m)
        X63PMax = m
        y63PMin = Y(m - 1)
        X63PMin = m - 1

        KS = (Sprungantwort_Max - Y(1)) / (Stellwertsprung_Max - y2(1))
        y63P = 0.63 * Differenz_Sprungantwort + Y(1)

```

```

    mG = (y63PMax - y63PMin) / (X63PMax - X63PMin)

    bG = y63PMin - mG * X63PMin

    Ts = ((y63P - bG) / mG) - T_Sprungantwort

    TTan = ((y63P - bG) / mG)

    Tt = T_Sprungantwort - T_Stellwertsprung

    txtKS.Text = KS
    txtTt.Text = Tt
    txtTs.Text = Ts + 10

    T63 = TTan
    T0 = T_Sprungantwort
    ySAMax = Sprungantwort_Max

    Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen
Exit For

End If
End If

Next m

Else
'Maximalwert des Sprungs aus der Spalte der CSV auslesen
Stellwertsprung_Max = 0

For j = 1 To UBound(y2)
    If y2(j) > Stellwertsprung_Max Then Stellwertsprung_Max = y2(j)
Next j

'Zeit (j) von Anfang des Sprungs berechnen
ySAJMax = y2(1)

For j = 1 To UBound(y2)
    If y2(j) > ySAJMax + 1 Or y2(j) < ySAJMax - 1 Then
        ySAJMax = y2(j)
        Exit For
    End If
Next j

T_Stellwertsprung = j - 1

Differenz_Sprungantwort = Sprungantwort_Max - Sprungantwort_Min

'Zeit (k) von Anfang der Sprungantwort berechnen
ySAKMax = Y(1)

For k = 1 To UBound(Y)
    If Y(k) < ySAKMax Then
        ySAKMax = Y(k - 1)
    End If
Next k

T_Sprungantwort = k - 1

y63PMax = 0

For m = 1 To UBound(Y)
    If Y(m) = 0.37 * Differenz_Sprungantwort + Y(1) Then
        y63PMax = Y(m)

        Tt = T_Sprungantwort - T_Stellwertsprung
        Ts = m - T_Sprungantwort

        KS = (Sprungantwort_Min - Y(1)) / (ySAJMax - y2(1))
        txtKS.Text = KS
        txtTt.Text = Tt
        txtTs.Text = Ts

        T63 = TTan
        T0 = T_Sprungantwort
        ySAMax = Sprungantwort_Min

```

```

        Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen
    Exit For

    Else

        If Y(m) < 0.37 * Differenz_Sprungantwort + Sprungantwort_Min Then
            y63PMax = Y(m)
            X63PMax = m
            y63PMin = Y(m - 1)
            X63PMin = m - 1

            KS = (Sprungantwort_Min - Y(1)) / (ySAJMax - y2(1))
            y63P = 0.37 * Differenz_Sprungantwort + Sprungantwort_Min

            mG = (y63PMax - y63PMin) / (X63PMax - X63PMin)

            bG = y63PMin - mG * X63PMin

            Ts = ((y63P - bG) / mG) - T_Sprungantwort

            TTan = ((y63P - bG) / mG)

            Tt = T_Sprungantwort - T_Stellwertsprung

            txtKS.Text = KS
            txtTt.Text = Tt
            txtTs.Text = Ts

            T63 = TTan
            T0 = T_Sprungantwort
            ySAMax = Sprungantwort_Min

            Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen
        Exit For
    End If
End If

Next m
End If

Call LogFileGeneration("Ermittelte Werte: " & vbTab & "KS = " & txtKS.Text & vbCrLf & vbTab _
    & vbTab & vbTab & "Tt = " & txtTt.Text & vbCrLf & vbTab & vbTab & vbTab _
    & "Ts = " & txtTs.Text & vbCrLf)

Else
    Result = MsgBoxT(Me, "Stoppen Sie vorher die Aufzeichnung!", _
        vbInformation, "Ermitteln nicht möglich")

End If

End Sub

```

'-----Auswertung, ob die Sprungantwort vom Wert her größer wird oder kleiner

```

Private Function Hoch(feld() As Double, maxWert As Double, minWert As Double) As Boolean
    Dim Ymin1 As Double
    Dim Zähler_hoch As Integer
    Dim Zähler_runter As Integer
    Dim i As Integer

    Ymin1 = feld(1)

    For i = 2 To UBound(feld())

        If feld(i) > Ymin1 Then
            Zähler_hoch = Zähler_hoch + 1
            Zähler_runter = 0
        End If

        If feld(i) < Ymin1 Then
            Zähler_runter = Zähler_runter + 1
            Zähler_hoch = 0
        End If

    Next i

    If Zähler_hoch > Abs(maxWert - minWert) / 5 Then Exit For

```

```

If Zähler_runter > Abs(maxWert - minWert) / 5 Then Exit For

Next i

If Zähler_hoch < Zähler_runter Then Hoch = False Else Hoch = True

End Function

```

'-----Optimale Parameter nach Ziegler und Nichols berechnen und in die Steuerung schreiben

```

Private Sub cmdZieglerNichols_Click()
Dim Result As Long
Dim KR As Double
Dim TN As Double
Dim TV As Double

If TempArray2(3) = 0 Then
If TempArray2(14) = True Then

KR = 0.9 * (txtTs.Text / (txtKS.Text * txtTt.Text))
TN = 3.33 * txtTt.Text

TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 4, "1", TN)
Else

KR = 0.9 * (txtTs.Text / (txtKS.Text * txtTt.Text))
TN = 3.33 * txtTt.Text * 1000

TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 4, "1", TN)

End If

Else
If TempArray2(14) = True And TempArray2(15) = True Then

KR = 1.2 * (txtTs.Text / (txtKS.Text * txtTt.Text))
TN = 2 * txtTt.Text
TV = 0.5 * txtTt.Text

TV = Round(TV, 0)
TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = True And TempArray2(15) = False Then

KR = 1.2 * (txtTs.Text / (txtKS.Text * txtTt.Text))
TN = 2 * txtTt.Text
TV = 0.5 * txtTt.Text * 1000

TV = Round(TV, 0)
TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = False And TempArray2(15) = True Then

KR = 1.2 * (txtTs.Text / (txtKS.Text * txtTt.Text))

```

```

    TN = 2 * txtTt.Text * 1000
    TV = 0.5 * txtTt.Text

    TV = Round(TV, 0)
    TN = Round(TN, 0)
    KR = Round(KR, 1)

    Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
    Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
    Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = False And TempArray2(15) = False Then

    KR = 1.2 * (txtTs.Text / (txtKS.Text * txtTt.Text))
    TN = 2 * txtTt.Text * 1000
    TV = 0.5 * txtTt.Text * 1000

    TV = Round(TV, 0)
    TN = Round(TN, 0)
    KR = Round(KR, 1)

    Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
    Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
    Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If
End If

Call LogFileGeneration("Optimiert nach Ziegler/Nichols: " & vbCrLf & Now)

End Sub

```

'-----Optimale Parameter nach Takahashi berechnen und in die Steuerung schreiben

```

Private Sub cmdTakahashi_Click()
Dim Result As Long
Dim KR As Double
Dim TN As Double
Dim TV As Double
Dim T As Double

If TempArray2(3) = 0 Then
    If TempArray2(14) = True Then

        KR = 0.9 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10) / 2))
        TN = 3.33 * (txtTt.Text + (TempArray2(10) / 1000) / 2)

        TN = Round(TN, 0)
        KR = Round(KR, 1)

        Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
        Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 4, "1", TN)
    Else

        KR = 0.9 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10) / 2))
        TN = 3.33 * (txtTt.Text + TempArray2(10) / 2)

        TN = Round(TN, 0)
        KR = Round(KR, 1)

        Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
        Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 4, "1", TN)

    End If

Else
    If TempArray2(14) = True And TempArray2(15) = True Then

        KR = 1.2 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10)))
        TN = 2 * (txtTt.Text + TempArray2(10) / 2) ^ 2 / (txtTt.Text + TempArray2(10) / 1000)
        TV = 0.5 * (txtTt.Text + TempArray2(10) / 1000)

        TV = Round(TV, 0)
        TN = Round(TN, 0)
    End If
End Sub

```

```

KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = True And TempArray2(15) = False Then

KR = 1.2 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10)))
TN = 2 * (txtTt.Text + TempArray2(10) / 2) ^ 2 / (txtTt.Text + TempArray2(10) / 1000)
TV = 0.5 * (txtTt.Text + TempArray2(10))

TV = Round(TV, 0)
TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = False And TempArray2(15) = True Then

KR = 1.2 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10)))
TN = 2 * (txtTt.Text + TempArray2(10) / 2) ^ 2 / (txtTt.Text + TempArray2(10))
TV = 0.5 * (txtTt.Text + TempArray2(10) / 1000)

TV = Round(TV, 0)
TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If

If TempArray2(14) = False And TempArray2(15) = False Then

KR = 1.2 * txtTs.Text / (txtKS.Text * (txtTt.Text + TempArray2(10)))
TN = 2 * (txtTt.Text + TempArray2(10) / 2) ^ 2 / (txtTt.Text + TempArray2(10))
TV = 0.5 * (txtTt.Text + TempArray2(10))

TV = Round(TV, 0)
TN = Round(TN, 0)
KR = Round(KR, 1)

Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + 2, "1", KR * 10)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 4, "1", TN)
Call writeValuesOPC("DB6", "TIME" & (pos - 1) * 100 + 6, "1", TV)

End If
End If

Call LogFileGeneration("Optimiert nach Takahashi: " & vbTab & Now)

End Sub

```

---

'-----CSV-Datei durch CommonDialog in PictureBox laden

```

Public Sub cmdKurveLaden_Click()
Dim i As Integer
Dim iFile%, strDataText$
Dim L() As String
Dim X() As Double
Dim Y() As Double
Dim y2() As Double
Dim v() As String
Dim fno, s As String
Dim xmin As Double, xmax As Double, ymin As Double, ymax As Double

CommonDialog1.CancelError = True
CommonDialog1.DialogTitle = "Datei laden"

```

```

CommonDialog1.InitDir = App.Path & "\CSV-Dateien\"
CommonDialog1.Filter = "(*.csv)|*.csv"

On Error Resume Next
CommonDialog1.ShowOpen

strDataFile = CommonDialog1.FileName

fno = FreeFile
Open CommonDialog1.FileName For Binary As #fno
s = Space(LOF(fno))
Get #fno, , s
Close #fno

L() = Split(s, vbCrLf)
ReDim X(1 To UBound(L) + 1)
ReDim Y(1 To UBound(L) + 1)
ReDim y2(1 To UBound(L) + 1)

For i = 0 To UBound(L)

    v() = Split(L(i), ";")
    If UBound(v) = 2 Then
        X(i + 1) = v(0)
        Y(i + 1) = v(1)
        y2(i + 1) = Val(v(2))
    End If
Next i

For i = 1 To UBound(X)

    If X(i) > xmax Then
        xmax = X(i)
    End If

    If X(i) < xmin Then
        xmin = X(i)
    End If

    If Y(i) > ymax Then
        ymax = Y(i)
    End If

    If Y(i) < ymin Then
        ymin = Y(i)
    End If

Next i

CommonDialog1.ShowColor           'Farbauswahl für Graphen anzeigen

Picture1.Scale (1, ymax + 1)-(1000, ymin - 1)   'Start und Ende der Skala festlegen
Picture1.ForeColor = CommonDialog1.Color

Picture1.PSet (X(1), y2(1))
For i = 2 To UBound(y2) - 1
Picture1.Line -(X(i), y2(i)), vbGreen
Next i

Picture1.PSet (X(1), Y(1))
For i = 2 To UBound(Y) - 1
Picture1.Line -(X(i), Y(i))
Next i

End Sub

```

---

'-----Anzeige in Picture1 löschen und Timer3 stoppen

```

Private Sub cmdKurvenLöschen_Click()
Dim Result As Long
Dim strPath$, strDataFile$

strPath = App.Path & "\CSV-Dateien\"
strDataFile = strPath & "Test.csv"

Result = MsgBoxT(Me, "Möchten Sie die Kurven wirklich löschen?", _

```



```

        vbYesNo + vbQuestion + vbCritical, "Kurven löschen")

If Result = vbYes Then

    Timer3 = False
    Picture1.Cls
    On Error Resume Next
    fso.DeleteFile strDataFile

Else

End If

X_Zaehler = 0

End Sub

```

---

#### '-----CSV-Datei durch CommonDialog speichern

```

Public Sub cmdKurveSpeichern_Click()
Dim Reglername As String, i As Integer
Static j&
Dim iFile%, strDataText$, Result&

j = j + 1
Reglername = "Stellwertsprung, " & frmReglerauswahl.DataGrid1.Columns(2)

CommonDialog1.CancelError = True
CommonDialog1.DialogTitle = "Datei speichern"
CommonDialog1.InitDir = App.Path & "CSV-Dateien\"
CommonDialog1.Filter = "(*.csv)|*.csv"
CommonDialog1.FileName = Reglername

On Error Resume Next
CommonDialog1.ShowSave

If Err = 0 Then

    iFile = FreeFile
    Open CommonDialog1.FileName For Append As #iFile

    If FileLen(CommonDialog1.FileName) > 0 Then

        Result = MsgBox("Die Datei " & Chr$(34) & Reglername & ".csv" & Chr$(34) & " existiert bereits!" & _
            " Möchten Sie die existierende Datei ersetzen?", _
            vbYesNo, "Ersetzen von Dateien bestätigen")

        If Result = vbYes Then

            Close #iFile
            Open CommonDialog1.FileName For Output As #iFile
            fso.DeleteFile strDataFile
        Else

            Exit Sub

        End If
    End If

    For i = 1 To UBound(Field(), 2)
        Print #iFile, Field(1, i) & ";" & Field(2, i) _
            & ";" & Field(3, i)
    Next i

    Close #iFile

End If

End Sub

```

---

#### '-----PictureBox um den Faktor 2 zoomen

```

Private Sub cmd_plus_Click()
Dim Result As Long
Dim i As Integer

```

```

Dim fno$, s$, L() As String, v() As String, X() As Double, Y() As Double, y2() As Double

If StopTrueFalse = True Then

    fno = FreeFile
    Open strDataFile For Binary As #fno
    s = Space(LOF(fno))
    Get #fno, , s
    Close #fno

    L() = Split(s, vbCrLf)
    ReDim X(1 To UBound(L) + 1)
    ReDim Y(1 To UBound(L) + 1)
    ReDim y2(1 To UBound(L) + 1)

    For i = 0 To UBound(L)

        v() = Split(L(i), ";")
        If UBound(v) = 2 Then
            X(i + 1) = v(0)
            Y(i + 1) = v(1)
            y2(i + 1) = v(2)
        End If

    Next i

    For i = 1 To UBound(X())
        X(i) = X(i) * 2
    Next i

    Open strDataFile For Output As #fno
    For i = 1 To UBound(X()) - 1
        Print #fno, X(i) & ";" & Y(i) & ";" & y2(i)
    Next i
    Close #fno

    Call TrendView_show

Else
    Result = MsgBoxT(Me, "Stoppen Sie vorher die Aufzeichnung!", _
        vbInformation, "Ermitteln nicht möglich")
End If

T63 = T63 * 2
T0 = T0 * 2
Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen

End Sub

```

---

'-----PictureBox um den Faktor 2 zurück zoomen

```

Private Sub cmd_minus_Click()
Dim Result As Long
Dim i As Integer
Dim fno$, s$, L() As String, v() As String, X() As Double, Y() As Double, y2() As Double

If StopTrueFalse = True Then

    fno = FreeFile
    Open strDataFile For Binary As #fno
    s = Space(LOF(fno))
    Get #fno, , s
    Close #fno

    L() = Split(s, vbCrLf)
    ReDim X(1 To UBound(L) + 1)
    ReDim Y(1 To UBound(L) + 1)
    ReDim y2(1 To UBound(L) + 1)

    For i = 0 To UBound(L)
        v() = Split(L(i), ";")

        If UBound(v) = 2 Then
            X(i + 1) = v(0)
            Y(i + 1) = v(1)
            y2(i + 1) = v(2)
        End If
    Next i

```

```
End If

Next i

For i = 1 To UBound(X())
    X(i) = X(i) / 2
Next i

Open strDataFile For Output As #fno
For i = 1 To UBound(X()) - 1
    Print #fno, X(i) & ";" & Y(i) & ";" & y2(i)
Next i
Close #fno

Call TrendView_show

Else
    Result = MsgBoxT(Me, "Stoppen Sie vorher die Aufzeichnung!", _
vbInformation, "Ermitteln nicht möglich")

End If

T63 = T63 / 2
T0 = T0 / 2
Picture1.Line (T0, ySAKMax)-(T63, ySAMax), vbGreen

End Sub
```

---

'-----Graphen in Picture1 drucken

```
Public Sub cmdPrint_Click()
Dim BeginPage, EndPage, NumCopies, Orientation
Dim i As Integer

    CommonDialog2.CancelError = True
    On Error GoTo ErrHandler
    cmdZurück.SetFocus

    CommonDialog2.ShowPrinter

    BeginPage = CommonDialog2.FromPage
    EndPage = CommonDialog2.ToPage
    NumCopies = CommonDialog2.Copies
    Orientation = CommonDialog2.Orientation = cdlPortrait

    For i = 1 To NumCopies
        Printer.PaintPicture Picture1.Image, 0, 0
        Printer.EndDoc
    Next
    Exit Sub

ErrHandler:
    ' Benutzer hat auf Abbrechen-Schaltfläche geklickt
    Exit Sub
End Sub
```

---

'-----Wenn Timer2 kommt Trendview\_show und csv\_Write aufrufen

```
Private Sub Timer3_Timer()
Dim Halt As Date

    Halt = Now
    Label5 = Format$(Halt - Start, "hh:mm:ss")

    Call ReadFromPLC1
    Call csv_WriteStellwert
    Call TrendView_show

End Sub
```

---

'-----Auf erste Form zurück gehen

```
Private Sub cmdZurück_Click()
    'Regler auf Automatikbetrieb stellen
    Call writeValuesOPC("DB6", "X" & (pos - 1) * 100 + 32 & ".3", "1", False)

    frmReglerauswahl.Show
    On Error Resume Next
    fso.DeleteFile strDataFile
    On Error GoTo 0
End Sub
```

---

'-----Timer für Verzögerung der Aufzeichnung

```
Private Sub Timer_Start_Timer()
    Timer_Start.Enabled = False
    If Not txtStellwertvorgabe = "" Then Call writeValuesOPC("DB6", "INT" & (pos - 1) * 100 + _
        16, "1", txtStellwertvorgabe.Text * 10)
End Sub
```

## Quellcode in mdlMain

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
'
'-----
'
' Source Code Administration Information
'
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
'
'-----
'
' File description
'
' PROGRAM:  Reglertuning
'
' FILENAME: mdlMain.bas
'
' FUNCTIONS: - Datenbankverbindung herstellen
'            - OPC-Gruppe kreieren
'            - Von PLC lesen
'            - MsgBox ohne zeitliche Unterbrechung
'            - In PLC schreiben
'
' COMMENTS:
'
' MODIFICATIONS:
'
'-----
'-----
Option Explicit
Option Base 1

Public cn As New ADODB.Connection

'Erstellen eines Recordsets zum Zugriff auf DB-Objekte
Public rs As New ADODB.Recordset
Public rs2 As New ADODB.Recordset
Public rs3 As New ADODB.Recordset

'Deklaration von fso, um auf Dateien zuzugreifen bzw. zu löschen
Public fso As New FileSystemObject

'Deklaration der OPC-Gruppe und der Items
Public mOPCGroup() As clsOPCGroup
Public TempArray1(17) As String
Public TempArray2() As Variant

'Deklaration des Zählerschrittes für die erste Zeile in der CSV-Datei
Public X_Zaehler As Double

'Deklaration des Operanden zur Berechnung der richtigen Adressen für den ausgewählten Regler
Public pos As Integer

'Deklaration des Operanden zum Auslesen der Nachkommastellen des Istwertes aus der Datenbank
Public Precision As Integer

'Deklaration des Namens für die CSV-Datei zur Kurvenaufzeichnung
Public strDataFile As String

'Deklaration der Bool-Variablen für Bedingungen, wenn bestimmte Prozeduren noch nicht ausgeführt wurden
Public SelChangeTrueFalse As Boolean
Public StartTrueFalse As Boolean
Public Start2TrueFalse As Boolean
Public StopTrueFalse As Boolean

'Deklaration der Sollwertgrenzen aus der Datenbank
Public SollHighLimit As Integer
Public SollLowLimit As Integer

'Field() als undimensioniertes Array deklarieren
Public Field() As String
```

## 'Deklaration der OPC-Verbindung

```
Public mstrOPCConnection As String
```

## 'Deklaration des Zählers der OPC-Gruppen

```
Private miGroupsCount%
```

## 'Deklaration der Funktion für die MessageBox ohne Time-Out

```
Private Declare Function MessageBox Lib "user32" Alias _
    "MessageBoxA" (ByVal hwnd As Long, ByVal lpText _
    As String, ByVal lpCaption As String, ByVal _
    wType As Long) As Long
```

## 'Deklaration der Funktion für die Datenbankverbindung in der Form frmVerbindung

```
Public Declare Function SQLDataSources Lib "ODBC32.DLL" (ByVal _
    henv As Long, ByVal fDirection As Integer, _
    ByVal szDSN As String, ByVal cbDSNMax As Integer, _
    pcbDSN As Integer, ByVal szDescription As String, _
    ByVal cbDescriptionMax As Integer, pcbDescription As Integer) _
    As Integer
```

```
Public Declare Function SQLAllocEnv Lib "ODBC32.DLL" (_
    env As Long) _
    As Integer
```

## 'Deklaration von Konstanten zum Abrufen der DSN und der API Funktion

```
Public Const SQL_SUCCESS As Long = 0
Public Const SQL_FETCH_NEXT As Long = 1
```

## '-----Verbindung über ODBC zur Datenbank aufnehmen

```
Public Sub main()
```

```
    On Error GoTo Error_Handler
```

```
    Set cn = New ADODB.Connection ' cn wird als neue ADO Datenbank-Verbindung anerkannt
```

```
    Set rs = New ADODB.Recordset ' rs wird als neuer ADO Recordset anerkannt
```

```
Verbinden:
```

```
    frmVerbindung.Show 1
```

```
With cn
```

```
    .ConnectionString = "DSN=" & frmVerbindung.cboDSNList.Text & ";USER=" & frmVerbindung.txtbenutzername.Text & " _
        ;Password=" & frmVerbindung.txtPasswort.Text ' Datenquelle
```

```
    .CursorLocation = adUseClient
```

```
    .Open
```

```
        ' Verbindung öffnen
```

```
End With
```

```
    frmReglerauswahl.Show
```

```
Exit Sub
```

```
Error_Handler:
```

```
    MsgBox "Benutzername oder Passwort falsch!", vbCritical, "Verbindungsfehler"
```

```
    GoTo Verbinden
```

```
End Sub
```

## '-----Neue OPC-Verbindungen schaffen und Items hinzufügen

```
Public Sub OPC_Connect(Adresse1 As String, Adresse2 As String, Adresse3 As String, Adresse4 As String, _
    Adresse5 As String, Adresse6 As String, Adresse7 As String, Adresse8 As String, _
    Adresse9 As String, Adresse10 As String, Adresse11 As String, Adresse12 As String, _
    Adresse13 As String, Adresse14 As String, Adresse15 As String, Adresse16 As String, _
    Adresse17 As String)
```

```
Dim NoErrors As Boolean
```

```
ReDim mOPCGroup(1)
```

```
    On Error GoTo ErrorHandler
```

```
    miGroupsCount = 1
```

```
    Set mOPCGroup(1) = New clsOPCGroup
```

```

mOPCGroup(1).OPCServerName = "OPC.SimaticNet"
mOPCGroup(1).OPCGroupName = "AxelTest1"
mOPCGroup(1).OPCUpdateRate = 1000           'Update-Rate von jeder OPC-Gruppe
mOPCGroup(1).OPCItemsCount = 17           'Anzahl der Items in der OPC-Gruppe
TempArray1(1) = Adresse1
TempArray1(2) = Adresse2
TempArray1(3) = Adresse3
TempArray1(4) = Adresse4
TempArray1(5) = Adresse5
TempArray1(6) = Adresse6
TempArray1(7) = Adresse7
TempArray1(8) = Adresse8
TempArray1(9) = Adresse9
TempArray1(10) = Adresse10
TempArray1(11) = Adresse11
TempArray1(12) = Adresse12
TempArray1(13) = Adresse13
TempArray1(14) = Adresse14
TempArray1(15) = Adresse15
TempArray1(16) = Adresse16
TempArray1(17) = Adresse17

```

```
NoErrors = mOPCGroup(1).AddOPCItems(TempArray1())
```

```
Exit Sub
ErrorHandler:
```

```
MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
vbCritical, "Fehler"
```

```
ErrorLog Err.Number, Err.Description, "mdlMain: OPC_Connect"
Resume Next
```

```
End Sub
```

---

```
'-----Gruppe lesen
```

```
Function ReadFromPLC1()
Dim NoErrors1 As Double
```

```
On Error GoTo ErrorHandler
```

```
NoErrors1 = mOPCGroup(1).ReadFromPLC(TempArray2())
```

```
Exit Function
ErrorHandler:
```

```
MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
vbCritical, "Fehler"
```

```
ErrorLog Err.Number, Err.Description, "mdlMain: Function ReadFromPLC1"
Resume Next
```

```
End Function
```

---

```
'-----Funktion für MsgBox ohne zeitliche Unterbrechung
```

```
'Messagebox mit TimeOut
Public Function MsgBoxT(Frm As Form, Prompt As String, _
Style As VbMsgBoxStyle, Titel As String, _
Optional mTimer As Timer = Nothing, _
Optional Show_MSeconds As Long = 0) As Long
```

```
If Not mTimer Is Nothing Then
mTimer.Interval = Show_MSeconds
mTimer.Enabled = True
End If
```

```
MsgBoxT = MsgBox(Frm.hwnd, Prompt, Titel, Style)
```

```
If Not mTimer Is Nothing Then
mTimer.Enabled = False
End If
```

```
End Function
```

---

## '-----Gruppe schreiben

```

Public Sub writeValuesOPC(Block1 As String, Adress1 As String, lenght1 As String, Value1 As Variant, Optional Block2 As
String, Optional Adress2 As String, Optional lenght2 As String, Optional Value2 As Variant)
Dim ItemsCount As Integer, strItems(2) As String, NoErrors As Boolean, mvValues(5) As Variant
ReDim Preserve mOPCGroup(miGroupsCount + 1)

    On Error GoTo ErrorHandler

    If Block1 = "" Or Adress1 = "" Or lenght1 = "" Then Exit Sub
    strItems(1) = "S7:[" & mstrOPCConnection & "]" & Block1 & "," & Adress1 & "," & lenght1
    ItemsCount = 1
    If Not (Block2 = "" Or Adress2 = "" Or lenght2 = "") Then
        strItems(2) = "S7:[" & mstrOPCConnection & "]" & Block2 & "," & Adress2 & "," & lenght2
        ItemsCount = 2
    End If
    Set mOPCGroup(miGroupsCount + 1) = New clsOPCGroup
    mOPCGroup(miGroupsCount + 1).OPCServerName = "OPC.SimaticNet"
    mOPCGroup(miGroupsCount + 1).OPCGroupName = "TempGroup"
    mOPCGroup(miGroupsCount + 1).OPCUpdateRate = 1000           'update rate for each opc group
    mOPCGroup(miGroupsCount + 1).OPCItemsCount = ItemsCount     'number of items in the opc group
    NoErrors = mOPCGroup(miGroupsCount + 1).AddOPCItems(strItems()) 'add one group with x items

    If NoErrors Then
        mvValues(1) = Value1
        mvValues(2) = Value2
        mOPCGroup(miGroupsCount + 1).WriteToPLC mvValues()
    End If
    mOPCGroup(miGroupsCount + 1).RemoveAllOPCItems
    Set mOPCGroup(miGroupsCount + 1) = Nothing

Exit Sub
ErrorHandler:

MsgBox "Fehlernummer " & Err.Number & Chr$(13) & Error$(Err), _
    vbCritical, "Fehler"

ErrorLog Err.Number, Err.Description, "mdlMain: writeValuesOPC"
Resume Next

End Sub

```



## Quellcode in mdlError

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
'
'-----
'
' Source Code Administration Information
'
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
'
'-----
'
' File description
'
' PROGRAM:  Reglertuning
'
' FILENAME: mdlError.bas
'
' FUNCTIONS: -ErrorLog – Erzeugen einer Error-Datei
'           -OPCLog – Datei erzeugen, wenn ein Fehler bei der OPC-Verbindung entsteht
'
' COMMENTS:
'
' MODIFICATIONS:
'
'-----
'-----
Option Explicit

'----- Erzeugen einer Error-Datei

Public Sub ErrorLog(IErrNumber&, Optional strDescription$, Optional strSubName$)
Dim iFile%, strPath$, strLogFile$, strErrorText$
Static IErrors&
    On Error GoTo ErrorHandler

    IErrors = IErrors + 1

    strPath = App.Path & "\Logging\"
    strLogFile = strPath & Format(Now(), "yyyy-mm-dd", vbMonday) & "_Error.txt"

    strErrorText = "Function:" & vbTab & strSubName & vbCrLf & _
        "Time:" & vbTab & vbTab & Now() & vbCrLf & _
        "Error count:" & vbTab & IErrors & vbCrLf & _
        "Error:" & vbTab & vbTab & IErrNumber & vbCrLf & _
        "Description:" & vbTab & strDescription

    iFile = FreeFile
    Open strLogFile For Append As #iFile
    Print #iFile,
'-----
'Print #iFile, "ComputerName : "; gsLocalHostName
Print #iFile, strErrorText
Print #iFile, vbCrLf
Close #iFile

Exit Sub

ErrorHandler:
If Err.Number = 76 Then ' path not exist
    Mkdir strPath
    Resume
Else
    MsgBox "Error: " & Err.Number & vbCrLf & vbCrLf & Err.Description, vbCritical, "Recipe OPC"
End If
On Error Resume Next
Resume Next
End Sub
```

'----- Datei erzeugen, wenn ein Fehler bei der OPC-Verbindung entsteht

```
Public Sub OPCLog(strGroup$, Optional strItem$)
Dim iFile%, strPath$, strLogFile$, strLogText$
Static ILogs&
    On Error GoTo ErrorHandler

    ILogs = ILogs + 1

    strPath = App.Path & "\Logging\"
    strLogFile = strPath & Format(Now(), "yyyy-mm-dd", vbMonday) & "_OPC.txt"

    strLogText = "Group:" & vbTab & vbTab & strGroup & vbCrLf & _
        "Item:" & vbTab & vbTab & strItem & vbCrLf & _
        "Time:" & vbTab & vbTab & Now() & vbCrLf & _
        "Logging count:" & vbTab & ILogs & vbCrLf

    iFile = FreeFile
    Open strLogFile For Append As #iFile
    Print #iFile,
"-----"
    'Print #iFile, "ComputerName : "; gsLocalHostName
    Print #iFile, strLogText
    Print #iFile, vbCrLf
    Close #iFile

    Exit Sub

ErrorHandler:
If Err.Number = 76 Then ' path not exist
    Mkdir strPath
    Resume
Else
    MsgBox "Error: " & Err.Number & vbCrLf & vbCrLf & Err.Description, vbCritical, "Recipe OPC"
End If
On Error Resume Next
Resume Next
End Sub
```

## Quellcode in mdLogFileGeneration

```
' Tuchenhagen Dairy Systems, Germany
' branch office Büchen
'
'-----
'
' Source Code Administration Information
'
' AUTHOR:   Axel Kern
' DATE:    19.09.06
' MODTIME: 19.09.06
' VERSION: 1.0.0
'
'-----
'
' File description
'
' PROGRAM:  Reglertuning
'
' FILENAME: mdLogFileGeneration.bas
'
' FUNCTIONS: - LogFile - create a logfile
'
'
' COMMENTS:
'
' MODIFICATIONS:
'
'-----
*****

Option Explicit

'----- Datei erzeugen und gewünschte Daten aus den Prozeduren in die Text-Datei schreiben

Public Sub LogFileGeneration(LoopName$, Optional strDescription$, Optional strSubName$)
Dim iFile%, strPath$, strLogFile$, strLogFileGenerationText$
Static ILogFileGeneration&
    On Error GoTo ErrorHandler

    ILogFileGeneration = ILogFileGeneration + 1

    strPath = App.Path & "\Log-File\"
    strLogFile = strPath & Format(Now(), "yyyy-mm-dd", vbMonday) & "_Log-File.txt"

    strLogFileGenerationText = vbCrLf & LoopName & vbCrLf & vbCrLf

    iFile = FreeFile
    Open strLogFile For Append As #iFile
    Print #iFile, "-----"
'    Print #iFile, "ComputerName : "; gsLocalHostName
    Print #iFile, strLogFileGenerationText
    Print #iFile, vbCrLf
    Close #iFile

Exit Sub

ErrorHandler:
If Err.Number = 76 Then ' path not exist
    Mkdir strPath
    Resume
Else
    MsgBox "Error: " & Err.Number & vbCrLf & vbCrLf & Err.Description, vbCritical, "Recipe OPC"
End If
On Error Resume Next
Resume Next
End Sub
```

## Erklärung zur Diplomarbeit

Name: Kern  
Vorname: Axel  
Matr.-Nr.: 1155246

Studiengang: Angewandte Automatisierungstechnik

An den Prüfungsausschuss  
des Fachbereichs Automatisierungstechnik  
der Universität Lüneburg  
Volgershall 1  
21339 Lüneburg

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Lüneburg, den 21.09.2006

.....

## SIMATIC

### Standard PID Control

#### Handbuch

Diese Dokumentation ist Bestandteil des Dokumentationspaketes  
**6ES7 830-2AA21-8AG0**

**Ausgabe 03/2003**

A5E00204508-02

Vorwort, Inhaltsverzeichnis

---

#### **Funktionsbausteine Standard PID Control**

---

Produktübersicht  
Standard PID Control **1**

---

Entwerfen von digitalen Regelungen **2**

---

Konfigurieren und Inbetriebsetzen  
von Standard PID Control **3**

---

Signalverarbeitung im Soll-/Istwertkanal und die PID-Reglerfunktionen **4**

---

Der kontinuierliche Regler  
(PID\_CP) **5**

---

Der Schrittreger  
(PID\_ES) **6**

---

Regleraufrufverteiler und Beispiele für Reglerkonfigurationen **7**

---

Technische Daten und Blockschaltbilder **8**

---

Parameterlisten von Standard PID Control **9**

---

#### **Konfiguration Standard PID Control**

---

Konfigurationswerkzeug (Tool) für Standard PID Control **10**

---

#### **Anhänge**

---

Literaturverzeichnis **A**

---

Glossar, Index

## Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgrad folgendermaßen dargestellt:



### Gefahr

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



### Warnung

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



### Vorsicht

bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

### Vorsicht

bedeutet, daß ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

### Achtung

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

## Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

## Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:



### Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

## Marken

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind Marken der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

## Copyright © Siemens AG 1999 - 2003 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

Siemens AG  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D- 90327 Nürnberg

Siemens Aktiengesellschaft

## Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 2003  
Technische Änderungen bleiben vorbehalten.

A5E00204508-02



# Vorwort

## Zweck des Handbuchs

Dieses Handbuch unterstützt Sie bei der Auswahl, Konfiguration, Parametrierung eines für Ihre Regelaufgabe optimalen Reglerbausteines.

Sie werden mit der Funktionsweise des Reglerbausteines sowie der Hantierung des Konfigurationswerkzeuges vertraut gemacht.

## Erforderliche Grundkenntnisse

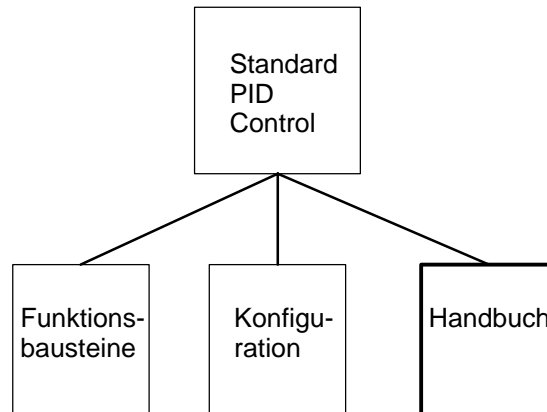
Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik und der Regelungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter dem Betriebssystem Windows 95/98/2000 bzw. NT vorausgesetzt. Da Standard PID Control auf der Basissoftware STEP 7 aufsetzt, sollten Sie auch Kenntnisse im Umgang mit der Basissoftware haben. Diese werden im Handbuch "Programmieren mit STEP 7 V5.1" vermittelt.

## Gültigkeitsbereich des Handbuchs

Das Handbuch ist gültig für die Pakete Standard PID Control FB V5.1 und Standard PID Control Tool V5.1.

## Einordnung in die Informationslandschaft



Das Softwarepaket "Standard-PID Control" umfasst drei Einzelprodukte:

- Das Produkt "Standard PID Control FB" umfasst hauptsächlich die beiden Reglerbausteine PID\_CP und PID\_ES .
- Das Produkt "Standard PID Control Tool" enthält hauptsächlich die Werkzeuge zum Konfigurieren der Reglerbausteine.

Das Produkt wird im Folgenden "Konfigurationswerkzeug" genannt.

- Dieses Handbuch ist ein eigenständiges Produkt und beschreibt sowohl das Produkt "Standard PID Control FB" als auch das Konfigurationswerkzeug "Standard PID Control Tool".

### Das Softwarepaket "Standard PID Control"

Das Softwarepaket "Standard PID Control" bietet ein umfassendes Konzept zur Realisierung von Regelfunktionen in den Automatisierungssystemen SIMATIC S7.

Der Regler ist in seinem vollen Leistungsumfang und mit allen Eigenschaften zur Signalverarbeitung fertig programmiert. Zur Anpassung einer Regelung an Ihre Prozessgegebenheiten müssen Sie lediglich gewünschte Teilfunktionen innerhalb dieser maximalen Funktionalität auswählen. Der Projektierungsaufwand schrumpft also auf das Weglassen (Ausblenden) nicht benötigter Funktionen zusammen. Bei diesen Arbeiten werden Sie durch das Konfigurationswerkzeug unterstützt.

Wegen der auf die Wahl - evtl. auch Ergänzung - von Grundfunktionen beschränkten Konfiguriermöglichkeiten ist das Konzept von Standard PID Control leicht beherrschbar. Auch der Anwender mit durchschnittlichen regelungstechnischen Kenntnissen ist in der Lage, in kurzer Zeit eine qualitativ hochwertige Regelung zu erstellen.



## **Wegweiser**

Das vorliegende Handbuch ist nach folgenden Themenbereichen gegliedert:

- Kapitel 1 gibt Ihnen eine Übersicht über Standard PID Control.
- Kapitel 2 zeigt Ihnen die Struktur und Arbeitsweise von Standard PID Control.
- Kapitel 3 hilft Ihnen beim Entwurf und Inbetriebsetzen von Standard PID Control.
- Kapitel 4 zeigt Ihnen die Signalverarbeitung im Soll-Istwertkanal und im Regler.
- Kapitel 5 zeigt Ihnen die Signalverarbeitung im kontinuierlichen Reglerausgang.
- Kapitel 6 zeigt Ihnen die Signalverarbeitung im Schrittreglerausgang.
- Kapitel 7 zeigt Ihnen das Arbeiten mit dem Regleraufrufverteiler und beispielhafte Reglerstrukturen.
- Kapitel 8 informiert Sie über die technischen Daten und Blockschaltbilder.
- Kapitel 9 informiert Sie über die Parameterlisten von Standard PID Control.
- Kapitel 10 gibt Ihnen eine Übersicht über das Konfigurationswerkzeug.
- Anhang A enthält das Literaturverzeichnis.
- Im Glossar sind wichtige Begriffe erklärt.
- Das Stichwortverzeichnis hilft Ihnen, Textstellen zu wichtigen Stichworten schnell zu finden.

## **Leserkreis**

- S7-Programmierer
- Regelungsprogrammierer
- Bedienpersonal
- Servicepersonal

## Lesekonventionen

Damit Sie die Informationen in diesem Handbuch leichter finden können, wurden bestimmte Konventionen festgelegt:

- Überfliegen Sie zunächst die Überschriften und die Titel am linken Rand, um schnell Hinweise auf den Inhalt zu erhalten.
- Ein Themenblock beantwortet entweder eine Fragestellung zur Funktionalität des Werkzeuges oder gibt Auskunft über erforderliche bzw. empfohlene Abläufe.
- Verweise auf weitere Behandlung eines Themas in anderen Kapiteln werden mit (*siehe Kapitel x.y*) dargestellt. Hinweise auf andere Dokumentationen sind mit Hilfe von Literaturnummern in Schrägstrichen /.../ angegeben. Anhand dieser Nummern können Sie dem Literaturverzeichnis am Ende des Handbuchs den genauen Titel der Dokumentation entnehmen.
- Handlungsanweisungen werden durch schwarze Punkte markiert.
- Handlungsabläufe werden durch Ordnungszahlen markiert oder explizit durch Schrittfolgen angegeben.
- Alternativen bei Handlungen oder Entscheidungen werden mit Strich markiert.

## Weitere Informationen

Dieses Handbuch ist als Nachschlagewerk konzipiert, das Ihnen Informationen zum Thema Standard-Regelungen liefert. Sie benötigen allerdings eventuell weitere Informationen, die Sie den Handbüchern */70/, /71/, /100/, /101/, /231/, /232/, /234/, /352/* entnehmen.

## Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

<http://www.siemens.com/automation/partner>

## Trainingscenter

Um Ihnen den Einstieg in das xxx und das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.

Telefon: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

## A&D Technical Support

Weltweit erreichbar zu jeder Tageszeit:



<b>Weltweit (Nürnberg)</b> <b>Technical Support</b>  Ortszeit: 0:00 bis 24:00 / 365 Tage Telefon: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@siemens.com GMT: +1:00		
<b>Europa / Afrika (Nürnberg)</b> <b>Authorization</b>  Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@siemens.com GMT: +1:00	<b>United States (Johnson City)</b> <b>Technical Support and Authorization</b>  Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +1 (0) 423 262 2522 Fax: +1 (0) 423 262 2289 E-Mail: simatic.hotline@sea.siemens.com GMT: -5:00	<b>Asien / Australien (Peking)</b> <b>Technical Support and Authorization</b>  Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +86 10 64 75 75 75 Fax: +86 10 64 74 74 74 E-Mail: adsupport.asia@siemens.com GMT: +8:00
Technical Support und Authorization sprechen generell Deutsch und Englisch.		

## **Service & Support im Internet**

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

<http://www.siemens.com/automation/service&support>

Dort finden Sie:

- der Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort über unsere Ansprechpartner-Datenbank.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Begriff "Leistungen" bereit.

# Inhaltsverzeichnis

	<b>Vorwort</b> .....	<b>iii</b>
	<b>Inhaltsverzeichnis</b> .....	<b>ix</b>
<b>1</b>	<b>Produktübersicht Standard PID Control</b> .....	<b>1-1</b>
1.1	Das Produkt "Standard PID Control" .....	1-1
1.2	Das Softwareprodukt "Standard PID Control" .....	1-3
1.3	Einsatzumgebung und Anwendungsbereich .....	1-5
<b>2</b>	<b>Entwerfen von digitalen Regelungen</b> .....	<b>2-1</b>
2.1	Prozesseigenschaften und Regelung .....	2-1
2.2	Identifikation der Streckeneigenschaften .....	2-5
2.3	Störgrößenaufschaltung .....	2-7
2.4	Mehrschleifige Regelungen .....	2-8
2.5	Struktur und Arbeitsweise von Standard PID Control .....	2-11
2.6	Signalflusspläne .....	2-15
<b>3</b>	<b>Konfigurieren und Inbetriebsetzen von Standard PID Control</b> .....	<b>3-1</b>
3.1	Definieren der Regelaufgabe .....	3-1
3.2	Projekt konfigurieren (Checkliste) .....	3-7
3.3	Parametrieren von Standard PID Control .....	3-10
3.4	Die Abtastzeit CYCLE .....	3-14
3.5	Das Aufrufverhalten von Standard PID Control .....	3-16
3.6	Wertebereiche und Signalanpassung (Normierung) .....	3-18
<b>4</b>	<b>Signalverarbeitung im Soll-/Istwertkanal und die PID-Reglerfunktionen</b> ....	<b>4-1</b>
4.1	Signalverarbeitung im Sollwertzweig .....	4-1
4.1.1	Sollwert-Generator (SP_GEN) .....	4-1
4.1.2	Zeitplangeber (RMP_SOAK) .....	4-3
4.1.3	Normierung des externen Sollwertes (SP_NORM) .....	4-12
4.1.4	FC-Aufruf im Sollwertzweig (SPFC) .....	4-15
4.1.5	Begrenzung der Änderungsgeschwindigkeit des Sollwertes (SP_ROC) .	4-17
4.1.6	Begrenzung des Absolutwertes der Führungsgröße (SP_LIMIT) .....	4-19
4.1.7	Sollwertverstellung über das Konfigurationswerkzeug .....	4-21
4.2	Signalverarbeitung im Istwertzweig .....	4-22
4.2.1	Normierung der Istwerteingabe .....	4-22
4.2.2	Glättung der Regelgröße (LAG1ST) .....	4-24
4.2.3	Bildung der Quadratwurzel (SQRT) .....	4-26
4.2.4	FC-Aufruf im Istwertzweig (PVFC) .....	4-28

4.2.5	Überwachung des Istwertes auf Grenzwerte (PV_ALARM) .....	4-30
4.2.6	Überwachung der Änderungsgeschwindigkeit des Istwertes (ROCALARM) .....	4-32
4.2.7	Istwertverstellung über das Konfigurationswerkzeug .....	4-34
4.3	Signalverarbeitung der Regeldifferenz (Regelabweichung) .....	4-35
4.3.1	Signalfilterung durch Totzonenfunktion (DEADBAND) .....	4-35
4.3.2	Überwachung der Regeldifferenz auf Grenzwerte (ER_ALARM) .....	4-37
4.4	Die PID-Reglerfunktionen .....	4-39
4.5	Signalverarbeitung im PID-Regleralgorithmus .....	4-46
4.5.1	Integrierer (INT) .....	4-46
4.5.2	Differenzierer (DIF) .....	4-51
<b>5</b>	<b>Der kontinuierliche Regler (PID_CP) .....</b>	<b>5-1</b>
5.1	Regelfunktionen des kontinuierlichen PID-Reglers .....	5-1
5.2	Signalverarbeitung der Stellgröße .....	5-3
5.2.1	Betriebsarten der Stellsignalbildung .....	5-3
5.2.2	Betriebsarten der Stellsignalbildung .....	5-5
5.2.3	FC-Aufruf im Stellgrößenzweig (LMNFC) .....	5-7
5.2.4	Begrenzung der Änderungsgeschwindigkeit des Stellwertes (LMN_ROC) .....	5-9
5.2.5	Begrenzung des Absolutwertes der Stellgröße (LMNLIMIT) .....	5-11
5.2.6	Normierung der Stellgröße auf das Format einer physikalischen Größe (LMN_NORM) .....	5-13
5.2.7	Stellwertausgabe im Peripherieformat (CRP_OUT) .....	5-15
5.2.8	Stellwertbeeinflussung über das Konfigurationswerkzeug .....	5-16
5.3	Kontinuierliche Regler in Kaskadenschaltungen .....	5-17
5.4	Impulsformerstufe (PULSEGEN) .....	5-19
<b>6</b>	<b>Der Schrittreger (PID_ES) .....</b>	<b>6-1</b>
6.1	Regelfunktionen des PID-Schrittregers .....	6-1
6.2	Stellgrößen beim Schrittreger mit Stellungsrückmeldung .....	6-5
6.2.1	Betriebsarten des Schrittregers .....	6-5
6.2.2	Stellwertbeeinflussung über das Konfigurationswerkzeug .....	6-8
6.2.3	Begrenzung des Absolutwertes der Stellgröße (LMNLIMIT) .....	6-9
6.2.4	Verarbeitung der Stellungsrückmeldung (LMNR_IN bzw. LMNR_PER) .....	6-11
6.2.5	Erzeugung der Stellsignale (QLMNUP/QLMNDN) .....	6-14
6.3	Stellgrößenverarbeitung beim Schrittreger ohne Stellungsrückmeldung .....	6-18
6.4	Schrittreger in Kaskadenschaltungen .....	6-25
<b>7</b>	<b>Regleraufrufverteiler und Beispiele für Reglerkonfigurationen .....</b>	<b>7-1</b>
7.1	Regleraufrufverteiler (LP_SCHED) .....	7-1
7.2	Example1: Schrittreger mit Streckensimulation .....	7-10
7.3	Example2: Kontinuierlicher Regler mit Streckensimulation .....	7-16
7.4	Example3: Mehrschleifige Verhältnisregelung .....	7-21
7.5	Example4: Mischungsregelung .....	7-24

7.6	Example5: Kaskadenregelung .....	7-27
7.7	Example6: Pulsegen: Kontinuierlicher Regler mit Impulsausgängen und Streckensimulation .....	7-30
<b>8</b>	<b>Technische Daten und Blockschaltbilder .....</b>	<b>8-1</b>
8.1	Technische Daten: Funktionsbausteine .....	8-1
8.2	Blockschaltbilder von Standard PID Control .....	8-3
<b>9</b>	<b>Parameterlisten von Standard PID Control .....</b>	<b>9-1</b>
9.1	Parameter des Funktionsbausteins PID_CP .....	9-2
9.2	Parameter des Funktionsbausteins PID_ES .....	9-11
9.3	Parameter der Funktion LP_SCHED .....	9-20
<b>10</b>	<b>Projektiersoftware für Standard PID Control .....</b>	<b>10-1</b>
<b>A</b>	<b>Literaturverzeichnis .....</b>	<b>A-1</b>
	<b>Glossar .....</b>	<b>Glossar-1</b>
	<b>Index .....</b>	<b>Index-1</b>





# Produktübersicht Standard PID Control

# 1

## 1.1 Das Produkt "Standard PID Control"

### Konzept von "Standard PID Control"

Das Softwareprodukt "Standard PID Control" besteht im Wesentlichen aus zwei **Funktionsbausteinen** (FBs), in denen die Algorithmen zur Bildung der Regel- und Signalverarbeitungs-Funktionen für kontinuierliche bzw. für Schrittreger enthalten sind. Es handelt sich also um eine reine Softwareregung, bei der ein Standard-Funktionsbaustein die komplette Funktionalität des Reglers verkörpert.

Das Verhalten des Reglers selbst und die Eigenschaften der Funktionen im Mess- und Stellkanal werden durch die numerischen Algorithmen des Funktionsbausteins realisiert bzw. nachgebildet. Die für diese zyklischen Berechnungen benötigten Daten sind in regelkreisspezifischen Datenbausteinen hinterlegt. Zur Erzeugung mehrerer Regler wird ein FB nur einmal benötigt.

Jeder Regler wird durch einen **Instanz-DB** repräsentiert, welcher applikationsabhängig zu erstellen ist. Bei Nutzung des Werkzeugs "Standard PID Control Tool" geschieht diese DB-Erstellung 'implizit'. Das bedeutet, die Auslegung eines spezifischen Reglers beschränkt sich auf das Festlegen der Struktur- und Wertparameter in den Bearbeitungsfenstern der Bedienoberfläche. Der Instanz-DB wird vom Konfigurationswerkzeug erstellt.

Die Berechnung der Algorithmen für einen bestimmten Regler erfolgt im Prozessor des S7-Automatisierungssystems (AS), und zwar in den eingestellten Zeitintervallen (Abtastzeiten). Die Berechnungsergebnisse und damit die aktualisierten Werte von Ein- und Ausgangsvariablen (Mess- und Stellgrößen) und Zustandsmeldungen (Grenzwerte) werden im zugehörigen Instanz-DB hinterlegt bzw. an die Prozessperipherie übergeben.

Bei Bearbeitung von vielen Regelkreisen, die je nach Trägheit der jeweiligen Regelstrecke unterschiedlich oft - dabei aber äquidistant - bearbeitet werden sollen, steht ein **Regleraufrufverteiler** (Loop Scheduler = LP\_SCHED) zur Verfügung, mit dessen Hilfe die Konfiguration umfangreicher Anlagenregelungen sehr übersichtlich und deshalb einfach wird. Außerdem wird eine gleichmäßige Auslastung der CPU gewährleistet.

## Übersicht über die Grundfunktionen

Bei vielen Regelungsaufgaben steht nicht allein der klassische PID-Regler als prozessbeeinflussendes Element im Vordergrund, sondern es werden auch hohe Anforderungen an die Signalverarbeitung gestellt.

Ein mit Hilfe des Softwarepakets "Standard PID Control" gebildeter Regler setzt sich deshalb aus einer Reihe von Teilfunktionen zusammen, die von Ihnen separat parametrierbar sind. Zusätzlich zum eigentlichen Regler mit dem PID-Algorithmus sind auch Funktionen zur Aufbereitung des Sollwerts und des Istwerts sowie zur Nachbearbeitung der errechneten Stellgröße integriert.

Hinzu kommen Anzeige- und Überwachungsfunktionen (im Übersichtsschema nicht dargestellt).

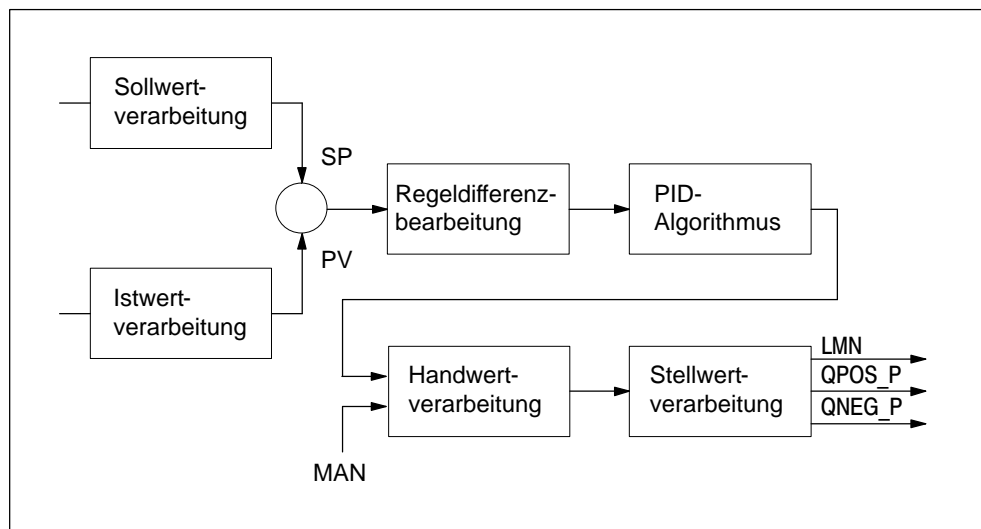


Bild 1-1 Funktionsübersicht des Softwarebausteins "Kontinuierlicher Regler"

## Erstellen der Regelung

Mit dem Softwarepaket "Standard PID Control" lässt sich ein Regler für eine bestimmte Regelungsaufgabe konfigurieren. Sein Funktionsvorrat kann eingeschränkt projektiert werden. Mittels sogenannter Strukturschalter können Teilfunktionen zu- bzw. abgeschaltet oder ganze Zweige unwirksam gemacht werden. Zu parametrieren sind anschließend nur noch die in der reduzierten Struktur verbliebenen Funktionsteile.

Die Erstellung einer Regelung von der Strukturierung über die Parametrierung bis hin zum zeitgerechten Aufruf durch das Systemprogramm ist weitgehend ohne Programmierung möglich. STEP 7-Kenntnisse sind erforderlich.

Die Struktur des Instanz-DB ist in *Kapitel 9* dieses Handbuchs offengelegt. Jedem Struktur- bzw. Wert-Parameter ist ein Datum d. h. eine Zeile vorbehalten. Durch Editieren der entsprechenden Eintragungen lassen sich sowohl Struktur als auch die gewünschten Eigenschaften der Regelung festlegen.

Diese Vorgehensweise wird jedoch wegen ihrer Unübersichtlichkeit nicht empfohlen. Das speziell für Standard PID Control konzipierte Konfigurationswerkzeug vereinfacht diese Aufgabe ganz wesentlich.

**Hinweis**

Das Konfigurationswerkzeug lässt sich nicht zur Parametrierung des Bausteins LP-SCHED einsetzen. Dessen Funktionalität wird ausschließlich durch Eingaben in den betreffenden Datenbaustein definiert.

## 1.2 Das Softwareprodukt "Standard PID Control"

### Produktstruktur: "Standard PID Control"

Nach der Installation des Produkts "Standard PID Control" finden Sie auf Ihrem PG/PC eine STEP 7-Bausteinbibliothek mit dem Namen "Standard PID Control". Diese enthält zwei Standard-Funktionsbausteine, eine Standard-Funktion, Mustervorlagen für Datenbausteine sowie das STEP 7-Projekt "zDt28\_03\_StdCon" mit 6 Beispielen und einem Getting Started.

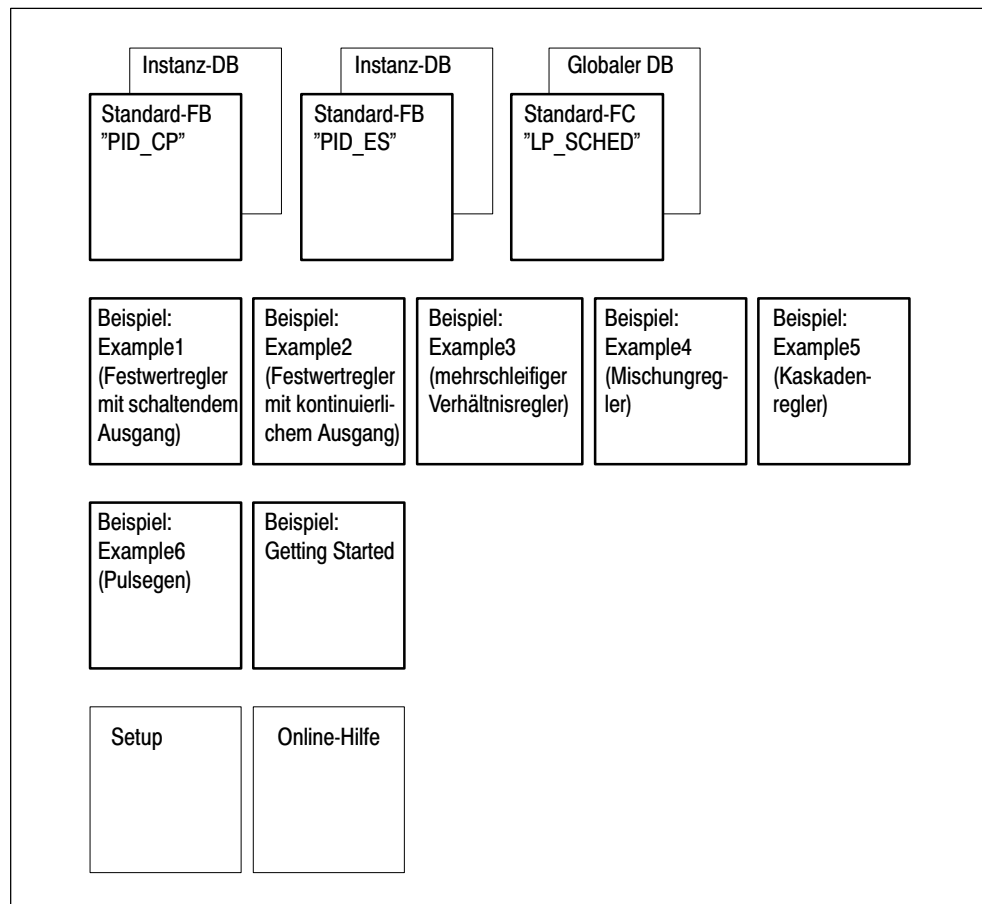


Bild 1-2 Inhalt des Softwarepaketes "Standard PID Control"

- Der Std-FB **PID\_CP** enthält alle regelungstechnischen Funktionen eines kontinuierlichen PID-Reglers einschließlich eines Impulsausgangs für proportionale Stellglieder.
- Der Std-FB **PID\_ES** enthält alle regelungstechnischen Funktionen eines PID-Reglers mit Dreipunkt-Schrittausgang.
- Die Std-FC **LP\_SCHED** steuert die Aufrufverteilung der einzelnen Regler innerhalb eines Weckalarm-OB bei Anwendungen mit vielen Regelkreisen. Der Baustein übernimmt auch die Initialisierung der Reglerstruktur beim Anlauf der CPU bzw. des AS.

Außerdem enthält das Softwarepaket ein **Setup-Programm** für die Installation der "Standard PID Control" auf PG/PC sowie eine **Online-Hilfe**, die Informationen (Nachschlagewissen) zu Teilfunktionen und Einzelparametern während der praktischen Aufgabenbearbeitung verfügbar macht.

### Vorkonfektionierte Anwendungs-Strukturen

Der Lieferumfang der "Standard PID Control" wird durch Datenstrukturen (Instanz-DBs) für die am häufigsten eingesetzten Reglertypen bzw. für die wichtigsten mehrschleifigen Regelungen ergänzt.

Auf diese betriebsfertig vorbereiteten Struktur-Beispiele (Example1 ... Example6) können Sie zurückgreifen, wenn Ihnen die Erstellung eines Reglers von Grund auf zu aufwändig erscheint oder wenn Sie bei der Erstellung von gekoppelten Reglerstrukturen Fehler vermeiden wollen.

Folgende Beispiel-Strukturen stehen zur Verfügung:

Kurzbezeichnung	Bereitgestellte Funktionalität	Kommentar
Example1	Festwertregler mit schaltendem Ausgang - für integrierende Stellglieder (z. B. Motorantriebe)	"PID-Schrittregler" mit Dreipunktverhalten
Example2	Festwertregler mit kontinuierlichem Ausgang - für proportionale Stellglieder	"Analog-PID-Regler"
Example3	Mehrschleifige Verhältnisregelung	Das Verhältnis zweier Regelgrößen wird konstant gehalten.
Example4	Mischungsregelung	Die zu mischenden Komponenten werden auf einen konstanten %-Satz gehalten und die Gesamtmenge wird geregelt.
Example5	Kaskadenregelung	Verbesserung des Regelverhaltens durch Einbeziehen von Regelgrößen in unterlagerten Regelkreisen
Example6	Kontinuierliche Regler mit Impulsausgängen und Streckensimulation	

### "Konfiguration von Standard PID Control"

Die Funktionen des Softwarepakets "Konfiguration Standard PID Control" werden im *Kapitel 10* dieses Handbuches beschrieben.

## 1.3 Einsatzumgebung und Anwendungsbereich

### Hardwareumgebung

Die mit dem Softwarepaket "Standard PID Control" erzeugten Regler sind auf folgenden Zielsystemen ablauffähig

- S7-300- und S7-400 (CPU mit Gleitpunkt und Weckalarm)
- C7-CPU's
- Win AC

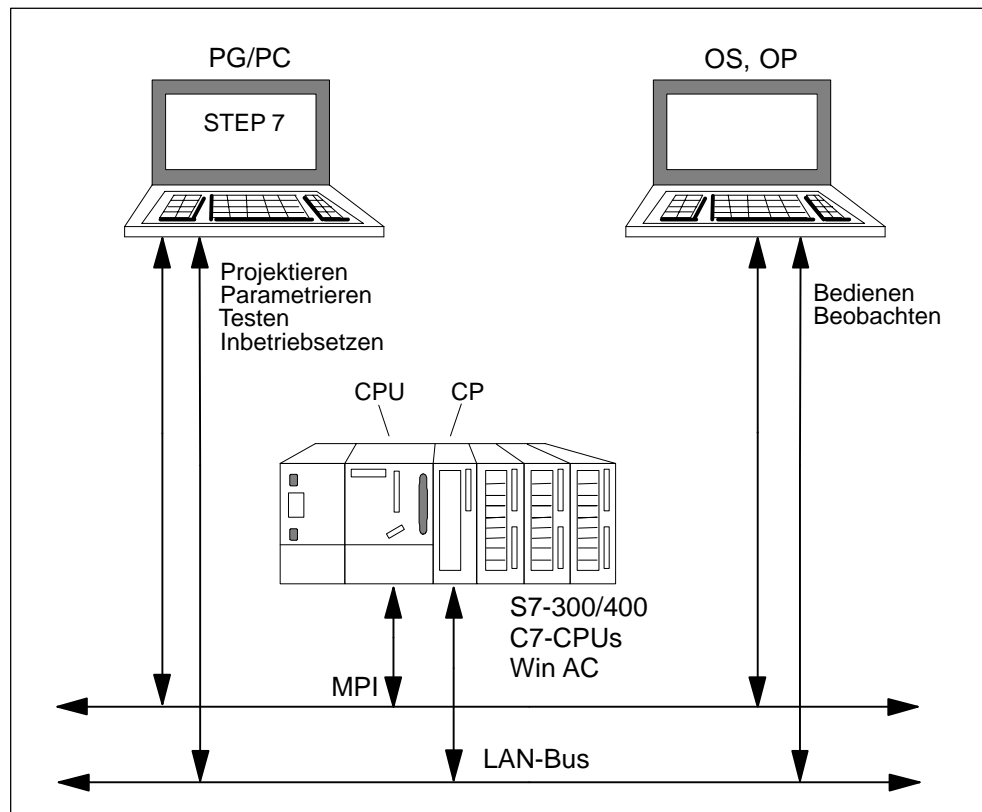


Bild 1-3 Einsatzumgebung des SW-Pakets "Standard PID Control"

### Softwareumgebung

Das Softwarepaket "Standard PID Control" ist für den Einsatz in der Programmgruppe STEP 7 konzipiert.

Die Erstellungssoftware für Standard-Regelungen kann lokal auf einem PG/PC oder aber in einem Netz auf einem zentralen Netzlaufwerk installiert werden.

## Der Systemrahmen

Da es sich bei der digitalen Realisierung von Reglerfunktionen immer um rechenaufwändige Operationen handelt (Wortverarbeitung), ist es wichtig, sich rechtzeitig ein Bild von der Belastung der zur Verfügung stehenden CPU zu machen. Dafür gibt es folgende Anhaltspunkte:

- Codeumfang eines Funktionsbausteins  
(PID\_CP bzw. PID\_ES):  $\leq 8$  KByte
- Daten pro Regler  $\leq 0,5$  KByte
- Eckdaten für minimale Laufzeiten (Bearbeitungszeiten) eines PID-Reglers auf unterschiedlichen Automatisierungssystemen finden Sie in *Kapitel 8.1* (Technische Daten).
- Die Größe des erforderlichen Bereichs im Anwenderspeicher und damit die theoretisch installierbare Anzahl der Regelkreise aufgrund der verfügbaren Speicherkapazität (bei 50 % Ausnutzung des Arbeitsspeichers durch die Regelungsaufgaben) ist den Technischen Daten (*siehe Kapitel 8.1*) zu entnehmen.
- Speicherbedarf für einen L-Stack besteht nicht.
- Alarme werden durch die Bearbeitung des Regelungs-FB nicht verzögert.

## Regleraufrufverteiler

Müssen viele Regler oder Regler mit großen Abtastzeiten aufgerufen werden, reicht der Umfang des Prioritätsklassenmodells in Bezug auf Weckalarm-OBs nicht aus. Mit dem Regleraufrufverteiler LP\_SCHED (Loop Scheduler) können in einem Weckalarm-OB mehrere Regler mit verschiedenen Abtastzeiten äquidistant aufgerufen werden.

Die Aufgaben des Aufrufverteilers sind:

- Steuerung des zeitlichen Aufrufs der einzelnen Regler innerhalb einer (Weckalarm-) Prioritätsklasse.
- Aufruf der installierten Standard-Regler beim Anlauf der CPU.

## Einsatzmöglichkeiten und -grenzen von Standard PID Control

Eine durch die Abarbeitung eines FB verkörperte Regelfunktion ist in Bezug auf ihre möglichen Einsatzfelder grundsätzlich neutral. Die Regelleistung und damit die Schnelligkeit der Bearbeitung von konkreten Regelkreisen hängt ausschließlich von der Leistung der verwendeten CPU ab.

Bei einer gegebenen CPU muss zwischen der Anzahl der Regler und der Häufigkeit, mit der die einzelnen Regler bearbeitet werden müssen, ein Kompromiss getroffen werden. Je schneller die aufgeschalteten Regelkreise sind, d. h. je öfter die Stellgrößen pro Zeiteinheit errechnet werden müssen, um so geringer ist die Anzahl der installierbaren Regler.

Die Standard-Funktionsbausteine PID\_CP und PID\_ES ermöglichen es, Software-regler zu generieren und zu betreiben, die auf dem konventionellen PID-Algorithmus von Standard PID Control basieren. Sonderfunktionen bezüglich der Verarbeitung der Prozesssignale im Regler sind nicht vorgesehen.

Einschränkungen bezüglich der Art der regelbaren Prozesse bestehen nicht. Es können sowohl träge Strecken (Temperaturen, Füllstände usw.) als auch sehr schnelle Strecken (Durchflüsse, Drehzahlen usw.) geregelt werden.

Einsatzformen von Standard PID Control:

- Festwertregelung mit P-, PI-, PD-, PID-Schrittregler;
- Festwertregelung mit kontinuierlichem P-, PI-, PD-, PID-Regler;
- Festwertregelung mit Störgrößenaufschaltung;
- Kaskadenregelung (Schrittregler nur im Folgekreis);
- Verhältnisregelung (zweischleifig);
- Mischungsregelung.

### **Funktionsspektrum von Standard PID Control**

Durch entsprechendes Projektieren der im Produkt "Standard PID Control" enthaltenen Funktionen lassen sich Regler mit nachstehenden Eigenschaften und Betriebsweisen generieren:

- Verstellung des Sollwertes durch Zeitplangeber (Zeitplanregelung);
- Begrenzung der Änderungsgeschwindigkeit des Führungsgröße und (bei Reglern mit kontinuierlichem Ausgang) der Stellgröße;
- Begrenzung der Absolutwerte der Führungsgröße und (bei Reglern mit kontinuierlichem Ausgang) der Stellgröße;
- Unterdrückung des Rauschens im Ist- oder Sollwertzweig durch Filterung des Regeldifferenz-Signals;
- Unterdrückung von höherfrequenten Schwingungen im Istwertzweig durch Glättung (Verzögerung) des Istwertsignals;
- Linearisierung von quadratischen Funktionen der Regelgröße (Durchflussregelung mit Differenzdruck-Gebern);
- Möglichkeit des Aufrufs von "eigenen Funktionen" im Sollwert-, Istwert- und/oder Stellwert-Zweig;
- Handbetrieb (Steuerung der Stellgröße von einem PG oder OP/OS aus)
- Überwachung der Regelgröße und/oder der Regeldifferenz auf Überschreitung von jeweils zwei oberen und zwei unteren Grenzwerten;
- Überwachung der Änderungsgeschwindigkeit der Regelgröße;
- Wahlweise P- und D-Wirkung in der Rückführung des Reglers.





# Entwerfen von digitalen Regelungen

# 2

## 2.1 Prozesseigenschaften und Regelung

### Prozesseigenschaften und Regler

Das statische Verhalten (Verstärkung) und die dynamischen Eigenschaften (Verzögerung, Totzeit, Integrationskonstante usw.) des zu regelnden Prozesses haben entscheidenden Einfluss auf die Art und das Zeitverhalten der Signalbearbeitung im Regler, der die jeweiligen Prozesszustände stabil halten bzw. nach vorgegebenem zeitlichen Verlauf ändern soll.

Unter den Gliedern eines Regelkreises nehmen die Regelstrecken eine Sonderstellung ein. Ihre Eigenschaften sind durch verfahrens-/maschinentechnische Gegebenheiten festgelegt und kaum zu beeinflussen. Ein gutes Reglergebnis kann also nur durch Auswahl eines für den Streckentyp am besten geeigneten Regler-typs sowie dessen Anpassung an das Zeitverhalten der Strecke erreicht werden.

Genauere Kenntnisse des Typs und der Kenndaten der Regelstrecke sind deshalb für die Auslegung bzw. den Entwurf des Reglers und der Dimensionierung seiner statischen (P-Einfluss) und dynamischen Parameter (I- und D-Einfluss) unerlässlich.

### Streckenanalyse

Für die Reglerauslegung benötigen Sie genauere Streckendaten, die Sie durch Aufnahme einer Übergangsfunktion nach Sprunganregung erhalten. Durch (graphische) Auswertung dieser (Zeit-)Funktion lassen sich dann Rückschlüsse für die Wahl der am besten geeigneten Reglerfunktion und für die Dimensionierung der einzustellenden Reglerparameter ableiten.

Bei der Durchführung dieser für eine Streckenanalyse erforderlichen Vorgänge werden Sie durch das Konfigurationswerkzeug weitgehend unterstützt.

Bevor wir jedoch die Nutzung der betreffenden Tools des Softwarepakets "Konfiguration von Standard PID Control" kurz beschreiben, im Folgenden ein Blick auf die häufigsten in technischen Prozessen und Anlagen vorkommenden Regelstrecken. Möglicherweise benötigen Sie diese Kenntnisse, um Anhaltspunkte für angemessene Vorgehensweisen bei der Analyse und ggf. auch Simulation der Prozesseigenschaften zu gewinnen.

## Typ und Charakteristik der Regelstrecke

Folgende Strecken werden näher analysiert:

- Strecke mit Ausgleich
- Strecke mit Ausgleich und Totzeit
- Strecke mit Integralwirkung

### Strecke mit Ausgleich

Die meisten Regelstrecken sind Strecken mit Ausgleich, d. h. nach einer sprungförmigen Änderung der Stellgröße nähert sich die Regelgröße nach einem Übergangsprozess einem neuen stabilen Wert, dem sogenannten Ausgleichswert. Das Zeitverhalten der Strecke kann demnach durch Aufzeichnung des zeitlichen Verlaufs der Regelgröße  $PV(t)$  nach Sprungänderung der Stellgröße  $LMN$  um einen Betrag  $> 1,5\%$  des Stellbereiches bestimmt werden.

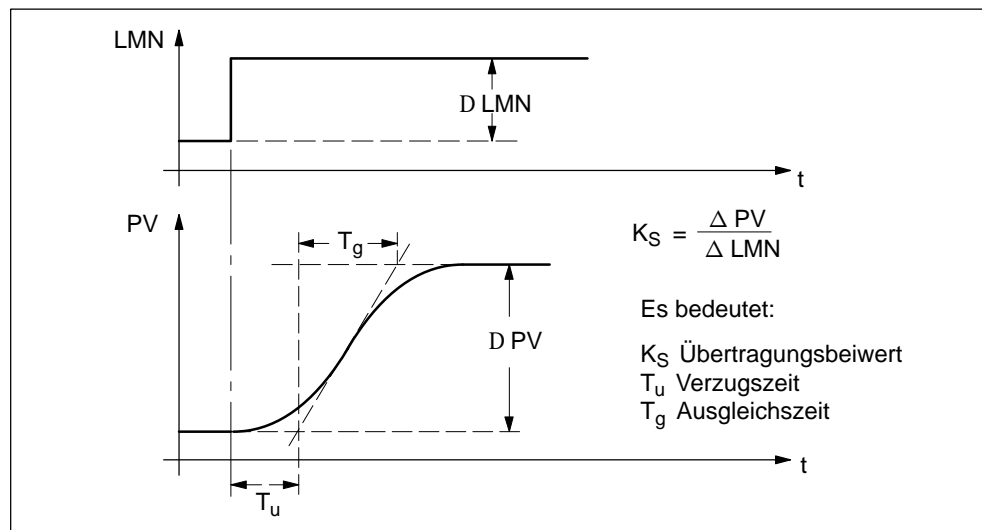


Bild 2-1 Übergangsfunktion einer Regelstrecke mit Ausgleich (PT-Strecke)

Ist das Streckenverhalten innerhalb des Stellbereiches linear, dann gibt der Übertragungsbeiwert  $K_S$  die Verstärkung der betreffenden Regelstrecke an. Aus dem Verhältnis von Verzugszeit zu Ausgleichszeit  $T_u/T_g$  lässt sich die Regelbarkeit der betreffenden Strecke abschätzen. Je kleiner dieser Wert d. h. je geringer die Verzugszeit im Verhältnis zur Ausgleichszeit ist, um so besser lässt sich der Prozess regeln.

Nach den Werten für  $T_u$  und  $T_g$  lässt sich das Zeitverhalten von Regelstrecken grob einteilen:

$T_u < 0,2 \text{ min}$  und  $T_g < 2 \text{ min}$  → schneller Prozess

$T_u > 0,5 \text{ min}$  und  $T_g > 5 \text{ min}$  → träger Prozess

Der absolute Wert der Ausgleichszeit hat deshalb direkten Einfluss auf die Abtastzeit des Reglers: je größer  $T_g$  – d. h. je träger die Prozessreaktion – um so größer kann die Abtastzeit gewählt werden.

### Strecke mit Ausgleich und Totzeit

Viele Prozesse mit Transportvorgängen (Rohrleitungen, Transportbänder usw.) haben ein Zeitverhalten, das dem Verlauf in Bild 2-2 ähnlich ist. Es verstreicht eine Anlaufzeit  $T_a$ , die sich aus der eigentlichen Totzeit und der Verzugszeit der Strecke mit Ausgleich zusammensetzt. Für die Regelbarkeit ist es außerordentlich wichtig, dass  $T_t$  klein gegenüber  $T_g$  ist bzw. das Verhältnis  $T_t/T_g \leq 1$  bleibt.

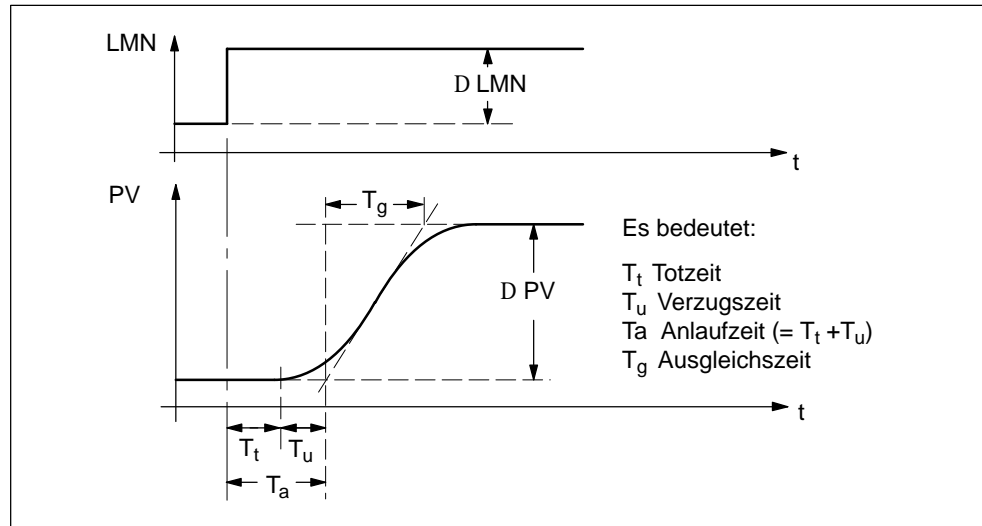


Bild 2-2 Sprungantwort einer Strecke mit Totzeit und Ausgleich ( $T_t$ -PT-Strecke)

Da der Regler während der Totzeit vom Geber keine Signaländerung empfängt, erfolgen seine Eingriffe zwangsläufig verspätet und mit entsprechend verringerter Regelqualität. Bei Einsatz eines Standard-Reglers können derartige Effekte nur durch günstigere Wahl der Einbaulage des Messfühlers beeinflusst werden.

## Strecke mit Integralwirkung

Hier ist die Steilheit der Rampe der Prozessvariablen (PV) nach Änderung des Stellwertes um einen festen Betrag umgekehrt proportional zum Wert der Integrationskonstante  $T_I$ .

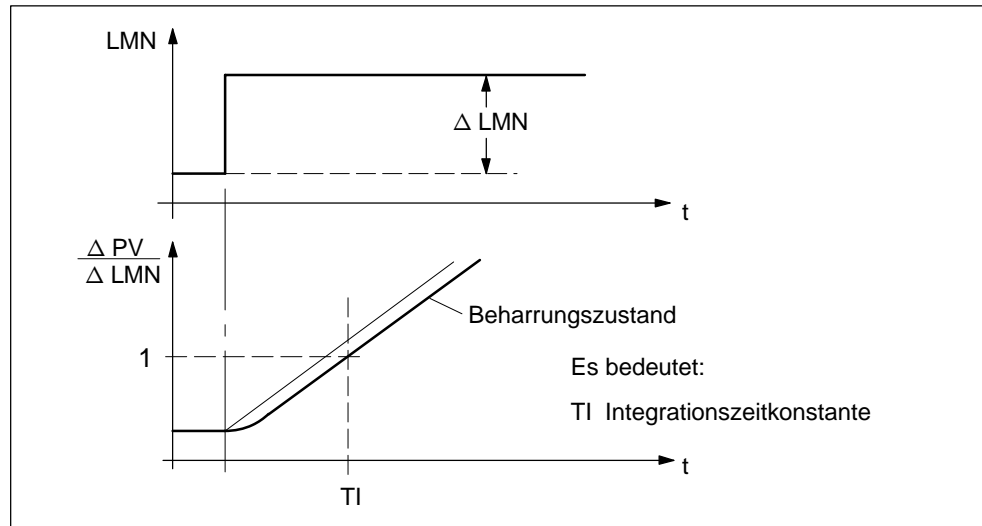


Bild 2-3 Übergangsfunktion einer Regelstrecke ohne Ausgleich (I-Strecke)

Prozesse mit I-Verhalten sind z. B. Niveaustrecken, bei denen der Füllstand je nach Öffnungsgrad des Stellgliedes unterschiedlich schnell angehoben oder abgesenkt wird. Wichtige Regelstrecken mit I-Verhalten stellen außerdem die häufig verwendeten motorischen Antriebe dar, bei denen die Änderungsgeschwindigkeit von Verstellwegen direkt proportional zur Drehzahl des Antriebs ist.

Treten bei einer Strecke mit Integralwirkung vor dem I-Glied keine Störgrößen auf (das ist meistens der Fall), dann sollte ein Regler ohne I-Anteil verwendet werden. Eine Störgrößeneinwirkung am Streckeneingang kann – ohne Verwendung eines I-Anteils im Regler – meist durch Störgrößenaufschaltung eliminiert werden.

## 2.2 Identifikation der Streckeneigenschaften

### Die Streckenidentifikation

Wie bereits erwähnt, sind zur Untersuchung und Identifizierung eines gegebenen Prozessverhaltens zwei Schritte erforderlich:

1. die Aufnahme der Übergangsfunktion der Regelstrecke nach Anregung durch eine sprungförmige Änderung der Stellgröße,
2. die Auswertung der aufgenommenen oder gespeicherten Übergangsfunktion zur Ermittlung einer geeigneten Reglerstruktur und der optimalen Reglerparameter.

### 1. Aufnahme der Übergangsfunktion

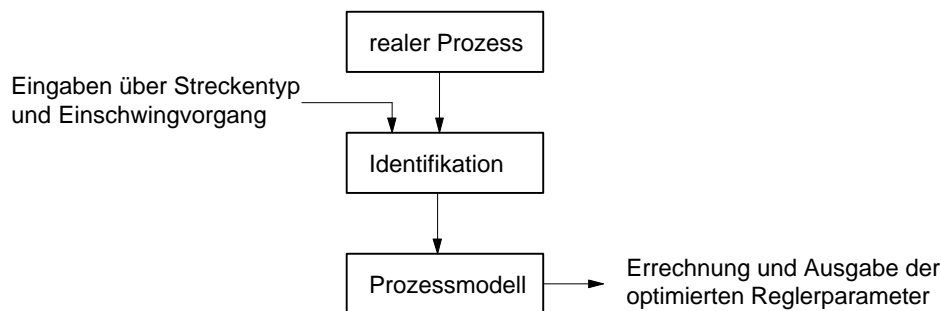
Bei der Ausführung von Schritt 1. werden Sie durch die im Konfigurationswerkzeug verfügbare Teilfunktion zur Prozess-Identifikation weitgehend unterstützt.

In den Dialogfenstern wird Ihnen, soweit erforderlich, in Kommentartexten das Hintergrundwissen zu Ihren Aktionen vermittelt. Eingabefenster oder Ausgabefenster werden in bestimmten Prozedurschritten automatisch geöffnet.

### 2. Ermittlung der Reglerkenndaten

Für die eigentliche Prozessidentifikation (Schritt 2.) beschränken sich Ihre Aktivitäten auf die Angabe des Einschwingverhaltens der Regelgröße nach Störungen im Regelkreis ("aperiodisch" oder "mit 10 % Überschwingung") und das Starten der automatischen Prozessidentifikation durch das System.





Folgendes Bild veranschaulicht die Methodik, mit der das System die Identifizierung eines Prozesses ausführt:



Die Ergebnisse dieses Identifikationsvorgangs werden Ihnen in einem Ausgabefenster angezeigt. Sie können hier die Übertragung der ermittelten PI- oder PID-Parameter in die Datenbasis veranlassen oder aber die Ergebnisse verwerfen und die Identifikationsprozedur mit anderen Streckendaten oder anderen Einstellungen wiederholen.

### Streckenidentifikation und Betriebsart des Regelkreises

Bei folgenden Betriebszuständen der Regelung kann für unterschiedliche Typen der Regelstrecke in der angegebenen Weise eine Prozessidentifikation ausgeführt werden:

	Daten- erfassung	Regelkreis	Strecke	Prozessanregung
1.	online	offen (Handbetrieb)	ohne I-Anteil	sprungförmige Stellgrößenänderung: 
2.	online	geschlossen (Automatikbetrieb)	ohne I-Anteil	sprungförmige Sollwertänderung: 
3.	online	offen (Handbetrieb)	mit I-Anteil	impulsförmige Stellgrößenänderung: 
4.	online	geschlossen (Automatikbetrieb)	mit I-Anteil	impulsförmige Sollwertänderung: 
5.	offline	Regelkreisdaten aus Archiv		

## 2.3 Störgrößenaufschaltung

### Störgrößenaufschaltung

Auf den Prozess einwirkende Störgrößen sind durch den Regler auszuregeln. Konstante Störgrößen werden durch Regler mit I-Anteil ausgeregelt. Das Regelverhalten wird dabei nicht beeinflusst.

Dynamische Störgrößen beeinflussen dagegen die Qualität einer Regelung weit aus stärker. Abhängig vom Angriffspunkt der Störung und den Zeitkonstanten nachfolgender Streckenanteile reagiert die Regelung mit Regeldifferenzen unterschiedlicher Größe und Dauer, die erst über die I-Wirkung im Regler wieder abgebaut werden.

Dieser Effekt lässt sich in den Fällen vermeiden, in denen die Störgröße durch Messung erfasst werden kann. Durch Aufschaltung der gemessenen Störgröße auf den Ausgang des Reglers lässt sich ein Kompensationseingriff ableiten, durch den die Regelung erheblich schneller auf diese Störgröße reagieren kann.

Der Standard-Regler besitzt einen Signaleingang für die Störgröße DISV. Diese Störgröße wird über einen Strukturschalter auf die Summationsstelle am Ausgang des PID-Reglers aufgeschaltet (Bild 2-4).

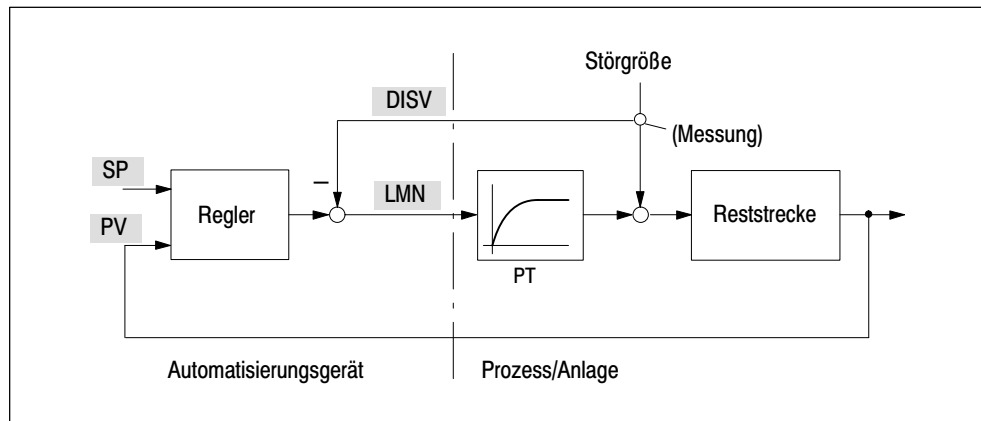


Bild 2-4 Kompensation einer Störgröße, die am Streckeneingang angreift  
(Signalbezeichnungen von Standard PID Control)

## 2.4 Mehrschleifige Regelungen

### Prozesse mit abhängigen Regelgrößen

Das Produkt "Standard PID Control" enthält vorbereitete Beispiele (Example3 ... Example5, siehe *Kapitel 7*), mit denen Sie auf einfache Weise mehrschleifige Regelungen realisieren können. Die Anwendung solcher Regelungsstrukturen bringt immer dann Vorteile, wenn Prozesse mit prozessgebundenen oder zahlenmäßig voneinander abhängenden Regelgrößen vorliegen.

Wie diese Reglerstrukturen aufgebaut sind und wie sie eingesetzt werden, sollen Ihnen nachstehende Erläuterungen zeigen.

### Mehrschleifige Verhältnisregelung (Example3)

Immer dann, wenn in Prozessen das Verhältnis zweier oder mehrerer Regelgrößen wichtiger ist als das Konstanthalten ihrer Absolutwerte, wird der Einsatz einer **Verhältnisregelung** erforderlich (Bild 2-5).

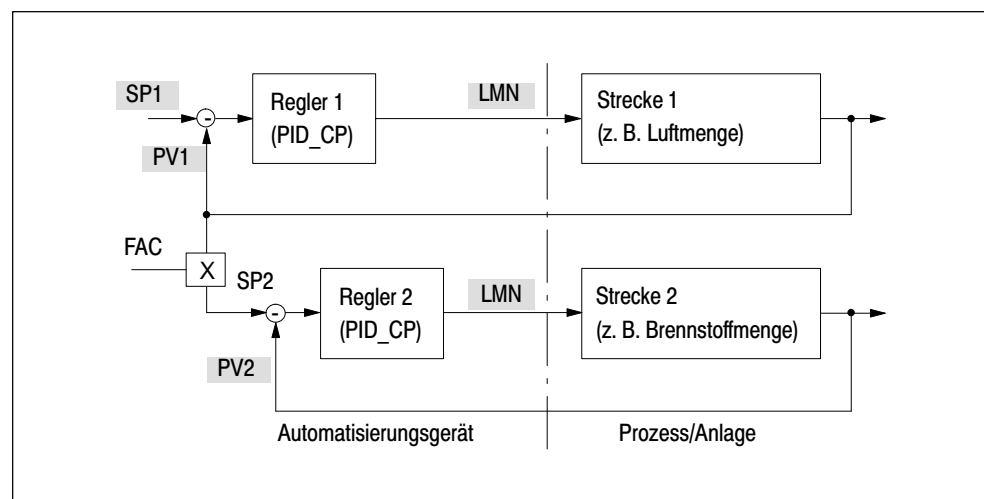


Bild 2-5 Verhältnisregelung mit zwei Regelkreisen (Example3)

Im Allgemeinen handelt es sich bei den in einem bestimmten Verhältnis zu regelnden Prozessgrößen um Durchflüsse bzw. Mengen, wie sie z. B. bei Verbrennungsprozessen vorkommen. In Bild 2-5 wird die im Regelkreis 2 geregelte Brennstoffmenge in einem über **FAC** einstellbaren Verhältnis der an **SP1** eingestellten Luftmenge nachgeführt.



## Mischungsregelung (Example4)

Bei einem Mischungsprozess muss einerseits die Gesamtmenge der zu mischenden Stoffe geregelt werden, andererseits müssen die Komponenten in bestimmten konstant zu haltenden Verhältnissen bzw. Prozentsätzen in die Gesamtmenge eingehen.

Aus diesen Forderungen ergibt sich (aus der Verhältnisregelung abgeleitet) eine Regelungsstruktur, in der für jede Mischungskomponente eine Mengenregelung vorgesehen ist. Deren Sollwerte werden hier nach Maßgabe der fest eingestellten Anteils- bzw. Verhältnisfaktoren (FAC) durch die Stellgröße des Gesamtmengenreglers beeinflusst (Bild 2-6).

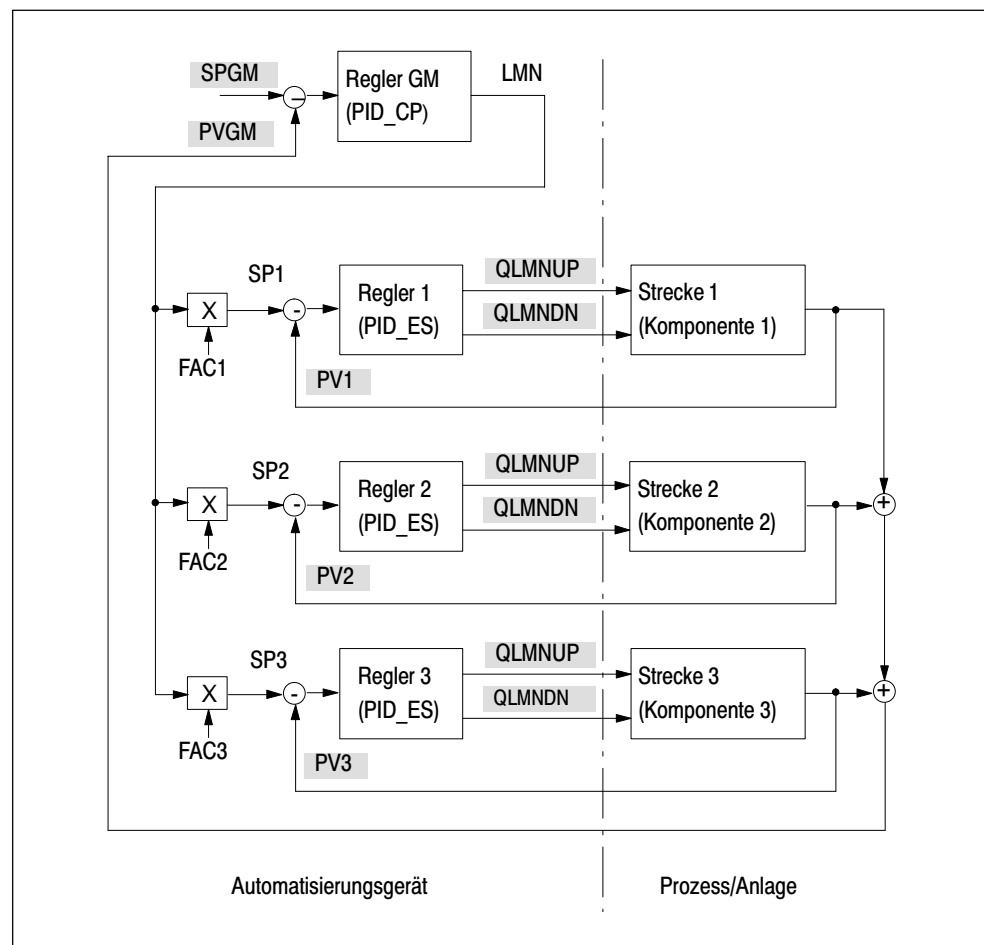


Bild 2-6 Mischungsregelung für drei Komponenten (Example\_4)

Die Reglerstruktur für die Mischungsregelung (Example4) enthält einen Regler mit kontinuierlichem Ausgang (PID\_CP) für die Regelung der Gesamtmenge GM sowie drei Schrittreger (PID\_ES) für die unterlagerte Regelung der Einzelkomponenten 1...3, die sich gemäß den eingestellten Faktoren FAC1... FAC3 zur Gesamtmenge zusammensetzen (Additionsstellen!).

## Kaskadenregelung (Example5)

Wenn sich in einem Prozess neben der eigentlichen zu regelnden Prozessgröße eine Zwischengröße (Hilfsregelgröße) ableiten lässt, die einer eigenen Regelung unterworfen werden kann, dann erhält man bezüglich der Hauptregelgröße meist bessere Regelergebnisse als mit einer einschleifigen Regelung.

Die Zwischengröße PV2 wird in einem unterlagerten Folgeregelkreis ausgegeregelt (Bild 2-7). Damit werden Störungen aus diesem Anlagenbereich vom Einfluss auf die qualitätsbestimmende Hauptregelgröße PV1 ferngehalten. Strukturbedingt werden innere Störgrößen schneller ausgegeregelt, da sie nicht die gesamte Regelstrecke durchlaufen müssen. Der Führungsregler kann in diesem Fall empfindlicher eingestellt sein, mit dem Vorteil einer schnelleren und genaueren Ausregelung auf den festen Sollwert SP.

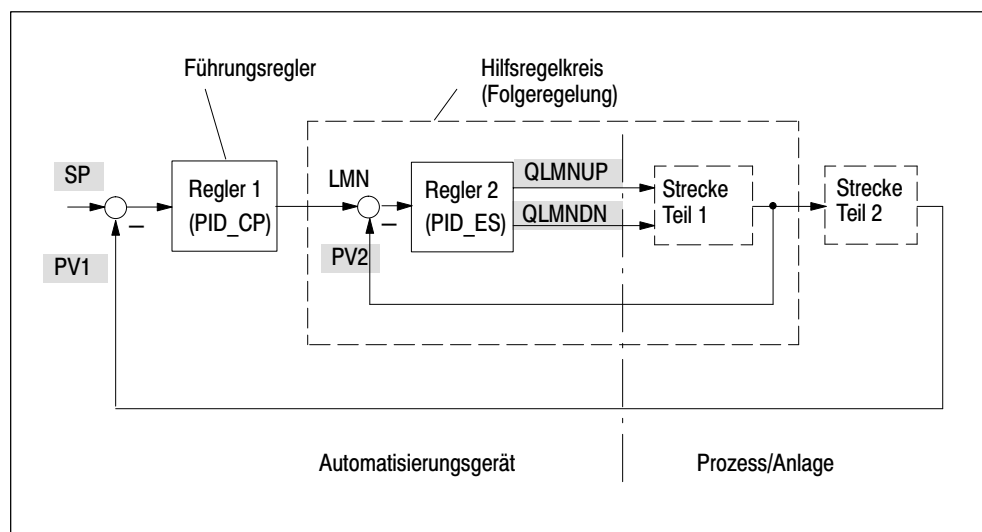


Bild 2-7 Zweischleifige Kaskadenregelung (Example5)

Die Reglerstruktur für die Kaskadenregelung (Example5) enthält einen Regler mit kontinuierlichem Ausgang (PID\_CP) für die Regelung der Führungsgröße des Hilfsregelkreises sowie einen Schrittreger (PID\_ES) für die unterlagerte Regelung der Zwischengröße PV2 (Folgeregler).

## 2.5 Struktur und Arbeitsweise von Standard PID Control

### Die Abtastregelung

Die mit "Standard PID Control" realisierbaren Regler sind grundsätzlich digitale Abtastregler (DDC-Regler, = Direct Digital Control). Abtastregler arbeiten zeitgesteuert, d. h. sie werden in immer gleichen Zeitabständen (der Abtastzeit CYCLE) bearbeitet. Die Abtastzeit bzw. die Häufigkeit, mit der ein bestimmter Regler bearbeitet wird, kann eingestellt werden.

Im Bild 2-8 ist der einfache Regelkreis mit dem Standard-Regler dargestellt. Anhand des Bildes sollen die Bezeichnungen der wichtigsten Kreisgrößen sowie deren in diesem Handbuch verwendeten Parameterbezeichnungen (Abkürzungen) noch einmal in Erinnerung gerufen werden.

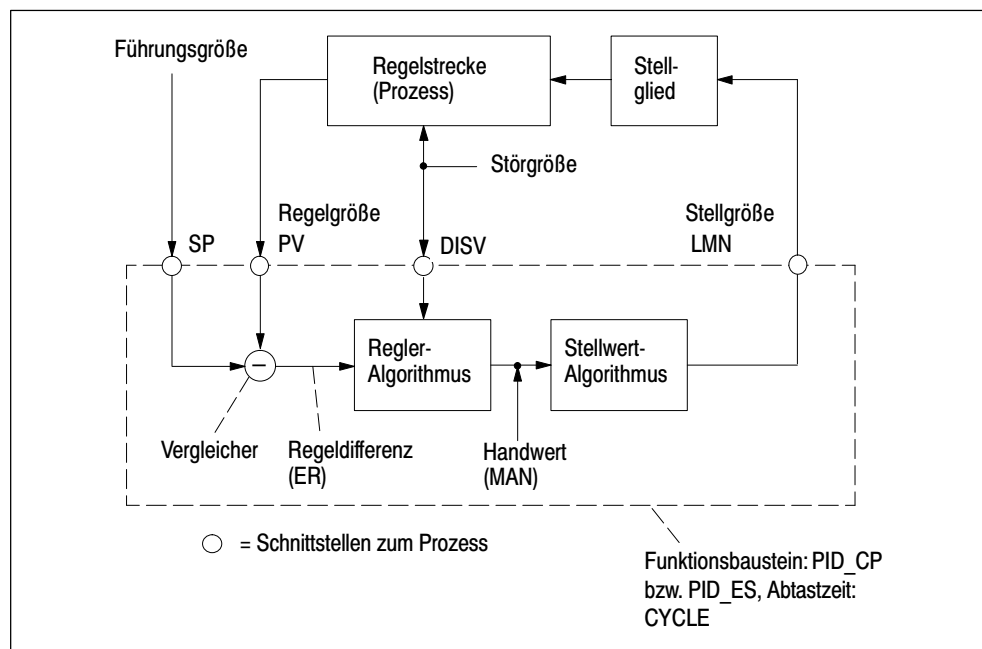


Bild 2-8 Abtastregler von Standard PID Control im Regelkreis

Bei den Regelfunktionen, die in den Funktionsbausteinen PID\_CP und PID\_ES realisiert werden, handelt es sich um reine Software-Regelungen. Deren Ein- und Ausgangswerte werden mit Hilfe von digitalen Algorithmen auf einer CPU bearbeitet.

Da die Bearbeitung der Reglerbausteine im Prozessor der CPU seriell erfolgt, können Eingangswerte nur zu diskreten Zeitpunkten erfasst und ebenso Ausgangswerte nur zu definierten Zeitpunkten ausgegeben werden. Dieses Verhalten ist das Charakteristikum der **Abtastregelung**.

## Regel-Algorithmus und konventionelle Regelung

Der Regel-Algorithmus im Prozessor bildet den Regler unter Echtzeitbedingungen nach. Zwischen den Abtastzeitpunkten reagiert die Regelung auf Änderungen der Prozessgröße PV nicht und die Stellgröße LMN bleibt unverändert.

Vorausgesetzt jedoch, die Abtastung geschieht in so kurzen Abständen, dass die Folge der Abtastwerte den kontinuierlichen Messgrößenverlauf annähernd getreu wiedergibt, lässt sich die digitale Regelung als eine quasikontinuierliche betrachten. Es können auch im Falle von "Standard PID Control" die üblichen Methoden zur Ermittlung von Struktur und Kennwerteinstellung wie bei kontinuierlichen Reglern angewendet werden.

Diese Voraussetzung für die Erstellung und Dimensionierung von Regelungen mit dem Regelungspaket "Standard PID Control" ist dann erfüllt, wenn die Abtastzeit (CYCLE) kürzer ist als 20 Prozent der Gesamtstreckenzeitkonstante.

Wird diese Bedingung eingehalten, lassen sich die Funktionen von "Standard PID Control" wie die von konventionellen Reglern beschreiben. Der gleiche Funktionsvorrat steht mit den gleichen Möglichkeiten für die Überwachung von Regelkreisgrößen und für das Tuning der Reglers zur Verfügung.

## Die Funktionalität von "Standard PID Control"

Die folgenden Bilder zeigen die vorkonfektionierten Reglerstrukturen von Standard PID Control als Blockschema. Im Bild 2-9 ist der kontinuierliche Regler mit den Signalverarbeitungszweigen für Ist- und Sollwert, dem Regler und dem Stellwertzweig dargestellt. Sie erkennen daran, welche Funktionen – nach der Signalumformung am Eingang – zu realisieren sind und welche nicht.

Das Funktionsinventar von "Standard PID Control" ist zwar starr, kann jedoch in allen Signalverarbeitungszweigen um jeweils eine eigene Funktion (FC) erweitert werden.

In den Bildern 2-10 und 2-11 wird die Stellwertbildung beim Schrittreger in den Varianten mit und ohne Stellungsrückmeldung dargestellt. Daraus wird ersichtlich, wie bei fehlender Stellungsrückmeldung ein quasi-stellungsproportionales Rückführsignal aus den Einschaltdauern der Binärausgänge erzeugt wird.

- Detaillierte Beschreibungen der jeweiligen Funktionen finden Sie in den *Kapiteln 4 bis 7* des Handbuches. Hintergrund- und Orientierungsinformationen sind außerdem in der Online-Hilfe verfügbar.
- Die Strukturbilder im folgenden Abschnitt enthalten Einzelheiten mit Parameterbezeichnungen und Struktur- bzw. Betriebsartenschaltern (*siehe Kapitel 2.6*).
- Eine genaue Darstellung des gesamten Signalflusses im kontinuierlichen Regler und im Schrittreger finden Sie in den Blockschaltbildern im *Kapitel 8.2*.

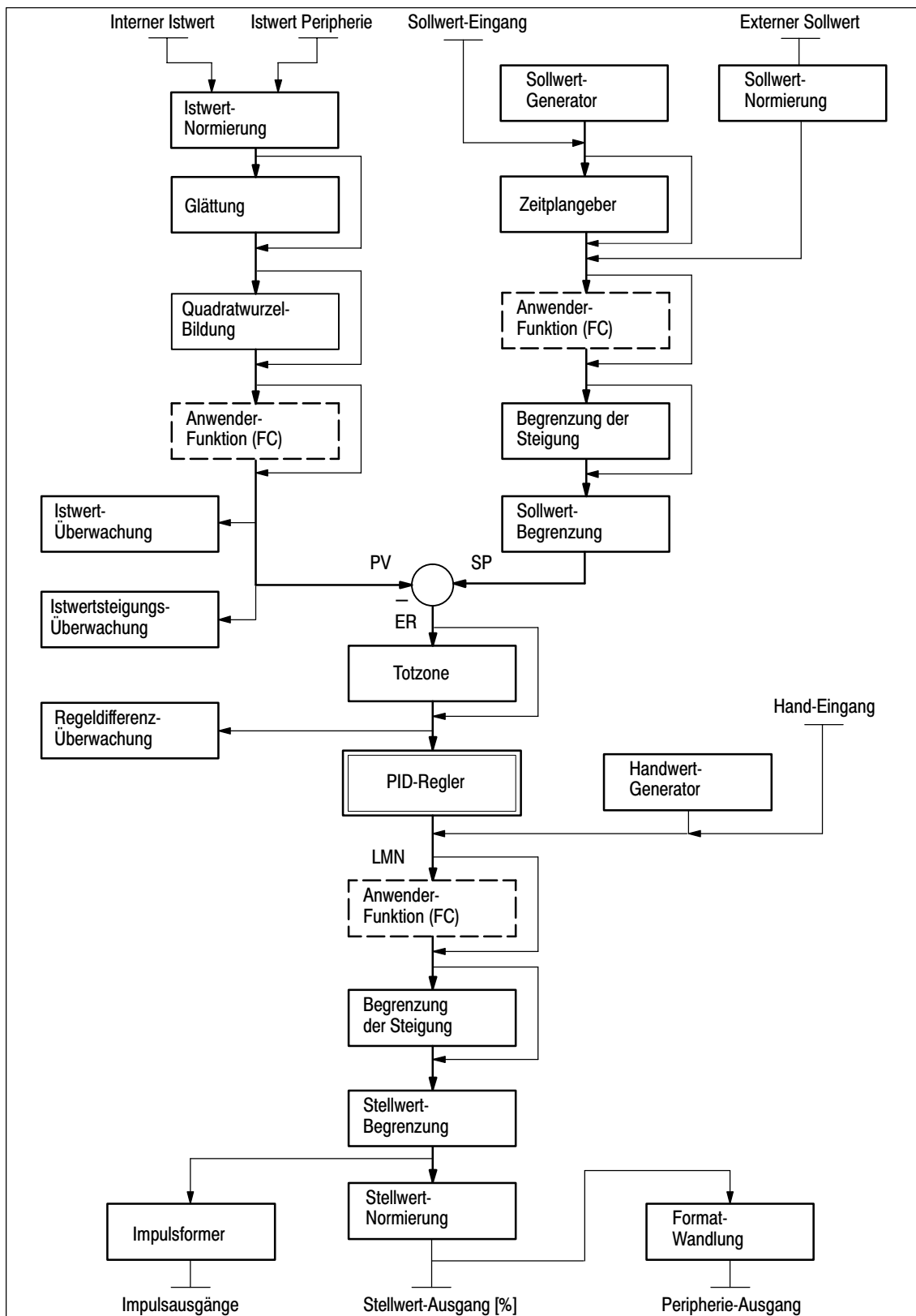


Bild 2-9 Funktionskette von Standard PID Control (kontinuierlicher Regler)

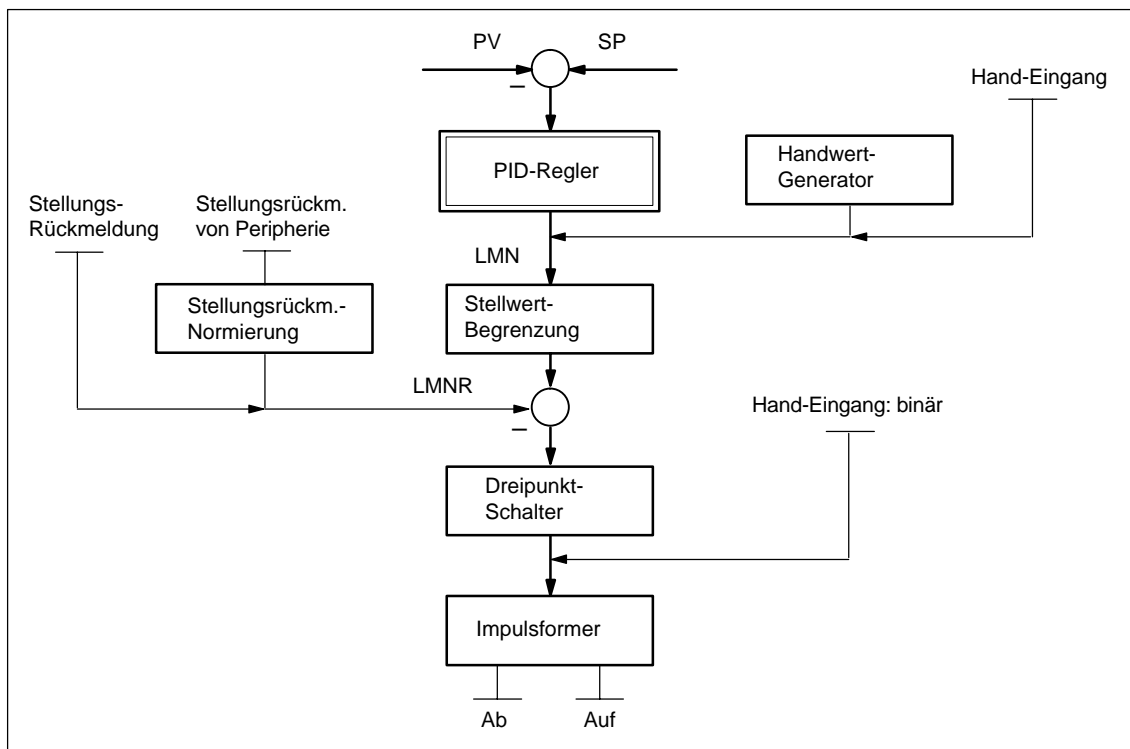


Bild 2-10 Stellwertzweig des Schrittreglers mit Stellungsrückmeldung

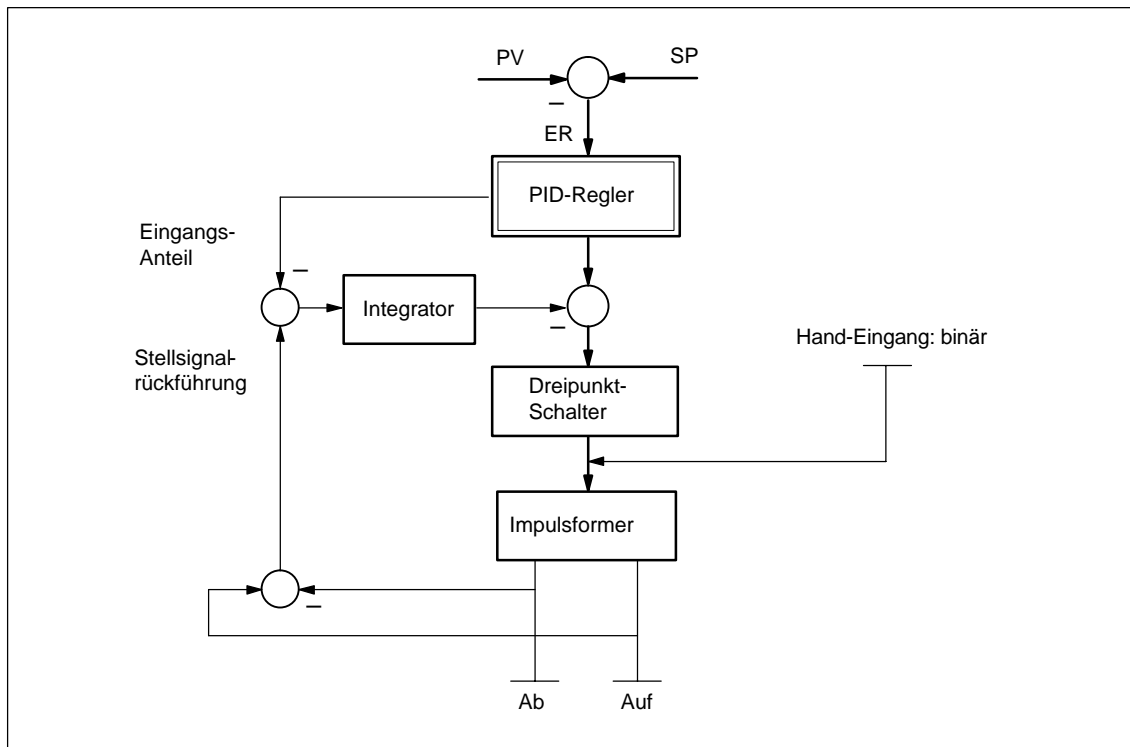


Bild 2-11 Stellwertzweig des Schrittreglers ohne Stellungsrückmeldung

## 2.6 Signalflusspläne

### Signalflussdarstellungen

Die nachstehenden Grafiken zeigen schematisierte Übersichten über die Funktionen von Standard PID Control. Deutlich ist die Vielzahl der (SW-)Schalter, mit deren Hilfe Sie die von Ihnen benötigte Funktionsauswahl in der Struktur bestimmen.

Analog zur Darstellung der Schalter im Konfigurationswerkzeug bedeutet der schwarze Punkt in den Schaltersymbolen, dass das Schaltsignal den nebenstehenden booleschen Wert (0 = FALSE oder 1 = TRUE) hat und dass der Signalweg jeweils über diesen Punkt durchgeschaltet ist. Die Schaltsignale (Binärsignale) sind durch gestrichelte Linien gekennzeichnet.

In den Plänen sind jeweils Teilfunktionen mit den durch die Voreinstellung der Schalterbits im Auslieferungszustand bestimmten Signalwegen dargestellt. Generell haben alle Schaltsignale in der Ausgangsparametrierung den Wert FALSE (Ausnahme P\_SEL sowie I\_SEL und MAN\_ON = TRUE).

Das bedeutet, es wird der Sollwert absolut über SP\_INT vorgegeben, das gleiche gilt für die Eingabe des Istwertes über PV\_IN. Als Regelfunktion ist ein normaler PI-Regler mit P-Funktion im Vorwärtszweig eingestellt. Der Regelkreis ist geöffnet, und die Stellgröße wird im Prozent-Bereich über den Eingang MAN beeinflusst. Alle übrigen Funktionen sind passiv oder – falls nicht abschaltbar – durch Vorbesetzung von "Rand"-Parametern bei Signalverlauf innerhalb des Mess- bzw. Arbeitsbereiches nicht wirksam.

### Symbole und Kennungen in den Signalflussplänen

Die Bezeichnungen der anschließbaren Prozessgrößen sind unterlegt dargestellt. Sie erkennen daran, wo die Reglerstruktur mit der S7-Peripherie oder direkt mit den Mess- und Stellgliedern des Prozesses verbunden werden kann.

Parameterbezeichnungen mit der Buchstabengruppe "OP" (z. B. SP\_OP/ SP\_OP\_ON) verweisen auf den an dieser Stelle möglichen Eingriff über das Konfigurationswerkzeug von Standard PID Control. Das Konfigurationswerkzeug hat eine eigene Schnittstelle zum Regler-FB.

Zwischengrößen im Signalverlauf können durch die mit Kreisen markierten Messpunkte MP1 ... MP12 beobachtet werden. Diese Zwischengrößen werden zum Abgleich von Werten vor Auslösung "stoßfreier" Umschaltvorgänge oder zur Einsichtnahme in aktuelle Zustände der betreffenden Regelung benötigt. Die Messpunktgrößen im Kurvenschreiber des Konfigurationswerkzeugs können (statisch und dynamisch) dargestellt werden.

Der Übersichtlichkeit wegen sind die Parameter zur Einstellung und Dimensionierung der Bearbeitungsvorgänge (Algorithmen) bei den einzelnen Funktionsblöcken jeweils mit angegeben. Wir verweisen dazu auf die Beschreibungen im Referenzteil und auf die Darstellung einzelner Teilfunktionen in den folgenden Abschnitten.

### Signalverarbeitung im Zweig der Führungsgröße

- **Festeinstellung des Sollwerts (SP\_GEN)**

Bei Festwertregelungen wird der Sollwert am Sollwertgenerator SP\_GEN per Schalterbetrieb gewählt und fest eingestellt.

- **Sollwerteinstellung per Zeitplan (RMP\_SOAK)**

Bei der Regelung von Prozessen auf unterschiedliche Sollwerte nach einem zeitgesteuertem Programm erzeugt der Zeitplangeber RMP\_SOAK nach entsprechender Parametrierung die gewünschte Fahrkurve der Führungsgröße und beeinflusst damit den Prozess im Sinne eines definierten Änderungsprofils der Regelgröße.

- **Änderungsbegrenzung der Führungsgröße (SP\_ROC)**

Die Umwandlung von Sollwertsprüngen in einen rampenförmigen Anstieg oder Abfall der Führungsgröße verhindert Eingangssprünge auf den Prozess. Die Funktion SP\_ROC begrenzt die Sollwertsteigung getrennt für An- und Abstieg sowie für positive und negative Werte der Führungsgröße.

- **Absolutwertbegrenzung der Führungsgröße (SP\_LIMIT)**

Um unzulässige Prozesszustände zu verhindern, wird der Einstellbereich der Führungsgröße durch SP\_LIMIT nach oben und unten begrenzt.

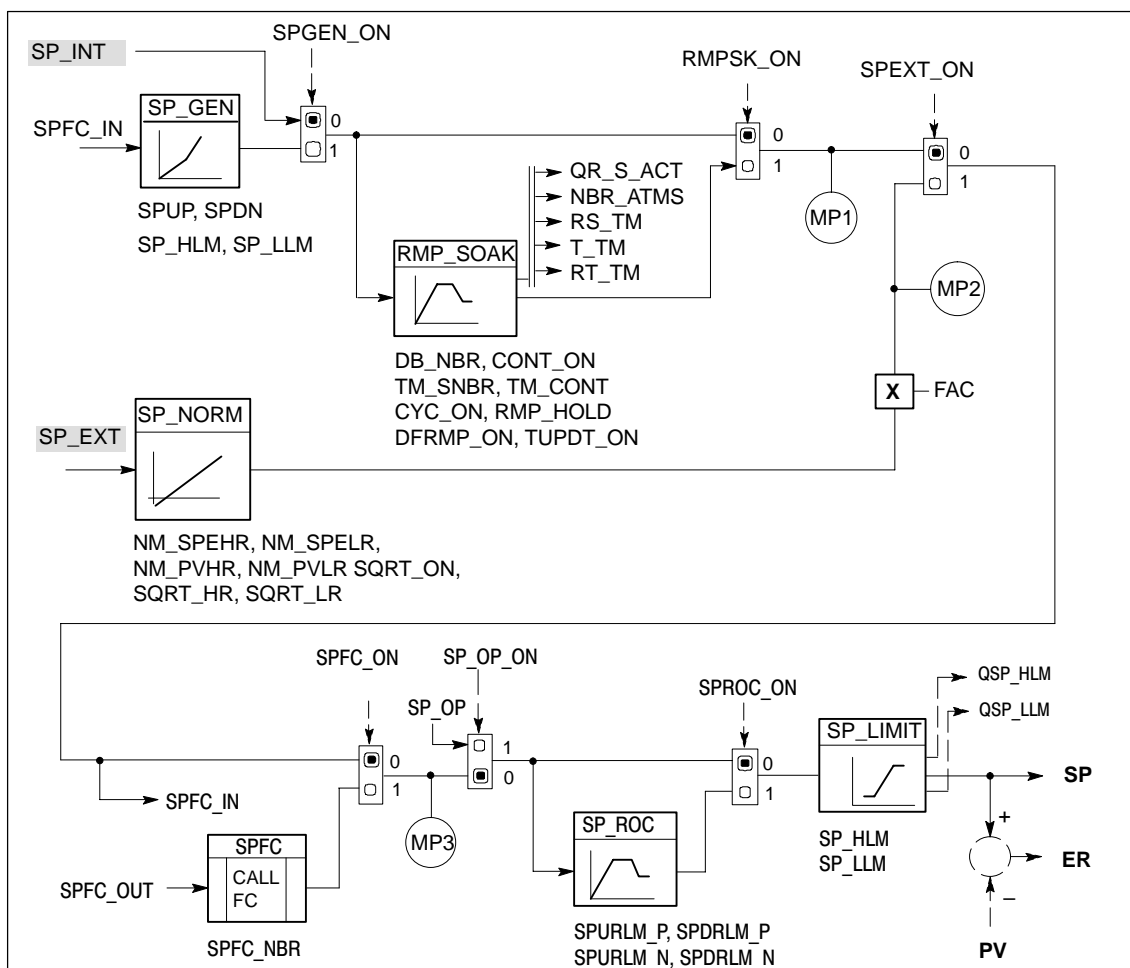


Bild 2-12 Signalflussplan der Sollwertverarbeitung



### Signalverarbeitung im Zweig der Regelgröße

- **Verzögerung der Regelgröße (LAG1ST)**

Zur Glättung von Prozesssignalen mit überlagerten schnellen Schwingungen (Noise) wird ein Verzögerungsglied erster Ordnung im Istwertzweig eingesetzt. Diese Funktion dämpft die analoge Regelgröße je nach eingestellter Zeitkonstante PV\_TMLAG mehr oder weniger stark. Störsignale werden so wirksam unterdrückt. Insgesamt wird dadurch jedoch auch die Zeitkonstante des Gesamtregelkreises erhöht, d. h. die Regelung wird träger.

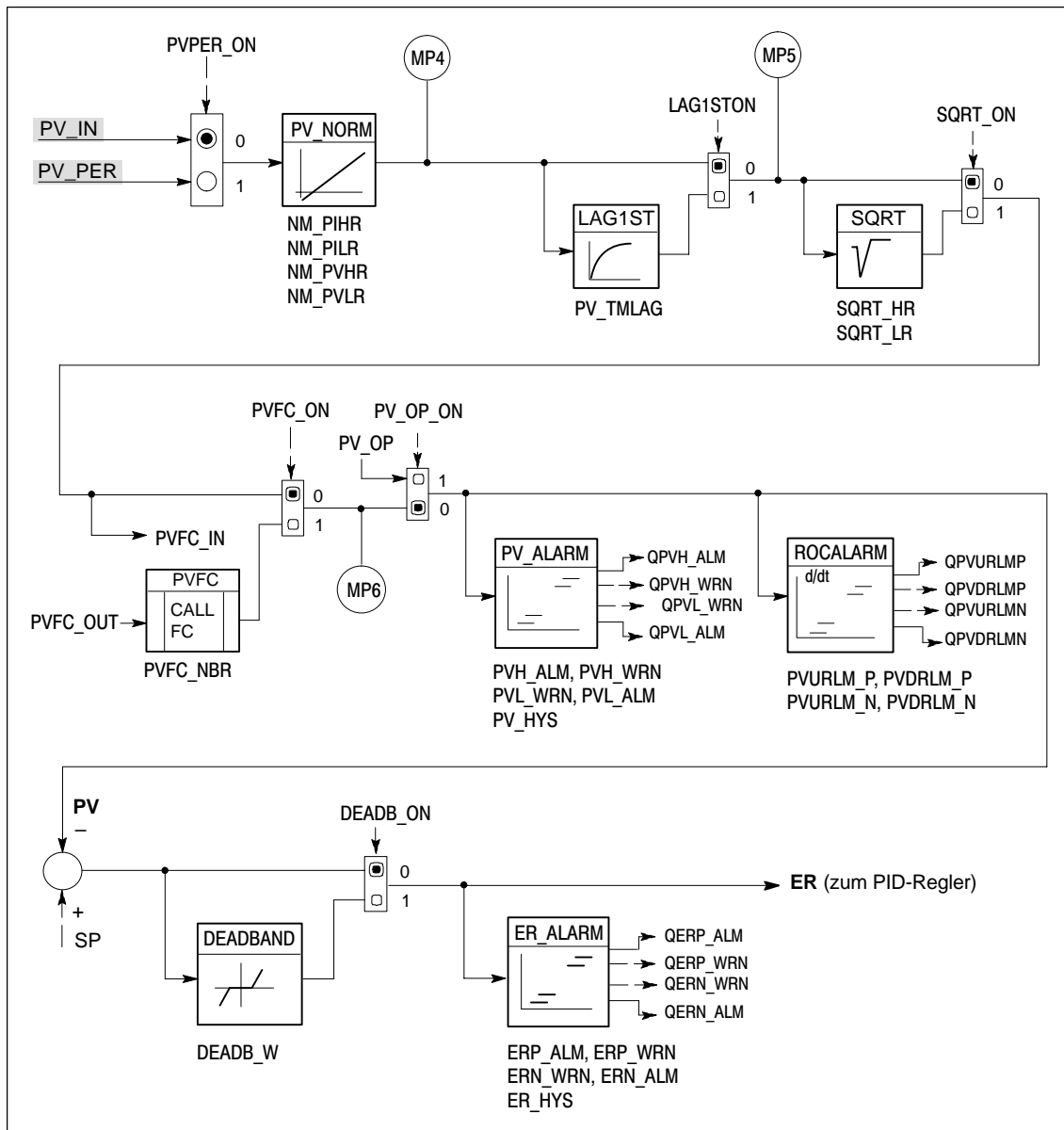


Bild 2-13 Signalflussplan der Istwert- und Regeldifferenzverarbeitung

- **Radizierung der Regelgröße (SQRT)**

Bei einer quadratischen Abhängigkeit des Messsignals vom physikalischen Wert (Durchflussmessung über Wirkdruckerfassung) ist es erforderlich, die Regelgröße durch Radizierung (Quadratwurzel-Algorithmus) zu linearisieren. Nur eine lineare Größe lässt sich mit dem linearen Sollwert für den Durchfluss vergleichen und im Regelalgorithmus verarbeiten. Im Istwertzweig ist dafür das wahlweise zuschaltbare Funktionsglied SQRT vorgesehen.

- **Überwachung der Regelgrößen-Steigung (ROCALARM)**

Wenn die Änderungsgeschwindigkeit der Regelgröße sehr groß bzw. zu groß wird, so deutet das auf gefährliche Prozesszustände hin, auf die das Automatisierungssystem gegebenenfalls reagieren muss. Zu diesem Zweck erzeugt die Funktion ROCALARM bei Überschreitung vorgegebener Steigungswerte durch PV (positiv oder negativ) entsprechende Alarmsignale, die entsprechend weiterverarbeitet werden können.

- **Überwachung der Absolutwerte von Regelgröße und Regeldifferenz**

Der Verlauf der Regelgröße und der Regeldifferenz wird durch die jeweilige Überwachungsfunktion PV-ALARM bzw. ER\_ALARM auf die Über-/Unterschreitung von jeweils zwei Grenzwerten überwacht.

- **Überlagerung durch Signalrauschen (DEADBAND)**

Um störendes Signalrauschen auf den Kanälen der Regelgröße oder der externen Führungsgröße auszufiltern, wird die Regeldifferenz über ein zuschaltbares Totzonenglied geführt. Je nach Stör-Amplitude kann hier eine unterschiedlich breite Unempfindlichkeitszone für die Signalübertragung eingestellt werden. Dabei ist eine von der Breite der eingestellten Unempfindlichkeitszone abhängige Verfälschung des übertragenen Signals in Kauf zu nehmen.

## Signalverarbeitung im PID-Regler

- **Normale PID-Reglerfunktion**

Die per aktuellem Schaltzustand in Bild 2-14 wirksame Struktur realisiert einen PI-Regler mit paralleler Verarbeitung der Signale des P- und des I-Anteils. D-SEL = TRUE ergänzt den Regelalgorithmus um die Parallelverarbeitung im D-Zweig. Über  $\pm$  GAIN wird der Proportionalbeiwert bzw. die Verstärkung des Reglers bestimmt, negatives Vorzeichen bedeutet Fallen der Stellgröße bei steigender Regelgröße.

- **PD in Rückführung**

Wird die Bildung des P- und D-Anteils in die Rückführung (PFDB\_SEL und DFDB\_SEL = TRUE) verlegt, dann verursachen sprungförmige Änderungen des Sollwertes keine Sprünge im Verlauf der Stellgröße. Der Faktor "−1" bewirkt eine negative Aufschaltung des Rückföhreinflusses.

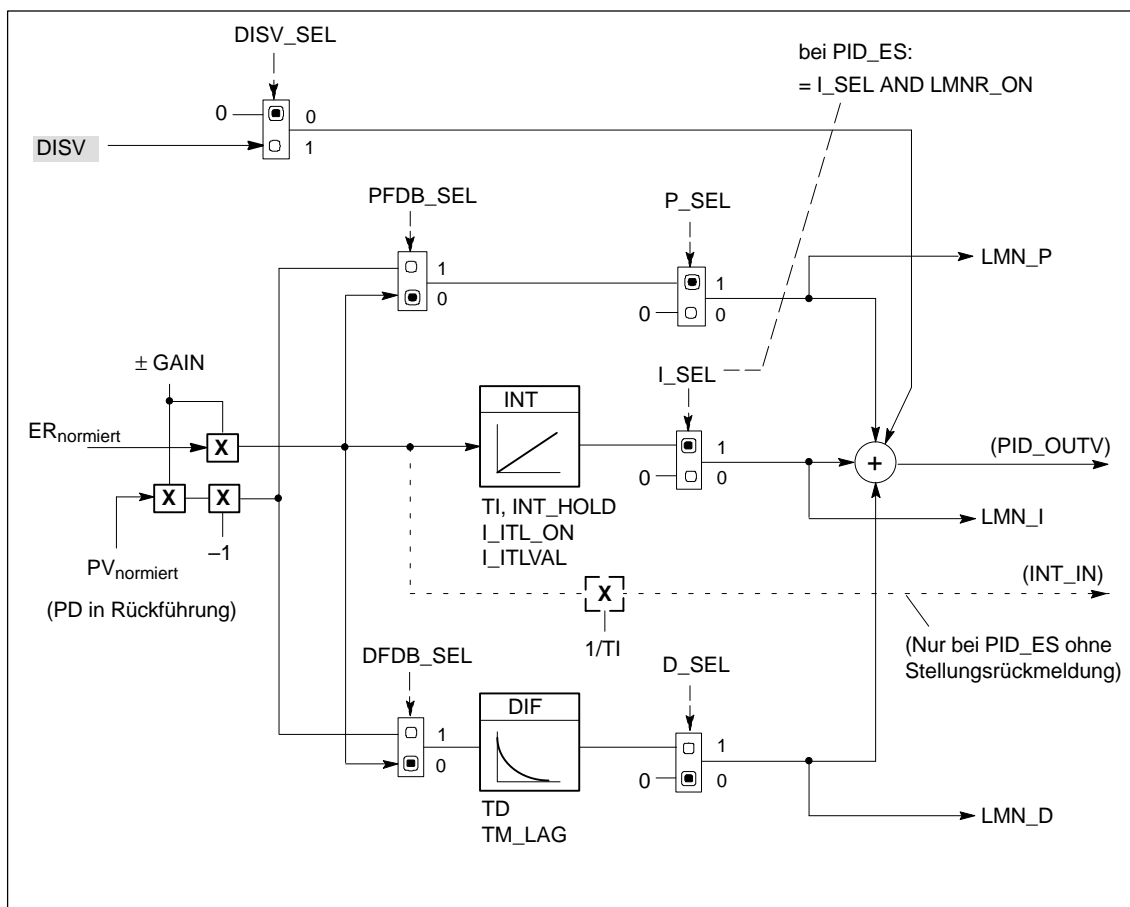


Bild 2-14 Signalflussplan der Regelfunktionen

## Signalverarbeitung im Zweig der analogen Stellgröße

- **Festeinstellung des Handwerts (MAN\_GEN)**

Bei Handbetrieb der Stellgröße (Regelkreis offen) wird der Stellwert am Handwertgenerator MAN\_GEN per Schalterbetrieb gewählt und fest eingestellt.

- **Änderungsbegrenzung der Stellgröße (LMN\_ROC)**

Die Umwandlung von sehr schnellen sprungförmigen Stellwertänderungen in einen rampenförmigen Anstieg oder Abfall der Stellgröße verhindert Eingangssprünge auf den Prozess. Die Funktion (LMN\_ROC) begrenzt die Stellwertsteigung getrennt für An- und Abstieg der Stellgröße.

- **Absolutwertbegrenzung der Stellgröße (LMNLIMIT)**

Um unzulässige Prozesszustände zu vermeiden, bzw. um eingeschränkte Verfahrbedingungen des Stellgliedes einzuhalten, wird der Einstellbereich der Stellgröße durch LMNLIMIT nach oben und unten begrenzt.

- **Aktivieren der Kaskadenregelung**

Das OR-Gatter erzeugt abhängig von der Kombination der Schaltzustände von Standard PID Control ein Freigabesignal für die Kaskadenkoppelung.

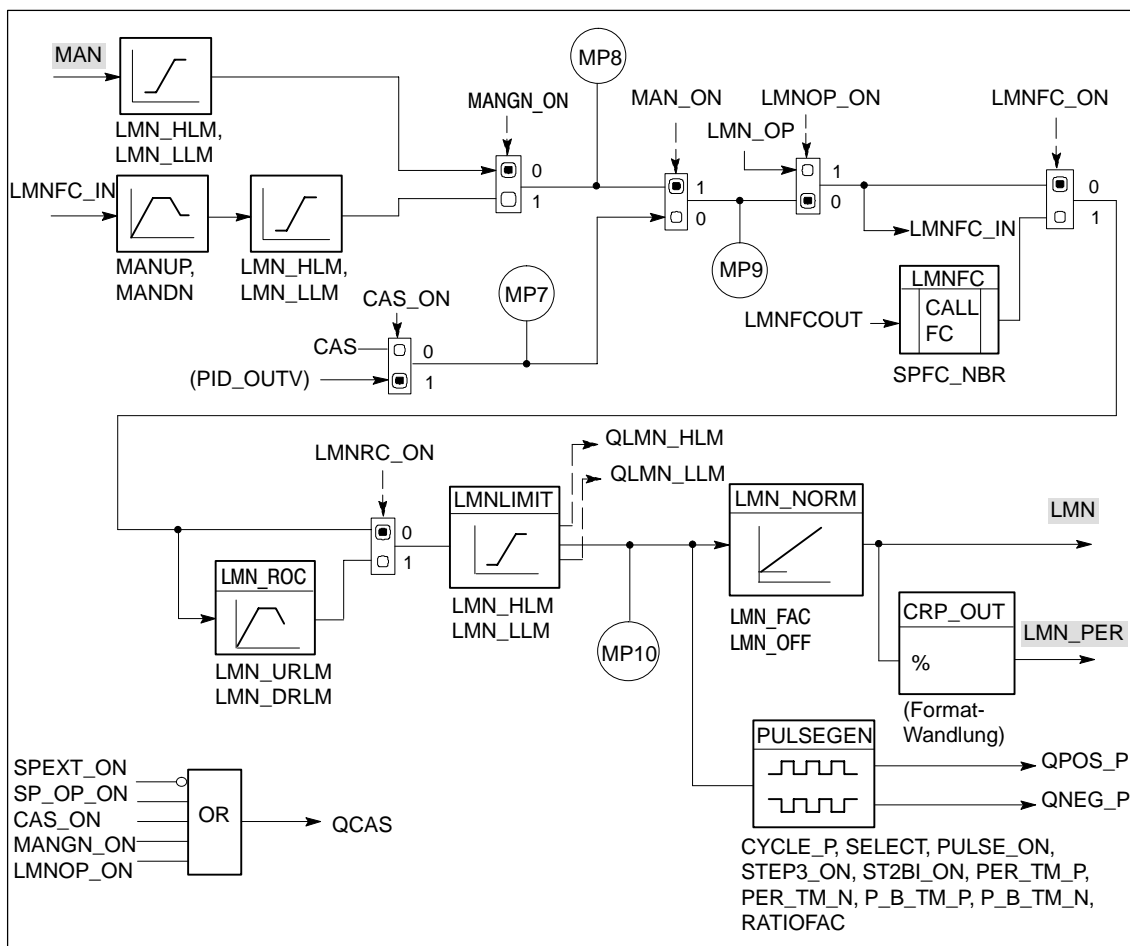


Bild 2-15 Signalflussplan der Stellsignalbildung des kontinuierlichen Reglers

## Stellsignalverarbeitung: Schrittreger mit Stellungsrückmeldung

- **Festeinstellung des Handwerts und Stellgrößenbegrenzung**

Die Funktionen zur Einstellung des Handwerts und zur Begrenzung des Absolutwerts der Ausgangsgröße sind die gleichen wie beim Regler mit kontinuierlichem Ausgang.

- **Bildung des binären Stellsignals (THREE\_ST, PULSEOUT)**

Der Dreipunktschalter THREE\_ST erzeugt je nach Vorzeichen der Eingangsdifferenz über die Pulsformerstufe PULSEOUT einen positiven oder negativen Ausgangsimpuls, der bis zum Verschwinden der Eingangsgröße anliegt. Die selbstadaptive Schalthysterese verhindert zu häufiges Schalten des Ausgangs.

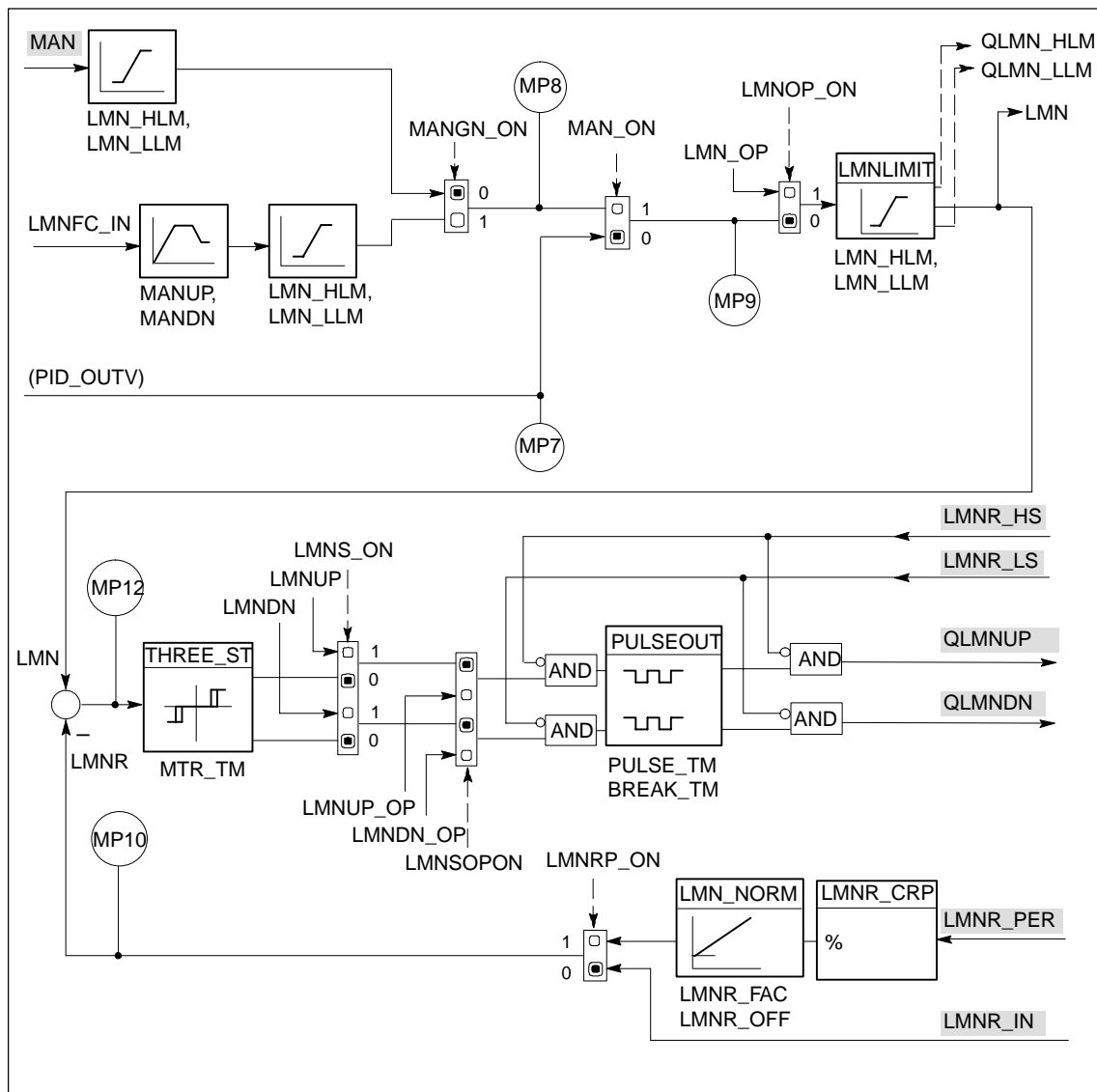


Bild 2-16 Signalflussplan der Stellgrößenbildung des Schrittregers mit Stellungsrückmeldung (LMNR\_ON = TRUE)

### Stellsignalverarbeitung: Schrittreger ohne Stellungsrückmeldung

- **Erzeugung des binären Stellsignals**

Die Bildung des Ausgangssignals durch Dreipunktschalter mit Hysterese und Impulsformerstufe ist bei allen Schrittreger identisch. Zeitparameter für die Berücksichtigung der Stellzeit des Motorantriebs (MTR\_TM) und die Einstellung der Impuls-/Pausendauer (PULSE\_TM und BREAK\_TM) sind einstellbar.

- **Simulation der Stellungsrückmeldung**

Die selbsttätige Ermittlung der Regelparameter durch die Optimierungsfunktion des Konfigurationswerkzeugs erfordert immer ein die Position des Stellgliedes repräsentierendes Signal als Eingangsgröße. Die dafür vorhandene Simulations-Funktion erfordert keine Parametrierung und ist für den Normalbetrieb des Schrittreger nicht relevant.

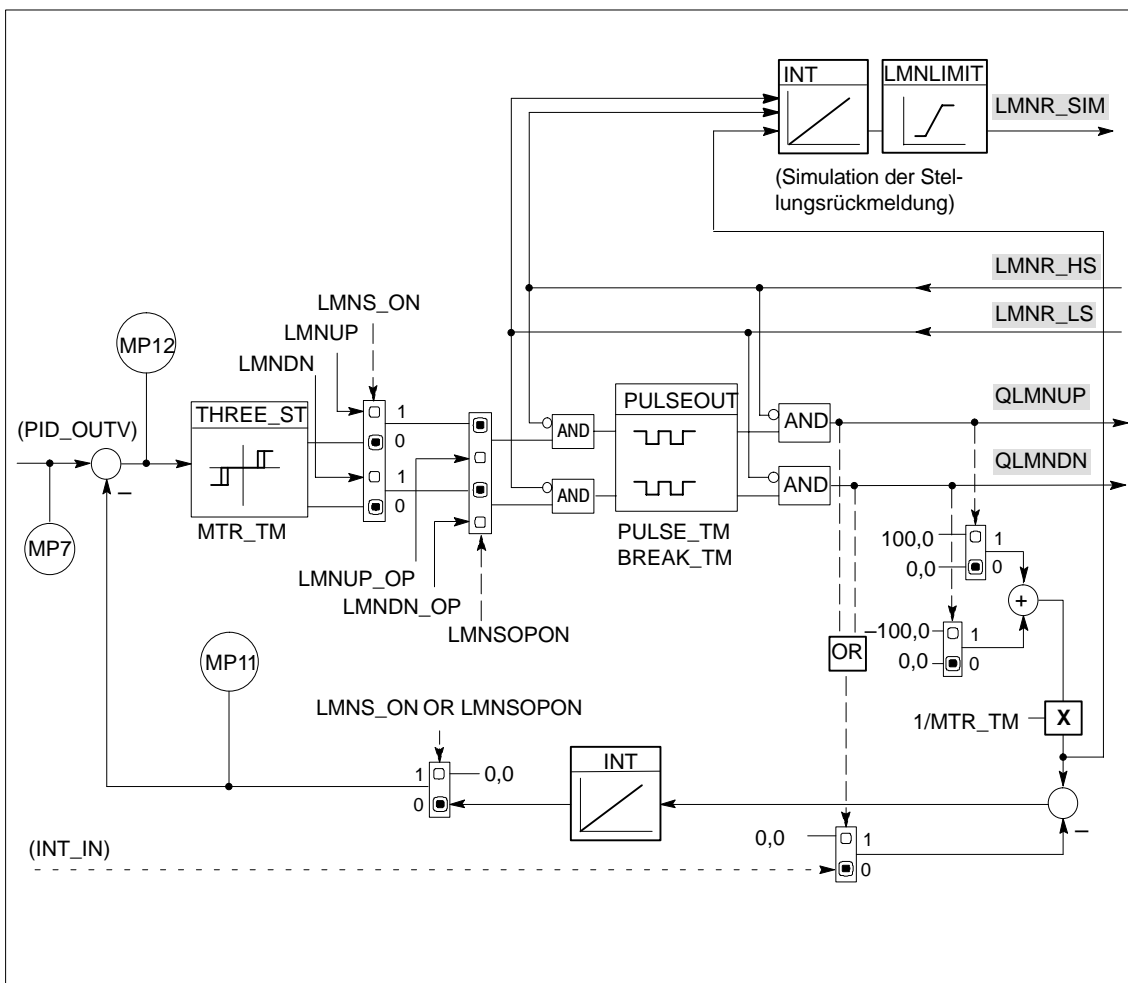


Bild 2-17 Blockschaubild Stellgrößenbildung des Schrittreger ohne Stellungsrückmeldung (LMNR\_ON = FALSE)

# Konfigurieren und Inbetriebsetzen von Standard PID Control

# 3

## 3.1 Definieren der Regelaufgabe

### Spezifizieren der Aufgabe

Vor der Realisierung eines Regelkreises mit Hilfe von Standard PID Control müssen Sie sich Klarheit verschaffen über die prozesstechnische Automatisierungsaufgabe, das einzusetzende Automatisierungssystem und das Bedien- und Überwachungsumfeld. Zur Spezifizierung der Aufgabe sind demnach erforderlich:

1. Kenntnisse über den zu regelnden Prozess, d. h. über die charakteristischen Kennwerte der Regelstrecke (Verstärkung, Ersatzzeitkonstante, Störgrößen usw.).
2. Festlegung der CPU, auf der Standard PID Control installiert und bearbeitet werden soll.
3. Definieren von Signalbearbeitungs- und Überwachungsfunktionen neben und zusätzlich zu den Kernfunktionen des Reglers.

Im Kapitel 2.1 wurde bereits auf Prozesseigenschaften und die Ermittlung von Kenngrößen für das Prozessverhalten eingegangen, so dass wir empfehlen, bei der Spezifizierung einer konkreten Aufgabe gegebenenfalls dort nachzulesen. Auch finden Sie dort Hinweise und Erläuterungen zur Ermittlung von Streckeneigenschaften und Reglerparametern mit Hilfe des Konfigurationswerkzeugs.

Der Einsatz des Konfigurationswerkzeugs entbindet Sie weitgehend von den Aufgaben (Punkt 1.) zur Ermittlung der Prozesskenngößen.

## Vorkenntnisse und Voraussetzungen

Da mit "Standard PID Control" Softwareregler auf der Grundlage von Standard-Funktionsbausteinen (hier PID\_CP bzw. PID\_ES) aus der S7-Bausteinwelt generiert werden, sind Kenntnisse über den Umgang mit S7-Bausteinen und den Aufbau von S7-Anwenderprogrammen (z. B. in der Programmiersprache S7-AWL) notwendig.

Obwohl die Funktionalität der zu realisierenden Regelung ausschließlich durch Parametrierung definiert wird, erfordert die Verschaltung des Reglerbausteins mit der Prozessperipherie sowie seine Einbindung in das Aufrufschema der CPU Kenntnisse, die in diesem Handbuches nicht behandelt werden.

Im einzelnen benötigen Sie:

- Informationen über das Arbeiten mit STEP 7 (*/231/*),
- Basiswissen über das Programmieren mit STEP 7 (*/232/*, */234/*),
- Daten über das verwendete Automatisierungssystem (*/70/*, */71/*, */100/*, */101/*).

## Die Regelstrecke

Bezüglich Art und Komplexität der mit Standard PID Control beherrschbaren Prozesse bestehen kaum Einschränkungen. Solange es sich um Eingrößensysteme ohne differenzierendes Übertragungsverhalten und ohne allpasshaltige Anteile handelt, können alle vorkommenden Streckentypen mit und ohne Ausgleich, d. h. mit I-Gliedern und ohne I-Glieder geregelt werden (Bild 3-1).



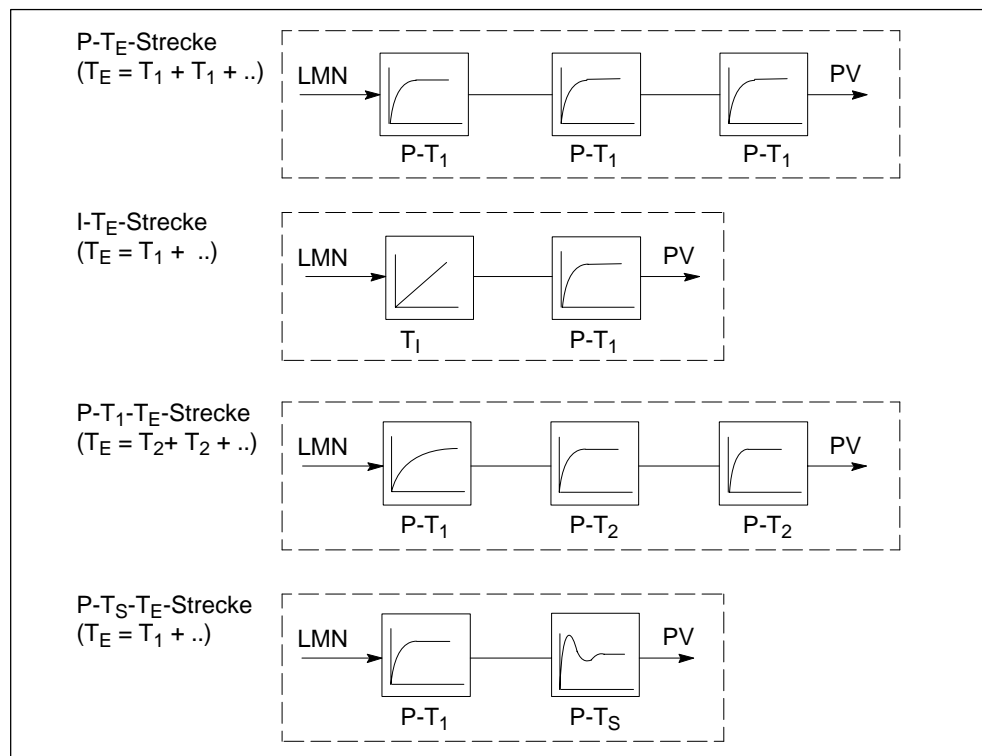


Bild 3-1 Mit Standard PID Control beherrschbare Regelstreckenarten

Die von Standard PID Control zu verarbeitende Prozessgröße PV ist in der Regel eine analoge physikalische Größe (Spannung, Strom, Widerstand usw.), die über eine zugeordnete S7 Analogeingabebaugruppe digitalisiert und in das einheitliche STEP 7 Peripheriesignal PV\_PER umgewandelt wird.

Die Werte dieser Signale werden in entsprechenden Speicherzellen bzw. Speicherbereichen des CPU-Anwenderspeichers abgelegt. Deren Adressierung kann absolut oder nach Eintrag in die Signalliste der CPU auch symbolisch erfolgen.

Liegt in Sonderfällen die Prozessgröße als Gleitpunktzahl vor, so kann dieser Wert als Regelgröße direkt mit dem Eingang PV\_IN verschaltet werden (Bild 3-2).

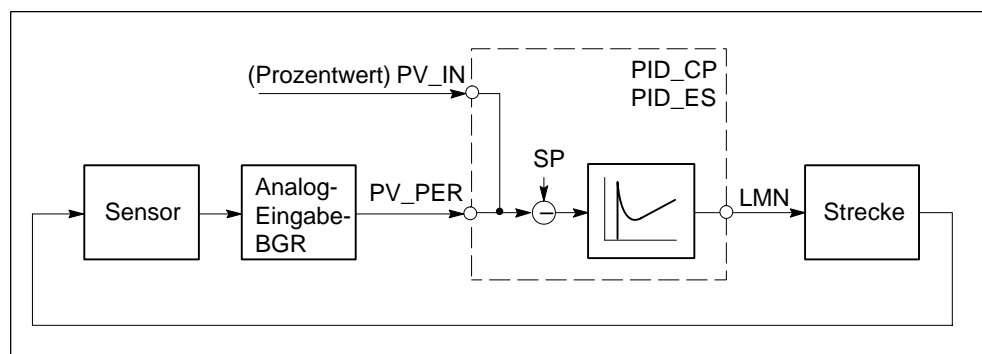


Bild 3-2 Verschaltung der Prozesssignale mit Standard PID Control

## Art des Stellgliedes

Um eine geeignete Konfiguration von Standard PID Control festzulegen, ist die Art des Stellgliedes wichtig, das zur Beeinflussung der betreffenden Prozessgröße verwendet wird. Die vom Stellglied geforderte Signalform bestimmt die Art der Signalbildung im Stellgrößenzweig (stetig oder unstetig) und damit die Art des einzusetzenden Reglers (kontinuierlicher Regler oder Schrittreger).

In den weitaus meisten Fällen werden Ventile oder Klappen zur Verstellung von Stoff- oder Energieströmen eingesetzt. Je nach Antrieb dieser Drosselorgane sind unterschiedliche Stellsignale erforderlich:

### 1. Proportionale Stellglieder mit stetigem Stellsignal

Proportional zum Wert der Stellgröße werden Öffnungsgrade, Drehwinkel oder Positionen eingenommen, d. h. innerhalb des Stellbereiches wirkt die Stellgröße in analoger Weise auf den Prozess ein.

Zu den Stellgliedern dieser Gruppe gehören federbelastete pneumatische Antriebe, aber auch motorische Antriebe mit Stellungsrückmeldung, bei denen ein Stellungsregelkreis gebildet wird.

### 2. Proportionale Stellglieder mit pulsbreitenmoduliertem Signal

Bei diesen Stellgliedern wird im Takt der Abtastzeit ein zum Wert der Stellgröße proportionales Impulslängensignal ausgegeben. Das bedeutet, das betreffende Stellglied – z. B. ein Heizwiderstand oder ein Kühlaggregat – wird takt synchron nach Maßgabe der Stellgröße unterschiedlich lange eingeschaltet.

Das Stellsignal kann entweder unipolar die Zustände "Ein" oder "Aus" annehmen oder bipolar z. B. die Werte "Auf/Zu", "Vorwärts/Rückwärts", "Beschleunigen/Bremsen" usw. repräsentieren.

### 3. Integral wirkende Stellglieder mit Dreipunktstellsignal

Häufig werden Stellglieder durch Motoren betätigt, bei denen die Dauer der Einschaltung proportional zum Verstellweg des Drosselorgans ist. Trotz unterschiedlichster Bauformen ist diesen Stellgliedern gemeinsam, dass sie der Wirkung eines I-Gliedes am Streckeneingang entsprechen. Standard PID Control mit Schrittausgang bietet die wirtschaftlichste Lösung beim Aufbau von Regelkreisen mit integral wirkenden Stellgliedern.

## Reglerwahl bezüglich der Art des Stellsignals

Standard PID Control bietet in Bezug auf die Art des erzeugten Stellsignals unterschiedliche Strukturen im Stellgrößenzweig.

- Stellglieder gemäß Punkt 1. und 2. der vorhergehenden Beschreibung werden mit dem Reglerbaustein PID\_CP geregelt. Falls ein impulsdauermoduliertes Signal benötigt wird, muss der Regler-FB noch durch den FB PULSEGEN ergänzt werden.
- Stellglieder mit I-Wirkung (Punkt 3.) werden mit dem Reglerbaustein PID\_ES geregelt. Wenn ein Rückmeldesignal der Stellgliedposition nicht verfügbar ist, wird die Reglerstruktur mit nachgebildetem Rückführsignal (LMNR\_ON = FALSE) eingesetzt.

Ist ein Signalgeber für die Position des Stellgliedes vorhanden, dann kann die Struktur mit Stellungsregelkreis (Bild 3-3 unten) konfiguriert werden (LMNR\_ON = TRUE).

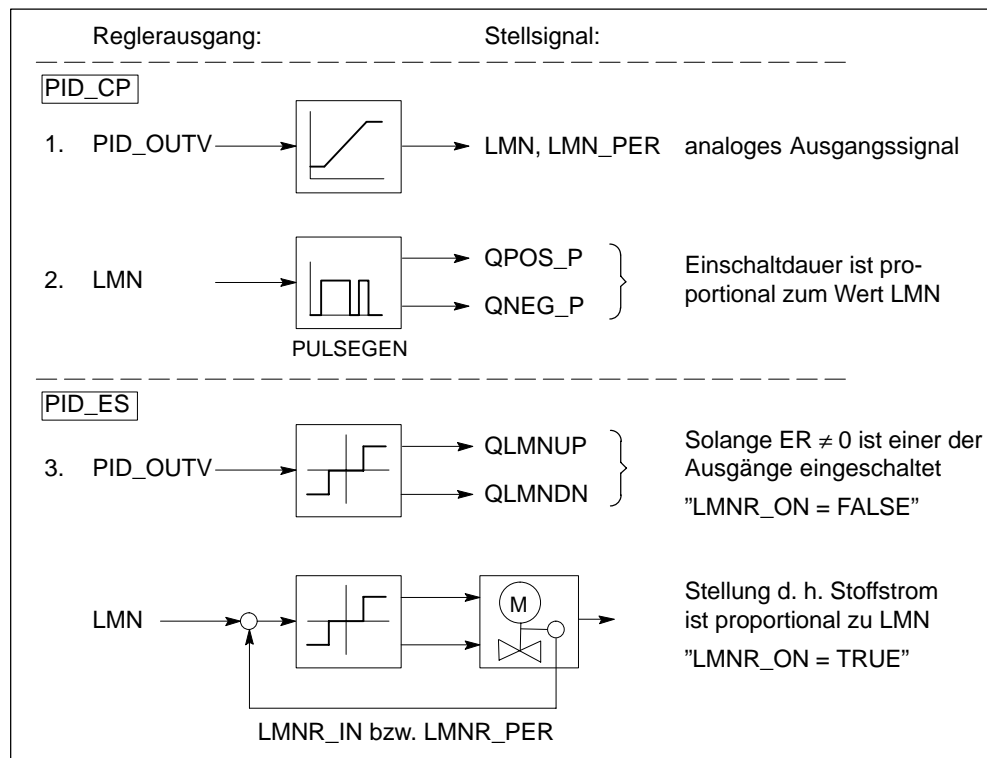


Bild 3-3 Stellausgänge von Standard PID Control

### Hinweis

Die Stellgrößen sind als digitale Zahlenwerte im Gleitpunkt- bzw. Peripherieformat oder als binäre Signalzustände repräsentiert. Je nach verwendetem Stellglied sind ausgangsseitig immer nachgeschaltete Baugruppen entsprechenden Typs zur Signalumformung und Bereitstellung der benötigten Stellenergie erforderlich.

## Stellsignal und Reglerbausteine

Der Zusammenhang zwischen Signalform der Stellgröße, Art der Regelung und zu deren Realisierung erforderlicher Konfiguration der Reglerbausteine ist in Tabelle 3-1 zusammengefasst.

Tabelle 3-1 Stellgröße, Art der Regelung und erforderliche Regler-Bausteine

Signalform der Stellgröße	Wertebereich	Art der Regelung	Reglerstruktur
proportional	Gleitpunkt 0.0 bis 100. 0 % bzw. Peripheriebereich	kontinuierlich	PID_CP
impulsdauermoduliert, bei 2-Pkt-Regelung Aus- gänge alternierend	bipolar bzw. unipolar positiver Ausgang: TRUE negativer Ausgang: FALSE	Dreipunkt-/Zwei- punkt-Regelung	PID_CP + PULSEGEN
dreipunkt-schaltend	Auf – 0 – Ab	Schrittregelung	PID_ES (LMNR_ON = FALSE)
dreipunkt-schaltend Stellungsrückmeldesignal	Auf – 0 – Ab 0 bis 100 % oder Periphe- riebereich	Schrittregelung mit Stellungsrück- meldung	PID_ES (LMNR_ON = TRUE)

Mit den vorstehenden Erläuterungen besitzen Sie nunmehr alle Informationen, um für Ihren konkreten Fall eine geeignete Konfiguration von Standard PID Control auszuwählen. Auf welcher Basis Sie das am besten machen und wie Sie interne Funktionen aktivieren und dimensionieren, beschreibt der folgenden Abschnitt.

## Ständig aktivierte nicht abschaltbare Funktionen

Die Funktionen zur Überwachung und Begrenzung der Signale in den Zweigen der Eingangs- und Ausgangssignalverarbeitung sind immer wirksam und können nicht abgeschaltet werden. Dazu gehören:

- die Sollwertbegrenzung SP\_LIMIT,
- die Istwertüberwachung auf Absolutwerte PV\_ALARM,
- die Istwertüberwachung auf Größe der Änderungsgeschwindigkeit ROC-ALARM,
- die Regeldifferenzüberwachung auf Absolutwerte ER\_ALARM,
- die Begrenzung des Ausgangssignals LMNLIMIT.

Wenn Sie sich für die Wahl eines der Reglerbausteine entschieden und dessen Ein-/Ausgänge definiert haben, müssen Sie sich deshalb immer um die Parametrierung der oben aufgeführten Funktionen kümmern.

---

### Hinweis

Die Vorbelegungswerte sind so gewählt (meist am Rande der zur Verfügung stehenden Arbeitsbereiche), dass der Betrieb auch ohne individuelle Parameterwahl aufgenommen werden kann. Die Parameter lassen sich dann anschließend an die Erfordernisse anpassen.

---

## 3.2 Projekt konfigurieren (Checkliste)

### Generieren des Regelungsprojektes durch Konfigurieren

Wenn Sie sich die geforderten Regel- und Überwachungsfunktionen erarbeitet haben (Informationen dazu in *Kapitel 2.5* und *3.1*), zeigen wir Ihnen in diesem Abschnitt die Vorgehensweisen zu deren Realisierung. Wir empfehlen einen Konfigurationsablauf in folgenden Schritten (Checkliste):

Schritt	Aktivität	Funktion in Standard PID Control	Erläuterungen und Bemerkungen
1.	Festlegen der Reglerbausteine bzw. der benötigten Bausteinkonfiguration für die gewünschte Regelungsstruktur. Auswählen und Kopieren einer Beispielkonfiguration, welche der zu realisierenden Konfiguration am nächsten kommt.	FB "PID_CP" oder "PID_ES" bzw. ein Beispiel aus Example1 bis Example6 oder Getting Started	– Kap. 3.3
2.	Auf Basis des ausgewählten Beispiels den gewünschten Regler durch Weglassen oder Hinzunehmen vorkonfektionierter oder eigenprogrammierter Funktionen konfigurieren.	– Im Blockschaltbild des Konfigurationswerkzeugs die Strukturschalter setzen; – oder im Instanz-DB die Schaltbits der Strukturschalter setzen (→Blockschaltbilder in Anhang A).	Die Datenstruktur der zugehörigen Instanz-DBs wird vom betreffenden FB geliefert.
3.	Abtastzeit und Aufrufverhalten des Regelkreises festlegen: – Anlaufverhalten über OB 100 festlegen – Abtastzeit und Aufrufebene bestimmen, ggf. das Zeitraster des Weckalarm-OB ändern – Abhängig von der Anzahl der Regelkreise in der CPU den Regleraufrufverteiler konfigurieren.	Parameter COM_RST  Parameter: CYCLE, Organisationsbautein: z. B. OB 3  Aufrufverteiler: LP_SCHED, in den Beispielen Example3 bis Example5 enthalten	– Kap. 3.4 und Kap. 3.5  – Kap. 7.1
4.	Für die Messbereichs- und Nullpunkt-Anpassung der Ein-/Ausgangssignale durch entsprechende Parametrierung der Normierungs und/oder Denormierungsfunktionen sorgen	– Normierung des externen Sollwerts (SP_NORM) – Normierung des externen Istwerts (PV_NORM) – Stellwert-Denormierung (LMN_NORM)	(→Kap. 4) – Kap. 3.6 – Kap. 3.6
5.	Sollwertzweig konfigurieren	– Sollwertgenerator (SP_GEN) – Zeiplangeber (RMP_SOAK) – Begrenzung der Sollwert-Änderungsgeschwindigkeit (SP_ROC) – Begrenzung der Sollwert-Absolutwerte (SP_LIMIT)	(→Kap. 4)  – Die Funktion ist ständig aktiv!

Schritt	Aktivität	Funktion in Standard PID Control	Erläuterungen und Bemerkungen
6.	Istwertzweig konfigurieren	<ul style="list-style-type: none"> <li>– Istwertverzögerung (LAG1ST)</li> <li>– Quadratwurzelbildung (SQRT)</li> <li>– Überwachung der Absolutwerte der Regelgröße (PV_ALARM)</li> <li>– Überwachung der Änderungsgeschw. der Regelgröße (ROC_ALARM)</li> </ul>	(→Kap. 4)  – Die Funktion ist ständig aktiv! – Die Funktion ist ständig aktiv!
7.	Bildung der Regeldifferenz konfigurieren	<ul style="list-style-type: none"> <li>– Totzone bzw. Unempfindlichkeitsbereich der Regeldifferenz (DEADBAND)</li> <li>– Überwachung der Regeldifferenz auf Absolutwerte (ER_ALARM)</li> </ul>	(→Kap. 4)  – Die Funktion ist ständig aktiv!
8.	Stellwertzweig für kontinuierlichen Regler konfigurieren   Stellwertzweig für Schrittreger konfigurieren	<ul style="list-style-type: none"> <li>– Handwertgenerator (MAN_GEN)</li> <li>– Begrenzung der Stellwertänderungsgeschwindigkeit (LMN_ROC)</li> <li>– Begrenzung der Stellwert-Absolutwerte (LMNLIMIT)</li> <li>– Handwertgenerator (MAN_GEN)</li> <li>– Bei Stellungsrückmeldung: Begrenzung der Stellwert-Absolutwerte (LMNLIMIT)</li> <li>– Betriebs-Parameter für Dreipunktglied und Pulsformerstufe (THREE_ST und PULSEOUT)</li> </ul>	(→Kap. 5)  – Die Funktion ist ständig aktiv!  (→Kap. 6) – Die Funktion ist ständig aktiv!
9.	Regler konfigurieren	<ul style="list-style-type: none"> <li>– PID-Reglerstruktur und PID-Parameter</li> <li>– Arbeitspunkt bei P- und PD-Reglern</li> <li>– Störgößenaufschaltung (DISV)</li> </ul>	(→Kap. 5)
10.	Im Bedarfsfall einfügen von Ergänzungsfunktionen – d. h. von anwendergeschriebenen FCs – in den Sollwert-, den Istwert- und/oder den Stellwertzweig.	<ul style="list-style-type: none"> <li>– SPFC (SPFC_ON = TRUE)</li> <li>– PVFC (PVFC_ON = TRUE)</li> <li>– LMNFC (LMNFC_ON = TRUE)</li> </ul>	
11.	Laden der konfigurierten Regelung in die CPU des AS	<ul style="list-style-type: none"> <li>– Im SIMATIC Manager das Projekt laden</li> </ul>	
12.	Falls gewünscht, Offlinetest der konfigurierten Regelung an der simulierten Verzögerungsstrecke 3. Ordnung durchführen.	<ul style="list-style-type: none"> <li>– Modellregelstrecke in Example1 und Example2 enthalten</li> </ul>	
13.	Verschalten der Bausteinein- und Bausteinausgänge der konfigurierten Regelung mit der Prozessperipherie.	<ul style="list-style-type: none"> <li>– Verschaltung der Ein-/Ausgänge mit den absoluten oder symbolischen Peripherieadressen im Anwenderspeicher der CPU programmieren.</li> </ul>	

Im Folgenden werden die Aktivitäten zur Konfiguration einzelner Funktionen bzw. Punkte der Checkliste, sofern sie der Erklärung bedürfen, näher erläutert.

In einem graphischen Parametrierplan finden Sie die Funktionalität von Standard PID Control mit allen Struktur- und Funktionsparametern zusammengefasst. Anhand dieses Plans können Sie für jede Funktion feststellen, welche Parameter es gibt und in welchen Bereichen sie eingestellt werden können.

### 3.3 Parametrieren von Standard PID Control

#### Parametrierplan für das Konfigurieren von Standard PID Control

Möchten Sie Ihre Konfiguration unmittelbar im Instanz-DB festlegen, dann liefern Ihnen die Parametrierpläne eine graphische Gesamtübersicht über die im Einzelfall auszuwählenden Funktionen und deren Parametrierung.

Zur Realisierung einer konkreten Regelung verweisen wir jedoch auf das Konfigurationswerkzeug, das Ihnen die Kontrolle über Ihre Eingaben und deren Vollständigkeit weitgehend abnimmt.

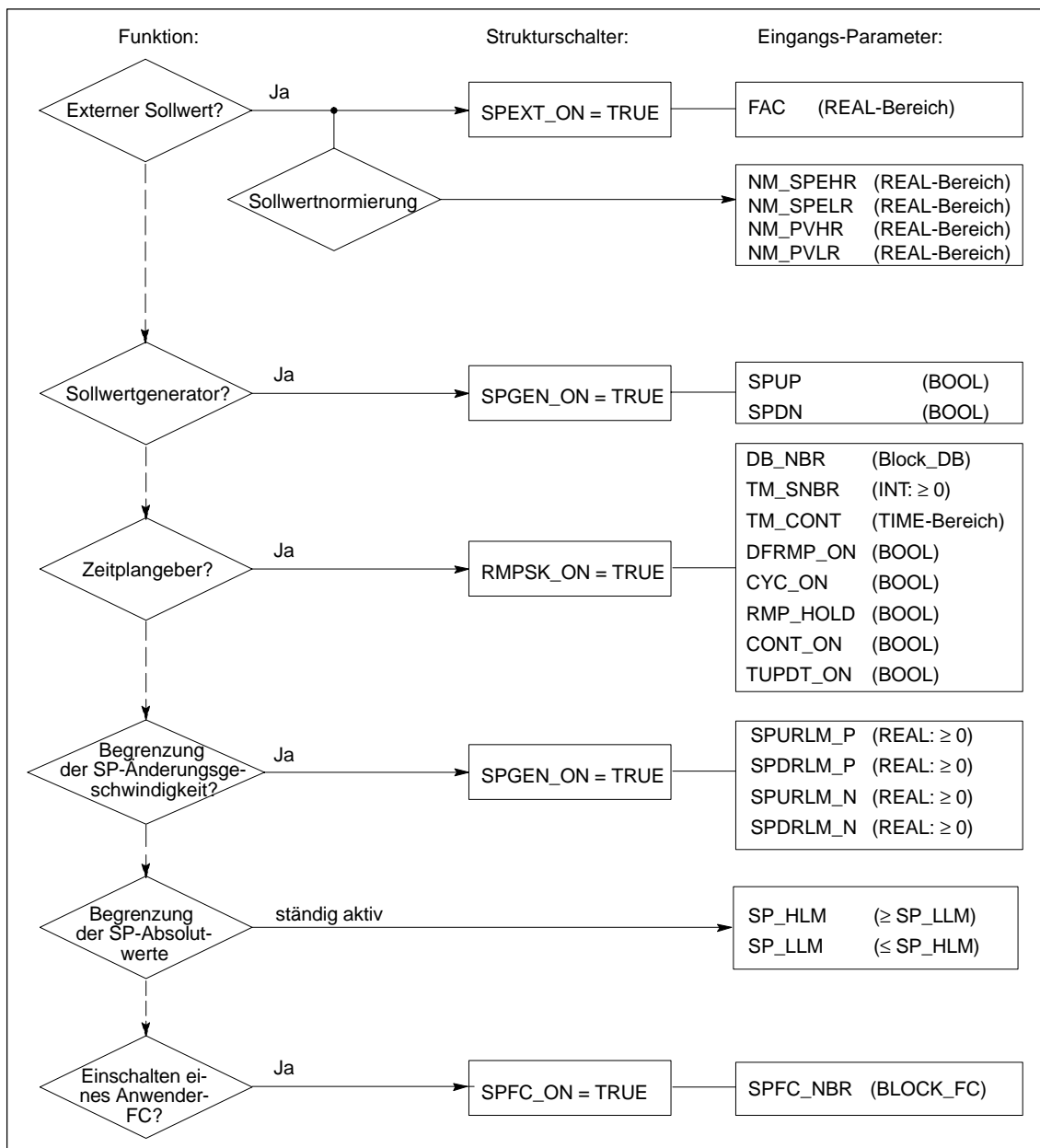


Bild 3-4 Konfiguration des Sollwertzweigs von Standard PID Control (Checkliste-P. 4. und 5.)



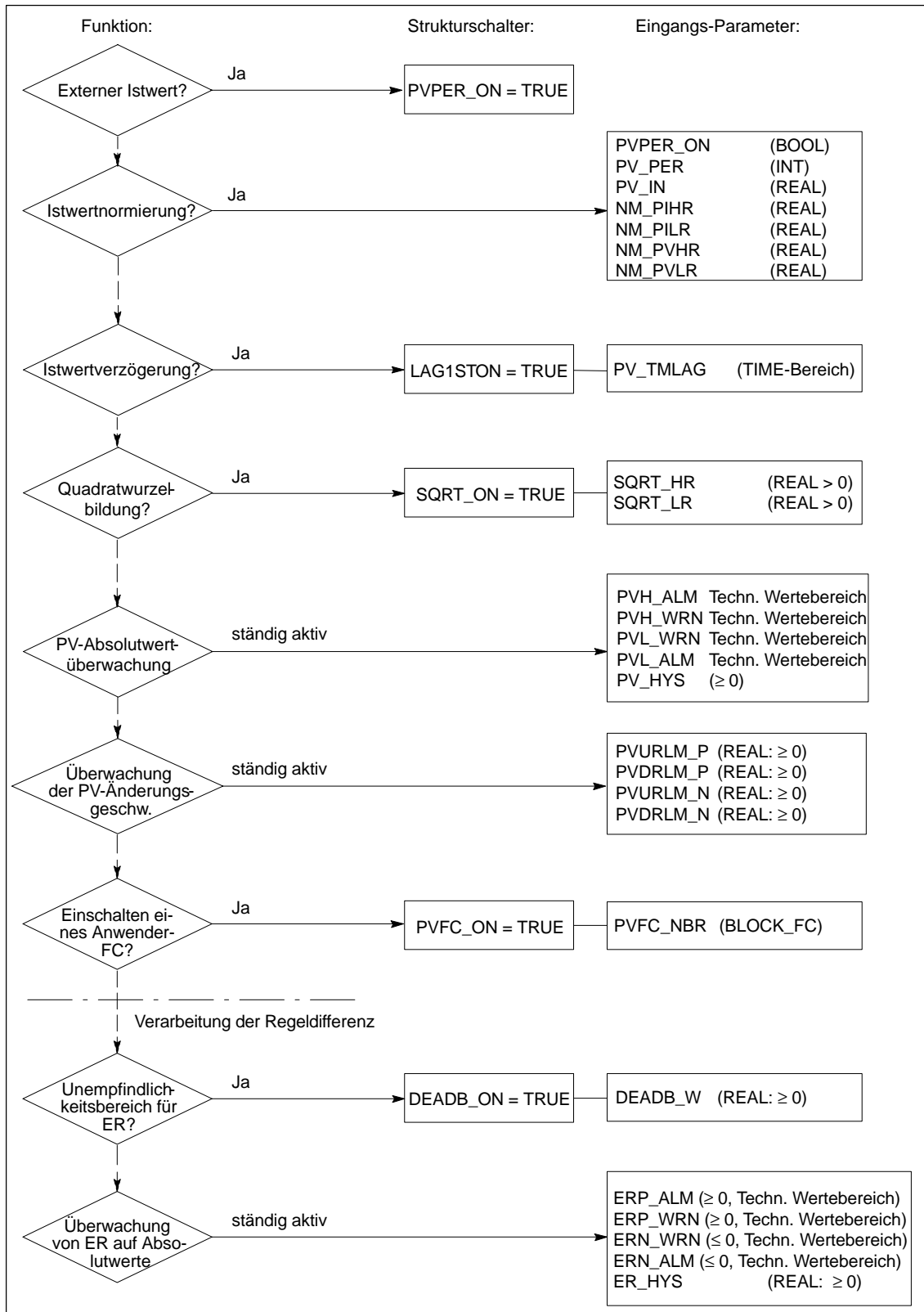


Bild 3-5 Konfiguration des Istwert- und Regeldifferenzzweigs von Standard PID Control (Checkl.-P. 6., 7. und 8.)

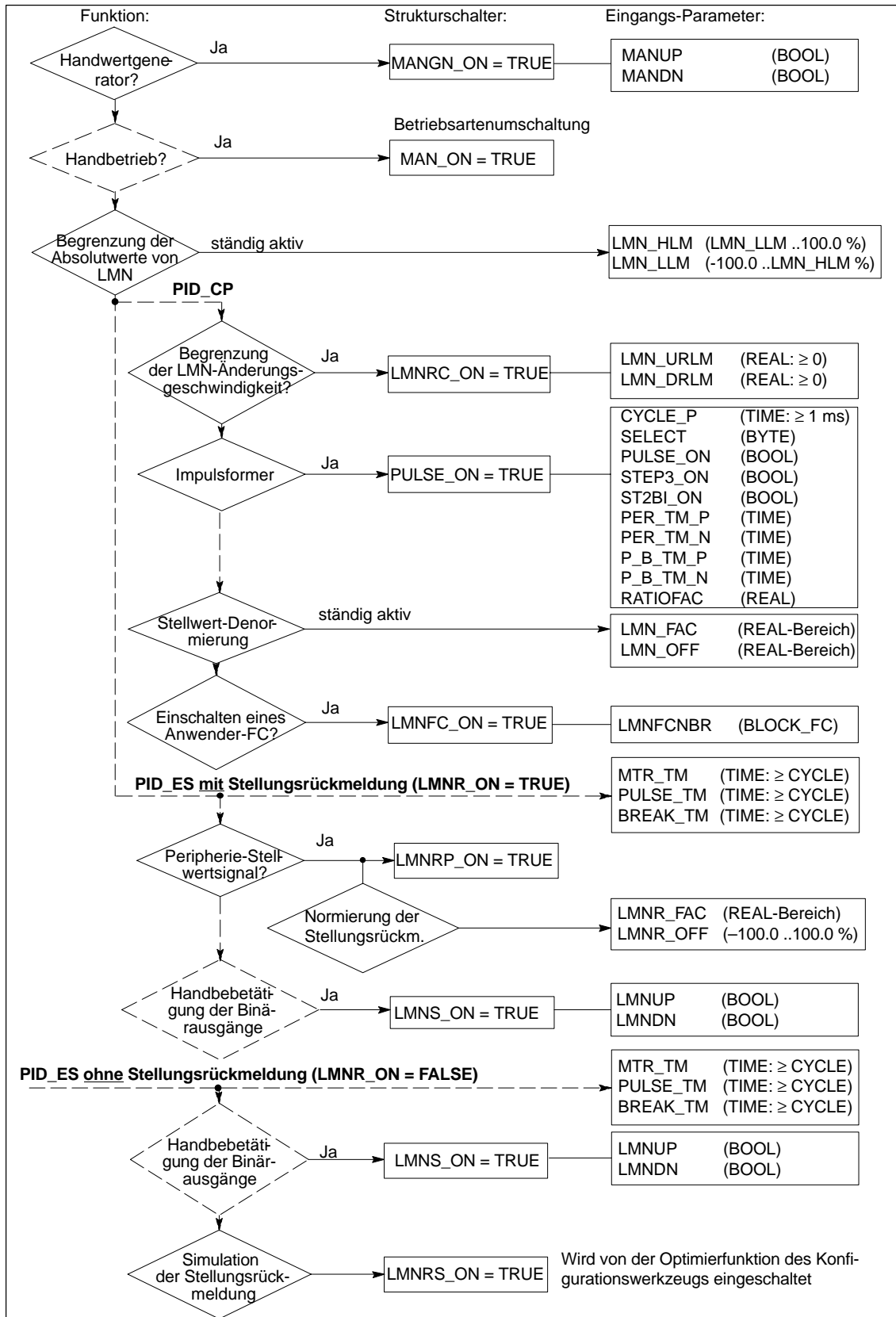


Bild 3-6 Konfiguration des Stellwertzweigs (Checklisten-P. 8.)

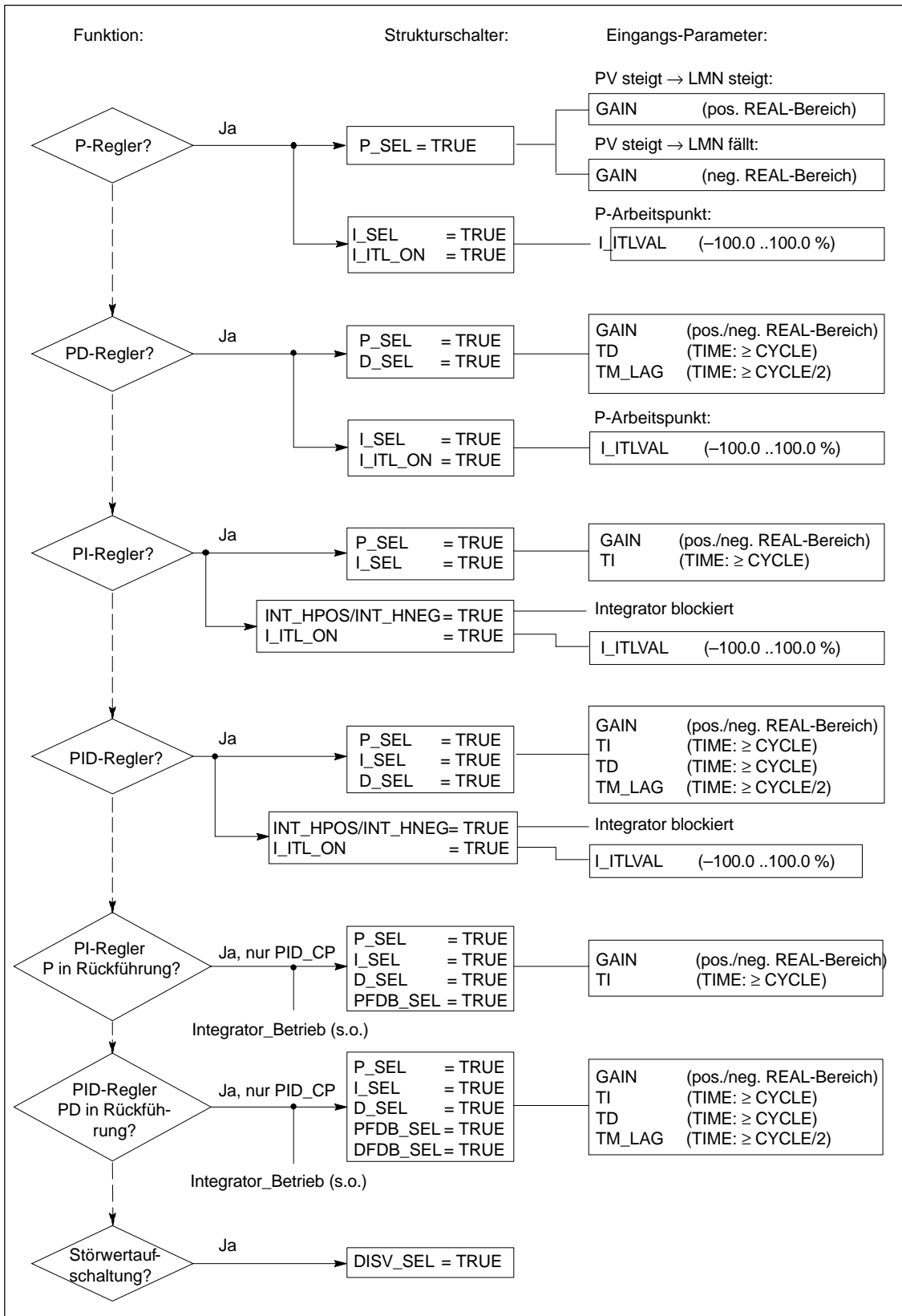


Bild 3-7 Konfiguration der Reglerfunktionen PID\_CP und PID\_ES (Checklisten-P. 9.)

## Das Konfigurationswerkzeug

Ist Ihnen der Aufwand für einen Konfigurierablauf gemäß Checkliste (*Kapitel 3.2*) oder nach den Angaben in vorstehenden Parametrierplänen zu unübersichtlich oder zu zeitaufwändig, dann empfehlen wir Ihnen, das "Konfigurationswerkzeug für Standard PID Control" zu Hilfe zu nehmen.

Das Konfigurationswerkzeug enthält folgende Tools, die für eine schnelle und dabei fehlerfreie Konfigurierung von Standard PID Control von großem Nutzen sind:

### Regelkreis-Editor

Im Blockschaltbild des Regelkreis-Editors sind die wichtigsten Funktionen von Standard PID Control als Blocksymbole dargestellt. Durch Anklicken der Schaltsymbole (dunkler Punkt) legen Sie darin auf einfachste Weise den gewünschten Signalfluss fest.

Nach Anklicken eines Funktionsblocks öffnet das System jeweils ein Dialogfeld, in dem Sie durch Eingaben in die Parameterfelder die betreffende Funktion dimensionieren. Wenn die betreffende Funktion im Blockschaltbild nicht explizit als Schaltfunktion dargestellt ist, aktivieren bzw. deaktivieren Sie sie über Schaltsymbole (Kreisymbol anklicken) bzw. Optionskästchen (ankreuzen).

## 3.4 Die Abtastzeit CYCLE

### Die Abtastzeit: CYCLE

Die Abtastzeit ist die Basis-Kenngröße für das dynamische Verhalten von Standard PID Control. Sie bestimmt, ob der betreffende Regler schnell genug auf Prozessänderungen reagiert und den Regler in allen Betriebszuständen beherrschen kann. Die zeitgebundenen Parameter von Standard PID Control werden jeweils durch den Wert der Abtastzeit nach unten begrenzt.

Die Wahl der Abtastzeit erfordert immer einen Kompromiss zwischen mehreren teils gegenläufigen Forderungen. Deshalb können hier dafür nur Richtwerte angegeben werden.

- Die CPU-abhängige Laufzeit für die Bearbeitung des Regelprogramms, d. h. für die Abarbeitung des Funktionsbausteins, stellt die unterste Grenze für die Abtastzeit  $CYCLE_{min}$  dar.
- Die tolerierbare Obergrenze für die Abtastzeit wird im Allgemeinen durch die Prozessdynamik festgelegt. Dabei ist die Prozessdynamik durch den Typ und die Kennwerte der Regelstrecke charakterisiert.

## Ersatz-Streckenzeitkonstante

Von wesentlichem Einfluss auf die Dynamik des Regelkreises ist die sogenannte Ersatzzeitkonstante der Regelstrecke ( $T_E$ ), die sich, nach Eingabe eines Sprungs DLMN, messtechnisch durch Aufnahme der Übergangsfunktion auf den Streckeneingang ermitteln lässt (Bild 3-8).

Der Streckenkennwert  $T_E$  ist eine brauchbare Annäherung für die effektive Verzögerung durch mehrere in Kette geschaltete P-T<sub>1</sub>-, P-T<sub>S</sub>- und T<sub>t</sub>-Glieder. Bei Reihenschaltungen von gleichen P-T<sub>1</sub>-Gliedern bildet er z. B. die Summe der Einzelzeitkonstanten.

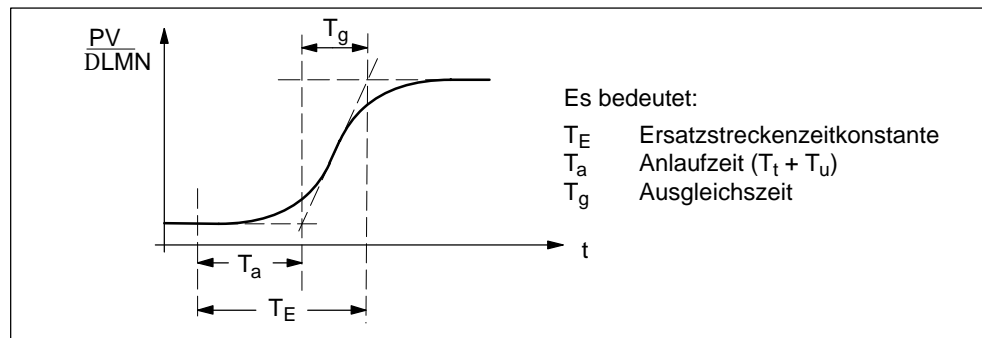


Bild 3-8 Ermittlung der Ersatz-Streckenzeitkonstante  $T_E$

## Abtastzeit-Abschätzung

Wird eine Mindestschnelligkeit der Regelung verlangt, kann man angeben, welchen Wert die Abtastzeit  $CYCLE_{max}$  nicht überschreiten darf.

Bei P-T<sub>E</sub>-Strecken mit überwiegend großer Zeitkonstante des ersten Verzögerungsgliedes  $T_1 \geq 0.5 T_E$  muss eingehalten werden:

$$CYCLE_{max} \leq 0.1 * T_E$$

Bei allen übrigen P-T<sub>E</sub>-Strecken genügt:

$$CYCLE_{max} \leq 0.2 * T_E$$

Für eine genaue Abschätzung der Abtastzeit siehe /352/.

## Faustformel für Wahl der Abtastzeit

Die Erfahrung hat gezeigt, dass eine Abtastzeit von ca.  $1/10$  der die Sprungantwort des geschlossenen Regelkreises bestimmenden Zeitkonstante  $T_{EG}$  zu einem mit dem konventionellen Analogregler vergleichbaren Ergebnis führt.

Die Gesamtzeitkonstante des geschlossenen Kreises wird, ähnlich wie in Bild 3-8 für die Strecke gezeigt, durch Sollwertsprung-Eingabe und Auswerten des Einschwingvorgangs von PV ermittelt.

$$CYCLE = \frac{1}{10} T_{EG}$$

## 3.5 Das Aufrufverhalten von Standard PID Control

### Bearbeitungsaufruf von Standard PID Control

Der Regler-Baustein muss, abhängig von der Abtastzeit des jeweiligen Reglers, unterschiedlich häufig, aber in gleichen zeitlichen Abständen aufgerufen werden. Das Betriebssystem der S7 ruft die Weckalarm-OBs zyklisch auf. Der Weckalarmzeittakt ist von 1 ms bis 1 Minute konfigurierbar. Die Standardeinstellung des OB 35 liegt bei 100 ms.

Sind mehrere Regler oder Regler mit größeren Abtastzeiten zu installieren, dann wird der Einsatz des Regleraufrufverteilers (LP\_SCHED) erforderlich.

#### Neustart:

Beim Aufruf des Regler-FB im Neustart (OB 100) wird das Neustartbit COM\_RST gesetzt und die Abtastzeit CYCLE übergeben. Das Bearbeiten der Initialisierungsroutine im FB stellt dann einen definierten Ausgangszustand von Standard PID Control sicher.

#### Wiederanlauf:

Bei Wiederanlauf wird von dem Betriebszustand ausgegangen, der zum Zeitpunkt der Unterbrechung geherrscht hat. Die Regelung arbeitet mit den Werten weiter, die im Augenblick der Unterbrechung berechnet waren.

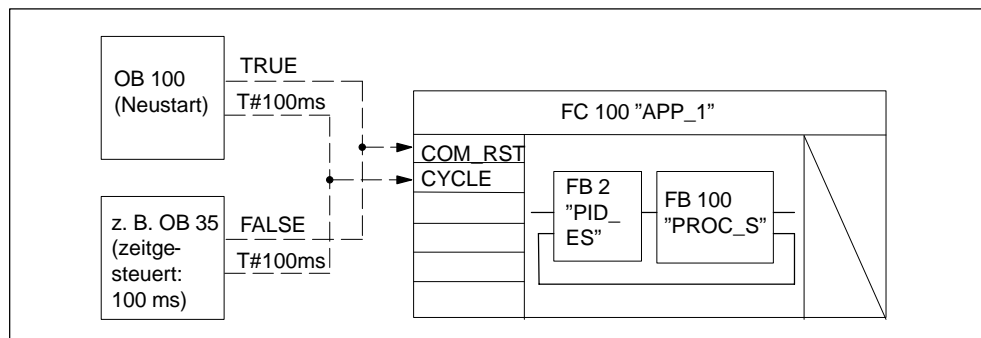


Bild 3-9 Verschaltung der Anlaufbausteine mit dem Beispiel APP\_1

#### Hinweis

Beim kontinuierlichen Regler PID\_CP ohne Impulsausgang wird die Abtastzeit über den Aufrufparameter CYCLE parametrier.

Beim kontinuierlichen Regler PID\_CP mit Impulsausgang wird am Aufrufparameter CYCLE\_P der Weckalarmzeittakt bzw. die über den Regleraufrufverteiler vorgegebene Abtastzeit parametrier (siehe Abschnitt 5.4.).

## Aufruf mit Regleraufrufverteiler

Wenn die Weckalarme des Prioritätsklassenmodells nicht ausreichen, um die gewünschte Anzahl von Reglern unterzubringen, oder wenn es Regler mit größeren Abtastzeiten als der längste Zeittakt der vorhandenen Weckalarme gibt, dann wird es erforderlich, einen Regleraufrufverteiler in den Weckalarm-OB einzufügen.

Der Aufrufverteiler LP\_SCHED macht es möglich, mehrere Regler in einer Weckalarmebene unterzubringen. Diese lassen sich unterschiedlich häufig, aber jeweils in gleichen Zeitabständen aufrufen (*siehe Kapitel 7.1*). Es wird eine gleichmäßigere Auslastung des Prozessors erreicht.

Die Eintragungen der Regleraufrufe in den globalen Datenbaustein mit der Nummer "DB\_NBR" legen fest, in welcher Reihenfolge und wie oft die einzelnen Regler bearbeitet werden (Bild 3-10). Einzelheiten zur Parametrierung des LP\_SCHED lesen Sie bitte im *Kapitel 7.1* dieses Handbuchs nach.

Die Parametrierung erfolgt mit STEP 7. Sie kann nicht mit dem Konfigurationswerkzeug vorgenommen werden.

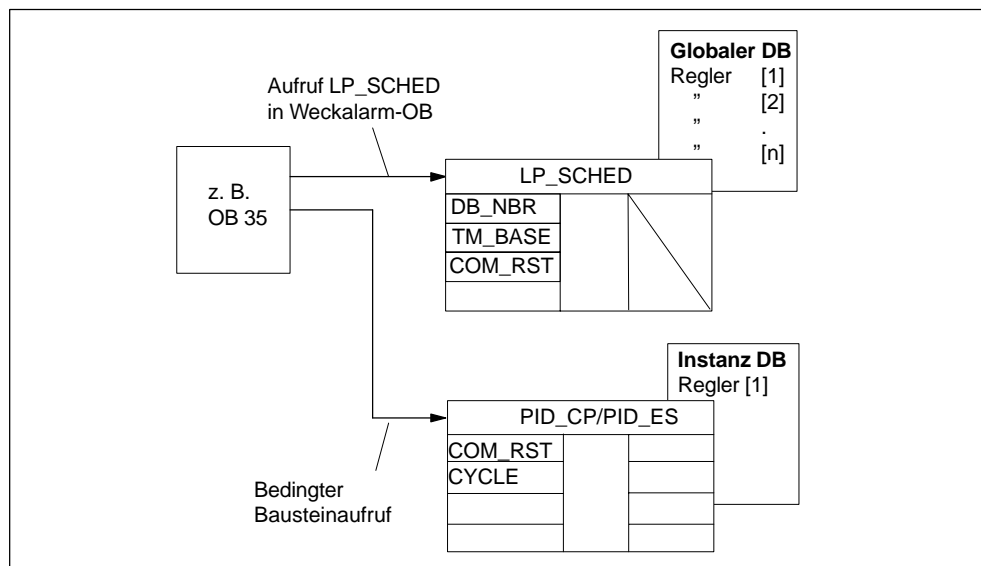


Bild 3-10 Regleraufruf über den Aufrufverteiler LP\_SCHED

## 3.6 Wertebereiche und Signalanpassung (Normierung)

### Interne Zahlendarstellung

Wenn die Algorithmen in den Funktionsbausteinen von Standard PID Control abgearbeitet werden, arbeitet der Prozessor mit Zahlen im Gleitpunktformat (REAL). Die Gleitpunktzahlen haben das Single Format nach ANSI/IEEE Std 754-1985:

Format:	DD (32 Bit)
Wertebereich:	$-3.37 * 10^{38}$ bis $-8.43 * 10^{-37}$ und $8.43 * 10^{-37}$ bis $3.37 * 10^{38}$

Dieser Zahlenbereich wird bei Parametern im Format REAL mit "gesamter Wertebereich" bezeichnet. Zur Vermeidung von Bereichsüberschreitungen bei der Verarbeitung wird das als analoge physikalische Größe anliegende Eingangssignal SP\_EXT in einem eingeschränkten Bereich als "Technischer Wertebereich" definiert:

Techn. Wertebereich:  $-10^5$  bis  $+10^5$

Zeitbehaftete Größen werden im Format TIME realisiert und verarbeitet. Ein Zeitwert stellt eine 32 Bit breite BCD-Zahl dar, bei der die vier höchstwertigen Bits für die Angabe des Zeitrasters reserviert sind.

Format:	DD (32 Bit)
Wertebereich:	0 bis +9 999 999 sec
Zeitraster:	10 ms, 100 ms, 1 sec, 10 sec

### Signalanpassung

Die Normierungsfunktion am Eingang für den externen Sollwert ermöglicht es, beliebige Kennlinien von Gebern oder Sensoren an den physikalischen Wertebereich von Standard PID Control anzupassen.



# Signalverarbeitung im Soll-/Istwertkanal und die PID-Reglerfunktionen

# 4

## 4.1 Signalverarbeitung im Sollwertzweig

### 4.1.1 Sollwert-Generator (SP\_GEN)

#### Anwendung

Mit Hilfe eines Mehr-/Weniger-Schalters können Sie den internen Sollwert einstellen. Der eingestellte Wert wird an MP1 zu beobachten.

#### Die Funktion SP\_GEN

Die Funktion SP\_GEN erzeugt einen Sollwert, der per Schalterbetrieb eingestellt und verändert werden kann. Über die Binäreingänge SPUP und SPDN lässt sich die Ausgangsgröße outv schrittweise vergrößern oder verkleinern.

Der Verstellbereich des Sollwertes wird durch die obere/untere Begrenzung SP\_HLM/SP\_LLM im Sollwertzweig eingeschränkt. Die Zahlenwerte der Grenzen (in Prozent) werden an den betreffenden Eingangsparametern eingestellt. Die Meldeausgänge QSP\_HLM und QSP\_LLM zeigen Überschreitungen dieser Grenzen an.

Für eine feinstufige Änderung sollte der Regler eine Abtastzeit von maximal 100 ms haben.

Die Änderungsgeschwindigkeit der Ausgangsgröße hängt von der Betätigungsdauer der Schalter SPUP bzw. SPDN und von den eingestellten Begrenzungen nach folgender Beziehung ab:

Während der ersten 3 s nach Setzen von SPUP bzw. SPDN ist:

$$\frac{doutv}{dt} = \frac{SP\_HLM - SP\_LLM}{100 \text{ s}}$$

danach:

$$\frac{doutv}{dt} = \frac{SP\_HLM - SP\_LLM}{10 \text{ s}}$$

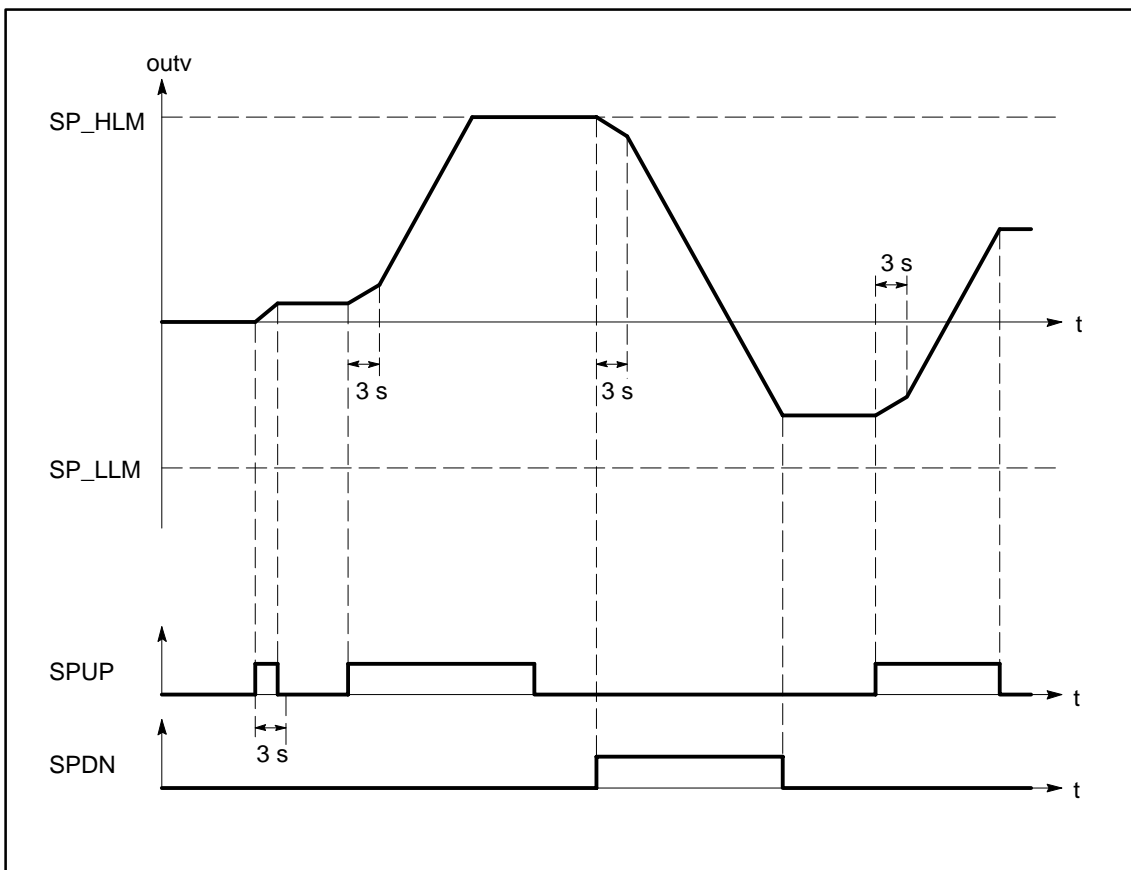


Bild 4-1 Änderung von OUTV in Abhängigkeit von den Schaltern SPUP und SPDN

Bei einer Abtastzeit von 100 ms und einem Sollwertbereich von  $-100,0$  bis  $100,0$  ändert sich z. B. der Sollwert in den ersten drei Sekunden um  $0,2$  pro Durchlauf. Bei längerer Tastung von SPUP geht die Änderungsgeschwindigkeit dann auf den zehnfachen Wert, hier also auf  $2$  pro Durchlauf (Bild 4-1).

### Anlauf- und Betriebsweise des Sollwertgenerators

- Bei Neustart wird der Ausgang outv auf  $0.0$  zurückgesetzt.
- Schalten Sie den Sollwertgenerator ein (SPGEN\_ON = TRUE), dann wird am Ausgang outv zunächst der Signalwert SPFC\_IN ausgegeben. Das Umschalten zum Sollwertgenerator aus einer anderen Betriebsart ist deshalb immer stoßfrei. Solange die Schalter SPUP und SPDN (Auf-/Ab-Tasten) nicht aktiviert werden, bleibt SPFC\_IN am Ausgang stehen.

## Parameter der Funktion SP\_GEN

Der Ausgangsparameter outv ist ein impliziter Parameter. Er ist am Konfigurationswerkzeug über den Messpunkt MP1 zu beobachten.

Parameter	Bedeutung	Zulässiger Wertebereich
SPFC_IN	Sollwert FC Eingang	technischer Wertebereich
SP_INT	interner Sollwert	technischer Wertebereich

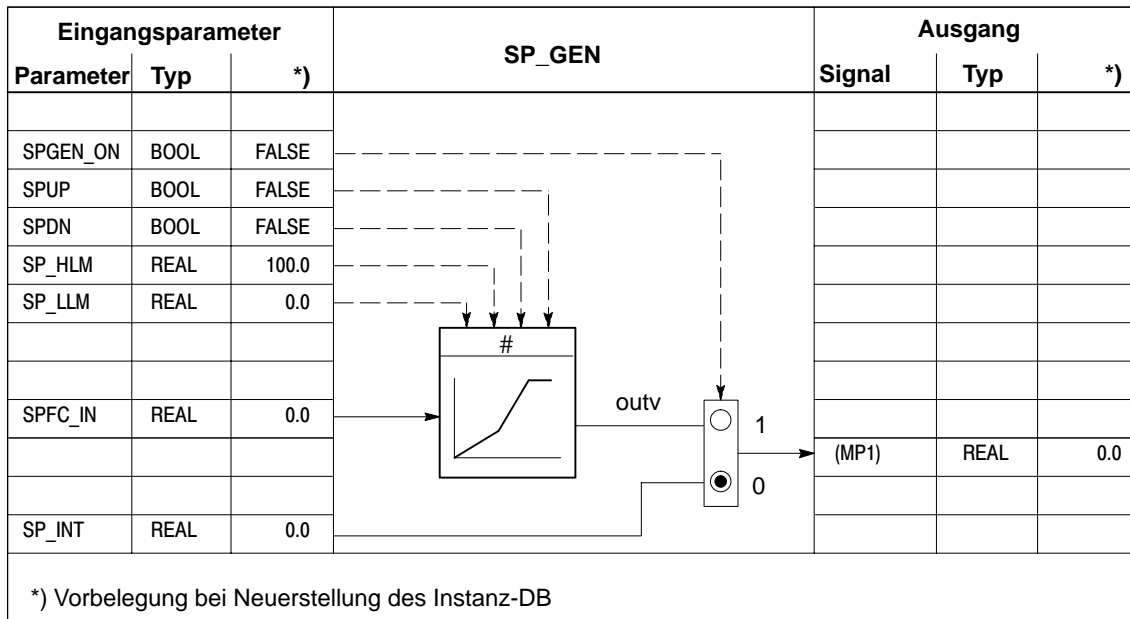


Bild 4-2 Funktionsschema und Parameter des Sollwert-Generators

### 4.1.2 Zeitplangeber (RMP\_SOAK)

#### Anwendung

Falls die Führungsgröße SP\_INT selbsttätig zeitabhängig geändert werden soll, z. B. bei der Regelung von Prozessen nach einem zeitgesteuerten Temperaturprogramm, so ist dies durch Projektieren einer entsprechenden Fahrkurve und Aktivieren des Zeitplangebers RMP\_SOAK möglich. Die Fahrkurve wird aus Geradenabschnitten mit max. 256 Stützpunkten gebildet.

## Die Funktion RMP\_SOAK

Der Zeitplangeber RMP\_SOAK im Sollwertzweig liefert eine nach einem definierten Zeitplan ablaufende Funktion der Ausgangsgröße outv (Bild 4-3). Gestartet wird diese Funktion durch Setzen des Eingangsbits RMPSK\_ON. Ist das Bit für die zyklische Wiederholung CYC\_ON gesetzt, dann wird die Funktion nach Ausgabe des letzten Stützpunktes outv[NBR\_PTS] wieder mit dem ersten Stützpunkt outv[1] angefangen. Zwischen dem letzten und dem ersten Stützpunkt wird bei zyklischer Wiederholung nicht interpoliert.

Der Ablauf des Zeitplans (der Fahrkurve) wird durch Festlegen einer Reihe von Stützpunkten in einem globalen Datenbaustein mit den Zeitwerten  $PI[i].TMV$  und den zugehörigen Ausgangswerten  $PI[i].OUTV$  definiert (Bild 4-3).

$PI[i].TMV$  gibt die Zeitabstände von Stützpunkt zu Stützpunkt an. Zwischen den Stützpunkten wird linear interpoliert.

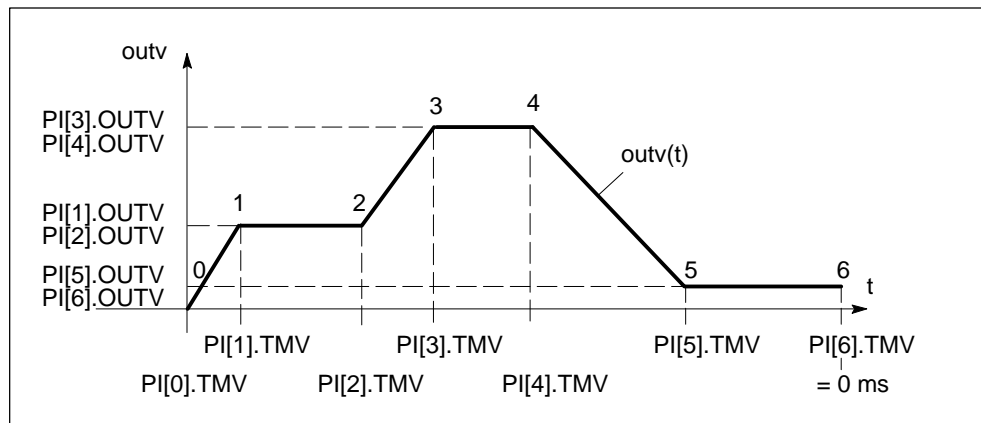


Bild 4-3 Fahrkurve mit Startpunkt und sechs Stützpunkten

### Hinweis

Bei  $n$  Stützpunkten ist der Zeitwert  $PI[n].TMV$  für den letzten Stützpunkt  $n = 0$  ms (Bearbeitungsende). Die Bearbeitungszeit einer Fahrkurve wird ausgehend vom Startwert bis auf Null heruntergerechnet.

### Hinweis

Beim Interpolieren der Fahrkurve zwischen den Stützpunkten kann es zum zeitweisen Verharren des Ausgangswerts kommen, wenn die Abtastzeit CYCLE sehr klein gegenüber der Zeit zwischen den Stützpunkten  $PI[n].TMV$  ist. Der Zeitplangeber kann aufgrund der CPU-Rechengenauigkeit nicht beliebig flache Fahrkurven linearisieren. Ist die Fahrkurve zu flach, verharrt der Ausgangswert eine Zeit lang am jeweiligen Stützpunkt, um nach einer gewissen Zeit mit der minimalen Steilheit zum nächsten Stützpunkt zu integrieren.

Abhilfe: Verkürzen Sie die Zeit zwischen den Stützpunkten, indem Sie weitere Stützpunkte einfügen. Sie nähern damit die ausgegebene Fahrkurve der gewünschten flachen Fahrkurve trapezförmig an.

## Zeitplanerstellung

- Die Stützpunktparameter NBR\_PTS, PI[i].TMV und PI[i].OUTV werden in einem globalen Datenbaustein hinterlegt.
- Der Parameter PI[i].TMV ist im Format der IEC-Zeit (TIME) anzugeben.
- Die Zählweise der maximal bis zu 256 Stützpunkt-Werte und -Zeiten wird in folgendem Bild deutlicher zu erkennen:

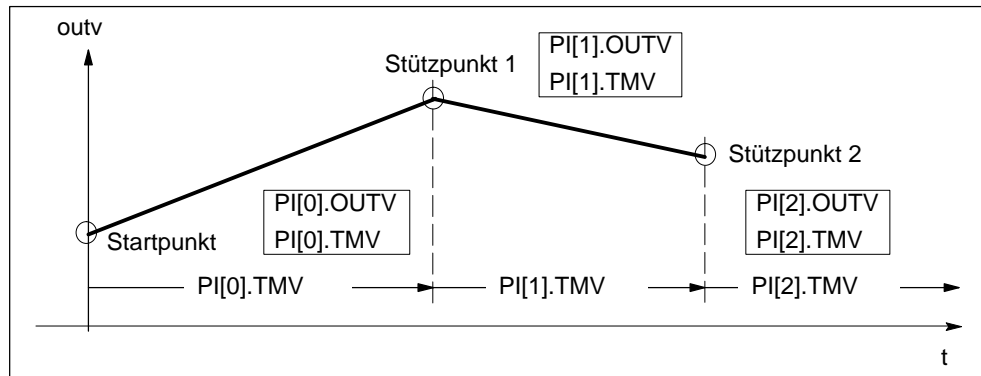


Bild 4-4 Zählweise der Stützpunkte und Zeiten

Der Zeitplangeber interpoliert im Normalbetrieb nach folgender Funktion mit  $0 \leq n \leq (\text{NBR\_PTS} - 1)$ :

$$\text{outv}(t) = \text{PI}[n + 1].\text{OUTV} - \frac{\text{RS\_TM}}{\text{PI}[n].\text{TMV}} (\text{PI}[n + 1].\text{OUTV} - \text{PI}[n].\text{OUTV})$$

## Projektierung der Fahrkurve

Die Anzahl der projizierten Stützpunkte (NBR\_PTS) und die den einzelnen Stützpunkten zugeordneten Werte für die Führungsgröße SP – zu beobachten an MP1 – werden in einem globalen Datenbaustein mit der Nummer DB\_NBR abgelegt (Tabelle 4-2). Die Ausgabe der Fahrkurve beginnt mit dem Startpunkt [0] und endet mit dem Stützpunkt [NBR\_PTS].

## Betriebsarten des Zeitplangebers

Über entsprechende Beeinflussung der Steuereingänge können folgende Betriebszustände und Betriebsarten des Zeitplangebers realisiert werden:

1. Zeitplangeber einschalten für einmaligen Durchlauf
2. Ausgang des Zeitplangebers mit festem Wert (z. B. SP\_INT) vorbelegen
3. Zyklischen Betrieb des Zeitplangebers einschalten
4. Bearbeitung des Zeitplangebers anhalten
5. Bearbeitungsschritt und -zeit vorgeben (die Restzeit RS\_TM und die Stützpunktnummer TM\_SNBR werden neu definiert)
6. Gesamtbearbeitungszeit und Gesamtrestzeit aktualisieren


## Betriebsarten

Zur Einstellung einer gewünschten Betriebsart gilt für die Wertigkeit der Steuereingänge folgende Wahrheitstabelle (Tabelle 4-1):

Tabelle 4-1 Betriebsarten des Zeitplangebers (RMP\_SOAK)

Betriebsart	RMPSK_ON	DFRMP_ON	RMP_HOLD	CONT_ON	CYC_ON	TUPDT_ON	Ausgangssignal OUTV
1. Zeitplangeber einschalten	TRUE	FALSE	FALSE		FALSE		outv(t) Endwert wird nach Bearbeitungsende gehalten.
2. Ausgang vorbelegen	TRUE	TRUE					SP_INT oder Ausgang von SP_GEN
3. Zyklischen Betrieb einschalten	TRUE	FALSE	FALSE		TRUE		outv(t) nach Ende: automatischer Start
4. Zeitplangeber anhalten	TRUE	FALSE	TRUE	FALSE			Aktueller Wert von outv(t) wird gehalten *)
5. Bearbeitungsschritt vorgeben Bearbeitungszeit vorgeben	TRUE	FALSE	TRUE	TRUE			outv (alt) *)
			FALSE				Mit neu vorgegebenen Werten weiterfahren
6. Gesamtzeit aktualisieren						FALSE	beeinflusst outv nicht
						TRUE	beeinflusst outv nicht

\*) Bis zum nächsten Stützpunkt hat die Fahrkurve nicht die vom Anwender parametrisierte Form!

 Die jeweils eingestellte Betriebsart wird unabhängig von der Wertigkeit der Steuersignale in den schraffierten Feldern ausgeführt.

## Einschalten des Zeitplangebers

Mit dem Wechsel RMPSK\_ON von FALSE nach TRUE wird die Zeitplangeberfunktion eingeschaltet (Softwareschalter im Blockschaltbild des Konfigurationswerkzeugs). Nach Erreichen des letzten Stützpunktes ist die Fahrkurve beendet. Bei erneutem Start der Funktion durch den Bediener muss RMPSK\_ON zunächst auf 'FALSE' und dann wieder auf 'TRUE' gesetzt werden.

Bei **Neustart** wird der Ausgang outv auf 0.0 zurückgesetzt und die Gesamtzeit bzw. Gesamtrestzeit ermittelt. Beim Übergang in den Normalbetrieb wird sofort die Fahrkurve vom Startpunkt aus nach eingestellter Betriebsart bearbeitet. Wird dies nicht gewünscht, muss der Parameter RMPSK\_ON im Neustart-OB auf FALSE gesetzt werden.



### Gefahr

Die Bausteinfunktion überprüft nicht, ob ein globaler DB mit der Nummer 'DB\_NBR' vorhanden ist und ob der Parameter 'Anzahl der Stützpunkte' NBR\_PTS zur DB-Länge passt. Bei inkorrekt parametrisierter CPU geht die CPU mit einem internen Systemfehler in **STOP**.

## Ausgang vorbelegen, Fahrkurve starten

Falls DFRMP\_ON = TRUE, dann wird der Ausgangswert des Zeitplangebers auf den Signalwert SP\_INT bzw. den Ausgangswert von SP\_GEN gesetzt. Mit DFRMP\_ON = FALSE läuft die Kurve ab diesem Wert los.

### Hinweis

Der Schalter DFRMP\_ON wirkt sich nur dann aus, wenn der Zeitplangeber eingeschaltet ist (RMPSK\_ON = TRUE).

Nach dem Umschalten DFRMP\_ON = FALSE wird outv linear vom eingestellten Sollwert (z. B. SP\_INT) zum Ausgangswert der aktuellen Stützpunktnummer PI[NBR\_ATMS].OUTV gefahren.

Die interne Zeitbearbeitung läuft auch bei durchgeschaltetem festen Sollwert (RMPSK\_ON = TRUE und DFRMP\_ON = TRUE) weiter.

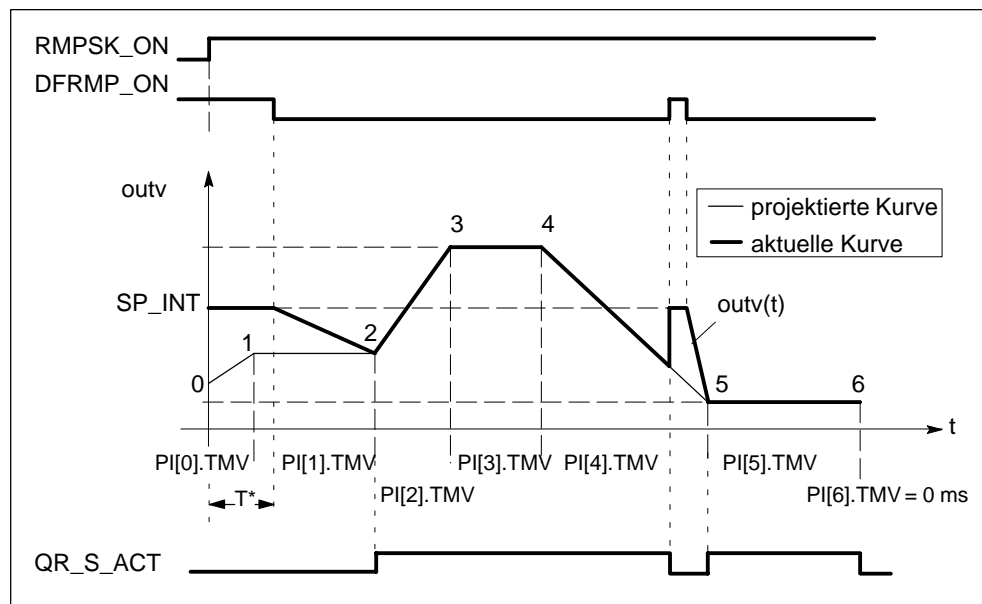


Bild 4-5 Beeinflussung der Fahrkurve durch das Vorbelegungssignal DFRMP\_ON

Beim Start der Fahrkurve RMPSK\_ON = TRUE wird so lange der feste Sollwert SP\_INT ausgegeben, bis DFRMP\_ON nach der Dauer von T\* von TRUE nach FALSE wechselt (Bild 4-5). Zu diesem Zeitpunkt ist die Zeit PI[0].TMV und ein Teil der Zeit PI[1].TMV abgelaufen. outv wird von SP\_INT nach PI[2].OUTV d. h. zum Stützpunkt 2 gefahren.

Erst ab dem Stützpunkt 2 wird die projektierte Fahrkurve ausgegeben, d. h. das Ausgangssignal QR\_S\_ACT erhält den Wert TRUE. Bei einem Wechsel des Vorbelegungssignals DFRMP\_ON von FALSE nach TRUE während der Fahrkurvenbearbeitung springt der ausgegebene Wert outv ohne Verzögerung auf SP\_INT bzw. den Ausgangswert von SP\_GEN.

### Zyklischer Betrieb eingeschaltet

Ist die Betriebsart 'Zyklische Wiederholung' (CYC\_ON = TRUE) eingeschaltet, dann kehrt der Zeitplangeber nach Ausgabe des letzten Stützpunktwertes automatisch zum Startpunkt zurück und beginnt einen neuen Durchlauf.

Zwischen dem letzten Stützpunkt und dem Startpunkt wird nicht interpoliert. Für stoßfreien Übergang muss gelten:  $PI[NBR\_PTS].OUTV = PI[0].OUTV$ .

### Zeitplangeber anhalten

Mit RMP\_HOLD = TRUE wird der Wert der Ausgangsgröße (einschließlich der Zeitbearbeitung) eingefroren. Bei Rücksetzen RMP\_HOLD = FALSE wird an der Unterbrechungsstelle  $PI[x].TMV$  fortgefahren.

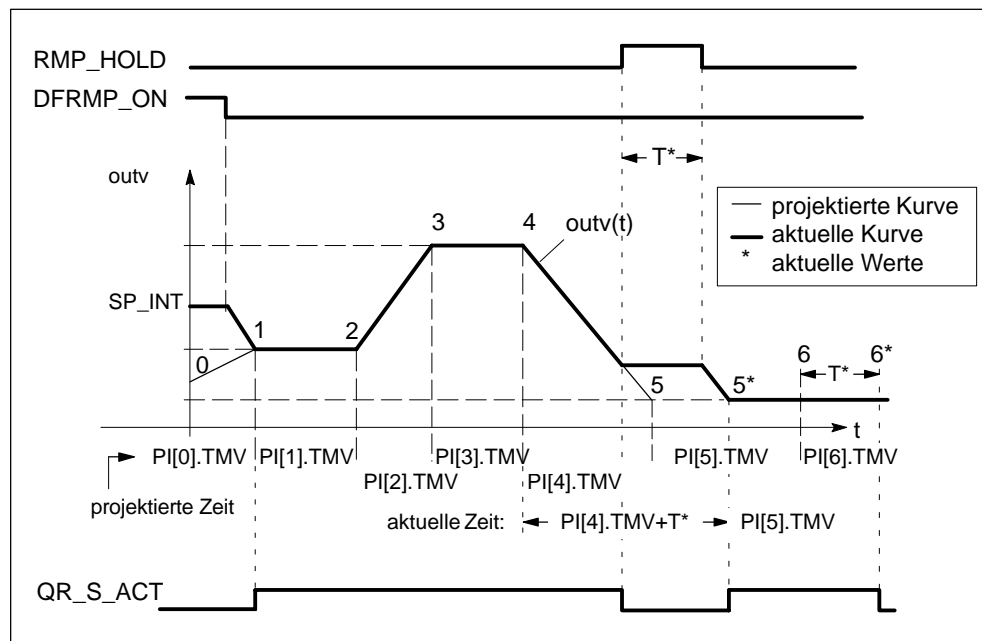


Bild 4-6 Beeinflussung der Fahrkurve durch das Haltesignal RMP\_HOLD

Die Bearbeitungszeit der Fahrkurve verlängert sich um die Haltezeit  $T^*$ . Die Fahrkurve hat vom Stützpunkt bis zum Signalwechsel von RMP\_HOLD (FALSE → TRUE) und vom Stützpunkt  $5^*$  bis zum Stützpunkt  $6^*$  den projektierten Verlauf, d. h. das Ausgangssignal QR\_S\_ACT hat den Wert TRUE (Bild 4-6).

Ist das Bit CONT\_ON gesetzt, fährt der angehaltene Zeitplangeber an der vorgegebenen Stelle  $TM\_CONT$  mit der Fahrkurvenbearbeitung fort.



### Bearbeitungsschritt und -zeit vorgeben

Ist der Steuereingang CONT\_ON für das Fortsetzen = TRUE gesetzt, dann wird an der Stelle TM\_CONT (Fortsetzzeitpunkt) mit dem Stützpunkt TM\_SNBR (Zielstützpunkt) weitergemacht. Der Zeit-Parameter TM\_CONT bestimmt die Restzeit, die der Zeitplangeber bis zum Zielstützpunkt TM\_SNBR benötigt.

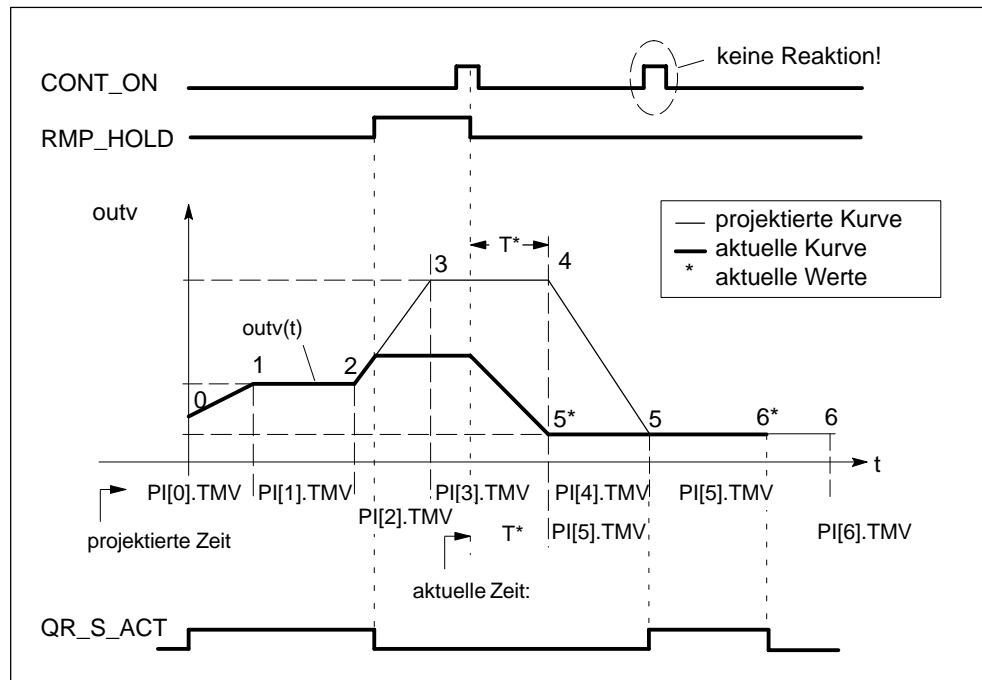


Bild 4-7 Beeinflussung der Fahrkurve durch das Haltesignal RMP\_HOLD und das Fortsetzungssignal CONT\_ON

Im Beispiel (Bild 4-7) gilt, dass bei  $RMP\_HOLD = TRUE$  sowie  $CONT\_ON = TRUE$  und Vorgabe von

Stützpunktnummer zum Fortsetzen  $TM\_SNBR = 5$

und Restzeit bis zum gewünschten Stützpunkt  $TM\_CONT = T^*$

für den Bearbeitungszyklus des Zeitplangebers die projektierten Stützpunkte 3 und 4 entfallen. Nach Signalwechsel  $RMP\_HOLD$  von TRUE nach FALSE wird der projektierte Kurvenverlauf erst ab Stützpunkt 5 wieder erreicht.

Der Ausgang  $QR\_S\_ACT$  wird nur dann gesetzt, wenn der Zeitplangeber die vom Anwender parametrisierte Fahrkurve abarbeitet.

### Gesamtzeit und Gesamtrestzeit aktualisieren

In jedem Zyklus werden die aktuelle Stützpunktnummer NBR\_ATMS, die aktuelle Restzeit bis zum Erreichen der Zeit RS\_TM, die Gesamtzeit T\_TM und die Gesamtrestzeit bis zum Erreichen des Fahrkurvenendes RT\_TM aktualisiert.

Bei Online-Änderungen von PI[n].TMV ändert sich die Gesamtzeit und die Gesamtrestzeit der Fahrkurve. Da die Berechnung von T\_TM und RT\_TM bei vielen Stützpunkten die Bearbeitungszeit des Funktionsbausteins stark vergrößert, wird sie nur nach Neustart oder bei TUPDT\_ON = TRUE durchgeführt. Die Zeitabschnitte PI[0 ... NBR\_PTS].TMV zwischen den einzelnen Stützpunkten werden aufsummiert und am Ausgang Gesamtzeit T\_TM und Gesamtrestzeit RT\_TM angezeigt.

Bitte beachten Sie, dass die Ermittlung der Gesamtzeiten relativ viel CPU-Laufzeit erfordert!

### Parameter der Funktion RMP\_SOAK

Der Ausgangsparameter outv ist ein impliziter Parameter und am Konfigurationswerkzeug über den Messpunkt MP1 zugänglich (siehe Bild 2-12).

Parameter	Bedeutung	Zulässiger Wertebereich
TM_SNB	Nummer des nächsten Stützpunktes	> 0 (dimensionslos)
TM_CONT	Weitermachzeitpunkt	gesamter Wertebereich
SP_INT	interner Sollwert	technischer Wertebereich

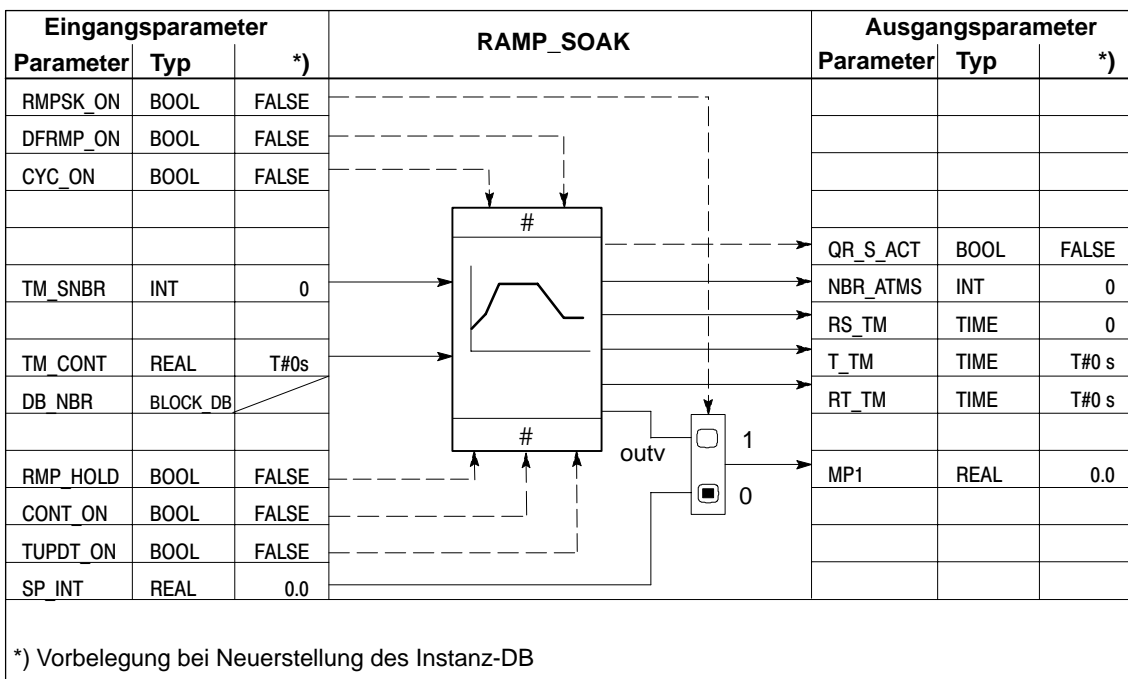


Bild 4-8 Funktionsschema und Parameter des Zeitplanners

Die Stützpunkt-Koordinaten sowie die Anzahl der Stützpunkte NBR\_PTS sind in einem globalen Datenbaustein abgelegt (Tabelle 4-2).

Tabelle 4-2 Globaler Datenbaustein (DB\_NBR), vorgelegt mit Startpunkt und vier Stützpunkten

Parameter	Datentyp	Kommentar	zul. Wertebereich	Vorbelegung
NBR_PTS	INT	Anzahl der Stützpunkte	1 bis 256	4
PI[0].OUTV	REAL	Ausgangswert [0]: Startpunkt	gesamter Wertebereich	0.0
PI[0].TMV	TIME	Zeitwert [0]: Startpunkt	gesamter Wertebereich	T#1 s
PI[1].OUTV	REAL	Ausgangswert [1]: Stützpunkt 1	gesamter Wertebereich	0.0
PI[1].TMV	TIME	Zeitwert [1]: Stützpunkt 1	gesamter Wertebereich	T#1 s
PI[2].OUTV	REAL	Ausgangswert [2]: Stützpunkt 2	gesamter Wertebereich	0.0
PI[2].TMV	TIME	Zeitwert [2]: Stützpunkt 2	gesamter Wertebereich	T#1 s
PI[3].OUTV	REAL	Ausgangswert [3]: Stützpunkt 3	gesamter Wertebereich	0.0
PI[3].TMV	TIME	Zeitwert [3]: Stützpunkt 3	gesamter Wertebereich	T#1 s
PI[4].OUTV	REAL	Ausgangswert [4]: Stützpunkt 4	gesamter Wertebereich	0.0
PI[4].TMV	TIME	Zeitwert [4]: Stützpunkt 4	gesamter Wertebereich	T#0 s

### 4.1.3 Normierung des externen Sollwertes (SP\_NORM)

#### Anwendung

Liegt der externe Sollwert nicht in der physikalischen Einheit des Istwerts vor (z. B. in % bei einer Reglerkaskade), dann müssen dieser Wert und sein Einstellbereich auf die physikalische Einheit des Istwertes normiert werden. Dies erfolgt über die Funktion "Normierung im Sollwertzweig".

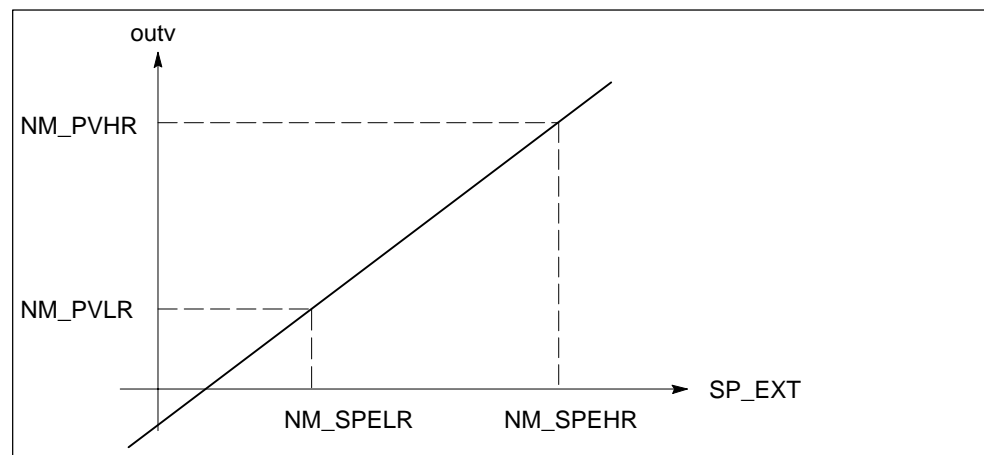
#### Die Funktion SP\_NORM

Die Funktion SP\_NORM normiert eine analoge Eingangsgröße. Der analoge externe Sollwert wird mit Hilfe der Normierungsgeraden in die Ausgangsgröße outv überführt. Der Ausgangswert OUTV ist am Konfigurationswerkzeug über den Messpunkt MP2 zugänglich (Bild 2-12).

Der Ausgangswert der Funktion ist wirksam, wenn der Steuereingang SPEXT\_ON = TRUE ist.

Zur eindeutigen Festlegung der Normierungsgeraden definieren Sie die folgenden vier Parameter:

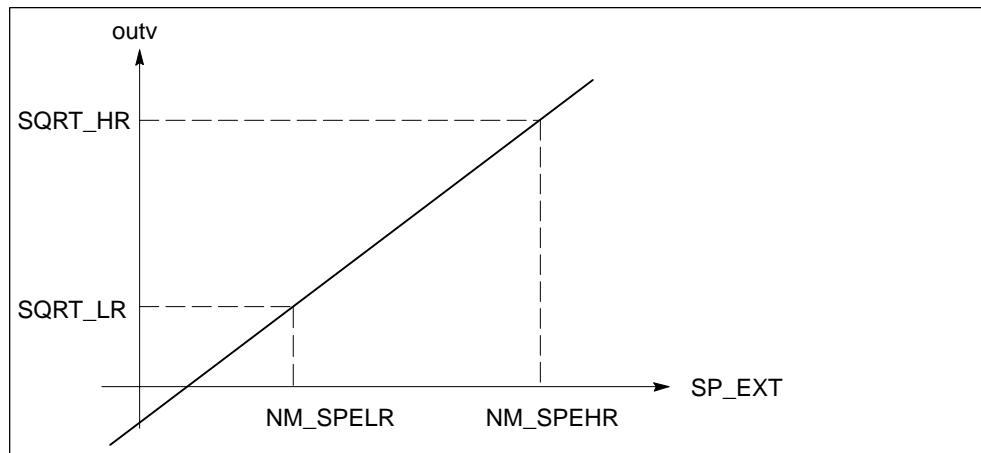
- die Obergrenze des Eingangswertes SP\_EXT: NM\_SPEHR
- die Untergrenze des Eingangswertes SP\_EXT: NM\_SPELR
- die Obergrenze des Ausgangswerts outv: NM\_PVHR  
(Diesen Wert geben Sie in der Normierfunktion des Istwerts vor.)
- die Untergrenze des Ausgangswerts outv: NM\_PVLR  
(Diesen Wert geben Sie in der Normierfunktion des Istwerts vor.)



Der Ausgangswert outv wird gemäß folgender Formel aus dem jeweiligen Eingangswert SP\_EXT berechnet:

$$\text{outv} = (\text{SP\_EXT} - \text{NM\_SPELR}) \times (\text{NM\_PVHR} - \text{NM\_PVLR}) / (\text{NM\_SPEHR} - \text{NM\_SPELR}) + \text{NM\_PVLR}$$

Für den Sonderfall einer eingeschalteten Wurzelfunktion im Istwertzweig gelten als Ober- und Untergrenze des Ausgangswerts outv die Normierungswerte der Wurzelfunktion (SQRT\_HR und SQRT\_LR).



Der Ausgangswert outv der Normierungsfunktion wird in diesem Fall gemäß folgender Formel aus dem jeweiligen Eingangswert SP\_EXT berechnet:

$$\text{outv} = (\text{SP\_EXT} - \text{NM\_SPELR}) \times (\text{SQRT\_HR} - \text{SQRT\_LR}) / (\text{NM\_SPEHR} - \text{NM\_SPELR}) + \text{SQRT\_LR}$$

Funktionsintern werden keine Werte begrenzt, und eine Prüfung der Parameter findet nicht statt. Falls Sie für NM\_SPEHR und NM\_SPELR denselben Wert eingeben, kann nach den obigen Formeln eine Division durch Null auftreten. Dies wird im Funktionsbaustein nicht abgefangen!

### Parameter der Funktion SP\_NORM

Der Ausgangsparameter outv ist ein impliziter Parameter und am Konfigurationswerkzeug nur über den Messpunkt MP2 zugänglich.

Parameter	Bedeutung	zul. Wertebereich
SP_EXT	externer Sollwert	technischer Wertebereich (physikalische Größe)
NM_SPEHR	Obergrenze des Eingangswerts SP_EXT	technischer Wertebereich (physikalische Einheit von SP_EXT)
NM_SPELR	Untergrenze des Eingangswerts SP_EXT	technischer Wertebereich (physikalische Einheit von SP_EXT)
NM_PVHR	Obergrenze des Ausgangswerts outv	technischer Wertebereich (physikalische Einheit des Istwerts bzw. dimensionslos, wenn Wurzelfunktion eingeschaltet ist)

Parameter	Bedeutung	zul. Wertebereich
NM_PVLR	Untergrenze des Ausgangswerts outv	technischer Wertebereich (physikalische Einheit des Istwerts bzw. dimensionslos, wenn Wurzelfunktion eingeschaltet ist)
SQRT_HR	Obergrenze des Ausgangswerts outv, falls im Istwertzweig eine Wurzelfunktion eingeschaltet ist	technischer Wertebereich
SQRT_LR	Untergrenze des Ausgangswerts outv, falls im Istwertzweig eine Wurzelfunktion eingeschaltet ist	technischer Wertebereich
SQRT_ON	Wurzelfunktion einschalten	TRUE, FALSE

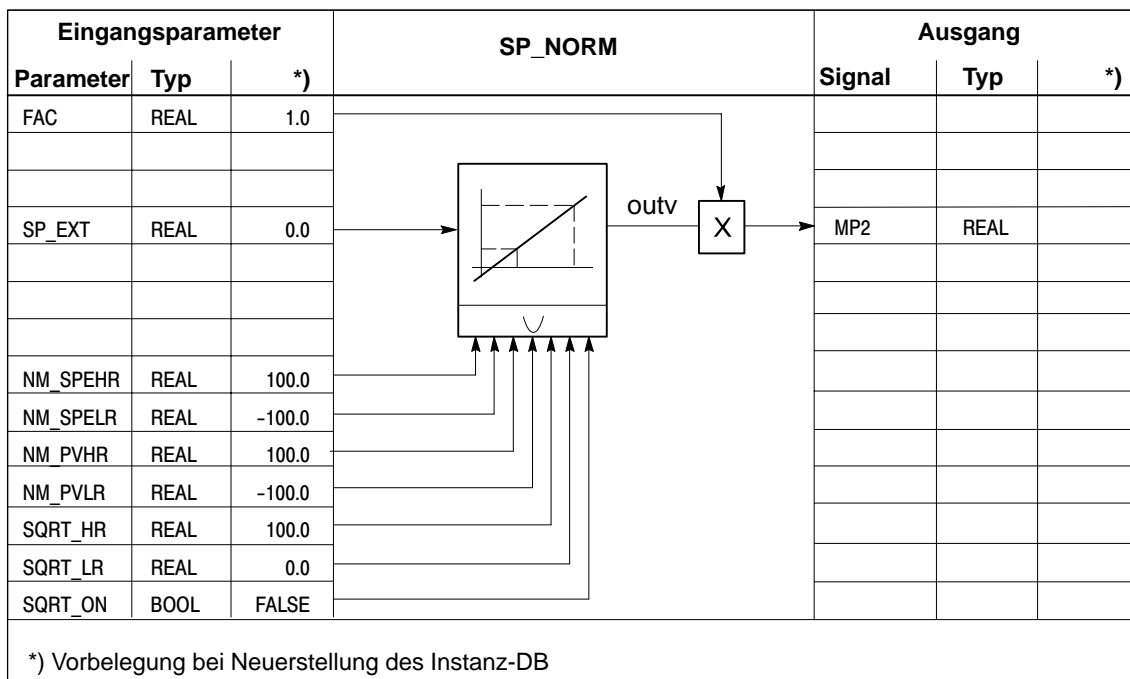


Bild 4-9 Funktionsschema und Parameter der Normierung der externen Führungsgröße

#### 4.1.4 FC-Aufruf im Sollwertweig (SPFC)

##### Anwendung

Durch Einfügen einer anwenderspezifischen Funktion in den Sollwertweig ist es möglich, eine extern vorgegebene Führungsgröße vor Aufschaltung auf den Regler einer Signalbehandlung (z. B. einer Signalverzögerung oder Linearisierung) zu unterziehen (Bild 2-12).

##### Die Funktion SPFC

Aktiviert man die Funktion SPFC durch `SPFC_ON = TRUE`, wird eine anwenderspezifische Funktion aufgerufen. Über den Parameter `SPFC_NBR` wird die Nummer der verwendeten FC eingegeben.

Der Regler führt einen Aufruf der-FC durch. Vorhandene Ein-/ Ausgangsparameter der Anwender-FC werden dabei nicht versorgt. Die Datenübertragung müssen Sie deshalb mit S7-AWL in der Anwender-FC programmieren. Nachstehend dafür ein Beispiel:

AWL	Erläuterung
<pre> FUNCTION "Anwender-FC" VAR_TEMP INV:REAL; OUTV:REAL; END_VAR BEGIN L      "Regler_DB".SPFC_IN T      #INV       //Anwenderfunktion OUTV=f(INV) L      #OUTV T      "Regler_DB".SPFC_OUT END_FUNCTION </pre>	

Der Wert von `SPFC_ON` bestimmt dann, ob in den Sollwertkanal an dieser Stelle eine frei programmierte Funktion in Form einer Standard-FC (z. B. eine Kennlinie) eingefügt wird oder ob der Sollwert ohne eine derartige Beeinflussung weiter verarbeitet wird.



##### Gefahr

Der Baustein überprüft nicht, ob eine FC vorhanden ist. Ist die FC nicht vorhanden, dann geht die CPU mit einem internen Systemfehler in STOP.

### Parameter der Funktion SPFC

Der Eingangswert SPFC\_IN ist ein impliziter Parameter. Dieser ist am Konfigurationswerkzeug entweder über den Messpunkt MP1 (Sollwert = SP\_INT) oder über den Messpunkt MP2 (Sollwert = SP\_EXT) zu beobachten. Der Ausgangswert ist am Messpunkt MP3 zugänglich.

Der Eingang SPFC\_IN ist dann auf den Sollwertzweig durchgeschaltet, wenn SPFC\_ON = FALSE ist (Vorbesetzung).

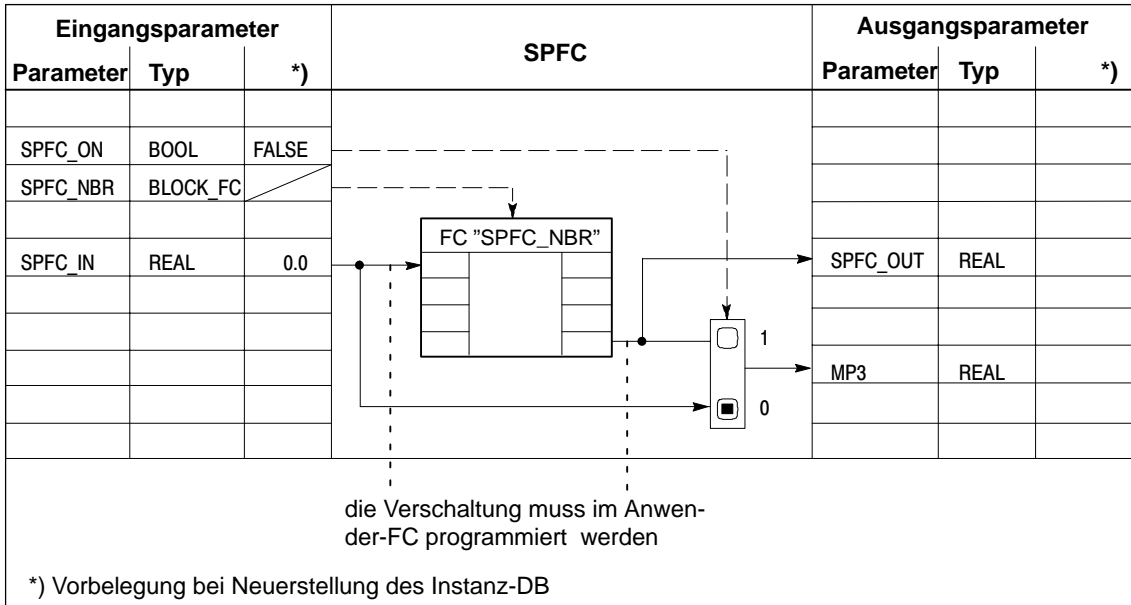


Bild 4-10 FC-Aufruf im Sollwertzweig



#### 4.1.5 Begrenzung der Änderungsgeschwindigkeit des Sollwertes (SP\_ROC)

##### Anwendung

Rampenfunktionen im Sollwertzweig werden dann verwendet, wenn der Prozess keine sprungförmigen Änderungen des Stellsignals verträgt, denn eine sprungförmige Sollwertänderung hat im Allgemeinen auch einen Stellwertsprung des Reglers zur Folge. Diese abrupten Stellwertänderungen sind z. B. dann zu vermeiden, wenn zwischen geregelter Motor und anzutreibender Last ein Getriebe zwischengeschaltet ist und ein zu schnelles Ansteigen der Motordrehzahl zu einer Überlastung des Getriebes führen würde.

##### Die Funktion SP\_ROC

Die Funktion SP\_ROC begrenzt die Änderungsgeschwindigkeit der im Regler verarbeiteten Sollwerte sowohl getrennt für den An- und Abstieg als auch getrennt für den positiven und den negativen Bereich der Führungsgröße.

An den vier Eingängen SPURLM\_P, SPDRLM\_P, SPURLM\_N und SPDRLM\_N werden die jeweiligen Begrenzungen für die Steigungen der Rampenfunktion im positiven und negativen Bereich der Führungsgröße eingegeben. Die Steigungen beziehen sich auf einen Anstieg bzw. Abfall pro Sekunde. Schnellere Sollwertänderungen werden auf diese Grenzgeschwindigkeiten verzögert.

Wird z. B. **SPURLM\_P** auf 10.0 [techn. Wertebereich/s] parametrisiert, so werden in jedem Abtastzyklus die folgenden Werte zum "Altwert" von outv addiert, und zwar solange  $inv > outv$  ist:

Abtastzeit 1 s	→	outv <sub>alt</sub> + 10
100 ms	→	outv <sub>alt</sub> + 1
10 ms	→	outv <sub>alt</sub> + 0.1

Die Funktionsweise der Signalbehandlung zeigt das folgende Bild anhand eines Beispiels. Aus Sprungfunktionen am Eingang inv(t) werden Rampenfunktionen am Ausgang outv(t).

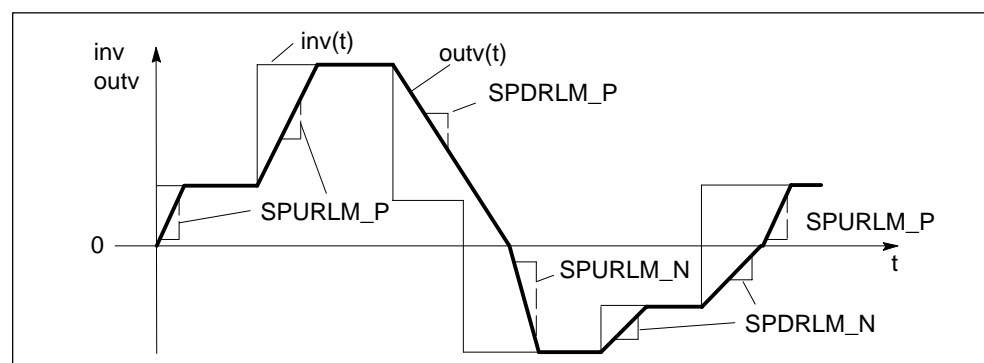


Bild 4-11 Begrenzung der Änderungsgeschwindigkeit der Führungsgröße SP(t)

Das Erreichen der Steigungsbegrenzungen wird nicht gemeldet!

### Parameter der Funktion SP\_ROC

Der Eingangswert *inv* ist ein impliziter Parameter und am Konfigurationswerkzeug nur über den Messpunkt MP3 zugänglich (Bild 2-12).

Der Ausgangswert *outv* ist am Konfigurationswerkzeug nicht zugänglich (siehe Bild 2-12).

Die Steigungswerte (pro Sekunde) werden immer positiv eingegeben.

Parameter	Rampe	Bedeutung	Zul. Wertebereich
SPURLM_P	OUTV > 0 und  OUTV  steigend	SP-Anstiegsbegrenzung im positiven Bereich	≥ 0 [techn. Wertebereich/s]
SPDRLM_P	OUTV > 0 und  OUTV  fallend	SP-Abstiegsbegrenzung im positiven Bereich	≥ 0 [techn. Wertebereich/s]
PURLM_N	OUTV < 0 und  OUTV  steigend	SP-Anstiegsbegrenzung im negativen Bereich	≥ 0 [techn. Wertebereich/s]
SPDRLM_N	OUTV < 0 und  OUTV  fallend	SP-Abstiegsbegrenzung im negativen Bereich	≥ 0 [techn. Wertebereich/s]

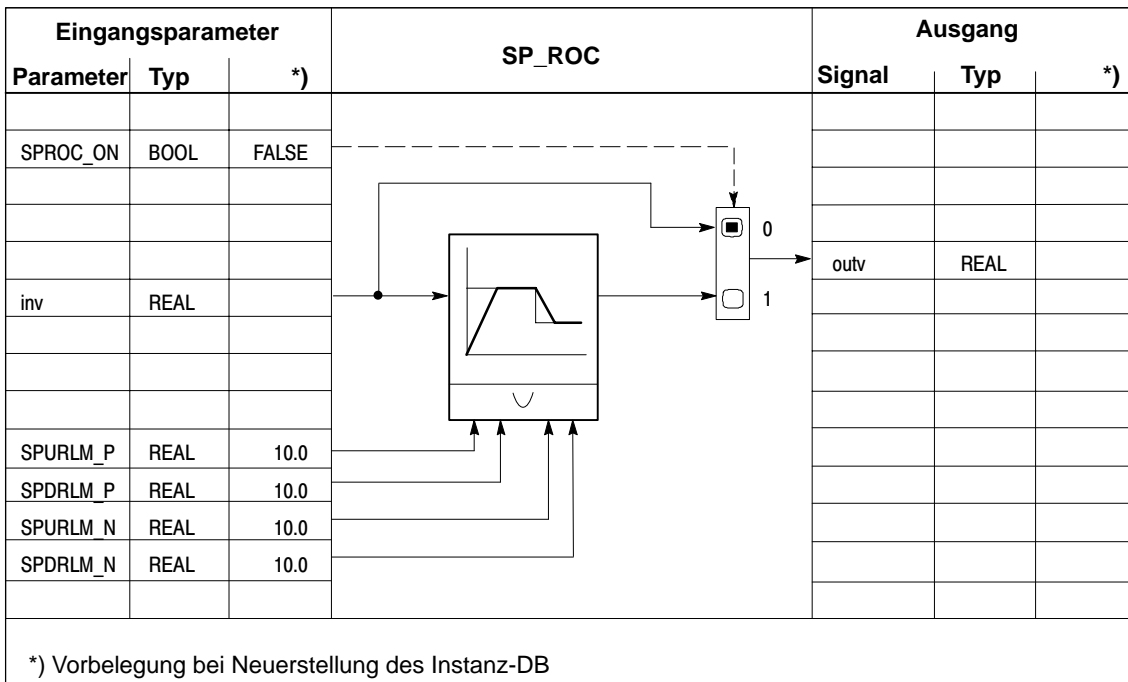


Bild 4-12 Funktionsschema und Parameter der Begrenzung der Sollwert-Änderungsgeschwindigkeit

#### 4.1.6 Begrenzung des Absolutwertes der Führungsgröße (SP\_LIMIT)

##### Anwendung

Der Einstellbereich der Führungsgröße bestimmt den Bereich, in dem die Regelgröße schwanken kann, d. h. den Bereich, in dem sich der Prozess im Rahmen zulässiger Zustandswerte bewegt.

Zur Vermeidung von kritischen oder unzulässigen Prozesszuständen wird deshalb im Sollwertzweig von Standard PID Control der Einstellbereich der Führungsgröße nach oben und unten begrenzt.

##### Die Funktion SP\_LIMIT

Die Funktion SP\_LIMIT begrenzt den Sollwert SP so lange auf den vorgebbaren unteren und oberen Wert SP\_LLM und SP\_HLM, wie die Eingangsgröße INV außerhalb dieser Begrenzungen liegt. Da die Funktion nicht abgeschaltet werden kann, muss bei der Konfiguration immer die Angabe einer Unter- und Obergrenze berücksichtigt werden.

Die Zahlenwerte der Grenzen werden an den Eingangsparametern für die untere und obere Begrenzung eingestellt. Bei Überschreitungen durch die Eingangsgröße  $inv(t)$  werden die zugehörigen Anzeigen über die Meldeausgänge (Bild 4-14) ausgegeben.

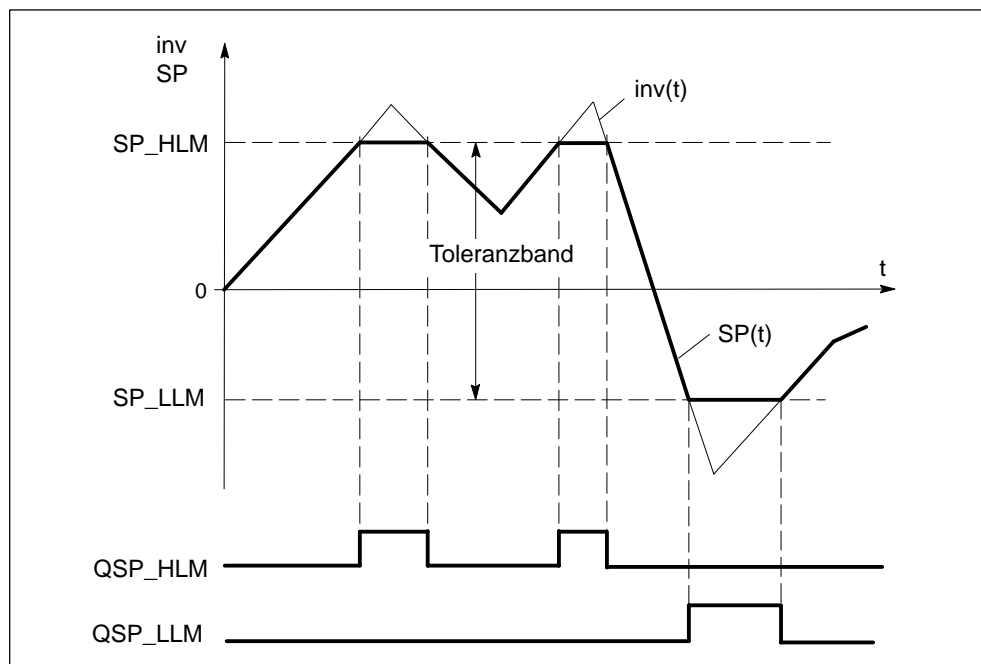


Bild 4-13 Absolutwertbegrenzung der Führungsgröße SP (t)

### Anlauf und Betriebsweise

- Bei Neustart werden alle Meldeausgänge nullgesetzt.
- Die Begrenzung arbeitet gemäß folgenden Beziehungen:

SP =	QSP_HLM =	QSP_LLM =	wenn:
SP_HLM	TRUE	FALSE	$inv \geq SP\_HLM$
SP_LLM	FALSE	TRUE	$inv \leq SP\_LLM$
INV	FALSE	FALSE	$SP\_HLM < inv < SP\_LLM$

Am Ausgang, d. h. am Parameter SP, wird der wirksame Sollwert von Standard PID Control angezeigt.

### Parameter der Funktion SP\_LIMIT

Der Eingangswert inv ist ein impliziter Parameter und am Konfigurationswerkzeug nur über den Messpunkt MP3 zugänglich.

Für ein sinnvolles Arbeiten der Begrenzungsfunktion muss gelten:

$$SP\_HLM > SP\_LLM$$

Parameter	Bedeutung	Zulässiger Wertebereich
SP_HLM	Obere Begrenzung der Führungsgröße	SP_LLM ... Obergrenze des techn. Wertebereichs
SP_LLM	Untere Begrenzung der Führungsgröße	Untergrenze des techn. Wertebereichs ... SP_HLM

Der Eingangsparameter inv ist ein impliziter Parameter und am Konfigurationswerkzeug nicht zugänglich (siehe Bild 2-12).

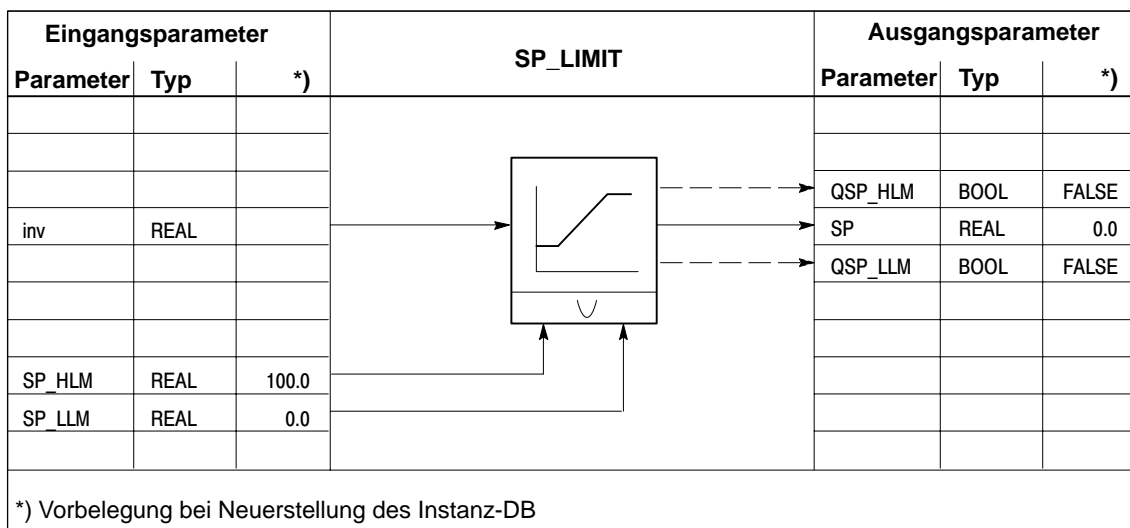


Bild 4-14 Funktionsschema und Parameter der Absolutwert-Begrenzung des Sollwertes

#### 4.1.7 Sollwertverstellung über das Konfigurationswerkzeug

##### SP-Anzeige und -Einstellung im Kreisbild

Das Konfigurationswerkzeug hat eine eigene Schnittstelle zum Regler-FB. Es ist deshalb jederzeit möglich, z. B. zu Testzwecken von einem PG/PC aus, auf dem das Konfigurationswerkzeug geladen ist, den Sollwertzweig zu unterbrechen und eigene Sollwerte SP\_OP vorzugeben (Bild 4-15).

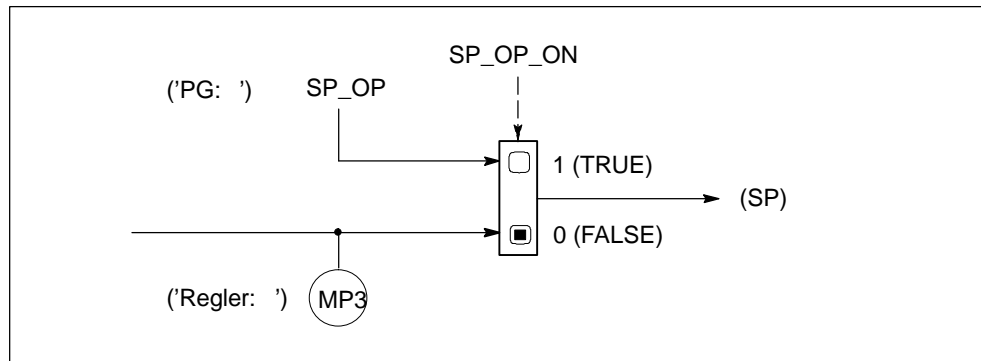


Bild 4-15 Eingriff in den Sollwertzweig durch das Konfigurationswerkzeug

Im Fenster des **Kreisbildes** steht dafür – bezeichnet mit **Sollwert** – eines der drei identischen Bedienfelder zur Verfügung. Hier wird unten ('Regler: ') der aktuelle am Messpunkt MP3 anliegende Sollwert ausgegeben. Im Feld darüber ('PG: ') wird der Parameter SP\_OP angezeigt und bedient.

##### Umschalten auf Sollwertvorgabe durch Konfigurationswerkzeug

Steht der Schalter im Konfigurationswerkzeug auf 'PG: ', dann wird im Regler-FB das Schaltsignal des Strukturschalters SPOP\_ON auf TRUE gesetzt und SP\_OP wird auf den Sollwert SP durchgeschaltet.

Ist im Sollwertzweig die Steigungsbegrenzung SP\_ROC eingeschaltet, kann stoßfrei zwischen den Schalterstellungen 'PG: ' und 'Regler: ' umgeschaltet werden. Der Wert, auf den zurückgeschaltet wird (MP3), kann im Anzeigefeld 'Regler: ' des **Kreisbildes** abgelesen werden. Mit der an SP\_ROC eingestellten Rampensteigung wird SP danach auf diesen Wert zu geführt.

Diese Bedieneingriffe wirken jeweils erst dann auf den Prozess, wenn sie durch Betätigen der Taste 'Senden' im **Kreisbild** zum Automatisierungsgerät übertragen werden.

## 4.2 Signalverarbeitung im Istwertzweig

### 4.2.1 Normierung der Istwerteingabe

#### Anwendung

Die Funktion "Normierung im Istwertzweig" dient zur Normierung des Eingangswertes PV\_PER bzw. PV\_IN auf die physikalische Einheit des Istwerts.

#### Die Funktion PV\_NORM

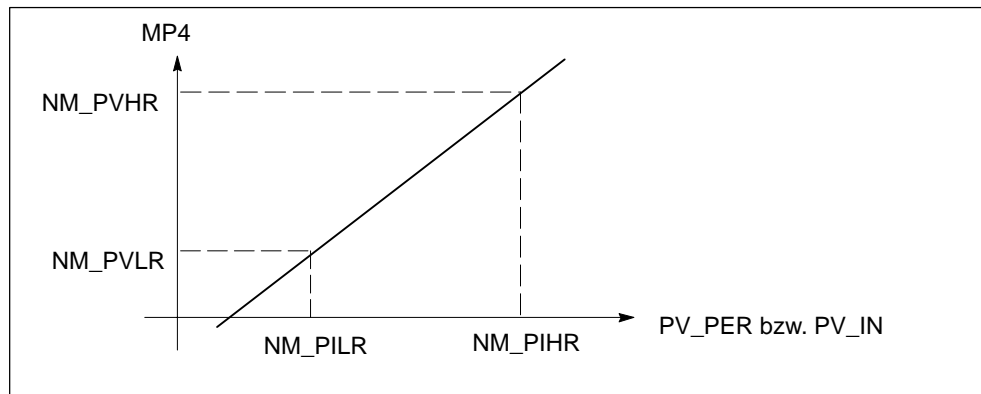
Die Funktion PV\_NORM normiert eine analoge Eingangsgröße. Über den Schalter PVPER\_ON legen Sie die zu normierende Eingangsgröße fest:

- PVPER\_ON = TRUE: Eingangsgröße ist der Istwert Peripherie PV\_PER
- PVPER\_ON = FALSE: Eingangsgröße ist der interne Istwert PV\_IN

Die Eingangsgröße wird mit Hilfe der Normierungsgeraden in die Ausgangsgröße MP4 überführt. Der Messpunkt MP4 ist am Konfigurationswerkzeug zugänglich (siehe Bild 2-13).

Zur eindeutigen Festlegung der Normierungsgeraden definieren Sie die folgenden vier Parameter:

- die Obergrenze des Eingangswertes PV\_PER bzw. PV\_IN: NM\_PIHR
- die Untergrenze des Eingangswertes PV\_PER bzw. PV\_IN: NM\_PILR
- die Obergrenze des Ausgangswerts MP4: NM\_PVHR
- die Untergrenze des Ausgangswerts MP4: NM\_PVLR



Der Ausgangswert MP4 wird nach folgender Formel aus dem jeweiligen Eingangswert PV\_PER bzw. PV\_IN berechnet:

$$MP4 = (PV\_PER - NM\_PILR) \times (NM\_PVHR - NM\_PVLR) / (NM\_PIHR - NM\_PILR) + NM\_PVLR$$

bzw.

$$MP4 = (PV\_IN - NM\_PILR) \times (NM\_PVHR - NM\_PVLR) / (NM\_PIHR - NM\_PILR) + NM\_PVLR$$

Funktionsintern werden keine Werte begrenzt, und eine Prüfung der Parameter findet nicht statt. Falls Sie für NM\_PIHR und NM\_PILR denselben Wert eingeben, kann nach den obigen Formeln eine Division durch Null auftreten. Dies wird im Funktionsbaustein nicht abgefangen!

### Normierung des Istwerts von Peripherie

Die Eingabe der Ober- und Untergrenze des Eingangswerts wird durch die Parametrieroberfläche unterstützt.

Für Spannungs-, Strom- und Widerstandsmessbereiche liegt die Nennbereichsobergrenze des Parameters PV\_PER (Peripherieeingang) immer auf dezimal 27648, die Nennbereichsuntergrenze auf 0 oder -27648.

Bei Temperaturenbaugruppen ist die Nennbereichsobergrenze variabel. Sie ist der jeweiligen Baugruppenbeschreibung zu entnehmen.

### Parameter der Funktionen CRP\_IN und PV\_NORM

Der Peripherieeingang PV\_PER ist dann auf den Istwertzweig durchgeschaltet, wenn PVPER\_ON = TRUE gesetzt ist. Am Messpunkt MP4 kann der normierte Peripherie-Istwert beobachtet werden (Bild 2-13).

Parameter	Bedeutung	Zulässiger Wertebereich
PV_PER	Istwert im Peripherieformat	
NM_PIHR	Obergrenze des Eingangswerts	technischer Wertebereich
NM_PILR	Untergrenze des Eingangswerts	technischer Wertebereich
NM_PVHR	Obergrenze des Ausgangswerts MP4	technischer Wertebereich (physikalische Einheit des Istwerts bzw. dimensionslos, falls Wurzelfunktion eingeschaltet)
NM_PVLR	Untergrenze des Ausgangswerts MP4	technischer Wertebereich (physikalische Einheit des Istwerts bzw. dimensionslos, falls Wurzelfunktion eingeschaltet)

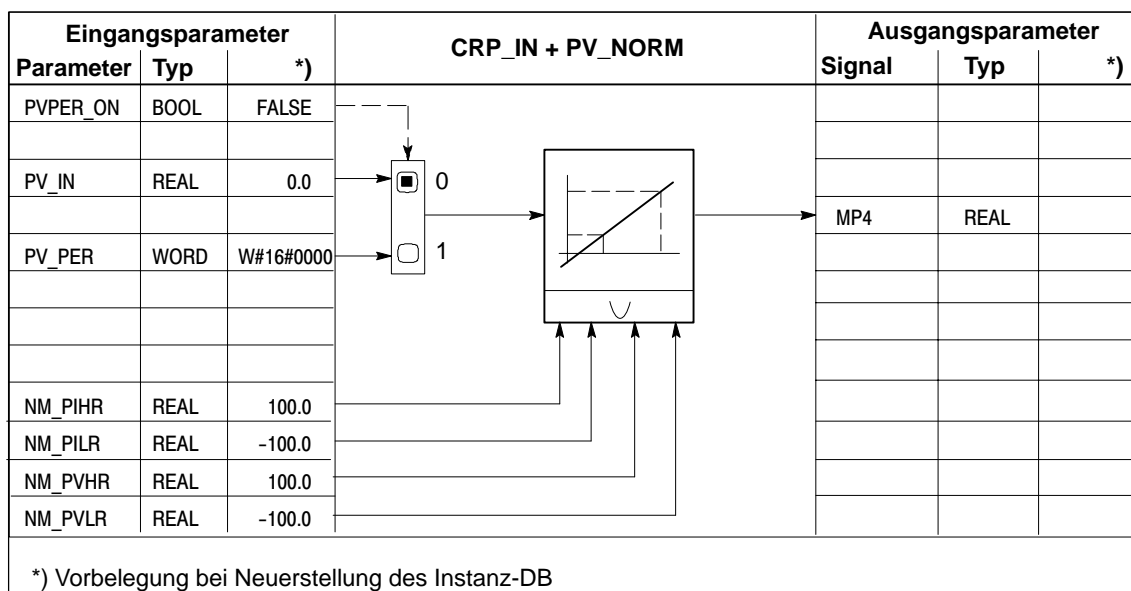


Bild 4-16 Funktionsschema und Parameter für die Normierung der physikalischen Regelgröße

## 4.2.2 Glättung der Regelgröße (LAG1ST)

### Anwendung

Die Funktion LAG1ST wird als Glättungs- und Verzögerungsglied für die Regelgröße eingesetzt. Störsignale können damit unterdrückt werden.

### Die Funktion LAG1ST

Durch zeitliche Verzögerung werden höherfrequente Schwankungen des Istwertsignals gedämpft und dadurch von der Verarbeitung im Regelalgorithmus - insbesondere von Auswirkungen auf die Bildung des Differenziereinflusses - ferngehalten. Das Maß der Signaldämpfung wird durch die einstellbare Zeitkonstante PV\_TMLAG bestimmt.

Die Dämpfungswirkung wird durch einen Verzögerungsalgorithmus 1. Ordnung erzielt.

Die Übertragungsfunktion im Laplace-Bereich ist:

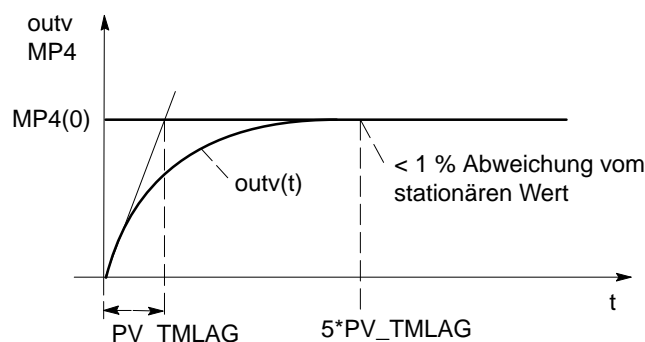
$$\frac{\text{outv}(s)}{\text{MP4}(s)} = \frac{1}{(1 + \text{PV\_TMLAG} * s)} \quad \text{mit } s = \text{Laplace-Operator}$$

Die Sprungantwort im Zeit-Bereich ist:

$$\text{outv}(t) = \text{MP4}(0) (1 - e^{-t/\text{PV\_TMLAG}})$$

Darin bedeutet:

MP4(0)	die Höhe des Istwertsprungs am Eingang
outv(t)	die Ausgangsgröße
PV_TMLAG	die Verzögerungszeitkonstante
t	Zeit





### Parametrier-Bedingungen

Für  $PV\_TMLAG \leq 0,5 * CYCLE$  ist keine Verzögerung mehr wirksam.

Eine Abtastzeit (CYCLE) von weniger als einem Fünftel der Verzögerungszeit ist erforderlich, um ein dem analogen Verhalten annähernd entsprechendes Verzögerungsverhalten zu erzielen.

### Parameter der Funktion LAG1ST

Der Ausgangswert outv ist ein impliziter Parameter und am Konfigurationswerkzeug nur über den Messpunkt MP5 zu beobachten (Bild 2-13).

Wenn  $LAG1STON = FALSE$ , dann ist der Peripherieeingang  $PV\_PER$  bzw. der interne Eingang  $PV\_IN$  verzögerungsfrei auf den Istwertzweig durchgeschaltet (Vorbesetzung).

Parameter	Bedeutung	Zulässiger Wertebereich
PV_TMLAG	Istwert-Verzögerungszeit	gesamter Wertebereich

Eingangsparameter			LAG1ST	Ausgangsparameter		
Parameter	Typ	*)		Signal	Typ	*)
LAG1STON	BOOL	FALSE		MP5	REAL	0.0
MP4	REAL	0.0				
PV_TMLAG	TIME	T#5s				
*) Vorbelegung bei Neuerstellung des Instanz-DB						

Bild 4-17 Glättung der Regelgröße

### 4.2.3 Bildung der Quadratwurzel (SQRT)

#### Anwendung

Liegt der von einem Geber gelieferte Istwert als physikalische Größe vor, die in einem quadratischem Zusammenhang mit der gemessenen Prozessgröße steht, dann muss der Verlauf der Regelgröße vor der Weiterverarbeitung im Regler linearisiert werden. Diese Aufgabe erfüllt die Funktion SQRT im Istwertzweig von Standard PID Control.

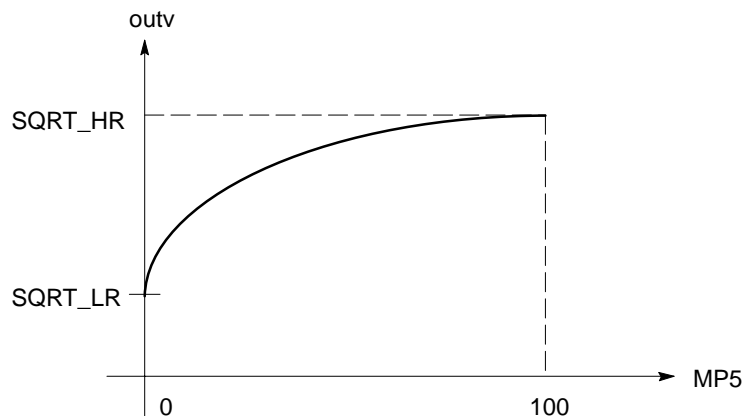
Das Messsignal muss immer dann durch Radizieren linearisiert werden, wenn Durchflussmessungen mit Hilfe von Messblenden oder Venturirohren usw. vorgenommen werden. Der gemessene Differenzdruck (Wirkdruck) ist dann proportional zum Quadrat des Durchflusses.

Über das Eingangssignal SQRT\_ON = TRUE wird die Wurzelfunktion im Istwertzweig aktiviert. Der Algorithmus der Wurzelfunktion hat folgende Form:

$$\text{outv} = \text{SQRT}(\text{MP5}) \times (\text{SQRT\_HR} - \text{SQRT\_LR}) / 10.0 + \text{SQRT\_LR}$$

Diese Formel sieht vor, dass der Eingangswert der Wurzel auf einen Zahlenbereich von 0 bis 100 normiert ist. Die Parameter NM\_PVHR und NM\_PVLR der Normierung im Istwertzweig müssen daher auf 100.0 und 0.0 parametrieren werden.

Die Wurzel aus diesem Wert ergibt einen Zahlenbereich von 0 bis 10. Über die Normierungswerte SQRT\_HR und SQRT\_LR wird dieser Zahlenbereich normiert auf den physikalischen Messbereich (SQRT\_LR bis SQRT\_HR).



#### Beispiel zur Normierung

Der Eingangswert PV\_IN des Reglers sei der Differenzdruck in mbar:

Messbereichs-Anfang NM_PILR	Messbereichs-Ende NM_PIHR	Wertebeispiel für PV_IN
20.0 mbar	200.0 mbar	150.0 mbar

Mit Hilfe der Normierfunktion PV\_NORM erfolgt die Berechnung des normierten Differenzdrucks, wobei NM\_PVHR = 100.0 und NM\_PVLR = 0.0 gewählt wird:

$$\begin{aligned}
 MP4 &= (PV\_IN - NM\_PILR) * (NM\_PVHR - NM\_PVLR) / \\
 &\quad (NM\_PIHR - NM\_PILR) + NM\_PVLR \\
 &= (PV\_IN - 20.0 \text{ mbar}) * (100.0 - 0.0) / \\
 &\quad (200.0 \text{ mbar} - 20.0 \text{ mbar}) + 0.0 \\
 &= (PV\_IN - 20.0 \text{ mbar}) * 100 / 180.0 \text{ mbar}
 \end{aligned}$$

Anfangswert von MP4	Endwert von MP4	Wertebeispiel für MP4
0.0	100.0	72.222

In diesem Beispiel wird keine Glättung verwendet und daher gilt: MP5 = MP4.

Für die Wurzel aus dem normierten Differenzdruck MP5 ergibt sich:

Anfangswert nach der Wurzel	Endwert nach der Wurzel	Wertebeispiel für SQRT(MP5)
0.0	10.0	8.498

Für den normierten Ausgangswert outv der Wurzelfunktion (physikalischer Durchfluss) folgt mit SQRT\_HR = 20000.0 m<sup>3</sup>/h und SQRT\_LR = 0.0 m<sup>3</sup>/h:

$$\begin{aligned}
 outv &= SQRT(MP5) * (SQRT\_HR - SQRT\_LR) / 10.0 + SQRT\_LR \\
 &= SQRT(MP5) * (20000.0 \text{ m}^3/\text{h} - 0.0 \text{ m}^3/\text{h}) / 10.0 + 0.0 \text{ m}^3/\text{h} \\
 &= 2000.0 \text{ m}^3/\text{h} * SQRT(MP5)
 \end{aligned}$$

Messbereichs-Anfang outv	Messbereichs-Ende outv	Wertebeispiel für outv
0.0 m <sup>3</sup> /h	20000.0 m <sup>3</sup> /h	16996.732 m <sup>3</sup> /h

Der Ausgangsparameter outv ist ein impliziter Parameter und am Konfigurationswerkzeug nicht zugänglich (siehe Bild 2-13).

Eingangsparameter			SQRT	Ausgang			
Parameter	Typ	*)		Signal	Typ	*)	
SQRT_ON	BOOL	FALSE					
(MP5)	REAL	0.0			(PVFC_IN)	REAL	0.0
SQRT_HR	REAL	100.0					
SQRT_LR	REAL	0.0					
*) Vorbelegung bei Neuerstellung des Instanz-DB							

Bild 4-18 Funktionsschema und Parameter für die Radizierung des Istwertsignals

#### 4.2.4 FC-Aufruf im Istwertzweig (PVFC)

##### Anwendung

Durch Einfügen einer anwenderspezifischen Funktion in den Istwertzweig kann die Regelgröße vor der Weiterverarbeitung im Regler einer Signalbehandlung, z. B. einer Signalverzögerung oder Linearisierung, unterzogen werden (Bild 2-13).

##### Die Funktion PVFC

Bei Aktivierung der Funktion PVFC durch PVFC\_ON = TRUE wird eine anwenderspezifische Funktion aufgerufen. Über den Parameter PVFC\_NBR wird die Nummer der verwendeten FC eingegeben.

Der Regler führt einen Aufruf der FC durch. Vorhandene Ein-/ Ausgangsparameter der Anwender-FC werden dabei nicht versorgt. Die Datenübertragung müssen Sie deshalb mit S7-AWL in der Anwender-FC programmieren. Nachstehend dafür ein Beispiel:

AWL	Erläuterung
<pre> FUNCTION "Anwender-FC" VAR_TEMP INV:REAL; OUTV:REAL; END_VAR BEGIN L      "Regler_DB".PVFC_IN T      #INV       //Anwenderfunktion OUTV=f(INV) L      #OUTV T      "Regler_DB".PVFC_OUT END_FUNCTION                     </pre>	

Der Wert von PVFC\_ON bestimmt dann, ob in den Istwertzweig an dieser Stelle eine frei programmierte Funktion in Form einer Standard-FC (z. B. eine Kennlinie) eingefügt wird oder ob der Istwert ohne eine derartige Beeinflussung weiterverarbeitet wird.



##### Gefahr

Der Baustein überprüft nicht, ob eine FC vorhanden ist. Ist die FC nicht vorhanden, dann geht die CPU mit einem internen Systemfehler in STOP.

### Parameter der Funktion PVFC

Der Eingangswert PVFC\_IN ist ein impliziter Parameter und am Konfigurationswerkzeug nicht zu beobachten. Der Ausgangswert ist am Messpunkt MP6 zugänglich (Bild 2-13).

Wenn PVFC\_ON = FALSE ist (Vorbereitung), dann ist der Eingang PVFC\_IN auf den Istwertzweig durchgeschaltet.

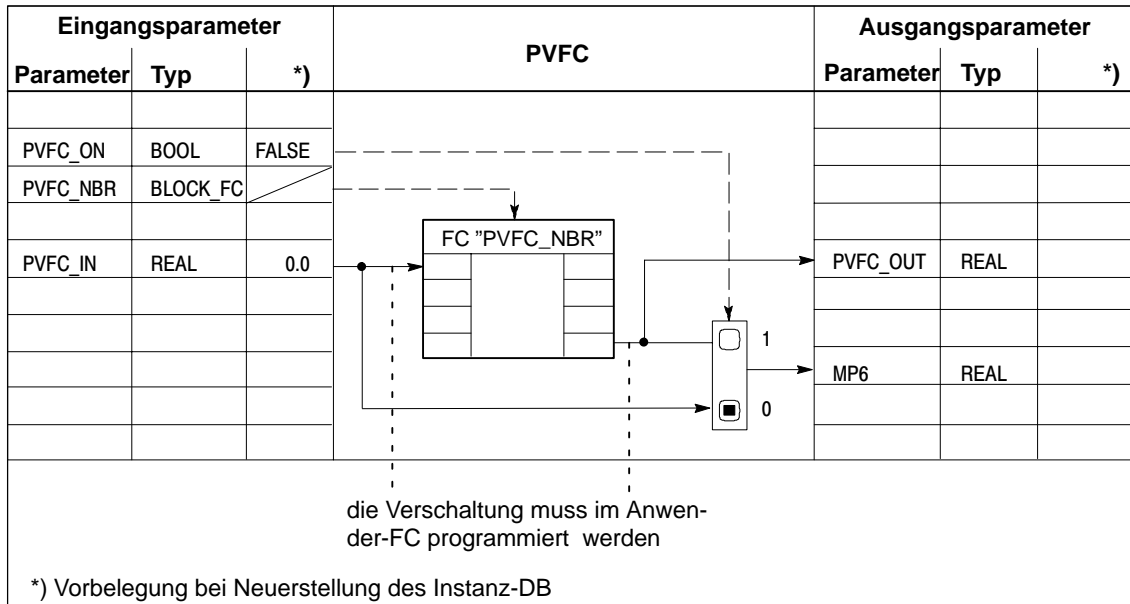


Bild 4-19 FC-Aufruf im Istwertzweig

## 4.2.5 Überwachung des Istwertes auf Grenzwerte (PV\_ALARM)

### Anwendung

Werden bestimmte Werte der Prozessgrößen (z. B. Drehzahl, Druck, Füllstand, Temperatur, ...) über- oder unterschritten, können bei Regelungen unzulässige Prozess- oder Anlagenzustände auftreten. In diesen Fällen wird die Funktion PV\_ALARM eingesetzt, um den Istwert auf Über-/Unterschreitung des zulässigen Betriebsbereiches zu überwachen. Die Grenzüberschreitungen werden erfasst und gemeldet, damit eine sinnvolle Reaktion ausgelöst werden kann.

### Die Funktion PV\_ALARM

Die Funktion PV\_ALARM überwacht die Regelgröße PV(t) auf vier vorgebbare Grenzen in zwei Toleranzbändern. Werden die Grenzen erreicht oder überschritten, meldet die Funktion an der ersten Grenze "Warnung" und an der zweiten Grenze "Alarm".

Die Zahlenwerte der Grenzen werden an den Eingangsparametern für "Warnung" und "Alarm" eingestellt (Bild 4-20). Überschreitet bzw. unterschreitet der Istwert (PV) diese Grenzen, dann werden die zugehörigen Ausgangsmeldebite QPVH\_ALM, QPVH\_WRN, QPVL\_WRN und QPVL\_ALM gesetzt (Bild 4-21).

Um bei geringfügigen Änderungen der Eingangsgröße oder bei Rundungsfehlern ein "Flackern" der Meldebite zu verhindern, wird eine Hysterese (Rückschaltdifferenz) PV\_HYS eingestellt. Die Regelgröße muss die Hysterese überwinden, bevor die Meldungen rückgesetzt werden.

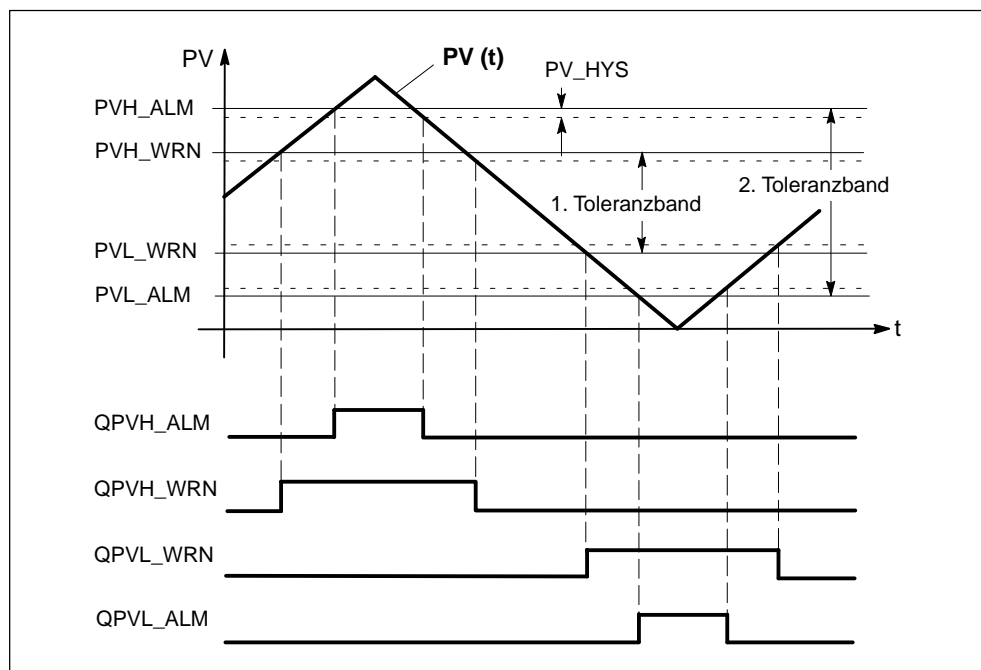


Bild 4-20 Überwachung der Regelgröße PV auf Grenzwertüber-/unterschreitung

### Anlauf und Betriebsweise

- Bei Neustart werden alle Meldeausgänge nullgesetzt.
- Die Grenzwertmeldung arbeitet entsprechend den folgenden Funktionen:

QPVH_ALM	QPVH_WRN	QPVL_WRN	QPVL_ALM	wenn:	und:
TRUE	TRUE	FALSE	FALSE	PV ↗ PV ↘	$PV \geq PVH\_ALM$ $PV \geq PVH\_ALM - PV\_HYS$
FALSE	TRUE	FALSE	FALSE	PV ↗ PV ↘	$PV \geq PVH\_WRN$ $PV \geq PVH\_WRN - PV\_HYS$
FALSE	FALSE	TRUE	FALSE	PV ↘ PV ↗	$PV \leq PVL\_WRN$ $PV \leq PVL\_WRN + PV\_HYS$
FALSE	FALSE	TRUE	TRUE	PV ↘ PV ↗	$PV \leq PVL\_ALM$ $PV \leq PVL\_ALM + PV\_HYS$

Für ein sinnvolles Arbeiten des Bausteins muss gelten:

$$PVL\_ALM < PVL\_WRN < PVH\_WRN < PVH\_ALM$$

### Parameter der Funktion PV\_ALARM

Die Funktion PV\_ALARM kann nicht abgeschaltet werden. Bei der Konfiguration von Standard PID Control ist deshalb besonders auf die Grenzwerteinstellung zu achten. Andernfalls werden Grenzüberschreitungen gemäß den voreingestellten Parametern (Bild 4-21) gemeldet.

Parameter	Bedeutung	Zulässiger Wertebereich
PVH_ALM	Oberer PV-Grenzwert 'Alarm'	Techn. Wertebereich
PVH_WRN	Oberer PV-Grenzwert 'Warnung'	Techn. Wertebereich
PVL_ALM	Unterer PV-Grenzwert 'Alarm'	Techn. Wertebereich
PVL_WRN	Unterer PV-Grenzwert 'Warnung'	Techn. Wertebereich
PV_HYS	PV-Rückschaltdifferenz (Hysterese)	$\geq 0$ [%]

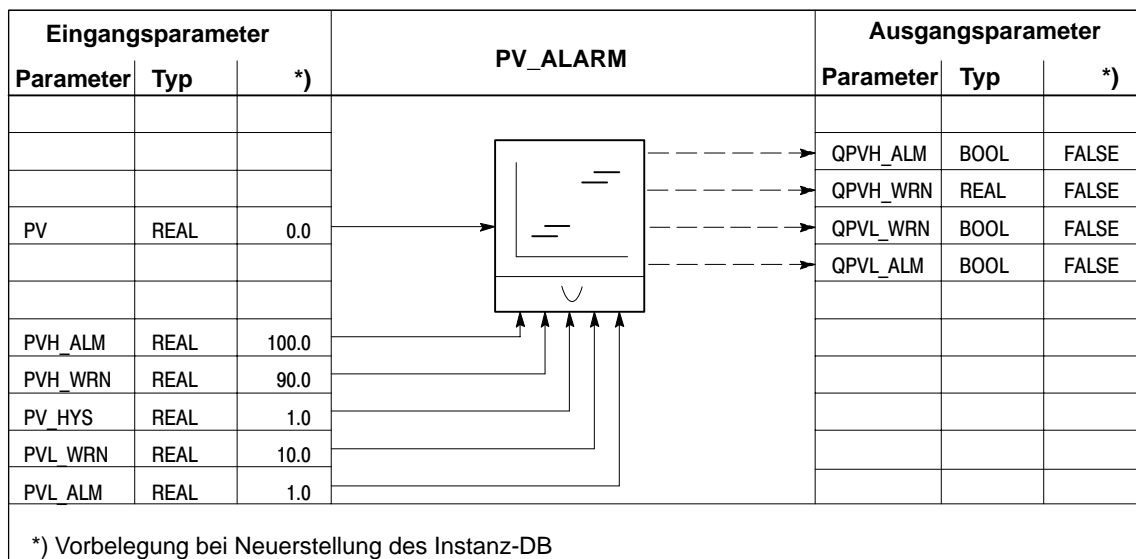


Bild 4-21 Funktionsschema und Parameter der Grenzwertüberwachung des Istwertes PV

## 4.2.6 Überwachung der Änderungsgeschwindigkeit des Istwertes (ROCALARM)

### Anwendung

Wird die Änderung der Prozessgröße (z. B. Drehzahl, Druck, Füllstand, Temperatur, ...) zu groß, können bei Regelungen unzulässige Prozess- oder Anlagenzustände auftreten. In diesen Fällen wird die Funktion ROCALARM eingesetzt, um den Istwert auf Über-/Unterschreiten einer zulässigen Änderungsgeschwindigkeit bzw. Steigung zu überwachen. Die Grenzüberschreitungen werden erfasst und gemeldet, damit eine sinnvolle Reaktion ausgelöst werden kann.

### Die Funktion ROCALARM

Die Funktion ROCALARM überwacht die Regelgröße  $PV(t)$  auf je nach Vorzeichen unterschiedlich vorgebbare Grenzen für die zulässige Änderungsgeschwindigkeit.

Die Zahlenwerte der Grenzsteigungen werden an den Eingangsparametern für "Steigen" und "Fallen" im positiven und negativen Bereich der Regelgröße eingestellt. Die Steigungen beziehen sich auf einen Anstieg bzw. Abfall in Prozent pro Sekunde.

Überschreitet die Änderungsgeschwindigkeit der Regelgröße diese Grenzen, dann werden die zugehörigen Ausgangsmeldebite  $QPVURLMP \dots QPVDRLMN$  gesetzt (Bild 4-22 und 4-23).

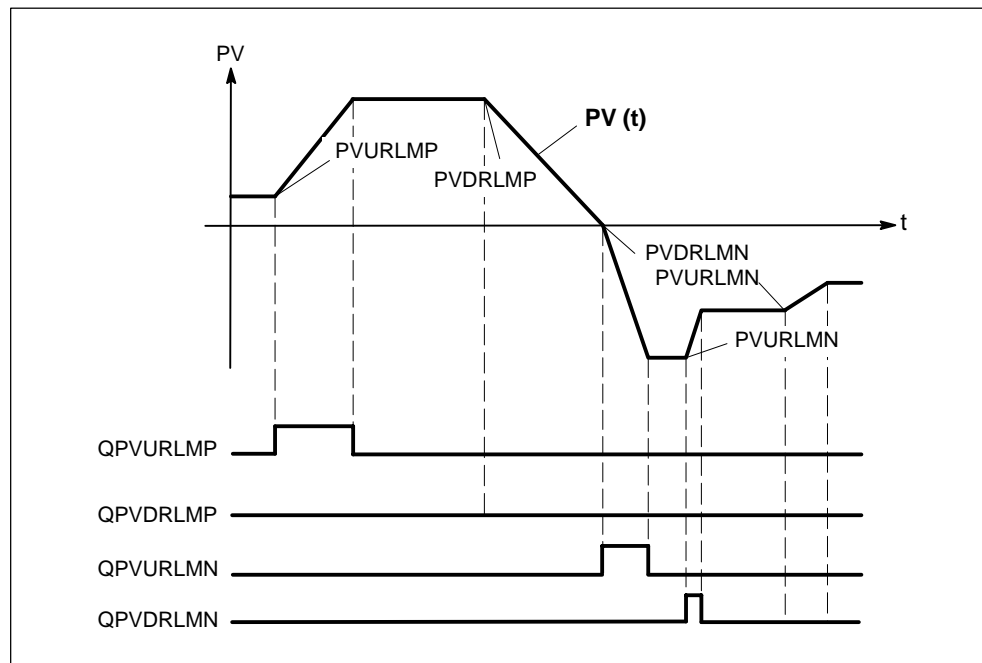


Bild 4-22 Überwachung der Änderungsgeschwindigkeit (Steigung) der Regelgröße  $PV(t)$  auf Grenzwertüber-/unterschreitung



Die Kennzeichnung der Rampen-Parameter erfolgt nach folgendem Schema:

Parameter	PV-Änderung
PVURLM_P	PV > 0 und  PV  steigend
PVDRLM_P	PV > 0 und  PV  fallend
PVURLM_N	PV < 0 und  PV  steigend
PVDRLM_N	PV < 0 und  PV  fallend

### Parameter der Funktion ROCALARM

Die Funktion ROCALARM kann nicht abgeschaltet werden. Bei der Konfiguration von Standard PID Control ist deshalb besonders auf die Grenzwerteinstellung zu achten. Andernfalls werden Grenzüberschreitungen gemäß den voreingestellten Parametern (Bild 4-23) gemeldet.

Parameter	Bedeutung	Zul. Wertebereich
PVURLM_P	PV-Anstiegsbegrenz. im pos. Bereich	$\geq 0$ [s]
PVDRLM_P	PV-Abstiegsbegrenz. im pos. Bereich	$\geq 0$ [s]
PVURLM_N	PV-Anstiegsbegrenz. im neg. Bereich	$\geq 0$ [s]
PVDRLM_N	PV-Abstiegsbegrenz. im neg. Bereich	$\geq 0$ [s]

Die Steigungswerte werden immer positiv eingegeben.

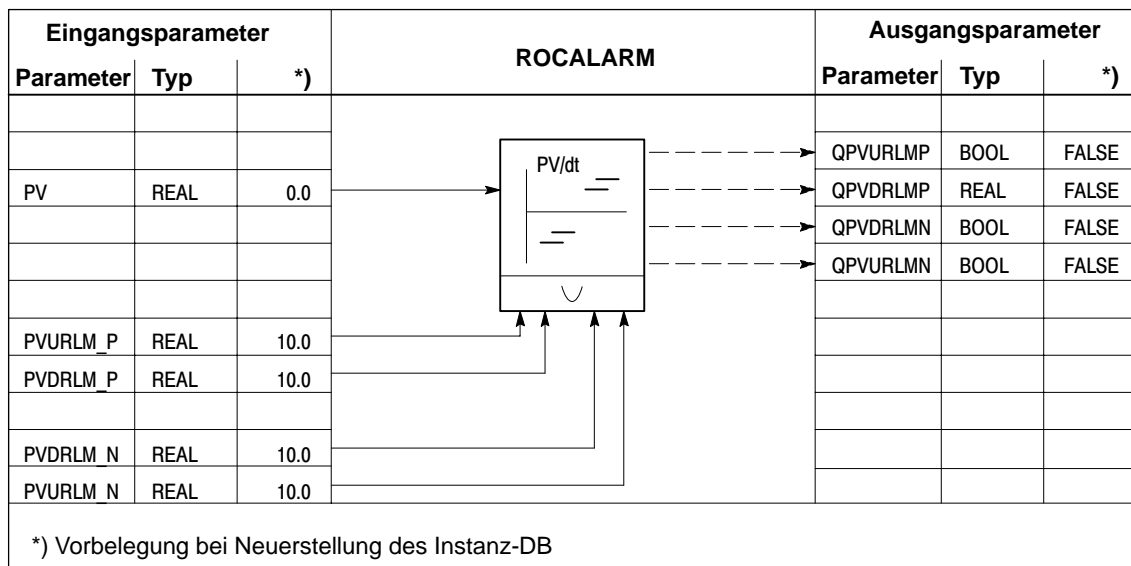


Bild 4-23 Funktionsschema und Parameter der Überwachung der Regelgröße PV(t) auf Geschwindigkeitsüberschreitung

## 4.2.7 Istwertverstellung über das Konfigurationswerkzeug

### PV-Anzeige und -Einstellung im Loop Monitor

Das Konfigurationswerkzeug hat eine eigene Schnittstelle zum Regler-FB. Es ist deshalb jederzeit möglich, z. B. zu Testzwecken von einem PG/PC aus, auf dem das Konfigurationswerkzeug geladen ist, den Regelgrößenzweig zu unterbrechen und eigene Istwerte PV\_OP vorzugeben (Bild 4-24).

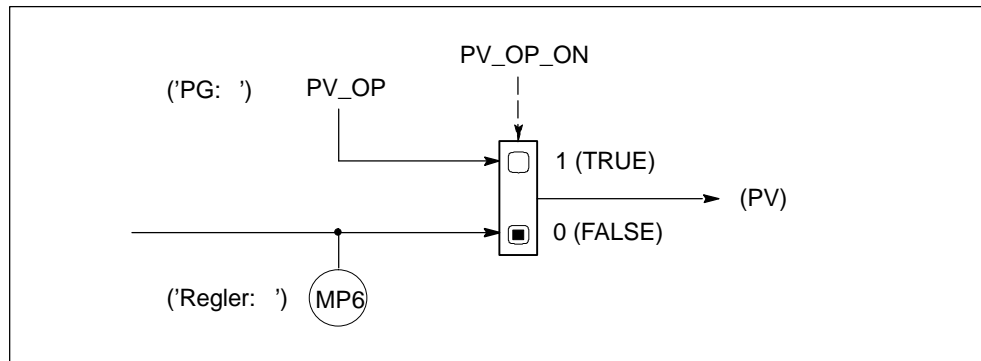


Bild 4-24 Eingriff in den Sollwertzweig von einer Bedieneinrichtung (OP) aus

Im Fenster des **Kreisbildes** steht dafür – bezeichnet mit **Istwert** – eines der drei identischen Bedienfelder zur Verfügung. Hier wird unten ('Regler: ') der aktuelle am Messpunkt MP6 anliegende Istwert ausgegeben. Im Feld darüber ('PG: ') wird der Parameter PV\_OP angezeigt und bedient.

### Umschalten auf Istwertvorgabe durch Konfigurationswerkzeug

Steht der Schalter im Konfigurationswerkzeug auf 'PG: ', dann wird im Regler-FB das Schaltsignal des Strukturschalters PVOP\_ON auf TRUE gesetzt und PV\_OP wird auf den Istwert PV durchgeschaltet.

Der Wert, auf den zurückgeschaltet wird (MP6), kann im Anzeigefeld 'Regler: ' des **Kreisbildes** abgelesen werden.

Diese Bedieneingriffe wirken jeweils erst dann auf den Prozess, wenn sie durch Betätigen der Taste 'Senden' im **Kreisbild** auf das Automatisierungsgerät übertragen werden.

## 4.3 Signalverarbeitung der Regeldifferenz (Regelabweichung)

### 4.3.1 Signalfilterung durch Totzonenfunktion (DEADBAND)

#### Anwendung

Bei Überlagerung der Regel- oder der Führungsgröße mit einem höherfrequenten Störsignal wird bei optimal eingestelltem Regler der Rauschanteil auch am Reglerausgang wirksam. Dies kann z. B. bei hoher Verstärkung des Reglers und eingeschalteter Differenzierung starke Schwankungen der Stellgröße verursachen. Wegen der vermehrten Schalthäufigkeit (Schrittregler) hat das einen erhöhten Verschleiß des Stellgliedes zur Folge.

Die Funktion vermindert im eingeschwungenen Zustand des Reglers den Rauschanteil im Signal der Regeldifferenz und vermindert dadurch ein unerwünschtes Oszillieren des Reglerausgangs.

#### Die Funktion DEADBAND

Die Funktion DEADBAND unterdrückt in einem einstellbaren Bereich kleine Schwankungen der Eingangsgröße um einen festgelegten Nullpunkt. Außerhalb dieser Schwankungsbreite steigt bzw. fällt die Regeldifferenz ER proportional zur Eingangsgröße. Die Breite der unwirksamen Zone (Totzone) kann mit dem Parameter DEADB\_W festgelegt werden. Die Totzonenbreite darf nur positive Werte annehmen.

Befindet sich die Eingangsgröße innerhalb der Totzone, wird am Ausgang der Wert 0 (Regeldifferenz = 0) ausgegeben. Erst wenn die Eingangsgröße diesen Unempfindlichkeitsbereich verlässt, steigt bzw. fällt der Ausgang um die gleichen Werte wie die Eingangsgröße inv. Daraus resultiert eine Verfälschung des übertragenen Signals auch außerhalb der Totzone. Das wird jedoch zur Vermeidung von Sprüngen an den Grenzen der Totzone in Kauf genommen (Bild 4-25). Die Verfälschung entspricht dem Wert DEADB\_W und ist deshalb leicht kontrollierbar.

Die Funktion DEADBAND arbeitet gemäß folgenden Funktionen:

$$\begin{aligned} (ER) &= inv + DEADB\_W && \text{für } inv < -DEADB\_W \\ (ER) &= 0 && \text{für } -DEADB\_W \leq inv \leq +DEADB\_W \\ (ER) &= inv - DEADB\_W && \text{für } inv > +DEADB\_W \end{aligned}$$

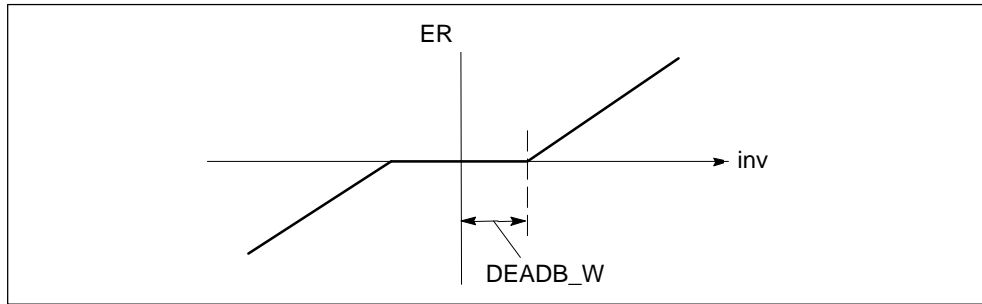


Bild 4-25 Rausch-Filterung des Regeldifferenzsignals ER durch Funktion mit Totzonenbildung

### Parameter der Funktion DEADBAND

Die Funktion DEADBAND kann abgeschaltet werden. Auswirkungen der Signalfilterung können am Ausgang 'ER' mit Hilfe des **Kurvenschreibers** (Konfigurationswerkzeug) beobachtet werden (Bild 2-13).

Der Parameter DEADB\_W kann zwischen 0.0 und der Obergrenze des techn. Wertebereichs gewählt werden.

Parameter	Bedeutung	Zulässiger Wertebereich
DEADB_W	Totzonenbreite (= Bereich Null bis Totzonenobergrenze)	0 bis Obergrenze des techn. Wertebereichs

Eingangsparameter			DEADBAND	Ausgangsparameter		
Parameter	Typ	*)		Parameter	Typ	*)
DEADB_ON	BOOL	FALSE		ER	REAL	0.0
inv	REAL					
DEADB_W	REAL	1.0				

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 4-26 Funktionsschema und Parameter der Funktion DEADBAND im Kanal der Regeldifferenz

### 4.3.2 Überwachung der Regeldifferenz auf Grenzwerte (ER\_ALARM)

#### Anwendung

Zu große Abweichungen der Regelgröße vom eingestellten Sollwert können unerwünschte Prozesszustände hervorrufen. In diesen Fällen wird die Funktion ER\_ALARM eingesetzt, um die Regeldifferenz auf Über-/Unterschreitung des zulässigen Betriebsbereiches zu überwachen. ER\_ALARM erfasst und meldet evtl. auftretende Grenzüberschreitungen, damit eine sinnvolle Reaktion ausgelöst werden kann.

#### Die Funktion ER\_ALARM

Die Funktion ER\_ALARM überwacht die Größe der Regeldifferenz  $ER(t)$  auf vier vorgebbare Grenzen in zwei Toleranzbändern. Werden die Grenzen erreicht oder überschritten, meldet die Funktion zunächst "Warnung" und an der zweiten Grenze "Alarm".

Die Zahlenwerte der Grenzen werden an den Eingangsparametern für "Warnung" und "Alarm" eingestellt (Bild 4-28). Überschreitet bzw. unterschreitet die Regeldifferenz (ER) diese Grenzen, dann werden die zugehörigen Ausgangsmeldebite QERN\_ALM ... QERP\_ALM gesetzt.

Um bei geringfügigen Änderungen der Eingangsgröße oder bei Rundungsfehlern ein "Flackern" der Meldebite zu vermeiden, wird eine Hysterese (Rückschaltdifferenz) ER\_HYS eingestellt. Diese Hysterese muss die Regeldifferenz überwinden, bevor die Meldungen rückgesetzt werden.

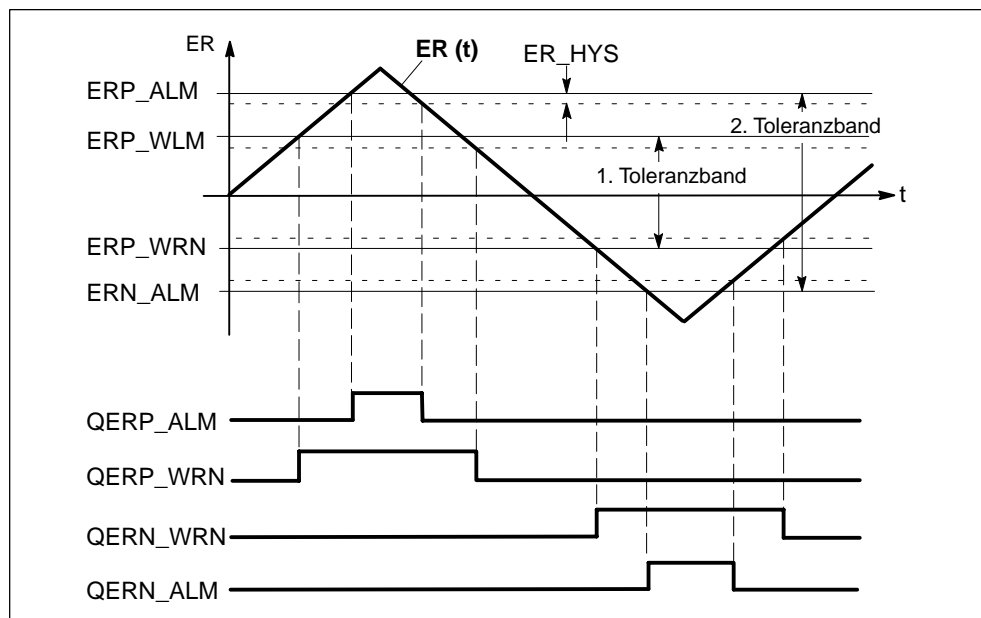


Bild 4-27 Überwachung der Regeldifferenz ER auf Grenzwertüber-/unterschreitung

### Anlauf und Betriebsweise

- Bei Neustart werden alle Meldeausgänge nullgesetzt.
- Die Grenzwertmeldung arbeitet entsprechend den folgenden Funktionen:

QERP_ ALM	QERP_ WRN	QERN_ WRN	QERN_ ALM	wenn:	und:
TRUE	TRUE	FALSE	FALSE	ER ↗ ER ↘	ER ≥ ERP_ ALM ER ≥ ERP_ ALM – ER_ HYS
FALSE	TRUE	FALSE	FALSE	ER ↗ ER ↘	ER ≥ ERP_ WRN ER ≥ ERP_ WRN – ER_ HYS
FALSE	FALSE	TRUE	FALSE	ER ↘ ER ↗	ER ≤ ERN_ WRN ER ≤ ERN_ WRN + ER_ HYS
FALSE	FALSE	TRUE	TRUE	ER ↘ ER ↗	ER ≤ ER_ ALM ER ≤ ERN_ ALM + ER_ HYS

Für ein sinnvolles Arbeiten des Bausteins muss gelten:

$$ERN\_ALM < ERN\_WRN < ERP\_WRN < ERP\_ALM$$

### Parameter der Funktion ER\_ALARM

Die Funktion ER\_ALARM zur Überwachung der Regeldifferenz kann nicht abgeschaltet werden. Bei der Konfiguration der Regelung ist deshalb besonders auf die Grenzwerteinstellung zu achten. Andernfalls werden Grenzüberschreitungen gemäß den voreingestellten Parametern (Bild 4-28) gemeldet.

Parameter	Bedeutung	Zulässiger Wertebereich
ERP_ ALM	Oberer ER-Grenzwert 'Alarm'	≥ 0.0, techn. Wertebereich
ERP_ WRN	Oberer ER-Grenzwert 'Warnung'	≥ 0.0, techn. Wertebereich
ERN_ WRN	Unterer ER-Grenzwert 'Warnung'	≤ 0.0, techn. Wertebereich
ERN_ ALM	Unterer ER-Grenzwert 'Alarm'	≤ 0.0, techn. Wertebereich

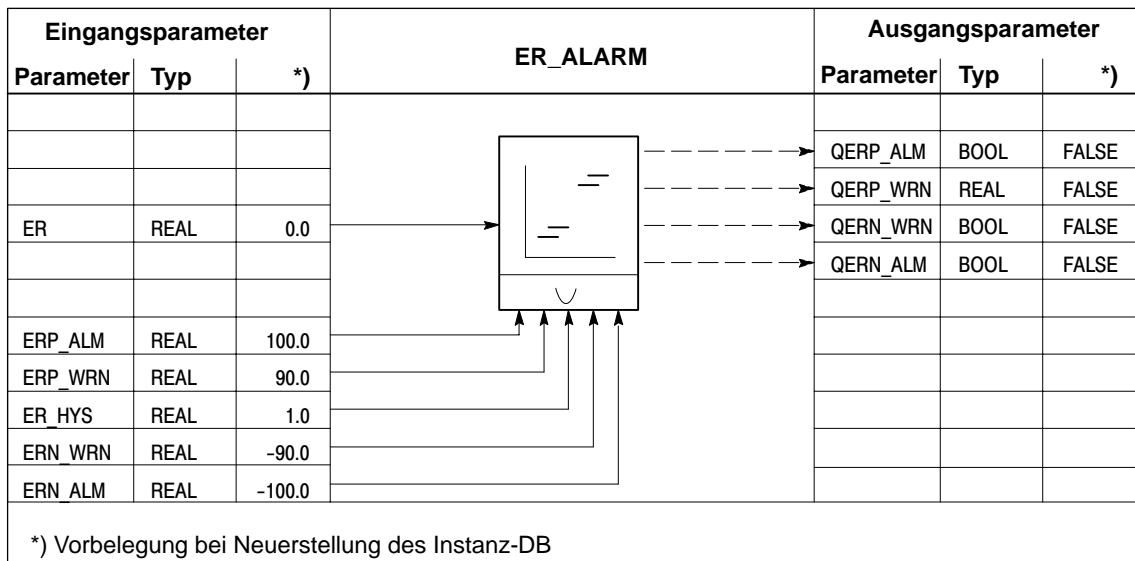


Bild 4-28 Funktionsschema und Parameter der Grenzwertüberwachung der Regeldifferenz ER

## 4.4 Die PID-Reglerfunktionen

### Normierung der Eingangsgrößen ER und PV

Die Eingangsgrößen ER und PV des PID-Reglers werden vor der Reglerbearbeitung normiert auf den Bereich 0 bis 100 gemäß folgenden Formeln:

- bei ausgeschalteter Wurzelfunktion (SQRT\_ON = FALSE):
  - $ER_{\text{normiert}} = ER * 100.0 / (NM\_PVHR - NM\_PVLR)$
  - $PV_{\text{normiert}} = (PV - NM\_PVLR) * 100.0 / (NM\_PVHR - NM\_PVLR)$
- bei eingeschalteter Wurzelfunktion (SQRT\_ON = TRUE):
  - $ER_{\text{normiert}} = ER * 100.0 / (SQRT\_HR - SQRT\_LR)$
  - $PV_{\text{normiert}} = (PV - SQRT\_LR) * 100.0 / (SQRT\_HR - SQRT\_LR)$

Diese Normierung wird durchgeführt, damit der Verstärkungsfaktor GAIN des PID-Reglers dimensionslos eingegeben werden kann. Bei einer Änderung der Ober- und Untergrenze des Physikalischen Messbereichs (z. B. von bar auf mbar) muss somit der Verstärkungsfaktor nicht geändert werden.

Die normierten Eingangsgrößen  $ER_{\text{normiert}}$  und  $PV_{\text{normiert}}$  sind nicht beobachtbar.

### Regelalgorithmus und Reglerstruktur

Im Zyklus der projektierten Abtastzeit wird die Stellgröße des kontinuierlich arbeitenden Reglers aus der Regeldifferenz im PID-Stellungsalgorithmus errechnet. Der Regler ist in reiner Parallelstruktur ausgeführt (Bild 4-29). Der Proportional-, Integral- oder Differentialanteil kann jeweils einzeln abgeschaltet werden.

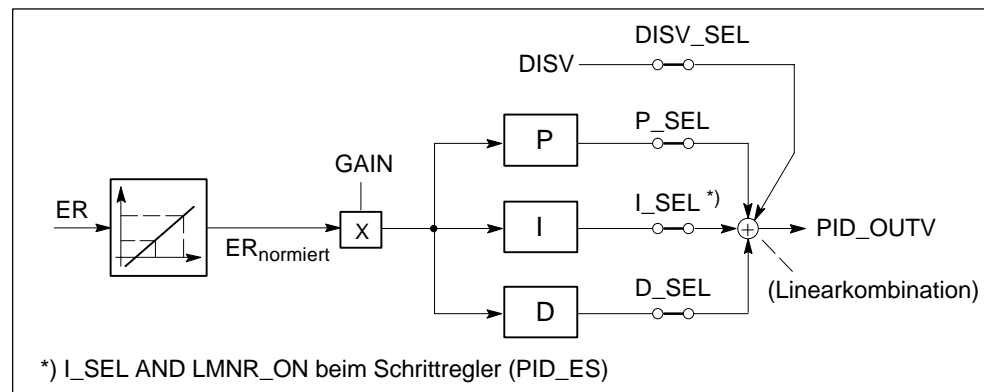


Bild 4-29 Regelalgorithmus von Standard PID Control (Parallelstruktur)

#### Störgrößenaufschaltung:

Dem Ausgangssignal des Reglers PID\_OUTV kann zusätzlich eine Störgröße **DISV** aufgeschaltet werden. Die Zu- bzw. Abschaltung erfolgt im PID-Fenster des Konfigurationswerkzeugs über den Strukturschalter DISV\_SEL bzw. durch "Störgröße ein".

**PD\_Anteil in der Rückführung:**

In der Parallelstruktur erhält jeder Anteil des Regelalgorithmus die Regeldifferenz als Eingangssignal. In dieser Struktur wirken Sollwertsprünge direkt auf den Regler. Die Stellgröße wird über den P- und den D-Anteil durch Sollwertsprünge unmittelbar beeinflusst. Eine andere Struktur des Reglers jedoch, bei der die Bildung des P- und des D-Anteils in die Rückführung verlegt wird, garantiert bei Sprungänderungen der Führungsgröße einen stoßfreien Verlauf der Stellgröße (Bild 4-30). In dieser Struktur verarbeitet der I-Anteil die Regeldifferenz als Eingangssignal, auf den P- und den D-Anteil wird nur die **negative** Regelgröße (Faktor = -1) aufgeschaltet.

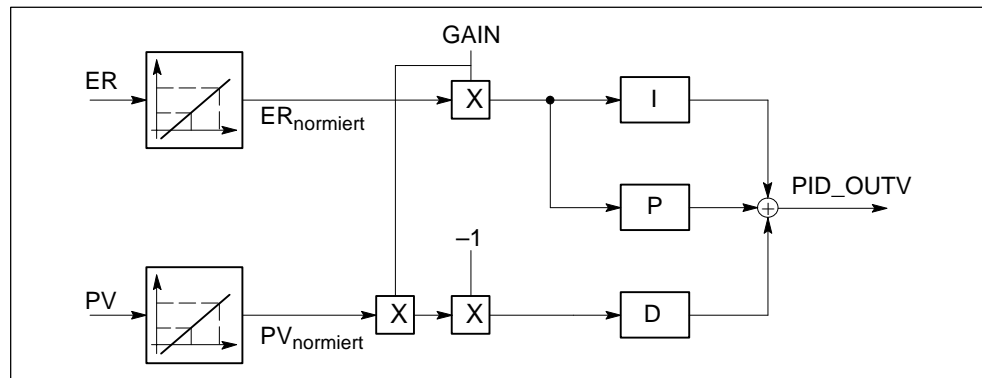


Bild 4-30 Regelalgorithmus mit P- und D-Anteil im Rückführzweig

**Festlegen der Reglerstruktur**

Um eine wirksame Reglerstruktur festzulegen, stehen insgesamt fünf Schalter (Tabelle 4-3). Die Einstellung dieser Strukturschalter erfolgt im Konfigurationswerkzeug durch Selektion der zu aktivierenden P-, I- und D-Wirkung und zwar für P und D wahlweise auch im Rückführzweig. Dies geschieht nach Selektion des PID-Reglerblocks (Blockschaltbild) im Bedienfenster "PID".

Tabelle 4-3 Festlegen der Reglerstruktur

Betriebsart \ Schalter	P_SEL	I_SEL *)	D_SEL	PFDB_SEL	DFDB_SEL
P-Regelung	TRUE	FALSE	FALSE	FALSE	FALSE
P-Regelung (P in Rückführung)	TRUE	FALSE	FALSE	TRUE	FALSE
PI-Regelung	TRUE	TRUE	FALSE	FALSE	FALSE
PI-Regelung (P in Rückführung)	TRUE	TRUE	FALSE	TRUE	FALSE
PD-Regelung	TRUE	FALSE	TRUE	FALSE	FALSE
PD-Regelung (P in Rückführung)	TRUE	FALSE	TRUE	FALSE	TRUE
PID-Regelung	TRUE	TRUE	TRUE	FALSE	FALSE
PID-Regelung (P/D in Rückführung)	TRUE	TRUE	TRUE	FALSE	TRUE

\*) Beim Schrittregler ohne Stellungsrückmeldung (PID\_ES mit LMNR\_ON = FALSE) ist der I-Anteil im PID-Algorithmus auf Null gesetzt.



## Reversierung der Reglerwirkung

Die **Reversierung**, d. h. die Umstellung des Reglers von der Zuordnung

- steigende Regelgröße PV(t) → fallende Stellgröße PID\_OUTV(t) auf
- steigende Regelgröße PV(t) → steigende Stellgröße PID\_OUTV(t)

wird durch Einstellung eines negativen Proportionalbeiwertes am Parameter GAIN vorgenommen. Das Vorzeichen dieses Parameterwertes bestimmt die Wirkungsrichtung des kontinuierlichen Reglers.

## P-Regelung

Beim P-Regler sind der I- und der D-Anteil abgeschaltet. (I\_SEL und D\_SEL = FALSE). Das bedeutet, dass bei Regeldifferenz ER = 0 auch das Ausgangssignal OUTV = 0 ist. Soll ein Arbeitspunkt ≠ 0, d. h. ein Zahlenwert für das Ausgangssignal bei Regeldifferenz Null eingestellt werden, so ist der I-Zweig zu aktivieren (Bild 4-31).

Im I-Anteil lässt sich über entsprechende Einstellung des Initialisierungswertes I\_ITLVAL ein Arbeitspunkt ≠ 0 für den P-Regler vorgeben. Setzen Sie dazu die Schalter 'I\_ITL\_ON' und 'I\_SEL' = TRUE.

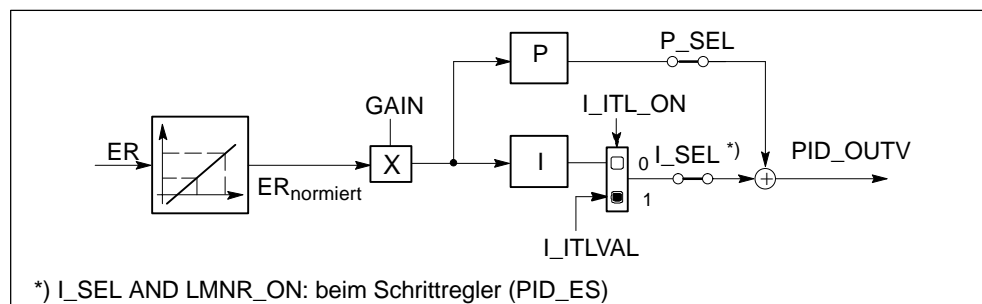


Bild 4-31 P-Regler mit Arbeitspunkteinstellung

Die Sprungantwort des P-Reglers im Zeitbereich ist:

$$\text{PID\_OUTV}(t) = \text{I\_ITLVAL} + \text{GAIN} * \text{ER}_{\text{normiert}}(t)$$

Darin ist:

PID_OUTV(t)	die Stellgröße bei Automatikbetrieb des Reglers
I_ITLVAL	der Arbeitspunkt des P-Reglers
GAIN	die Reglerverstärkung
ER_normiert(t)	die normierte Regeldifferenz

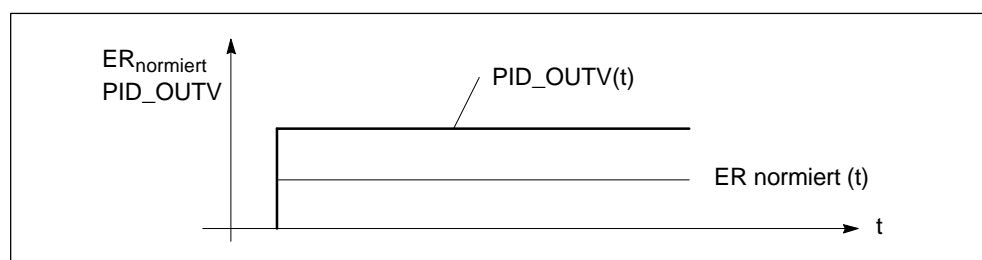


Bild 4-32 Sprungantwort des P-Reglers

## PI-Regelung

Beim PI-Regler ist der D-Anteil abgeschaltet. ( $D\_SEL = FALSE$ ). Ein PI-Regler verstellt über den I-Anteil die Ausgangsgröße  $PID\_OUTV$  so lange, bis die Regeldifferenz  $ER = 0$  geworden ist. Das gilt jedoch nur, wenn die Ausgangsgröße dabei die Grenzen des Stellbereiches nicht überschreitet.

Die Sprungantwort im Zeitbereich (Bild 4-33) ist:

$$PID\_OUTV(t) = GAIN * ER_{normiert}(0) \left( 1 + \frac{1}{TI} * t \right)$$

Darin bedeutet:

$PID\_OUTV(t)$	die Stellgröße bei Automatikbetrieb des Reglers
$GAIN$	die Reglerverstärkung
$ER_{normiert}(0)$	die Sprunghöhe der normierten Regeldifferenz
$TI$	Integrationszeitkonstante

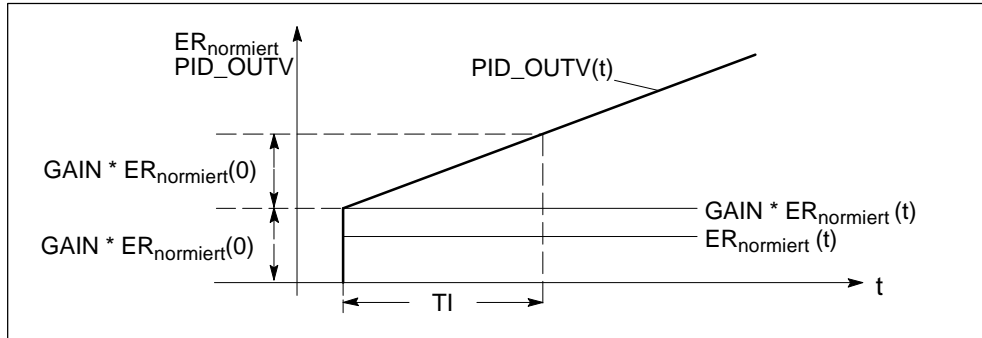


Bild 4-33 Sprungantwort des PI-Reglers

Um "stoßfrei" vom Handbetrieb in den Automatikbetrieb des PI-Reglers umzuschalten, wird das Ausgangssignal  $LMNFC\_IN - LMN\_P - DISV$  bei Beeinflussung der Stellgröße von Hand auf den internen Speicher des Integrierers geschaltet (Bild 4-34). Beim Schrittreger mit Stellungsrückmeldung wird der Integrator auf das Ausgangssignal  $LMN$  nachgeführt.

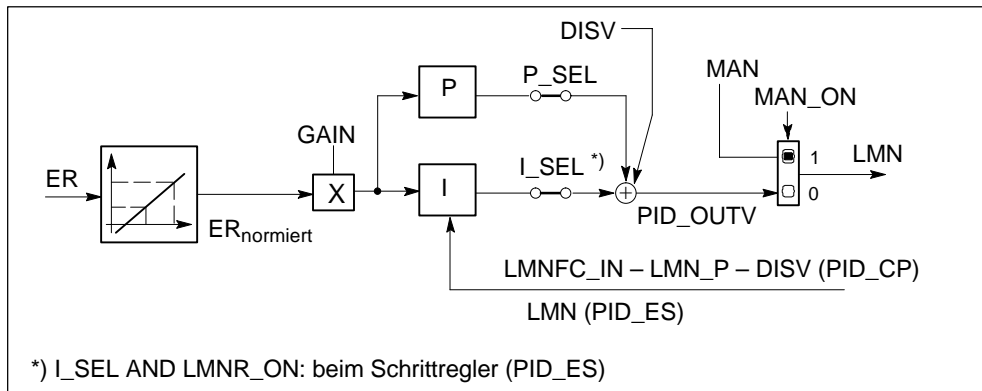


Bild 4-34 PI-Regler mit stoßfreier Umschaltung Hand- → Automatikbetrieb

Schalten Sie den P-Anteil über P-SEL ab, um eine reine I-Regelung zu realisieren.

## PD-Regelung

Beim PD-Regler ist der I-Anteil abgeschaltet (I-SEL = FALSE). Das bedeutet, dass bei Regeldifferenz  $ER = 0$  auch das Ausgangssignal  $OUTV = 0$  ist. Soll ein Arbeitspunkt  $\neq 0$ , d. h. ein Zahlenwert für das Ausgangssignal bei Regeldifferenz Null eingestellt werden, so muss der I-Zweig aktiviert werden (Bild 4-31).

Im I-Anteil lässt sich über entsprechende Einstellung des Initialisierungswertes I\_ITLVAL ein Arbeitspunkt  $\neq 0$  für den PD-Regler vorgeben. Dazu sind die Schalter 'I\_ITL\_ON' und 'I\_SEL' auf TRUE zu setzen.

Der PD-Regler bildet die Eingangsgröße  $ER(t)$  proportional auf das Ausgangssignal ab und addiert den durch Differentiation von  $ER(t)$  gebildeten D-Anteil dazu, der nach der Trapezregel (Padé-Näherung) doppelgenau berechnet wird. Das Zeitverhalten wird durch die Differentiationszeitkonstante (Vorhaltzeit) TD bestimmt.

Zur Signalglättung und Unterdrückung von Störsignalen ist eine Verzögerung 1. Ordnung (einstellbare Zeit: TM\_LAG) in den Algorithmus zur Bildung des D-Anteils integriert. Meist genügt ein kleiner Wert für TM\_LAG, um den gewünschten Erfolg zu erzielen. Wird  $TM\_LAG \leq CYCLE/2$  parametrisiert, dann ist die Verzögerung abgeschaltet.

Die Sprungantwort im Zeitbereich (Bild 4-35) ist:

$$PID\_OUTV(t) = GAIN * ER_{normiert}(0) \left( 1 + \frac{TD}{TM\_LAG} * e^{-\frac{t}{TM\_LAG}} \right)$$

Darin bedeutet:

PID_OUTV(t)	die Stellgröße bei Automatikbetrieb des Reglers
GAIN	die Reglerverstärkung
$ER_{normiert}(0)$	die Sprunghöhe der normierten Regeldifferenz
TD	Differentiationszeit
TM_LAG	Verzögerungszeit

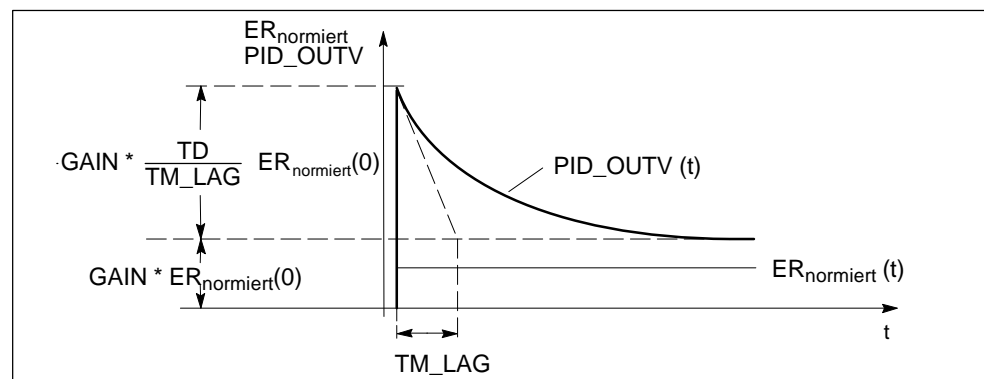


Bild 4-35 Sprungantwort des PD-Reglers

## PID-Regelung

Beim PID-Regler sind der P-, I- und D-Anteil eingeschaltet (P\_SEL = TRUE, I\_SEL = TRUE, D\_SEL = TRUE). Ein PID-Regler verstellt die Ausgangsgröße PID\_OUTV über den I-Anteil so lange, bis die Regeldifferenz ER = 0 geworden ist. Das gilt jedoch nur, wenn die Ausgangsgröße dabei die Grenzen des Stellbereiches nicht überschreitet. Werden Stellwertbegrenzungen überschritten, behält der I-Anteil den an der Grenze erreichten Wert bei (Anti Reset Wind-up).

Der PID-Regler bildet die normierte Eingangsgröße  $ER_{normiert}(t)$  proportional auf das Ausgangssignal ab und addiert die durch Differentiation und Integration von  $ER_{normiert}(t)$  gebildeten Anteile dazu, die nach der Trapezregel (Padé-Näherung) doppeltgenau berechnet werden. Das Zeitverhalten wird durch die Differentiationszeit (Vorhaltzeit) TD und die Integrationszeit (Nachstellzeit) TI bestimmt.

Zur Signalglättung und Unterdrückung von Störsignalen ist eine Verzögerung 1. Ordnung (einstellbare Zeitkonstante: TM\_LAG) in den Algorithmus zur Bildung des D-Anteils integriert. Meist genügt ein kleiner Wert für TM\_LAG, um den gewünschten Erfolg zu erzielen. Wird  $TM\_LAG \leq CYCLE/2$  parametrisiert, dann ist die Verzögerung abgeschaltet.

Die Sprungantwort im Zeitbereich (Bild 4-36) ist:

$$PID\_OUTV(t) = GAIN * ER_{normiert}(0) \left( 1 + \frac{1}{TI} * t + \frac{TD}{TM\_LAG} * e^{-\frac{t}{TM\_LAG}} \right)$$

Darin bedeutet:

PID_OUTV(t)	die Stellgröße bei Automatikbetrieb des Reglers
$ER_{normiert}(0)$	die Sprunghöhe der normierten Regeldifferenz
GAIN	die Reglerverstärkung (= GAIN)
TI	Integrationszeit
TD	Differentiationszeit
TM_LAG	Verzögerungszeit

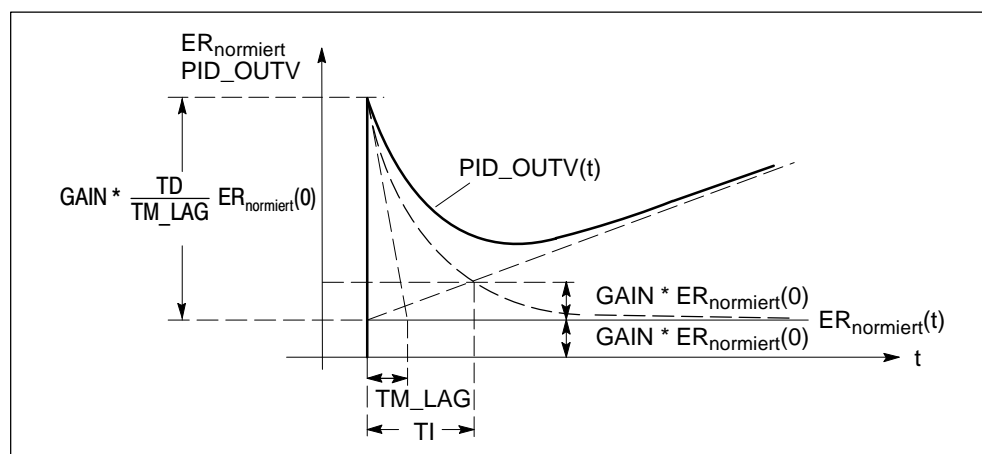


Bild 4-36 Sprungantwort des PID-Reglers

---

### Hinweis

Bei einer Änderung von TD sollten Sie auch TM\_LAG entsprechend anpassen.

Empfehlung:  $5 \leq (TM / TM\_LAG) \leq 10$

---

## Einsatz und Parametrierung des PID-Reglers

Die meisten der in der Verfahrens- und Prozessindustrie vorkommenden Regelstrecken lassen sich mit der PI-/PID-Reglerfunktion von Standard PID Control regeln. Nur in Spezialfällen sind zusätzliche Methoden und Maßnahmen zur Beherrschung der betreffenden Regelung erforderlich (*siehe Kapitel 1.2, weitere S7-Softwarepakete für Regelungsaufgaben*).

Ein großes praktisches Problem ist jedoch nach wie vor die Parametrierung des PI-/PID-Reglers, d. h. die "richtigen" Einstellwerte für die Reglerparameter zu finden. Die Qualität dieser Parametrierung ist von ausschlaggebender Bedeutung für die aufgabengemäße Wirkung der PID-Regelung und erfordert entweder große praktische Erfahrung, Spezialkenntnisse oder einen hohen Zeitaufwand.

Mit dem **Konfigurationswerkzeug** entfallen jedoch die genannten Schwierigkeiten. Die darin enthaltene Funktion **PID-Regler optimieren** ermöglicht die Ersteinstellung der Reglerparameter durch adaptive Inbetriebnahme. Hierbei wird nach einer Streckenidentifikation das Prozessmodell ermittelt und anschließend daraus möglichst günstige (optimale) Einstellwerte für die Reglerparameter errechnet. Dieser weitgehend automatische Vorgang enthebt den Anwender der lästigen Aufgabe, den installierten PID-Regler online von Hand zu "trimmen".

## 4.5 Signalverarbeitung im PID-Regleralgorithmus

### 4.5.1 Integrierer (INT)

#### Anwendung

Die Funktion des Integrierers wird in Standard-Reglern für PI- und PID-Regelungen zur Bildung des I-Anteils eingesetzt. Die Integratorwirkung in diesen Reglern sorgt dafür, dass durch Nachstellung des Arbeitspunktes die Regeldifferenz für beliebige Werte der Stellgröße zu 0 werden kann.

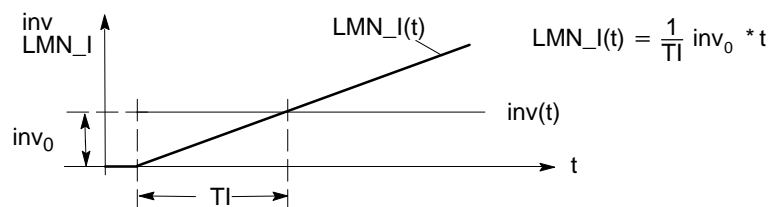
#### Die Funktion INT

Die Integrier-Funktion erzeugt ein Ausgangssignal, dessen Änderungsgeschwindigkeit sich proportional zur Änderung des Absolutwertes der Eingangsgröße verhält. Das Zeitverhalten wird durch die Integrationszeitkonstante (Nachstellzeit) TI bestimmt.

Die Übertragungsfunktion im Zeitbereich ist:

$$\text{OUTV}(t) = \frac{1}{T_I} \int \text{inv}(t) dt$$

Die Sprungantwort auf einen Eingangssprung  $\text{inv}_0$  ist:



Darin bedeutet:

$\text{LMN}_I(t)$	die Ausgangsgröße des Integrierers
$\text{inv}_0$	die Sprunghöhe am Integrierereingang
$T_I$	Integrationszeitkonstante

## Zulässige Bereiche für TI und CYCLE

Wegen der begrenzten Genauigkeit der in der CPU errechneten REAL-Zahlen kann beim Integrieren folgender Effekt auftreten: wurde die Abtastzeit CYCLE der Regelung im Vergleich zur Integrationszeit TI zu klein gewählt und ist der Eingangswert  $inv$  des Integrierers gegenüber seinem Ausgangswert OUTV klein, dann spricht der Integrierer nicht an und bleibt auf seinem momentanen Ausgangswert stehen.

Dieser Effekt lässt sich vermeiden, wenn bei der Parametrierung folgende Dimensionierungsregel beachtet wird:

$$CYCLE > 10^{-4} * TI$$

Damit reagiert der Integrierer noch auf Änderungen der Eingangswerte, die im Bereich von millionstel Promille der aktuellen Ausgangsgröße liegen:

$$inv > 10^{-10} * OUTV$$

Damit das Übertragungsverhalten des Integrier-Algorithmus dem analogen Verhalten entspricht, sollte die Abtastzeit kleiner als 20 % der eingestellten Integrationszeit sein bzw. TI mindestens den fünffachen Wert der gewählten Abtastzeit haben:

$$CYCLE < 0,2 * TI$$

Der Algorithmus lässt Werte für die Abtastzeit bis  $CYCLE \leq 0,5 * TI$  zu.

## Anlauf und Betriebsarten

- **I-Anteil vorbelegen**

Wird I\_ITL\_ON = TRUE aktiviert, dann wird der Vorbelegungswert I\_ITLVAL auf den Ausgang durchgeschaltet. Beim Übergang in den Normalbetrieb durch zurücksetzen von I\_ITL\_ON = FALSE beginnt der Integrierer von I\_ITLVAL aus mit der Integration seiner Eingangsgröße (Bild 4-37).

- Kontinuierlicher Regler PID\_CP

Im Handbetrieb wird der I-Anteil des Reglers so nachgeführt, dass der Regler bei der Umschaltung auf Automatikbetrieb mit einer sinnvollen Stellgröße beginnt. Folgende Einstellungen sind wählbar:

**Stoßfreie Umschaltung von Hand nach Automatik:**

Mit SMOO\_CHG = TRUE (Voreinstellung) wird der I-Anteil im Handbetrieb so gesetzt, dass die Stellgröße bei der Hand-Automatik-Umschaltung zunächst unverändert bleibt. Eine anstehende Regelabweichung wird langsam ausgeregelt.

**Keine stoßfreie Umschaltung von Hand nach Automatik:**

Mit SMOO\_CHG = FALSE wird der I-Anteil im Handbetrieb so gesetzt, dass die Stellgröße bei der Hand-Automatik-Umschaltung ausgehend vom Handstellwert einen Sprung (durch den P- und D-Anteil) macht. Die Sprunghöhe entspricht der Stellgrößenänderung bei einem Sollwertsprung vom aktuellen Istwert auf den aktuellen Sollwert. Die anstehende Regelabweichung wird schneller ausgeregelt. Dies ist z. B. bei Temperaturregelstrecken wünschenswert.

Ist jedoch der P-Anteil in die Rückführung gelegt (PFDB\_SEL = TRUE), dann wirkt nur der Istwert auf den P-Anteil. Die Stellgröße macht daher – wie auch bei einem Sollwertsprung - bei einer Hand-Automatik-Umschaltung keinen Sprung durch den P-Anteil; die Umschaltung ist stoßfrei. Das gleiche gilt auch für den D-Anteil, falls dieser in die Rückführung gelegt wurde (DFDB\_SEL = TRUE).

- Schrittreger PID\_ES

Der I-Anteil wird im Handbetrieb so gesetzt, dass das Stellglied bei der Hand-Automatik-Umschaltung ausgehend von der aktuellen Stellung um die Sprunghöhe des P-Anteils verfahren wird. Die Sprunghöhe des P-Anteils entspricht der Stellgrößenänderung bei einem Sollwertsprung vom aktuellen Istwert auf den aktuellen Sollwert. Ist jedoch der P-Anteil in die Rückführung gelegt (PFDB\_SEL = TRUE), dann wirkt nur der Istwert auf den P-Anteil. Die Stellgröße macht daher – wie auch bei einem Sollwertsprung – bei einer Hand-Automatik-Umschaltung keinen Sprung durch den P-Anteil; die Umschaltung ist stoßfrei. Der D-Anteil wird im Handbetrieb auf Null gesetzt und bleibt auch bei der Hand-Automatik-Umschaltung Null.

- **Handbetrieb**

Bei Handbeeinflussung des Stellsignals, d. h. wenn MAN\_ON, LMNOP\_ON oder CAS\_ON = TRUE, wird der interne Speicherwert des Integrierers dem Wert LMNFC\_IN – LMN\_P – DISV nachgeführt (Bild 4-34). Beim Schrittreger mit Stellungsrückmeldung (PID\_ES) wird der Integrator auf das Ausgangssignal LMN nachgeführt.



• **Integrator anhalten**

Über die Binäreingänge INT\_HPOS und INT\_HNEG kann der Integrierer in positive bzw. negative Richtung blockiert werden. Dies kann sinnvoll sein bei Reglerkaskaden. Gerät z. B. die Stellgröße des Folgeregler in die obere Begrenzung, kann somit eine weitere Vergrößerung der Stellgröße des Führungsregler durch dessen Integrator verhindert werden. Sie realisieren dies durch die folgenden Anweisungen:

AWL	Erläuterung
U	"Folgeregler".QLMN_HLM
=	"Fuehrungsregler".INT_HPOS
U	"Folgeregler".QLMN_LLM
=	"Fuehrungsregler".INT_HNEG

• **Integrieren**

Ist der Schalter I\_SEL = TRUE gesetzt, dann ist der Integrierbetrieb ausgehend vom Wert I\_ITLVAL aktiviert. Das dynamische Verhalten der Funktion wird durch den Wert der Integrationszeitkonstante TI bestimmt.

Im ausgeschalteten Zustand (I\_SEL = FALSE) wird der I-Anteil, d. h. der interne Speicher und der Ausgang LMN\_I des Integrators auf Null gesetzt.

Betriebsart \ Schalter	I_ITL_ON	MAN_ON oder LMNOP_ON	INT_HPOS	INT_HNEG
I-Anteil (LMN_I) vorbelegen	TRUE	beliebig	beliebig	beliebig
Handbetrieb	FALSE	TRUE	beliebig	beliebig
Integrator in pos. Richtung blockieren	FALSE	FALSE	TRUE	FALSE
Integrator in neg. Richtung blockieren	FALSE	FALSE	FALSE	TRUE
Integrator in beide Richtungen blockieren	FALSE	FALSE	TRUE	TRUE
Integrieren	FALSE	FALSE	FALSE	FALSE

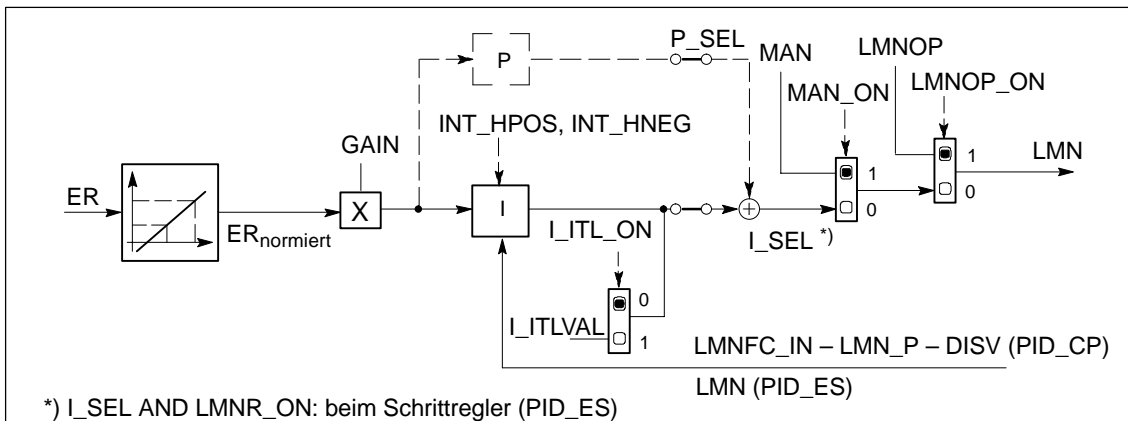


Bild 4-37 Betriebsarten des Integrierers im PI-/PID-Regler

### Begrenzungsverhalten

Der Ausgang und der Speicher des Integrierers wird durch die obere und untere Begrenzung LMN\_HLM und LMN\_LLM begrenzt (Anti Reset Wind-up).

### Parameter der Funktion INT

Der Ausgangswert OUTV des Integrierers ist am Parameter LMN\_I zu beobachten.

Parameter	Bedeutung	Zulässiger Wertebereich
TI	Integrationszeit	$\geq 5 * \text{CYCLE}$
I_ITLVAL	Initialisierungswert für I-Anteil	-100.0 bis +100.0 [%]

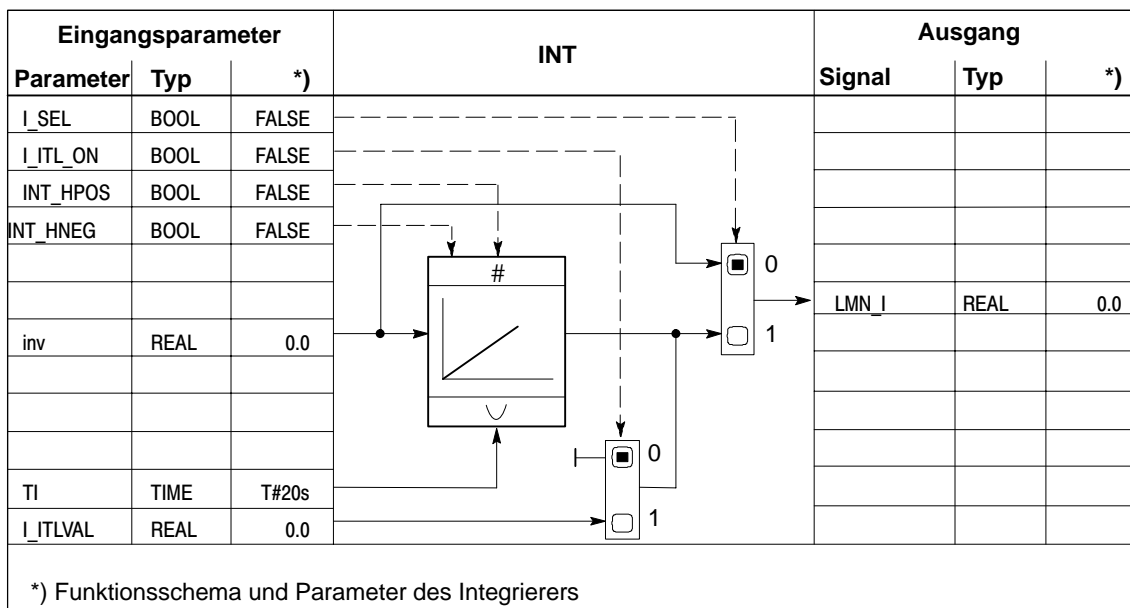


Bild 4-38 Funktionsschema und Parameter des Integrierers

## 4.5.2 Differenzierer (DIF)

### Anwendung

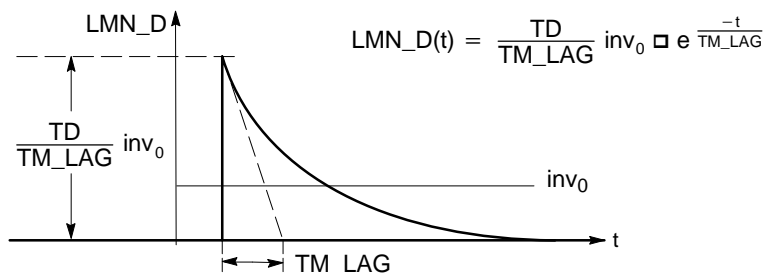
Die Funktion des Differenzierers wird eingesetzt, um den D-Anteils in Standard-Reglern für PD- und PID-Regelungen zu bilden. Die Regelgröße wird dynamisch differenziert.

### Die Funktion DIF

Die Differenzier-Funktion erzeugt ein Ausgangssignal, dessen Größe sich proportional zur Änderungsgeschwindigkeit der Eingangsgröße ändert. Das Zeitverhalten wird durch die Differenzierzeitkonstante (Vorhaltzeit) TD und die Verzögerungszeit des Differenzierers TM\_LAG bestimmt.

Zur Signalglättung und Unterdrückung von Störsignalen ist eine Verzögerung 1. Ordnung integriert, deren Zeitkonstante am Parameter TM\_LAG eingestellt wird.

Die Sprungantwort auf einen Eingangssprung  $inv_0$  ist:



Darin bedeutet:

LMN_D(t)	die Ausgangsgröße des Differenzierers
$inv_0$	die Sprunghöhe am Differenzierereingang
TD	Differenzierzeitkonstante
TM_LAG	Verzögerungszeitkonstante

### Zulässige Bereiche für TD und CYCLE

Damit der Differenzierer bzw. sein Berechnungsalgorithmus in der CPU korrekt arbeiten können, müssen bezüglich der zulässigen Einstellbereiche bei der Parametrierung der Zeitkonstanten folgende Bedingungen eingehalten werden:

$$TD \geq CYCLE \text{ und} \\ TM\_LAG \geq 0,5 * CYCLE$$

Ist für TD ein Wert  $< CYCLE$  eingestellt, dann arbeitet der Differenzierer so, als würde TD den Wert CYCLE haben.

Ist für TM\_LAG ein Wert  $< 0,5 * CYCLE$  eingestellt, dann arbeitet der Differenzierer ohne Verzögerung. Ein Eingangssprung wird dann mit dem Faktor TD/CYCLE multipliziert und dieser Wert als "Nadelimpuls" auf den Ausgang gegeben. Das heißt im folgenden Bearbeitungszyklus wird LMN\_D wieder auf Null zurückgenommen.

### Anlauf und Betriebsarten

- **Handbetrieb**

Wenn eine stoßfreie Umschaltung von Hand nach Automatik gewählt wurde (SMOO\_CHG = TRUE), dann wird der D-Anteil im Handbetrieb auf Null gesetzt. Die Umschaltung auf Automatik erfolgt ohne Stellgrößensprung.

Wenn keine stoßfreie Umschaltung von Hand nach Automatik gewählt wurde (SMOO\_CHG = FALSE), dann wird der D-Anteil im Handbetrieb auf einen Wert gesetzt, der der anstehenden Regeldifferenz entspricht. Die Umschaltung auf Automatik erfolgt mit Stellgrößensprung, die Regeldifferenz wird schneller ausgeglichen.

- **Differenzieren**

Ist der Schalter D\_SEL = TRUE gesetzt, dann ist der Differenzierbetrieb aktiviert. Das dynamische Verhalten der Funktion wird durch den Wert der Differenzierzeitkonstante TD und der Verzögerungszeit TM\_LAG bestimmt.

Im ausgeschalteten Zustand (D\_SEL = FALSE) wird der D-Anteil, d. h. der interne Speicher und der Ausgang LMN\_D des Differenzierers nullgesetzt.

Betriebsart \ Schalter	MAN_ON oder LMNOP_ON
Handbetrieb	TRUE
Differenzieren	FALSE

### Parameter der Funktion DIF

Der Ausgangswert des Differenzierers ist am Parameter LMN\_D zu beobachten.

Parameter	Bedeutung	Zulässiger Wertebereich
TD	Differenzierzeit	≥ CYCLE
TM_LAG	Verzögerungszeit des D-Anteils	≥ 0,5 * CYCLE

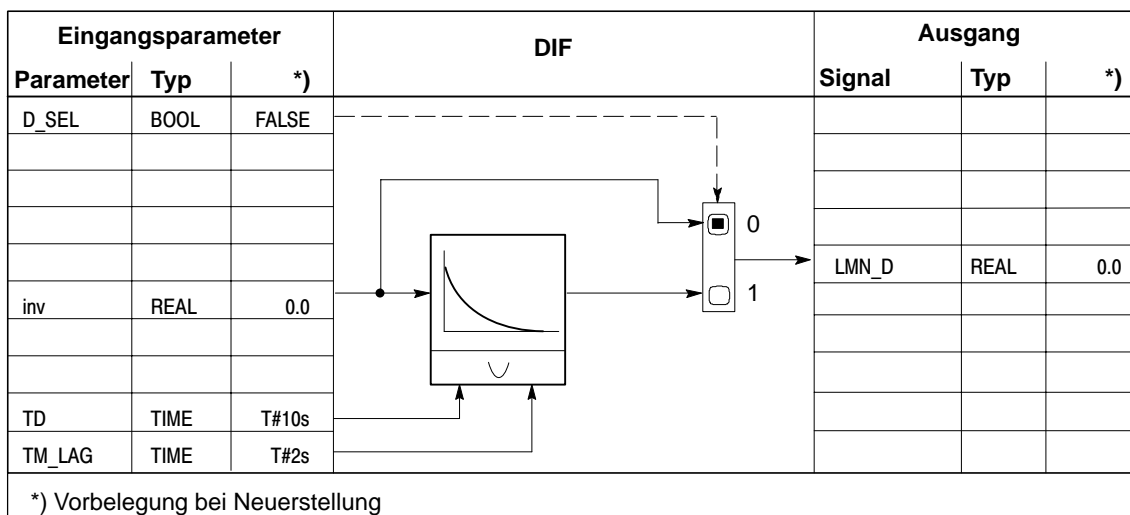


Bild 4-39 Funktionsschema und Parameter des Differenzierers

## Der kontinuierliche Regler (PID\_CP)

### 5.1 Regelfunktionen des kontinuierlichen PID-Reglers

#### Der Funktionsbaustein PID\_CP

Neben den Funktionen im Soll- und Istwertzeig realisiert der Funktionsbaustein (FB) einen fertigen PID-Regler mit kontinuierlichem Stellgrößen-Ausgang und Beeinflussungsmöglichkeit des Stellwertes von Hand. Teilfunktionen lassen sich zu- oder abschalten.

Mit dem FB haben Sie die Möglichkeit, technische Prozesse und Anlagen mit kontinuierlichen Ein- und Ausgangsgrößen auf den Automatisierungssystemen SIMATIC S7 zu regeln. Der Regler kann als Festwertregler einzeln oder auch in mehrschleifigen Regelungen als Kaskaden-, Mischungs- oder Verhältnisregler eingesetzt werden.

#### Funktionsschema des kontinuierlichen Reglers

Die Arbeitsweise basiert auf dem PID-Regelalgorithmus des Abtastreglers mit analogem Ausgangssignal, gegebenenfalls um eine Impulsformerstufe zur Bildung von pulsbreitenmodulierten Ausgangssignalen für Zwei- oder Dreipunktregelungen mit proportionalen Stellgliedern ergänzt.

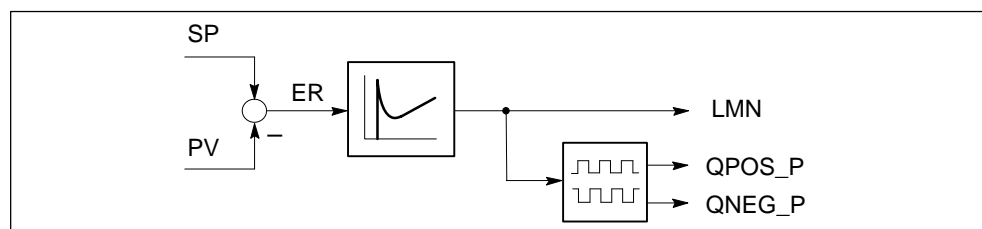


Bild 5-1 Funktionsschema des Reglers mit kontinuierlichem Stellsignal  
(Softwarepaket "Standard PID Control")

### **Neustart/Wiederanlauf**

Der FB PID\_CP verfügt über eine Initialisierungsroutine, die durchlaufen wird, wenn der Eingangs-Parameter COM\_RST = TRUE gesetzt ist.

#### **Zeitplangeber (RMP\_SOAK)**

Bei eingeschaltetem Zeitplangeber werden die Zeitabschnitte DB\_NBR PI[0 ... NBR\_PTS].TMV zwischen den Stützpunkten aufsummiert und am Ausgang die Gesamtzeit T\_TM und Gesamtrestzeit RT\_TM angezeigt.

Bei Online-Änderungen von PI[n].TMV oder bei Vorgabe von TM\_CONT und TM\_SNBR ändern sich Gesamtzeit und Gesamtrestzeit der Fahrkurve. Da die Berechnung von T\_TM und RS\_TM bei vielen Stützpunkten die Bearbeitungszeit der Funktion RMP\_SOAK stark vergrößert, wird diese Berechnung nur nach Neustart oder bei TUPDT\_ON = TRUE durchgeführt.

#### **Integrierer (INT)**

Der Integrierer wird beim Anlauf auf den Initialisierungswert I\_ITLVAL gesetzt und der I-Anteil am Ausgang LMN\_I ausgegeben. Beim Aufruf in einer Weckalarmebene arbeitet er von diesem Wert aus weiter. Alle anderen Ausgänge werden auf ihre Vorbelegungswerte gesetzt.

## 5.2 Signalverarbeitung der Stellgröße

### 5.2.1 Betriebsarten der Stellsignalbildung

#### Handbetrieb und Betriebsarten-Umschaltung

Bei Standard PID Control gibt es neben der Betriebsart "Automatik" mit auf den Ausgang durchgeschaltetem Ausgang des PID-Algorithmus (PID\_OUTV) die zwei Betriebsarten, bei denen das Stellsignal manuell beeinflusst werden kann: "Handbetrieb ohne Schalterbedienung" und "Handbetrieb über Mehr-/Weniger-Schalter" (MAN\_GEN).

Über den Parameter MAN kann eine externe Stellgrößenbeeinflussung aufgeschaltet oder vom Anwenderprogramm aus vorgegeben werden. Der Eingangswert MAN wird auf die Stellgrößenbegrenzungen LMN\_HLM (obere) und LMN\_LLM (untere) begrenzt.

Die Struktur der Handwert-Erzeugung und -Aufschaltung geht aus Bild 5-2 hervor. Beim Einschalten von MAN\_GEN aus einer anderen Betriebsart wird der momentan wirksame Stellwert am Ausgang von MAN\_GEN übernommen. Die Umschaltung zum Handwertgenerator ist damit immer stoßfrei.

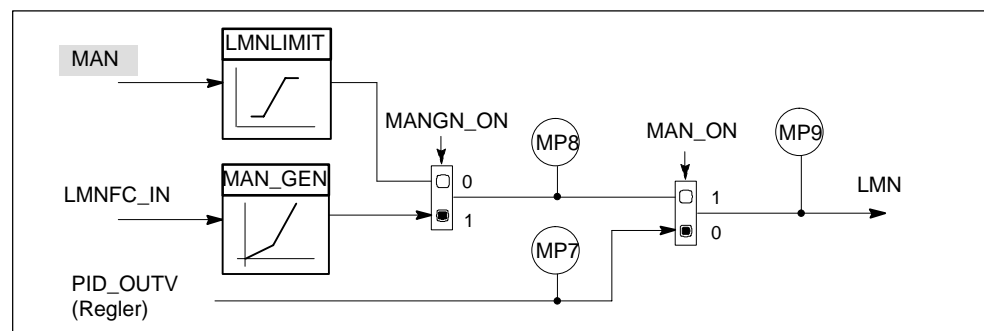


Bild 5-2 Handwert-Erzeugung bei Standard PID Control

#### Automatikbetrieb

Ist `MAN_ON = FALSE` (Blockschaltbild im Konfigurationswerkzeug) gewählt, dann wird der Stellwert des PID-Algorithmus auf den Ausgang durchgeschaltet. Im Handbetrieb (`MAN_ON = TRUE`) wird der I-Anteil des Reglers so nachgeführt, dass der Regler bei der Umschaltung auf Automatikbetrieb mit einer sinnvollen Stellgröße beginnt (siehe "Anlauf und Betriebsarten" im Abschnitt 4.5.1). Der Ausgang des PID-Algorithmus ist am Messpunkt MP7 hinterlegt.

Im Automatikbetrieb wird der Handwert MAN der Stellgröße (abzüglich D-Anteil) nachgeführt. Bei der Umschaltung auf Handbetrieb bleibt die Stellgröße daher auf dem zuletzt berechneten Wert stehen. Sie kann per Bedienung verändert werden.

### Handbetrieb ohne Schalterbedienung

In dieser Betriebsart (MANGN\_ON = FALSE und MAN\_ON = TRUE) wird der Handstellwert als Absolutwert am Eingang MAN eingegeben. Der Handstellwert wird am Messpunkt MP8 angezeigt.

### Handbetrieb mit Schalterbedienung

In dieser Betriebsart (MANGN\_ON = TRUE und MAN\_ON = TRUE) wird der aktuelle Stellwert über die Schalter von MAN\_GEN innerhalb der Stellgrößenbegrenzungen vergrößert oder verkleinert.

### Schaltung der Betriebsarten

Folgende Tabelle zeigt die möglichen Betriebsarten des kontinuierlichen Reglers mit den erforderlichen Werten der strukturbestimmenden Schalter.

Tabelle 5-1 Betriebsarten des kontinuierlichen Reglers

Betriebsart \ Schalter	MANGN_ON	MAN_ON
Automatikbetrieb	beliebig	FALSE
Handbetrieb ohne Schalterbedienung	FALSE	TRUE
Handbetrieb über Mehr-/Weniger-Schalter	TRUE	TRUE



## 5.2.2 Betriebsarten der Stellsignalbildung

### Anwendung

Stellwertbeeinflussung von Hand mit Hilfe eines Mehr-/Weniger-Schalters. Der eingestellte Wert wird an MP8 simultan angezeigt.

### Die Funktion MAN\_GEN

Die Funktion MAN\_GEN erzeugt einen Stellwert, der per Schalterbetrieb eingestellt und verändert werden kann. Über die Binäreingänge MANUP und MANDN lässt sich die Ausgangsgröße outv schrittweise vergrößern oder verkleinern.

Der Verstellbereich des Stellwertes wird durch die obere/untere Grenze LMN\_HLM/LMN\_LLM in der Begrenzungsfunktion LMNLIMIT eingeschränkt. Die Zahlenwerte der Grenzen (in Prozent) werden an den betreffenden Eingangsparametern eingestellt. Für eine feinstufige Änderung sollte der Regler eine Abtastzeit von maximal 100 ms haben.

Die Änderungsgeschwindigkeit der Ausgangsgröße hängt von der Betätigungsdauer von MANUP bzw. MANDN und von den eingestellten Begrenzungen nach folgender Beziehung ab: Während der ersten 3 s nach Setzen von MANUP bzw. MANDN ist:

$$\text{die Steigung von outv} = \frac{\text{LMN\_HLM} - \text{LMN\_LLM}}{100 \text{ s}}$$

$$\text{danach:} = \frac{\text{LMN\_HLM} - \text{LMN\_LLM}}{10 \text{ s}}$$

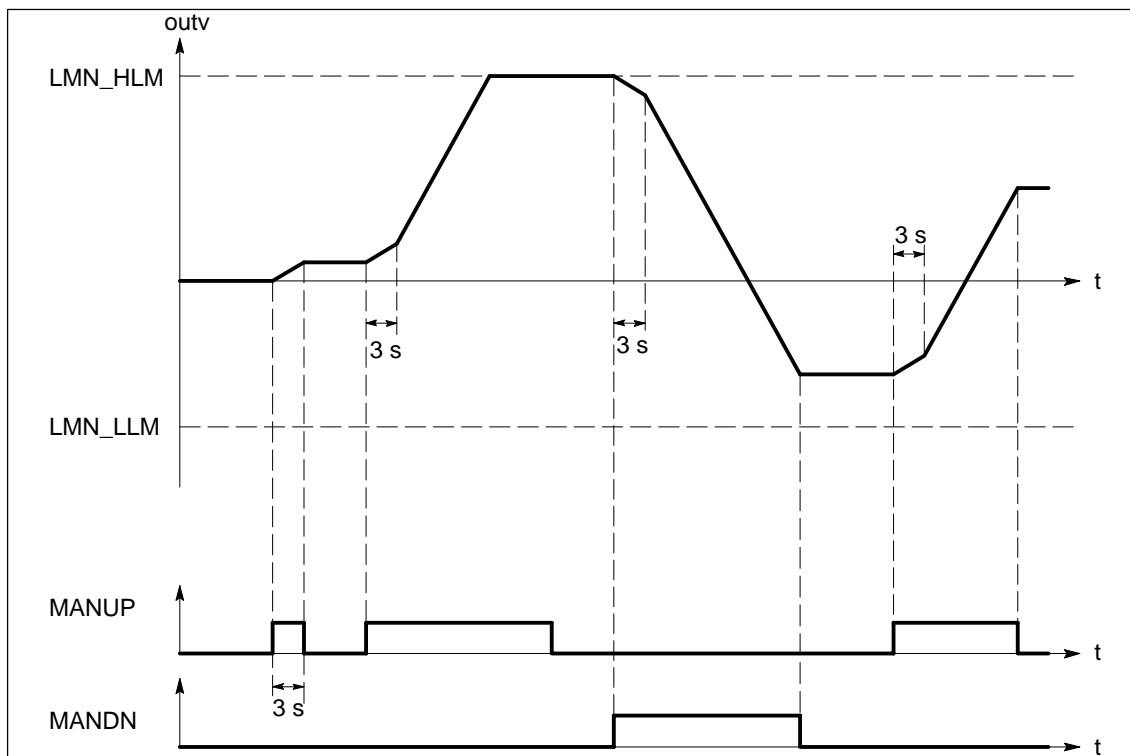


Bild 5-3 Änderung des Stellwertes in Abhängigkeit von den Schaltern MANUP und MANDN

Bei einer Abtastzeit von 100 ms und einem Stellwertbereich von –100,0 bis 100,0 Prozent ändert sich z. B. der Stellwert in den ersten drei Sekunden um 0,2 % pro Durchlauf. Bei längerer Tastung von MANUP geht die Änderungsgeschwindigkeit dann auf den zehnfachen Wert, hier also auf 2 % pro Durchlauf (Bild 5-3).

### Anlauf und Betriebsweise des Stellwertgenerators

- Bei Neustart wird der Ausgang outv auf 0.0 zurückgesetzt.
- Schalten Sie den Stellwertgenerator (MANGN\_ON = TRUE) ein, dann wird am Ausgang outv zunächst der Signalwert LMNFC\_IN ausgegeben. Deshalb wird immer stoßfrei aus einer anderen Betriebsart zum Stellwertgenerator umgeschaltet. Solange MANUP oder MANDN (Auf-/Ab-Tasten des Konfigurationswerkzeugs) nicht aktiviert werden, bleibt LMNFC\_IN am Ausgang stehen.

### Parameter der Funktion MAN\_GEN

Der Ausgangsparameter outv ist ein impliziter Parameter. Er ist am Konfigurationswerkzeug über den Messpunkt MP8 zugänglich.

Parameter	Bedeutung	Zulässiger Wertebereich
MAN	Hand-Stellwert	– 100.0 bis 100.0 [%]

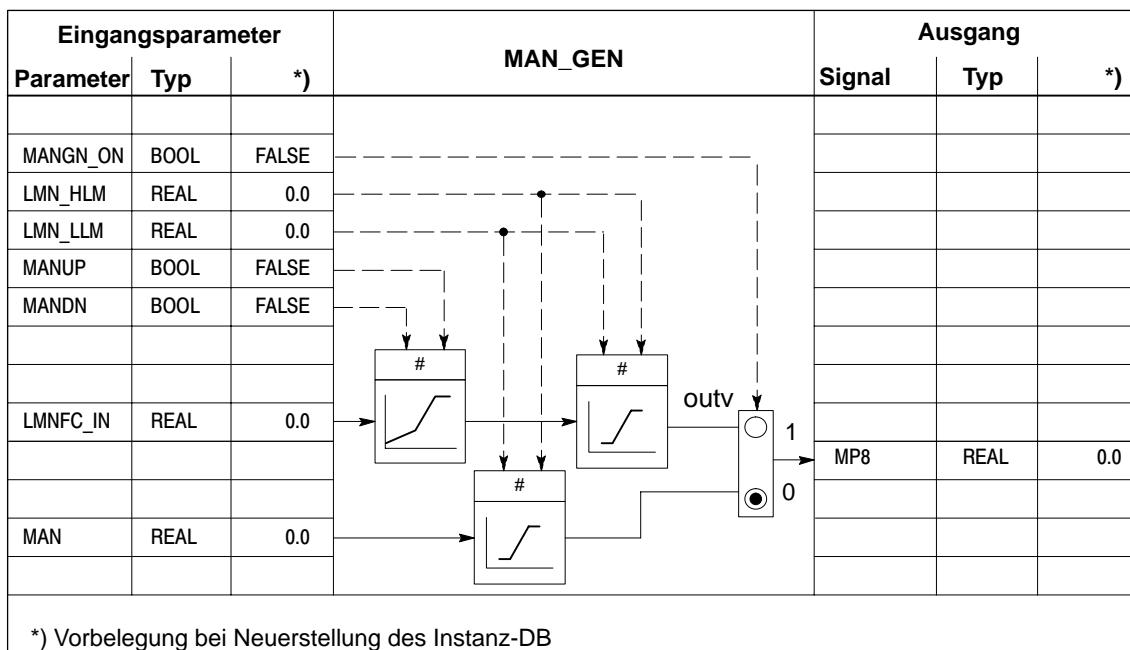


Bild 5-4 Funktionsschema und Parameter des Stellwertgenerators

### 5.2.3 FC-Aufruf im Stellgrößenzweig (LMNFC)

#### Anwendung

Fügen Sie eine anwenderspezifische Funktion in den Stellgrößenzweig ein, können Sie die im Regler gebildete Stellgröße PID\_OUTV vor Aufschaltung auf den Ausgang des Reglers einer Signalbehandlung (z. B. einer Signalverzögerung) unterziehen.

#### Die Funktion LMNFC

Aktivieren Sie die Funktion LMNFC durch `LMNFC_ON = TRUE`, wird eine anwenderspezifische Funktion aufgerufen. Über den Parameter `LMNFCNBR` wird die Nummer der verwendeten FC eingegeben.

Der Regler führt einen Aufruf der Anwender-FC durch. Vorhandene Ein-/ Ausgangsparameter der Anwender-FC werden dabei nicht versorgt. Die Datenübertragung müssen Sie deshalb mit S7-AWL programmieren. Nachstehend dafür ein Beispiel:

AWL	Erläuterung
<pre> FUNCTION "Anwender-FC" VAR_TEMP INV:REAL; OUTV:REAL; END_VAR BEGIN L      "Regler_DB".LMNFC_IN T      #INV       //Anwenderfunktion OUTV=f(INV) L      #OUTV T      "Regler_DB".LMNFC_OUT END_FUNCTION </pre>	

Der Wert von `LMNFC_ON` bestimmt, ob an dieser Stelle eine frei programmierte Funktion in Form einer Standard-FC (z. B. ein PT-Glied) in den Stellgrößenzweig eingefügt wird oder ob der Stellwert ohne eine derartige Beeinflussung weiterverarbeitet wird (Bild 2-15).



#### Gefahr

Der Baustein überprüft nicht, ob eine FC vorhanden ist. Ist die FC nicht vorhanden, dann geht die CPU mit einem internen Systemfehler in STOP.

### Parameter der Funktion LMNFC

Der Eingangswert LMNFC\_IN ist ein impliziter Parameter. Dieser ist am Konfigurationswerkzeug an LMNFC\_IN bzw. über den Messpunkt **MP9** zu beobachten.

Der Ausgangswert outv ist ebenfalls ein impliziter Parameter und über das Konfigurationswerkzeug nicht beobachtbar (siehe Bild 2-15).

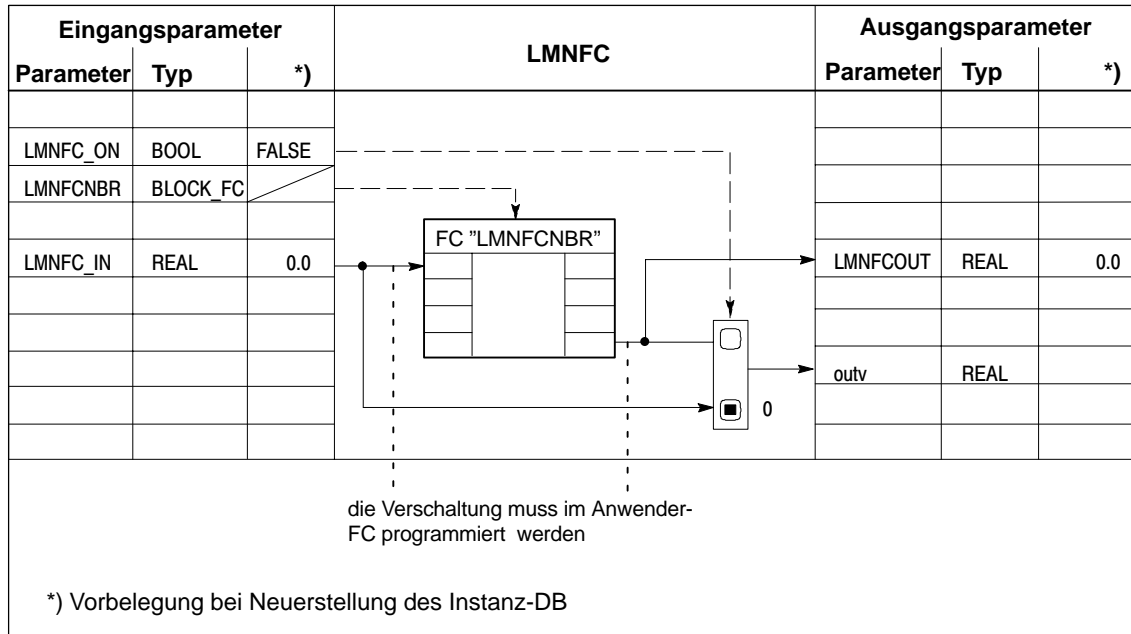


Bild 5-5 FC-Aufruf im Stellwertweig

## 5.2.4 Begrenzung der Änderungsgeschwindigkeit des Stellwertes (LMN\_ROC)

### Anwendung

Rampenfunktionen im Stellgrößenzweig werden verwendet, wenn der Prozess keine sprungförmigen Änderungen des Streckeneingangssignals verträgt. Abrupte Änderungen des Stellwertes sind z. B. zu vermeiden, wenn zwischen geregelter Motor und anzutreibender Last ein Getriebe zwischengeschaltet ist und ein zu schnelles Ansteigen der Motordrehzahl zu einer Überlastung des Getriebes führt.

### Die Funktion LMN\_ROC

Die Funktion LMN\_ROC begrenzt, getrennt für den An- und Abstieg der Stellgröße, die Änderungsgeschwindigkeit der Stellwerte am Ausgang des Reglers. Für den gesamten Wertebereich sind, vom Nullpunkt aus gesehen, zwei Rampen mit steigenden oder fallenden Werten parametrierbar. Die Funktion ist eingeschaltet, wenn LMNRC\_ON = TRUE.

An den beiden Eingängen LMN\_URLM und LMN\_DRLM werden die jeweiligen Begrenzungen für die Steigungen der Rampenfunktion im positiven und negativen Bereich der Stellgröße eingegeben. Die Steigungen beziehen sich auf einen Anstieg bzw. Abfall in Prozent pro Sekunde. Schnellere Stellwertänderungen werden auf diese Grenzgeschwindigkeiten verzögert.

Wird z. B. 'LMN\_URLM' auf 10.0 [%/s] parametrierd, so werden in jedem Abtastzyklus die folgenden Werte zum "Altwert" von outv addiert, und zwar solange  $|inv| > |outv|$  ist:

Abtastzeit 1 s	→	outv <sub>alt</sub> + 10 %
100 ms	→	outv <sub>alt</sub> + 1 %
10 ms	→	outv <sub>alt</sub> + 0,1 %

Bild 5-6 zeigt die Funktionsweise der Signalbehandlung. Aus Sprungfunktionen am Eingang inv (t) werden Rampenfunktionen am Ausgang outv (t).

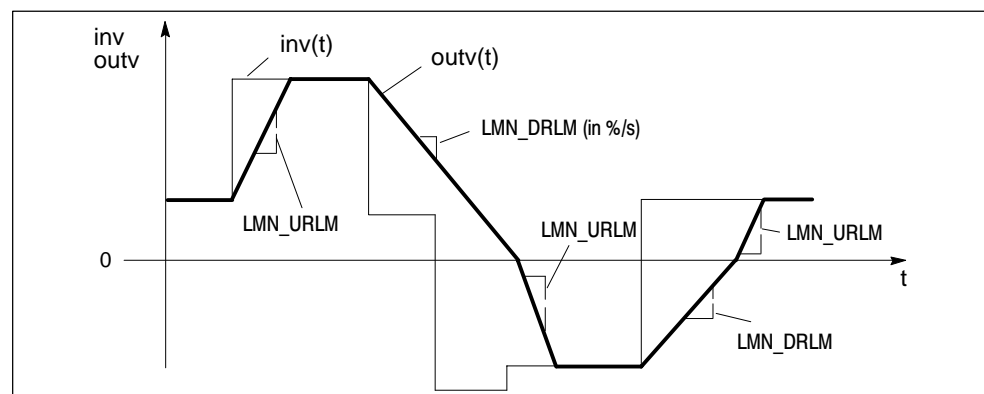


Bild 5-6 Begrenzung der Änderungsgeschwindigkeit der Stellgröße LMN (t)

Das Erreichen der Steigungsbegrenzungen wird nicht gemeldet!

Die Kennzeichnung der Rampen-Parameter erfolgt nach folgendem Schema:

Parameter	Rampe
LMN_URLM	outv  steigend
LMN_DRLM	outv  fallend

### Parameter der Funktion LMN\_ROC

Der Eingangswert inv und der Ausgangswert outv sind implizite Parameter. Sie sind am Konfigurationswerkzeug nicht zu beobachten (Bild 2-15).

Parameter	Bedeutung	Zulässiger Wertebereich
LMN_URLM	Stellwert-Anstiegsbegrenzung	$\geq 0$ [%/s]
LMN_DRLM	Stellwert-Abstiegsbegrenzung	$\geq 0$ [%/s]

Die Steigungswerte (in % pro Sekunde) werden immer positiv eingegeben.

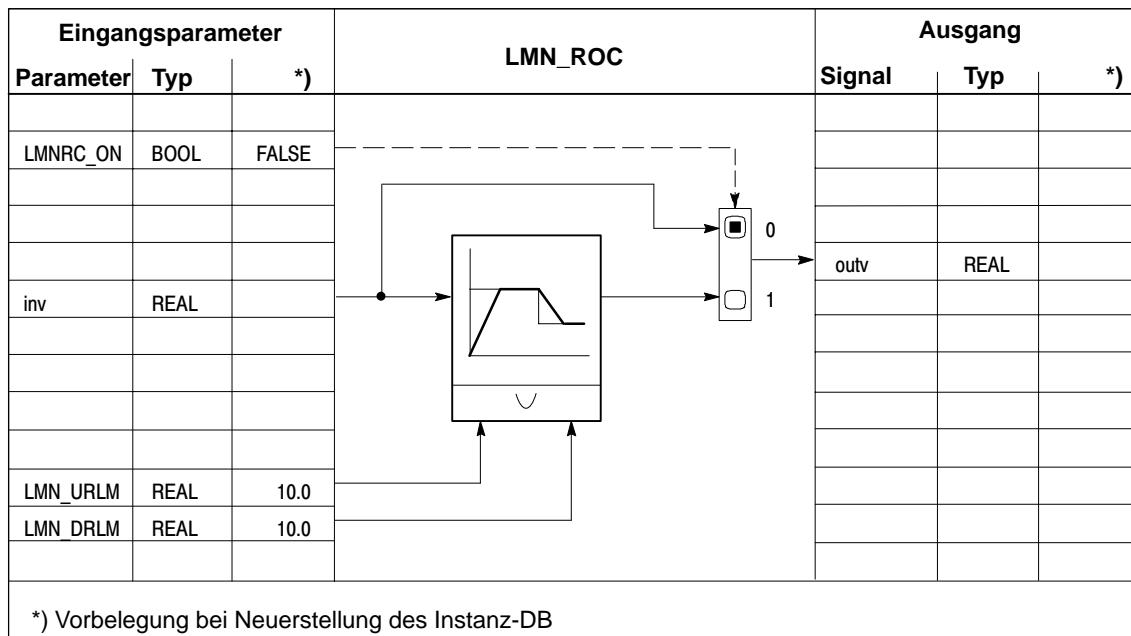


Bild 5-7 Funktionsschema und Parameter der Begrenzung der Stellgrößen-Änderungsgeschwindigkeit

## 5.2.5 Begrenzung des Absolutwertes der Stellgröße (LMNLIMIT)

### Anwendung

Der Arbeitsbereich, d. h. der Bereich, in dem sich das Stellglied im Rahmen zulässiger Stellwerte bewegen kann, wird vom Einstellbereich der Stellwertes bestimmt. Da sich, abhängig von der Art des Stellgliedes, die Grenzen für erlaubte Stellwerte meist nicht mit der 0 %- oder der 100 %- Grenze des Stellbereiches decken, sind oft weitere Bereicheeinschränkungen erforderlich.

Um im jeweiligen Prozess unzulässige Zustände zu vermeiden, wird im Stellwertzweig der Regelung der Einstellbereich für die Stellgröße nach oben und unten begrenzt.

### Die Funktion LMNLIMIT

Die Funktion 'LMNLIMIT' begrenzt die Stellgröße  $LMN(t)$  auf den vorgebbaren unteren und oberen Wert  $LMN\_HLM$  und  $LMN\_LLM$ . Die Eingangsgröße  $inv$  muss jedoch außerhalb dieser Begrenzungen liegen. Da die Funktion nicht abgeschaltet werden kann, muss bei der Konfiguration immer die Angabe einer Unter- und Obergrenze berücksichtigt werden.

Die Zahlenwerte der Grenzen (in Prozent) werden an den Eingangsparametern für die untere und obere Begrenzung eingestellt. Bei Überschreitungen durch die Eingangsgröße  $inv(t)$  werden die zugehörigen Anzeigen über die Meldeausgänge (Bild 2-15) ausgegeben.

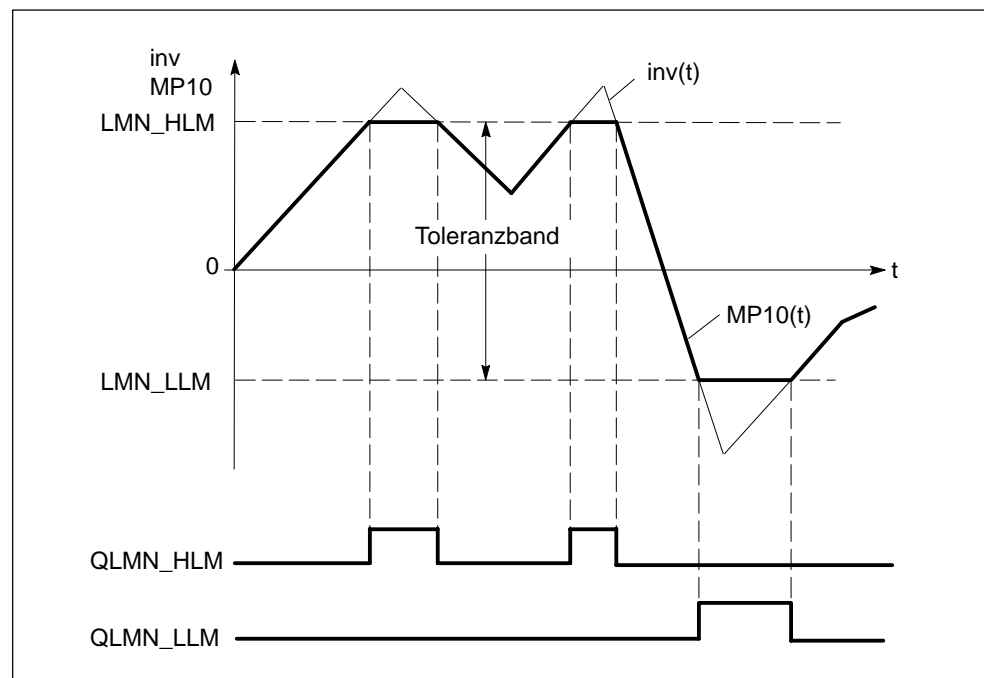


Bild 5-8 Absolutwertbegrenzung der Stellgröße  $LMN(t) = MP10(t)$

### Anlauf und Betriebsweise

- Bei Neustart werden alle Meldeausgänge nullgesetzt.
- Die Begrenzung arbeitet gemäß folgenden Beziehungen:

LMN =	QLMN_HLM	QLMN_LLM	wenn:
LMN_HLM	TRUE	FALSE	$inv \geq LMN\_HLM$
LMN_LLM	FALSE	TRUE	$inv \leq LMN\_LLM$
inv	FALSE	FALSE	$LMN\_HLM < inv < LMN\_LLM$

Am Ausgang d. h. am Parameter LMN bzw. am Messpunkt MP10 wird der wirk-same Stellwert der Regelung angezeigt.

### Parameter der Funktion LMNLIMIT

Der Eingangswert inv ist ein impliziter Parameter. Er ist am Konfigurationswerk-zeug nur über den Parameter LMNFC\_IN oder über den Messpunkt **MP9** zugäng-lich.

Für ein sinnvolles Arbeiten der Begrenzungsfunktion muss gelten:

$$LMN\_HLM > LMN\_LLM$$

Parameter	Bedeutung	Zulässiger Wertebereich
LMN_HLM	Obere Begrenzung der Stell-größe	LMN_LLM bis 100.0 [%]
LMN_LLM	Untere Begrenzung der Stell-größe	-100.0 bis LMN_HLM [%]

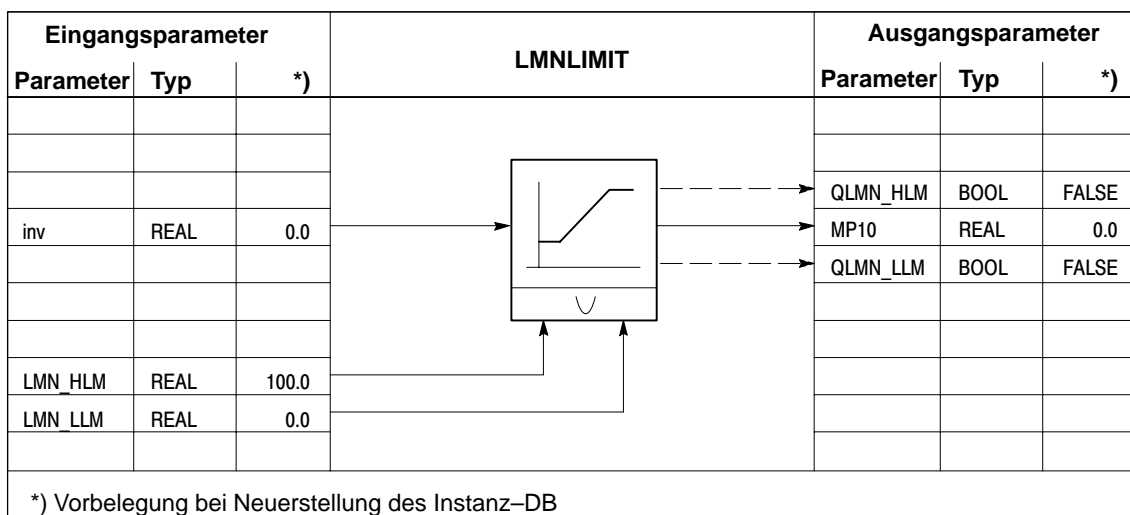


Bild 5-9 Funktionsschema und Parameter der Absolutwert-Begrenzung des Stellwertes



## 5.2.6 Normierung der Stellgröße auf das Format einer physikalischen Größe (LMN\_NORM)

### Anwendung

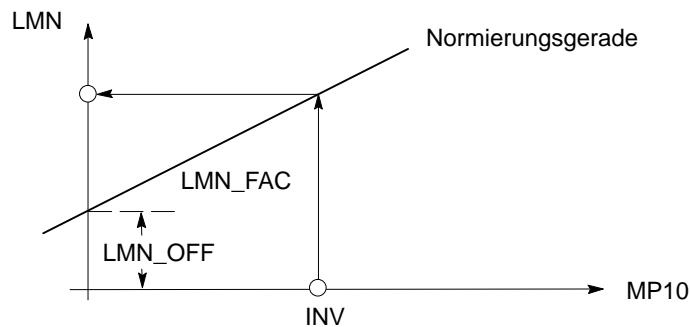
Wird am Streckeneingang eine physikalische Größe als Stellgröße gefordert, dann müssen die Gleitpunktwerte im Bereich 0 bis 100 % auf den physikalischen Bereich (z. B. 150 bis 3000 U/min) der Stellgröße denormiert werden.

### Die Funktion LMN\_NORM

Die Funktion LMN\_NORM denormiert die analoge Ausgangsgröße des Reglers. Es wird der analoge Stellwert an der Normierungsgeraden in die Ausgangsgröße LMN überführt. Der Ausgangswert ist am Konfigurationswerkzeug am Parameter **LMN** zu beobachten.

Zur eindeutigen Festlegung der Normierungsgeraden interne Prozentwerte (im REAL-Format)  $\Rightarrow$  externe physikalische Werte sind zwei Parameter zu definieren:

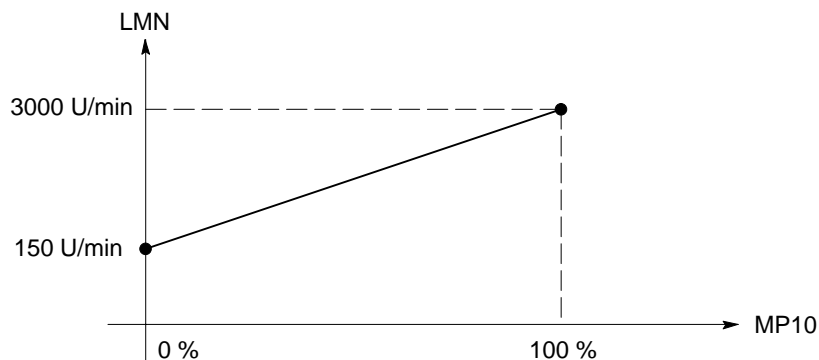
- der Faktor (für die Steigung): **LMN\_FAC**
- der Offset der Normierungsgeraden im Nullpunkt: **LMN\_OFF**



Der Normierungswert wird gemäß folgender Funktion aus dem jeweiligen Eingangswert MP10 berechnet:

$$LMN = MP10 * LMN\_FAC + LMN\_OFF$$

Für das oben genannte Beispiel gilt:



$$LMNFAC = \frac{(3000 - 150)U/min}{(100 - 0)\%} = 28,5 \frac{U/min}{\%}$$

$$LMNOFF = 150U/min$$

Funktionsintern werden keine Werte begrenzt, eine Prüfung der Parameter findet nicht statt.

### Parameter der Funktion LMN\_NORM

Der Ausgang ist ein impliziter Parameter und am Konfigurationswerkzeug an **LMN** zu beobachten (Bild 2-15).

Um die Steigung der Abbildungsgeraden zur Denormierung auf die physikalische Größe am Ausgang LMN zu definieren, kann der Parameter LMN\_FAC im gesamten technischen Wertebereich gewählt werden.

Parameter	Bedeutung	Zulässiger Wertebereich
LMN_FAC	Stellwertfaktor (Steigung der Normierungsgeraden)	gesamter Wertebereich (dimensionslos)
LMN_OFF	Stellwert-Offset	techn. Wertebereich (physikalische Größe)

Eingangsparameter			LMN_NORM	Ausgang		
Parameter	Typ	*)		Signal	Typ	*)
MP10	REAL	0.0		LMN	REAL	0.0
LMN_FAC	REAL	1.0				
LMN_OFF	REAL	0.0				
*) Vorbelegung bei Neuerstellung des Instanz-DB						

Bild 5-10 Funktionsschema und Parameter für die Stellwert-Normierung auf eine physikalische Größe

## 5.2.7 Stellwertausgabe im Peripherieformat (CRP\_OUT)

### Anwendung

Wird der Stellwert an eine Analogausgabebaugruppe übergeben, dann muss der im Gleitpunktformat (Prozentbereich) vorliegende Zahlenwert der internen Stellgröße auf den Zahlenwert des mit dem Ausgang LMN\_PER verschalteten Datenwortes umgesetzt werden. Diese Aufgabe übernimmt die Funktion **CRP\_OUT**.

### Die Funktion CRP\_OUT

Die Funktion CRP\_OUT setzt den am Eingang LMN anliegenden Gleitpunktwert der Stellgröße in einen auf das Peripherieformat denormierten Wert um. Eine Prüfung auf positiven/negativen Überlauf und auf Erreichen des Über-/Untersteuerungsbereiches wird dabei nicht durchgeführt. Baugruppentypen werden nicht berücksichtigt.

Folgende Tabelle gibt einen Überblick über Bereiche und Zahlenwerte vor und nach der Bearbeitung durch den Umform- und Normierungsalgorithmus der Funktion CRP\_OUT.

Stellwert LMN in %	Peripheriewert LMN_PER
118,515	32767
100,000	27648
0,003617	1
0,000	0
-0,003617	-1
-100,000	-27648
-118,519	-32768

### Parameter der Funktion CRP\_OUT

Der Eingangswert ist ein impliziter Parameter im Gleitpunktformat. Dieser ist am Konfigurationswerkzeug über den Ausgang LMN beobachtbar.

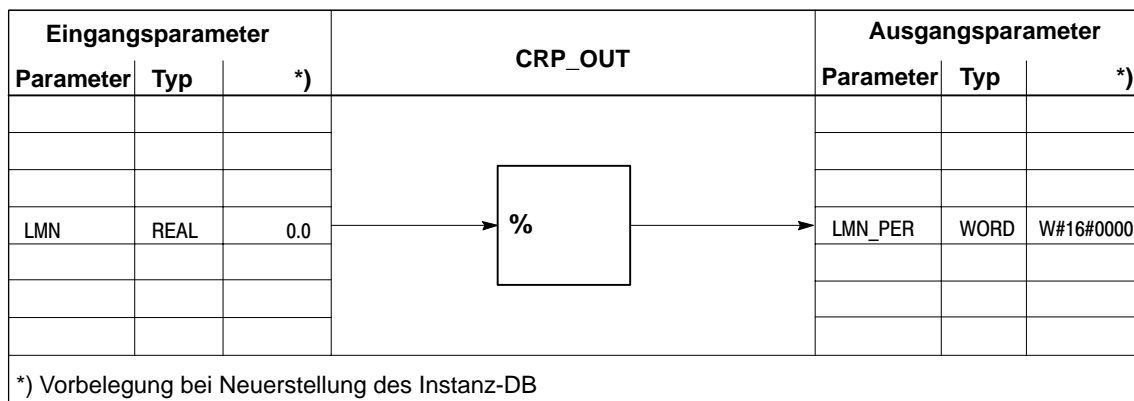


Bild 5-11 Funktionsschema und Parameter der Stellgrößen-Denormierung auf das Peripherieformat

## 5.2.8 Stellwertbeeinflussung über das Konfigurationswerkzeug

### LMN-Anzeige und -Einstellung im Kreisbild

Das Konfigurationswerkzeug hat eine eigene Schnittstelle zum Regler-FB. Es ist deshalb jederzeit möglich, (z. B. zu Testzwecken von einem PG/PC aus auf dem das Konfigurationswerkzeug geladen ist) den Stellgrößenzweig zu unterbrechen und eigene Stellwerte LMN\_OP vorzugeben (Bild 5-12).

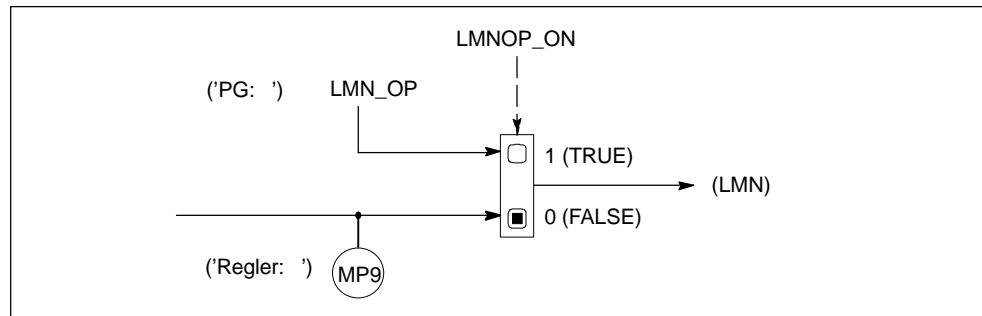


Bild 5-12 Eingriff in den Stellwertzweig über das Konfigurationswerkzeug

Im Fenster des **Kreisbildes** steht dafür – bezeichnet mit **Stellwert** – eines der drei identischen Bedienfelder zur Verfügung. Hier wird oben ('Regler: ') der aktuelle am Messpunkt MP9 anliegende Stellwert ausgegeben. Im Feld darunter ('PG: ') wird der Parameter LMN\_OP angezeigt und bedient.

### Umschalten auf Stellwertvorgabe durch das Konfigurationswerkzeug

Steht der Schalter im Konfigurationswerkzeug auf 'PG: ', dann wird im Regler-FB das Schaltsignal des Strukturschalters LMNOP\_ON auf TRUE gesetzt und LMN\_OP wird auf den Stellwert LMN durchgeschaltet.

Wenn im Stellgrößenzweig die Steigungsbegrenzung LMN\_ROC eingeschaltet ist, kann stoßfrei zwischen den Schalterstellungen 'PG: ' und 'Regler: ' umgeschaltet werden. Der Wert, auf den zurückgeschaltet wird (MP9), kann im Anzeigefeld 'Regler: ' des **Kreisbildes** abgelesen werden. Nach Maßgabe der an LMN\_ROC eingestellten Rampensteigung wird LMN danach auf diesen Wert zugeführt.

Diese Bedieneingriffe wirken erst dann auf den Prozess, wenn sie durch Betätigen der Taste 'Senden' im **Kreisbild** zum Automatisierungssystem übertragen werden.

## 5.3 Kontinuierliche Regler in Kaskadenschaltungen

### Auftrennung der Reglerkaskade

In einer Kaskade stehen mehrere Regler direkt miteinander in Beziehung. Sie müssen daher Vorkehrungen treffen, damit nach Auftrennung der Kaskadenstruktur an einer beliebigen Stelle der Kaskadenbetrieb wieder problemlos aufgenommen werden kann.

In Kaskadenregelungen wird deshalb in den unterlagerten Reglern über eine Oder-Logik aus den Zustandssignalen der Schalter im Sollwert- und Stellgrößenzweig ein Signal QCAS gebildet. Dieses Signal betätigt in den überlagerten Reglern einen Schalter, der die Regler in den Nachführbetrieb überführt. Die Nachführgröße ist dabei immer der Sollwert des unterlagerten Kreises (Bild 5-13).

Die Umschaltung vom Nachführ- in den Automatikbetrieb erfolgt stoßfrei wie bei der Hand- Automatikumschaltung.

Der kontinuierliche Regler (PID\_CP) kann in überlagerten Regelkreisen als Führungsregler und in unterlagerten Regelkreisen als Folgeregler eingesetzt werden.

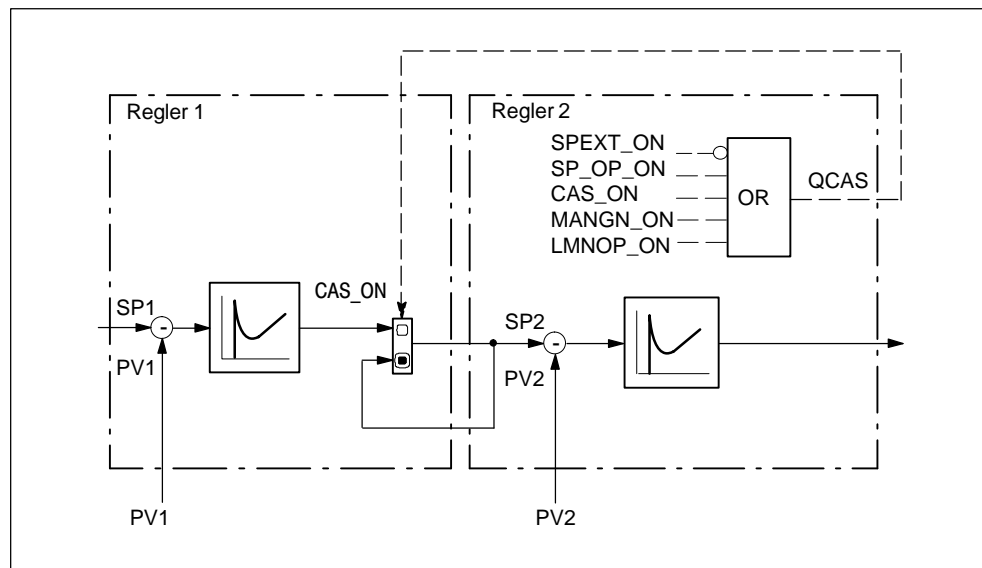


Bild 5-13 Zweischleifige Kaskadenregelung

### Hinweis

Die Verschaltung vom Stellwert des Führungsreglers LMN muss immer auf den externen Sollwert SP\_EXT des Folgereglers gehen.

### Bausteinverschaltung

Nachstehendes Bild zeigt das Prinzip der Regler- bzw. Bausteinverschaltung in mehrschleifigen Kaskaden.

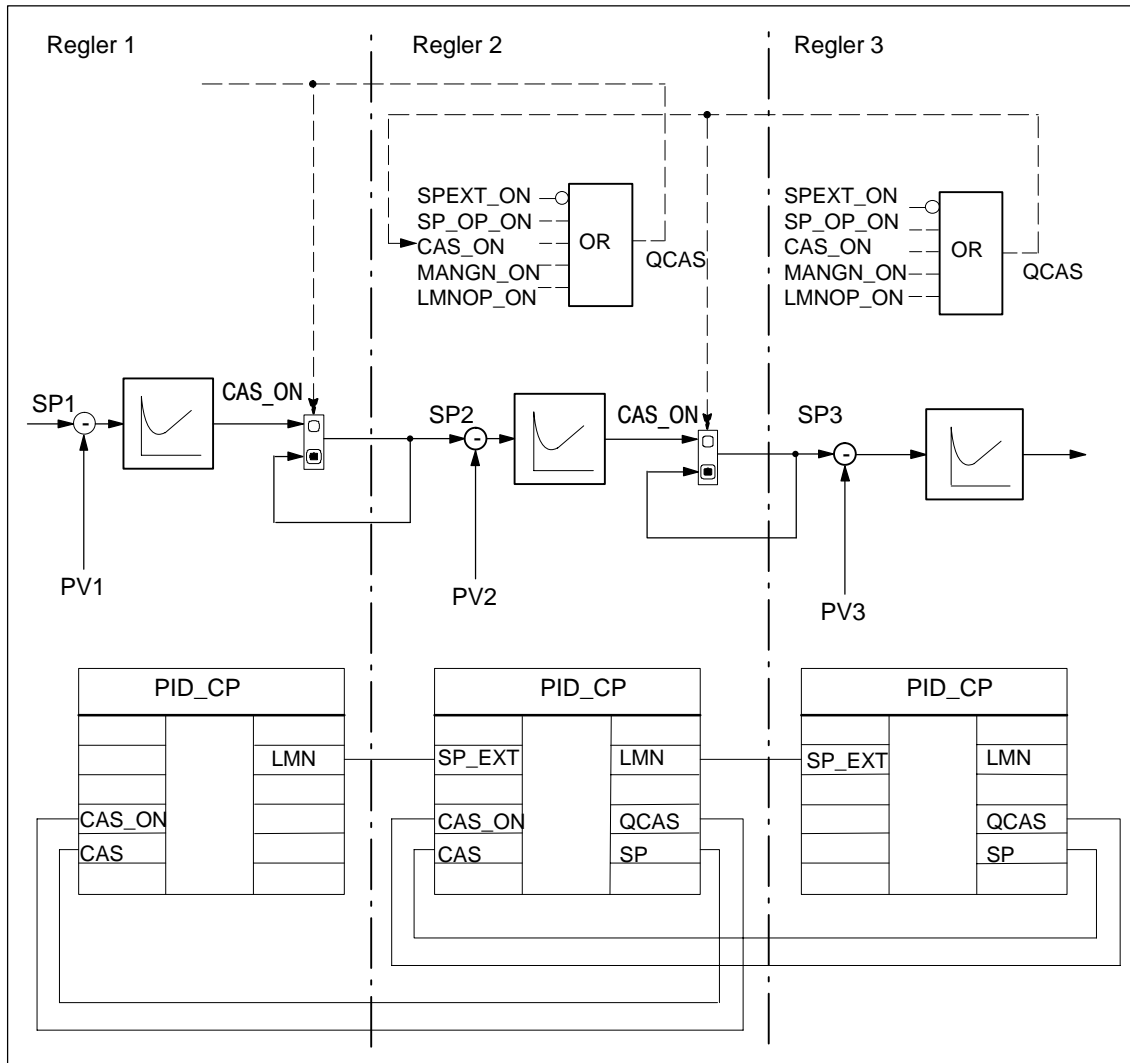


Bild 5-14 Verschaltung einer Kaskade mit zwei unterlagerten Regelkreisen

## 5.4 Impulsformerstufe (PULSEGEN)

### Anwendung

Die Impulsformerstufe des Standard-FB "PID\_CP" generiert die Impulsausgabe eines kontinuierlichen Reglers, um über Impulse bei Standard PID Control proportionale Stellglieder ansteuern zu können. Es lassen sich damit PID-Zweipunkt- und -Dreipunktregler mit Pulsbreitenmodulation aufbauen.

### Der Impulsformer

Die Impulsformerstufe des Standard-FB "PIC\_CP" transformiert die Eingangsgröße "Stellwert des PID-Reglers am Messpunkt MP10" durch Modulation der Impulsbreite in eine Impulsfolge mit konstanter Periodendauer, die in PER\_TM parametrisiert werden muss.

Die Dauer eines Impulses pro Periodendauer ist proportional der Eingangsgröße. Dabei ist der durch PER\_TM parametrisierte Zyklus nicht identisch mit dem Bearbeitungszyklus des Impulsformers. Vielmehr setzt sich ein Zyklus "PER\_TM" aus mehreren Bearbeitungszyklen des Impulsformers zusammen, wobei die Anzahl der Impulsformeraufrufe pro PER\_TM-Zyklus ein Maß für die Genauigkeit der Impulsbreite darstellt.

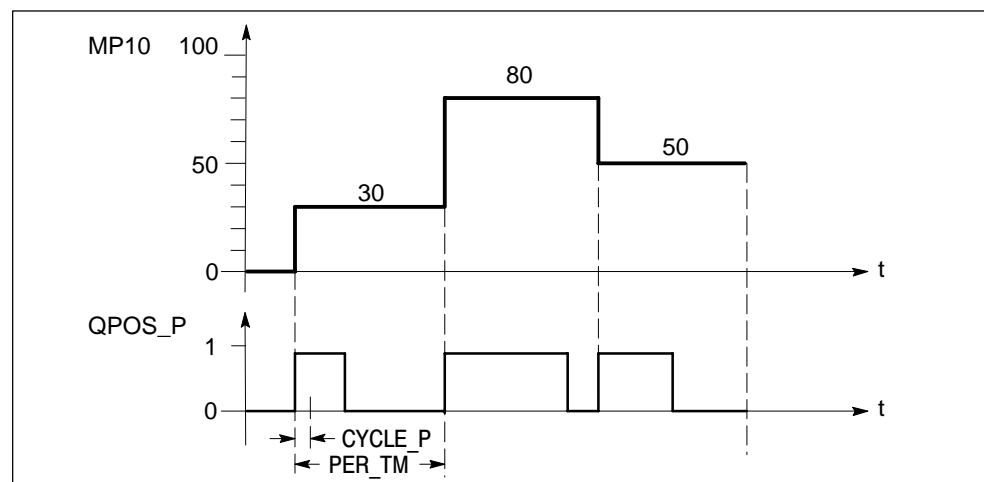


Bild 5-15 Impulsbreitenmodulation

Eine Eingangsgröße von 30 % und 10 Aufrufe des Impulsformers pro PER\_TM bedeuten also:

- "Eins" am Ausgang QPOS für die ersten drei Aufrufe des Impulsformers (30 % von 10 Aufrufen),
- "Null" am Ausgang QPOS für sieben weitere Aufrufe des Impulsformers (70 % von 10 Aufrufen).

## Reglerabtastzeit CYCLE und Impulsrasterbreite CYCLE\_P

Wenn Sie die Impulsformerstufe eingeschaltet haben (PULSE\_ON = TRUE), dann müssen Sie am Eingang CYCLE\_P den Zeittakt des aufrufenden Weckalarm-OB vorgeben. Die Dauer des erzeugten Impulses beträgt stets ein ganzzahliges Vielfaches dieses Wertes.

Am Eingang CYCLE geben Sie die Abtastzeit für die übrigen Regelungsfunktionen des PID\_CP vor. Der Funktionsbaustein PID\_CP ermittelt die Zeitumsetzung und bearbeitet die Regelungsfunktionen mit der Abtastzeit CYCLE.

Dabei müssen Sie darauf achten, dass CYCLE ein ganzzahliges Vielfaches von CYCLE\_P ist. Falls Sie diese Bedingung nicht einhalten, rundet der Funktionsbaustein PID\_CP die Abtastzeit für die Regelungsfunktionen auf ein ganzzahliges Vielfaches von CYCLE\_P auf. Die zeitabhängigen Funktionen (z. B. Glättung, Integration, Differentiation) arbeiten dann nicht ganz korrekt.

CYCLE kann kleiner gewählt werden als die Periodendauer PER\_TM\_P bzw. PER\_TM\_N. Dies ist dann sinnvoll, wenn einerseits eine möglichst große Periodendauer gewünscht wird, um die Stellglieder zu schonen, und andererseits die Abtastzeit aufgrund einer schnellen Regelstrecke klein sein muss.

Als sinnvoller Wert für die Abtastzeit CYCLE gilt wie beim kontinuierlichen Regler ohne Impulsformerstufe, dass CYCLE nicht kleiner werden darf als ca. 10% der dominierenden Streckenzeitkonstanten des geregelten Systems.

Beispiel für die Wirkung der Parameter CYCLE\_P, CYCLE und PER\_TM\_P bzw. PER\_TM\_N:

PER\_TM\_P = 10 s, CYCLE = 1 s, CYCLE\_P = 100 ms.

Jede Sekunde wird ein neuer Stellwert berechnet, alle 100 ms erfolgt der Vergleich des Stellwertes mit der bislang ausgegebenen Impulslänge bzw. Pausenlänge.

Wenn ein Impuls ausgegeben wird und der berechnete Stellwert größer ist als die bisherige Impulslänge / PER\_TM\_P, wird der Impuls verlängert. Andernfalls wird kein Impulssignal mehr ausgegeben. Wenn kein Impuls ausgegeben wird und  $(100\% - \text{berechneter Stellwert})$  größer ist als die bisherige Pausenlänge / PER\_TM\_P, wird die Pause verlängert. Andernfalls wird ein Impulssignal ausgegeben.

Aufgrund eines besonderen Verfahrens der Pulsformung bewirkt eine Vergrößerung bzw. Verkleinerung der Stellgröße während der Periode eine Verlängerung bzw. Verkürzung des ausgegebenen Impulses. Ist in diesem Fall (CYCLE < PER\_TM\_P) die Periodendauer so groß parametrisiert, dass sie zum Schwingen des Istwertes führen würde, dann wird die wirksame Periodendauer vom Funktionsbaustein PID\_CP selbständig auf einen sinnvollen Wert reduziert.

## Genauigkeit der Pulsformung

Je kleiner die Impulsrasterbreite CYCLE\_P gegenüber der Periodendauer PER\_TM\_P (bzw. PER\_TM\_N) ist, desto genauer ist die Pulsbreitenmodulation. Für eine hinreichend genaue Regelung sollte folgende Bedingung erfüllt sein:

$$\text{CYCLE\_P} \leq \frac{\text{PER\_TM}}{20 \dots 50}$$



## Realisierung von sehr kurzen Impulsrasterbreiten

Bei einer schnellen Regelstrecke sind sehr kleine Impulsrasterbreiten (z. B. 10 ms) notwendig. Aufgrund der Programmlaufzeit ist es in diesem Fall nicht sinnvoll, die Regelungsteile im selben Weckalarm-OB zu bearbeiten wie die Berechnung der Impulsausgabe. Die Bearbeitung der Regelungsfunktionen verlagern Sie dann entweder in den OB 1 oder in einen langsameren Weckalarm-OB (Die Bearbeitung der Regelungsfunktion im OB 1 ist nur dann sinnvoll, wenn die Zykluszeit des OB 1 deutlich kleiner ist als die Abtastzeit CYCLE des Reglers.)

Über den Parameter SELECT bestimmen Sie, welcher Programmteil bearbeitet werden soll. Die folgende Tabelle verschafft Ihnen einen Überblick über die Parametrierung des Eingangsparameters SELECT:

SELECT	Verwendete Funktionalität des Bausteins	Zugrundeliegende Methode
0	Regelungsteil und Impulsausgabe	Regelungsteil und Impulsausgabe in ein und demselben Baustein
1	Aufruf im OB 1 (Regelungsteil)	Regelungsteil im OB 1, Impulsausgabe in schnellem Weckalarm-OB
2	Aufruf in Weckalarm-OB (Impulsausgabe)	
3	Aufruf in langsamem Weckalarm-OB (Regelungsteil)	Regelungsteil in langsamem Weckalarm-OB, Impulsausgabe in schnellem Weckalarm-OB
2	Aufruf in schnellem Weckalarm-OB (Impulsausgabe)	

Im Folgenden werden die in der obigen Tabelle angedeuteten Methoden zur Realisierung sehr kurzer Impulsrasterbreiten näher erklärt.

- Regelungsfunktion im OB 1, Impulsausgabe in Weckalarm-OB

Beim Aufruf des FB "PID\_CP" mit SELECT = 2 erfolgt die Berechnung der Impulsausgabe und die Prüfung, ob seit der letzten Bearbeitung des Regelungsteils die an CYCLE parametrierte Abtastzeit schon abgelaufen ist. Falls diese Abtastzeit abgelaufen ist, beschreibt der FB die Variable QC\_ACT im Instanz-DB mit dem Wert TRUE.

Beim Aufruf des FB "PID\_CP" mit SELECT = 1 erfolgt die Bewertung der Variablen QC\_ACT im Instanz-DB: Wenn QC\_ACT den Wert FALSE hat, wird der Baustein sofort wieder beendet; er hat also nur eine sehr kleine Laufzeit benötigt. Wenn QC\_ACT den Wert TRUE hat, wird der Regelungsteil einmalig durchlaufen, und anschließend setzt der FB QC\_ACT wieder zurück.

Diese Vorgehensweise bewirkt, dass die Abtastzeit für die Regelungsfunktion des FB "PID\_CP" nicht exakt eingehalten werden kann. Sie schwankt um die Laufzeit des OB 1 (einschließlich aller Unterbrechungen). Dieses Verfahren ist daher nur geeignet, wenn die Laufzeit des OB 1 klein ist gegenüber der Abtastzeit CYCLE.

- Regelungsfunktion in langsamem, Impulsausgabe in schnellem Weckalarm-OB

Beim Aufruf des FB "PID\_CP" mit SELECT = 2 erfolgt grundsätzlich die Berechnung der Impulsausgabe

Beim Aufruf des FB "PID\_CP" mit SELECT = 3 erfolgt stets die Bearbeitung des Regelungsteils.

### Hinweis

Damit der Aufruf des FB "PID\_CP" mit seinen vielen Formaloperanden nicht zweimal ausprogrammiert werden muss, empfiehlt es sich, diesen einmalig in einer FC zu programmieren, die als Eingangsparameter ebenfalls einen Parameter SELECT hat. Diesen Eingangsparameter verschalten sie auf den SELECT-Eingang des FB "PID\_CP". Im OB 1 bzw. im Weckalarm-OB rufen Sie dann nur diese FC auf.

Diese Vorgehensweise ist änderungsfreundlich und spart Programmspeicher.

### Betriebsarten des Reglers mit Impulsausgang

Je nach Parametrierung des Impulsformers können PID-Regler mit Dreipunktverhalten oder mit bipolarem bzw. unipolarem Zweipunktausgang konfiguriert werden. Nachstehende Tabelle zeigt die Einstellung der Schalterkombinationen für die möglichen Betriebsarten.

Betriebsart \ Schalter	MAN_ON	STEP3_ON	ST2BI_ON
Dreipunktregelung	FALSE	TRUE	beliebig
Zweipunktregelung mit bipolarem Stellbereich (-100 % bis 100 %)	FALSE	FALSE	TRUE
Zweipunktregelung mit unipolarem Stellbereich (0 % bis 100 %)	FALSE	FALSE	FALSE
Handbetrieb	TRUE	beliebig	beliebig

### Dreipunktregelung

In der Betriebsart "Dreipunktregelung" können drei Zustände des Stellsignals erzeugt werden, z. B. je nach Stellglied und Prozess: Mehr - Aus - Weniger, Vorwärts - Stop - Rückwärts, Heizen - Aus - Kühlen usw. Nach Anforderung des zu regelnden Prozesses werden die Zustandswerte der binären Ausgangssignale QPOS\_P und QNEG\_P den jeweiligen Betriebszuständen des Stellgliedes zugeordnet. Die Tabelle zeigt zwei Beispiele.

	heizen vorwärts	aus stop	kühlen rückwärts
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

Eine Dimensionierung der Mindestimpuls- bzw. Mindestpausendauer P\_B\_TM\_P kann sehr kurze Ein- oder Ausschaltzeiten verhindern, die die Lebensdauer von Schaltgliedern und Stelleinrichtungen beeinträchtigen. (Bild 5-16). Dabei wird der proportionalen Ausgangskennlinie, über die die Impulsdauer berechnet wird, ein Ansprechwert für die Impulsausgabe überlagert.

**Hinweis**

Kleine Absolutwerte der Eingangsgröße "Stellwert des PID-Reglers am Messpunkt MP10", die eine Impulsdauer kleiner als  $P\_B\_TM\_P$  erzeugen würden, werden unterdrückt. Für große Eingangswerte, die eine Impulsdauer größer als  $PER\_TM\_P - P\_B\_TM\_P$  erzeugen würden, wird die Impulsdauer auf 100 % bzw. -100 % gesetzt.

Es werden Einstellwerte  $P\_B\_TM\_P \leq 0,1 * PER\_TM\_P$  empfohlen.

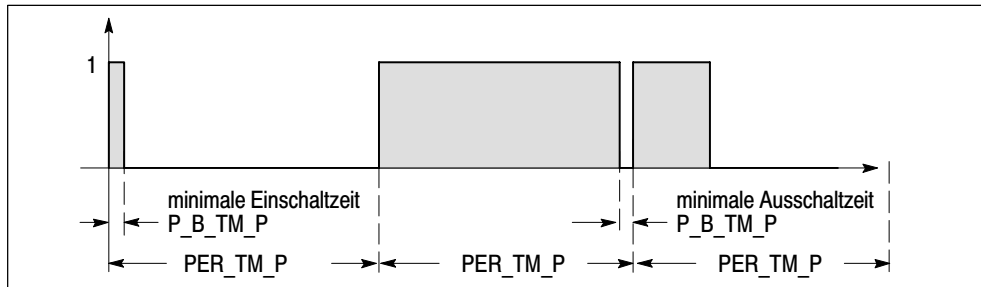


Bild 5-16 Schaltverhalten des Impulsausgangs

Die Dauer der positiven oder negativen Impulse errechnet sich aus der Eingangsgröße "Stellwert des PID-Reglers am Messpunkt MP10" (in %) mal Periodendauer:

$$\text{Impulsdauer} = \frac{MP10}{100} * PER\_TM\_P[s]$$

Durch die Unterdrückung von Mindestimpuls- bzw. -Pausendauer erhält die Umformkennlinie Knickpunkte im Anfangs- und Endbereich (Bild 5-17).

Die zuvor getroffenen Aussagen gelten ebenso für  $P\_B\_TM\_N$  und  $PER\_TM\_N$  (siehe Bild 5-17).

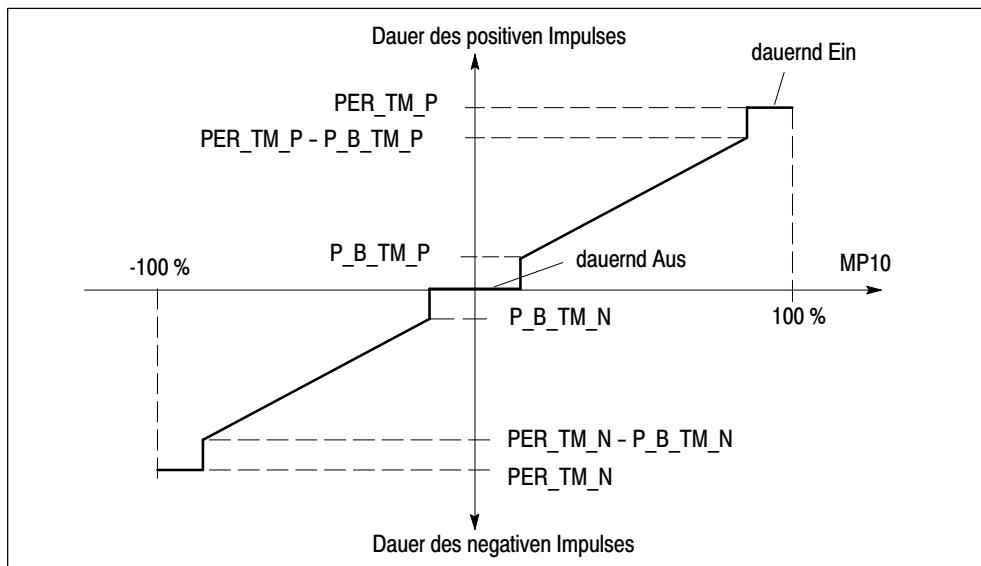


Bild 5-17 Symmetrische Kennlinie des Dreipunktreglers (Verhältnissfaktor = 1)

### Dreipunktregelung unsymmetrisch

Über den Verhältnissfaktor **RATIOFAC** kann das Verhältnis der Dauer von positiven zu negativen Impulsen verändert werden. Bei einem thermischen Prozess lassen sich damit z. B. unterschiedliche Streckenzeitkonstanten für Heizen und Kühlen berücksichtigen.

Soll bei gleichem Absolutwert "Stellwert des PID-Reglers am Messpunkt MP10" die Impulsdauer am negativen Impulsausgang gegenüber dem positiven Impuls verkürzt werden, so ist ein Verhältnissfaktor kleiner 1 einzustellen (Bild 5-18):

pos. Impuls > neg. Impuls:  $RATIOFAC < 1$

Impulsdauer negativ:  $\frac{MP10}{100} * PER\_TM\_N * RATIOFAC$

Impulsdauer positiv:  $\frac{MP10}{100} * PER\_TM\_P$

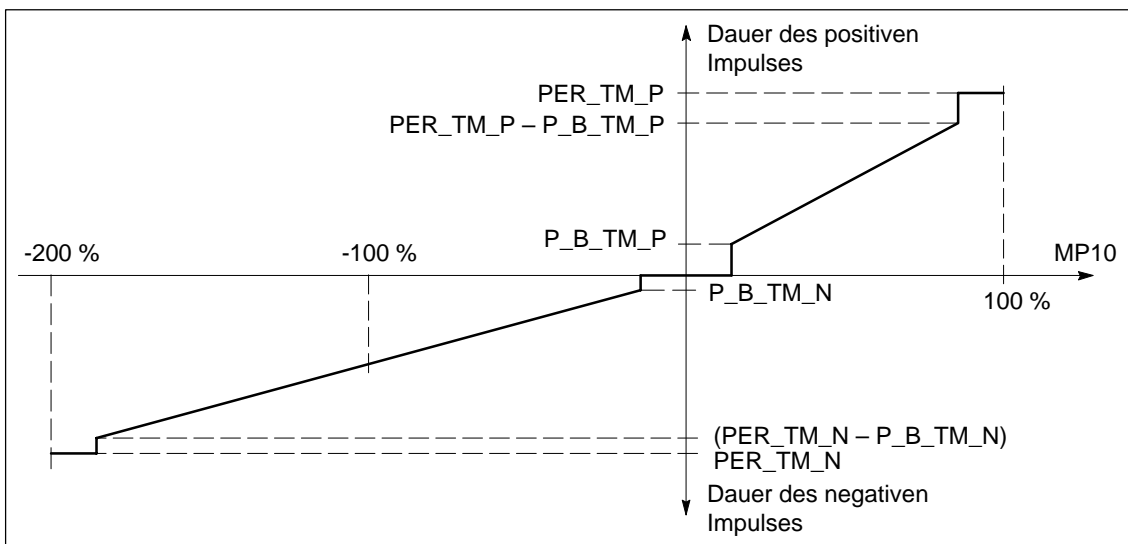


Bild 5-18 Unsymmetrische Kennlinie des Dreipunktreglers (Verhältnissfaktor = 0.5)

Soll umgekehrt bei gleichem Absolutwert  $|MP10|$  die Impulsdauer am positiven Impulsausgang gegenüber dem negativen Impuls verkürzt werden, so ist ein Verhältnissfaktor größer 1 einzustellen:

pos. Impuls < neg. Impuls:  $RATIOFAC > 1$

Impulsdauer negativ:  $\frac{MP10}{100} * PER\_TM\_N$

Impulsdauer positiv:  $\frac{MP10 * PER\_TM\_P}{100 * RATIOFAC}$

Rechnerisch bedeutet dies, dass bei  $RATIOFAC < 1$  der Ansprechwert für negative Impulse mit dem Verhältnissfaktor multipliziert und bei  $RATIOFAC > 1$  der Ansprechwert für positive Impulse durch den Verhältnissfaktor dividiert wird.

**Hinweis**

Bei unsymmetrischer Dreipunktregelung  $RATIOFAC \neq 1$  müssen Sie die Stellwertgrenzen nach folgenden Formeln anpassen:

**RATIOFAC < 1:**

$$LMN\_HLM = 100$$

$$LMN\_LLM = -100 * (1 / RATIOFAC)$$

**RATIOFAC > 1:**

$$LMN\_HLM = 100 * RATIOFAC$$

$$LMN\_LLM = -100$$

Beispiele:

RATIOFAC = 1	RATIOFAC = 0,5	RATIOFAC = 2,0
LMN_HLM = 100	LMN_HLM = 100	LMN_HLM = 200
LMN_LLM = -100	LMN_LLM = -200	LMN_LLM = -100

**Zweipunktregelung**

Bei der Zweipunktregelung wird nur der positive Impulsausgang QPOS\_P von PIC\_CP mit dem betreffenden Ein/Aus-Stellglied verbunden. Je nach genutztem Stellbereich,  $MP10 = -100.0$  bis  $+100.0$  % oder  $MP10 = 0.0$  % bis  $100.0$  %, hat der Zweipunktregler einen bipolaren oder einen unipolaren Stellbereich (Bild 5-19 und Bild 5-20).

Bei unipolarer Betriebsart darf MP10 nur Werte zwischen 0.0 und 100% annehmen.

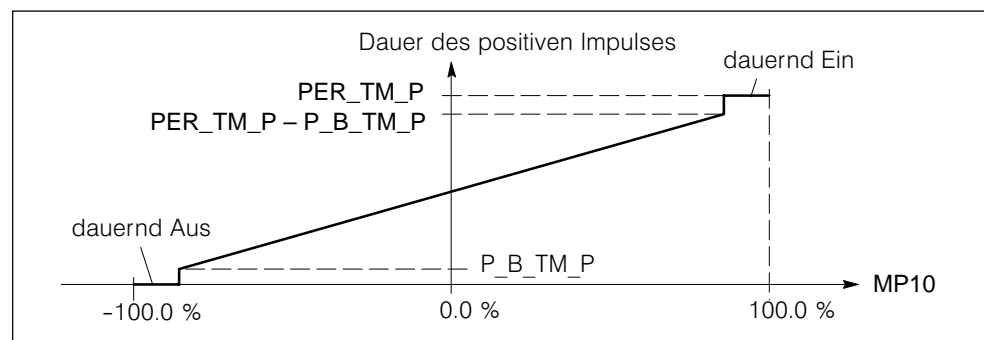


Bild 5-19 Zweipunktregler mit bipolarem Stellbereich (-100 % bis 100 %)

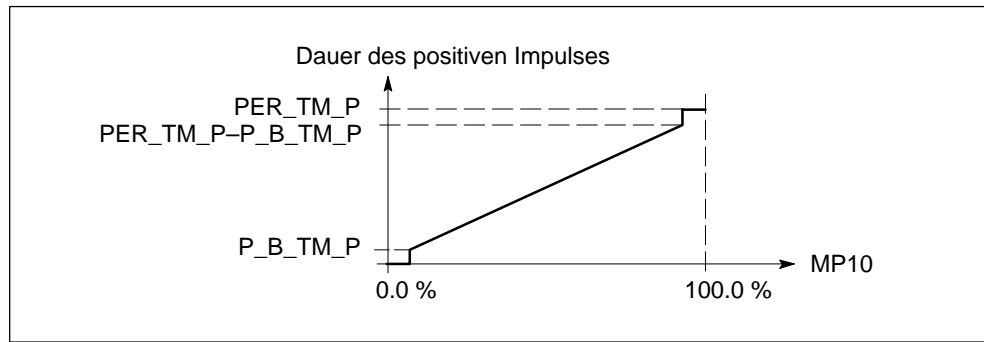


Bild 5-20 Zweipunktregler mit unipolarem Stellbereich (0 % bis 100 %)

An QNEG\_P steht das negierte Ausgangssignal zur Verfügung für den Fall, dass die Verschaltung des Zweipunktreglers im Regelkreis ein logisch invertiertes Binärsignal für die Stellimpulse erfordert.

	Ein	Aus
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

### Parameter der Impulsformerstufe

Die Werte der Eingangsparameter werden am Baustein "PID\_CP" nicht begrenzt. Eine Prüfung der Parameter findet nicht statt.

Bei Neustart werden alle Parameter auf Null gesetzt.

Parameter	Bedeutung	Zulässiger Wertebereich
CYCLE_P	Abtastzeit der Impulsformerstufe	$\geq 1$ ms
SELECT	Auswahlschalter für die im aktuellen Bausteinaufruf zu bearbeitenden Funktionsteile (nur relevant, falls PULS_ON = TRUE) 0 (default): PID und Pulsgenerator 1: PID (Bausteinaufruf im OB 1) 2: Pulsgenerator (Bausteinaufruf in Weckalarm-OB) 3: PID (Bausteinaufruf in Weckalarm-OB)	
QC_ACT	Anzeige, ob beim nächsten Bausteinaufruf der Regelungsteil bearbeitet wird (nur relevant, falls SELECT den Wert 0 oder 1 hat)	
QPOS_P	Pulsgenerator: positiver Puls ein	
QNEG_P	Pulsgenerator: negativer Puls ein	
PULSE_ON	Pulsgenerator ein	
STEP3_ON	Pulsgenerator: Dreipunktregelung einschalten	

Parameter	Bedeutung	Zulässiger Wertebereich
ST2BI_ON	Pulsgenerator: Zweipunktregelung für bipolaren Stellbereich einschalten (für unipolaren Stellbereich muss STEP3_ON = FALSE sein)	
PER_TM_P	Pulsgenerator: Periodendauer des positiven Pulses	
PER_TM_N	Pulsgenerator: Periodendauer des negativen Pulses	
P_B_TM_P	Pulsgenerator: Mindestimpuls- bzw. Mindestpausendauer des positiven Pulses	
P_B_TM_N	Pulsgenerator: Mindestimpuls- bzw. Mindestpausendauer des negativen Pulses	
RATIOFAC	Verhältnissfaktor für unsymmetrische Kennlinien	0.1 bis 10.0 (dimensionslos)

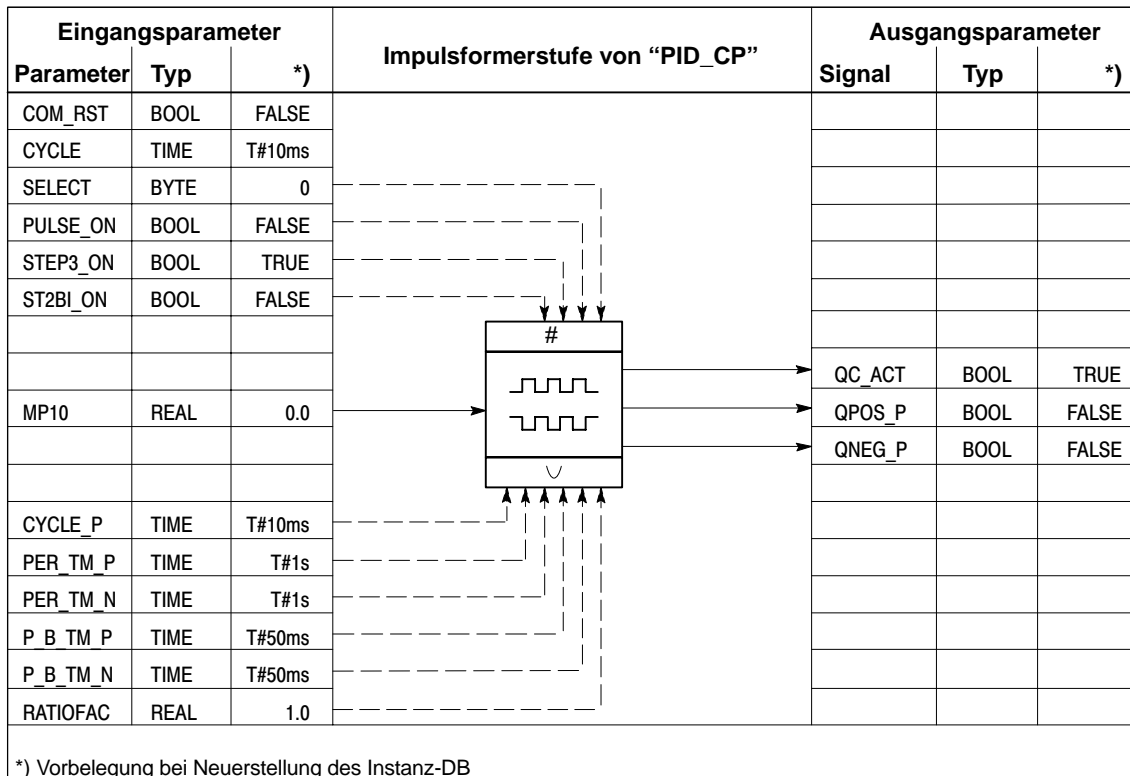


Bild 5-21 Funktionsschema und Parameter des Impulsformers





## Der Schrittreger (PID\_ES)

### 6.1 Regelfunktionen des PID-Schrittregers

#### Der Funktionsbaustein PID\_ES

Neben den Funktionen im Soll- und Istwertzweig realisiert der Funktionsbaustein (FB 2) einen fertigen PID-Regler mit binärem Stellwert-Ausgang. Es ist möglich, den Stellwert von Hand zu beeinflussen. Teilfunktionen lassen sich zu- oder abschalten.

Der FB ermöglicht es, technische Prozesse und Anlagen mit integrierenden Stellgliedern auf den Automatisierungssystemen SIMATIC S7 zu regeln. Der Regler kann als Festwertregler einzeln oder in unterlagerten Regelkreisen bei Kaskaden-, Mischungs- oder Verhältnisregelungen eingesetzt werden, jedoch nicht als Führungsregler.

Die Signalverarbeitung im Soll-/Istwertzweig sowie bei der Behandlung und Überwachung der Regeldifferenz ist identisch mit der des kontinuierlichen Reglers. Die betreffenden Detail-Beschreibungen finden Sie deshalb für beide Reglertypen in *Kapitel 4* dieses Handbuches.

## Normierung der Eingangsgrößen ER und PV

Die Eingangsgrößen ER und PV des PID-Reglers werden vor der Reglerbearbeitung normiert auf den Bereich 0 bis 100 gemäß folgenden Formeln:

- bei ausgeschalteter Wurzelfunktion (SQRT\_ON = FALSE):
  - $ER_{\text{normiert}} = ER * 100.0 / (NM\_PVHR - NM\_PVLR)$
  - $PV_{\text{normiert}} = (PV - NM\_PVLR) * 100.0 / (NM\_PVHR - NM\_PVLR)$
- bei eingeschalteter Wurzelfunktion (SQRT\_ON = TRUE):
  - $ER_{\text{normiert}} = ER * 100.0 / (SQRT\_HR - SQRT\_LR)$
  - $PV_{\text{normiert}} = (PV - SQRT\_LR) * 100.0 / (SQRT\_HR - SQRT\_LR)$

Diese Normierung wird durchgeführt, damit der Verstärkungsfaktor GAIN des PID-Reglers dimensionslos eingegeben werden kann. Bei einer Änderung der Ober- und Untergrenze des Physikalischen Messbereichs (z. B. von bar auf mbar) muss somit der Verstärkungsfaktor nicht geändert werden.

Die normierten Eingangsgrößen  $ER_{\text{normiert}}$  und  $PV_{\text{normiert}}$  sind nicht beobachtbar.

## Funktionsschema des Schrittregers mit Stellungsrückmeldung im Regelkreis

Die Arbeitsweise des Schrittregers mit Stellungsrückmeldung basiert auf dem PID-Regelalgorithmus und wird um die Funktionsglieder zur Erzeugung der binären Ausgangssignale aus dem analogen Stellsignal des Reglers (Bild 6-1) ergänzt.

Das Dreipunktglied formt dabei Abweichungen zwischen Stellgröße und Stellungsrückmeldung je nach Vorzeichen in positive oder negative Impulse des Ausgangssignals um, die z. B. auf einen motorischen Ventilantrieb gegeben werden. Praktisch handelt es sich dabei um eine Kaskadenregelung mit einem unterlagerten Stellungsregelkreis.

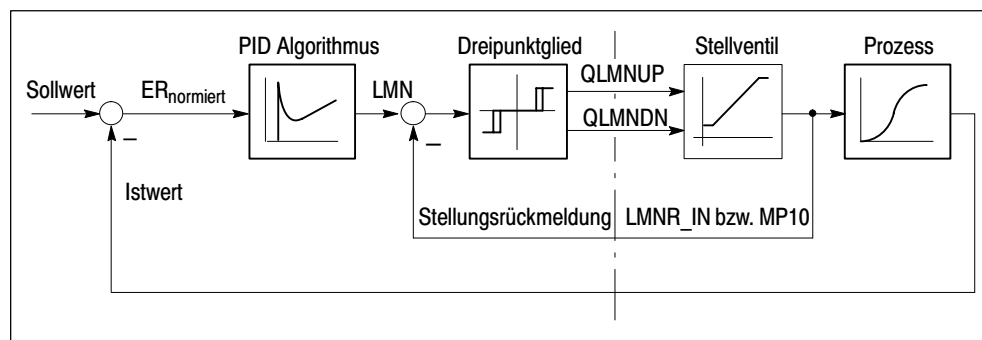


Bild 6-1 Schrittreger mit Stellungsrückmeldung

## Funktionsschema des Schrittregers ohne Stellungsrückmeldung im Regelkreis

Der I-Anteil des Schrittregers ohne Stellungsrückmeldung wird in einem Integrator in der Rückführung berechnet. Das mit dem Reglerausgang des PD-Reglers verglichene Rückführsignal wird hierbei durch Aufintegrieren von folgenden zwei Signalanteilen gebildet:

- Signalkomponente für die simulierte Stellungsrückmeldung:  $\frac{\pm 100.0}{MTR\_TM}$
- Signalkomponente für den I-Anteil:  $(\text{Sollwert} - \text{Istwert})_{\text{normiert}} * \frac{GAIN}{TI}$

Das Rückführsignal ist somit die Differenz zwischen simulierter Stellungsrückmeldung und I-Anteil.

Das Dreipunktglied formt Abweichungen zwischen Stellgröße und Rückführsignal je nach Vorzeichen in positive oder negative Impulse des Ausgangssignals um, die z. B. auf einen motorischen Ventiltrieb gegeben werden.

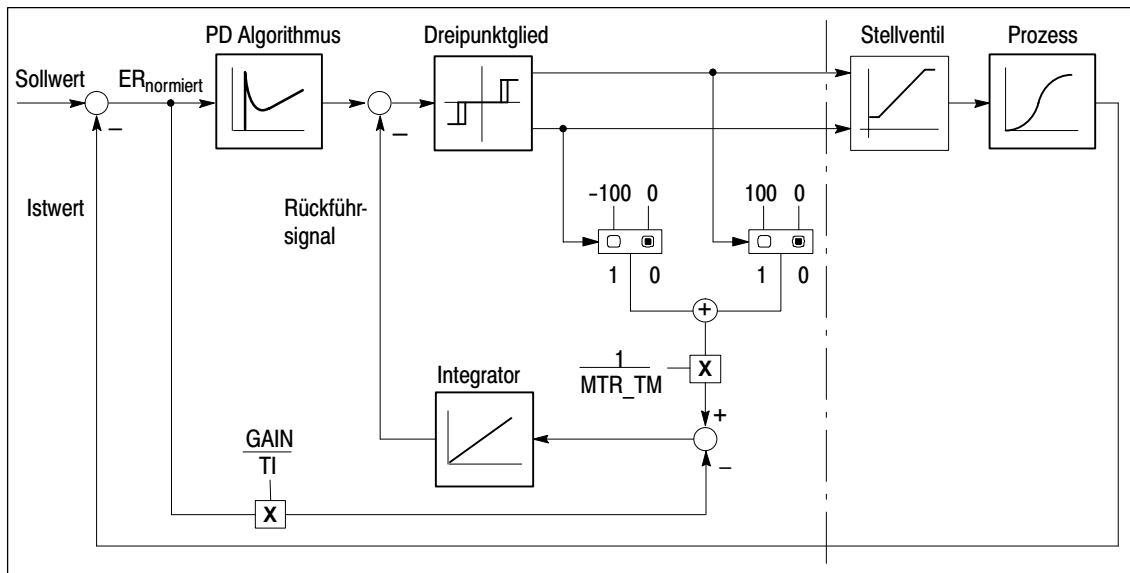


Bild 6-2 Funktionsschema des Schrittregers ohne Stellungsrückmeldung

### **Neustart/Wiederanlauf**

Der FB PID\_ES verfügt über eine Initialisierungsroutine, die durchlaufen wird, wenn der Eingangs-Parameter COM\_RST = TRUE gesetzt ist.

#### **Zeitplangeber (RMP\_SOAK)**

Bei eingeschaltetem Zeitplangeber werden die Zeitabschnitte DB\_NBR PI[0 ... NBR\_PTS].TMV zwischen den Stützpunkten aufsummiert und am Ausgang Gesamtzeit T\_TM und Gesamtrestzeit RT\_TM angezeigt.

Bei Online-Änderungen von PI[n].TMV oder bei Vorgabe von TM\_CONT und TM\_SNBR ändern sich Gesamtzeit und Gesamtrestzeit der Fahrkurve. Da die Berechnung von T\_TM und RS\_TM bei vielen Stützpunkten die Bearbeitungszeit der Funktion RMP\_SOAK stark vergrößert, wird diese Berechnung nur nach Neustart oder bei TUPDT\_ON = TRUE durchgeführt.

#### **Integrierer (INT)**

Der Integrierer wird beim Anlauf intern auf den Initialisierungswert I\_ITLVAL gesetzt. Beim Aufruf in einer Weckalarmebene arbeitet er von diesem Wert aus weiter.

Alle anderen Ausgänge werden auf ihre Vorbesetzungswerte gesetzt.

## 6.2 Stellgrößen beim Schrittreger mit Stellungsrückmeldung

### 6.2.1 Betriebsarten des Schrittreglers

#### Struktur des Schrittreglers

Der Schrittreger (PID\_ES) mit Stellungsrückmeldung setzt sich aus zwei Teilen zusammen: dem mit kontinuierlichen Signalen arbeitenden Reglerteil, der weitgehend identisch ist mit der Struktur im Funktionsbaustein PID\_CP, und einem zweiten Teil, in dem die binären Stellsignale erzeugt und in dem über die Stellungsrückmeldung gleichzeitig ein Stellungsregelkreis gebildet wird (Bild 6-3).

Der Ausgang des PID-Algorithmus wirkt als Führungsgröße auf diese Stellungsregelung und gibt damit die Position des motorischen Stellgliedes vor.

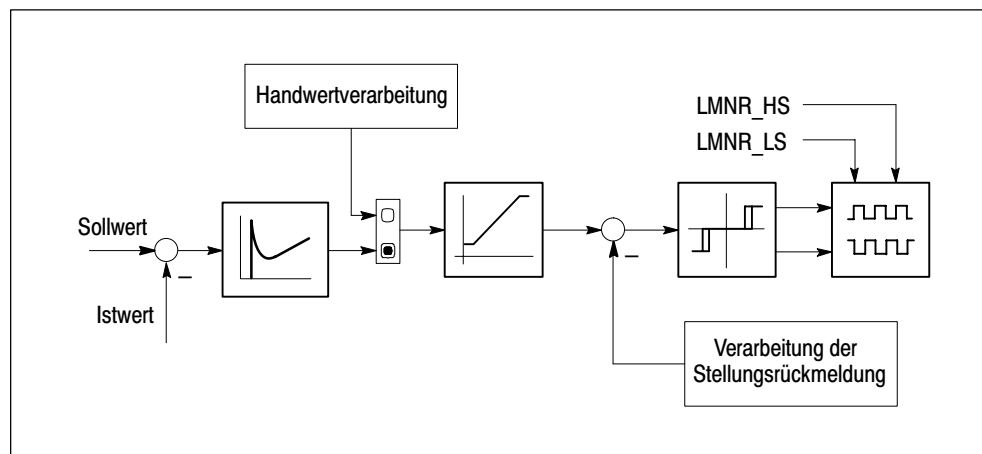


Bild 6-3 Schrittreger mit Stellungsrückmeldung

Um Übersteuerungen des Antriebs zu vermeiden, können dessen Anschlagssignale (LMNR\_HS/LMNR\_LS) zur Verriegelung der Reglerausgänge herangezogen werden (Bild 2-16). Liefert der Antrieb keine Anschlagssignale, dann müssen die Eingangsparameter LMNR\_HS und LMNR\_LS = FALSE gesetzt werden.

#### Hinweis

Sind keine Anschlagssignale vorhanden, kann der Regler nicht erkennen, ob ein mechanischer Anschlag erreicht ist. Es besteht die Möglichkeit, dass der Regler z. B. Signale zum Öffnen des Ventils ausgibt, obwohl es sich bereits am oberen Anschlag befindet.

## Betriebsarten des Schrittreger

- **Anwahl: Schrittreger mit Stellungsrückmeldung**

Immer wenn – abhängig von der Art des Stellantriebs – eine Stellungsrückmeldung zur Verfügung steht, wird die Reglerstruktur nach Bild 6-3 durch Setzen von LMNR\_ON = TRUE aktiviert.

Falls am motorischen Stellglied kein Positionssignal abgenommen werden kann, ist durch LMNR\_ON = FALSE die Struktur des Schrittreger ohne Stellungsrückmeldung zu wählen (*siehe Kapitel 6.3*).

---

### Hinweis

Der Betriebsartenschalter LMNR\_ON darf im Online-Betrieb des Regler nicht betätigt werden!

---

- **Betriebsarten**

Der Schrittreger kann in den gleichen Betriebsarten wie der stetige Regler betrieben werden, nämlich der Betriebsart "Automatik" d. h. Regelung im geschlossenen Kreis und der Betriebsart "Hand" d. h. Beeinflussung des Stellgliedes von Hand bei offenem Regelkreis. Die Handsignalbildung durch Absolutwertaufschaltung (MAN) oder Stellwertgenerator (MAN\_GEN) wird beim Schrittreger durch die Möglichkeit der direkten Schaltung der Ausgangssignale über LMNS\_ON erweitert.

### Automatikbetrieb

Ist MAN\_ON = FALSE gewählt, wird der Stellwert des PID-Algorithmus auf das Dreipunktglied geschaltet. Die Umschaltung von Hand- auf Automatikbetrieb bewirkt einen Sprung des Stellwertes LMN. Dies wirkt sich jedoch nicht nachteilig aus, da der Prozess über das integrierend wirkende Stellglied rampenförmig angesteuert wird. Der Ausgang des PID-Algorithmus ist am Messpunkt MP7 hinterlegt.

### Handwertnachführung im Automatikbetrieb

Im Automatikbetrieb wird der I/O-Parameter MAN der Stellung des Stellgliedes (LMNR\_IN, wenn LMNRP\_ON = FALSE oder MP10, wenn LMNRP\_ON = TRUE) nachgeführt. Bei der Umschaltung auf Handbetrieb bleibt die Stellgröße daher auf dem Wert stehen, der der Stellung des Stellgliedes entspricht. Sie kann nun per Bedienung verändert werden.

### Handbetrieb

Neben der Betriebsart "Automatik" gibt es bei der Schritt-Regelung drei Betriebsarten, in denen das Stellsignal manuell beeinflusst werden kann:

- Handbetrieb ohne Schalterbedienung über das Signal MAN,
- Handbetrieb über Mehr-/Weniger-Schalter" (MAN\_GEN),
- Handbetrieb durch direkte Schaltung der Binärausgänge.

Die Struktur der Handwert-Erzeugung und -Aufschaltung geht aus Bild 6-4 hervor. Über den Parameter MAN (-100.0 % bis 100.0 %) kann eine externe Stellgrößenbeeinflussung als Absolutwert aufgeschaltet oder vom Anwenderprogramm aus vorgegeben werden.

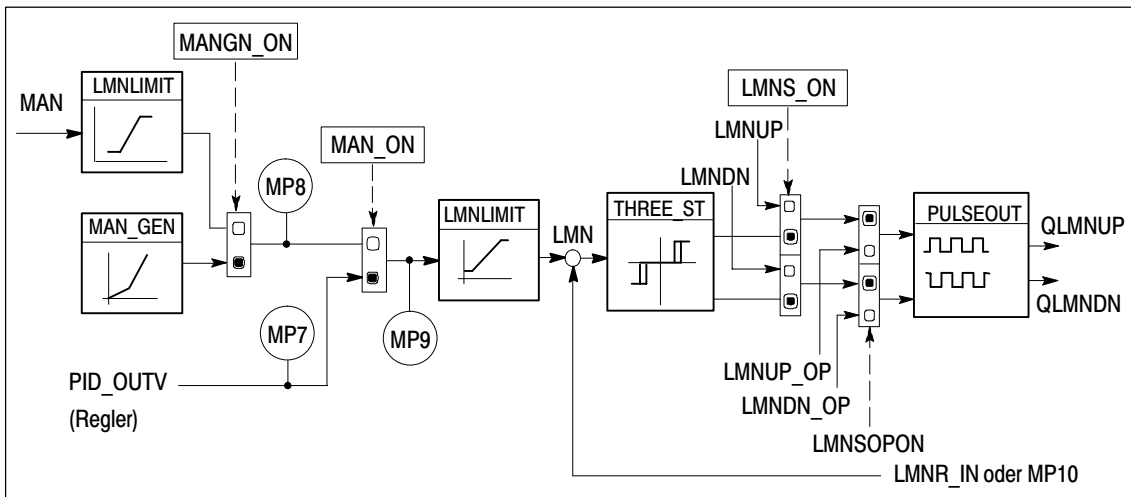


Bild 6-4 Betriebsarten und Handstellwert-Erzeugung beim Schrittreger mit Stellungsrückmeldung

Wird MAN\_GEN aus einer anderen Betriebsart eingeschaltet, dann wird der Stellwert am Ausgang (MP9) übernommen. Die Umschaltung zum Handwertgenerator ist damit immer stoßfrei. Der Handstellwert kann innerhalb der Grenzen LMN\_HLM und LMN\_LLM vergrößert oder verkleinert werden.

Wegen der direkten Einwirkung auf die Zustände der Ausgangssignale hat ein Tastbetrieb des Stellgliedes über LMNUP bzw. LMNDN immer Vorrang. Bei Rückschaltung LMNS\_ON = FALSE wird eine beliebige eingeschaltete Betriebsart "stoßfrei" übernommen.

### Schaltung der Betriebsarten

Folgende Tabelle zeigt die möglichen Betriebsarten des Schrittregers mit den erforderlichen Werten der strukturbestimmenden Schalter.

Tabelle 6-1 Betriebsarten des Schrittregers

Betriebsart \ Schalter	MNGN_ON	MAN_ON	LMNS_ON
Automatikbetrieb	beliebig	FALSE	FALSE
Handbetrieb ohne Schalterbedienung	FALSE	TRUE	FALSE
Handbetrieb über Schalter MAN_GEN	TRUE	TRUE	FALSE
Handbetrieb über Impulsschalter	beliebig	beliebig	TRUE

## 6.2.2 Stellwertbeeinflussung über das Konfigurationswerkzeug

### LMN-Anzeige und -Einstellung im Kreisbild

Das Konfigurationswerkzeug hat eine eigene Schnittstelle zum Regler-FB. Es ist deshalb jederzeit möglich, von einem PG/PC aus über das Konfigurationswerkzeug den Stellgrößenzweig zu unterbrechen und eigene Stellwerte vorzugeben (Bild 6-5).

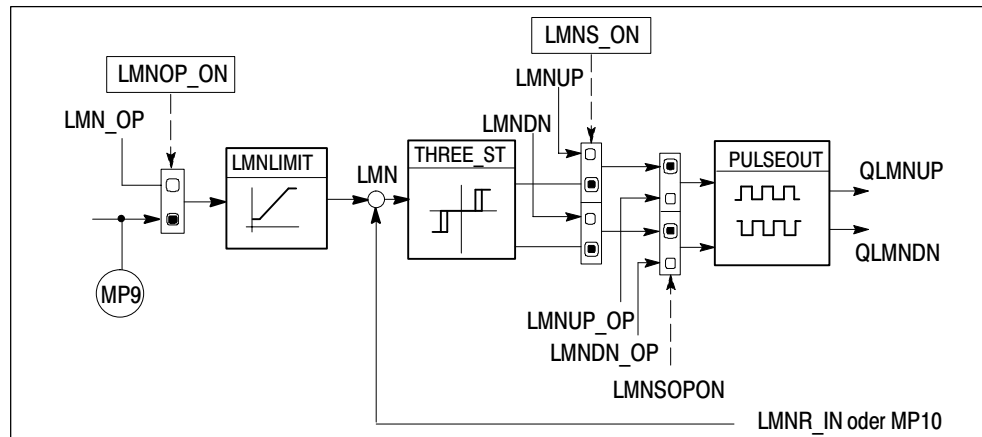


Bild 6-5 Eingriffe in den Stellwertzweig über das Konfigurationswerkzeug

Im Fenster des Kreisbildes steht dafür – bezeichnet mit 'Stellwert' – eines der drei identischen Bedienfelder zur Verfügung. Hier wird oben ('Regler: ') der aktuelle am Messpunkt MP9 anliegende Stellwert ausgegeben. Im Feld darunter ('PG: ') wird der Parameter LMN\_OP angezeigt und bedient.

### Umschalten auf Stellwertvorgabe durch Konfigurationswerkzeug

Steht der Schalter im Konfigurationswerkzeug auf 'PG: ', dann wird im Regler-FB das Schaltsignal des Strukturschalters LMNOP\_ON auf TRUE gesetzt und LMN\_OP wird auf den Stellwert durchgeschaltet.

Steht der Schalter "Regler:/PG:" im Feld Stellsignale auf "PG:", wird der Parameter LMNSOPON=TRUE gesetzt, und die Stellsignalausgänge können über die Parameter LMNUP\_OP (Hoch) bzw. LMNDN\_OP (Tief) im Kreisbild bedient werden. Dies gilt für den Schrittreger mit und ohne Stellungsrückmeldung.

Diese Bedieneingriffe wirken jeweils erst dann auf den Prozess, wenn sie durch Betätigen der Taste 'Senden' im Kreisbild auf das Automatisierungssystem übertragen werden.



### 6.2.3 Begrenzung des Absolutwertes der Stellgröße (LMNLIMIT)

#### Anwendung

Der Einstellbereich der Stellwertes bestimmt den Arbeitsbereich der Stellgröße, d. h. den Bereich, in dem sich das Stellglied im Rahmen zulässiger Stellwerte bewegen kann. Da sich – abhängig von der Art des Stellgliedes – die Grenzen für erlaubte Stellwerte meist nicht mit der 0 %- oder der 100 %- Grenze des Stellbereiches decken, sind oft weitere Bereichseinschränkungen erforderlich.

Um im jeweiligen Prozess unzulässige Zustände zu vermeiden, wird im Stellwertzweig des Schrittreglers der Einstellbereich für die Stellgröße nach oben und unten begrenzt.

#### Die Funktion LMNLIMIT

Die Funktion LMNLIMIT begrenzt die Stellgröße  $LMN(t)$  auf den unteren und oberen Wert  $LMN\_HLM$  und  $LMN\_LLM$ . Diese Werte können vorgegeben werden. Die Eingangsgröße  $inv$  muss allerdings außerhalb dieser Begrenzungen liegen. Da die Funktion nicht abgeschaltet werden kann, muss die Angabe einer Unter- und Obergrenze bei der Konfiguration immer berücksichtigt werden.

Die Zahlenwerte der Grenzen (in Prozent) werden an den Eingangsparametern für die untere und obere Begrenzung eingestellt. Bei Überschreitungen durch die Eingangsgröße  $inv(t)$  werden die zugehörigen Anzeigen über die Meldeausgänge (Bild 6-7) ausgegeben.

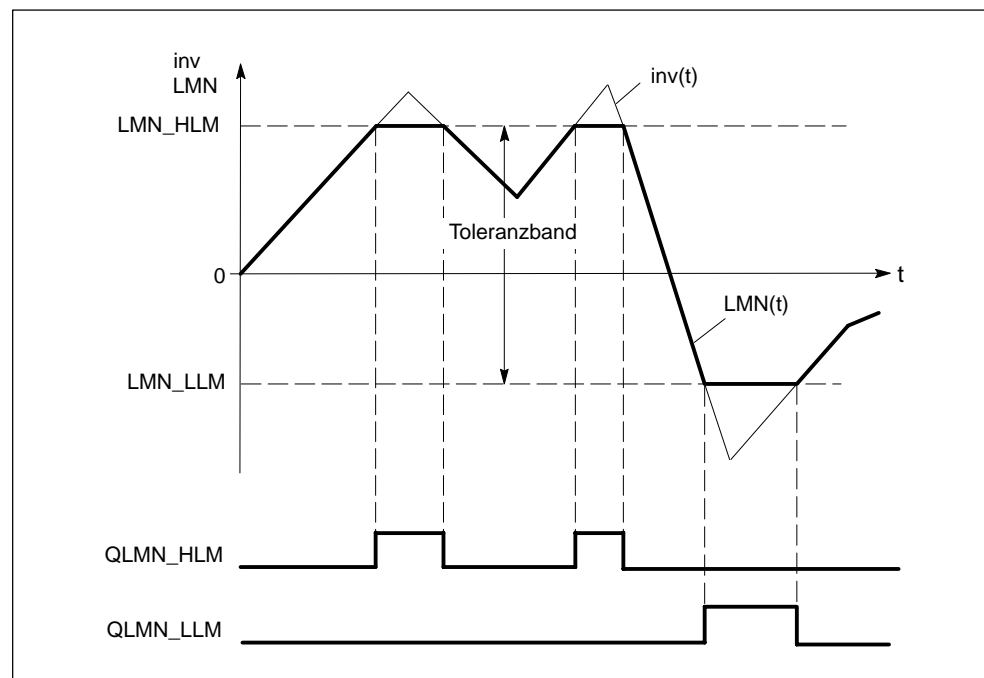


Bild 6-6 Absolutwertbegrenzung der Stellgröße  $LMN(t)$

### Anlauf und Betriebsweise

- Bei Neustart werden alle Meldeausgänge nullgesetzt.
- Die Begrenzung arbeitet gemäß folgenden Beziehungen:

LMN =	QLMN_HLM	QLMN_LLM	wenn:
LMN_HLM	TRUE	FALSE	$INV \geq LMN\_HLM$
LMN_LLM	FALSE	TRUE	$INV \leq LMN\_LLM$
INV	FALSE	FALSE	$LMN\_HLM \leq INV \leq LMN\_LLM$

Am Ausgang d. h. am Parameter LMN wird der wirksame Stellwert angezeigt.

### Parameter der Funktion LMNLIMIT

Der Eingangswert INV ist ein impliziter Parameter, der am Konfigurationswerkzeug über den Messpunkt **MP9** beobachtet werden kann.

Für ein sinnvolles Arbeiten der Begrenzungsfunktion muss gelten:

$$LMN\_HLM > LMN\_LLM$$

Parameter	Bedeutung	Zulässiger Wertebereich
LMN_HLM	Obere Begrenzung der Stellgröße	LMN_LLM bis 100.0 [%]
LMN_LLM	Untere Begrenzung der Stellgröße	-100.0 bis LMN_HLM [%]

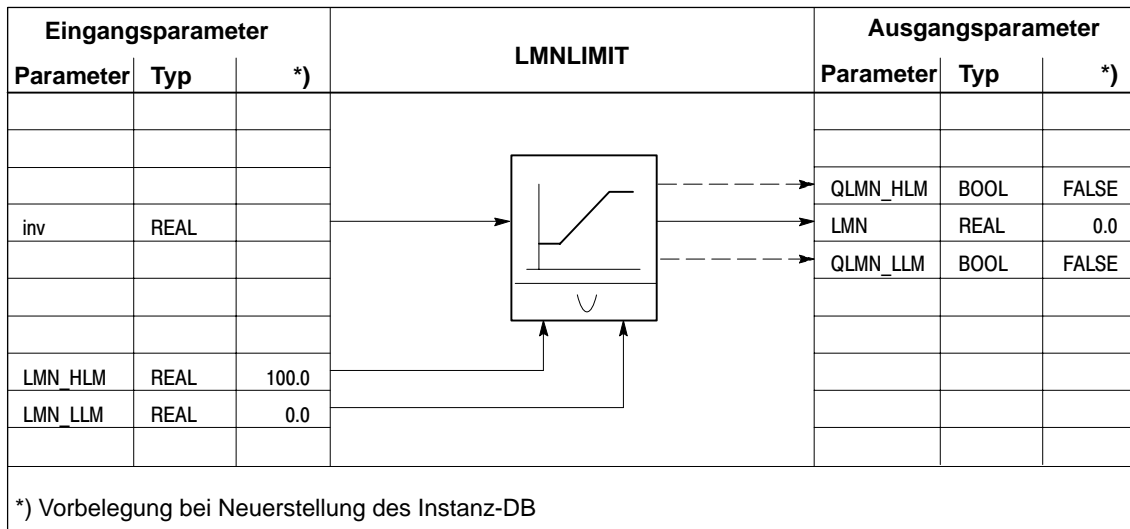


Bild 6-7 Funktionsschema und Parameter der Absolutwert-Begrenzung des Stellwertes

## 6.2.4 Verarbeitung der Stellungsrückmeldung (LMNR\_IN bzw. LMNR\_PER)

### Signalanpassung

Für die Aufschaltung der Stellungsrückmeldung auf den Vergleicher im Stellwertzweig des Schrittreger stehen, je nach Format des zu verarbeitenden Wertes, Eingänge mit entsprechender Signalverarbeitung zur Verfügung (Bild 6-8). Über LMNR\_PER lassen sich Signale im Format der SIMATIC-Peripherie und über LMNR\_IN Signale im Gleitpunktformat aufschalten.

Am Messpunkt MP10 wird der zugehörige interne Wert im Prozentbereich hinterlegt.

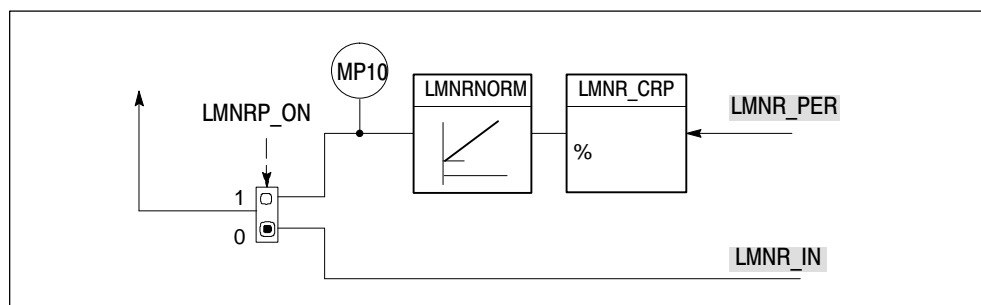


Bild 6-8 Verarbeitung der Stellungsrückmeldung beim Schrittreger

### Die Funktion LMNR\_CRP

Wird der zu verarbeitende Wert der Stellungsrückmeldung von einer Analogeingabegruppe geliefert, dann muss der Zahlenwert des Peripherie-Datenwortes in einen Zahlenwert im Gleitpunktformat (Prozentbereich) umgesetzt werden.

Die Funktion LMNR\_CRP setzt diesen am Eingang LMNR\_PER anliegenden Zahlenwert der Stellungsrückmeldung in einen auf Prozent normierten Gleitpunktwert um. Eine Prüfung auf positiven/negativen Überlauf, Erreichen des Über-/Untersteuerungsbereiches und auf Drahtbruch wird dabei nicht durchgeführt.

Folgende Tabelle gibt einen Überblick über Bereiche und Zahlenwerte vor und nach der Bearbeitung durch den Umform- und Normierungsalgorithmus der Funktion LMNR\_CRP.

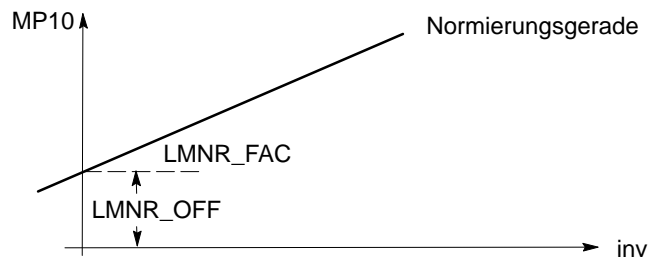
Peripheriewert LMNR_PER	Ausgangswert in %
32767	118,515
27648	100,000
1	0,003617
0	0,000
-1	-0,003617
-27648	-100,000
-32768	-118,519

## Die Funktion LMNRNORM

Liegt die Stellungsrückmeldung als physikalische Größe vor (z. B. 240 bis 800 mm oder 0 bis 60  $\pm^\circ$ ), dann muss der bereits als Gleitpunktwert (in %) gewandelte Rückmeldeeingang auf den für die Weiterverarbeitung erforderlichen internen Gleitpunktwert im Bereich 0 bis 100 % normiert werden.

Zur eindeutigen Festlegung der Normierungsgeraden sind folgende Parameter zu definieren:

- der Faktor (für die Steigung): **LMNR\_FAC**
- der Offset der Normierungsgeraden im Nullpunkt: **LMNR\_OFF**



Der Normierungswert MP10 (Bild 6-8) wird nach folgender Beziehung aus dem jeweiligen Eingangswert inv (LMNR\_PER) berechnet:

$$MP10 = inv * LMNR\_FAC + LMNR\_OFF$$

## Anlauf

Die Funktion ist wirksam, wenn der Steuereingang LMNRP\_ON = TRUE ist. Funktionsintern werden keine Werte begrenzt. Eine Prüfung der Parameter findet nicht statt.

## Parameter der Funktionen LMNR\_CRP und LMNRNORM

Der Peripherieeingang LMNR\_PER ist dann auf den Rückmeldezweig durchgeschaltet, wenn LMNRP\_ON = TRUE gesetzt ist. Der Wert – im internen Format – von LMNR\_PER ist am Messpunkt MP10 zugänglich.

Parameter	Bedeutung	Zulässiger Wertebereich
LMNR_PER	Rückmeldewert im Peripherieformat	
LMNR_FAC	Steigung der Normierungsgeraden am Eingang der Stellungsrückmeldung LMNR_PER	Technischer Wertebereich (dimensionslos)
LMNR_OFF	Nullpunkt der LMNR-Normierungsgeraden	-100.0 bis + 100.0 [%]

Eingangsparameter			LMNR_CRP + LMNRNORM	Ausgangsparameter			
Parameter	Typ	*)		Signal	Typ	*)	
LMNRP_ON	BOOL	FALSE		LMNR	REAL		
LMNR_IN	REAL	0.0					
LMNR_PER	WORD	W#16#0000					
LMNR_FAC	REAL	1.0					
LMNR_OFF	REAL	0.0					

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 6-9 Funktionsschema und Parameter der Peripheriewert-Umformung für die Stellungsrückmeldung

## 6.2.5 Erzeugung der Stellsignale (QLMNUP/QLMNDN)

### Signalverarbeitung

Die Differenz zwischen Stellwert LMN und Stellungsrückmeldung LMNR wird auf das Dreipunktglied mit Hysterese THREE\_ST geschaltet. Der nachgeschaltete Impulsformer PULSEOUT sorgt dafür, dass eine Mindestimpuls- und Mindestpausendauer eingehalten wird, um die Stellglieder zu schonen. (Bild 6-10). Ein Ansprechen der Endlagenschalter des Stellgliedes (LMNR\_HS/LMNR\_LS) verursacht eine Sperrung des betreffenden Ausgangs.

Auch bei Handbetätigung der binären Ausgangssignale (LMNS\_ON = TRUE oder LMNSOPON = TRUE) werden die Mindestimpulszeit PULSE\_TM und die Mindestpausendauer BREAK\_TM mit berücksichtigt. Ist der zugeordnete Endlagenschalter gesetzt, dann wird auch bei Handbetrieb der Ausgang gesperrt.

Sind beide Signalschalter zur Stellwertsignalbedienung gesetzt (LMNUP = LMNDN = TRUE bzw. LMNUP\_OP = LMNDN\_OP = TRUE), wird an den Ausgängen PLMNUP und QLMNDN immer FALSE ausgegeben.

Der direkte Wechsel von "Stellwertsignal Auf" (QLMNUP = TRUE, QLMNDN = FALSE) nach "Stellwertsignal Zu" (QLMNUP = FALSE, QLMNDN = TRUE) ist nicht möglich. Der Impulsformer fügt einen Zyklus mit QLMNUP = QLMNDN = FALSE ein.

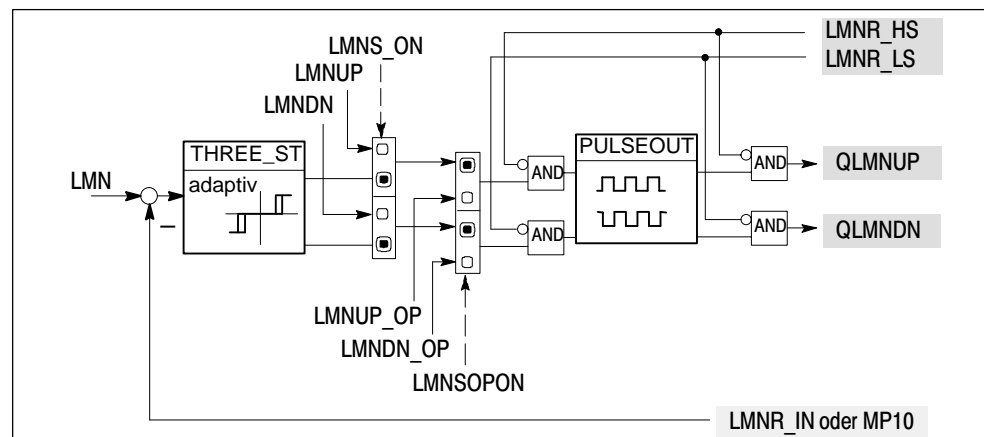


Bild 6-10 Erzeugung des binären Stellsignals beim Schrittreger mit Stellungsrückmeldung

## Das Dreipunktglied mit Hysterese THREE\_ST

Die Abweichung zwischen den Werten vom Stellsignal des Reglers und der aktuell erreichten Position des Stellgliedes bildet die Eingangsgröße des Dreipunktgliedes. An dessen Ausgang werden zwei Binärsignale erzeugt, die je nach Größe und Vorzeichen des Differenzwertes am Eingang sich im Zustand gesetzt bzw. rückgesetzt befinden.

Der Dreipunktschalter THREE\_ST reagiert auf das Eingangssignal INV entsprechend folgender Beziehungen (ThrOn = Einschaltsschwelle, ThrOff = Ausschaltsschwelle) und nimmt dabei jeweils eine der drei möglichen Kombinationen seiner Ausgangssignale UP/DOWN an (Bild 6-11):

UP	DOWN	Eingangskombination
TRUE	FALSE	$INV \geq ThrOn$ oder $INV > (ThrOff \text{ und } UP_{alt} = TRUE)$
FALSE	TRUE	$INV \leq -ThrOn$ oder $INV < (-ThrOff \text{ und } DOWN_{alt} = TRUE)$
FALSE	FALSE	$ INV  \leq ThrOff$

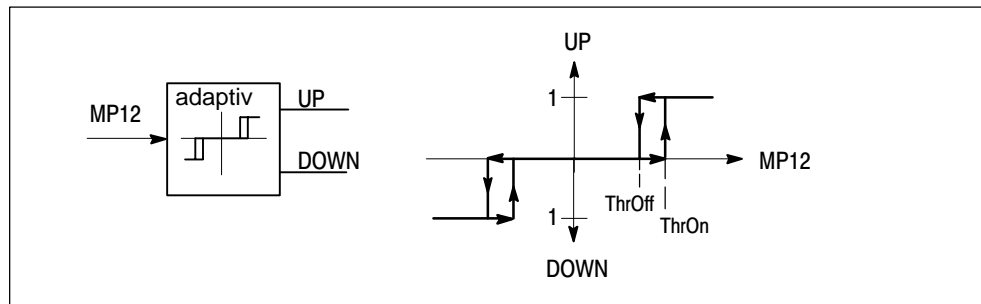


Bild 6-11 Funktionsschema des Dreipunktgliedes THREE\_ST

Die Abschaltsschwelle ThrOff muss größer sein als die Änderung der Stellungsrückmeldung nach einer Impulsdauer. Dieser Wert ist abhängig von der Stellzeit des Motors MTR\_TM und wird wie folgt berechnet:

$$ThrOff = 0.5 * \frac{110}{MTR\_TM} * CYCLE$$

PULSE\_TM muss ein ganzzahliges Vielfaches von CYCLE sein.

### Hinweis

Eine zu groß eingestellte Motorstellzeit (10% über der realen Stellzeit) bewirkt ein ständiges Ein- und Ausschalten der Stellwertsignale QLMNUP und QLMNDN.

### Adaption der Ansprechschwelle ThrOn

Um bei der Ausregelung größerer Regeldifferenzen die Schalthäufigkeit zu reduzieren, wird die Ansprechschwelle +ThrOn während des Betriebes automatisch angepasst, während ThrOff konstant bleibt. ThrOn ist begrenzt auf:

$$\text{Min ThrOn} = \frac{100}{\text{MTR\_TM}} * \text{MAX} (\text{PULSE\_TM}, \text{CYCLE} )$$

$$\text{Max ThrOn} = 10$$

Die Adaption der Ansprechschwelle ist bei reinen P-, D- oder PD-Reglern abgeschaltet. Hierbei ist:

$$\text{ThrOn} = \text{Min ThrOn}.$$

### Der Impulsformer PULSEOUT

Der Impulsformer sorgt dafür, dass beim Setzen und Ausschalten der Ausgangsimpulse jeweils ein Mindestwert für die Impulsdauer bzw. für die Pausendauer eingehalten wird.

Zur Schonung des Stellgliedes können deshalb eine Mindestimpulsdauer PULSE\_TM und eine Mindestpausendauer BREAK\_TM parametrisiert werden. Die Dauer der Ausgangsimpulse QLMNUP bzw. QLMNDN ist immer größer als PULSE\_TM und die Unterbrechungsdauer zwischen zwei Impulsen immer größer als BREAK\_TM. Das Bild 6-12 zeigt die Arbeitsweise von PULSEOUT am Beispiel des UP-Signals.

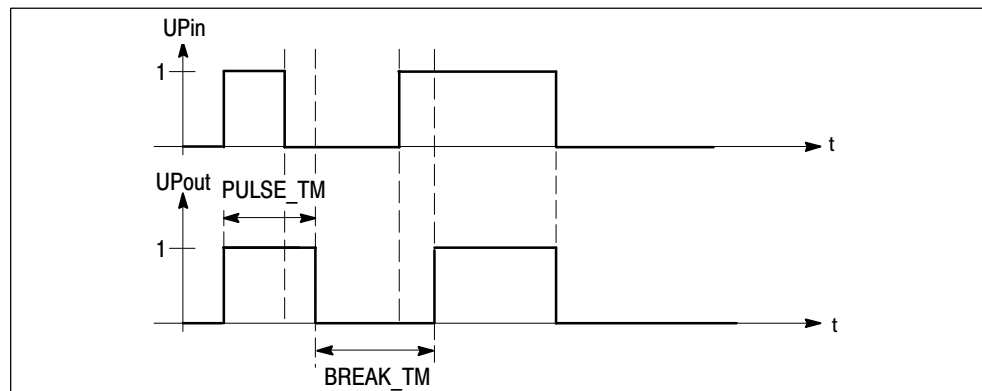


Bild 6-12 Arbeitsweise des Impulsformers PULSEOUT



### Parameter von THREE\_ST und PULSEOUT

Die Einstellwerte der Parameter PULSE\_TM und BREAK\_TM müssen ein ganzzahliges Vielfaches der Zykluszeit CYCLE sein. Bei kleineren Einstellwerten ist für die Mindestimpuls- und Mindestpausendauer die Zykluszeit CYCLE wirksam.

Eingangsparameter			THREE_ST PULSEOUT	Ausgangsparameter			
Parameter	Typ	*)		Signal	Typ	*)	
LMNSOPON	BOOL	FALSE					
LMNUP_OP	BOOL	FALSE					
LMNDN_OP	BOOL	FALSE					
LMNS_ON	BOOL	TRUE					
LMNUP	BOOL	FALSE					
LMNDN	BOOL	FALSE					
LMNR_HS	BOOL	FALSE					
LMNR_LS	BOOL	FALSE					
MP12	REAL	0.0			QLMNUP	BOOL	FALSE
					QLMNDN	BOOL	FALSE
MTR_TM	TIME	T#30s					
PULSE_TM	TIME	T#3s					
BREAK_TM	TIME	T#3s					

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 6-13 Funktionsschema und Parameter der Stellsignalerzeugung beim Schrittreger

## 6.3 Stellgrößenverarbeitung beim Schrittreger ohne Stellungsrückmeldung

### Struktur und Funktion des Schrittreglers

Der Schrittreger (PID\_ES) ohne Stellungsrückmeldung setzt sich aus zwei Teilen zusammen: dem mit kontinuierlichen Signalen arbeitenden PD-Reglerteil und einem zweiten Teil, in dem die binären Stellsignale aus der Differenz von PD-Anteil und Rückführung erzeugt werden (Bild 6-14).

Der Integrator in der Rückführung dieses Schrittreglers summiert das Differenzsignal aus  $\pm 100/\text{MTR\_TM}$  und  $\text{ER}_{\text{normiert}} * \text{GAIN}/\text{TI}$  auf. Am Ausgang des Integrators steht die Differenz aus gedachter Motorstellung und dem I-Anteil. Im eingeschwungenen Zustand wird der Ausgang des Integrators und der PD-Anteil zu Null. Da der Eingang des Dreipunktgliedes auch zu Null wird, bleiben die binären Stellsignale QLMNUP und QLMNDN auf FALSE stehen.

Der I-Anteil des PID-Algorithmus ist abgeschaltet. Beim Schrittreger ohne Stellungsrückmeldung ist es nicht vorgesehen, den I-Anteil vorzubelegen oder einzufrieren. Der Handbetrieb über den Parameter MAN entfällt hier wegen der fehlenden Information über die Lage des Stellgliedes.

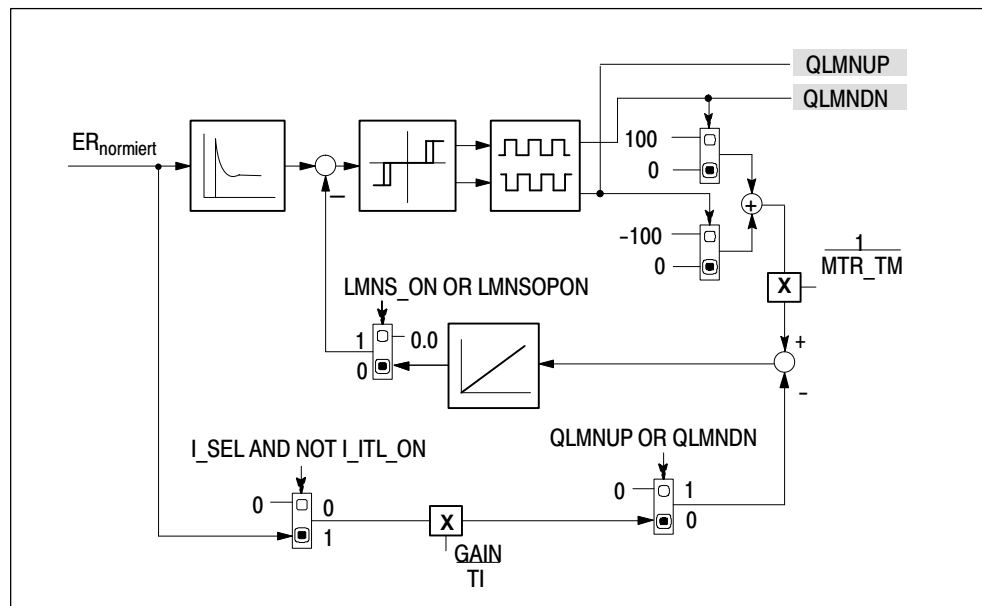


Bild 6-14 Schrittreger ohne Stellungsrückmeldung

Um Übersteuerungen des Antriebs zu vermeiden, können dessen Anschlagsignale (LMNR\_HS/LMNR\_LS) zur Verriegelung der Reglerausgänge herangezogen werden (Bild 2-17). Liefert der Antrieb keine Anschlagsignale, dann müssen die Eingangsparameter LMNR\_HS und LMNR\_LS = FALSE gesetzt werden.

### Hinweis

Wenn keine Anschlagssignale vorhanden sind, kann der Regler nicht erkennen, ob ein mechanischer Anschlag erreicht wurde. Es besteht die Möglichkeit, dass der Regler z. B. Signale zum Öffnen des Ventils ausgibt, obwohl es sich bereits am oberen Anschlag befindet.

## Betriebsarten des Schrittreger

- **Anwahl: Schrittreger ohne Stellungsrückmeldung**

Falls ein Rückmeldesignal der Stellgliedposition nicht zur Verfügung steht, wird die Reglerstruktur nach Bild 6-14 durch Setzen von LMNR\_ON = FALSE aktiviert.

- **Betriebsarten**

Wegen der fehlenden Information über die Lage des Stellgliedes entfällt beim Schrittreger ohne Stellungsrückmeldung der Handbetrieb über den Parameter MAN bzw. den Handstellwertgeber MAN\_GEN. Neben der Betriebsart "Automatik", d. h. Regelung im geschlossenen Kreis, ist der Betriebszustand "Hand" mit direkter Tastung der Ausgangsimpulse bei offenem Regelkreis durch LMNS\_ON = TRUE einstellbar.

### Handbetrieb

Bei eingeschaltetem Handbetrieb (LMNS\_ON = TRUE) können die binären Ausgangssignale QLMNUP und QLMNDN über die Schalter LMNUP und LMNDN gesetzt werden (Bild 6-15). Die Mindestimpulszeit PULSE\_TM sowie die Mindestpausendauer werden dabei eingehalten.

Ist einer der Endlagenschalter LMNR\_HS bzw. LMNR\_LS gesetzt, so wird – auch bei Handbetrieb – das entsprechende Ausgangssignal gesperrt.

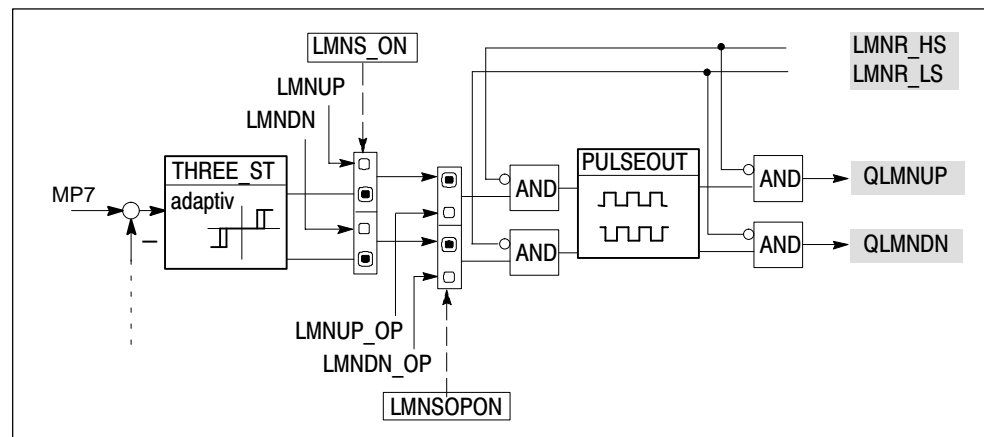


Bild 6-15 Handbetrieb beim Schrittreger ohne Stellungsrückmeldung

Der Tastbetrieb des Stellgliedes durch LMNUP bzw. LMNDN wirkt direkt auf die Ausgangssignale und hat daher immer Vorrang. Bei Rückschaltung LMNS\_ON = FALSE geht der Regler "stoßfrei" in den Automatikbetrieb über.

Folgende Tabelle zeigt die möglichen Betriebsarten des Schrittregers ohne Stellungsrückmeldung:

Tabelle 6-2 Betriebsarten des Schrittregers ohne Stellungsrückmeldung

Betriebsart	Schalter	LMNS_ON
Automatikbetrieb		FALSE
Handbetrieb der binären Ausgangssignale		TRUE

### Erzeugung der Stellsignale QLMNUP/QLMNDN

Die Differenz zwischen dem PD-Anteil des Reglers und der Rückführgröße (MP11) wird auf das Dreipunktglied mit Hysterese THREE\_ST geschaltet. Der nachgeschaltete Impulsformer PULSEOUT sorgt dafür, dass eine Mindestimpuls- und Mindestpausendauer eingehalten wird, um die Stellglieder zu schonen (Bild 6-16). Das Ansprechen der Endlagenschalter des Stellgliedes (LMNR\_HS/LMNR\_LS) verursacht eine Sperrung des betreffenden Ausgangs.

Auch bei Handbetätigung der binären Ausgangssignale (LMNS\_ON = TRUE oder LMNSOPON = TRUE) werden die Mindestimpulszeit PULSE\_TM und die Mindestpausendauer BREAK\_TM mit berücksichtigt (Bild 6-15). Ist der zugeordnete Endlagenschalter gesetzt, dann wird auch bei Handbetrieb der Ausgang gesperrt.

Sind beide Signalschalter zur Stellwertsignalbedienung gesetzt (LMNUP = LMNDN = TRUE bzw. LMNUP\_OP = LMNDN\_OP = TRUE), wird an den Ausgängen PLMNUP und QLMNDN immer FALSE ausgegeben.

Der direkte Wechsel von "Stellwertsignal Auf" (QLMNUP = TRUE, QLMNDN = FALSE) nach "Stellwertsignal Zu" (QLMNUP = FALSE, QLMNDN = TRUE) ist nicht möglich. Der Impulsformer fügt einen Zyklus mit QLMNUP = QLMNDN = FALSE ein.

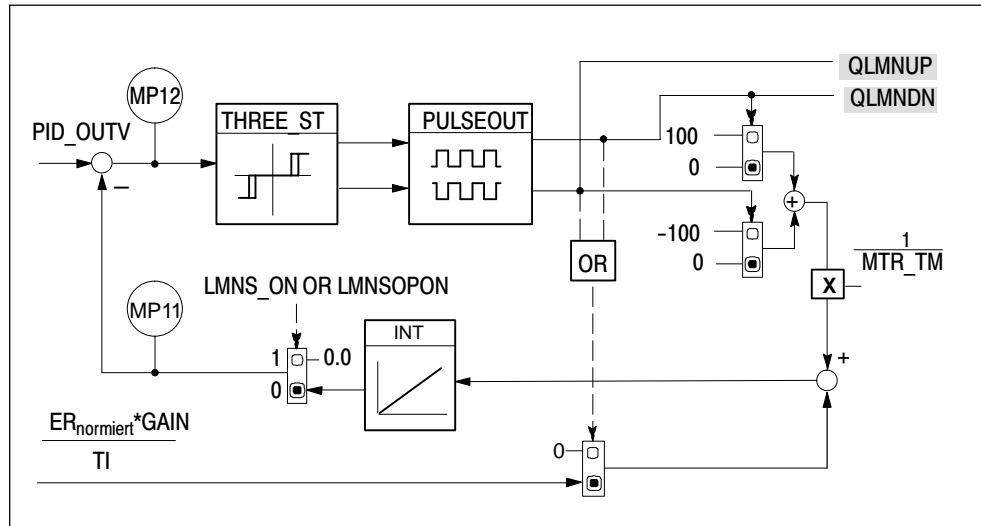


Bild 6-16 Erzeugung des binären Stellsignals beim Schrittreger ohne Stellungsrückmeldung

### Das Dreipunktglied mit Hysterese THREE\_ST

Das Eingangssignal des Dreipunktgliedes wird durch die Abweichung zwischen den Werten vom PD-Anteil des Reglers und der Rückführgröße gebildet. An dessen Ausgang werden zwei Binärsignale erzeugt, die sich, je nach Größe und Vorzeichen des Differenzwertes, am Eingang im Zustand gesetzt bzw. rückgesetzt befinden.

Der Dreipunktschalter THREE\_ST reagiert auf das Eingangssignal MP12 (PD-Anteil - Rückführung) entsprechend folgender Beziehungen (ThrOn = Einschalt-schwelle, ThrOff = Ausschalt-schwelle) und nimmt dabei jeweils eine der drei möglichen Kombinationen seiner Ausgangssignale UP/DOWN an (Bild 6-17):

UP	DOWN	Eingangskombination
TRUE	FALSE	$INV \geq ThrOn$ oder $INV > (ThrOff \text{ und } UP_{alt} = TRUE)$
FALSE	TRUE	$INV \leq ThrOn$ oder $INV < (-ThrOff \text{ und } DOWN_{alt} = TRUE)$
FALSE	FALSE	$INV \leq ThrOff$

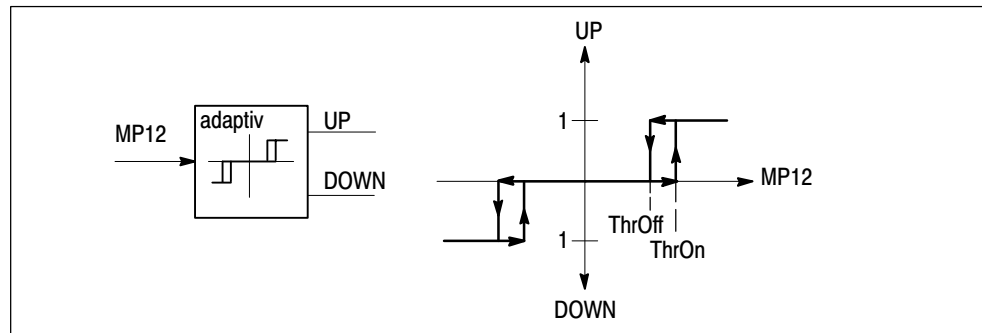


Bild 6-17 Funktionsschema des Dreipunktgliedes THREE\_ST

Die Abschaltenschwelle ThrOff muss größer sein als die Positionsänderung des Stellgliedes (in %) nach einer Impulsdauer. Dieser Wert ist abhängig von der Stellzeit des Motors MTR\_TM und wird wie folgt berechnet:

$$\text{ThrOff} = 0.5 * \frac{110}{\text{MTR\_TM}} * \text{CYCLE}$$

### Adaption der Ansprechschwelle ThrOn

Um bei der Ausregelung größerer Regeldifferenzen die Schalhäufigkeit zu reduzieren, wird die Ansprechschwelle +ThrOn während des Betriebes automatisch angepasst, während ThrOff konstant bleibt. ThrOn ist begrenzt auf:

$$\text{Min ThrOn} = \frac{100}{\text{MTR\_TM}} * \text{MAX}(\text{PULSE\_TM}, \text{CYCLE})$$

$$\text{Max ThrOn} = 10$$

Die Adaption der Ansprechschwelle ist bei reinen P-, D- oder PD-Reglern abgeschaltet. Hierbei ist:

$$\text{ThrOn} = \text{Min ThrOn.}$$

### Der Impulsformer PULSEOUT

Der Impulsformer hat die gleiche Funktionalität wie der Impulsformer beim Schrittregler mit Stellungsrückmeldung (*siehe Kapitel 6.2.5*).

## Simulation der Stellungsrückmeldung

Steht keine Stellungsrückmeldung als messbare Größe zur Verfügung, so kann diese auch simuliert werden (LMNRS\_ON = TRUE). Die Optimierung der PID-Reglerparameter mit Hilfe des Konfigurationswerkzeugs benötigt immer das Signal der Stellungsrückmeldung als Eingangsgröße.

Die Nachbildung der Stellungsrückmeldung geschieht durch einen Integrator mit der Motorstellzeit MTR\_TM als Integrationszeitkonstante (Bild 6-18). Im Zustand LMNRS\_ON = FALSE wird am Ausgang des Integrators LMNR\_SIM der Startwert des Parameters LMNRSVAL ausgegeben. Nach dem Umschalten von auf TRUE läuft die Simulation mit diesem Startwert an.

Ist LMNR\_HS = TRUE, wird die Integration nach oben angehalten, ist LMNR\_LS = TRUE, wird sie nach unten angehalten. Ein Abgleich der simulierten Stellungsrückmeldung an den Endlagen erfolgt nicht.

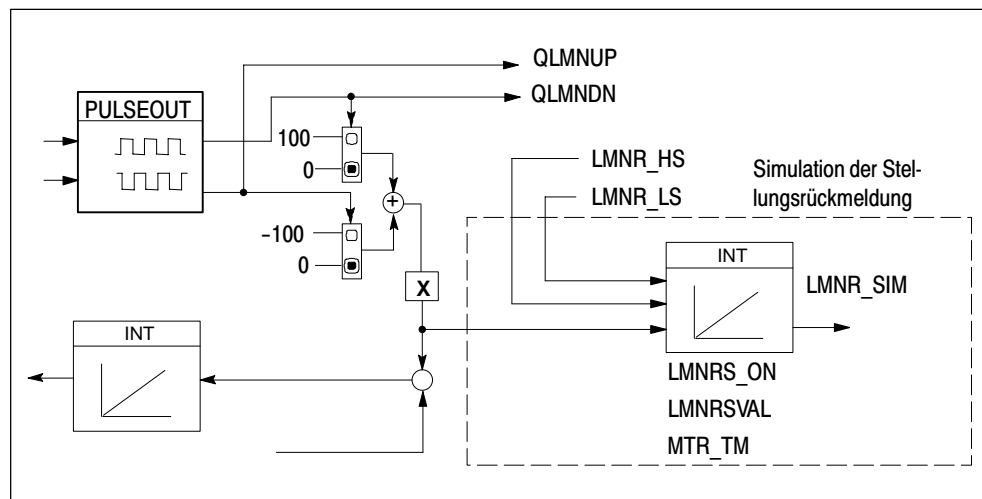


Bild 6-18 Simulation der Stellungsrückmeldung

### Hinweis

Die Stellungsrückmeldung wird lediglich simuliert. Sie muss nicht mit der wirklichen Position des Stellgliedes übereinstimmen. Ist eine reale Stellungsrückmeldung vorhanden, dann sollte diese in jedem Falle genutzt werden.





## 6.4 Schrittreger in Kaskadenschaltungen

### Auftrennung der Reglerkaskade

In einer Kaskade stehen mehrere Regler direkt miteinander in Beziehung. Sie müssen daher Vorkehrungen treffen, damit nach Auftrennung der Kaskadenstruktur an einer beliebigen Stelle der Kaskadenbetrieb problemlos wieder aufgenommen werden kann.

In Kaskadenregelungen wird deshalb in den unterlagerten Reglern über eine Oder-Logik aus den Zustandssignalen der Schalter im Sollwert- und Stellgrößenzweig ein Signal QCAS gebildet. Dieses Signal betätigt in den überlagerten Reglern einen Schalter, der die Regler in den Nachführbetrieb überführt. Die Nachführgröße ist dabei immer der Sollwert SP des unterlagerten Kreises (Bild 6-20).

Die Umschaltung vom Nachführ- in den Automatikbetrieb erfolgt stoßfrei wie bei der Hand- Automatikumschaltung.

### Hinweis

Schrittreger (PID\_ES) können in Kaskadenschaltungen nur als Folgeregler in unterlagerten Regelkreisen eingesetzt werden.

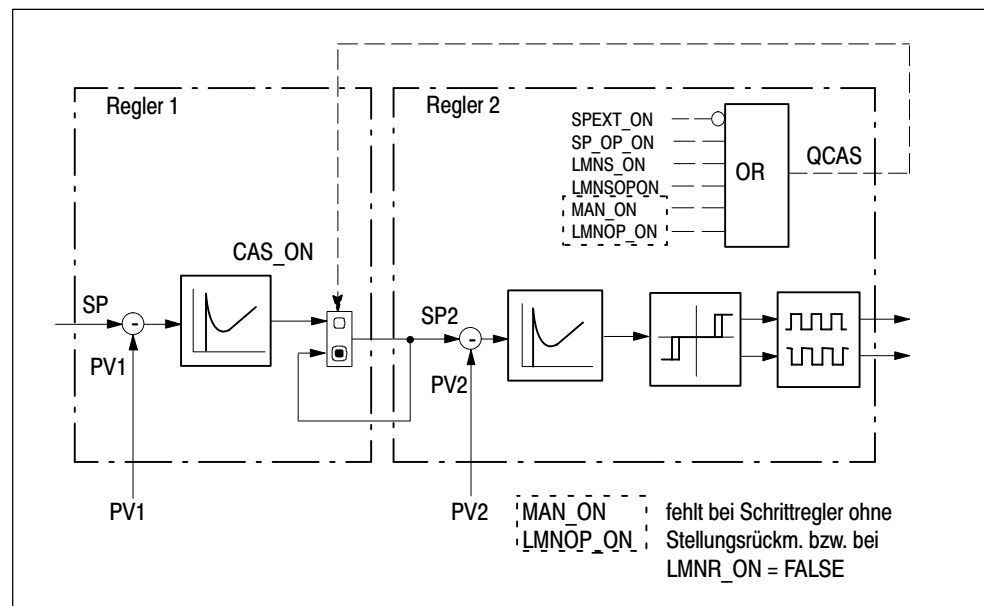


Bild 6-20 Zweischleifige Kaskadenregelung mit Schrittreger

### Bausteinverschaltung

Nachstehendes Bild zeigt das Prinzip der Regler- bzw. Bausteinverschaltung in mehrschleifigen Kaskaden.

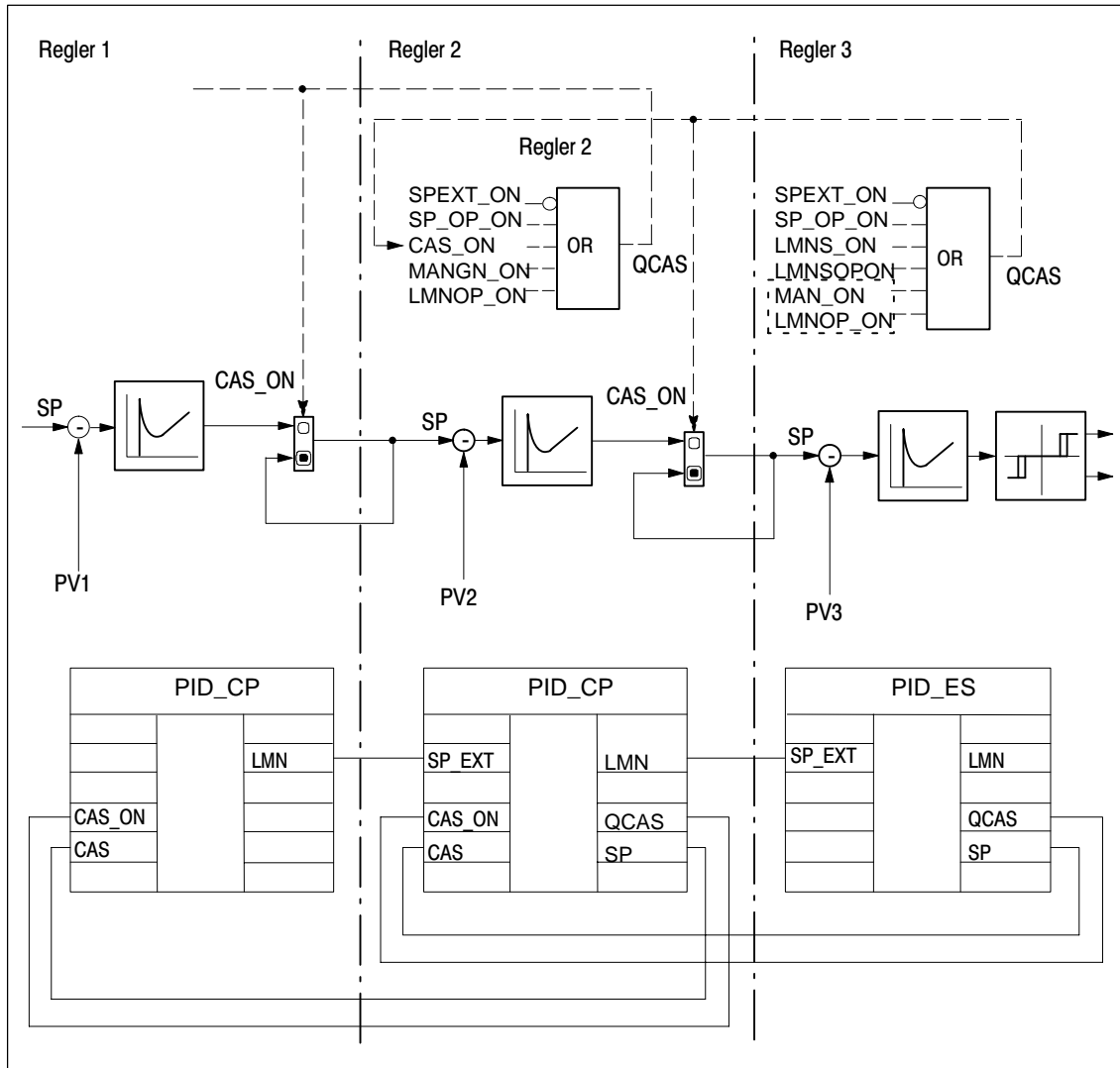


Bild 6-21 Verschaltung einer Kaskade mit zwei unterlagerten Regelkreisen und Schrittregler

# Regleraufrufverteiler und Beispiele für Reglerkonfigurationen

# 7

## 7.1 Regleraufrufverteiler (LP\_SCHED)

### Anwendung

Der Regleraufrufverteiler LP\_SCHED wird verwendet, wenn die Zahl der Weckalarme einer CPU nicht ausreicht, um die gewünschten (verschiedenen) Abtastzeiten zu realisieren. Er ermöglicht den Aufruf von bis zu 256 Regelkreisen mit Abtastzeiten, die ein ganzzahliges Vielfaches des Weckalarmzeittakts betragen.

### Übersicht

Die Funktion "LP\_SCHED" liest aus dem globalen Datenbaustein "DB\_LOOP" die von Ihnen vorgegebenen Parameter, berechnet sich daraus die für die Aufrufverteilung notwendigen Variablen und legt diese wieder im DB "DB\_LOOP" ab.

Sie müssen die FC "LP\_SCHED" in einem Weckalarm-OB aufrufen. Danach müssen Sie in demselben Weckalarm-OB einen bedingten Aufruf aller zugehörigen Regelkreise programmieren. Die Bedingung für den Aufruf der einzelnen Regelkreise wird von der FC "LP\_SCHED" ermittelt und im DB "DB\_LOOP" hinterlegt. Der Aufruf der Regelkreis-FBs "PID\_CP" bzw. "PID\_ES" kann nicht durch die FC "LP\_SCHED" erfolgen, da beim Aufruf der FBs deren Ein- und Ausgangsparameter belegt werden müssen.

Sie können im laufenden Betrieb den Aufruf einzelner Regelkreise manuell sperren und darüber hinaus einzelne Regelkreise rücksetzen.

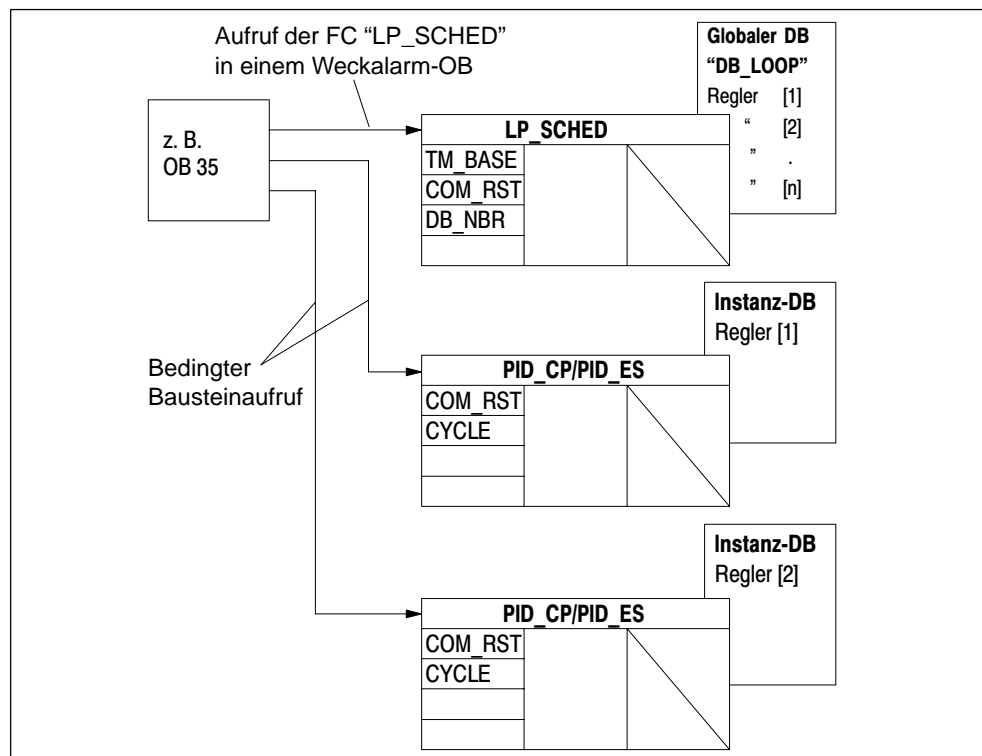


Bild 7-1 Prinzip des Regleraufrufs über den Aufrufverteiler LP\_SCHED am Beispiel von zwei Regelkreisen

### Aufbau des DB "DB\_LOOP"

Tabelle 7-1 Globaler Datenbaustein "DB\_LOOP" für den Regleraufruf

Parameter	Typ	Wertebereich	Beschreibung
GLP_NBR	INT	1...256	größte Regelkreisnummer
ALP_NBR	INT	1...256	aktuelle Regelkreisnummer
<b>Regelkreis Nr. 1</b>			
MAN_CYC [1]	TIME	≥20ms	von Ihnen vorgegebene Abtastzeit
MAN_DIS [1]	BOOL		man. Regleraufruf sperren
MAN_CRST [1]	BOOL		man. Neustart setzen
ENABLE [1]	BOOL		Freigabe
COM_RST [1]	BOOL		Neustart
ILP_COU [1]	INT		interner Regelkreiszähler
CYCLE [1]	TIME	≥20ms	von der FC "LP_SCHED" berechnete Abtastzeit
<b>Regelkreis Nr. 2</b>			
MAN_CYC [2]	TIME	≥20ms	von Ihnen vorgegebene Abtastzeit
MAN_DIS [2]	BOOL		man. Regleraufruf sperren
...	..		...

Kurze Übersicht:

- Sie müssen die Variablen GLP\_NBR und MAN\_CYC[x],  $x = 1, \dots, \text{GLP\_NBR}$  parametrieren.
- MAN\_DIS[x] dient zum Sperren des Aufrufs von Regelkreis x im laufenden Betrieb.
- MAN\_CRST[x] dient zum Anstoß eines Initialisierungslaufs für den Regelkreis x im laufenden Betrieb.
- In die Variable ENABLE[x] trägt die FC "LP\_SCHED" die Aufrufbedingung für den Regelkreis x ein.
- Die Variablen COM\_RST[x] und CYCLE[x] werden von der FC "LP\_SCHED" beschrieben. Sie dienen zur Verschaltung auf die Eingänge COM\_RST und CYCLE der Regelkreis-FBs.
- Die Variablen ALP\_NBR und ILP\_COU[x] sind interne Variablen der FC "LP\_SCHED". Sie können Ihnen zur Beobachtung der Funktion "LP\_SCHED" nützlich sein.

### Parametrierung der Regleraufrufe im DB "DB\_LOOP"

Sie müssen die Parametrierung des Regleraufrufverteilers ohne Unterstützung des Konfigurationswerkzeugs vornehmen, brauchen den DB "DB\_LOOP" jedoch nicht komplett neu zu erstellen. Er ist als Kopiervorlage in der Bibliothek "Standard PID Control" enthalten.

Sie müssen die folgenden Variablen im DB "DB\_LOOP" parametrieren:

- GLP\_NBR: Anzahl der Regelkreise (bzw. Regelkreis-FBs), deren Aufruf von der FC "LP\_SCHED" verwaltet wird (max. 256)
- MAN\_CYC[x],  $x = 1, \dots, \text{GLP\_NBR}$ : die von Ihnen gewünschte Abtastzeit für die einzelnen Regelkreise. Bitte beachten Sie dabei für jeden Regelkreis die unten angegebene Bedingung für MAN\_CYC[x]. Sonst kann die parametrierte Abtastzeit nicht garantiert werden.

Falls Sie für einen oder mehrere Regelkreise die zugehörigen Elemente des Feldes MAN\_CYC im laufenden Betrieb ändern, wird diese Änderung mit dem nächsten Aufruf der FC "LP\_SCHED" wirksam.

### Hinzufügen weiterer Regelkreise

Wenn Sie einen oder mehrere Regelkreise in den DB "DB\_LOOP" einfügen wollen, öffnen Sie diesen DB mit dem DB-Editor. Im Menü "Ansicht" wählen Sie die Deklarationssicht. Jetzt können Sie den ARRAY-Bereich der Variablen ändern, z. B. 1, ... 4 anstatt 1, ... 3. (Auf dieselbe Weise können Sie Regelkreise auch wieder entfernen.)

Nachdem Sie im Menü "Ansicht" wieder auf "Datensicht" umgeschaltet haben, müssen Sie nun die Variable GLP\_NBR anpassen und die gewünschte Abtastzeit für jeden Regelkreis überprüfen (MAN\_CYC[x],  $x = 1, \dots, \text{GLP\_NBR}$ ). Dabei ist wiederum die unten angegebene Bedingung für MAN\_CYC einzuhalten.

## Aufruf der FC "LP\_SCHED" in Ihrem Programm

Die FC "LP\_SCHED" müssen Sie vor allen Regelkreis-FBs aufrufen.

Bei der Wertzuweisung an die Eingangsparameter beachten Sie das Folgende:

- TM\_BASE: An diesem Eingang geben Sie den Zeittakt des Weckalarm-OB an, in dem die FC "LP\_SCHED" aufgerufen wird.
- COM\_RST: Bei einem Anlauf der CPU müssen Sie die FC "LP\_SCHED" einmal mit COM\_RST = TRUE aufrufen. Sie macht dann einen Initialisierungslauf und nimmt dabei die unter "CPU-Anlauf" beschriebenen Vorbelegungen vor. Im zyklischen Betrieb (Weckalarm) müssen Sie die FC "LP\_SCHED" mit COM\_RST = FALSE aufrufen.
- DB\_NBR: An diesem Eingang geben Sie die Nummer des DB "DB\_LOOP" an, auf den die FC "LP\_SCHED" zugreifen soll.

Nach dem Aufruf der FC "LP\_SCHED" müssen Sie alle zugehörigen Regelkreis-FBs bedingt aufrufen. Die Bearbeitung eines Regelkreis-FB soll erfolgen, wenn das jeweilige ENABLE-Bit im DB "DB\_LOOP" den Wert TRUE hat. Dieses Bit wurde von der FC "LP\_SCHED" zuvor beschrieben. Falls der Regelkreis-FB bearbeitet wurde, müssen Sie dem ENABLE-Bit nach der Bearbeitung den Wert FALSE zuweisen.

Beim Aufruf der Regelkreis-FBs müssen Sie deren Eingangsparameter COM\_RST und CYCLE mit den Variablen COM\_RST[x] und CYCLE[x] des DB "DB\_LOOP" verschalten. CYCLE[x] enthält die tatsächliche Abtastzeit des Regelkreises x und wird von der FC "LP\_SCHED" bei jedem Durchlauf beschrieben. Falls Sie die im Folgenden genannte Bedingung für die Parametrierung der Variable MAN\_CYC[x] beachtet haben, enthält CYCLE[x] denselben Wert wie MAN\_CYC[x]. Andernfalls steht in CYCLE[x] der Wert, der sich ergibt, wenn man MAN\_CYC[x] auf das nächste ganzzahlige Vielfache von  $TM\_BASE * GLP\_NBR$  abrundet.

Im Folgenden ist ein Beispiel zum Aufruf der FC "LP\_SCHED" und zum bedingten Aufruf eines Regelkreis-FB angegeben.

AWL	
CALL "LP_SCHED"	
TM_BASE:=	Hier wird der Zeittakt des Weckalarms parametrieret. Bsp.: T#100ms oder #CYCLE mit CYCLE = Eingangsparameter des Bausteins, in dem die FC LP_SCHED aufgerufen wird.
COM_RST:=	Hier wird der FC "LP_SCHED" mitgeteilt, ob ein Initialisierungslauf der aufgerufenen Regelkreise stattfinden soll. Bsp.: FALSE oder #COM_RST mit COM_RST = Eingangsparameter des Bausteins, in dem die FC LP_SCHED aufgerufen wird.
DB_NBR	Hier wird die Nummer des DB "DB_LOOP" parametrieret, den die FC "LP_SCHED" bearbeiten soll. Bsp.: "DB_LOOP" mit DB_LOOP = in der Symboltabelle vergebener Name des DB.
U "DB_LOOP".LOOP_DAT[1].ENABLE	
SPBN M002	Regelkreisaufruf, falls ENABLE = TRUE
CALL FBx,DBy	
COM_RST:= "DB_LOOP".LOOP_DAT[1].COM_RST	
:	Formaloperandenliste
:	Formaloperandenliste
CYCLE:= "DB_LOOP".LOOP_DAT[1].CYCLE	
:	Formaloperandenliste
:	Formaloperandenliste
CLR	
= "DB_LOOP".LOOP_DAT[1].ENABLE	ENABLE-Bit zurücksetzen
M002:	weiter im Programm, z. B. bedingter Aufruf des nächsten Regelkreis-FB
:	

### Impulsformer in Verbindung mit LP\_SCHED

Wenn Sie beim kontinuierlichen Regler PID\_CP den Impulsformer eingeschaltet haben, muss statt des Parameters CYCLE die Impulsrasterbreite CYCLE\_P mit dem Parameter LOOP\_DAT[x].CYCLE beschrieben werden.

### Bedingung für die Parametrierung der Abtastzeit

Die FC "LP\_SCHED" kann pro Aufruf maximal einen Regelkreis bearbeiten. Es vergeht daher die Zeit

$$TM\_BASE * GLP\_NBR,$$

bis die erstmalige Bearbeitung sämtlicher Regelkreise abgeschlossen ist. Bei der Parametrierung der gewünschten Abtastzeiten  $MAN\_CYC[x]$  müssen Sie daher für jeden Regelkreis folgende Bedingung einhalten:

Die Abtastzeit des Regelkreises  $x$  muss ein ganzzahliges Vielfaches aus dem Produkt von Zeitbasis und der Anzahl der zu bearbeitenden Regler sein.

$$MAN\_CYC[x] \stackrel{!}{=} GV (TM\_BASE * GLP\_NBR), x = 1, \dots, GLP\_NBR$$

Die tatsächliche Abtastzeit  $CYCLE[x]$  des Regelkreises  $x$  wird von der FC "LP\_SCHED" bei jedem Durchlauf aus  $MAN\_CYC[x]$  wie folgt ermittelt:

- Falls Sie die obige Vorschrift eingehalten haben, ist die tatsächliche Abtastzeit  $CYCLE[x]$  mit der von Ihnen vorgegebenen Abtastzeit  $MAN\_CYC[x]$  identisch.
- Falls Sie die obige Bedingung nicht eingehalten haben, steht in  $CYCLE[x]$  der Wert, der sich ergibt, wenn man  $MAN\_CYC[x]$  auf das nächste ganzzahlige Vielfache von  $TM\_BASE * GLP\_NBR$  abrundet.

### Beispiel für eine Aufrufverteilung

Das folgende Beispiel zeigt die Aufrufreihenfolge von vier Regelkreisen in einem Weckalarm-OB. Pro Einheit der Zeitbasis  $TM\_BASE$  wird maximal ein Regelkreis bearbeitet. Die Aufrufreihenfolge ergibt sich aus der Reihenfolge der Regelkreisdaten im DB "DB\_LOOP".

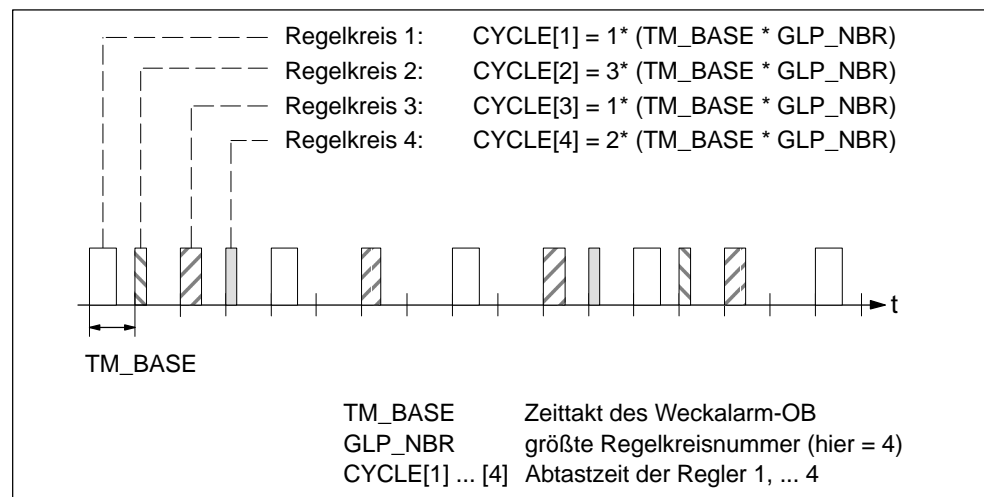


Bild 7-2 Aufrufreihenfolge von vier unterschiedlich häufig bearbeiteten Regelkreisen



## Aufruf von mehr als einem Regelkreis-FB pro Weckalarmzeittakt

Wenn in einem Durchlauf eines Weckalarm-OB mehr als ein Regelkreis bearbeitet werden soll, dann darf die FC "LP\_SCHED" auch mehrmals aufgerufen werden. Alle Aufrufe dieser FC müssen vor dem Aufruf der Regelkreis-FBs erfolgen. Am Eingangsparameter TM\_BASE der FC "LP\_SCHED" müssen Sie dann den Zeittakt des Weckalarm-OB dividiert durch die Anzahl der FC-Aufrufe eingeben.

Beispiel: Die FC LP\_SCHED werde im OB 35 zweimal aufgerufen; der OB 35 werde alle 100 ms bearbeitet. Dann muss der Eingangsparameter TM\_BASE mit 50 ms parametrierung werden.

## Laufzeiten

Bitte beachten Sie, dass die Summe der Laufzeiten der FC "LP\_SCHED" und der Regelkreis-FBs, die in einem Durchlauf eines Weckalarm-OB bearbeitet werden, nicht größer sein darf als der Zeittakt des Weckalarm-OB.

---

### Hinweis

Der Baustein überprüft weder, ob ein globaler DB mit der Nummer DB\_NBR wirklich vorhanden ist, noch, ob der Parameter GLP\_NBR (größte Regelkreisnummer) zur Länge des Datenbausteins passt. Bei inkorrekt Parametrierung geht die CPU mit "internem Systemfehler" in STOP.

---

## Bedienungen im laufenden Betrieb

Folgende Änderungen im DB "DB\_LOOP" sind im laufenden Betrieb erlaubt, wenn dabei nur der jeweilige Parameter geändert und nicht der komplette DB auf die CPU geladen wird:

- Sperren einzelner Regelkreise  
Durch Beschreiben der Variable MAN\_DIS[x] mit dem Wert TRUE wird die Bearbeitung des Regelkreises x im laufenden Betrieb gesperrt. Die FC "LP\_SCHED" setzt das ENABLE-Bit dieses Regelkreises so lange nicht mehr auf TRUE, bis Sie MAN\_DIS[x] wieder mit dem Wert FALSE beschreiben.
- Initialisieren eines Regelkreises  
Durch Beschreiben der Variable MAN\_CRST[x] mit dem Wert TRUE können Sie einen einzelnen Regelkreis neu starten: In diesem Fall beschreibt die FC "LP\_SCHED" bei der nächsten Bearbeitung des Regelkreises x die Variable COM\_RST[x] mit TRUE. Bei der übernächsten Bearbeitung dieses Regelkreises beschreibt die FC "LP\_SCHED" die Variablen MAN\_CRST[x] und COM\_RST[x] mit FALSE.
- Änderung der Abtastzeit eines Regelkreises  
Den Parameter MAN\_CYC[x] des DB "DB\_LOOP" dürfen Sie im laufenden Betrieb ändern.

### Hinweis

Wird ein Regelkreis eingefügt oder gelöscht, d. h. wird der ganze DB "DB\_LOOP" neu auf die CPU geladen, ohne dass die CPU einen Anlauf durchführen soll, dann müssen Sie die internen Regelkreiszähler ILP\_COU[x], x = 1, ... GLP\_NBR und den Parameter für die aktuelle Regelkreisnummer ALP\_NBR mit Null vorbelegen.

---

### CPU-Anlauf

Bei einem Anlauf der CPU müssen Sie die FC "LP\_SCHED" aus dem zugehörigen Anlauf-OB heraus aufrufen und dabei dem Eingang COM\_RST den Wert TRUE zuweisen. Im Weckalarm-OB müssen Sie diesem Eingang wieder den Wert FALSE zuweisen. Die FC "LP\_SCHED" verfügt über eine Initialisierungsroutine, die gestartet wird, wenn der Eingangs-Parameter COM\_RST den Wert TRUE hat. In diesem Initialisierungslauf werden die folgenden Vorbelegungen im DB "DB\_LOOP" vorgenommen:

- Aktuelle Regelkreisnummer: ALP\_NBR = 0
- Freigabe: ENABLE[x] = NOT MAN\_DIS[x], x = 1, ... GLP\_NBR
- Abtastzeit: CYCLE[x] erhält den Wert, der sich ergibt, wenn man MAN\_CYC[x] auf das nächste ganzzahlige Vielfache von TM\_BASE \* GLP\_NBR abrundet, x = 1, ... GLP\_NBR
- Regelkreisinitialisierung: COM\_RST[x] = TRUE, x = 1, ... GLP\_NBR
- Interner Regelkreiszähler: ILP\_COU[x] = 0, x = 1, ... GLP\_NBR

Nach dem Aufruf der FC "LP\_SCHED" im Anlauf-OB rufen Sie dort die Regelkreis-FBs bedingt auf, damit diese ihre Initialisierungen vornehmen.

### Beobachtung der FC "LP\_SCHED"

In die Variable ALP\_NBR des DB "DB\_LOOP" trägt die FC "LP\_SCHED" die Nummer des nächsten zu bearbeitenden Regelkreises ein. Die Nummer des jeweiligen Regelkreises ergibt sich aus der Plazierung seiner Aufrufdaten in der Reihenfolge der Eintragungen im DB (siehe Tabelle 7-1).

Die Variable ILP\_COU[x] ist der interne Regelkreiszähler der FC "LP\_SCHED". Er enthält die Zeitdauer bis zum nächsten Aufruf des entsprechenden Regelkreises. Die Zeiteinheit von ILP\_COU ist das Produkt aus Zeitbasis TM\_BASE und Anzahl der Regelkreise GLP\_NBR. Ist ILP\_COU = 0, dann setzt die FC "LP\_SCHED" das ENABLE-Bit des betreffenden Regelkreises.

### Wenn Regelkreise wider Erwarten nicht aufgerufen werden

Wenn die Funktion "LP\_SCHED" aufgerufen wird, aber einzelne Regelkreise nicht bearbeitet werden, kann dies folgende Ursachen haben:

- Durch Beschreiben der Variable MAN\_DIS[x] mit dem Wert TRUE ist die Regelkreisbearbeitung im laufenden Betrieb gesperrt.
- Im Parameter GLP\_NBR ist die Anzahl der FBs bzw. der Regelkreise, die von der FC "LP\_SCHED" bearbeitet werden sollen, zu klein angegeben.
- Die von Ihnen vorgegebenen Abtastzeiten MAN\_CYC[x] der einzelnen Regelkreise dürfen nicht kleiner sein als das Produkt aus Zeitbasis TM\_BASE und Anzahl der Regelkreise GLP\_NBR. Regelkreise, bei denen diese Bedingung verletzt wird, werden nicht bearbeitet.

### Parameter der FC "LP\_SCHED"

Die Funktion "LP\_SCHED" steuert den Aufruf einzelner Regler innerhalb eines Weckalarm-OB.

Die Werte der Eingangsparameter werden im Baustein nicht begrenzt. Eine Prüfung der Parameter findet nicht statt.

Eingangsparameter			LP_SCHED	Ausgangsparameter		
Parameter	Typ	*)		Parameter	Typ	
TM_BASE	TIME	100 ms				
COM_RST	BOOL	FALSE				
DB_NBR	BLOCK_DB	DB 1				
*) Vorbelegung bei Neuerstellung des Instanz-DB						

Bild 7-3 Blockschaltbild und Parameter der Funktion LP\_SCHED

## 7.2 Example1: Schrittreger mit Streckensimulation

### Anwendung

Das Beispiel Example1 umfasst einen Standard-Schrittreger (PID\_ES) in Verbindung mit einer simulierten Regelstrecke, die aus einem integrierenden Stellglied und einem nachgeschalteten Verzögerungsglied dritter Ordnung (PT3) besteht.

Mit Hilfe von Example1 ist es möglich, auf einfache Weise einen Schrittreger zu generieren und diesen in allen Eigenschaften im Offline-Zusammenspiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

Das Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit diskontinuierlichem Ausgang, wie sie bei der Regelung von Strecken mit motorischen Stellgliedern sehr häufig eingesetzt werden, leicht zu verstehen. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Durch entsprechende Wahl ihrer Parameter nähern Sie die Strecke an die Eigenschaften des realen Prozesses an. Mit Hilfe des Konfigurationswerkzeugs kann dann durch Identifikation der Modellstrecke ein Satz von geeigneten Reglerkenn-daten gefunden werden.

### Funktionen von Example1

Das Beispiel Example1 setzt sich im Wesentlichen aus den beiden Funktionsbausteinen PID\_ES und PROC\_S zusammen. PID\_ES verkörpert dabei den verwendeten Standard-Regler, und PROC\_S simuliert eine Regelstrecke mit den Funktionsgliedern "Ventil" und PT3 (Bild 7-4). Dem Regler werden dabei neben der Regelgröße Informationen über die Position des Stellgliedes und ggf. erreichte An-schlagsignale übermittelt.

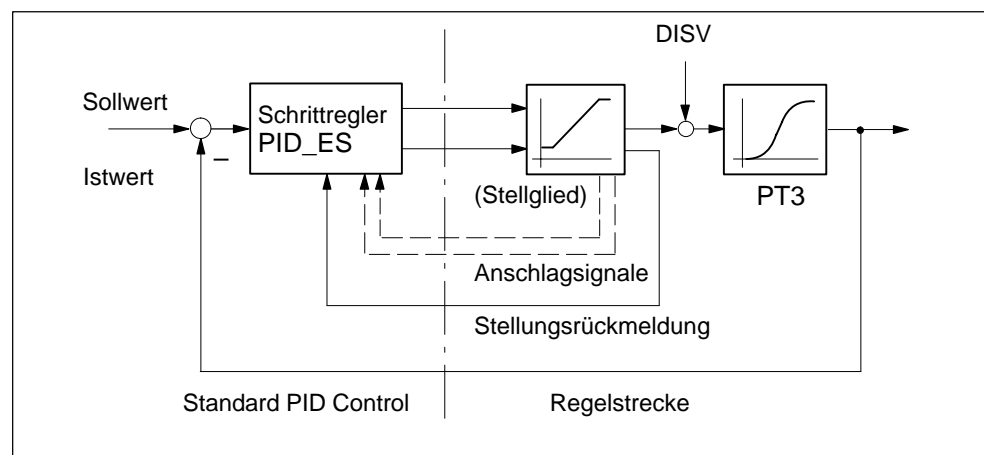


Bild 7-4 Beispiel Example1, Regelkreis

Der Funktionsbaustein PROC\_S bildet eine Reihenschaltung nach, die aus dem integrierenden Stellglied und drei Verzögerungsgliedern 1. Ordnung besteht (Bild 7-5). Zum Ausgangssignal des Stellgliedes wird immer die Störgröße **DISV** hinzuaddiert, sodass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können. Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen.

Der Parameter für die Motorstellzeit **MTR\_TM** definiert die Zeit, welche das Stellglied für den Durchlauf von Anschlag zu Anschlag benötigt.

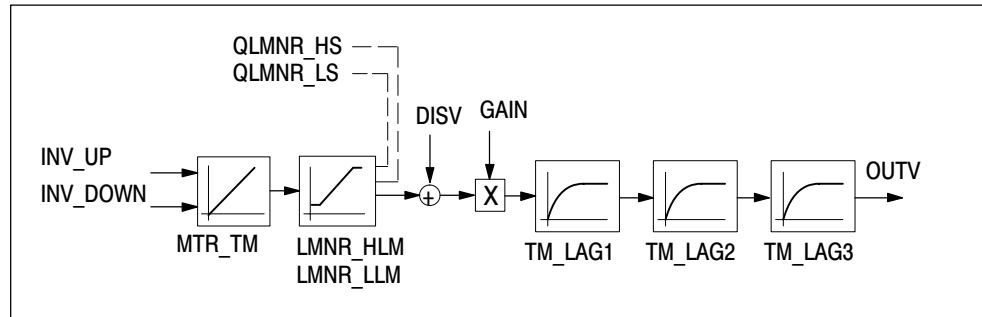


Bild 7-5 Aufbau und Parameter des Strecken-Bausteins PROC\_S

### Bausteinstruktur

Das Beispiel Example1 ist aus der Funktion APP\_1, die die Bausteine für den Regler und die simulierte Strecke umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-2 Bausteine des Beispiels Example1

Baustein	Name (in der Symboleiste)	Beschreibung
OB 100		Neustart-OB
OB 35		Zeitgesteuerter OB: 100 ms
FC 100	APP_1	Beispiel 1
FB 2	PID_ES	Schrittregler
FB 100	PROC_S	Strecke für Schrittregler
DB 100	PROCESS	Instanz-DB zu PROC_S
DB 101	CONTROL	Instanz-DB zu PID_ES

Den beiden Funktionsbausteinen (Bild 7-6) sind die Instanz-Datenbausteine DB 100 für die Strecke und DB 101 für den Regler zugeordnet.

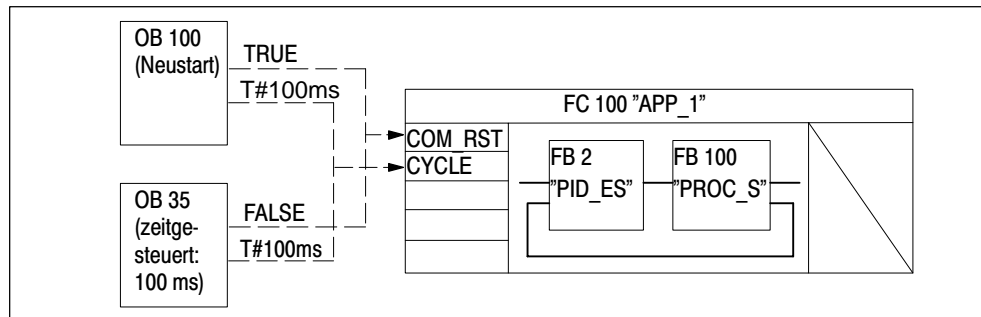


Bild 7-6 Bausteine des Beispiels 1: Verschaltung und Aufruf

## Die Parameter des Streckenmodells

Die Parameter des Reglerbausteins PID\_ES und ihre Bedeutung sind in Kapitel 6 beschrieben. Die Parameter des Streckenbausteins PROC\_S sind in der folgenden Tabelle aufgelistet.

Tabelle 7-3 Parameter des Streckenbausteins "PROC\_S" (DB100: FB100)

Parameter	Typ	Wertebereich	Beschreibung
INV_UP	BOOL		Eingangssignal auf (mehr)
INV_DOWN	BOOL		Eingangssignal ab (weniger)
COM_RST	BOOL		Neustart
CYCLE	TIME	$\geq 1\text{ms}$	Abtastzeit
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkung
MTR_TM	TIME		Motorstellzeit
LMNR_HLM	REAL	LMNR_LLM ... 100.0 [%]	obere Grenze der Stellungsrückmeldung
LMNR_LLM	REAL	-100.0...LMNR_HLM [%]	untere Grenze der Stellungsrückmeldung
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 3
OUTV	REAL		Ausgangsgröße
LMNR	REAL		Stellungsrückmeldung
QLMNR_HS	BOOL		Stellglied am oberen Anschlag
QLMNR_LS	BOOL		Stellglied am unteren Anschlag

Nach **Neustart** werden die Ausgangsgröße OUTV sowie alle internen Speichergrößen auf Null gesetzt.

### Verschaltung und Aufruf des Beispiels Example1

Wie der Schrittreger über die Funktion FC 100 intern mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus Bild 7-7 hervor.

Selbstverständlich lässt sich durch Öffnen der Verbindung LMNR – LMNR\_IN auch eine Schrittregerung ohne Stellungsrückmeldung realisieren.

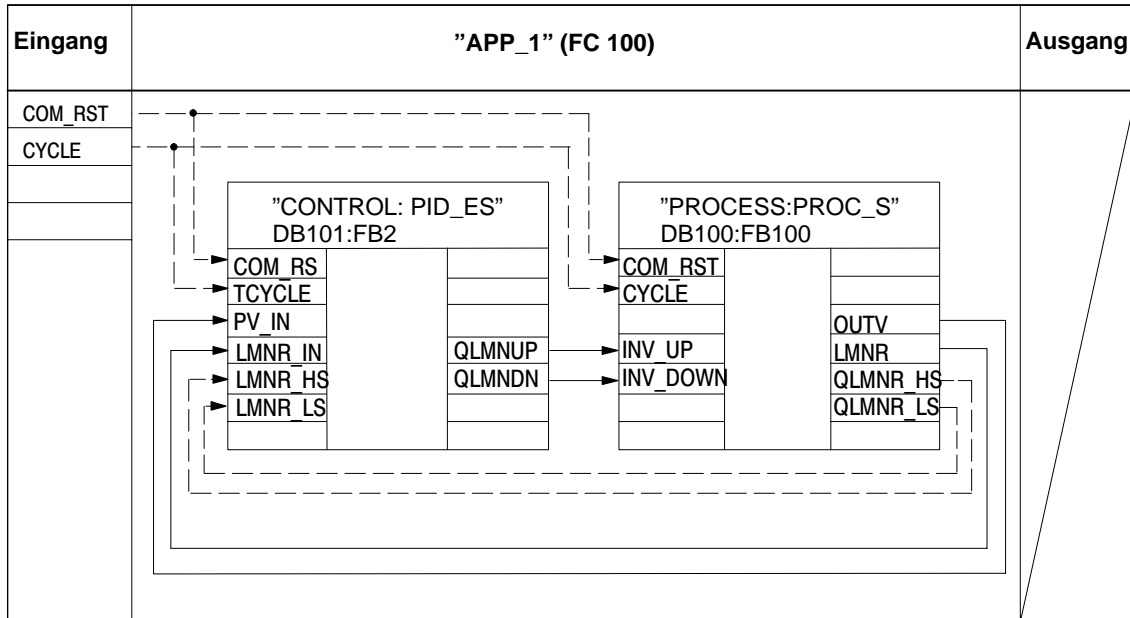


Bild 7-7 Verschaltung und Aufruf der FC 100 (APP\_1)

**Parameter der Modellregelstrecke für Schrittregler**

Bild 7-8 zeigt das Funktionsschema und die Parameter der Regelstrecke.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in *Kapitel 3.5* beschrieben.

Eingangsparameter			PROC_S (FB 100)	Ausgangsparameter			
Parameter	Typ	*)		Signal	Typ	*)	
COM_RST	BOOL	FALSE					
CYCLE	TIME	T#1s					
GAIN	REAL	0.0					
DISV	REAL	0.0					
INV_UP	BOOL	FALSE			QLMNR_HS	BOOL	FALSE
INV_DOWN	BOOL	FALSE			OUTV	REAL	0.0
					QLMNR_LS	BOOL	FALSE
					LMNR	REAL	0.0
LMNR_HLM	REAL	100.0					
LMNR_LLM	REAL	0.0					
MTR_TM	TIME	T#30s					
TM_LAG1	TIME	T#10s					
TM_LAG2	TIME	T#10s					
TM_LAG3	TIME	T#10s					

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 7-8 Funktionsschema und Parameter des Streckenmodells PROC\_S

**Parameter und Sprungantwort**

Anhand einer konkreten Parametrierung des Schrittreglers mit PI-Wirkung und eingeschalteter Totzone wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke 3. Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit bilden annähernd das Verhalten eines schnellen Temperaturprozesses oder einer Füllstandsregelung nach.

Einstellung einer der Verzögerungszeiten  $TM\_LAGx = 0$  s reduziert die Ordnung der Strecke um ein Grad.



Das Kurvendiagramm (Konfigurationswerkzeug) zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Sollwertänderung von 60 Prozent (Bild 7-9). Die Tabelle enthält die aktuell eingestellten Werte der relevanten Parameter für Regler und Strecke.

Parameter	Typ	Parametrierung	Beschreibung
<b>Regler:</b>			
CYCLE	TIME	100ms	Abtastzeit
GAIN	REAL	0.31	Proportionalbeiwert
TI	TIME	19.190s	Integrationszeit
MTR_TM	TIME	20s	Motorstellzeit
PULSE_TM	TIME	100ms	Mindestimpulsdauer
BREAK_TM	TIME	100ms	Mindestpausendauer
DEADB_ON	BOOL	TRUE	Totzone einschalten
DEADB_W	REAL	0.5	Totzonbreite
<b>Regelstrecke:</b>			
GAIN	REAL	1.5	Streckenverstärkung
MTR_TM	TIME	20s	Motorstellzeit
TM_LAG1	TIME	10s	Verzögerungszeit 1
TM_LAG2	TIME	10s	Verzögerungszeit 2
TM_LAG3	TIME	10s	Verzögerungszeit 3

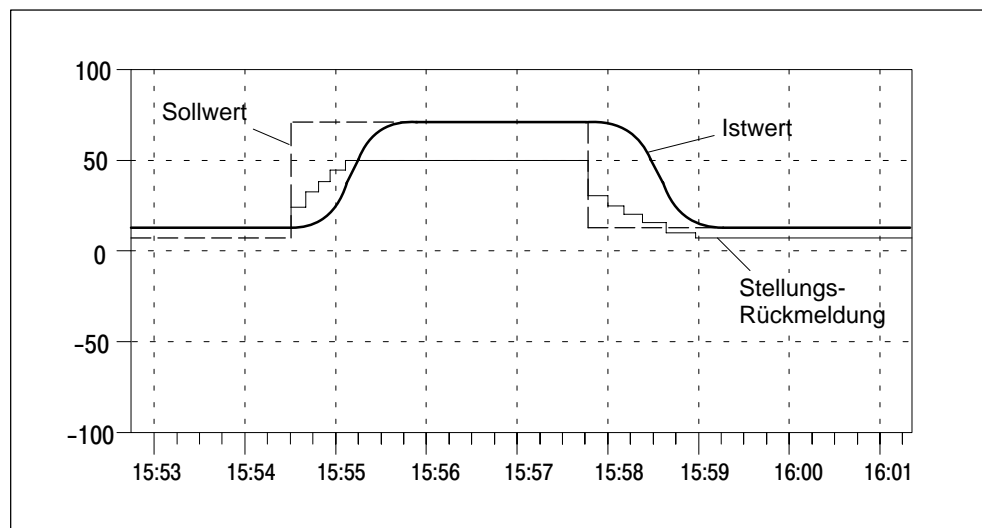


Bild 7-9 Regelkreis mit Schrittregler nach Sollwertsprung

## 7.3 Example2: Kontinuierlicher Regler mit Streckensimulation

### Anwendung

Das Beispiel Example2 umfasst einen kontinuierlichen Standard-Regler (PID\_CP) in Verbindung mit einer simulierten Regelstrecke, die aus einem Verzögerungsglied dritter Ordnung (PT3) besteht.

Mit Hilfe von Example2 ist es möglich, auf einfache Weise einen kontinuierlichen PID-Regler zu generieren und diesen in allen Eigenschaften im Offline-Zusammenpiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

Das Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit analogem Ausgangssignal, wie sie bei der Regelung von Strecken mit proportional wirkenden Stellgliedern eingesetzt werden, leicht zu verstehen. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Nach Annäherung der Strecke an die Eigenschaften des realen Prozesses durch entsprechende Wahl ihrer Parameter kann durch Identifikation der Modellstrecke mit Hilfe des Konfigurationswerkzeugs ein Satz von geeigneten Reglerkenndaten gefunden werden.

### Funktionen von Example2

Das Beispiel Example2 setzt sich im Wesentlichen aus den beiden Funktionsbausteinen PID\_CP (FB 1) und PROC\_C (FB 100) zusammen. PID\_CP verkörpert dabei den verwendeten Standard-Regler, und PROC\_C simuliert eine Regelstrecke mit Ausgleich dritter Ordnung (Bild 7-10).

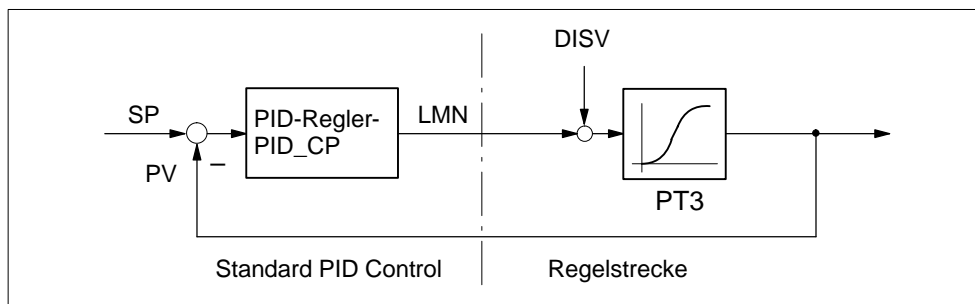


Bild 7-10 Beispiel Example2, Regelkreis

Der Funktionsbaustein PROC\_C bildet eine Reihenschaltung aus drei Verzögerungsgliedern 1. Ordnung nach (Bild 7-11). Zum Ausgangssignal des Stellgliedes wird immer die Störgröße **DISV** hinzuaddiert, sodass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können. Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen.

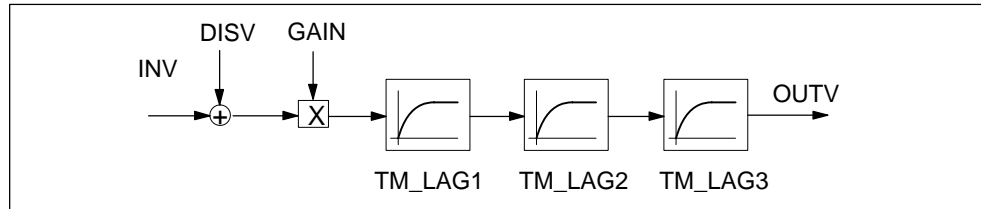


Bild 7-11 Aufbau und Parameter des Strecken-Bausteins PROC\_C

### Bausteinstruktur

Das Beispiel Exampel2 ist aus der Funktion APP\_2, die die Bausteine für den Regler und die simulierte Strecke umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-4 Bausteine des Beispiels Example2

OB 100		Neustart-OB
OB 35		Zeitgesteuerter OB: 100 ms
FC 100	APP_2	Beispiel 2
FB 1	PID_CP	Kontinuierlicher PID-Regler
FB 100	PROC_C	Strecke für kontinuierlichen Regler
DB 100	PROCESS	Instanz-DB zu PROC_C
DB 101	CONTROL	Instanz-DB zu PID_CP

Den beiden Funktionsbausteinen (Bild 7-12) sind die Instanz-Datenbausteine DB 100 für die Strecke und DB 101 für den Regler zugeordnet.

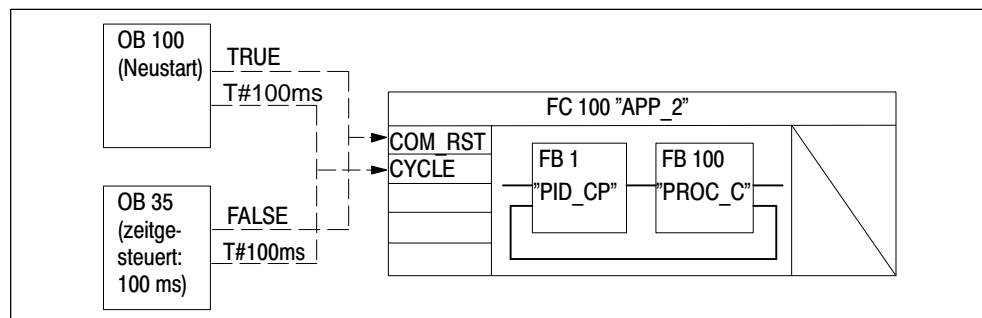


Bild 7-12 Bausteine des Beispiels Example2: Verschaltung und Aufruf

## Die Parameter des Streckenmodells

Die Parameter des Reglerbausteins PID\_CP und ihre Bedeutung sind in *Kapitel 6* beschrieben. Die Parameter des Streckenbausteins PROC\_C sind in der folgenden Tabelle aufgelistet.

Tabelle 7-5 Parameter des Streckenbausteins "PROC\_C" (DB 100: FB 100)

Parameter	Typ	Wertebereich	Beschreibung
INV	REAL		Eingangsgröße
COM_RST	BOOL		Neustart
CYCLE	TIME	$\geq 1\text{ms}$	Abtastzeit
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkungsfaktor
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 3
OUTV	REAL		Ausgangsgröße

## Verschaltung und Aufruf des Beispiels Example2

Wie der kontinuierliche Regler über die Funktion FC 100 intern mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus Bild 7-13 hervor.

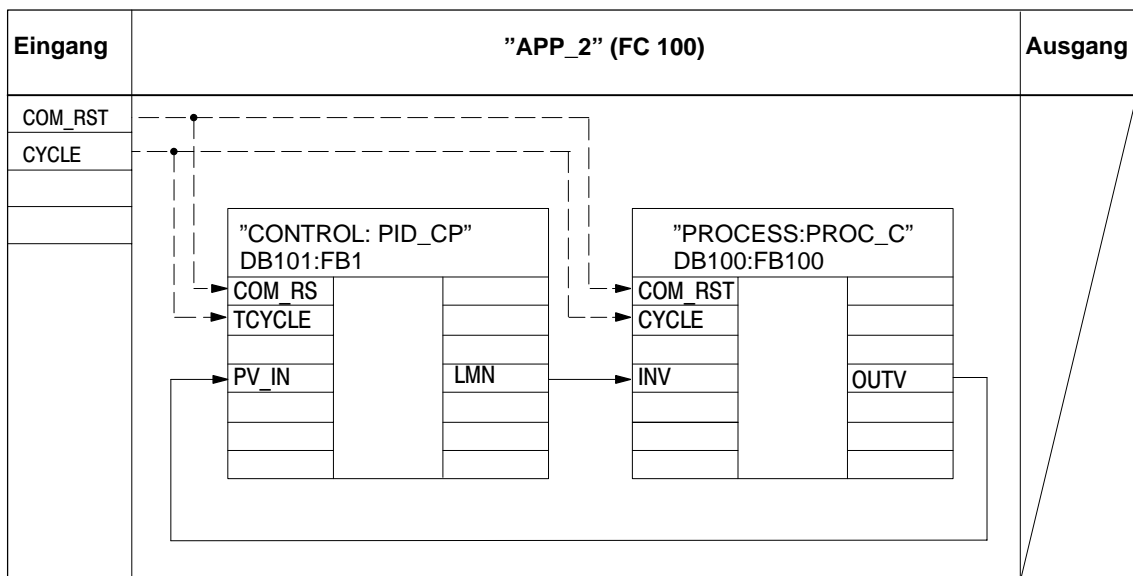


Bild 7-13 Verschaltung und Aufruf der FC 100 (APP\_2)

### Parameter der Modellregelstrecke für kontinuierliche Regler

Bild 7-14 zeigt das Funktionsschema und die Parameter der Regelstrecke.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in *Kapitel 3.5* beschrieben.

Eingangsparameter			PROC_C (FB 100)	Ausgangsparameter		
Parameter	Typ	*)		Signal	Typ	*)
COM_RST	BOOL	FALSE				
CYCLE	TIME	T#1s				
GAIN	REAL	0.0				
DISV	REAL	0.0				
INV	REAL	0.0			OUTV	REAL
						0.0
TM_LAG1	TIME	T#10s				
TM_LAG2	TIME	T#10s				
TM_LAG3	TIME	T#10s				

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 7-14 Funktionsschema und Parameter des Streckenmodells PROC\_C

### Parameter und Sprungantwort

Anhand einer konkreten Parametrierung eines kontinuierlichen Reglers mit PID-Wirkung wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke 3. Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit bilden annähernd das Verhalten einer Druckregelung oder einer Füllstandsregelung nach.

Einstellung einer der Verzögerungszeiten  $TM\_LAGx = 0$  s reduziert die Ordnung der Strecke um ein Grad.

Das Kurvendiagramm (Konfigurationswerkzeug) zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Reihe von Sollwertänderungen von jeweils 20 Prozent des Messbereiches (Bild 7-15). Die Tabelle enthält die aktuell eingestellten Werte der relevanten Parameter für Regler und Strecke.

Parameter	Typ	Parametrierung	Beschreibung
<b>Regler:</b>			
CYCLE	TIME	100ms	Abtastzeit
GAIN	REAL	0.31	Proportionalbeiwert
TI	TIME	22.720s	Integrationszeit
TD	TIME	5.974s	Differenzierzeit
TM_LAG	TIME	1.195s	Verzögerungszeit des D-Anteils
<b>Regelstrecke:</b>			
GAIN	REAL	1.5	Streckenverstärkung
TM_LAG1	TIME	10s	Verzögerungszeit 1
TM_LAG2	TIME	10s	Verzögerungszeit 2
TM_LAG3	TIME	10s	Verzögerungszeit 3

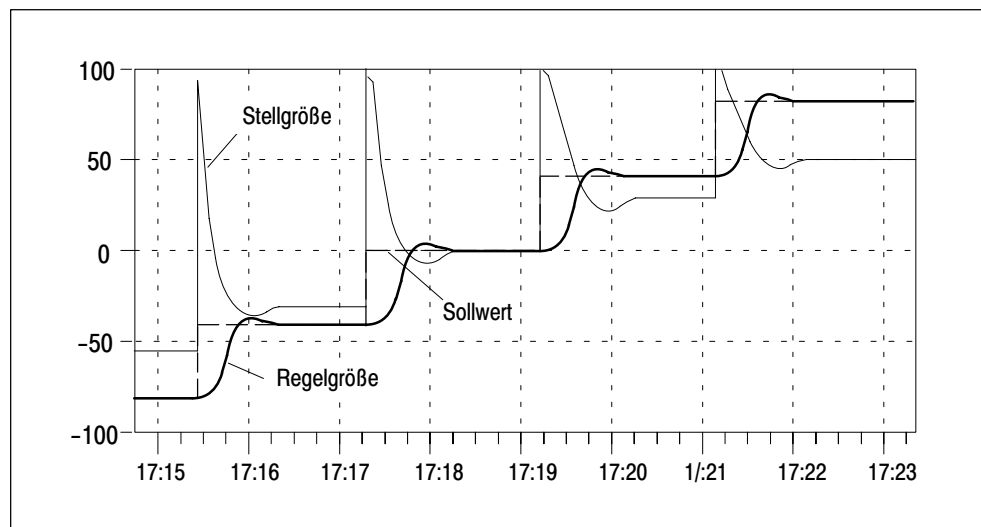


Bild 7-15 Regelung mit kontinuierlichem Regler und Sollwertsprüngen über den ganzen Messbereich

## 7.4 Example3: Mehrschleifige Verhältnisregelung

### Anwendung

Das Beispiel Example3 enthält all die Bausteine, die zur Konfiguration einer zweischleifigen Verhältnisregelung erforderlich sind.

Mit Hilfe von Example3 ist es möglich, auf einfache Weise eine bei Verbrennungsprozessen häufig benötigte Verhältnisregelung für zwei Komponenten zu generieren. Die Struktur lässt sich leicht auf die Regelung von mehr als zwei Regelgrößen in konstantem Verhältnis zueinander erweitern.

### Funktionen von Example3

Das Beispiel Example3 umfasst den Regleraufrufverteiler (LP\_SCHED) mit zugehörigem globalen Datenbaustein (DB-LOOP) sowie den Funktionsbaustein (FB 1) für kontinuierliche Standard-Regler mit zwei Instanz-DBs für die Konfigurationsdaten der beiden Regler.

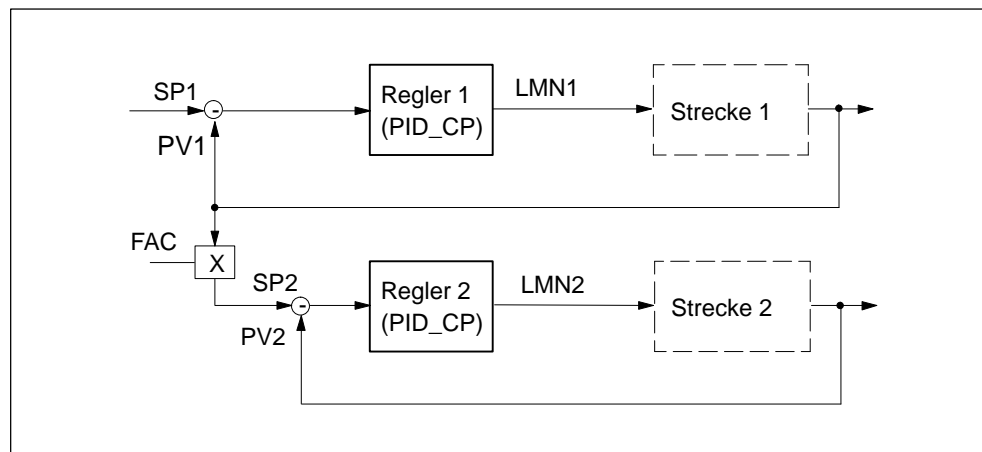


Bild 7-16 Verhältnisregelung mit zwei Regelkreisen (Example3)

Die Regler (Bild 7-16) werden über den Regleraufrufverteiler aus der Weckalarmebene mit 100 ms Zeittakt zu jeweils zyklischen Zeitpunkten aufgerufen.

Der Regler 1 wirkt dabei als Führungsregler für die Einstellung des Sollwertes zur Regelung der zweiten Prozessgröße. Dadurch bleibt das Verhältnis zwischen PV1 und PV2 auch dann konstant, wenn durch Störeinflüsse die Prozessgröße PV1 dynamisch schwankt.

## Baueinstruktur

Das Beispiel Example3 ist aus der Funktion APP\_3, die die Bausteine für den Aufrufverteiler und die beiden Regler umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-6 Bausteine des Beispiels Example3

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 100		Neustart-OB
OB 35		Zeitgesteuerter OB: 100 ms
FC 100	APP_3	Beispiel 3
FC 1	LP_SCHED	Regleraufrufverteiler
FB 1	PID_CP	Kontinuierlicher PID-Regler
DB 1	DB_LOOP	Glob. DB für Aufrufdaten zu LP_SCHED
DB 100	CONTROL1	1. Instanz-DB zu PID_CP
DB 101	CONTROL2	2. Instanz-DB zu PID_CP

Dem Funktionsbaustein PID\_CP (FB 1) sind die beiden Instanz-Datenbausteine DB 100 und DB 101 für die Realisierung der zwischenschleifigen Verhältnisregelung zugeordnet.

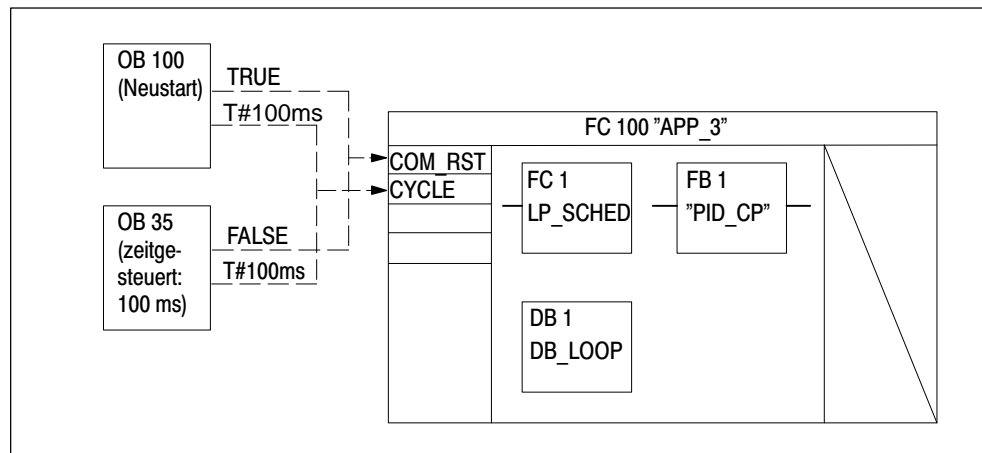


Bild 7-17 Bausteine des Beispiels Example3: Verschaltung und Aufruf



### Parametrierung des Beispiels Example3

Wie die PID-Regler über die Funktion FC 100 intern mit dem Regleraufrufverteiler und untereinander verschaltet sind, geht aus Bild 7-18 hervor.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in *Kapitel 3.5* beschrieben.

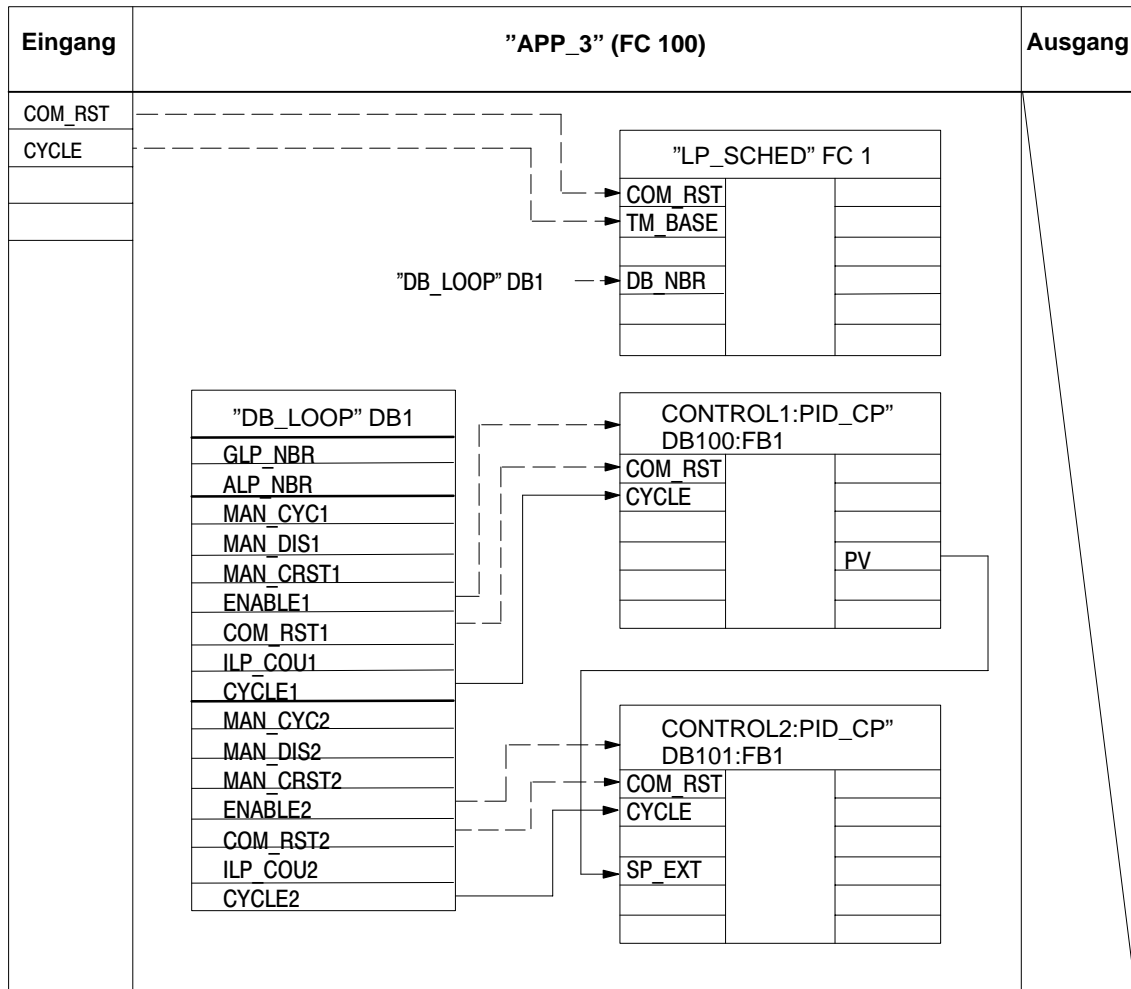


Bild 7-18 Schaltbild und Parameter der Funktion APP\_3

## 7.5 Example4: Mischungsregelung

### Anwendung

Das Beispiel Example4 enthält all die Bausteine, die zur Konfiguration einer Mischungsregelung mit einer Haupt- und zwei Nebenkomponten benötigt werden.

Mit Hilfe von Example4 ist es möglich, auf einfache Weise eine bei Mischprozessen erforderliche Regelung der Gesamtmenge mit konstanten Anteilen der jeweils in die Mischung eingehenden Einzelmengen (für drei Komponenten) zu generieren. Die Struktur lässt sich leicht auf die Regelung von mehr als drei Mischkomponenten erweitern.

### Funktionen von Example4

Das Beispiel Example4 enthält den Regleraufrufverteiler (LP\_SCHED) mit zugehörigem globalen Datenbaustein (DB-LOOP) sowie den Funktionsbaustein (FB 1) für kontinuierliche Standard-Regler und den Funktionsbaustein (FB 2) für Schrittreger zusammen mit vier Instanz-DBs für die Konfigurationsdaten der vier Regler.

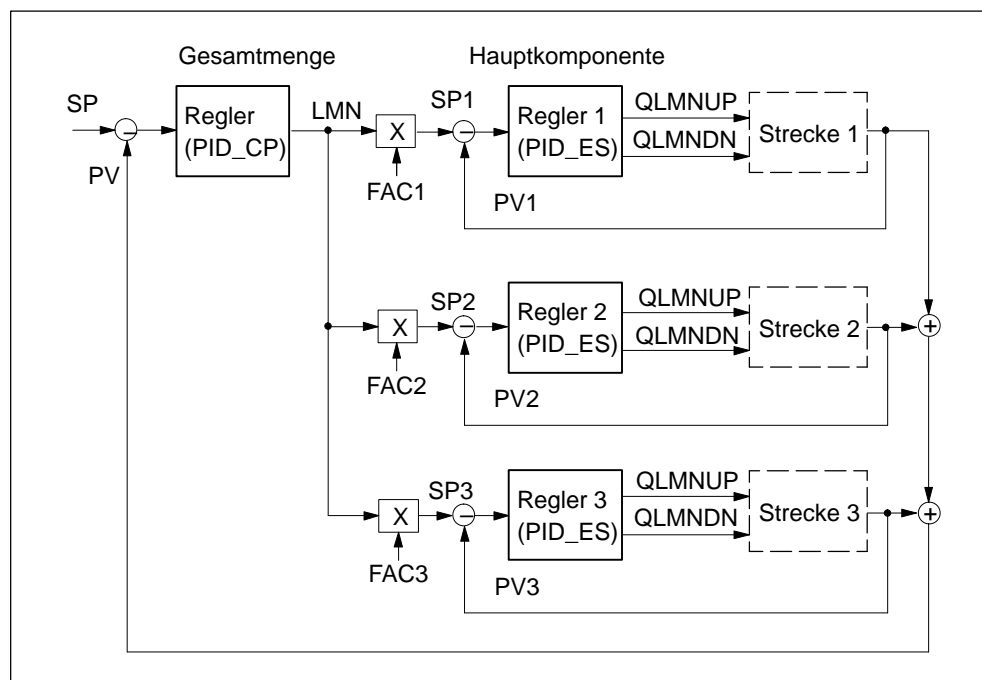


Bild 7-19 Mischungsregelung für drei Komponenten (Example4)

Die vier Regler werden über den Regleraufrufverteiler aus der Weckalarmebene mit 100 ms Zeittakt zu jeweils zyklischen Zeitpunkten aufgerufen.

Der Regler für die Gesamtmenge mit kontinuierlichem Ausgang (PID\_CP) wirkt dabei als Führungsregler auf die Einstellung der Sollwerte, d. h. auf die Menge der jeweiligen Komponente. Die Mengen der Hauptkomponente und der zwei Nebenkomponten werden in Example4 durch Schrittreger (PID\_ES) entsprechend den Anteilseinstellungen an FAC1...3 geregelt. Dabei ist Bedingung, dass sich die für die Mischungsfaktoren FAC1...FAC3 parametrisierten Werte zu 100 % addieren.

## Bausteinstruktur

Das Beispiel Example4 ist aus der Funktion APP\_4, die die Bausteine für den Aufrufverteiler und die vier Regler umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-7 Bausteine des Beispiels Example4

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 100		Neustart-OB
OB 35		Zeitgesteuerter OB: 100 ms
FC 100	APP_4	Beispiel 4
FC 1	LP_SCHED	Regleraufrufverteiler
FB 1	PID_CP	Kontinuierlicher PID-Regler
FB 2	PID_ES	Schrittregler
DB 1	DB_LOOP	Glob. DB für Aufrufdaten zu LP_SCHED
DB 100	CONT_C1	Instanz-DB zu PID_CP
DB 101	CONT_S1	1. Instanz-DB zu PID_ES
DB 102	CONT_S2	2. Instanz-DB zu PID_ES
DB 103	CONT_S3	3. Instanz-DB zu PID_ES

Dem Funktionsbaustein PID\_ES (FB 2) sind drei Instanz-Datenbausteine (DB 101, DB 102 und DB 103) für die Realisierung der Mengenregelungen der drei Einzelkomponenten zugeordnet.

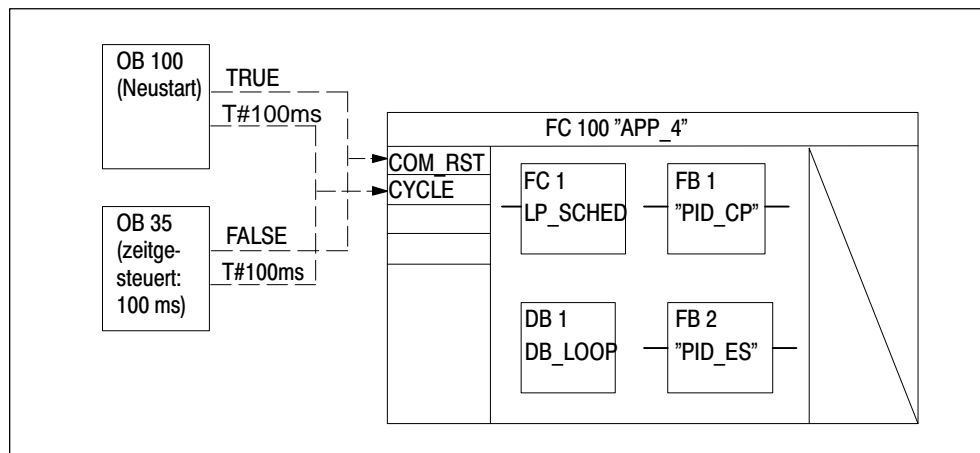


Bild 7-20 Bausteine des Beispiels Example4: Verschaltung und Aufruf

### Parametrierung des Beispiels Example4

Wie die Regler über die Funktion FC 100 intern mit dem Regleraufrufverteiler und untereinander verschaltet sind, geht aus Bild 7-21 hervor.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in Kapitel 3.5 beschrieben.

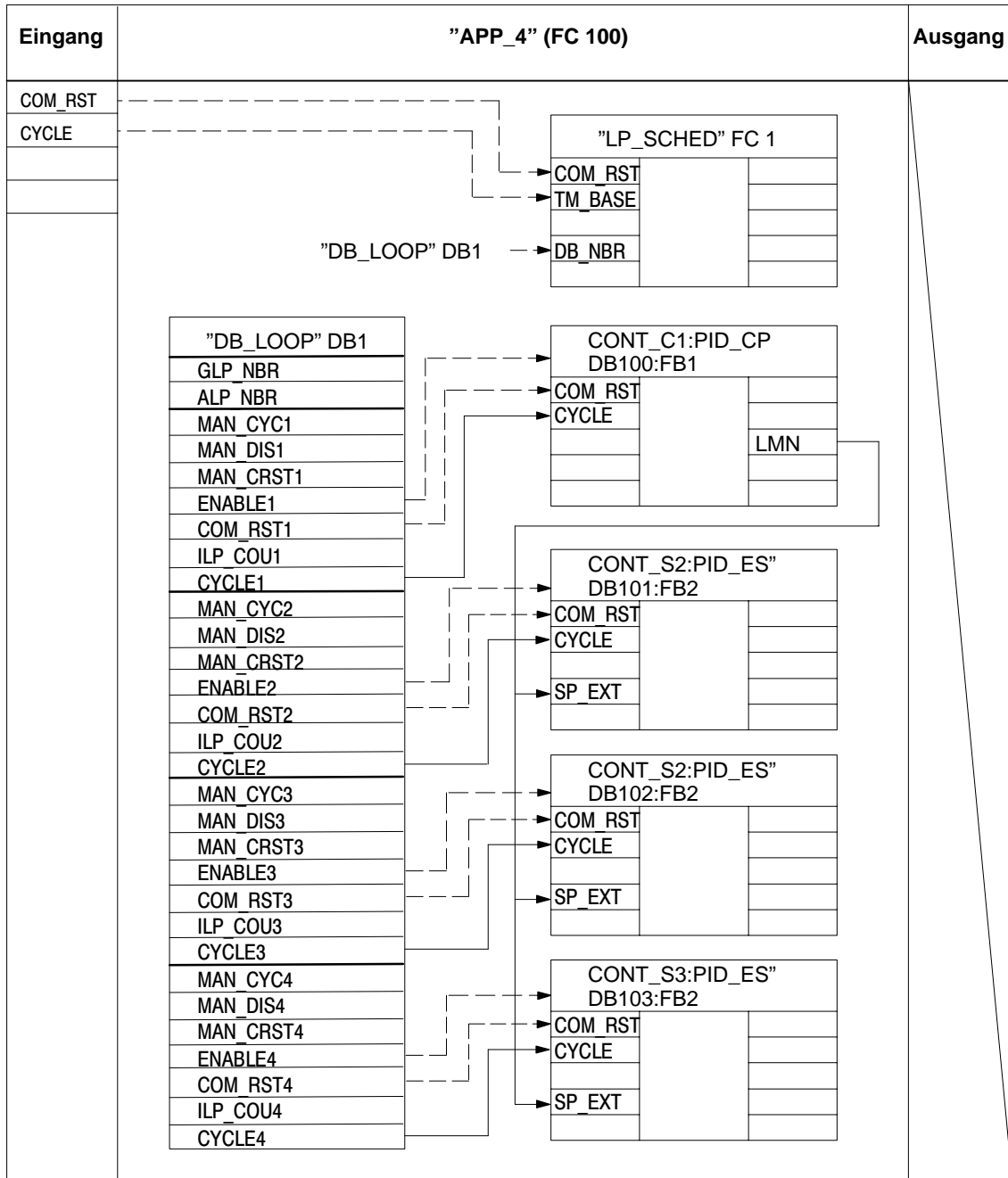


Bild 7-21 Schaltbild und Parameter der Funktion APP\_4

## 7.6 Example5: Kaskadenregelung

### Anwendung

Das Beispiel Example5 enthält alle die Bausteine, die zur Konfiguration einer Kaskadenregelung mit einer Haupt- und einer Hilfsregelgröße benötigt werden.

Mit Hilfe von Example5 ist es möglich, auf einfache Weise eine Kaskadenregelung mit einem Führungs- und einem Folgeregelkreis zu generieren. Die Struktur lässt sich leicht auf mehr als einen Folgeregelkreis erweitern.

### Funktionen von Example5

Das Beispiel Example5 enthält den Regleraufrufverteiler (LP\_SCHED) mit zugehörigem globalen Datenbaustein (DB-LOOP), die Funktionsbausteine FB 1 für den kontinuierlichen Standard-Regler (Führungsregler) sowie FB 2 für den Schrittreger (Folgeregler) mit den beiden Instanz-DBs für die Konfigurationsdaten der Regler.

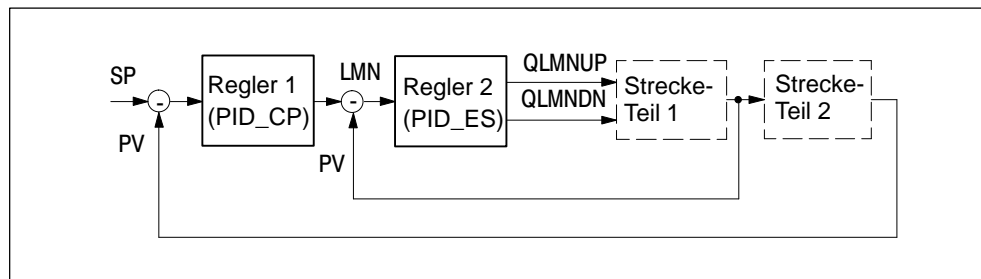


Bild 7-22 Zweiseifige Kaskadenregelung (Example5)

Die Regler werden über den Regleraufrufverteiler aus der Weckalarmebene mit 100 ms Zeittakt zu jeweils zyklischen Zeitpunkten aufgerufen.

Der Regler mit kontinuierlichem Ausgang (PID\_CP) wirkt dabei als Führungsregler auf den Sollwert des Folgereglers in der Weise, dass die Hauptregelgröße am Ausgang von Streckenteil 2 auf dem Führungswert SP gehalten wird. Störungen, die auf den Streckenteil 1 einwirken, regelt der Schrittreger im Hilfsregelkreis (PID\_ES) ohne Beeinflussung der Hauptregelgröße PV aus.

## Baueinstruktur

Das Beispiel Example5 ist aus der Funktion APP\_5, die die Bausteine für den Aufrufverteiler und die beiden Regler umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-8 Bausteine des Beispiels Example5

Bauein	Name (in der Symbolleiste)	Beschreibung
OB 100		Neustart-OB
OB 35		Zeitgesteuerter OB: 100 ms
FC 100	APP_5	Beispiel 5
FC 1	LP_SCHED	Regleraufrufverteiler
FB 1	PID_CP	Kontinuierlicher PID-Regler
FB 2	PID_ES	Schrittregler
DB 1	DB_LOOP	Glob. DB für Aufrufdaten zu LP_SCHED
DB 100	CONT_C	Instanz-DB zu PID_CP
DB 101	CONT_S	Instanz-DB zu PID_ES

Den Funktionsbausteinen PID\_CP und PID\_ES sind jeweils der Instanz-Datenbauein DB 100 bzw. DB 101 zugeordnet

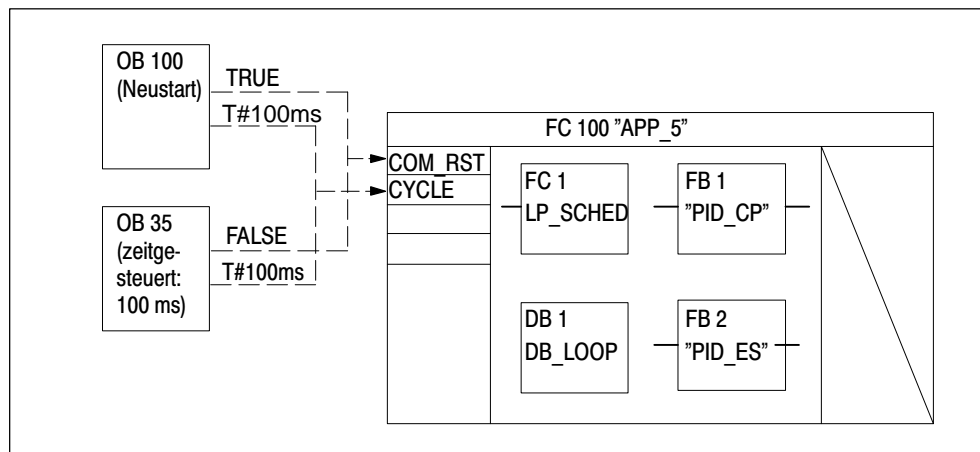


Bild 7-23 Baueine des Beispiels Example5: Verschaltung und Aufruf

### Parametrierung des Beispiels Example5

Wie die Regler über die Funktion FC 100 intern mit dem Regleraufrufverteiler und untereinander verschaltet sind, geht aus Bild 7-24 hervor.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in Kapitel 3.5 beschrieben.

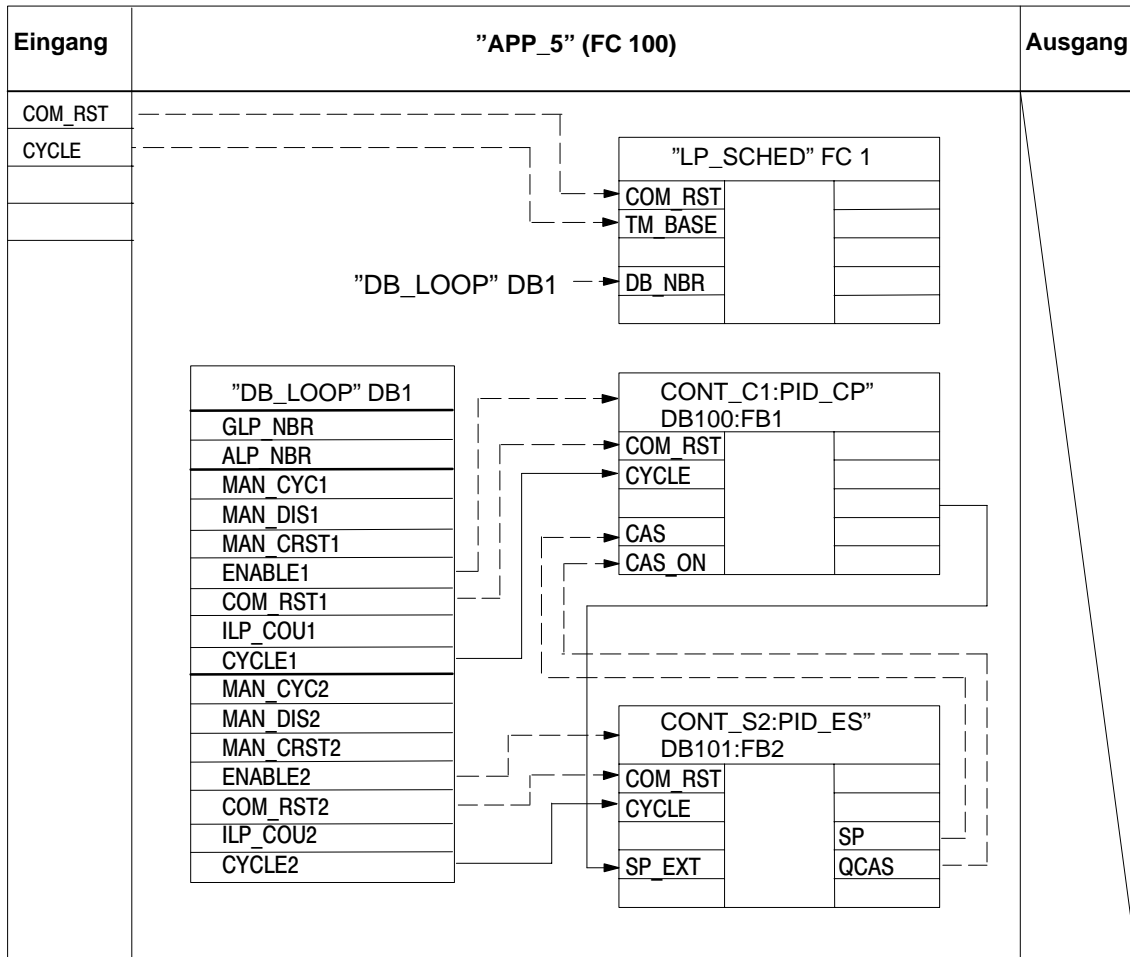


Bild 7-24 Schaltbild und Parameter der Funktion APP\_5

## 7.7 Example6: Pulsegen: Kontinuierlicher Regler mit Impulsausgängen und Streckensimulation

### Anwendung

Das Beispiel (Pulsegen) umfasst einen kontinuierlichen Regler (PID\_CP) mit positivem und negativem Impulsausgang in Verbindung mit einer simulierten Regelstrecke, die aus einem Verzögerungsglied dritter Ordnung (PT3) besteht.

Mit Hilfe des Beispiels Pulsegen ist es möglich, auf einfache Weise einen kontinuierlichen PID-Regler mit Impulsausgängen zu generieren und diesen in allen Eigenschaften im Zusammenspiel mit einer typischen Streckenanordnung zu parametrieren und zu erproben.

Das Beispiel ermöglicht es, die Arbeitsweise und Konfiguration von Reglern mit binären Impulsausgängen, wie sie bei der Regelung von Strecken mit proportional wirkenden Stellgliedern eingesetzt werden, leicht zu verstehen. Solche Regler kommen z. B. bei Temperaturstrecken mit elektrischer Heizung zum Einsatz. Es ist deshalb auch für Einführungs- bzw. Schulungszwecke anwendbar.

Nach Annäherung der Strecke an die Eigenschaften des realen Prozesses durch entsprechende Wahl ihrer Parameter kann durch Identifikation der Modellstrecke mit Hilfe des Konfigurationswerkzeugs ein Satz von geeigneten Reglerkenndaten gefunden werden.

### Funktionen von Example6

Das Beispiel Example6 setzt sich im Wesentlichen aus den beiden Funktionsbausteinen PID\_CP (FB 1) und PROC\_CP (FB 100) zusammen. PID\_CP verkörpert dabei den verwendeten Regler einschließlich Impulsformer, und PROC\_CP simuliert eine Regelstrecke mit Ausgleich dritter Ordnung (Bild 7-25).

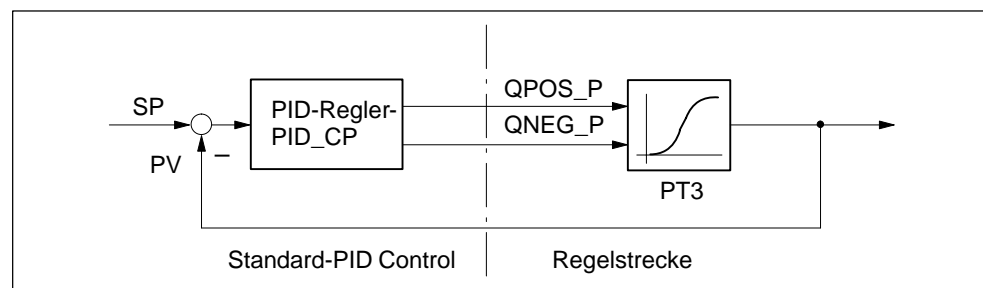


Bild 7-25 Beispiel Example6, Regelkreis

Der Funktionsbaustein PROC\_CP bildet eine Reihenschaltung aus drei Verzögerungsgliedern 1. Ordnung nach (Bild 7-26). Als Eingangssignal der Regelstrecke wirken nicht nur die Impulseingänge POS\_P und NEG\_P, sondern es wirkt als zusätzliches Eingangssignal die Störgröße **DISV**, sodass an dieser Stelle Streckenstörungen manuell aufgeschaltet werden können. Über den Faktor **GAIN** lässt sich die statische Streckenverstärkung bestimmen.



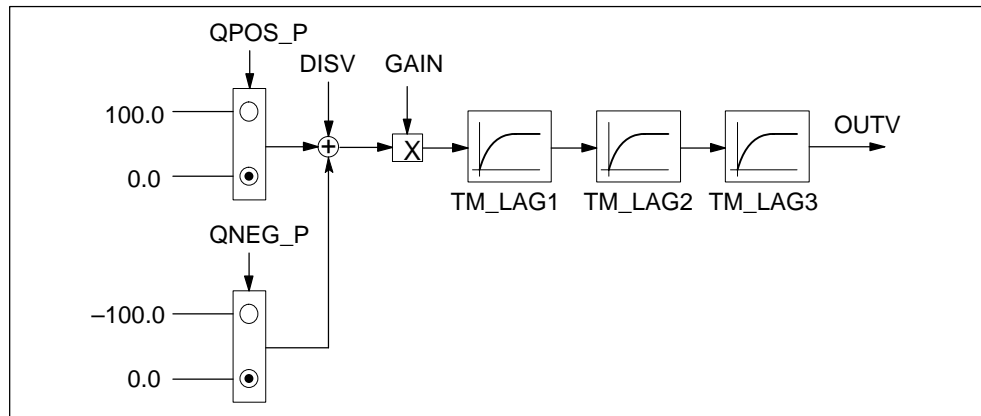


Bild 7-26 Aufbau und Parameter des Strecken-Bausteins PROC\_CP

### Bausteinstruktur

Das Beispiel Pulsegen ist aus der Funktion APP\_Pulsegen, die die Bausteine für den Regler und die simulierte Strecke umfasst, sowie aus den Aufrufbausteinen für Neustart (OB 100) und eine Weckalarmebene (OB 35 mit 100 ms Zeittakt) zusammengesetzt.

Tabelle 7-9 Bausteine des Beispiels Example6

Baustein	Name (in der Symbolleiste)	Beschreibung
OB 100	RESTART	Neustart-OB
OB 35	CYC_INT1	Zeitgesteuerter OB: 100 ms
FC 100	APP_Pulsegen	Beispiel 6
FB 1	PID_CP	Kontinuierlicher PID-Regler mit Impulsformer
FB 100	PROC_CP	Strecke für kontinuierlichen Regler mit Impulseingängen
DB 100	PROCESS	Instanz-DB zu PROC_C
DB 101	CONTROL	Instanz-DB zu PID_CP

Die beiden Funktionsbausteine (Bild 7-27) sind die Instanz-Datenbausteine PROCESS DB 100 für die Strecke und CONTROL DB 101 für den Regler zugeordnet.

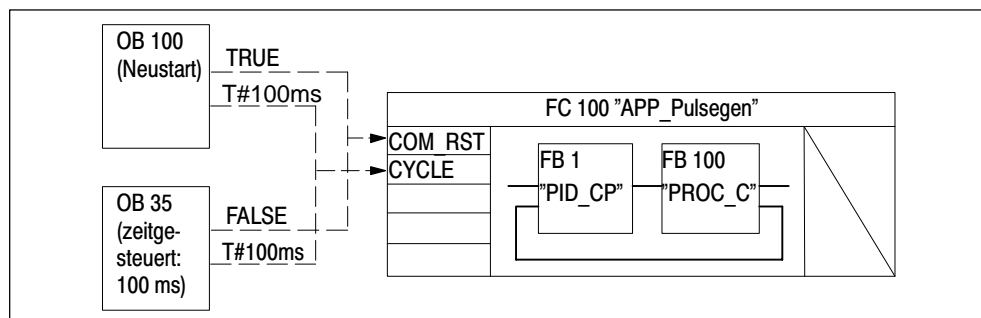


Bild 7-27 Bausteine des Beispiels Example6: Verschaltung und Aufruf

## Die Parameter des Streckenmodells

Die Parameter des Reglerbausteins PID\_CP und ihre Bedeutung sind in Kapitel 6 beschrieben. Die Parameter des Streckenbausteins PROC\_CP sind in der folgenden Tabelle aufgelistet.

Tabelle 7-10 Parameter des Streckenbausteins "PROC\_CP" (DB 100: FB 100)

Parameter	Typ	Wertebereich	Beschreibung
DISV	REAL		Störgröße
GAIN	REAL		Streckenverstärkungsfaktor
TM_LAG1	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 1
TM_LAG2	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 2
TM_LAG3	TIME	$\geq \text{CYCLE}/2$	Verzögerungszeit 3
POS_P	BOOL		Positiver Impuls
NEG_P	BOOL		Negativer Impuls
COM_RST	BOOL		Neustart
CYCLE	TIME	$\geq 1\text{ms}$	Abtastzeit
OUTV	REAL		Ausgangsgröße

## Verschaltung und Aufruf von Example6

Wie der kontinuierliche Regler über die Funktion FC 100 intern mit dem Streckenmodell zu einem Regelkreis verschaltet ist, geht aus Bild 7-28 hervor.

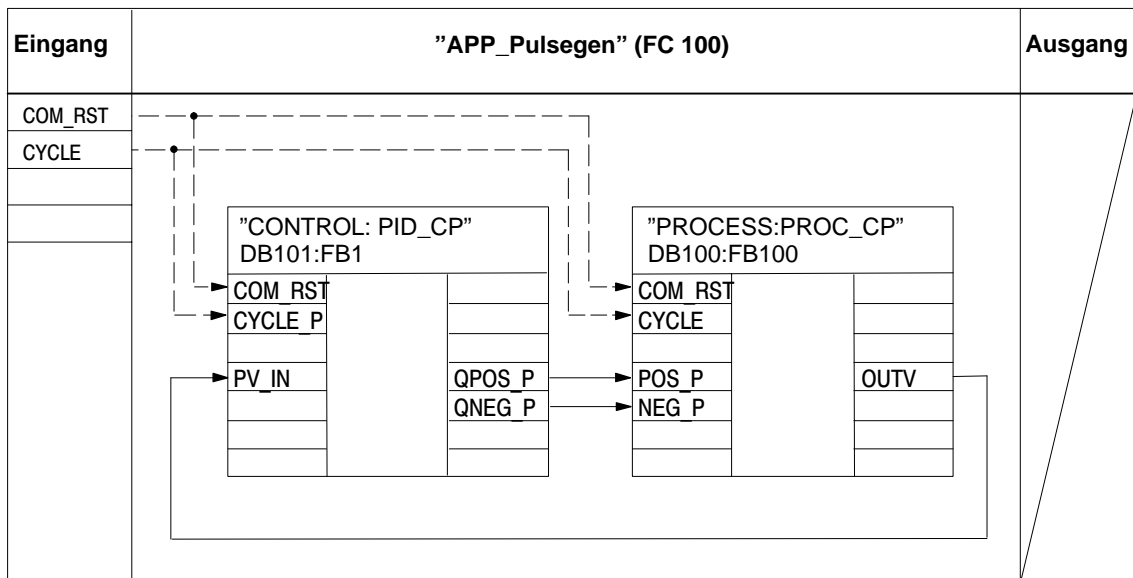


Bild 7-28 Verschaltung und Aufruf der FC 100 (APP\_Pulsegen)

### Parameter der Modellregelstrecke für kontinuierliche Regler

Bild 7-29 zeigt das Funktionsschema und die Parameter der Regelstrecke.

Bei **Neustart** bzw. **Wiederanlauf** verhält sich die Regelung, wie in *Kapitel 3.5* beschrieben.

Eingangsparameter			PROC_CP (FB 100)	Ausgangsparameter			
Parameter	Typ	*)		Signal	Typ	*)	
COM_RST	BOOL	FALSE					
CYCLE	TIME	T#1s					
GAIN	REAL	0.0					
DISV	REAL	0.0					
POS_P	BOOL	FALSE			OUTV	REAL	0.0
NEG_P	BOOL	FALSE					
TM_LAG1	TIME	T#10s					
TM_LAG2	TIME	T#10s					
TM_LAG3	TIME	T#10s					

\*) Vorbelegung bei Neuerstellung des Instanz-DB

Bild 7-29 Funktionsschema und Parameter des Streckenmodells PROC\_CP

### Parameter und Sprungantwort

Anhand einer konkreten Parametrierung eines kontinuierlichen Reglers mit PID-Wirkung wird die Reaktion eines Regelkreises mit simulierter PT-Regelstrecke 3. Ordnung gezeigt. Die eingestellten Streckenparameter mit jeweils 10 s Verzögerungszeit realisieren eine schnellere Regelstrecke, als es bei einer Temperaturregelung in der Praxis entspricht. Anhand der relativ schnellen Strecke lässt sich jedoch die Funktion des Reglers schneller testen. Durch Änderung der Verzögerungszeitkonstanten kann die Eigenschaft der simulierten Strecke jedoch einfach an eine reale Regelstrecke angenähert werden.

Das Kurvendiagramm (Konfigurationswerkzeug) zeigt das Übergangs- und Einschwingverhalten des geschlossenen Regelkreises nach einer Reihe von Sollwertänderungen von jeweils 20 % des Messbereichs (Bild 7-30). Dabei ist die kontinuierliche Stellgröße des Reglers abgebildet, nicht die Impulsausgänge. Die Tabelle enthält die aktuell eingestellten Werte der relevanten Parameter für Regler und Strecke.

Parameter	Typ	Parametrierung	Beschreibung
<b>Regler:</b>			
CYCLE	TIME	1s	Abtastzeit des Reglers
CYCLE_P	TIME	100ms	Abtastzeit
GAIN	REAL	1.535	Proportionalbeiwert
TI	TIME	22.720s	Integrationszeit
TD	TIME	5.974s	Differenzierzeit
TM_LAG	TIME	1.195s	Verzögerungszeit des D-Anteils
<b>Regelstrecke:</b>			
GAIN	REAL	1.5	Streckenverstärkung
TM_LAG1	TIME	10s	Verzögerungszeit 1
TM_LAG2	TIME	10s	Verzögerungszeit 2
TM_LAG3	TIME	10s	Verzögerungszeit 3

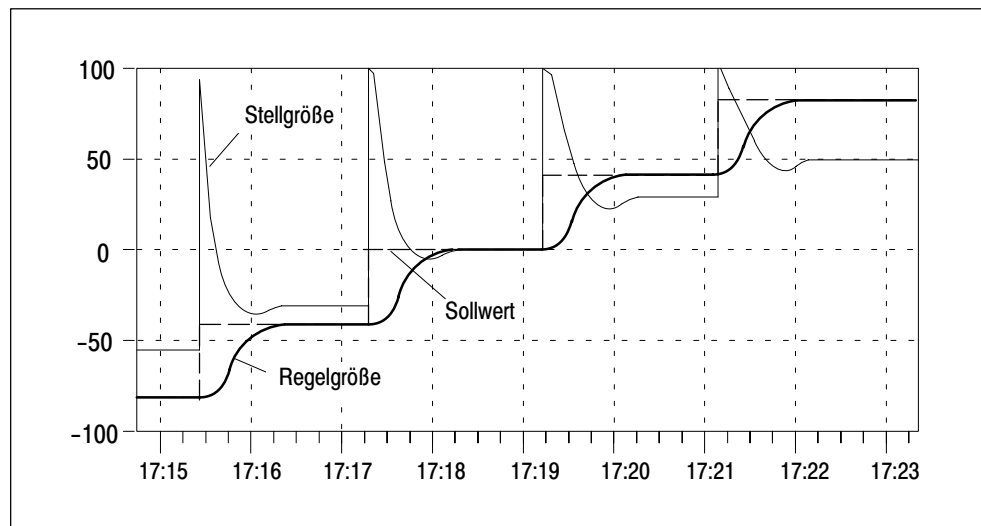


Bild 7-30 Regelung mit kontinuierlichem Regler mit Impulsausgängen und Sollwertsprüngen über den ganzen Messbereich

# Technische Daten und Blockschaltbilder

# 8

## 8.1 Technische Daten: Funktionsbausteine

### CPU-Belastung

Um die Belastung einer bestimmten CPU durch die Installation von Standard PID Control abschätzen zu können, gibt es folgende Anhaltspunkte:

- Für eine beliebig Anzahl von Reglern muss im Anwenderspeicher der CPU der Regler-FB nur einmal vorhanden sein.
- Pro Regler benötigen Sie einen DB mit ca. 0,5 KByte
- Eckdaten für typische Laufzeiten (Bearbeitungszeiten) der Bausteine bei Parametrierung gemäß der Vorbelegung im Regelbetrieb:

Bau- stein- name	Randbedingungen	Bearbeitungszeit (in ms) CPU 315-2AG10	Bearbeitungszeit (in ms) CPU 416-2XK02
PID_CP	typische Randbedingungen	1,3	0,14
PID_ES	ohne Stellungsrückmeldung, ty- pische Randbedingungen	1,5	0,16

### Arbeitsspeicher-Belegung

Die Größe des erforderlichen Bereichs im Anwenderspeicher und damit die Anzahl der Regelkreise, die aufgrund der verfügbaren Speicherkapazität theoretisch installiert werden könnten, ist folgenden Tabellen zu entnehmen:

Baustein- name		Ladespeicher- bedarf	Arbeitsspeicher- bedarf	Lokaldaten
PID_CP	FB 1	8956 Bytes	7796 Bytes	122 Bytes
PID_ES	FB 2	9104 Bytes	7982 Bytes	152 Bytes
LP_SCHED	FC 1	1064 Bytes	976 Bytes	20 Bytes

Instanz-DB bzw. Globaler DB	Ladespeicherbedarf	Arbeitsspeicherbedarf
DB zu PID_CP	1168 Bytes	510 Bytes
DB zu PID_ES	1124 Bytes	484 Bytes
DB_LOOP (bei 5 Regelkreisen)	184 Bytes	100 Bytes
DB_RMPSK (mit Startpunkt und 4 Stützstellen)	142 Bytes	78 Bytes

### Abtastzeit

Die kleinste einstellbare Abtastzeit ist abhängig von der Leistungsklasse der eingesetzten CPU.

---

#### Hinweis

Die begrenzte Rechengenauigkeit schränkt die realisierbare Abtastzeit nach unten ein. Bei kleiner werdender Abtastzeit nehmen die Konstanten der Algorithmen immer kleinere Zahlenwerte an. Dies kann zu einer falschen Berechnung der Stellgröße führen.

Empfehlung:

S7-300:            Abtastzeit  $\geq$  20 ms

S7-400:            Abtastzeit  $\geq$  5 ms

---

### Regler-Aufrufverhalten

Abhängig von der Abtastzeit muss die Bearbeitung des Funktionsbausteins für einen bestimmten Regelkreis in konstanten Zeitabständen aufgerufen werden. Das Betriebssystem der S7 ruft die Weckalarm-OBs zyklisch auf.

Die Abtastzeit und die Weckalarmzeit müssen übereinstimmen.

## 8.2 Blockschaltbilder von Standard PID Control

### Konventionen zu Parameter- und Blocknamen

Für die Bezeichnungen der Parameter- und Blocknamen wurden nicht mehr als acht Zeichen gewählt. Dies spart Schreibarbeit bei der Implementierung unter S7-GRAPH bzw. -SCL und erlaubt eine platzsparende Darstellung der Parameterfelder auf dem Bildschirm.

Die Bezeichnungen orientieren sich in erster Linie an den Bedürfnissen des internationalen Marktes und an der Norm IEC 1131-3. Für die Parameterbezeichnungen von Standard PID Control gelten folgende Konventionen:

SP	setpoint	Sollwert, Führungsgröße
PV	process variable	Istwert (Messwert), Regelgröße
ER	error signal	Regeldifferenz
LMN	manipulated variable	Stellgröße (auszugebendes analoges Stellsignal)
DISV	disturbance variable	Störgröße
MAN	manual value	Hand(stell)wert
CAS	cascade	Kaskade
SQRT	square root	Quadratwurzel
.._ROC	rate of change	Steigung (Änderungsgeschwindigkeit)
Q..	(Q steht für 'O')	allgemeiner Ausgang vom Typ BOOL
.._INT	(internal value)	intern
.._EXT	(external value)	extern
.._ON		boolesche Größe = Schaltsignal
..URLM	uprate limit	Anstiegsbegrenzung
..DRLM	downrate limit	Abstiegsbegrenzung

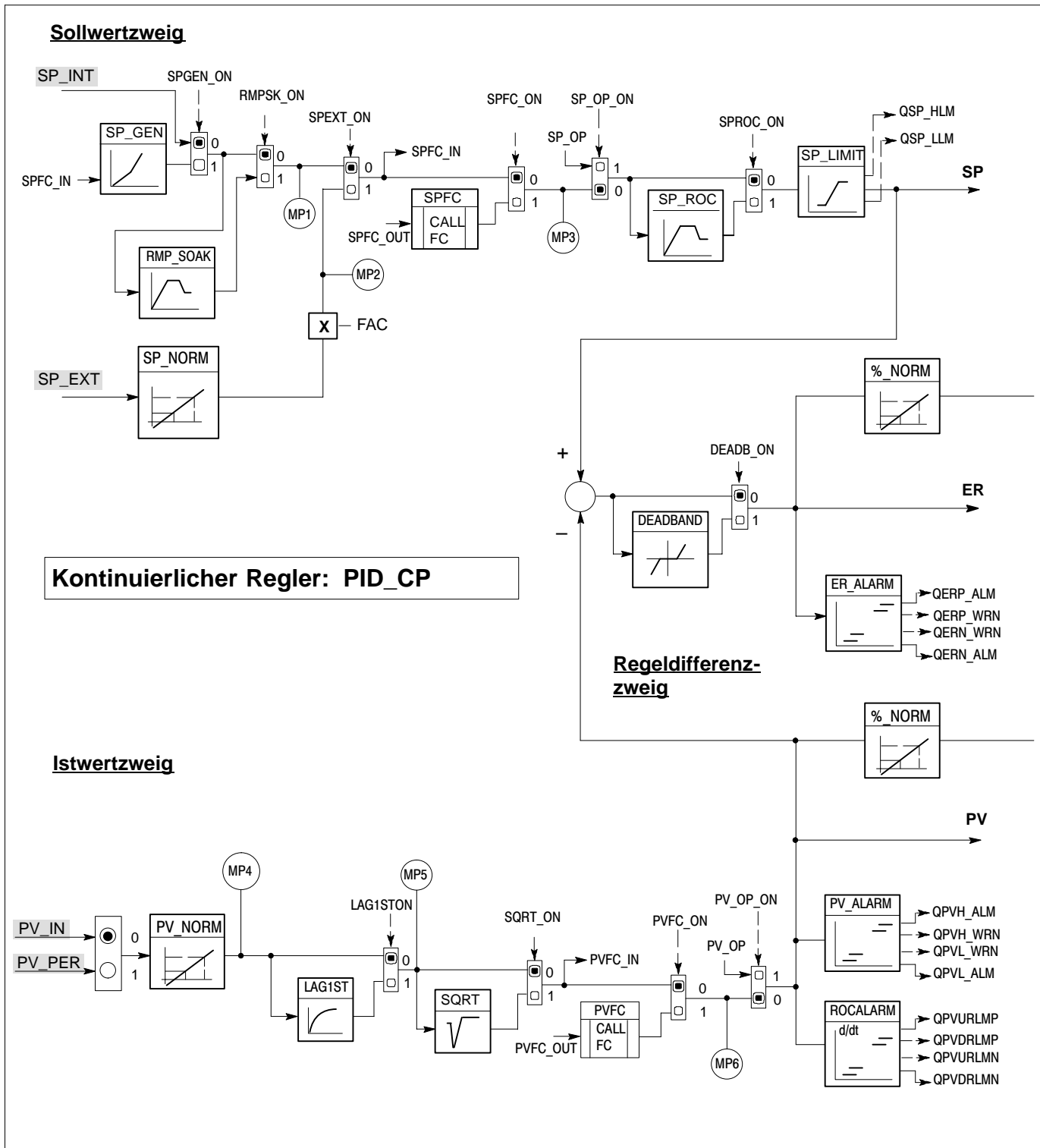
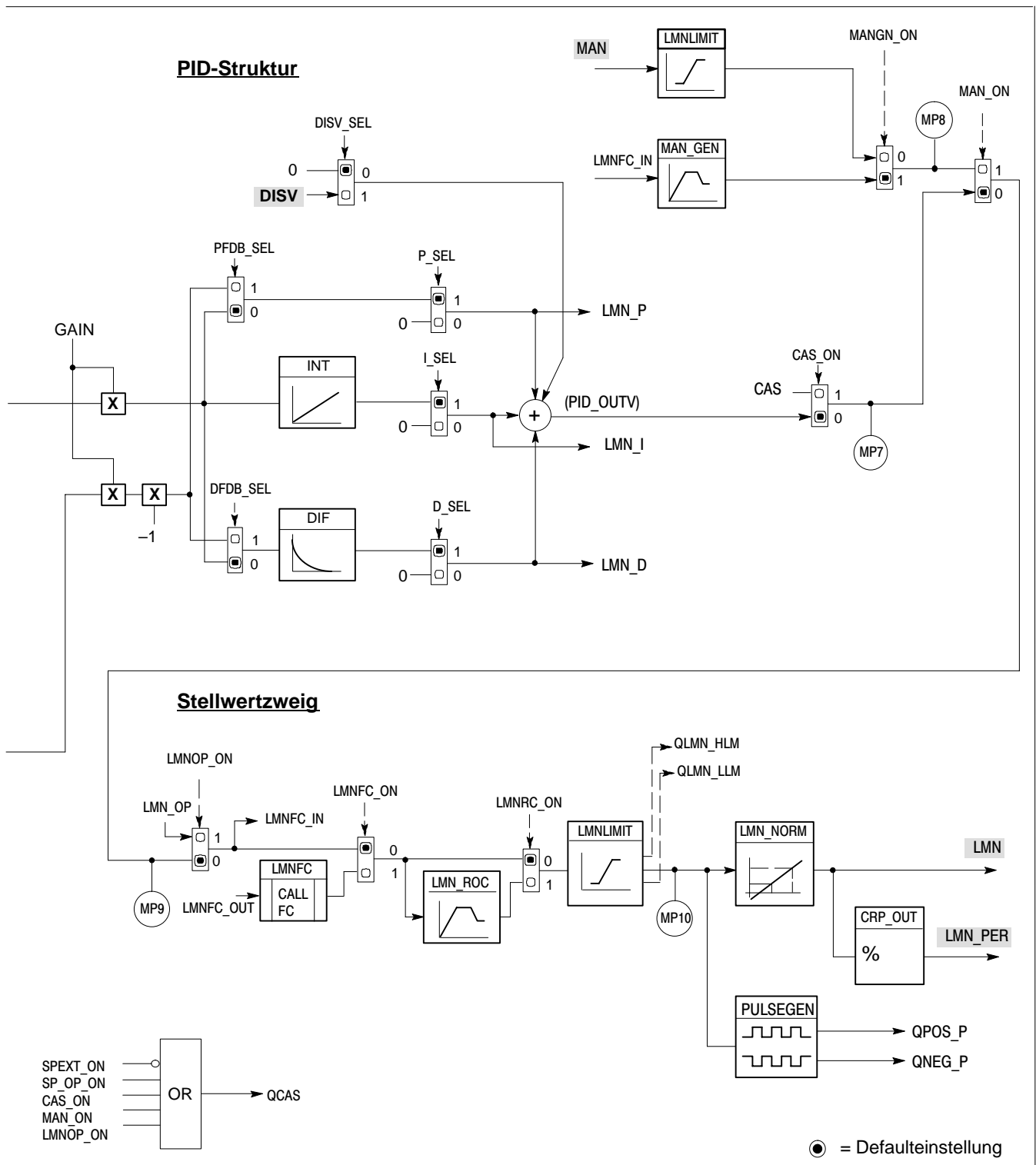


Bild 8-1 Blockschaltbild des kontinuierlichen Reglers: PID\_CP





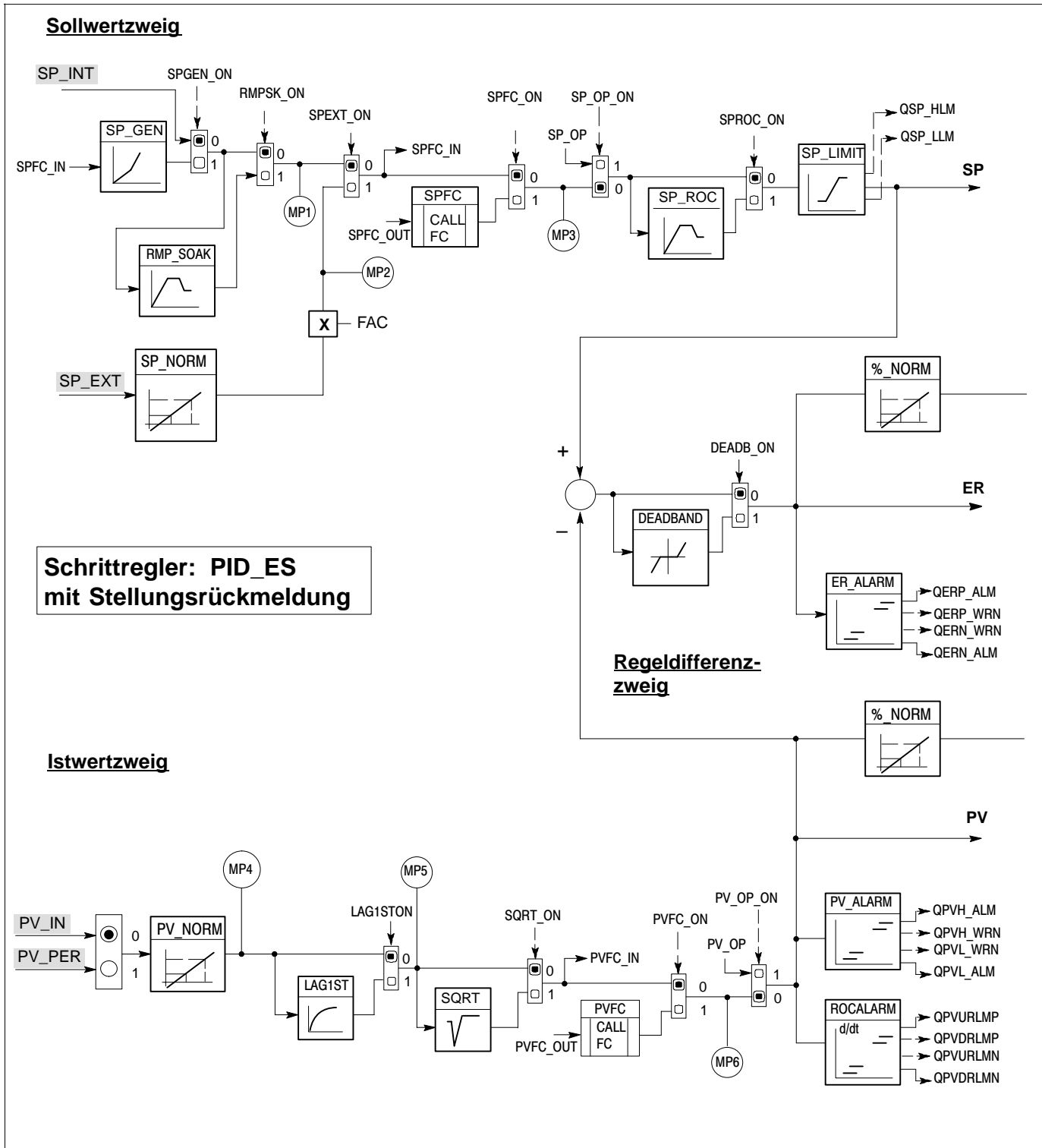
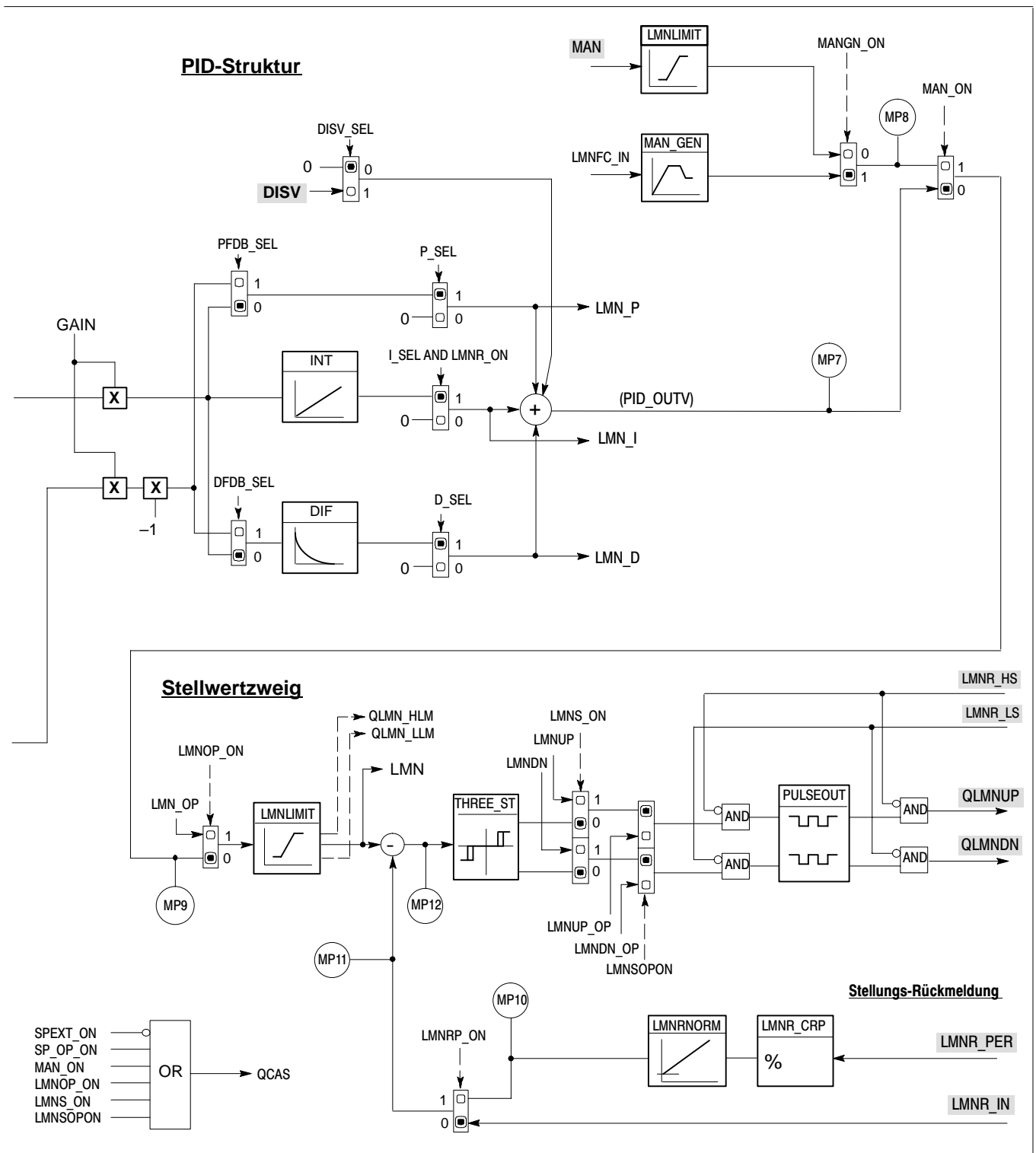


Bild 8-2 Blockschaltbild des Schrittreglers: PID\_ES (mit Stellungsrückmeldung "LMNR = TRUE")



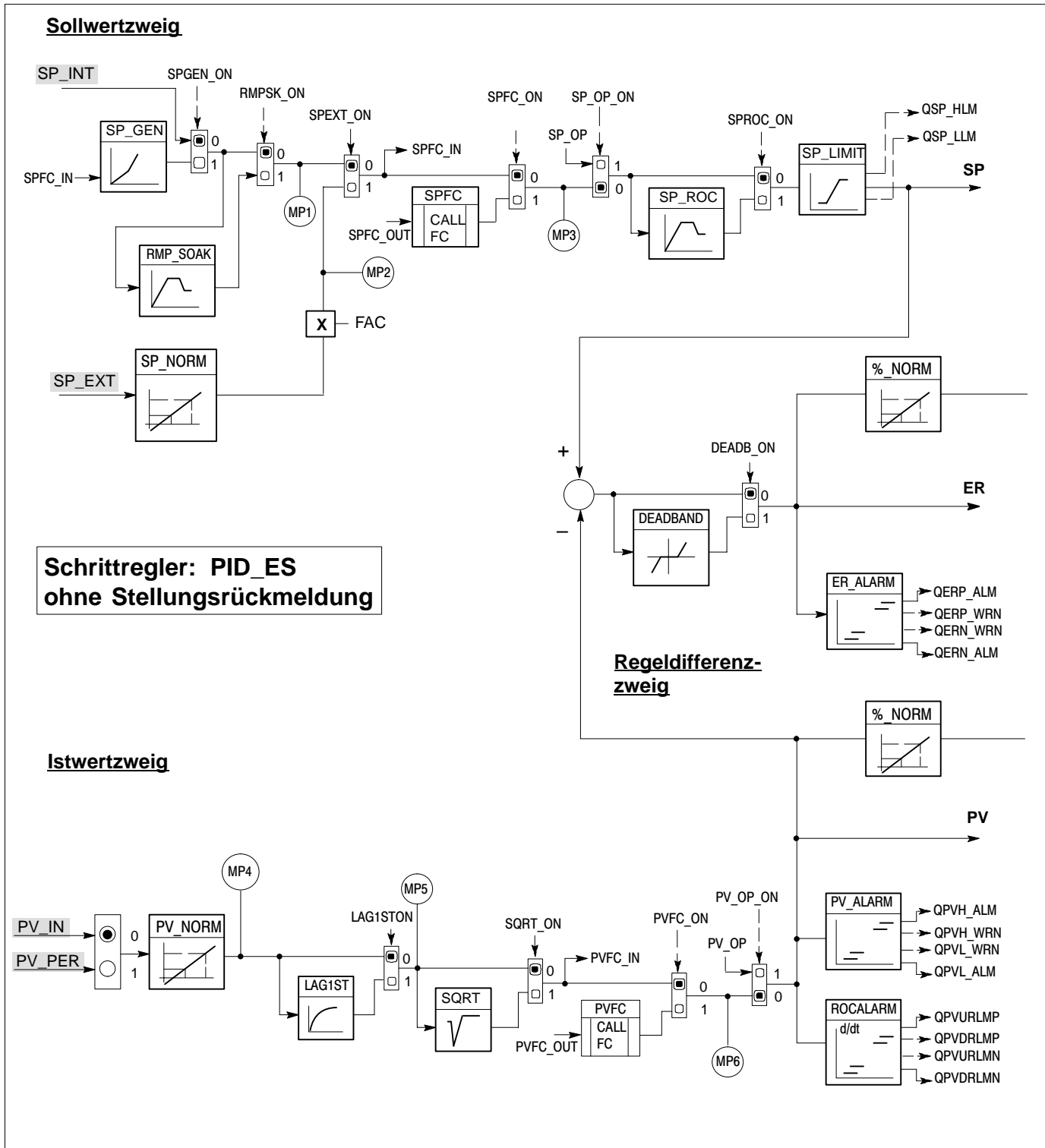
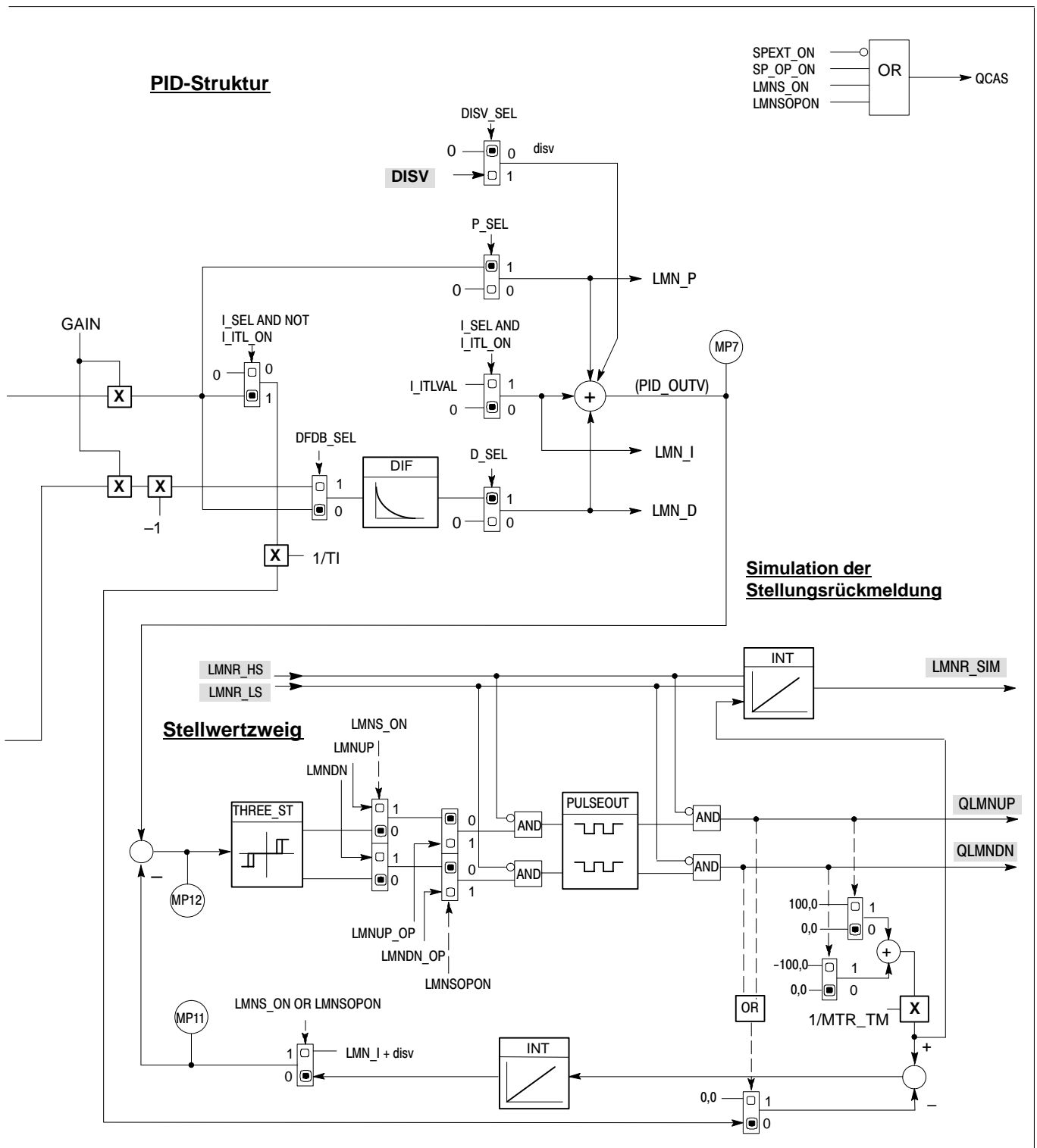


Bild 8-3 Blockschaltbild des Schrittreglers: PID\_ES (ohne Stellungsrückmeldung "LMNR\_ON = FALSE")





---

## Hinweis

- Die in diesem Anhang abgedruckten Parameterlisten geben bezüglich Reihenfolge und Inhalt die Struktur der Instanzdatenbausteine zu den jeweiligen SIMATIC S7 Standard-Funktionsbausteinen wieder.
  - Für jeden Parameter ist der **Zulässige Wertebereich** angegeben.  
"Gesamter Wertebereich" bedeutet: für den betreffenden STEP 7-Operandentyp festgelegter Zahlenbereich.  
"Technischer Wertebereich" bedeutet: eingeschränkter Wertebereich, welcher die Realität ausreichend genau repräsentiert, hier  $-10^5$  bis  $+10^5$ . Dadurch werden bei Operationen mit den betreffenden Wertparametern unhandlich große bzw. kleine Zahlenräume vermieden.
  - Alle Parameter haben bei Neuerstellung des Instanz-DB als Default-Wert die angegebene **Vorbelegung**.  
Diese Werte sind so gewählt, dass bei unveränderter Übernahme aller Voraussetzungen nach keine kritischen Zustände auftreten können.  
Mit Hilfe von STEP 7 können Sie die Vorbelegung durch einen beliebigen anderen Wert im **Zulässigen Wertebereich** ersetzen. Einen bequemeren Weg dafür bietet jedoch die Benutzung des Konfigurationswerkzeugs mit seinen Parametrierfunktionen.
  - Hinweise zu Namensgebungen und den Konventionen für die Parameterbezeichnungen finden Sie in *Kapitel 8.2*.
-

## 9.1 Parameter des Funktionsbausteins PID\_CP

PID_CP		
COM_RST		LMN
I_SEL		LMN_PER
D_SEL		SP
MAN_ON		PV
CAS_ON		QCAS
SELECT		QC_ACT
CYCLE		QPOS_P
CYCLE_P		QNEG_P
SP_INT		
SP_EXT		
PV_IN		
PV_PER		
GAIN		
TI		
TD		
TM_LAG		
DISV		
CAS		
SP_HLM		
SP_LLM		
LMN_HLM		
LMN_LLM		
DB_NBR		
SPFC_NBR		
PVFC_NBR		
LMNFCNBR		
MAN		MAN

Tabelle 9-1 Eingangparameter PID\_CP (kontinuierlicher Regler)

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
COM_RST	BOOL	Neustart (Initialisierungsroutine des FB wird bearbeitet)		FALSE
I_SEL	BOOL	I-Anteil einschalten		TRUE
D_SEL	BOOL	D-Anteil einschalten		FALSE
MAN_ON	BOOL	Handbetrieb einschalten (Regelkreis aufgetrennt, LMN von Hand eingest.)		TRUE
CAS_ON	BOOL	Kaskadenbetrieb einschalten (verschaltet mit QCAS des unterlagerten Reglers)		FALSE



Tabelle 9-1 Eingangsparmeter PID\_CP (kontinuierlicher Regler), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
SELECT	BYTE	falls PULS_ON = TRUE: 0: PID und Pulsgenerator 1: PID (Bst.-Aufruf im OB 1) 2: Pulsgenerator (Bst.-Aufruf in Weckalarm-OB) 3: PID (Bst.-Aufruf in Weckalarm-OB)	0, 1, 2, 3	0
CYCLE	TIME	Abtastzeit (Zeit zwischen zwei Bausteinaufrufen = konstant) Achten Sie unbedingt darauf, dass Sie diesen Parameter mit der Weckalarmtaktzeit des OB, in dem der FB "PID_CP" läuft, parametrieren! Andernfalls arbeiten die zeitabhängigen Funktionen nicht korrekt. (Ausnahme: Sie verwenden eine Zeitumsetzung, z. B. über den Regleraufrufverteiler.)	> 20 ms (S7-300)	T#1s
CYCLE_P	TIME	Abtastzeit des Pulsgenerators Achten Sie unbedingt darauf, dass Sie diesen Parameter mit der Weckalarmtaktzeit des OB, in dem der FB "PID_CP" läuft, parametrieren! Andernfalls arbeiten die zeitabhängigen Funktionen nicht korrekt. (Ausnahme: Sie verwenden eine Zeitumsetzung, z. B. über den Regleraufrufverteiler.)		T#10 ms
SP_INT	REAL	interner Sollwert (zur Sollwertvorgabe mittels B&B-Funktionen)	techn. Wertebereich (physikalische Größe)	0.0
SP_EXT	REAL	externer Sollwert (SP im Gleitpunktformat)	techn. Wertebereich (physikalische Größe)	0.0
PV_IN	REAL	Istwert-Eingang (PV im Gleitpunktformat)	techn. Wertebereich (physikalische Größe)	0.0
PV_PER	INT	Istwert von Peripherie (PV im Peripherieformat)		W#16#0000
GAIN	REAL	Proportionalbeiwert (= Reglerverstärkung)	gesamter Wertebereich (dimensionslos)	2.0
TI	TIME	Integrationszeit	$TI \geq CYCLE$	T#20s
TD	TIME	Differenzierzeit	$TD \geq CYCLE$	T#10s
TM_LAG	TIME	Verzögerungszeit des D-Anteils	$TM\_LAG \geq CYCLE/2$	T#2s
DISV	REAL	Störgröße	-100.0 ... 100.0	0.0
CAS	REAL	Eingang für Kaskadenbetrieb (Verschaltung mit PV des unterlager-ten Reglers)	techn. Wertebereich (physikalische Größe)	0.0

Tabelle 9-1 Eingangsparmeter PID\_CP (kontinuierlicher Regler), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
SP_HLM	REAL	Sollwert obere Begrenzung	techn. Wertebereich (physikalische Größe)	100.0
SP_LLM	REAL	Sollwert untere Begrenzung	techn. Wertebereich (physikalische Größe)	0.0
LMN_HLM	REAL	Stellwert: obere Begrenzung	LMN_LLM ... 100.0	100.0
LMN_LLM	REAL	Stellwert: untere Begrenzung	-100.0 ... LMN_HLM	0.0
DB_NBR	BLOCK_DB	Datenbausteinnummer (DB mit den Stützpunkten der Fahrkurve)		DB 1
SPFC_NBR	BLOCK_FC	Sollwert FC-Nummer (selbstdefinierter FC im Sollwertzweig)		FC 0
PVFC_NBR	BLOCK_FC	Istwert FC-Nummer (selbstdefinierter FC im Istwertzweig)		FC 0
LMNFCNBR	BLOCK_FC	Stellwert FC-Nummer (selbstdefinierter FC im Stellwertzweig)		FC 0

Tabelle 9-2 Ausgangsparmeter PID\_CP (kontinuierlicher Regler)

Parameter	Datentyp	Erläuterung	Vorbelegung
LMN	REAL	Stellwert (Stellwert im Gleitpunktformat)	0.0
LMN_PER	INT	Stellwert für Peripherie (LMN im Peripherieformat)	W#16#0000
SP	REAL	Sollwert (effektiv wirksamer Sollwert)	0.0
PV	REAL	Istwert (Ausgabe des effektiven Istwertes bei Kaskadebetrieb)	0.0
QCAS	BOOL	Signal für Kaskadebetrieb (wird mit CAS_ON des überlagerten Reglers verschaltet)	FALSE
QC_ACT	BOOL	Anzeige, ob beim nächsten Bausteinaufruf der Regelungsteil bearbeitet wird (nur relevant, wenn SELECT den Wert 0 oder 1 hat)	TRUE
QPOS_P	BOOL	Pulsgenerator: positiver Puls ein	FALSE
QNEG_P	BOOL	Pulsgenerator: negativer Puls ein	FALSE

Tabelle 9-3 Durchgangsparameter PID\_CP (kontinuierlicher Regler)

Parameter	Datentyp	Erläuterung	Vorbelegung
MAN	REAL	Hand-Stellwert (zur Stellwertvorgabe mittels B&B-Funktionen)	0.0

Tabelle 9-4 Statische Bausteindaten PID\_CP (Eingänge)

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
PVH_ALM	REAL	Istwert: oberer Grenzwert 'Alarm'	PVH_WRN...100.0	100.0
PVH_WRN	REAL	Istwert: oberer Grenzwert 'Warnung'	PVL_WRN... PVH_ALM	90.0
PVL_WRN	REAL	Istwert: unterer Grenzwert 'Warnung'	PVL_ALM... PVH_WRN	-90.0
PVL_ALM	REAL	Istwert: unterer Grenzwert 'Alarm'	-100.0...PVL_WRN	-100.0
SPGEN_ON	BOOL	Sollwertbedienung einschalten (Verstellen des Sollwerts über Auf-/Ab-Schalter)		FALSE
SPUP	BOOL	Sollwert hoch		FALSE
SPDN	BOOL	Sollwert tief		FALSE
RMPSK_ON	BOOL	Zeitplangeber einschalten (Sollwert wird als Fahrkurve vorgegeben)		FALSE
SPEXT_ON	BOOL	Externer Sollwert einschalten (zur Verschaltung mit anderen Reglerbausteinen)		FALSE
MANGN_ON	BOOL	Handbedienung einschalten (LMN per Schalter eingestellt)		FALSE
MANUP	BOOL	Hand-Stellwert Hoch		FALSE
MANDN	BOOL	Hand-Stellwert Tief		FALSE
DFRMP_ON	BOOL	Ausgang des Zeitplangebers vorbelegen (der Ausgang wird mit SP_INT vorbelegt)		FALSE
CYC_ON	BOOL	Zyklische Wiederholung einschalten (Fahrkurve wird automatisch wiederholt)		FALSE
RMP_HOLD	BOOL	Zeitplangeber anhalten (Ausgang des Zeiplangebers wird eingefroren)		FALSE
CONT_ON	BOOL	(Zeitplan) weitermachen (Fahrkurve wird an nächster Stützstelle fortgesetzt)		FALSE
TUPDT_ON	BOOL	Gesamtzeit neu berechnen (Gesamtzeit der Fahrkurve wird neu berechnet)		FALSE

Tabelle 9-4 Statische Bausteindaten PID\_CP (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
SPFC_ON	BOOL	Sollwert-FC aufrufen		FALSE
SPROC_ON	BOOL	Rampenfunktion einschalten (Begrenzung der Änderungsgeschwindigkeit von SP)		FALSE
PVPER_ON	BOOL	Istwert von Peripherie einschalten (Verschaltung mit Peripheriebaugruppen)		FALSE
LAG1STON	BOOL	Verzögerungsglied 1. Ordnung einschalten		FALSE
SQRT_ON	BOOL	Quadratwurzelfunktion einschalten		FALSE
PVFC_ON	BOOL	Istwert-FC aufrufen		FALSE
DEADB_ON	BOOL	Totzone einschalten (Kleine Störungen und Rauschen werden gefiltert)		FALSE
P_SEL	BOOL	P-Anteil einschalten		TRUE
PFDB_SEL	BOOL	P-Anteil in Rückführung schalten		FALSE
INT_HPOS	BOOL	Einfrieren des I-Anteils in positiver Richtung		FALSE
INT_HNEG	BOOL	Einfrieren des I-Anteils in negativer Richtung		FALSE
I_ITL_ON	BOOL	I-Anteil vorbesetzen		FALSE
DFDB_SEL	BOOL	D-Anteil in Rückführung schalten		FALSE
DISV_SEL	BOOL	Störgröße aufschalten		FALSE
LMNFC_ON	BOOL	Stellwert-FC aufrufen		FALSE
LMNRC_ON	BOOL	Stellwert-Rampenfunktion einschalten (Begrenzung Änderungsgeschwindigkeit von LMN)		FALSE
SMOO_CHG	BOOL	Stoßfreie Umschaltung von Hand nach Automatik		TRUE
PULSE_ON	BOOL	Pulsgenerator ein		FALSE
STEP3_ON	BOOL	Pulsgenerator: Dreipunktregelung einschalten		TRUE
ST2BI_ON	BOOL	Pulsgenerator: Zweipunktregelung für bipolaren Stellbereich einschalten (für unipolaren Stellbereich muss STEP3_ON = FALSE sein)		FALSE
TM_SNBR	INT	Nr. des nächsten Stützpunktes zum Weitermachen	≥0 (dimensionslos)	0
TM_CONT	TIME	Weitermachzeitpunkt (Zeit nach Stützpunkt TM_SNBR zu der der Zeitplangeber die Fahrkurve weiter bearbeitet)	gesamter Wertebereich (dimensionslos)	T#0s

Tabelle 9-4 Statische Bausteindaten PID\_CP (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
FAC	REAL	Faktor (Verhältnis- oder Mischungsfaktor)	gesamter Wertebereich (dimensionslos)	1.0
NM_SPEHR	REAL	Sollwert-Normierung: Arbeitsbereich Eingang oben		100.0
NM_SPELR	REAL	Sollwert-Normierung: Arbeitsbereich Eingang unten		-100.0
SPFC_OUT	REAL	Sollwert FC-Ausgang (wird mit dem Ausgang des FC im Sollwertzweig verschaltet)	-100.0 ... 100.0	0.0
SPURLM_P	REAL	Sollwert-Anstiegsbegrenzung im positiven Bereich	$\geq 0$ [physikalische Größe/s]	10.0
SPDRLM_P	REAL	Sollwert-Abstiegsbegrenzung im positiven Bereich	$\geq 0$ [physikalische Größe/s]	10.0
SPURLM_N	REAL	Sollwert-Anstiegsbegrenzung im negativen Bereich	$\geq 0$ [physikalische Größe/s]	10.0
SPDRLM_N	REAL	Sollwert-Abstiegsbegrenzung im negativen Bereich	$\geq 0$ [physikalische Größe/s]	10.0
NM_PIHR	REAL	Istwert-Normierung: Messbereich Eingang oben		100.0
NM_PILR	REAL	Istwert-Normierung: Messbereich Eingang unten		-100.0
NM_PVHR	REAL	Istwert-Normierung: Messbereich Ausgang oben		100.0
NM_PVLR	REAL	Istwert-Normierung: Messbereich Ausgang unten		-100.0
PV_TMLAG	TIME	Istwertverzögerungszeit (Verzögerung des PT1-Gliedes im PV-Zweig)	gesamter Wertebereich	T#5s
SQRT_HR	REAL	Quadratwurzel: Arbeitsbereich Ausgang oben		100.0
SQRT_LR	REAL	Quadratwurzel: Arbeitsbereich Ausgang unten		0.0
PVFC_OUT	REAL	Istwert FC-Ausgang (wird mit dem Ausgang des FC im Istwertzweig verschaltet)	-100.0 ... 100.0	0.0
PVURLM_P	REAL	Istwert-Anstiegsbegrenzung im positiven Bereich	$\geq 0$ [/s]	10.0
PVDRLM_P	REAL	Istwert-Abstiegsbegrenzung im positiven Bereich	$\geq 0$ [/s]	10.0
PVURLM_N	REAL	Istwert-Anstiegsbegrenzung im negativen Bereich	$\geq 0$ [/s]	10.0
PVDRLM_N	REAL	Istwert-Abstiegsbegrenzung im negativen Bereich	$\geq 0$ [%/s]	10.0

Tabelle 9-4 Statische Bausteindaten PID\_CP (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
PV_HYS	REAL	Istwert Hysterese (Vermeidet 'Flackern' der Überwachungsanzeige)	$\geq 0$	1.0
DEADB_W	REAL	Totzonenbreite (= Bereich Null bis Totzonenobergrenze) (bestimmt Größe des Unempfindlichkeitsbereichs)	0.0 ... 100.0	1.0
ERP_ALM	REAL	Regeldifferenz: positiver Grenzwert 'Alarm'	0 ... 200.0	100.0
ERP_WRN	REAL	Regeldifferenz: positiver Grenzwert 'Warnung'	0 ... 200.0	90.0
ERN_WRN	REAL	Regeldifferenz: negativer Grenzwert 'Warnung'	-200.0 ... 0	-90.0
ERN_ALM	REAL	Regeldifferenz: negativer Grenzwert 'Alarm'	-200.0 ... 0	-100.0
ER_HYS	REAL	Regeldifferenz Hysterese (Vermeidet 'Flackern' der Überwachungsanzeige)	$\geq 0$ [%]	1.0
I_ITLVAL	REAL	Initialisierungswert für I-Anteil	-100.0 ... 100.0 [%]	0.0
LMNFCOUT	REAL	Stellwert FC-Ausgang (wird mit dem Ausgang des FC im Stellwertzweig verschaltet)	-100.0 ... 100.0 [%]	0.0
LMN_URLM	REAL	Stellwert-Anstiegsbegrenzung	$\geq 0$ [%/s]	10.0
LMN_DRLM	REAL	Stellwert-Abstiegsbegrenzung	$\geq 0$ [%/s]	10.0
LMN_FAC	REAL	Stellwertfaktor (Faktor zur Anpassung des Stellwertbereichs)	gesamter Wertebereich (dimensionslos)	1.0
LMN_OFF	REAL	Stellwert-Offset (Nullpunkt der Stellwert-Normierung)	gesamter Wertebereich (dimensionslos)	0.0
PER_TM_P	TIME	Pulsgenerator: Periodendauer des positiven Pulses		T#1s
PER_TM_N	TIME	Pulsgenerator: Periodendauer des negativen Pulses		T#1s
P_B_TM_P	TIME	Pulsgenerator: Mindestimpuls- bzw. Mindestpausendauer des positiven Pulses		T#50ms
P_B_TM_N	TIME	Pulsgenerator: Mindestimpuls- bzw. Mindestpausendauer des negativen Pulses		T#50ms
RATIOFAC	REAL	Pulsgenerator: Verhältnisfaktor (Verhältnis von positiver Impulsdauer und negativer Impulsdauer)	0.1 ... 10.0 (dimensionslos)	1.0
PHASE	INT	Phase vom PID Self Tuner		0

Tabelle 9-5 Statische Lokaldaten PID\_CP (Ausgänge)

Parameter	Datentyp	Erläuterung	Vorbelegung
QPVH_ALM	BOOL	Istwert: oberer Grenzwert 'Alarm' angesprochen	FALSE
QPVH_WRN	BOOL	Istwert: oberer Grenzwert 'Warnung' angesprochen	FALSE
QPVL_WRN	BOOL	Istwert: unterer Grenzwert 'Warnung' angesprochen	FALSE
QPVL_ALM	BOOL	Istwert: unterer Grenzwert 'Alarm' angesprochen	FALSE
QR_S_ACT	BOOL	Zeittabelle für Fahrkurve wird bearbeitet	FALSE
QSP_HLM	BOOL	Sollwert: obere Begrenzung angesprochen	FALSE
QSP_LLM	BOOL	Sollwert: untere Begrenzung angesprochen	FALSE
QPVURLMP	BOOL	Istwert: Anstiegsgrenzwert im positiven Bereich angesprochen	FALSE
QPVDRMLP	BOOL	Istwert: Abstiegsgrenzwert im positiven Bereich angesprochen	FALSE
QPVURLMN	BOOL	Istwert: Anstiegsgrenzwert im negativen Bereich angesprochen	FALSE
QPVDRLMN	BOOL	Istwert: Abstiegsgrenzwert im negativen Bereich angesprochen	FALSE
QERP_ALM	BOOL	Regeldifferenz: positiver Grenzwert 'Alarm' angesprochen	FALSE
QERP_WRN	BOOL	Regeldifferenz: positiver Grenzwert 'Warnung' angesprochen	FALSE
QERN_WRN	BOOL	Regeldifferenz; negativer Grenzwert 'Warnung' angesprochen	FALSE
QERN_ALM	BOOL	Regeldifferenz: negativer Grenzwert 'Alarm' angesprochen	FALSE
QLMN_HLM	BOOL	Stellwert: obere Begrenzung angesprochen	FALSE
QLMN_LLM	BOOL	Stellwert: untere Begrenzung angesprochen	FALSE
NBR_ATMS	INT	Nummer des vom Zeitplangeber aktuell angefahrenen Stützpunktes	0
RS_TM	TIME	"Aktuelle Restzeit" der Fahrkurve bis zum nächsten Stützpunkt	T#0s
T_TM	TIME	"Gesamtzeit" der Fahrkurve	T#0s
RT_TM	TIME	"Gesamtrestzeit" = Zeit bis zum Ende der Fahrkurve	T#0s
ER	REAL	Regeldifferenz	0.0
LMN_P	REAL	P-Anteil	0.0
LMN_I	REAL	I-Anteil	0.0
LMN_D	REAL	D-Anteil	0.0
SPFC_IN	REAL	Sollwert FC-Eingang (wird mit Eingang des selbstdefinierten FC verschaltet)	0.0
PVFC_IN	REAL	Istwert FC-Eingang (wird mit Eingang des selbstdefinierten FC verschaltet)	0.0
LMNFC_IN	REAL	Stellwert FC-Eingang (wird mit Eingang des selbstdefinierten FC verschaltet)	0.0

Tabelle 9-6 Vom Konfigurierwerkzeug benutzte statische Lokaldaten PID\_CP

Parameter	Datentyp	Erläuterung	Vorbelegung
SP_OP_ON	BOOL	Sollwertbedienung einschalten (der Wert von SP_OP wird als Sollwert übernommen)	FALSE
PV_OP_ON	BOOL	Istwertbedienung einschalten (der Wert von PV_OP wird als Sollwert übernommen)	FALSE
LMNOP_ON	BOOL	Stellwertbedienung einschalten (der Wert von LMN_OP wird als Sollwert übernommen)	FALSE
SP_OP	REAL	Sollwertbedienung am Konfigurierwerkzeug	0.0
PV_OP	REAL	Istwertbedienung am Konfigurierwerkzeug	0.0
LMN_OP	REAL	Stellwertbedienung am Konfigurierwerkzeug	0.0
MP1	REAL	Messpunkt 1: Interner Sollwert	0.0
MP2	REAL	Messpunkt 2: Externer Sollwert	0.0
MP3	REAL	Messpunkt 3: Unbegrenzter Sollwert	0.0
MP4	REAL	Messpunkt 4: Istwert von der Peripheriebaugruppe	0.0
MP5	REAL	Messpunkt 5: Istwert nach dem Verzögerungsglied 1. Ordnung	0.0
MP6	REAL	Messpunkt 6: Wirksamer Istwert (PV)	0.0
MP7	REAL	Messpunkt 7: Stellwert vom PID-Algorithmus	0.0
MP8	REAL	Messpunkt 8: Hand-Stellwert	0.0
MP9	REAL	Messpunkt 9: Unbegrenzter Stellwert	0.0
MP10	REAL	Messpunkt 10: Begrenzter Stellwert	0.0

Die vom Konfigurierungswerkzeug genutzten statischen Lokaldaten stehen zu Beginn des Wertebereiches der statischen Lokaldaten.

---

#### Hinweis

Alle anderen statischen Lokaldaten dürfen nicht beeinflusst werden.

---



## 9.2 Parameter des Funktionsbausteins PID\_ES

PID_ES		
COM_RST		QLMNUP
I_SEL		QLMNDN
D_SEL		QCAS
MAN_ON		LMN
LMNR_HS		SP
LMNR_LS		PV
CYCLE		
SP_INT		
SP_EXT		
PV_IN		
PV_PER		
GAIN		
TI		
TD		
TM_LAG		
DISV		
LMNR_IN		
LMNR_PER		
SP_HLM		
SP_LLM		
LMN_HLM		
LMN_LLM		
DB_NBR		
SPFC_NBR		
PVFC_NBR		
MAN		MAN

Tabelle 9-7 Eingangparameter PID\_ES (Schrittregler)

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
COM_RST	BOOL	Neustart (Initialisierungsroutine des FB wird bearbeitet)		FALSE
I_SEL	BOOL	I-Anteil einschalten		TRUE
D_SEL	BOOL	D-Anteil einschalten		FALSE
MAN_ON	BOOL	Handbetrieb einschalten (Regelkreis aufgetrennt, LMN von Hand eingest.)		TRUE
LMNR_HS	BOOL	Oberes Anschlagsignal der Stellungsrückmeldung		FALSE
LMNR_LS	BOOL	Unteres Anschlagsignal der Stellungsrückmeldung		FALSE

Tabelle 9-7 Eingangsparmeter PID\_ES (Schrittregler), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
CYCLE	TIME	Abtastzeit (Zeit zwischen Bausteinaufrufen = konstant) Achten Sie unbedingt darauf, dass Sie diesen Parameter mit der Weckalarmtaktzeit des OB, in dem der FB "PID_CP" läuft, parametrieren! Andernfalls arbeiten die zeitabhängigen Funktionen nicht korrekt. (Ausnahme: Sie verwenden eine Zeitumsetzung, z. B. über den Regleraufrufverteiler.)	≥20 ms (S7-300)	T#1s
SP_INT	REAL	interner Sollwert (zur Sollwertvorgabe mittels B&B-Funktionen)	techn. Wertebereich (physikalische Größe)	0.0
SP_EXT	REAL	externer Sollwert (SP im Gleitpunktformat)	techn. Wertebereich (physikalische Größe)	0.0
PV_IN	REAL	Istwert-Eingang (PV im Gleitpunktformat)	techn. Wertebereich (physikalische Größe)	0.0
PV_PER	INT	Istwert von Peripherie		W#16#0000
GAIN	REAL	Proportionalbeiwert (= Reglerverstärkung)	gesamter Wertebereich (dimensionslos)	2.0
TI	TIME	Integrationszeit	TI ≥ CYCLE	T#20s
TD	TIME	Differenzierzeit	TD ≥ CYCLE	T#10s
TM_LAG	TIME	Verzögerungszeit des D-Anteils	TM_LAG ≥ CYCLE/2	T#2s
DISV	REAL	Störgröße	-100.0 ... 100.0 [%]	0.0
LMNR_IN	REAL	Stellungsrückmeldung (LMNR im Gleitpunktformat)	0.0 ... 100.0 [%]	0.0
LMNR_PER	WORD	Stellungsrückmeldung von Peripherie (LMNR im Peripherieformat)		W#16#0000
SP_HLM	REAL	Sollwert obere Begrenzung	techn. Wertebereich (physikalische Größe)	100.0
SP_LLM	REAL	Sollwert untere Begrenzung	techn. Wertebereich (physikalische Größe)	0.0
LMN_HLM	REAL	Stellwert: obere Begrenzung	LMN_LLM ... 100.0 [%]	100.0
LMN_LLM	REAL	Stellwert: untere Begrenzung	0.0 ... LMN_HLM [%]	0.0
DB_NBR	BLOCK_DB	Datenbausteinnummer (DB mit den Stützpunkten der Fahrkurve)		DB 1
SPFC_NBR	BLOCK_FC	Sollwert FC-Nummer (selbstdefinierter FC im Sollwertzweig)		FC 0
PVFC_NBR	BLOCK_FC	Istwert FC-Nummer (selbstdefinierter FC im Istwertzweig)		FC 0

Tabelle 9-8 Ausgangsparameter PID\_ES (Schrittregler)

Parameter	Datentyp	Erläuterung	Vorbelegung
QLMNUP	BOOL	Stellwertsignal hoch	FALSE
QLMNDN	BOOL	Stellwertsignal tief	FALSE
QCAS	BOOL	Signal für Kaskadebetrieb (wird mit CAS_ON des überlagerten Reglers verschaltet)	FALSE
LMN	REAL	Stellwertsignal (nach Regelalgorithmus)	0.0
SP	REAL	Sollwert (effektiv wirksamer Sollwert)	0.0
PV	REAL	Istwert (Ausgabe des effektiven Istwertes bei Kaskadebetrieb)	0.0

Tabelle 9-9 Durchgangsparameter PID\_ES (Schrittregler)

Parameter	Datentyp	Erläuterung	Vorbelegung
MAN	REAL	Hand-Stellwert (zur Stellwertvorgabe mittels B&B-Funktionen)	0.0

Tabelle 9-10 Statische Lokaldaten PID\_ES (Eingänge)

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
PVH_ALM	REAL	Istwert: oberer Grenzwert 'Alarm'	PVH_WRN...100.0	100.0
PVH_WRN	REAL	Istwert: oberer Grenzwert 'Warnung'	PVL_WRN... PVH_ALM	90.0
PVL_WRN	REAL	Istwert: unterer Grenzwert 'Warnung'	PVL_ALM... PVH_WRN	-90.0
PVL_ALM	REAL	Istwert: unterer Grenzwert 'Alarm'	-100.0...PVL_WRN	-100.0
SPGEN_ON	BOOL	Sollwertbedienung einschalten (Verstellen des Sollwerts über Auf-/Ab-Schalter)		FALSE
SPUP	BOOL	Sollwert hoch		FALSE
SPDN	BOOL	Sollwert tief		FALSE
RMPSK_ON	BOOL	Zeitplangeber einschalten (Sollwert wird als Fahrkurve vorgegeben)		FALSE
SPEXT_ON	BOOL	Externer Sollwert einschalten (zur Verschaltung mit anderen Reglerbausteinen)		FALSE

Tabelle 9-10 Statische Lokaldaten PID\_ES (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
MANGN_ON	BOOL	Handbedienung einschalten (LMN per Schalter eingestellt)		FALSE
MANUP	BOOL	Hand-Stellwert Hoch		FALSE
MANDN	BOOL	Hand-Stellwert Tief		FALSE
LMNS_ON	BOOL	Handbetrieb der Stellsignale einschalten		FALSE
LMNUP	BOOL	Stellwertsignal Hoch (das Ausgangssignal QLMNUP wird bedient)		FALSE
LMNDN	BOOL	Stellwertsignal Tief (das Ausgangssignal QLMNDN wird bedient)		FALSE
DFRMP_ON	BOOL	Ausgang des Zeitplangebers vorbelegen (der Ausgang wird mit SP_INT vorbelegt)		FALSE
CYC_ON	BOOL	Zyklische Wiederholung einschalten (Fahrkurve wird automatisch wiederholt)		FALSE
RMP_HOLD	BOOL	Zeitplangeber anhalten (Ausgang des Zeiplangebers wird eingefroren)		FALSE
CONT_ON	BOOL	(Zeitplan) weitermachen (Fahrkurve wird an nächster Stützstelle fortgesetzt)		FALSE
TUPDT_ON	BOOL	Gesamtzeit neu berechnen (Gesamtzeit der Fahrkurve wird neu berechnet)		FALSE
SPFC_ON	BOOL	Sollwert-FC aufrufen		FALSE
SPROC_ON	BOOL	Rampenfunktion einschalten (Begrenz. der Änderungsgeschwindigkeit von SP)		FALSE
PVPER_ON	BOOL	Istwert von Peripherie einschalten (Verschaltung mit Peripheriebaugruppen)		FALSE
LAG1STON	BOOL	Verzögerungsglied 1. Ordnung einschalten		FALSE
SQRT_ON	BOOL	Quadratwurzelfunktion einschalten		FALSE
PVFC_ON	BOOL	Istwert-FC aufrufen		FALSE
DEADB_ON	BOOL	Totzone einschalten (Kleine Störungen und Rauschen werden gefiltert)		FALSE
P_SEL	BOOL	P-Anteil einschalten		TRUE
PFDB_SEL	BOOL	P-Anteil in Rückführung schalten		FALSE
INT_HPOS	BOOL	I-Anteil in positiver Richtung einfrieren		FALSE
INT_HNEG	BOOL	I-Anteil in negativer Richtung einfrieren		FALSE

Tabelle 9-10 Statische Lokaldaten PID\_ES (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
I_ITL_ON	BOOL	I-Anteil vorbesetzen		FALSE
DFDB_SEL	BOOL	D-Anteil in Rückführung schalten		FALSE
DISV_SEL	BOOL	Störgröße aufschalten		FALSE
LMNR_ON	BOOL	Stellungsrückmeldung einschalten (Betriebsarten: Schrittreger mit/ohne Stellungsrückmeldung) <u>Nicht</u> im Regelbetrieb umschalten!		FALSE
LMNRP_ON	BOOL	Stellungsrückmeldung von Peripherie einschalten		FALSE
TM_SNBR	INT	Nr. des nächsten Stützpunktes zur Fortsetzung der Fahrkurve	$\geq 0$ (dimensionslos)	0
TM_CONT	TIME	Zeitspanne bis zur Fortsetzung der Fahrkurve (Zeitspanne, die abläuft, bevor der Zeitplangeber nach einer Unterbrechung die Fahrkurve am Stützpunkt TM_SNBR weiter bearbeitet)	gesamter Wertebereich (dimensionslos)	T#0s
FAC	REAL	Faktor (Verhältnis- oder Mischungsfaktor)	gesamter Wertebereich (dimensionslos)	1.0
NM_SPEHR	REAL	Sollwert-Normierung: Eingang oben		100.0
NM_SPELR	REAL	Sollwert-Normierung: Eingang unten		-100.0
SPFC_OUT	REAL	Sollwert FC-Ausgang (wird mit dem Ausgang des FC im Sollwertzweig verschaltet)	-100.0 ... 100.0	0.0
SPURLM_P	REAL	Sollwert-Anstiegsbegrenzung im positiven Bereich	$\geq 0$ [/s]	10.0
SPDRLM_P	REAL	Sollwert-Abstiegsbegrenzung im positiven Bereich	$\geq 0$ [/s]	10.0
SPURLM_N	REAL	Sollwert-Anstiegsbegrenzung im negativen Bereich	$\geq 0$ [/s]	10.0
SPDRLM_N	REAL	Sollwert-Abstiegsbegrenzung im negativen Bereich	$\geq 0$ [/s]	10.0
NM_PIHR	REAL	Istwert-Normierung: Eingang oben		100.0
NM_PILR	REAL	Istwert-Normierung: Eingang unten		-100.0
NM_PVHR	REAL	Istwert-Normierung: Ausgang oben		100.0
NM_PVLR	REAL	Istwert-Normierung: Ausgang unten		-100.0
PV_TMLAG	TIME	Istwertverzögerungszeit (Verzögerung des PT1-Gliedes im PV-Zweig)	gesamter Wertebereich	T#5s
SQRT_HR	REAL	Quadratwurzel: Messbereich Ausgang oben		100.0
SQRT_LR	REAL	Quadratwurzel: Messbereich Ausgang unten		0.0
PVFC_OUT	REAL	Istwert FC-Ausgang (wird mit dem Ausgang des FC im Istwertzweig verschaltet)	-100.0 ... 100.0	0.0

Tabelle 9-10 Statische Lokaldaten PID\_ES (Eingänge), Fortsetzung

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
PVURLM_P	REAL	Istwert-Anstiegsbegrenzung im positiven Bereich	0 [/s]	10.0
PVDRLM_P	REAL	Istwert-Abstiegsbegrenzung im positiven Bereich	$\geq 0$ [/s]	10.0
PVURLM_N	REAL	Istwert-Anstiegsbegrenzung im negativen Bereich	$\geq 0$ [/s]	10.0
PVDRLM_N	REAL	Istwert-Abstiegsbegrenzung im negativen Bereich	$\geq 0$ [/s]	10.0
PV_HYS	REAL	Istwert Hysterese (Vermeidet 'Flackern' der Überwachungsanzeige)	$\geq 0$	1.0
DEADB_W	REAL	Totzonenbreite (bestimmt Größe des Unempfindlichkeitsbereichs)	0.0 ... 100.0	1.0
ERP_ALM	REAL	Regeldifferenz: positiver Grenzwert 'Alarm'	0 ... 200.0	100.0
ERP_WRN	REAL	Regeldifferenz: positiver Grenzwert 'Warnung'	0 ... 200.0	90.0
ERN_WRN	REAL	Regeldifferenz: negativer Grenzwert 'Warnung'	-200.0 ... 0	-90.0
ERN_ALM	REAL	Regeldifferenz: negativer Grenzwert 'Alarm'	-200.0 ... 0	-100.0
ER_HYS	REAL	Regeldifferenz Hysterese (Vermeidet 'Flackern' der Überwachungsanzeige)	$\geq 0$	1.0
I_ITLVAL	REAL	Initialisierungswert für I-Anteil	-100.0 ... 100.0 [%]	0.0
LMNR_FAC	REAL	Stellungsrückmeldungs-Faktor (Faktor zur Anpassung des Rückmeldungsbereichs)	gesamter Wertebereich (dimensionslos)	1.0
LMNR_OFF	REAL	Stellungsrückmeldungs-Offset (Nullpunkt der Rückmeldungsnormierung)	-100.0 ... 100.0 [%]	0.0
PULSE_TM	TIME	Mindestimpulsdauer	= n * CYCLE /n=0,1,2...	T#3s
BREAK_TM	TIME	Mindestpausendauer	= n * CYCLE /n=0,1,2...	T#3s
MTR_TM	TIME	Motorstellzeit	$\geq$ CYCLE	T#30s
PHASE	INT	Phase vom PID Self Tuner		0

Tabelle 9-11 Statische Lokaldaten PID\_ES (Ausgänge)

Parameter	Datentyp	Erläuterung	Vorbelegung
QPVH_ALM	BOOL	Istwert: oberer Grenzwert 'Alarm' angesprochen	FALSE
QPVH_WRN	BOOL	Istwert: oberer Grenzwert 'Warnung' angesprochen	FALSE
QPVL_WRN	BOOL	Istwert: unterer Grenzwert 'Warnung' angesprochen	FALSE
QPVL_ALM	BOOL	Istwert: unterer Grenzwert 'Alarm' angesprochen	FALSE
QR_S_ACT	BOOL	Zeittabelle für Fahrkurve wird bearbeitet	FALSE
QSP_HLM	BOOL	Sollwert: obere Begrenzung angesprochen	FALSE
QSP_LLM	BOOL	Sollwert: untere Begrenzung angesprochen	FALSE
QPVURLMP	BOOL	Istwert: Anstiegsgrenzwert im positiven Bereich angesprochen	FALSE
QPVDRMLP	BOOL	Istwert: Abstiegsgrenzwert im positiven Bereich angesprochen	FALSE
QPVURLMN	BOOL	Istwert: Anstiegsgrenzwert im negativen Bereich angesprochen	FALSE
QPVDRLMN	BOOL	Istwert: Abstiegsgrenzwert im negativen Bereich angesprochen	FALSE
QERP_ALM	BOOL	Regeldifferenz: positiver Grenzwert 'Alarm' angesprochen	FALSE
QERP_WRN	BOOL	Regeldifferenz: positiver Grenzwert 'Warnung' angesprochen	FALSE
QERN_WRN	BOOL	Regeldifferenz; negativer Grenzwert 'Warnung' angesprochen	FALSE
QERN_ALM	BOOL	Regeldifferenz: negativer Grenzwert 'Alarm' angesprochen	FALSE
QLMN_HLM	BOOL	Stellwert: obere Begrenzung angesprochen	FALSE
QLMN_LLM	BOOL	Stellwert: untere Begrenzung angesprochen	FALSE
NBR_ATMS	INT	Nummer des vom Zeitplangeber aktuell angefahrenen Stützpunktes	0
RS_TM	TIME	"Aktuelle Restzeit" der Fahrkurve bis zum nächsten Stützpunkt	T#0s
T_TM	TIME	"Gesamtzeit" = abgelaufene Zeit der Fahrkurve	T#0s
RT_TM	TIME	"Gesamtrestzeit" = Zeit bis zum Ende der Fahrkurve	T#0s
ER	REAL	Regeldifferenz	0.0
LMN_P	REAL	P-Anteil	0.0
LMN_I	REAL	I-Anteil	0.0
LMN_D	REAL	D-Anteil	0.0
SPFC_IN	REAL	Sollwert FC-Eingang (wird mit Eingang des selbstdefinierten FC verschaltet)	0.0
PVFC_IN	REAL	Istwert FC-Eingang (wird mit Eingang des selbstdefinierten FC verschaltet)	0.0

Tabelle 9-12 Vom Konfigurierwerkzeug benutzte statische Lokaldaten (Schrittregler PID\_ES)

Parameter	Datentyp	Erläuterung	Vorbelegung
SP_OP_ON	BOOL	Sollwertbedienung einschalten (der Wert von SP_OP wird als Sollwert übernommen)	FALSE
PV_OP_ON	BOOL	Istwertbedienung einschalten (der Wert von PV_OP wird als Sollwert übernommen)	FALSE
LMNOP_ON	BOOL	Stellwertbedienung einschalten (der Wert von LMN_OP wird als Sollwert übernommen)	FALSE
LMNSOPON	BOOL	Stellwertsignalbedienung einschalten (LMNUP_OP und LMNDN_OP werden als Stellsignale übernommen)	FALSE
LMNUP_OP	BOOL	Stellwertsignal Hoch	FALSE
LMNDN_OP	BOOL	Stellwertsignal Tief	FALSE
LMNRS_ON	BOOL	Simulation der Stellungsrückmeldung einschalten	FALSE
SP_OP	REAL	Sollwertbedienung am Konfigurierwerkzeug	0.0
PV_OP	REAL	Istwertbedienung am Konfigurierwerkzeug	0.0
LMN_OP	REAL	Stellwertbedienung am Konfigurierwerkzeug	0.0
LMNRSVAL	REAL	Startwert der simulierten Stellungsrückmeldung	0.0
LMNR_SIM	REAL	Aktueller Wert der simulierten Stellungsrückmeldung	0.0
MP1	REAL	Messpunkt 1: Interner Sollwert	0.0
MP2	REAL	Messpunkt 2: Externer Sollwert	0.0
MP3	REAL	Messpunkt 3: Unbegrenzter Sollwert	0.0
MP4	REAL	Messpunkt 4: Istwert von der Peripheriebaugruppe	0.0
MP5	REAL	Messpunkt 5: Istwert nach dem Verzögerungsglied 1. Ordnung	0.0
MP6	REAL	Messpunkt 6: Wirksamer Istwert (PV)	0.0
MP7	REAL	Messpunkt 7: Stellwert vom PID-Algorithmus	0.0
MP8	REAL	Messpunkt 8: Hand-Stellwert	0.0
MP9	REAL	Messpunkt 9: Unbegrenzter Stellwert	0.0
MP10	REAL	Messpunkt 10: Stellungsrückmeldung Peripherie	0.0
MP11	REAL	Messpunkt 11: Rückkopplungswert (LMNR_ON = FALSE) Stellungsrückmeldung (LMNR_ON = TRUE)	0.0
MP12	REAL	Messpunkt 12: Dreipunktglied Eingang	0.0

Die vom Konfigurierungswerkzeug genutzten statischen Lokaldaten stehen am Beginn des Wertebereiches der statischen Lokaldaten.

---

#### Hinweis

Alle anderen statischen Lokaldaten dürfen nicht beeinflusst werden.

---



Tabelle 9-13 Funktion RMP\_SOAK (PID\_CP und PID\_ES): Globaler Datenbaustein (DB\_NBR), vorgelegt mit Startpunkt und vier Stützpunkten

Parameter	Datentyp	Kommentar	zulässiger Wertebereich	Vorbelegung
NBR_PTS	INT	Anzahl der Stützpunkte	0 ... 255	4
PI[0].OUTV	REAL	Ausgangswert [0]: Startpunkt	gesamter Wertebereich	0.0
PI[0].TMV	TIME	Zeitwert [0]: Startpunkt	gesamter Wertebereich	T#1 s
PI[1].OUTV	REAL	Ausgangswert [1]: Stützpunkt 1	gesamter Wertebereich	0.0
PI[1].TMV	TIME	Zeitwert [1]: Stützpunkt 1	gesamter Wertebereich	T#1 s
PI[2].OUTV	REAL	Ausgangswert [2]: Stützpunkt 2	gesamter Wertebereich	0.0
PI[2].TMV	TIME	Zeitwert [2]: Stützpunkt 2	gesamter Wertebereich	T#1 s
PI[3].OUTV	REAL	Ausgangswert [3]: Stützpunkt 3	gesamter Wertebereich	0.0
PI[3].TMV	TIME	Zeitwert [3]: Stützpunkt 3	gesamter Wertebereich	T#1 s
PI[4].OUTV	REAL	Ausgangswert [4]: Stützpunkt 4	gesamter Wertebereich	0.0
PI[4].TMV	TIME	Zeitwert [4]: Stützpunkt 4	gesamter Wertebereich	T#0 s

### 9.3 Parameter der Funktion LP\_SCHED

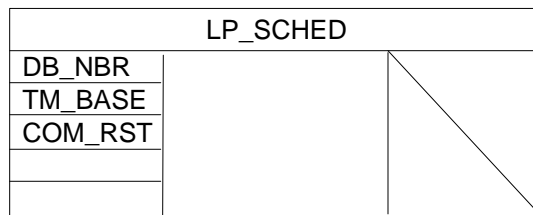


Bild 9-1 Funktion LP\_SCHED

Tabelle 9-14 Eingangsparmeter von LP\_SCHED

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
TM_BASE	TIME	Zeitbasis (Zeittakt der Weckalarmebene, in der LP-SCHED aufgerufen wird)	≥20 ms (S7-300) ≥5 ms (S7-400)	100 ms
COM_RST	BOOL	Neustart (Neustartroutine des LP_SCHED wird bearbeitet)		FALSE
DB_NBR	BLOCK_DB	Datenbausteinnummer (DB mit den Aufrufdaten der Regelkreise)		DB 1

Tabelle 9-15 Globaler Datenbaustein "DB\_NBR"

Parameter	Datentyp	Erläuterung	zulässiger Wertebereich	Vorbelegung
GLP_NBR	INT	größte Regelkreisnummer	1 ... 256	2
ALP_NBR	INT	aktuelle Regelkreisnummer	1 ... 256	0
LOOP_DAT[1] MAN_CYC	TIME	Regelkreisdaten [1]: manuelle Abtastzeit	≥20 ms (S7-300) ≥5 ms (S7-400)	T#1s
LOOP_DAT[1] MAN_DIS	BOOL	Regelkreisdaten [1]: manuell Regleraufruf sperren		FALSE
LOOP_DAT[1] MAN_CRST	BOOL	Regelkreisdaten [1]: manuell Neustart setzen (Anwender kann den betreffenden Regelkreis rücksetzen)		FALSE
LOOP_DAT[1] ENABLE	BOOL	Regelkreisdaten [1]: Reglerfreigabe (Anwender muss den bedingten Aufruf des Regelkreises programmieren)		FALSE
LOOP_DAT[1] COM_RST	BOOL	Regelkreisdaten [1]: Neustart (dieser Parameter wird mit COM_RST des Regelkreises verschaltet)		FALSE
LOOP_DAT[1] ILP_COU	INT	Regelkreisdaten [1]: Interner Regelkreiszähler (interne Zählvariable)		0
LOOP_DAT[1] CYCLE	TIME	Regelkreisdaten [1]: Abtastzeit	≥20 ms (S7-300) ≥5 ms (S7-400)	T#1s
...		...	...	...
...		...	...	...

# Projektiersoftware für Standard PID Control

# 10

## Voraussetzungen

STEP 7 muss korrekt auf Ihrem PG/PC installiert sein.

## Lieferform

Die Auslieferung der Software erfolgt auf CD.

## Installation

So installieren Sie die Software:

1. Legen Sie die CD mit Standard PID Control Tool in das Laufwerk ein.
2. Starten Sie unter WINDOWS den Dialog zur Installation von Software durch Doppelklick auf das Symbol "Software" in "Systemsteuerung".
3. Wählen Sie im Dialog das Laufwerk und die Datei Setup.exe aus und starten Sie den Installationsvorgang.

Dabei wird das Konfigurationswerkzeug auf Ihrem PG/PC installiert.

4. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationswerkzeug anzeigt.

## Liesmich-Datei auslesen

Gegebenenfalls sind wichtige aktuelle Informationen über die gelieferte Software in einer Liesmich-Datei abgelegt. Diese Datei finden Sie im Startmenü von WINDOWS unter SIMATIC\STEP7\Hinweise.

## Zweck

Das Konfigurationswerkzeug soll Ihnen die Parametrierung und Inbetriebnahme des Standard-Reglerbausteines erleichtern, sodass Sie sich verstärkt dem eigentlichen Regelungsproblem zuwenden können.

Mit dem Konfigurationswerkzeug können Sie die Standard-Reglerbausteine

- PID\_CP (Regler mit kontinuierlichem Ausgang)
- PID\_ES (Regler mit Ausgang für Schritttregelung)

parametrieren und damit optimal an die Prozesseigenschaften der Regelstrecke anpassen.

## Die Funktionen des Konfigurationswerkzeuges

Die Leistungen des Konfigurationswerkzeuges sind in einzelne Funktionen aufgeteilt. Jede dieser Funktionen läuft in einem eigenen Fenster ab. Eine Funktion kann auch mehrfach aufgerufen werden, d. h. Sie können z. B. die Kreisbilder von mehreren Reglern simultan darstellen.

### Regler beobachten

Mit Hilfe der Funktion **Kurvenschreiber** können Sie für von Ihnen ausgewählte Variablen des Regelkreises den Werteverlauf über einen definierten Zeitraum aufzeichnen und darstellen lassen. Es sind bis zu vier Variablen gleichzeitig darstellbar.

Mit der Funktion **Kreisbild** können Sie sich die relevanten Regelkreisgrößen (Sollwert, Stellwert und Istwert) eines von Ihnen ausgewählten Reglers darstellen lassen. Weiter werden Grenzwertüberschreitungen der Prozessvariablen angezeigt.

### Regler optimieren

Mit Hilfe der Funktion **Regleroptimierung** können Sie für einen vorgegebene Regelkreis die optimale Reglereinstellung ermitteln. In einem experimentellen Verfahren werden die Kennparameter der Regelstrecke ermittelt. Daraus werden die optimalen Reglerparameter berechnet und zu Ihrer weiteren Verwendung bereitgestellt.

Es spielt dabei keine Rolle, ob die aufgezeichneten Einschwingvorgänge von einem Regler stammen, der auf einen simulierten Prozess wirkt oder der online den realen Prozess beeinflusst.

### Regler bedienen

Über die Funktion **Kreisbild** können Sie die relevanten Regelkreisgrößen des aktuell angezeigten Reglers verändern bzw. neu vorgeben.

### Integrierte Hilfe

Zum Konfigurationswerkzeug gibt es eine integrierte Hilfe, die Sie unterstützt. Sie haben folgende Möglichkeiten, die integrierte Hilfe aufzurufen:

- Über den Menübefehl **Hilfe > Hilfethemen**
- Durch Drücken der Taste F1
- Durch Anklicken des Hilfe-Buttons in den einzelnen Masken
- Über den Menübefehl **Hilfe > Hilfe zum Kontext** und anschließender Auswahl des Funktionsblocks oder Parameters, für den Sie Hilfe benötigen
- Über den Button "Hilfe" (Pfeil mit Fragezeichen) in der Symbolleiste und anschließender Auswahl des Funktionsblocks oder Parameters, für den Sie Hilfe benötigen

Wenn Sie mit der Maus auf ein Eingabefeld oder eine Verbindungslinie im Hauptfenster zeigen, werden Ihnen der Parametername und die Adresse im Datenbaustein angezeigt. Wenn Sie den Baustein online geöffnet haben, wird Ihnen zusätzlich der Online-Wert der Variablen angezeigt.

# Literaturverzeichnis

# A

- /70/** Handbuch: *Automatisierungssystem S7-300, Aufbauen*
- /71/** Referenzhandbuch: *Automatisierungssysteme S7-300, M7-300 Baugruppendaten*
- /100/** Installationshandbuch: *Automatisierungssysteme S7-400, M7-400, Aufbauen*
- /101/** Referenzhandbuch: *Automatisierungssysteme S7-400, M7-400 Baugruppendaten*
- /231/** Handbuch: *Hardware konfigurieren und Verbindungen projektieren mit STEP 7 V5.1*
- /232/** Referenzhandbuch: *Anweisungsliste (AWL) für S7-300/400,*
- /234/** Handbuch: *Programmieren mit STEP 7 V5.1*
- /352/** J. Gißler, M. Schmid: *Vom Prozess zur Regelung. Analyse, Entwurf, Realisierung in der Praxis.* Siemens AG. ISBN 3-8009-1551-0.



# Glossar

## **Abtastregler** (sampling controller)

Mit Abtastregler bezeichnet man einen Regler, der in konstanten Zeitabständen die analogen Eingangsgrößen (Sollwert, Istwert) erfasst, diese bis zum nächsten Abtastzeitpunkt speichert und die Stellgröße berechnet.

## **Abtastzeit $T_A$** (sample time)

Die Abtastzeit ist die Zeitspanne zwischen zwei Abtastzeitpunkten bzw. Bearbeitungszyklen des Regelalgorithmus für einen bestimmten Mess-/Regelkanal. Diese Zeitabstände sind konstant und an das Zeitverhalten der Regelstrecke anpassbar:

$$T_A = \text{CYCLE.}$$

## **Analogwert Ein-/Ausgabe** (analog in-/output)

Der Analogwert Ein-/Ausgabe (CRP\_IN bzw. CRP\_OUT) ist ein Algorithmus zur Wandlung eines Eingangswertes im Peripherie(daten-)format in einen Gleitpunktwert und Normierung auf Prozent, entsprechend Wandlung eines internen Prozentwertes in das Peripherieformat der Ausgangsgröße.

## **Anpassfaktor** (alignment factor)

Bei einer Verhältnisregelung werden mit dem Anpassfaktor FAC die Führungsgrößen der Regelkreise aneinander angepasst, und zwar so, dass das eingestellte Verhältnis dem tatsächlichen Verhältnis der beiden Regelgrößen entspricht (→ Verhältnisregelung).

Bei einer Mischungsregelung wird mit dem Anpassfaktor FAC der gewünschte Mengenanteil der einzelnen Komponenten eingestellt. Die Summe der Mischungsfaktoren FAC muss dabei 1 sein (→ Mischungsregelung).

**Ansprechschwelle des Schrittregler-Ausgangs** (response threshold)

Die Ansprechschwelle des Schrittreglers wird im Dreipunktglied (THREE\_ST) automatisch angepasst. Dies führt zu einer Reduzierung der Impulse und schont die Stellventile. Zusätzlich kann die Länge der Impulse und der Impulspause über die Mindestimpulsdauer bzw. die Mindestpausendauer eingestellt werden.

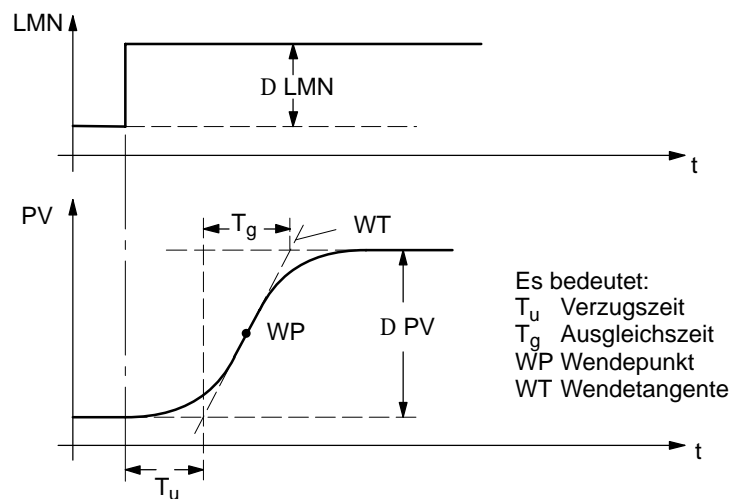
Die Mindestimpulsdauer (PULSE\_TM) oder die Mindestpausendauer (BREAK\_TM) bestimmen, wie lange ein Ausgang mindestens ein- oder ausgeschaltet wird.

**Arbeitspunkt** (operating point)

Der Arbeitspunkt bezeichnet den Stellwert, bei dem die Abweichung der Regelgröße vom Sollwert zu Null wird. Von Bedeutung ist dieser Wert bei Reglern ohne I-Anteil, bei denen im Beharrungszustand eine bleibende Regeldifferenz zur Aufrechterhaltung des benötigten Stellwertes notwendig ist. Soll diese Differenz verschwinden, dann ist der Arbeitspunktparameter entsprechend anzupassen.

**Ausgleichszeit** (control settling time)

Bei einer Sprungantwort einer PT-Strecke (= Strecke mit Ausgleich) höherer Ordnung versteht man unter der Ausgleichszeit den Abschnitt, den die Wendetangente auf den durch Anfangs- und Endwert gelegten Parallelen zur Zeitachse abschneidet.



Übergangsfunktion einer PT3-Strecke

**Ausregelzeit** (control settling time)

Die Ausregelzeit ist die Zeitspanne vom Verlassen des bisherigen Beharrungszustandes bis zum letztmaligen Eintauchen der Regelgröße in das Toleranzband von  $\pm 5\%$  um den Sollwert nach Führungsgrößenänderungen oder Störungen.



**Auswahlregelung** (selection control)

Die Auswahlregelung wird bei Prozessen eingesetzt, die bei verschiedenen Betriebsbedingungen unterschiedliche Regelungsstrukturen verlangen. Es ist ein Kriterium zur Auslösung des Umschaltvorganges erforderlich.

**Automatikbetrieb** (automatic mode)

Der Regler arbeitet und berechnet die Stellgröße mit dem Ziel, die Regeldifferenz zu minimieren (Regelkreis geschlossen).

**AS** (PC)

Automatisierungssystem, bestehend aus einer oder mehreren Zentralprozessoreinheit(en) (CPU), den Peripherieeinheiten mit digitalen/analogen Ein- und/oder Ausgängen, den Einheiten zur Kopplung und Kommunikation mit weiteren Systemeinheiten und ggf. einer Stromversorgung.

**Begrenzung** (limiter)

Algorithmus (Funktion) zur Einschränkung des Wertebereiches stetiger Größen auf vorgebbare obere/untere Begrenzungswerte.

**D-Anteil** (derivative component)

Mit D-Anteil bezeichnet man den Differentialanteil des Reglers.

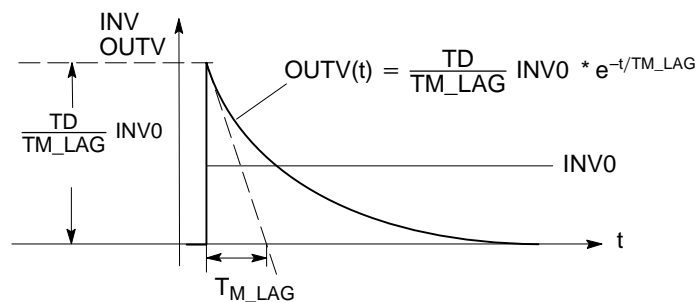
D-Glieder allein sind zur Regelung ungeeignet, da sie bei Einstellung der EingangsgroÙe auf einen Beharrungswert kein Ausgangssignal abgeben.

**DDC** (direct digital control)

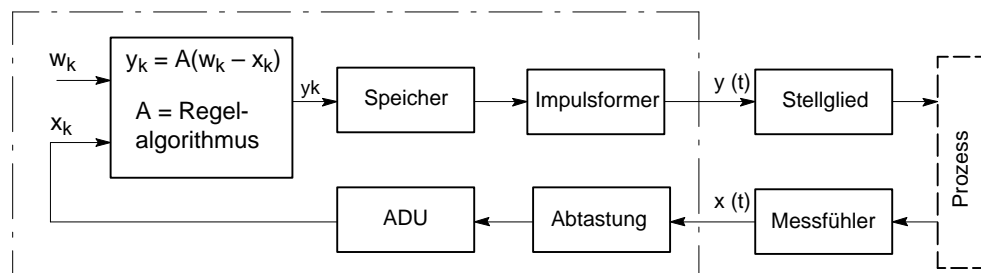
DDC ist eine diskrete Regelung, bei der die Regeldifferenz jeweils zum Abtastzeitpunkt aktualisiert wird (→ Abtastzeit, → Digitale Regelung).

**Differenzierung** (derivative action)

Ein Verfahren (Algorithmus) zur Differenzierung einer analogen Größe, wobei das Zeitverhalten durch die Differenzierzeit  $TD$  (= Vorhaltzeit) bestimmt wird. Das Ausgangssignal des Differenziergliedes ist proportional zur zeitlichen Ableitung seines Eingangssignals. Zur Unterdrückung von differentiellen Spitzen bzw. von Störsignalen ist eine Verzögerung erster Ordnung  $TM\_LAG$  vorgesehen. Die Sprungfunktion hat folgende Form:

**Digitale Regelung (Abtastregelung)** (digital control)

Regler der in konstanten Zeitabständen ( $\rightarrow$  Abtastzeit) einen neuen Wert der Regelgröße (Istwert) erfasst und danach abhängig vom Wert der aktuellen Regeldifferenz einen neuen Wert für die Stellgröße berechnet.

**Dreipunktregler** (three step controller)

Regler, bei dem die Ausgangsgröße nur drei diskrete Zustände annehmen kann: z. B. "heizen – aus – kühlen" oder "rechts – Stillstand – links" ( $\rightarrow$  Schrittreger).

**Fahrkurve**

$\rightarrow$  Zeitplangeber

**Festwertregelung** (fixed setpoint control)

Als Festwertregelung bezeichnet man eine Regelung mit einer festen, nur gelegentlich veränderten Führungsgröße. Sie regelt im Prozess auftretende Störgrößen aus.

**Folgeregelung** (follow-up control)

Als Folgeregelung bezeichnet man eine Regelung, bei welcher der Führungswert laufend von außen beeinflusst wird (unterlagerter Regler einer mehrschleifigen Regelung). Aufgabe des Folge-reglers ist, die lokale Regelgröße möglichst schnell und genau mit der Führungsgröße zur Deckung zu bringen.

**Führungsgröße** (setpoint)

Die Führungsgröße gibt den gewünschten Wert bzw. Verlauf der interessierenden Prozessgröße vor. Ihr Momentanwert heißt → Sollwert (SP).

**Führungsregler** (master controller)

Der Führungsregler ist ein überlagerter Regler einer mehrschleifigen Regelung. Erzeugt den Führungs-(Soll)wert für den nachgeschalteten (Folge-)Regler (→ Kaskadenregelung).

**Führungsverhalten** (master control behavior)

Das Führungsverhalten ist das Verhalten (zeitlicher Verlauf) der Regelgröße im geschlossenen Regelkreis nach einer (sprungförmigen) Änderung des Sollwertes.

**Grenzwertmelder** (limit alarm monitor)

Algorithmus (Funktion) zur Überwachung einer analogen Größe auf vier vorgebbare Grenzen. Bei Erreichen und Über-/Unterschreiten dieser Grenzen wird ein zugehöriges Warnsignal (1. Grenze) oder Alarmsignal (2. Grenze) erzeugt. Zur Verhinderung von Signalflackern ist die Ausschaltsschwelle (Rückschaltdifferenz) der Grenzsignale über einen Parameter für die Hysterese einstellbar.

**Handbetrieb** (manual processing)

Beeinflussung des Wertes der Stellgröße (LMN) durch Handbedienung. Der aktuelle Stellwert wird vom Bediener oder STEP 7-Anwenderprogramm als Prozentwert des Stellbereiches vorgegeben.

Die Umschaltung zwischen Automatik- und Handbetrieb kann 'stoßfrei' erfolgen, wenn rampenförmige An- und Abstiegsbegrenzungen der Stellgröße vorgesehen werden (Funktion: LMN\_ROC).

**Handstellwert** (manual value)

Bei aufgetrenntem Regelkreis (→ Handbetrieb) manuell absolut oder inkrementell (über Schalterbedienung) vorgegebener Wert der Stellgröße (in % des Stellbereichs).

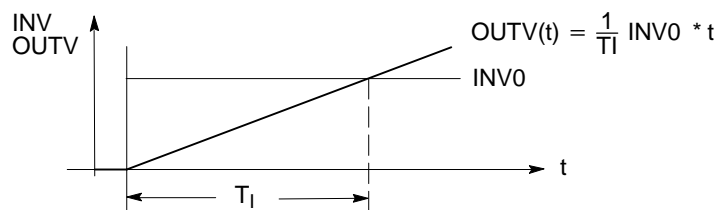
**I-Anteil** (integral component)

Integralanteil des Reglers.

Nach einer sprungförmigen Änderung der Regelgröße (bzw. Regeldifferenz) ändert sich die Ausgangsgröße rampenförmig über der Zeit, und zwar mit einer Änderungsrate, die dem Integrierbeiwert  $K_I (= 1/T_I)$  proportional ist. Der Integralanteil bewirkt im geschlossenen Regelkreis, dass die Reglerausgangsgröße so lange verstellt wird, bis die Regeldifferenz zu Null geworden ist.

**Integration** (integral action)

Verfahren (Algorithmus) zur Integration einer analogen Größe, wobei das Zeitverhalten durch die Integrationszeit  $T_I$  (Nachstellzeit) bestimmt wird. Das Ausgangssignal des Integrierers ändert sich geschwindigkeitsproportional zur statischen Änderung des Eingangssignals. Der Integrierbeiwert  $K_I = 1/T_I$  ist ein Maß für die Anstiegsgeschwindigkeit des Ausgangssignals bei von Null verschiedenem Eingangssignal. Die Sprungfunktion hat folgende Form:

**Impulsbreitenmodulation** (pulse width modulation)

Die Impulsbreitenmodulation ist ein Verfahren zur Beeinflussung der Stellgröße bei schaltendem Ausgang. Der errechnete Stellwert in Prozent wird in eine proportionale Einschaltdauer (ED) des betreffenden Stellausgangs umgeformt, z. B. ist 100 %  $ED = T_A$  bzw. = CYCLE.

**Interpolation** (interpolation)

Die Interpolation ist ein Verfahren zur Ermittlung von Zwischenwerten aufgrund bekannter Randwerte eines Intervalls (→ Zeiplangeber).

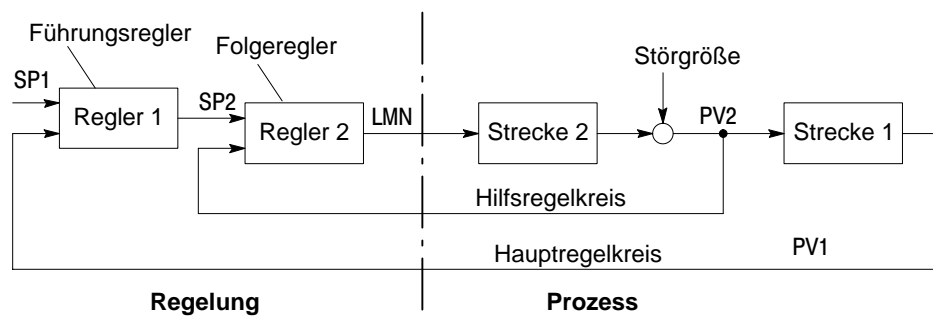
**Istwert** (process variable)

Momentanwert der Regelgröße PV

### Kaskadenregelung (cascade control)

Die Kaskadenregelung ist eine Hintereinanderschaltung von Reglern, wobei der erste Regler (Führungsregler) den nachgeschalteten Reglern (Folgeregler) den Sollwert vorgibt bzw. deren Sollwerte gemäß der aktuellen Regeldifferenz der Hauptregelgröße beeinflusst.

Durch Einbeziehen von zusätzlichen Prozessgrößen lässt sich das Regelergebnis mit einer Kaskadenregelung verbessern. Dazu wird an geeigneter Stelle eine Hilfsregelgröße PV2 erfasst und diese auf den Führungssollwert (Ausgang des Führungsreglers SP2) geregelt. Der Führungsregler regelt den Istwert PV1 auf den Festen Sollwert SP1 und stellt dazu SP2 so ein, dass dieses Ziel möglichst schnell und überschwingungsfrei erreicht wird.



### Konfiguration (configuration)

Werkzeug (Software) zur Erstellung und Parametrierung einer Standard-Regelung sowie zur Optimierung des Reglers mit Hilfe der aus einer Streckenidentifikation gewonnenen Daten.

### Korrekturprofil (adjustment profile)

Bei Mischungs- und Kaskadenregelungen mit mehreren Folgekreisen kann deren Sollwertbeeinflussung durch jeweils einen spezifischen Bewertungsfaktor (FAC) dimensioniert werden. Dieser kennzeichnet jeweils das Maß für die Stärke des Eingriffs an dieser Stelle, woraus sich insgesamt das Korrekturprofil ergibt.

### Kreisverstärkung (loop gain)

Die Kreisverstärkung ist ein Produkt aus Proportionalbeiwert (GAIN) und Verstärkung der Regelstrecke ( $K_S$ )

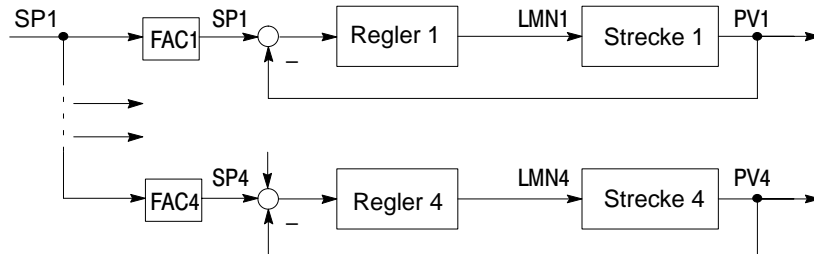
### Lineare Skalierung (linear scaling)

Die lineare Skalierung ist eine Funktion zur Umwandlung bzw. Korrektur von Prozesswerten.

Algorithmus: Ausgang = Eingang \* FACTOR + OFFSET

### Mischungsregelung (blending control)

Die Mischungsregelung ist eine Regelungsstruktur, bei welcher der Sollwert für die Gesamtmenge SP prozentual auf die gewünschten Mengenanteile der einzeln geregelten Komponenten umgerechnet wird. Die Summe der Mischungsfaktoren FAC muss dabei 1 sein.



### Modulare Regelung (modular PID control)

Die modulare Regelung ist eine Regelungsstruktur, bei der signalverarbeitende und regelungstechnische Funktionen vom Anwender freizügig zusammengestellt werden können. Der so konfigurierte Regler lässt sich dadurch auch strukturell optimal an die jeweilige Aufgabenstellung anpassen (eigenes S7-Softwarepaket).

### Nachstellzeit $T_N$ (reset time)

Die Nachstellzeit bestimmt das Zeitverhalten des I-Anteils im PI- oder PID-Regler ( $T_N = T_I$ ).

### Neustart (complete restart)

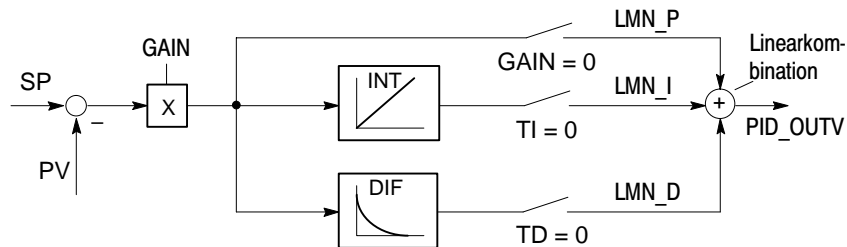
Bei Neustart einer Regelung kann von einem definierten Anfangszustand aus gestartet werden. Die Ausgangsparameter und lokalstatischen Daten des Reglers werden in der 'Neustarroutine' vorbesetzt.

### Normierung (physical norm)

Die Normierung ist ein Verfahren (Algorithmus) zur Umwandlung (Normierung) der physikalischen Werte einer Prozessgröße in die (intern verarbeiteten) Prozentwerte der Standard-Regelung und Umwandlung umgekehrt am Ausgang. Die Normierungsgerade wird durch den Anfangswert (OFFSET) und die Steigung (FACTOR) festgelegt.

**Parallelstruktur** (parallel structure)

Die Parallelstruktur ist eine spezielle Art der Signalverarbeitung im Regler (Art der mathematischen Bearbeitung). P-, I- und D-Anteil werden als interaktionsfrei parallel wirkend berechnet und dann summiert.

**P-Regler** (P algorithm)

Algorithmus zur Errechnung eines Ausgangssignals, bei welchem ein proportionaler Zusammenhang zwischen Regeldifferenz und Stellgrößenänderung besteht. Merkmale: bleibende Regeldifferenz, an Totzeitstrecken nicht zu verwenden.

**PI-Regler** (PI algorithm)

Algorithmus zur Errechnung eines Ausgangssignals, bei dem sich die Änderung der Stellgröße zusammensetzt aus einem zur Regeldifferenz proportionalen Anteil und einem I-Anteil, welcher dem Wert der Regeldifferenz und der Zeit proportional ist. Merkmale: keine bleibende Regeldifferenz, schnellere Ausregelung als beim I-Regler, für viele Strecken geeignet.

**PID-Regler** (PID algorithm)

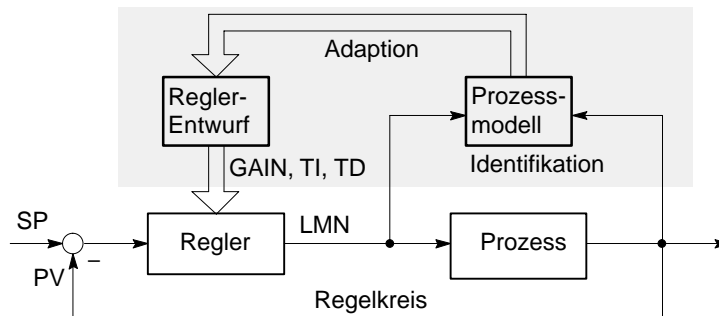
Algorithmus zur Errechnung eines Ausgangssignals, das durch Multiplikation, Integration und Differentiation aus der Regeldifferenz gebildet wird. Der PID-Algorithmus ist in reiner  $\rightarrow$  Parallelstruktur ausgeführt. Merkmal: hohe Regelgüte erreichbar, solange die Totzeit der Regelstrecke nicht größer ist als die Summe der übrigen Zeitkonstanten.

**Physikalische Normierung** (physical norm)

$\rightarrow$  Normierung

**Prozessidentifikation** (process identification)

Die Prozessidentifikation ist eine Funktion des Konfigurationswerkzeugs, welche Informationen über das Übertragungsverhalten und die Struktur des Prozesses liefert. Als Ergebnis wird ein geräteunabhängiges Prozessmodell ermittelt, das den Prozess in seinem statischen und dynamischen Verhalten beschreibt. Davon abgeleitet werden optimale Werte für die Reglerkenngrößen errechnet (Reglerentwurf).

**Quadratwurzel**

→ Radizierung

**Radizierung** (square root)

Mit der Quadratwurzelfunktion SQRT können quadratische Kennlinien linearisiert werden.

**Rampengeber** (rate of change, ROC)

Verfahren zur Begrenzung der Änderungsgeschwindigkeit analoger Größen, getrennt für den An- und Abstieg. Aus Sprungfunktionen am Eingang werden Funktionen mit endlicher Steigung am Ausgang.

**Regeldifferenz** (error signal, ER)

Die Regeldifferenz ist eine Funktion zur Bildung der Regeldifferenz  $ER = SP - PV$ . An der Vergleichsstelle wird die Differenz zwischen gewünschtem Sollwert und dem tatsächlich vorhandenen Istwert gebildet. Dieser Wert wird dem Regelalgorithmus als Eingang übergeben. Alte Bezeichnung: Regelabweichung.

**Regeldifferenzüberwachung** (error signal monitoring)

Die Regeldifferenzüberwachung ist eine Funktion zur Überwachung der Werte (Amplitude) der Regeldifferenz auf vier vorgebbare Grenzen. Bei Erreichen und Über-/Unterschreiten dieser Grenzen wird ein zugehöriges Warnsignal (1. Grenze) oder Alarmsignal (2. Grenze) erzeugt. Zur Verhinderung von Signalflickern ist die Ausschaltswelle (Rückschaltdifferenz) der Grenzschnelle über einen Parameter für die Hysterese einstellbar.



**Regleinrichtung** (Control device)

Gesamtheit von Regler, Stellgerät und Aufnehmer (Messeinrichtung) für die Regelgröße.

**Regelgröße** (process variable, PV)

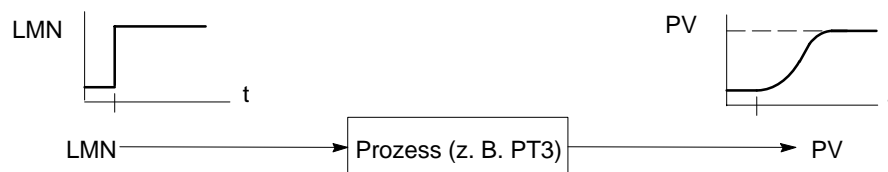
Prozessgröße (Ausgangsgröße der Regelstrecke), die an den Momentanwert der Führungsgröße angeglichen werden soll. Ihr Momentanwert heißt Istwert.

**Regelkreis** (control loop)

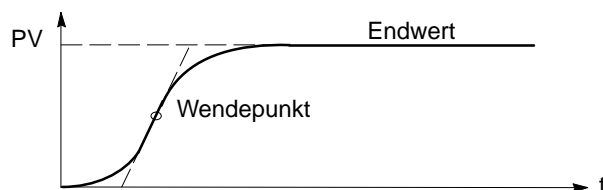
Mit Regelkreis bezeichnet man die Verbindung des Streckenausgangs (Regelgröße) mit dem Reglereingang und des Reglerausgangs (Stellgröße) mit dem Prozesseingang, so dass Regler und Prozess einen geschlossenen Wirkungskreis bilden.

**Regelstrecke** (process unit)

Mit Regelstrecke bezeichnet man den Anlagenteilbereich, in dem die Regelgröße von der Stellgröße beeinflusst wird, z. B. durch Änderung der Stellenergie oder des Massenstroms.

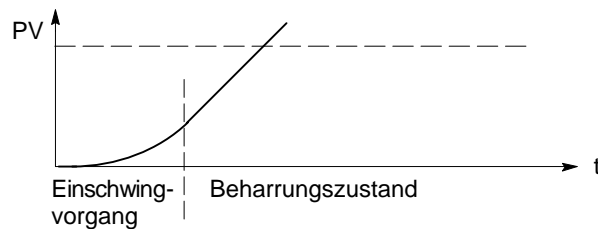
**Regelstrecke mit Ausgleich** (self-regulating process)

Man spricht von einer Strecke mit Ausgleich, wenn die Sprungantwort der Regelgröße einen stationären Endwert erreicht (PT-Verhalten).



**Regelstrecke ohne Ausgleich** (not balanced process)

Man spricht von einer Strecke ohne Ausgleich, wenn die Steigung der Regelgröße als Sprungantwort auf eine Störung oder Stellgrößenänderung im Beharrungszustand proportional zum Eingangssprung ist (I-Verhalten).

**Regler** (closed-loop controller)

Ein Regler ist eine Einrichtung, welche die Regeldifferenz fortlaufend erfasst (Vergleicher) und ggf. eine zeitabhängige Funktion zur Bildung des Stellsignals (Ausgangsgröße) erzeugt, mit dem Ziel, die Regeldifferenz schnell und überschwingungsfrei zum Verschwinden zu bringen.

**Regleraufrufverteiler** (loop scheduler)

Der Regleraufrufverteiler organisiert das zeitliche Aufrufverhalten mehrerer Regler in einer Weckalarmebene sowie den Aufruf der Reglerstrukturen bei Neustart. Wird dann eingesetzt, wenn viele Regler, die in den Weckalarmebenen nicht untergebracht werden können, oder Regler mit großen Abtastzeiten aufgerufen werden sollen.

**Reglerparameter** (control parameter)

Reglerparameter sind Kennwerte für die statische und dynamische Anpassung des Reglerverhaltens an die gegebenen Strecken- bzw. Prozesseigenschaften.

**Schrittregler** (step controller)

Der Schrittregler ist ein quasistetiger Regler mit diskontinuierlichem Ausgang (und motorischem Stellantrieb mit I-Wirkung). Das Stellsignal hat Dreipunktverhalten, z. B. aufwärts – halt – abwärts oder Schließen – Konstanthalten – Öffnen (→ Dreipunktregler).

**Signalflussplan** (signal flow graph)

Der Signalflussplan stellt die aus regelungstechnischer Sicht wichtigen Wirkzusammenhänge einer Regeleinrichtung oder eines Prozesses dar. Der Signalflussplan besteht aus Übertragungsblöcken, die das Übertragungsverhalten der realen Regelkreisglieder kennzeichnen und aus Wirkungslinien, welche die Wirkungsrichtungen darstellen.

**Sollwert** (setpoint value, SP)

Der Sollwert ist der Wert, den die Regelgröße durch Einwirken eines Reglers annehmen soll.

**Sollwertgenerator** (setpoint value generator)

Mit Sollwertgenerator bezeichnet man eine Funktion zur getasteten Veränderung des Sollwertes mit Hilfe von Schaltern. In den ersten 3 Sekunden nach Betätigen des Gebers beträgt die Änderungsgeschwindigkeit nur 10 % der entgültigen Geschwindigkeit, die proportional zur Größe des zulässigen Verstellbereiches ist.

**Standard-Regelung** (standard PID control)

Eine Standardregelung ist eine fertige feste Reglerstruktur, die alle Funktionen einer Regleranwendung enthält. Der Anwender kann durch "Softwareschalter" Funktionen zu- oder abschalten.

**Stellgerät** (process control unit)

Mit Stellgerät bezeichnet man den Teil des Regelkreises, der zum Beeinflussen der Stellgröße am Prozesseingang dient.

**Stellgröße** (manipulated variable, LMN)

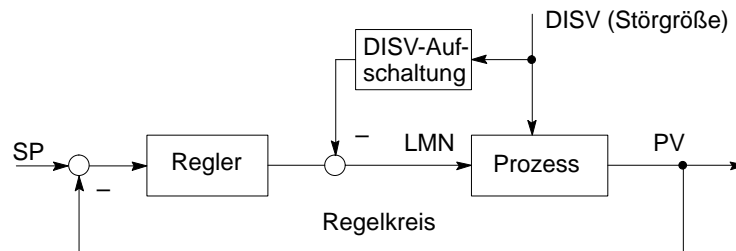
Die Stellgröße ist die Ausgangsgröße des Reglers bzw. die Eingangsgröße der Regelstrecke. Das Stellsignal kann den Stellbereich analog als Prozentwert oder als Pulsbreitenwert bzw. Impulsdauer abbilden. Bei integrierendem Stellglied (z. B. Motor) genügt es, binäre Auf-/Ab- bzw. Vor-/Zurück-Schaltsignale zu liefern.

**Störgröße** (disturbance variable, DISV)

Alle Einflussgrößen auf die Regelgröße – mit Ausnahme der Stellgröße – werden als Störgrößen bezeichnet. Additive Einflüsse auf das Streckenausgangssignal lassen sich durch Überlagerung mit dem Stellsignal kompensieren.

**Störgrößenaufschaltung** (controller with feedforward control)

Die Störgrößenaufschaltung ist ein Verfahren zur Verringerung/Beseitigung des Einflusses einer dominierenden (messbaren) Störgröße (z. B. Außentemperatur) auf den Regelkreis. Aus der gemessenen Störgröße DISV wird ein Korrekturereingriff abgeleitet, mit dem auf Änderungen von DISV schneller reagiert werden kann. Im Idealfall wird der Einfluss voll kompensiert, ohne dass der Regler selbst einen Korrekturvorgang (über I-Teil) ausführen muss.

**Strecke** (process)

→ Regelstrecke

**Totzeit** (dead time)

Mit Totzeit bezeichnet man die zeitliche Verzögerung der Regelgrößenreaktion auf Störungen oder Stellwertänderungen bei Prozessen mit Transportvorgängen. Die Eingangsgröße eines Totzeitgliedes wird um den Wert der Totzeit zeitlich versetzt am Ausgang ausgegeben.

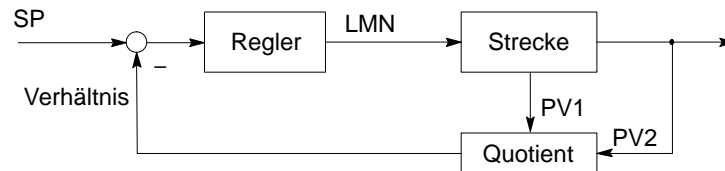
**Trapezregel** (trapeze rule)

Methode zur algorithmischen Nachbildung von kontinuierlichen I-, D- und Verzögerungs-Gliedern durch rekursive Differenzenberechnung. Bei Verwendung der Trapezregel wird der Regelalgorithmus des digitalen Reglers in seinem Verhalten dem eines analogen Reglers ähnlicher.

## Verhältnisregelung (ratio controller)

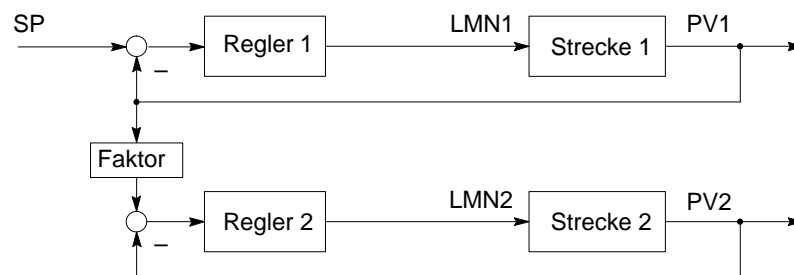
- Einschleifige Verhältnisregelung (single loop ratio controller)

Eine einschleifige Verhältnisregelung wird dann eingesetzt, wenn für einen Vorgang (z. B. Drehzahlregelung) das Verhältnis zweier Regelgrößen wichtiger ist als die Absolutwerte der Regelgrößen.



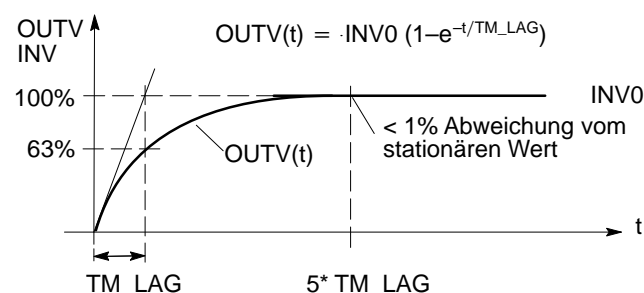
- Mehrschleifige Verhältnisregelung (multiple loop ratio controller)

Bei einer zweischleifigen Verhältnisregelung wird das Verhältnis der beiden Prozessgrößen PV1 und PV2 konstant gehalten. Dazu wird der Sollwert des 2. Regelkreises aus der Regelgröße des 1. Regelkreises berechnet. Auch bei einer dynamischen Änderung der Prozessgröße  $x_1$  ist gewährleistet, dass das vorgegebene Verhältnis eingehalten wird.



## Verzögerung 1. Ordnung (first order lag)

Die Verzögerung 1. Ordnung ist eine Funktion zur Dämpfung (zeitlichen Verzögerung) der Änderungen einer analogen Regelgröße. Die Verzögerungszeitkonstante  $TM\_LAG$  gibt an, nach welcher Zeit das Ausgangssignal auf 63 % des stationären Endwertes hochgelaufen ist. Das Übertragungsverhältnis im eingeschwungenen Zustand ist 1 : 1.



**Vorhaltezeit  $T_V$**  (derivative time)

Die Vorhaltezeit bestimmt das Zeitverhalten des D-Anteils im PD- oder PID-Reglers ( $T_V = TD$ ).

**Wertebereich** (value range)

Der Regler arbeitet intern mit Prozentwerten im Gleitpunktformat (z. B. -100,0 bis +100,0). An bestimmten Eingangsparametern, z. B. an externen Sollwerten, können auch physikalische Werte im Gleitpunktbereich von STEPP 7 eingegeben werden (→ Zahlendarstellung).

**Wiederanlauf** (restart)

Bei Wiederanlauf einer Regelung wird von den erfassten Ergebnissen und dem Betriebszustand ausgegangen, der zum Zeitpunkt der Unterbrechung geherrscht hat. Das bedeutet, dass die Regelung mit den zum Zeitpunkt der Unterbrechung berechneten Werten weiterarbeitet.

**Zeitplangeber** (ramp soak)

Der Zeitplangeber ist eine Funktion zur Erzeugung von Fahrkurven der Führungsgröße nach einem festen Zeitplan (Programm). Die zeitabhängige Verlauf der Ausgangsgröße wird über Stützpunkte und lineare Interpolation definiert. Die Fahrkurve kann zyklisch wiederholt werden.

**Zweipunktregler** (two step controller)

Mit Zweipunktregler bezeichnet man einen Regler, bei dem die Stellgröße nur zwei Zustände (z. B. ein – aus) annehmen kann.

# Index

## A

Abtastregelung, 2-11  
Abtastregler, Glossar-1  
Abtastzeit, 2-11, 3-14, 8-2, Glossar-1  
    Abschätzung, 3-15  
    Faustformel, 3-15  
Analogwert-Eingabe, Glossar-1  
Anlaufbausteine, 3-16  
Anlaufzeit, 2-3  
Anpassfaktor, Glossar-1  
Ansprechschwelle, 6-16, 6-22, Glossar-2  
    selbsttätige Adaption, 6-22  
Anwenderspeicher, 1-6  
Arbeitspunkt, Glossar-2  
Arbeitspeicherbelegung, 8-1  
Aufrufverhalten, 3-16  
Ausgleichszeit, 2-2, Glossar-2  
Ausregelzeit, Glossar-2  
Auswahl der Reglerstruktur, 3-6  
Automatikbetrieb, 5-3  
    Schrittregler, 6-6

## B

Bearbeitungsaufruf des Regler-FB, 3-16  
Beispiel Example1 (Schrittregler)  
    Anwendung, 7-10  
    Baueinstruktur, 7-11  
    Funktionalität, 7-10  
    Parameter des Streckenmodells, 7-12, 7-14  
    Sprungantwort des Regelkreises, 7-14  
    Verschaltung und Aufruf, 7-12, 7-13  
Beispiel Example2 (Kontinuierlicher Regler)  
    Anwendung, 7-16  
    Baueinstruktur, 7-17  
    Funktionalität, 7-16  
    Parameter des Streckenmodells, 7-18, 7-19  
    Sprungantworten des Regelkreises, 7-19  
    Verschaltung und Aufruf, 7-17, 7-18  
Beispiel Example3 (Verhältnisregelung)  
    Anwendung, 7-21  
    Baueinstruktur, 7-22  
    Funktionalität, 7-21  
    Parametrierung, 7-23  
    Verschaltung und Aufruf, 7-22

Beispiel Example4 (Mischungsregelung)

    Anwendung, 7-24  
    Baueinstruktur, 7-25  
    Funktionalität, 7-24  
    Parametrierung, 7-26  
    Verschaltung und Aufruf, 7-25

Beispiel Example5 (Kaskadenregelung)

    Anwendung, 7-27  
    Baueinstruktur, 7-28  
    Funktionalität, 7-27  
    Parametrierung, 7-29  
    Verschaltung und Aufruf, 7-28

Beispiel Example6 (Pulsegen)

    Anwendung, 7-30  
    Baueinstruktur, 7-31  
    Funktionalität, 7-30  
    Parameter des Streckenmodells, 7-32, 7-33  
    Sprungantworten des Regelkreises, 7-33  
    Verschaltung und Aufruf, 7-31, 7-32

Beispiele, vorkonfektionierte Anwendungen,  
    1-4

Betriebsarten-Umschaltung, 5-3

## C

Checkliste, 3-7  
CPU-Belastung, 8-1  
CRP\_OUT, 5-15

## D

D-Anteil, Glossar-3  
Dämpfung, 4-24  
Daten pro Regler, 1-6  
DDC, Glossar-3  
DEADBAND, 4-35  
    Parameter, 4-36  
DIF, Parameter, 4-52  
Differentiationszeit TD, 4-51  
Differenzier-Funktion, 4-51  
Differenzierer (DIF), 4-51  
    Anlauf und Betriebsarten, 4-52  
Differenzierzeit (Einstellbereiche), 4-51  
Digitale Regelung, Glossar-4

Dreipunktglied, 6-15, 6-21  
  Ansprechschwelle, 6-16  
Dreipunktregelung, 5-22  
  Kennlinie, 5-23  
  unsymmetrische Kennlinie, 5-24  
Dreipunktregler, Glossar-4

## E

Einsatzformen, 1-7  
ER\_ALARM, 4-37  
  Parameter, 4-38  
Ersatz-Streckenzeitkonstante, Ermittlung, 3-15

## F

Fahrkurve, 4-4  
  anhalten, fortsetzen, 4-9  
  projektieren, 4-5  
  starten, 4-7  
Festwertregelung, Glossar-5  
Folgeregelung, Glossar-5  
Folgeregler, 2-10  
Führungsgröße, Glossar-5  
  Anwender-FC verschalten, 4-15  
  Begrenzung des Einstellbereichs, 4-19  
  eigene Funktion einfügen, 4-15  
  Rampenfunktion, 4-17  
  Steigungsbegrenzung, 4-17  
  Verstellung über Konfigurationswerkzeug,  
    4-21  
Führungsgrößenbegrenzung, Meldeausgänge,  
  4-19  
Führungsregler, 2-10, Glossar-5  
Führungsverhalten, Glossar-5  
Funktionalität im Auslieferungszustand, 2-15  
Funktionsbaustein  
  PID\_CP, 5-1  
  PID\_ES, 6-1  
Funktionsspektrum, 1-7  
Funktionsübersicht, 2-12

## G

Grenzwerte für PV, 4-30  
Grenzwertmelder, Glossar-5

## H

Handbetrieb, 5-3, Glossar-5  
  Schrittregler (mit Rückmeldung), 6-6  
  Schrittregler (ohne Rückmeldung), 6-19  
Handstellwert (Handwert), Glossar-6  
Handwert-Erzeugung, 5-3  
Handwert-Generator  
  Anlauf und Betriebsweise, 5-6  
  Verstellbereich, 5-5  
  Verstellgeschwindigkeit, 5-5  
Hardwareumgebung, 1-5  
Hilfsregelgröße, 2-10

## I

I-Anteil, Glossar-6  
I-Strecke, 2-4  
Impulsausgang, Schaltverhalten, 5-23  
Impulsbreitenmodulation, 5-19, Glossar-6  
Impulsformer, 5-19, 6-16  
  Arbeitsweise, 6-16, 6-17  
  Betriebsarten, 5-22  
  Impulsrasterbreite, 5-20  
Inbetriebnahmewerkzeug, 10-1  
Instanz-Datenbaustein, 1-1  
Integrationszeit (TI), 4-46  
  zulässige Bereiche für TI und CYCLE, 4-47  
Integrator (INT), 4-46  
  Anlauf und Betriebsarten, 4-48  
  Begrenzungsverhalten, 4-50  
  Parameter, 4-50  
Integrier-Funktion, 4-46  
Istwert, Glossar-6  
  anwenderspezifische Funktion (FC), 4-28  
  Grenzwertüberwachung, 4-30  
  Normierung, 4-22  
  Steigungsüberwachung, 4-32  
  Umschalten auf Konfigurationswerkzeug,  
    4-34  
  Verzögerung, 4-24  
Istwert-Überwachung, Hysterese, 4-30

## K

Kaskadenregelung, 2-10, Glossar-7  
  Bausteinverschaltung, 5-18, 6-26



Kaskadenregelung (Example5), 7-27  
 Bausteinstruktur, 7-28  
 Kaskadenschaltungen, 5-17  
 Kaskadenumschaltung, 5-17, 6-25  
 Konfiguration, Glossar-7  
 Istwert-/Regeldifferenzweig, 3-11  
 Reglerfunktionen, 3-13  
 Sollwertweig, 3-10  
 Stellwertweig, 3-12  
 Vorgehensweisen, 1-2  
 Konfigurationswerkzeug, 3-14  
 Kontinuierlicher Regler  
 Betriebsarten-Schaltung, 5-4  
 Differenzierer, 4-51  
 Funktionsschema, 5-1  
 Integrierer, 4-46  
 Kaskadenschaltungen, 5-17  
 Neustart/Wiederanlauf, 5-2  
 P-Regelung, 4-41  
 PD-Regelung, 4-43  
 PI-Regelung, 4-42  
 PID-Regelung, 4-44  
 Regelfunktionen, 5-1  
 Wirkungsrichtung, 4-41  
 Kontinuierlicher Regler (Example2), 7-16  
 Bausteinstruktur, 7-17  
 Korrekturprofil, Glossar-7  
 Kreisbild, 10-2  
 Kreisverstärkung, Glossar-7  
 Kurvenschreiber, 10-2

**L**

LAG1ST, 4-24  
 Parameter, 4-25  
 Laufzeit (Regler-FB), 8-1  
 Laufzeit pro Regler (Eckdaten), 1-6  
 Liesmich-Datei, 10-1  
 LMN\_NORM, 5-13  
 Parameter, 5-14  
 LMN\_ROC, 5-9  
 Parameter, 5-10  
 LMNFC, 5-7  
 Parameter, 5-8  
 LMNLIMIT, 5-11  
 Parameter, 5-12, 6-10  
 LMNR\_CRP, 6-11  
 Parameter, 6-12  
 LMNRNORM, 6-12  
 Parameter, 6-12

LP\_SCHED  
 Parameter, 7-9  
 Parameterliste, 9-20

**M**

MAN\_GEN, 5-5, 5-6  
 Mehrschleifige Regelungen, 1-4, 2-8  
 Mengengerüst, 1-6  
 Mindestimpulsdauer, 5-22  
 Mindestpausendauer, 5-22  
 Mischungsregelung, 2-9, Glossar-8  
 Regelungsstruktur, 2-9  
 Mischungsregelung (Example4), 7-24  
 Bausteinstruktur, 7-25  
 Modulare Regelung, Glossar-8

**N**

Nachstellzeit, Glossar-8  
 Neustart, 3-16  
 Normierung, 3-18, Glossar-8  
 Regelgröße, 4-22  
 Sollwert, 4-12  
 Stellgröße, 5-13  
 Stellungsrückmeldung, 6-12  
 Normierungsfunktion, 3-18, 5-13  
 Normierungsgerade, 4-22, 5-13, 6-12

**P**

P-Regelung  
 Arbeitspunkt, 4-41  
 Sprungantwort, 4-41  
 Parallelstruktur (PID), Glossar-9  
 Parametrierplan, 3-10  
 PD-Regler  
 Arbeitspunkt, 4-43  
 Sprungantwort, 4-43  
 Verzögerung der D-Wirkung, 4-43  
 PD-Wirkung in Rückführung, 4-40  
 PI-Regler  
 Integrierer bei Handbetrieb, 4-42  
 Sprungantwort, 4-42  
 PID-Regler  
 Parametrierung, 4-45  
 Regelalgorithmus, 4-39  
 Reglerstruktur, 4-40  
 Sprungantwort, 4-44

- PID\_CP
    - Ausgangsparameter, 9-4
    - Eingangsparameter, 9-2, 9-5
    - Statische Lokaldaten (Ausgänge), 9-9
    - Statische Lokaldaten (Eingänge), 9-5
    - Statische Lokaldaten für Konfigurationswerkzeug, 9-10
  - PID\_ES
    - Ausgangsparameter, 9-13
    - Eingangsparameter, 9-11
    - Statische Lokaldaten (Ausgänge), 9-17
    - Statische Lokaldaten (Eingänge), 9-13
    - Statische Lokaldaten für Konfigurationswerkzeug, 9-18
  - Prioritätsklassenmodell, 3-17
  - Produktstruktur, 1-3
  - Projekt konfigurieren, 3-7
  - Prozesse mit I-Verhalten, 2-4
  - Prozesseigenschaften und Regelung, 2-1
  - Prozessidentifikation, Glossar-10
  - Prozessverhalten, 3-1
    - regelbare Streckenarten, 3-3
  - PULSEGEN, 5-19
    - Parameter, 5-26
  - PULSEGEN (Example6), 7-30
    - Baueinstruktur, 7-31
  - PULSEOUT, 6-16
    - Parameter, 6-17
  - Pulsformung, Genauigkeit, 5-20
  - PV-Grenzwertmeldung, Betriebsweise, 4-31
  - PV\_ALARM, 4-30
    - Parameter, 4-31
  - PV\_NORM, 4-22
    - Parameter, 4-23
  - PVFC, 4-28
    - Parameter, 4-29
- R**
- Radizierung, Glossar-10
  - Rampengeber, Glossar-10
  - Regel-Algorithmus, 2-12
  - Regelaufgabe, Spezifizierung, 3-1
  - Regelbarkeit, 2-2
  - Regeldifferenz, Glossar-10
    - Grenzwertüberwachung, 4-37
    - Signalfilterung (Totzone), 4-35
  - Regeldifferenz-Überwachung, Glossar-10
    - Funktionsweise, 4-38
    - Hysterese, 4-37
- Regelgröße, Glossar-11
    - Anwender-FC verschalten, 4-28
    - eigene Funktion (FC) einfügen, 4-28
    - Glättungsfunktion, 4-24
    - Grenzwertüberwachung, 4-30
    - Quadratwurzelbildung, 4-26
    - Steigungsüberwachung, 4-32
    - Überwachung der Änderungsgeschwindigkeit, 4-32
    - Verstellung über Konfigurationswerkzeug, 4-34
  - Regelkreis, Glossar-11
  - Regelkreis-Editor, 3-14
  - Regelstrecke, Ersatzzeitkonstante, 3-15
  - Regelstrecke (Prozess), Glossar-11
  - Regelung erstellen (Prinzip), 1-2
  - Regelung konfigurieren, 3-7
  - Regelung von Mischprozessen, 2-9
  - Regler, Glossar-12
  - Regler-FB, Codeumfang, 1-6
  - Regler-Konfiguration, Vorgehensweise (Checkliste), 3-7
  - Regleraufrufverhalten, 8-2
  - Regleraufrufverteiler, 1-1, 1-6, 3-17, 7-1, Glossar-12
    - globaler Datenbaustein, 7-2
  - Reglerauslegung, 2-1
  - Reglerauswahl, 3-5
  - Reglerkonfiguration, Vorkenntnisse, 3-2
  - Regleroptimierung, 10-2
  - Reglerparameter (Kenngrößen), Glossar-12
  - Reglerstruktur festlegen, 4-40
  - RMP\_SOAK, 4-4
  - ROCALARM, 4-32
    - Parameter, 4-33
- S**
- Schrittregler, Glossar-12
    - Betriebsarten-Schaltung, 6-7
    - Betriebsarten-Umschaltung, 6-6
    - Funktionsschema, 6-2
    - Kaskadenschaltungen, 6-25
      - mit Stellungsrückmeldung, 2-21
      - Neustart/Wiederanlauf, 6-4
      - ohne Stellungsrückmeldung, 2-22, 6-3
    - Regelfunktionen, 6-1
    - Struktur, 6-5
  - Schrittregler (Example1), 7-10
    - Baueinstruktur, 7-11

- Schrittregler ohne Stellungsrückmeldung
  - Betriebsarten, 6-19
  - Erzeugung der Stellsignale, 6-20
  - Parameter der Stellgrößenverarbeitung, 6-24
  - Signalverarbeitung der Stellgröße, 6-18
  - Struktur und Funktion, 6-18
- Signalanpassung, 3-18
- Signalfussplan, Glossar-13
- Signalfusspläne, 2-15
  - Zeichen und Symbole, 2-15
- Signalverarbeitung
  - analoge Stellgröße, 2-20
  - binäre Stellsignale, 6-14
  - Führungsgröße, 2-16
  - im Istwertzweig, 4-22
  - im Sollwertzweig, 4-1
  - Kontinuierlicher Regler, 4-46
  - PID-Regler, 2-19
  - Regeldifferenz, 4-35
  - Regelgröße, 2-17
  - Stellgröße des Schrittreglers, 6-5
  - Stellgröße des kontinuierlichen Reglers, 5-3
  - Stellungsrückmeldung, 6-11
  - Stellwert des Schrittreglers, 2-21
- Signalwandlung, internes Format → Peripherieformat, 5-15
- Simulation der Stellungsrückmeldung, 6-23
- Softwareumgebung, 1-5
- Sollwert, Glossar-13
  - anwenderspezifische Funktion (FC), 4-15
  - Begrenzung absolut, 4-19
  - Begrenzung der Änderungsgeschwindigkeit, 4-17
  - Umschalten auf Konfigurationswerkzeug, 4-21
- Sollwert-Generator, 4-1
  - Anlauf und Betriebsweise, 4-2
  - Parameter, 4-3
  - Verstellbereich, 4-1
  - Verstellgeschwindigkeit, 4-1
- Sollwertbegrenzung, Funktionsweise, 4-20
- Sollwertgenerator, Glossar-13
- Sollwertnormierung, 4-12
- SP\_GEN, 4-1
- SP\_LIMIT, 4-19
  - Parameter, 4-20
- SP\_NORM, 4-12
  - Parameter, 4-13
- SP\_ROC, 4-17
  - Parameter, 4-18
- SPFC (Anwender-FC), 4-15
  - Parameter, 4-16
- Standard PID Control, 1-1
  - Arbeitsweise, 2-11
  - Blockschaltbilder, 8-3
  - Einführung, iv
  - Funktionsschema, 1-2
  - Konzept, 1-1
  - Softwarepakete, 1-3
  - Struktur, 2-11
  - Übersichtsbilder, 2-12
- Standard-Funktionsbaustein, Regler-FB, 1-1
- Standard-Regelung, Glossar-13
  - Aufrufverhalten, 3-16
  - Beispiele, 1-4
  - Einsatzmöglichkeiten und -grenzen, 1-6
  - Einsatzumgebung, 1-5
  - Funktionalität, 1-7
  - Funktionsübersicht, 1-2
  - Grundfunktionen, 1-2
  - Produktstruktur, 1-3
  - ständig aktive Funktionen, 3-6
- Stellausgänge, 3-5
- Stellgerät, Glossar-13
- Stellglied, 3-4
  - Anschlagsignale, 6-18
  - Art des Stellsignals, 3-4
- Stellgröße, Glossar-13
  - Begrenzung absolut, 5-11
  - Begrenzung der Änderungsgeschwindigkeit, 5-9
  - Begrenzung des Einstellbereichs, 5-11, 6-9
  - eigene Funktion (FC) einfügen, 5-7
  - Rampenfunktion, 5-9
  - Signalformen, 3-4
  - Steigungsbegrenzung, 5-9
  - Verstellung über Konfigurationswerkzeug, 5-16
- Stellgröße (Schrittregler), Verstellung über Konfigurationswerkzeug, 6-8
- Stellgrößenbegrenzung, Meldeausgänge, 5-11, 6-9
- Stellgrößenbildung, Impulsausgabe, 5-19
- Stellsignal, Reglerauswahl, 3-5
- Stellsignalbildung
  - Betriebsarten des Schrittreglers, 6-5
  - Betriebsarten des stetigen Reglers, 5-3, 5-5
- Stellungsrückmeldung, 2-21
  - Signalanpassung (Normierung), 6-11
  - Simulation, 2-22, 6-23

- Stellwert
    - anwenderspezifische Funktion (FC), 5-7
    - Umschalten auf Konfigurationswerkzeug, 5-16
  - Stellwert (Schrittregler), Umschalten auf Konfigurationswerkzeug, 6-8
  - Stellwertbegrenzung, Funktionsweise, 5-12, 6-10
  - Stellwertnormierung, 5-13
  - Störgröße, Glossar-13
  - Störgrößenaufschaltung, 2-7, 4-39, Glossar-14
    - Wirkungsweise, 2-7
  - Störgrößenerfassung, 2-7
  - Strecke mit Ausgleich (PT-Strecke), 2-2
  - Strecke mit Totzeit, 2-3
  - Strecke ohne Ausgleich, 2-4
  - Strecken-Eigenschaften, 2-1
  - Streckenanalyse, 2-1
  - Streckeneigenschaften, 3-2
  - Streckenidentifikation, 2-5
    - Methodik, 2-5
    - Regelkreis offline/online, 2-6
    - Reglerkenndaten, 2-5
    - Übergangsfunktion, 2-5
  - Streckensimulation (APP\_Pulsegen), 7-31
  - Streckensimulation (Example1), 7-11
  - Streckensimulation (Example2), 7-17
  - Struktur-Beispiele, 1-4
  - Stützpunkte, 4-5
  - Systemrahmen, 1-6
- T**
- Teilfunktionen, 1-2
    - Schaltbilder, 2-15
  - THREE\_ST, 6-15, 6-21
  - TM\_LAG, 4-43, 4-51
  - Toleranzbänder, 4-30, 4-37
  - Tool
    - integrierte Hilfe, 10-2
    - Softwarevoraussetzungen, 10-1
- Totzeit, 2-3, Glossar-14
  - Totzonenfunktion, Wirkungsweise, 4-35
  - Totzonenglied, 4-35
  - Trapezregel, Glossar-14
- V**
- Verhältnisregelung, 2-8, Glossar-15
    - zwei Regelkreise, 2-8
  - Verhältnisregelung (Example3), 7-21
    - Baueinstruktur, 7-22
  - Verzögerung 1. Ordnung, Glossar-15
  - Verzögerung der D-Wirkung (TM\_LAG), 4-43
  - Verzögerungsglied, 4-24
  - Verzögerungszeitkonstante (TM\_LAG), 4-51
  - Verzugszeit, 2-2
  - Vorhaltzeit, Glossar-16
- W**
- Weckalarm-OB, 3-16
  - Wertebereich, Glossar-16
    - techn. Bereich, 3-18
    - Zeiten, 3-18
  - Wiederanlauf, 3-16
- Z**
- Zahlendarstellung, 3-18
  - Zeitplanerstellung, Stützpunktparameter, 4-5
  - Zeitplangeber, 4-3, Glossar-16
    - anhalten, 4-8
    - Ausgang vorbelegen, 4-7
    - Betriebsarten, 4-5, 4-6
    - einschalten, 4-6
    - Online-Änderungen, 4-10
    - Parameter, 4-10
    - zyklischer Betrieb, 4-8
  - Zweipunktregelung, 5-25
  - Zweipunktregler, Glossar-16