

VDE

VERLAG

*Serge Zacher*

# SPS-Programmierung mit Funktionsbausteinsprache



Automatisierungstechnische  
Anwendungen

## Vorwort

Nach dem heutigen Stand der Technik lassen sich speicherprogrammierbare Steuerungen (SPS) überall für die Automatisierung einsetzen. Mit SPS werden sowohl einzelne Anlagen binär gesteuert, als auch gesamte Industrieprozesse regelungstechnisch erfasst. Die Vernetzung führt zu immer größeren Systemen – mit entsprechend hohem Aufwand. Dem Aufbau solcher SPS-Systeme kommt eine enorme Bedeutung zu, weil in erster Linie die Sicherheit auf dem Spiel steht.

Auf die Darstellung der SPS als Regler wird im Buch besonderer Wert gelegt. Der Schwerpunkt liegt bei der Behandlung der grafischen Funktionsbausteinsprache (Kapitel 2), einer der fünf Programmiersprachen der internationalen Norm *IEC 61131-3*, die gleichzeitig als europäische und deutsche Norm *DIN EN 61131* gilt. Als Basis für die Implementierung der Funktionsbausteinsprache (FBS), die auch unter ihrer genormten Bezeichnung *FBD (Function Block Diagram)* bekannt ist, dient das Programm *Concept* (Kapitel 3).

Das Kapitel 4 befasst sich mit den auf SPS-Basis implementierten PID- und Fuzzy-Reglern. Ein Projektbeispiel der Prozessautomatisierung mit dazugehörigen Aufgaben der Identifikation, automatischen Auswertung der Sprungantworten, Messungen, Regelungen, Steuerungen und wird im Kapitel 5 aufgeführt. Die verwendeten regelungstechnischen Begriffen und Formelzeichen orientieren sich an der geltenden *DIN 19226*.

Mit dem Programm *Lookout* lassen sich die im Buch entwickelten SPS-Funktionsbausteine auf dem PC schnell und übersichtlich visualisieren und bedienen. Im Kapitel 5 wird auf dem Einstieg in das Programm *Lookout* eingegangen und dieser durch einfache Übungen erleichtert.

Während die klassischen SPS-Programme sich mit vorprogrammierten Steuerungen befassen, bieten die im Buch beschriebenen *selbstlernenden Steuerungen* (Kapitel 6 und 7) ein hohes Maß an Flexibilität:

- Der *Adress-Master* ordnet die SPS-Adressen zu den physikalischen Ein-/Ausgängen, erkennt mögliche Fehler und korrigiert die Zuordnung während des Betriebs.
- Die *selbsteinstellende Regelung* erkennt die Güte des Regelkreisverhaltens und berechnet automatisch die Kennwerte eines PID-Reglers.
- Der Fuzzy-Regler mit *Feed-Forward-Neuronen* stellt seine Kennwerte automatisch nach dem konventionellen Betragsoptimum ein.

Das Buch richtet sich an Fachleute auf den Gebieten MSR- und Leittechnik, an Studenten der Elektrotechnik und des Maschinenbaus an den Fachhochschulen sowie an Ingenieure der Automatisierungstechnik. Es bietet vor allem Kenntnisse der grafischen Programmiersprache für die Prozessautomatisierung mit SPS und soll einen Überblick über die neuen Verfahren der *Fuzzy-* und *Neuroregelung* geben. Dafür ist im Kapitel 8 eine Bibliothek der entworfenen Funktionsbausteine zusammengestellt.

Das Buch ist für das Selbststudium konzipiert. Für jedes Kapitel findet man Übungsaufgaben mit Lösungen, die für die Programmierung mit *Concept* und *Lookout* entwickelt sind.

Alle im Buch beschriebenen Funktionsbausteine sind dem Leser unter <http://www.it-4-u.de> zur Verfügung gestellt. Über diese Adresse kann man das Programm *Concept* direkt von den Web-Seiten der Hersteller-Firma Modicon, USA, oder Schneider Electric GmbH herunterladen sowie die Hinweise über das Programm *Lookout* und dessen Vertreiber, Firma National Instruments Germany GmbH, finden.

Die Installation dieser Programme auf PC soll das Gesamtverständnis der Methoden fördern und die Erprobung des gelernten Stoffes auch ohne SPS ermöglichen. Unter <http://www.it-4-u.de> kann man sich bei tiefergehenden Fragen zum Buch direkt an den Autor über ein virtuelles Seminar per *Internet - Diskussionsforum* wenden.

Dieses Buch entstand während der Lehrtätigkeit des Autors im Fachgebiet Automatisierungstechnik an der Fachhochschule Wiesbaden. Alle Programme wurden mehrmals getestet, unter anderem auf der CeBit'97 und der Hannover Messe'99. Jedoch ist das Programmmaterial mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor und der Verlag übernehmen infolgedessen keine Haftung für die Benutzung dieses Programmmaterials.

Der Autor ist für fachliche Anregungen und Kritik der Leserinnen und Leser sehr dankbar.

Wiesbaden, im August 2000

Prof. Dr. S. Zacher

# 1 Grundlagen der Automatisierung mit SPS und PC

In der Anfangszeit der Automatisierung technischer Prozesse wurden verbindungsprogrammierte Steuerungen (VPS) eingesetzt. Das Anpassen der rein elektromechanischen Elemente solcher Steuerungen geschah durch die Verdrahtung. Die Änderungen der Steuerung ohne Eingriff in die Verdrahtung waren nicht möglich.

Hingegen erlauben die speicherprogrammierbare Steuerungen (SPS), die aus einem Automatisierungsgerät (ein Mikroprozessor mit allen benötigten Elementen) und einem Programmiergerät (meist ein PC) bestehen, ein Programm festzulegen, das für verschiedenen Aufgaben genützt und beliebig oft kopiert werden kann. Die Verdrahtung der Eingangs- und Ausgangsseite des Automatisierungsgeräts kann unverändert bleiben. Da die SPS von Bedingungen und Ergebnissen jedes Programmschrittes abhängig ist, wird diese Steuerungsart als Ablaufsteuerung bezeichnet. Die Themen SPS und Grundlagen der Steuerungstechnik werden in etlichen Büchern wie [1, 2, 4, 5, 17] behandelt.

Moderne Steuerungssysteme verarbeiten nicht nur binäre Signale, wie es bei den klassischen SPS der Fall war, sondern auch Bytes, Worte und Doppelworte und können miteinander durch Bussysteme gekoppelt werden. Ein PC wird heute nicht nur als ein separates Programmiergerät (PG), das über eine serielle Schnittstelle mit dem Automatisierungsgerät verbunden ist, sondern als eine Anzeige- und Bedienkomponente (ABK) für Simulation und Visualisierung des Prozessablaufs benutzt. Mit der Entwicklung von Soft-SPS, PC-basierten- oder Coprozessor-Lösungen [3, 13] verwischt sich die Grenze zwischen SPS und PC.

## 1.1 SPS als Steuerungsgerät

### 1.1.1 Bauformen der SPS

Eine SPS besteht aus folgenden vier Baugruppen:

- Mikroprozessor mit seinen Speichern (Zentraleinheit, CPU)
- Ein-/ Ausgabeeinheiten (Wandler, Filter, galvanische Trennung usw.)
- Daten- und Adressbus
- Netzversorgung.

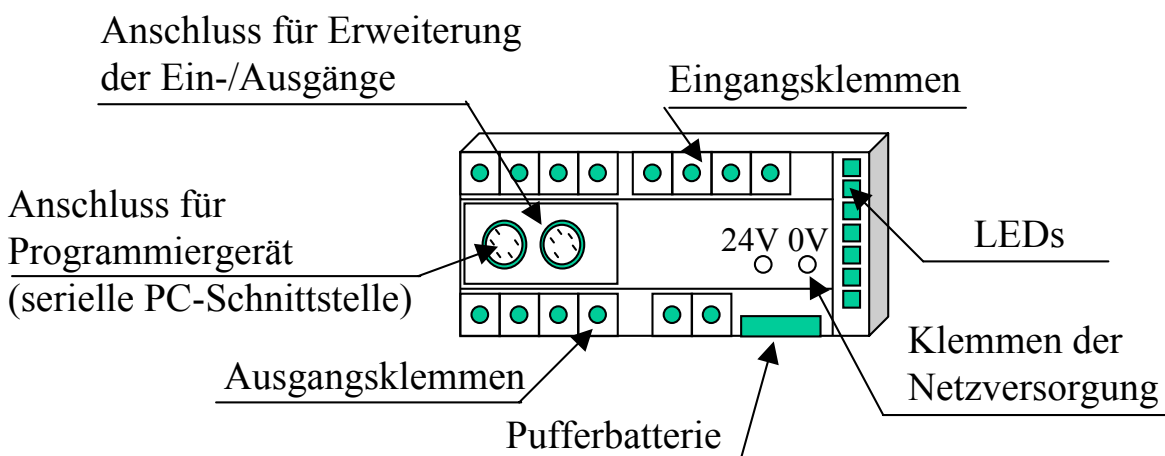
Die einfachsten SPS mit nur wenigen Ein-/Ausgängen werden als Steckkarten in einen Rechner (z.B. PC-Bus) eingesteckt. Für kleine Steuerungsaufgaben werden alle vier Baugruppen aus Kostengründen in einem gemeinsamen Gehäuse in Form

einer Kompakt-SPS untergebracht. **Bild 1.1** zeigt die Ansicht einer Kompakt-SPS mit Ein-/Ausgangsklemmen und mit Anschlüssen zu PC und zu lokalen Erweiterungen LE. Die Zustände der binären Ein-/Ausgänge werden auf der Frontplatte durch Leuchtdioden angezeigt.

Eine Kompakt-SPS kann auch binäre Eingänge für Alarm- oder Interrupt-Funktionen vorsehen, die das Hauptprogramm unterbrechen und die entsprechenden Unterprogramme aufrufen. Durch Kopplung mit externen Modulen können die Ein- und Ausgänge beim zentralen Aufbau erweitert werden. Beispielsweise bietet die Kompakt-SPS PS 4-201-MM der Firma Moeller acht digitale Eingänge und sechs digitale Ausgänge, zwei analoge Eingänge und einen analogen Ausgang sowie zwei Alarmeingänge und einen Zähler-Eingang. Außerdem kann das Grundgerät durch sechs lokale Erweiterungen LE 4 oder externe Module EM 4-2xx auf 110 Ein-/Ausgänge erweitert werden.

Für umfangreiche Steuerungsaufgaben werden modular ausbaufähige Systeme angeboten, die aus mehreren steckbaren Baugruppen bestehen. Die geschirmten Module werden in einem Baugruppenträger zusammengefasst, z.B. bei PS 416 von Moeller, SIMATIC S7 von Siemens und TSX Quantum von Schneider Electric.

Durch den modularen Aufbau im gleichen Format, wie im *Europaformat*, mit mehrpoligen Steckkontakten können die Baugruppen mit dem Bussystem verbunden und kombiniert werden. Beispielsweise lassen sich bis zu 19 modulare SPS PS 416 im Basis- oder Erweiterungs-Baugruppenträger zusammenfassen und auf sieben Netzwerkstränge mit je 30 Teilnehmern erweitern.



**Bild 1.1** Schematische Darstellung der Frontplatte einer Kompakt-SPS

## 1.2 SPS als Regler

### 1.2.1 Vergleich: Regelung und Steuerung

Die *Regelungs-* und die *Steuerungstechnik* haben das gemeinsame Ziel, eine Anlage zu automatisieren, jedoch wird diese von jeder Disziplin aus einem anderen Blickwinkel betrachtet (**Tabelle 1.1**).

Die Regelgröße (Istwert) wird bei einer Regelung fortlaufend erfasst und mit der Führungsgröße (Sollwert) verglichen. Dagegen wirkt die vorprogrammierte Steuerung so auf die Strecke ein, dass diese im gewünschten Sinn beeinflusst wird.

<b>Regelung</b>	<b>Steuerung</b>
Die Regelung erfolgt in einem geschlossenen Regelkreis	Die Steuerung geschieht in einem offenem Wirkungsweg
Ein Regelkreis kann instabil werden	Keine Schwingungskreise vorliegen
Die Beschreibung von Störgrößen ist nicht nötig: Die Auswirkung vieler Störgrößen kann kompensiert werden.	Nur mathematisch beschreibbare Störgrößen werden kompensiert. Es dürfen keine Änderungen der Kennlinien auftreten.
Der Aufwand von benötigten Stell- und Messgliedern ist hoch	Der Aufwand von benötigten Messungen ist gering
Typischerweise analoge Signale	Typischerweise binäre Signale
Theoretische Grundlagen: Differentialgleichungen oder Frequenzkennlinien	Theoretische Grundlagen: Logik (Boolesche Algebra)

**Tabelle 1.1** Unterschiede zwischen konventioneller Regelung und Steuerung

Heutzutage geht der Trend für beide Automatisierungsarten in Richtung digitaler Gerätetechnik. Damit wird die aus der Steuerungstechnik stammende SPS sehr breit für Regelungszwecke verwendet.

Nachfolgend ist die aktuelle Situation im SPS-Bereich beschrieben:

- Bei SPS-Systemen kommen meistens die Standard-PID-Regelalgorithmen zum Einsatz. Ein Beispiel ist die Bausteinbibliothek `CONT_CTL` des Programms `Concept V2.11 XL` mit Funktionsbausteinen von Standard-, Kaskaden-, Zweipunkt- und Dreipunktreglern sowie Autotuning. Die Methoden wie adaptive oder prädiktive Regelung sind in SPS nur selten zu finden.

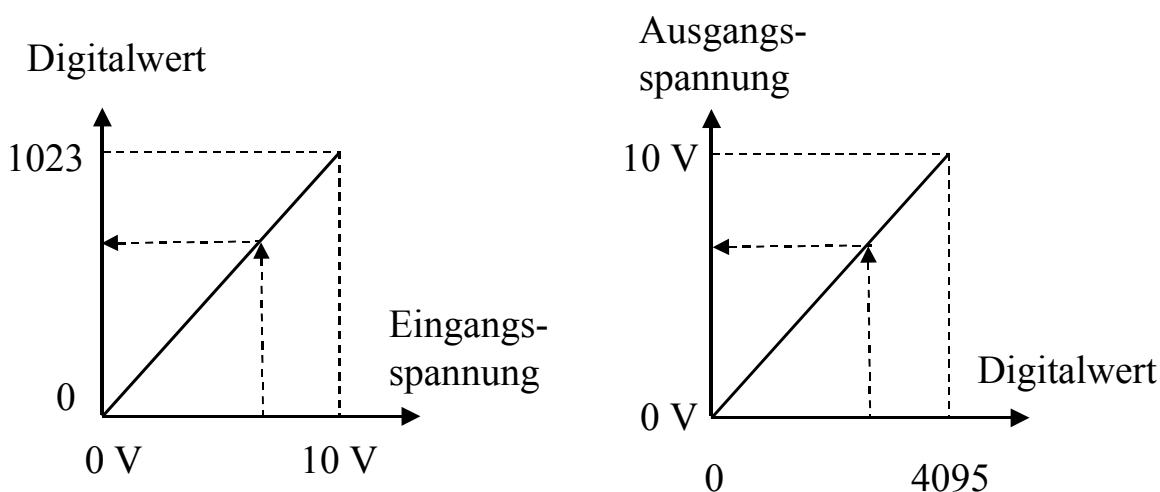
- Die Fuzzy-Regelung wird von vielen Herstellern angeboten. Die Bausteine sind preiswert und leicht zu testen. Die Einarbeitung dauert nicht lange. Trotz dieser Vorteile findet die Fuzzy-Regelung nur begrenzte industrielle Anwendung.
- Die künstlichen neuronalen Netze verlangen von der SPS umfangreiche Berechnungen, was zu Problemen führen kann. Für neuronale Algorithmen werden die übergeordneten Prozessoren von ABK wie WinCC benutzt. Die in diesem Buch entwickelten Neuro-Algorithmen lassen sich auch in CPU der SPS ausführen.

### 1.2.2 Analogwertverarbeitung

Ein Analog-Digital-Umsetzer (A/D) wandelt eine elektrische Messgröße am Eingang der SPS in einen Digitalwert um. Die Umsetzung geschieht mit einer begrenzten Auflösung, denn nicht jedem Analogwert kann ein Digitalwert zugeordnet werden.

Die Digitalwerte werden am Ausgang von SPS binär ausgegeben oder durch Digital-Analog-Umsetzer (D/A) in Strom- oder Spannung umgesetzt.

*Beispiel:* Die A/D-Auflösung der PS 4 (Moeller GmbH, [5]) ist 10 Bit (**Bild 1.5**). Die D/A-Auflösung beträgt 12 Bit. Die Eingangsspannung von 10 V wird in einen Digitalwert von 1023 umgesetzt. Dagegen entspricht die Ausgangsspannung von 10 V einem Digitalwert von 4095. Bei einer Temperaturmessung im Bereich von 20 °C bis 50 °C wird beispielsweise die Temperatur von 35 °C durch den digitalen Eingangswert von 512 abgebildet.



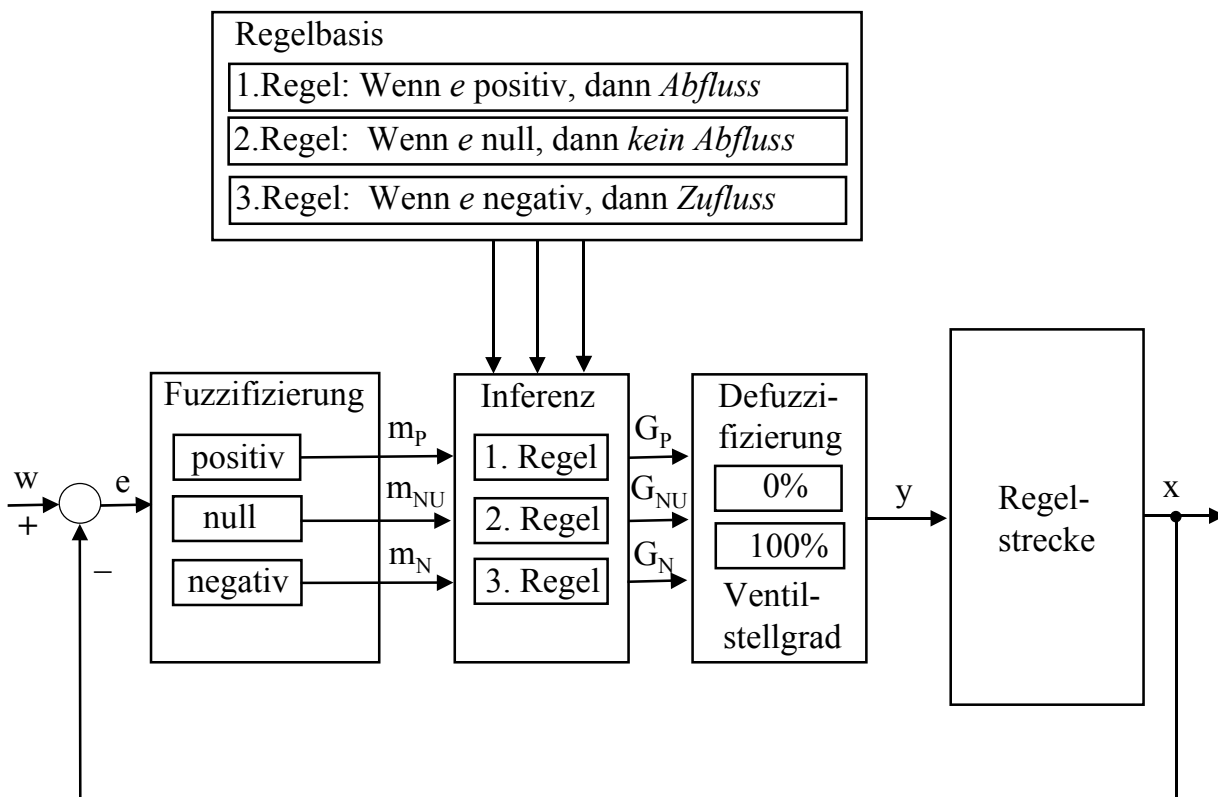
**Bild 1.5** Analogwertverarbeitung mit Auflösung von 10 Bit am Eingang (links) und 12 Bit am Ausgang (rechts)

## 1.2.4 Fuzzy-Regelung

Die Regelung mit Fuzzy-Logik basiert auf praktischen Erfahrungen und kommt ohne mathematische Formeln aus.

Die Struktur eines Fuzzy-Reglers ist im **Bild 1.8** dargestellt. Der aktuelle *scharfe* Wert  $e_{\text{akt}}$  der Regeldifferenz  $e(t)$  wird am Eingang des Fuzzy-Reglers zunächst fuzzifiziert (*Fuzzifizierung*). Darunter versteht man eine Umsetzung physikalischer Größen in Terme linguistischer Variablen. Damit werden die Zugehörigkeitsgrade  $m_P$ ,  $m_{\text{NU}}$  und  $m_N$  von linguistischen Variablen *positiv*, *null* und *negativ* gebildet. Der Fuzzy-Regler wertet alle Regeln der *Regelbasis* für jeden linguistischen Wert der Regeldifferenz aus und bildet mit Hilfe der logischen Operationen den Erfüllungsgrad  $G$  jeder Regel (*Inferenz*). Danach ermittelt der Regler die Fuzzy-Menge der Stellgrößen für jeder Regel und bestimmt daraus als letzten Schritt einen resultierenden *scharfen* Wert der Stellgröße  $y_{\text{akt}}$  (*Defuzzifizierung*), mit dem eine Stelleinrichtung angesteuert werden kann.

Die Fuzzy-Blöcke des Reglers werden von SPS-Herstellern in Form von Funktionsbausteinen angeboten. Dem Benutzer wird lediglich die Wahl der Stützpunkten und die Parametrierung von Bausteinen überlassen.



**Bild 1.8** Beispielhaftes Wirkungsplan eines Regelkreises mit Fuzzy-Regler



### 1.3.3 IEC-Norm und DIN EN 61131-3

Um eine weitgehend einheitliche und herstellerunabhängige Programmierung von Steuerungsaufgaben zu ermöglichen, wurde die Norm IEC 1131 *Programmable Controller* von der *International Electrotechnical Commission* (1993) festgelegt. Diese Norm wurde als europäische Norm EN 61131 anerkannt und hat seit 1994 gleichzeitig den Status einer deutschen Norm DIN EN 61131 [15].

Der Teil 3 der Norm DIN EN 61131-3 definiert folgende Programmiersprachen:

- Ablaufsprache AS oder SFC (Sequential Function Chart)
- Anweisungsliste AWL oder IL (Instruction List)
- Funktionsbausteinsprache FBS (engl. Function Block Diagram FBD)
- Kontaktplan KOP oder LD (Ladder Diagram)
- Strukturierter Text ST (Structured Text)

Die Sprachelemente der verschiedenen Sprachen sind nach der Norm nur an einer Stelle definiert. So sind z.B. die Funktionen aus AS und FBS zusammengefasst und können durch Funktionen aus ST und KOP ergänzt werden. Man kann die Sprache auswählen, die zu einer einfachen Darstellung der Steuerungsaufgabe führt und eine optimale Lösung unterstützt, z.B. für Ablaufsteuerungen die Sprachen AS und FBS, dagegen KOP und FBS zur Darstellung komplexer Funktionen in einem Kontaktplan.

Durch die Vereinheitlichung von Programmiersprachen ist es möglich, die SPS verschiedener Hersteller mit gleichen Sprachelementen zu programmieren. Programme, die in einer Programmiersprache entwickelt wurden, sollen nach IEC-Norm auch in den anderen vier IEC-Programmiersprachen darstellbar sein.

Ein normiertes Softwaresystem soll das Wiederherstellen von Programmen erleichtern. Dadurch lassen sich die Kosten für die Entwicklung von Anwenderprogrammen verringern sowie Einarbeitungszeiten und Schulungskosten reduzieren. Wegen bestimmter Einschränkungen bieten jedoch derzeit noch nicht alle SPS-Hersteller die Übersetzungsmöglichkeit in vollem Umfang an.

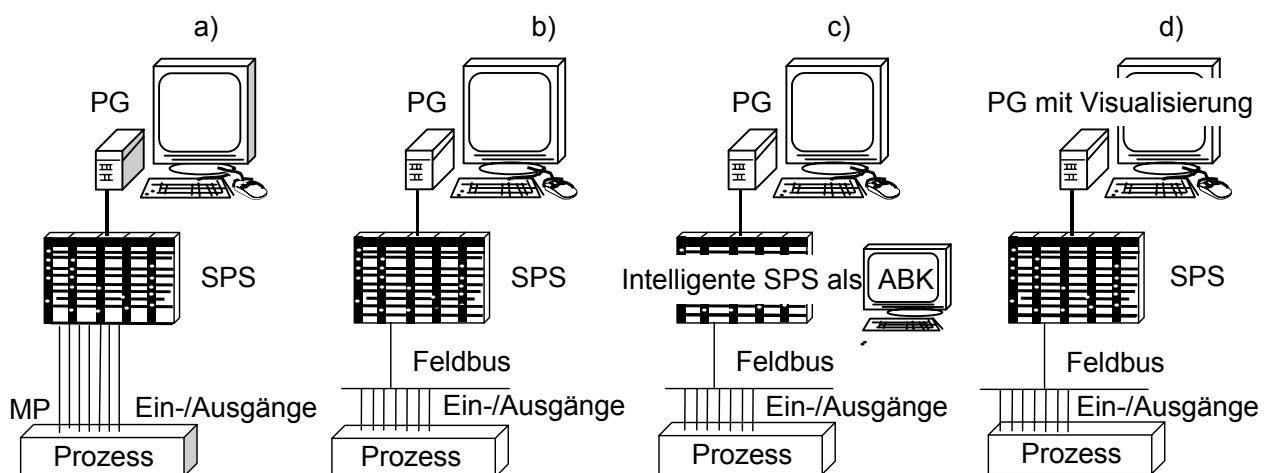
Ein wesentlicher Vorteil der modernen Steuerungstechnik ist die Möglichkeit einer PC-Simulation des Anwenderprogramms, in dem das Programm – bevor es in der SPS übergeben wird – in einen PC-Simulator geladen wird. Mit Hilfe der Animation werden aktuelle Werte von Variablen, Zuständen und Verbindungen eingeblendet oder farbig angezeigt.

## 1.4 Automatisierungsstrukturen

### 1.4.1 Verbindung SPS-PC

Für die Verbindung einer oder mehrerer SPS mit dem PC bieten sich folgende vier Möglichkeiten an [3, 6, 11, 13]:

- PC als Programmiergerät (PG) für Einzel-SPS. Dabei kann die SPS mit einer klassischen Mehrpunktverbindung (MP) prozessnah eingesetzt (**Bild 1.14a**) oder mit Hilfe eines Feldbusses und einer Bus-Master-Karte weit vom Prozess betrieben werden (**Bild 1.14b**). Im PC ist typischerweise eine Offline-Simulation oder Animation zum Testen des Anwenderprogramms vorhanden.
- PC als PG für Einzel-SPS, die einen eigenen Prozessor als Anzeige-/Bedien-Komponente (ABK) besitzt (**Bild 1.14c**). Solche SPS-PC-Einheiten, die autark für die Aufgaben wie Steuerung, Regelung, Kommunikation, Visualisierung eingesetzt werden können, nennt man intelligente Experte. Diese zeichnen sich durch kleine Zykluszeiten aus.
- PC als PG und ABK für Einzel-SPS mit einer Feldbusverbindung (**Bild 1.14d**). Neben Offline-Simulation übernimmt der PC die Funktionen der Online-Steuerung und der Prozessvisualisierung.
- PC als PG und ABK über ein Bussystem für ein hierarchisches SPS-System. Die Systemkomponenten kommunizieren typischerweise sowohl untereinander über einen seriellen Bus als auch mit über- oder untergeordneten Ebenen über einen Systembus. Einem Prozessleitsystem ähnlich entsteht ein SPS-System mit SPS als PNK (prozessnahe Komponenten) in der Feldebene und PC als ABK in der Prozessleitebene.



**Bild 1.14** PC als Programmiergerät (PG) für Einzel-SPS

## 4.2 Standard-SPS-Regler

### 4.2.1 PID-Regler

Ein PID-Regler als Funktionsbaustein PIDP1 der Bibliothek CONT\_CTL, Gruppe CLC, wurde im Abschnitt 1.2.3 als Standard-Regleralgorithmus beschrieben. Nachfolgend wird der EFB PIDP1 parametrisiert und getestet.

Is Beispiel soll ein PI-Regler mit folgenden Kennwerten programmiert werden:

$$K_{PR} = 3,5$$

$$T_n = 5 \text{ s}$$

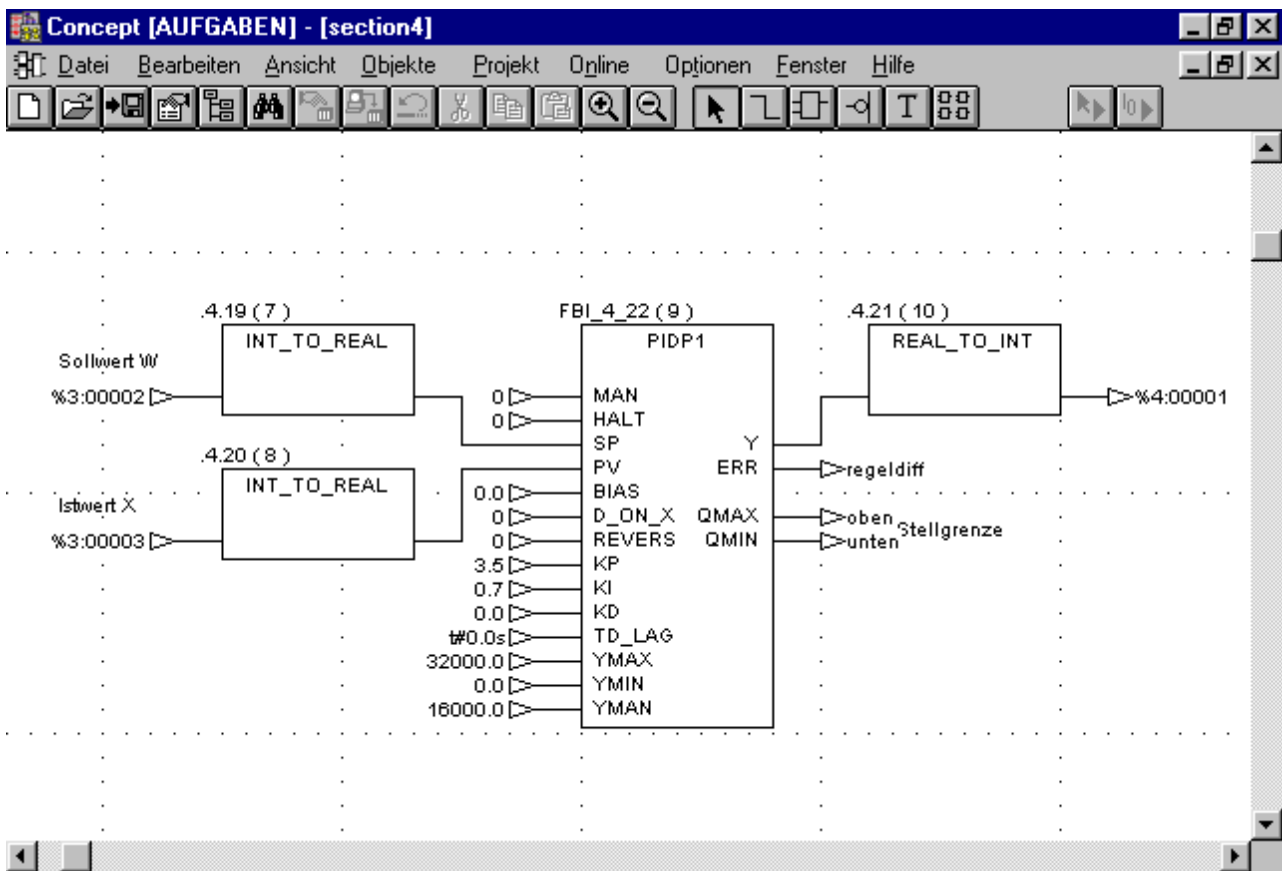
$$T_V = 0 \text{ s}$$

Um den EFB PIDP1 zu parametrieren, werden die Kennwerte umgerechnet und als REAL-Literale deklariert:

$$KP = K_{PR} = 3.5$$

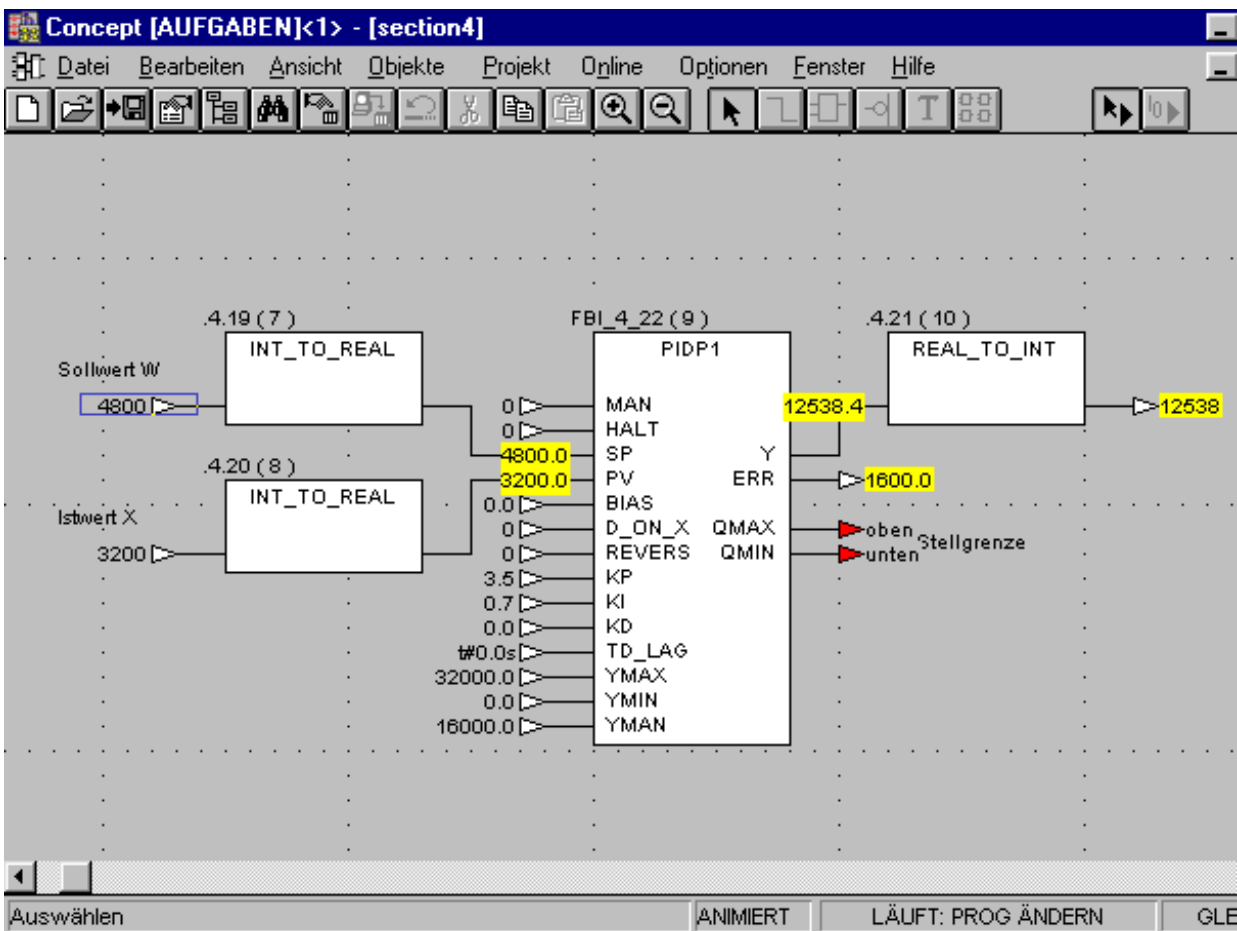
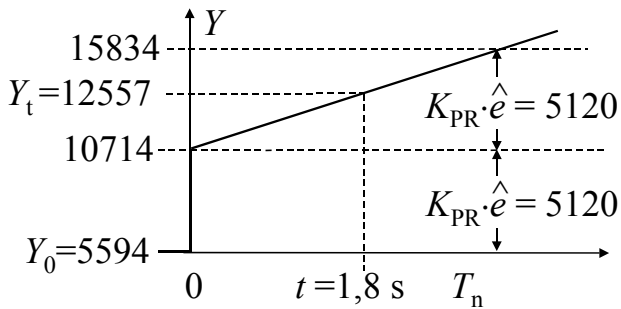
$$KI = K_{PR} / T_n = 0,7 \text{ s}^{-1}$$

$$KD = 0,0 \text{ s}$$



**Bild 4.1** Programmierung eines PID-Reglers mit EFB PIDP1

Das Programm wird mit dem PC-Simulator ohne angeschlossene SPS simuliert. Zum Beginn der Simulation werden die beiden direkten Adressen mit dezimalen Werten, z.B. mit 3200,0 belegt, was dem Arbeitspunkt von  $W_0 = X_0 = 1 \text{ V}$  entspricht. Zum Zeitpunkt  $t = 0$  wird der Sollwert (Adresse %3:00002) um 0,5 V, d.h. um  $0,5 \cdot 3200,0$  geändert und auf den Wert von  $3200,0 + 1600,0 = 4800,0$  gesetzt. Aus dem Arbeitspunkt  $Y_0 = 5594$  soll die Stellgröße theoretisch bei  $t = 1,8 \text{ s}$  einen Wert von  $Y_t = 12557$  erreichen (**Bild 4.2**). Dies wird durch die Simulation bestätigt, allerdings mit einem Simulationsfehler von  $12557 - 12538 = 19$ .



**Bild 4.2** Die theoretisch zu erwartende Sprungantwort (oben) und das Simulationsergebnis (unten) zum Zeitpunkt  $t = 1,8 \text{ s}$  nach dem Eingangssprung von 1600,0.

## 4.2.2 Adaptive Regler

Die Umrechnung von Kennwerten des Reglers  $KI = K_{PR} / T_n$  und  $KD = K_{PR} \cdot T_V$  kann von EFB MUL\_REAL und DIV\_REAL (Bibliothek IEC, Gruppe *Arithmetic*) übernommen werden.

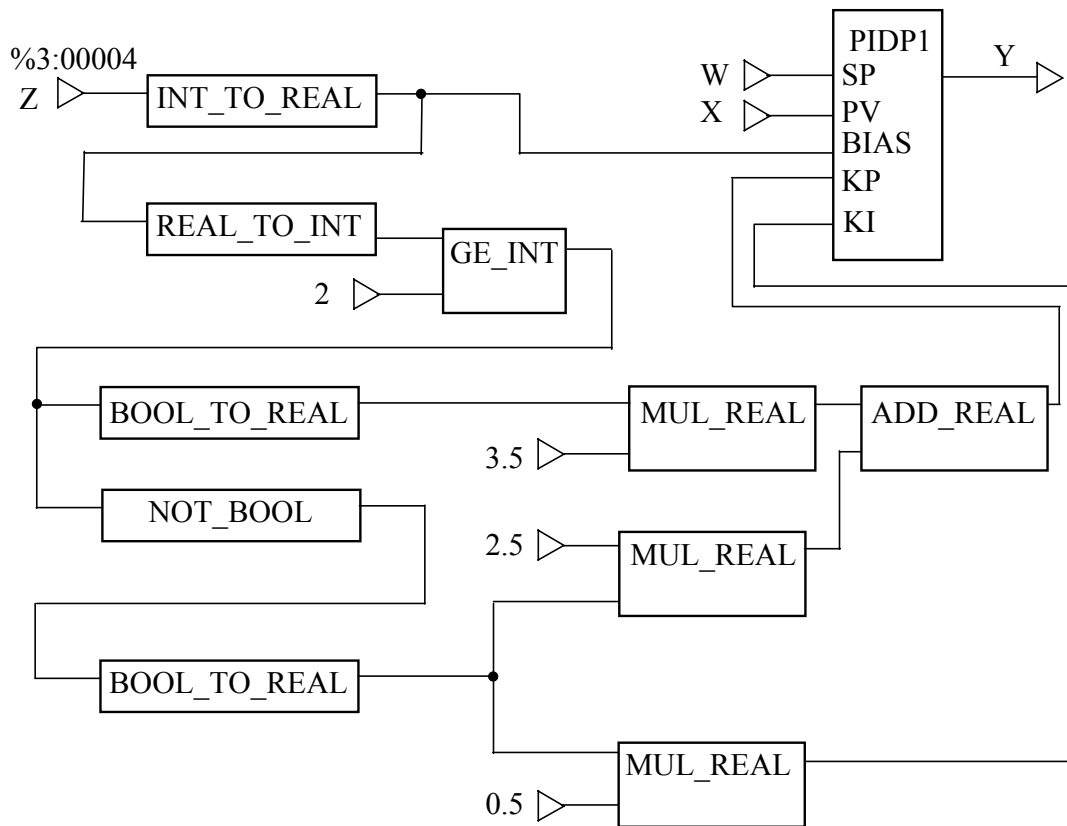
Diese Idee – die Kennwerte eines SPS-Reglers online zu ändern – lässt sich weiter entwickeln, indem die Kennwerte des Reglers ständig der veränderlichen Regeldifferenz oder Störgröße angepasst werden.

Als Beispiel wird ein adaptiver Regler mit dem EFB PIDP1 programmiert. Es wird angenommen, dass im Regelkreis eine Störgröße  $Z$  wirkt. Die Störgröße kann erfasst und der Eingangsadresse %3:00004 des SPS-Reglers gegeben werden.

Der Regler soll nach folgendem Algorithmus programmiert werden:

- Die Störgröße  $Z > 2$  soll mit dem P-Regler mit  $K_{PR} = 3,5$  ausgeregelt werden.
- Ist die Störgröße  $Z \leq 2$ , so wird statt P-Regler ein PI-Regler mit  $K_{PR} = 2,5$  und  $T_n = 5$  s eingesetzt, d.h. mit  $K_I = 2,5 / 5\text{s} = 0,5 \text{ s}^{-1}$ .

Das Programm des adaptiven SPS-Reglers ist im **Bild 4.3** schematisch dargestellt. Den EFB GE\_INT findet man in der IEC-Bibliothek unter *Comparison*.



**Bild 4.3** Adaptiver SPS-Regler mit Störgrößenkompensation

### 4.2.3 Zweipunktregler

Zweipunktreglern enthalten als Kennwerte die Stellgrößen  $Y_{\min}$ ,  $Y_{\max}$  und die Schaltdifferenz  $X_d$ . Wird ein Zweipunktregler  $G_R$  mit Rückführung  $G_r$  konfiguriert, so kommen auch deren Kennwerte in Betracht.

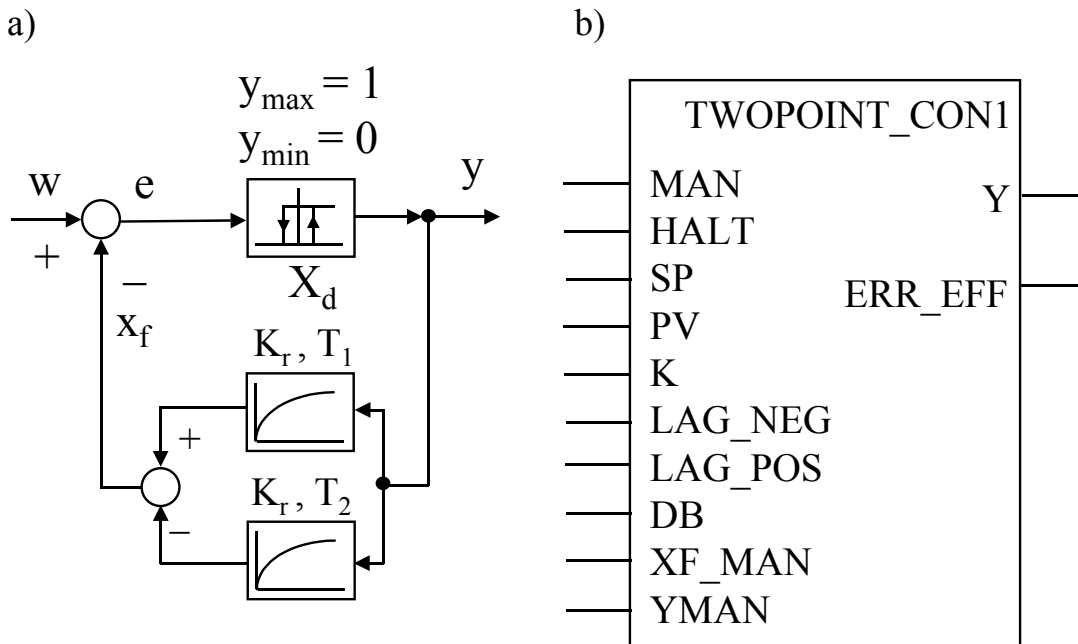
Als Rückführung wird oft eine Parallelschaltung von zwei P-T1-Glieder mit Kennwerten  $K_r$ ,  $T_1$  und  $T_2$  verwendet (**Bild 4.4**). Daraus entsteht ein vollständiger PID-Regler mit Kennwerten  $K_{PR}$ ,  $T_n$  und  $T_V$ .

Die Übertragungsfunktion des PID-Reglers kann in einer multiplikativen oder in einer additiven Form dargestellt werden:

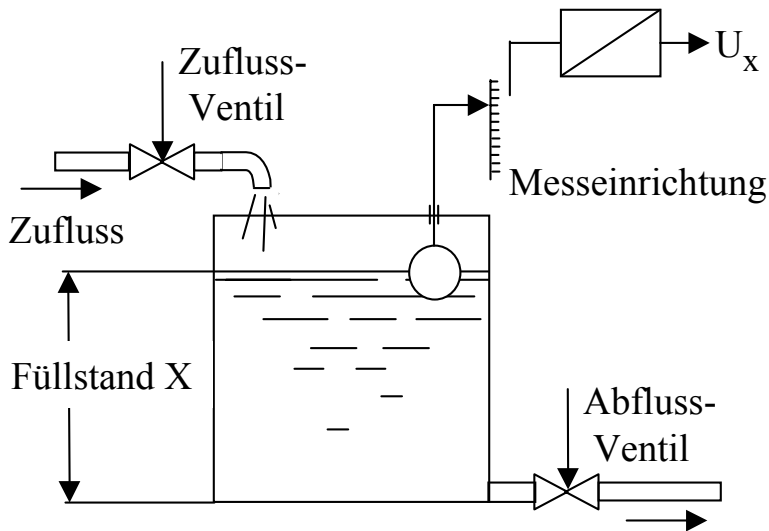
$$G_R(s) = \frac{K_{PR} (1 + sT_n)(1 + sT_V)}{sT_n} \quad \text{oder} \quad G_R(s) = K_{PR} \left( 1 + \frac{1}{sT_n} + sT_V \right)$$

Je nach beforzgte Form ergibt sich für den Zusammenhang zwischen den Regler- und Rückführungs-Kennwerten:

$$G_r(s) = \frac{K_r}{1 + sT_1} - \frac{K_r}{1 + sT_2} = \frac{K_r + K_r T_2 \cdot s - K_r - K_r T_1 \cdot s}{(1 + sT_1)(1 + sT_2)} = \frac{K_r(T_2 - T_1) \cdot s}{(1 + sT_1)(1 + sT_2)}$$



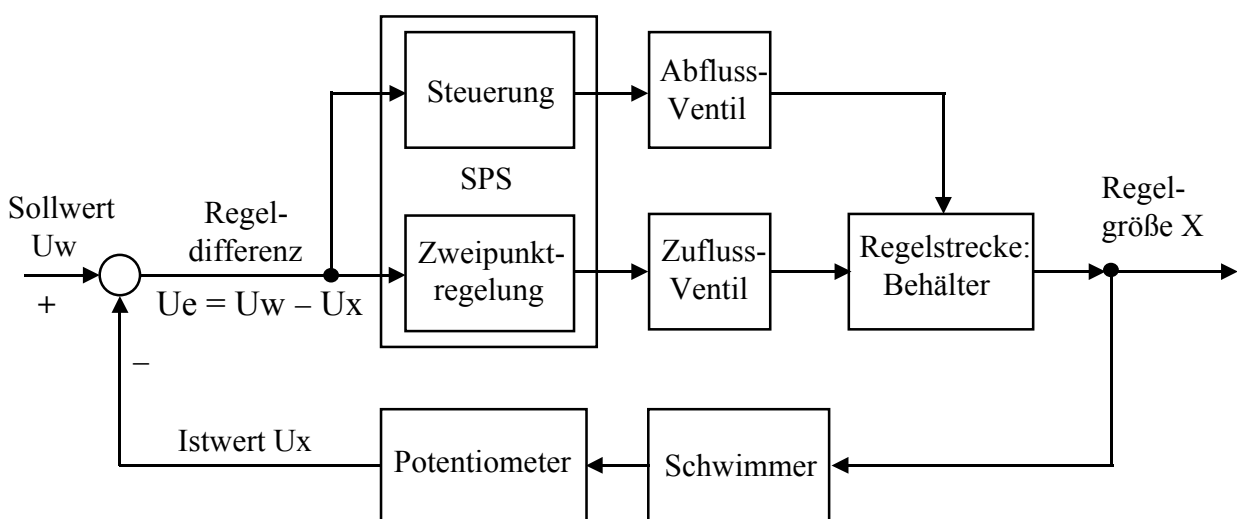
**Bild 4.4** Zweipunktregler mit Rückführung: a) Wirkungsplan des PID-Reglers; b) EFB



**Bild 4.5** Wasserbehälter als Beispiel einer Regelstrecke

Als Beispiel möge ein Wasserbehälter mit den zugehörigen Mess- und Stelleinrichtungen (**Bild 4.5**) dienen. Der Füllstand  $X$  ändert sich, wenn das Abflussventil oder das Zuflussventil betätigt wird. Schwimmer und Potentiometer stellen die Messeinrichtung dar. Die Ausgangsspannung des Potentiometers  $U_x$  wird zur SPS geleitet, mit dem Sollwert  $U_w$  verglichen (**Bild 4.6**) und auf einem konstanten Wert gehalten, d.h.  $U_w = U_x$ . Die Regelung soll mit einer SPS-Anlage als PI-Zweipunktregler erfolgen.

Der Wasserbehälter als Regelstrecke ist ein I-T1-Glied mit gegebenen Parametern:  $K_{PS} \cdot K_{IS} = 0,1 \text{ s}^{-1}$  und  $T = 2 \text{ s}$ .



**Bild 4.6** Wirkungsplan der Regelung und Steuerung des Wasserbehälters mit SPS

Die Übertragungsfunktion der Regelstrecke:

$$G_S(s) = \frac{K_{PS} \cdot K_{IS}}{s \cdot (1 + sT)}$$

Die Übertragungsfunktion des aufgeschnittenen Regelkreises

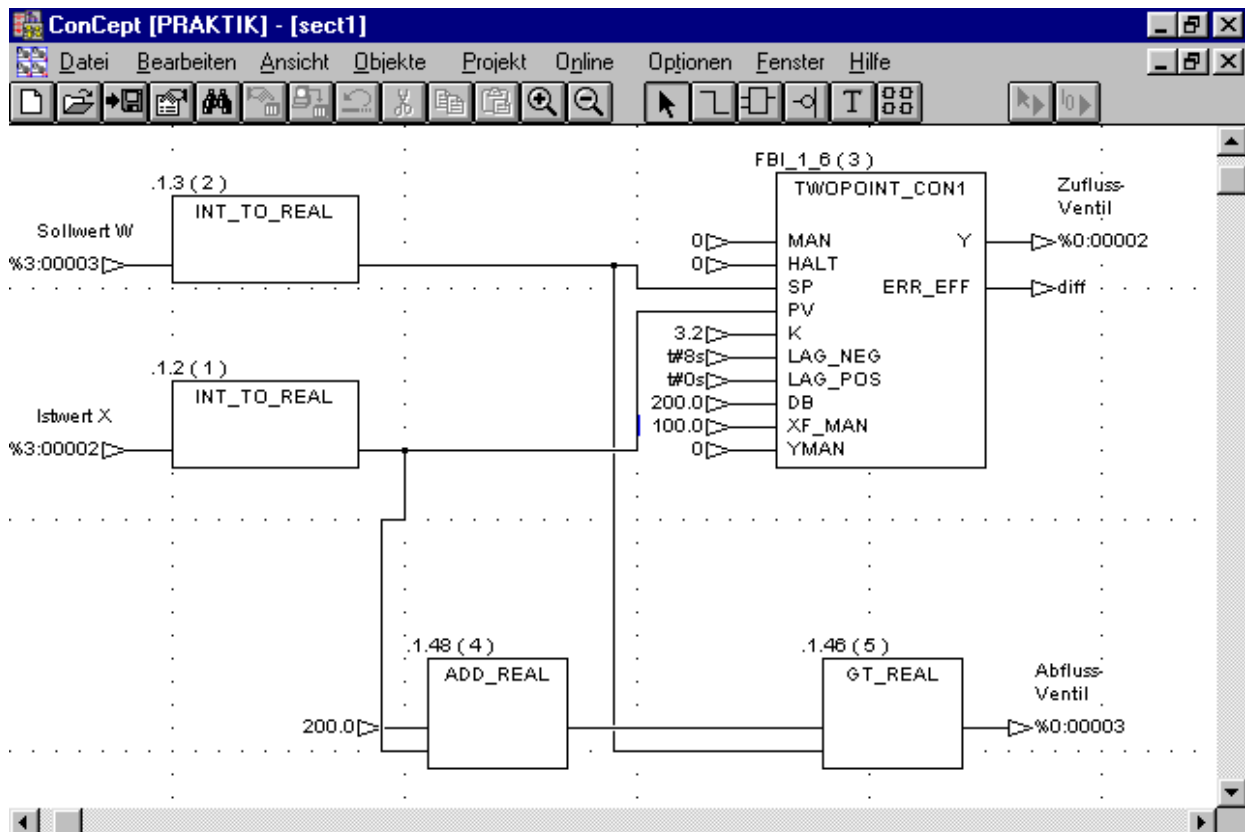
$$G_0(s) = \frac{K_{PR} \cdot (1 + sT_n)}{sT_n} \cdot \frac{K_{PS} \cdot K_{IS}}{s \cdot (1 + sT)}$$

lässt den Regler nach dem symmetrischen Optimum mit  $T_n = 4 \cdot T = 8$  s und

$$K_{PR} = \frac{1}{2 \cdot K_{PS} \cdot K_{IS} \cdot T_n} = 2,5$$

einstellen. Daraus folgen die Kennwerte der Rückführung (Bild 4.7):  $T_1 = 0$ ,

$$T_2 = T_n = 8 \text{ s und } K_r = \frac{1}{K_{PR}} \cdot \frac{T_2 + T_1}{T_2 - T_1} = 3,2$$



**Bild 4.7** Beispiel einer Füllstandsregelung mit dem Zweipunktregler



## 4.4 Simulierte Regelstrecke

Mit dem IEC-Simulator kann man nicht nur den Regler sondern auch die Regelstrecke simulieren. Die CONT\_CTL-Bibliothek stellt die geeigneten EFB zur Verfügung.

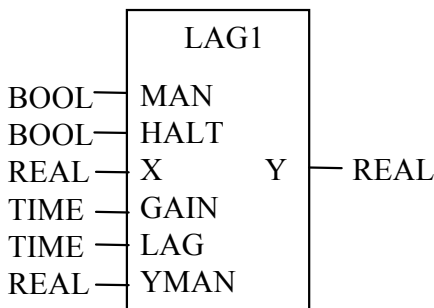
### 4.4.1 P-T1-Glied

Ein Verzögerungsglied 1. Ordnung kann mit EFB LAG, LAG1 und LAG\_Filter simuliert werden. **Bild 4.18** zeigt als Beispiel den EFB LAG1 mit der Übertragungsfunktion

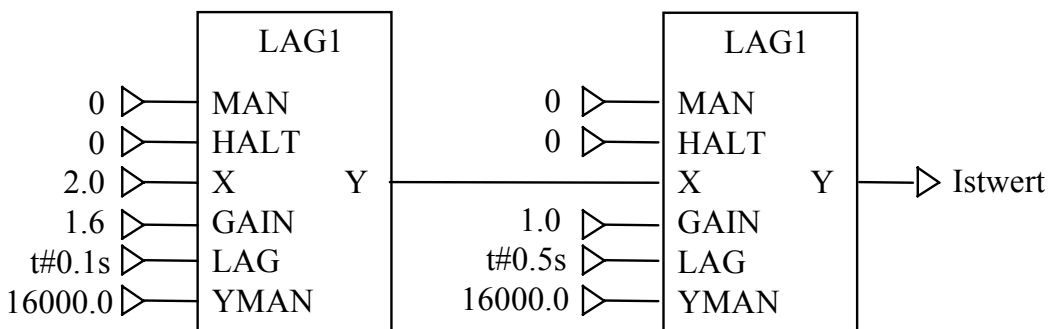
$$G_S(s) = \frac{K_{PS}}{1 + sT}$$

Der Proportionalbeiwert  $K_{PS}$  wird als GAIN und die Zeitkonstante  $T$  als LAG bezeichnet.

Das Beispiel einer Regelstrecke, die aus zwei hintereinander geschalteten P-T1-Gliedern mit Parametern  $K_{PS1} = 1,6$ ,  $T_1 = 0,1$  s,  $K_{PS2} = 1,0$  und  $T_2 = 0,5$  s besteht, ist im **Bild 4.18** dargestellt.



**Bild 4.18** EFB LAG1 der CONT\_CTL-Bibliothek als P-T1-Glied



**Bild 4.19** Beispiel einer Regelstrecke mit zwei P-T1-Gliedern

## 4.6 Aufgaben zum Kapitel 4

### Aufgabe 4.1

Ein PID-Regler soll mit der Bausteinbibliothek CONT\_CTL programmiert werden. Wenn die Stellgröße  $y \geq (Y_{\max} - 1 \text{ V})$  ist, soll der Regler auf einen manuellen Betrieb mit  $Y_{\text{man}} = 5 \text{ V}$  umgeschaltet werden.

Die Kennwerte des Reglers sind unten gegeben:

P-Anteil:	$K_{\text{PR}}$	$= 2$
I-Anteil:	$K_{\text{I}}$	$= 0,4 \text{ s}^{-1}$
D-Anteil:	$K_{\text{D}}$	$= 0,2 \text{ s}$
Verzögerung:	$T_{\text{R}}$	$= 0,01 \text{ s}$
Stellbereich: von	$Y_{\text{min}}$	$= 0 \text{ V}$ bis $Y_{\text{max}} = 10 \text{ V}$

Die Verbindung der SPS mit dem Prozess erfolgt nach folgenden Adressen:

Variable	Adresse
Regelgröße $x$	%3:00003
Führungsgröße $w$	%3:00002
Stellgröße $y$	%4:00001
Regeldifferenz $e$	%4:00002
Begrenzung der Stellgröße $Q_{\text{max}}$	%0:00001
Begrenzung der Stellgröße $Q_{\text{min}}$	%0:00002

### Aufgabe 4.2

Ein Reaktionsgefäß mit dem Zulaufventil  $V_{\text{Zu}}$  (BOOL-Variable *Zulauf*) und dem Ablaufventil  $V_{\text{Ab}}$  (BOOL-Variable *Ablauf*) wird als Regelstrecke betrachtet.

Die Belegung von Adressen:

- Istwert des Füllstands  $X$  – analoge Adresse %3:00004
- Sollwert des Füllstands – digitale Adresse %3:00005.

Eine Steuerung zur Messung der Füllzeiten mit einer Schaltdifferenz (Hysterese) von  $\pm 0,2 \text{ V}$  soll programmiert werden:

- Anfangszustand: Reaktionsgefäß ist leer und beide Ventile sind geschlossen
- Das Zulaufventil  $V_{\text{Zu}}$  wird geöffnet
- Die Füllzeit  $t_{\text{füll}}$  bis zum gegebenen Sollwert wird gemessen.

## **5 Projektbeispiel: Prozessautomatisierung**

### **5.1 Aufgabenstellung**

#### **5.1.1 Entwurfsstufen eines Projekts**

Die Automatisierung einer Anlage besteht aus folgenden Funktionen: Messen – Steuern – Regeln – Bedienen – Visualisieren. Bei der Prozessautomatisierung zählt hierzu auch die Vernetzung einzelner SPSen in einem verteilten Steuerungssystem.

Die Entwurfsstufen einer speicherprogrammierbaren Steuerung werden üblicherweise in folgender Reihenfolge durchgeführt:

- Konfigurierung
- Programmierung
- Test und Inbetriebnahme
- Diagnose

Die Aufgaben einer Regelung werden auf eine ganz andere Art gelöst:

- Identifikation der Regelstrecke, d.h. die experimentelle Untersuchung, die mathematische Beschreibung und die Bestimmung der Prozessparameter
- Vereinfachung des Prozessmodells, z.B. die Linearisierung oder die Entkopplung mehrerer Regelgrößen
- Entwurf des Regelkreises
- Realisierung des Regelalgorithmus

Unter der letzten Entwurfsstufe versteht man heute die Parametrierung einer SPS oder PLS. Damit verschmelzen die beiden Entwurfsvorgänge – die Entwicklung einer Regelung und die SPS-Programmierung – zu einer Projektaufgabe der Prozessautomatisierung.

Komplexe Automatisierungsprojekte bedürfen einer sorgfältigen Aufteilung des Gesamtprojektes in Teilprojekte. Wenn die jeweiligen Aufgabenstellungen formuliert und die Schnittstellen zwischen den einzelnen Teilaufgaben definiert sind, kann das Projekt auf mehrere Entwickler aufgeteilt und von ihnen parallel bearbeitet werden. Die letzte Stufe ist die Einbindung der Teil-Lösungen zur Gesamt-Lösung, d.h. von Einzelprojekten zum lauffähigen Automatisierungssystem und ein Integrationstest.

Nachfolgend werden einige Entwurfsstufen der Prozessautomatisierung mit SPS am konkreten, vereinfachten Beispiel eines Rührkesselreaktors dargestellt.

## 5.4 Teilprojekt: Entwurf einer Regelung

Der Einfachheit halber soll der Prozess mit zwei getrennten Regelkreisen (also keine Zweigrößenregelung, kein Beobachter...) geregelt werden. Die dazugehörigen Regelgrößen sind:

- Füllstand als *Füll\_ist* (Signal AI1) mit dem Sollwert *Füll\_soll* (Signal AI3)
- Temperatur als *Temp\_ist* (Signal AI2) mit dem Sollwert *Temp\_soll* (AI4)

Die Adresse %4:00004 von *Heiz\_Zulauf* ist der SPS Momentum angepasst.

### 5.4.1 Temperaturregelkreis

Ein PIDP1-Funktionsbaustein als PID-Regler wird für den Temperaturkreis eingesetzt. Das Programm ist zu groß, um in ein einziges Fenster zu passen. Ein Teil des Programms ist im **Bild 5.7** zu sehen. Am Ausgang der Reglers ist ein DFB UMFORM zwecks Visualisierung mit Lookout geschaltet. Der DFB wandelt die analogen Werte in digitale um. Der DFB ADIGMOM dient der Konfiguration der E/A-Baugruppe 170 AMM 090 00 der SPS Momentum. Das Ventil *Produkt\_Zulauf* wird mittels einer binären Steuerung betätigt.

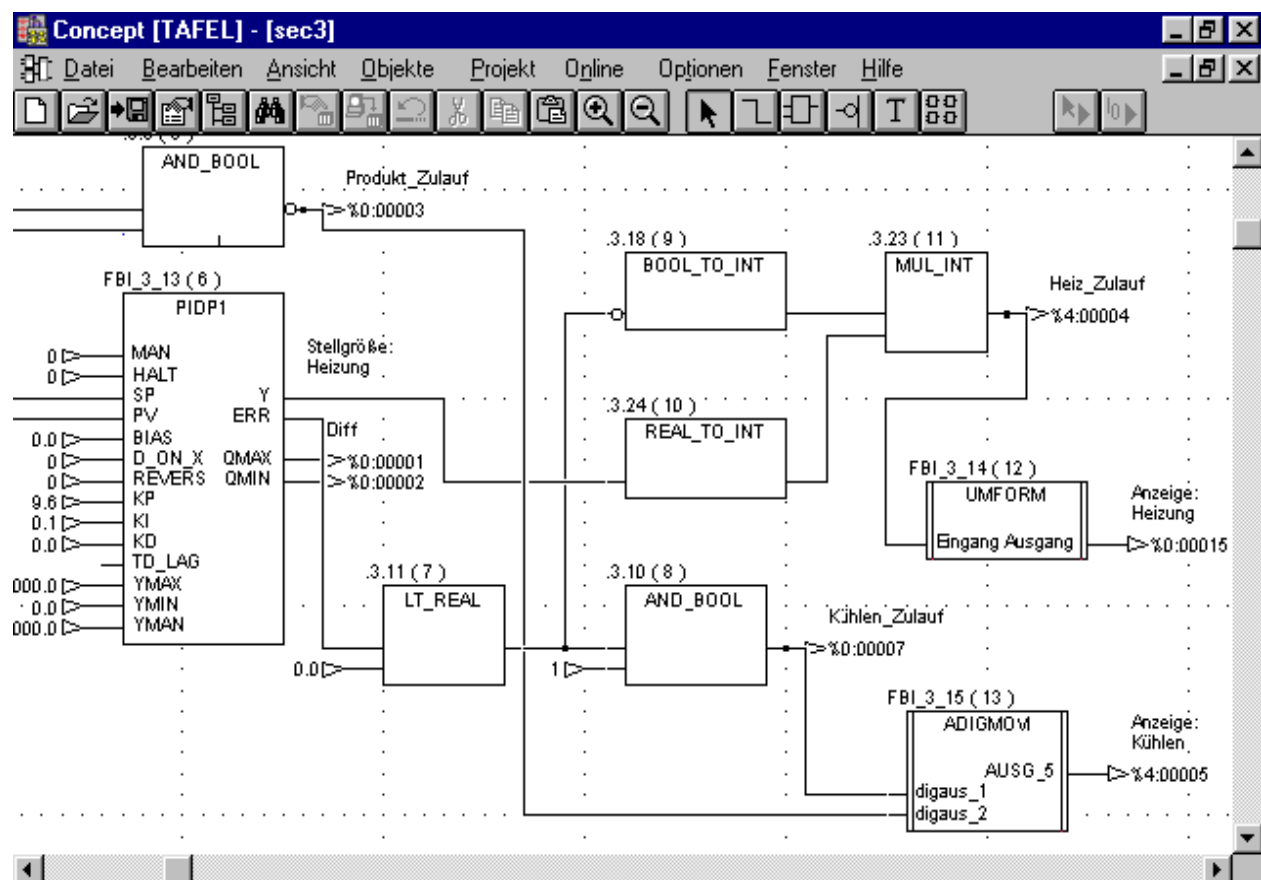
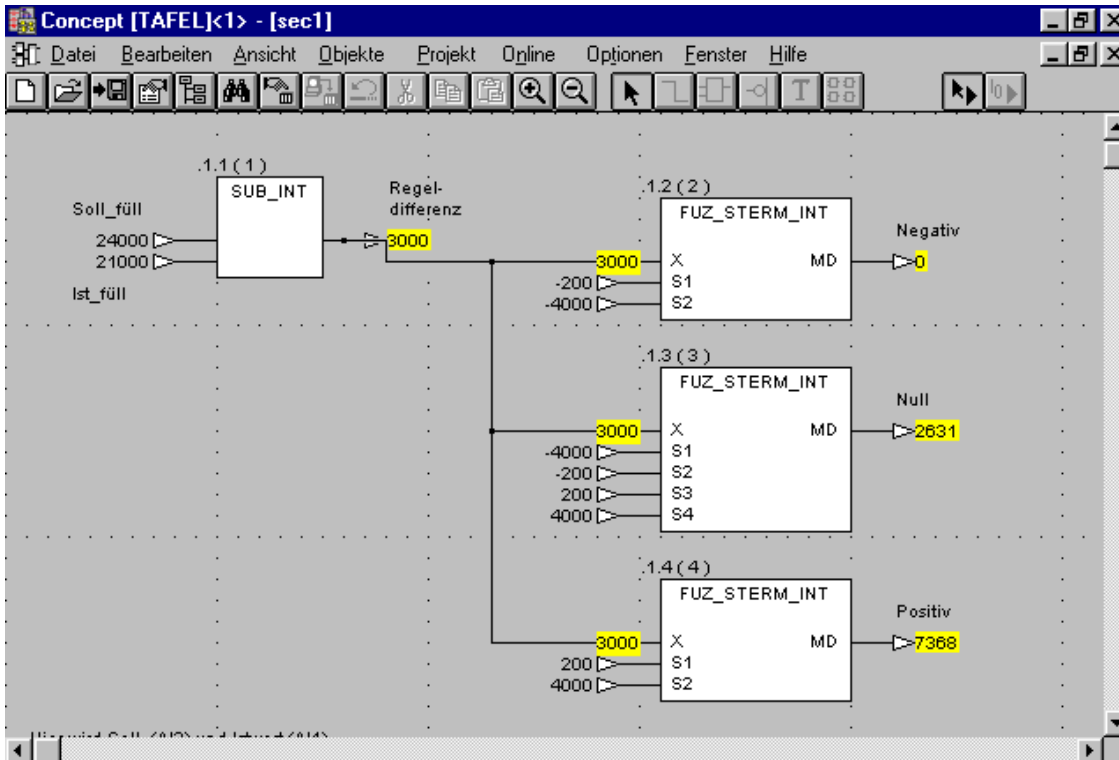


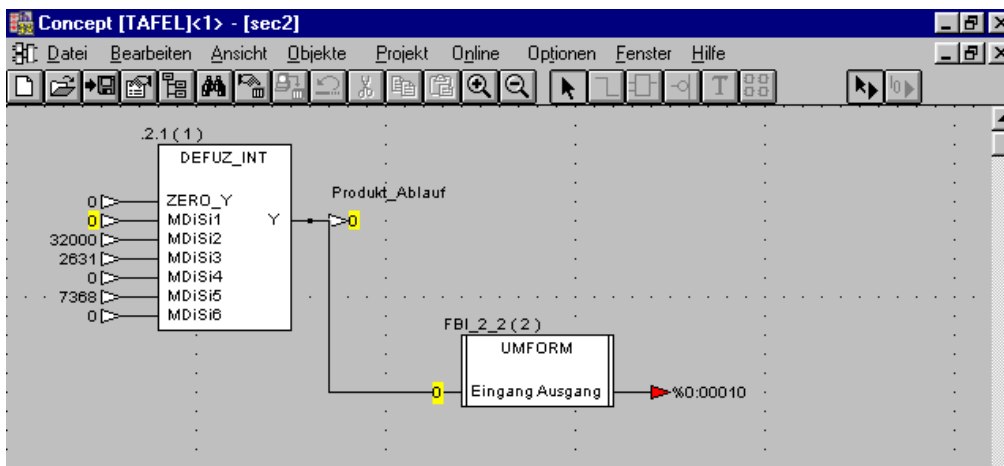
Bild 5.7 Die Temperaturregelung des Reaktors mit dem PID-Regler

## 5.4.2 Füllstandsregelkreis

Der Füllstand wird mit einem Fuzzy-Regler geregelt. Im **Bild 5.8** ist die Sektion mit den Fuzzifizierung gezeigt. Deutlich sind die zwei aktiven Terme *Null* und *Positiv* zu sehen. Für die gleiche aktuelle Regeldifferenz ist im **Bild 5.9** die Defuzzifizierung abgebildet.



**Bild 5.8** Die Füllstandsregelung des Reaktors mit dem Fuzzy-Regler



**Bild 5.9** Defuzzifizierung: Im aktuellen Zustand ist die Stellgröße *Produkt\_Ablauf* = 0.

## 5.6 Prozessvisualisierung

Die grafische Abbildung des Rührkesselreaktors und die Anzeige-/Bedienfunktionen werden mit Lookout realisiert. Wie im Abschnitt 5.1.2 schon erwähnt wurde, ist das Programm Lookout für die SPS-PC-Verbindung über Modbus gut geeignet. Bevor der Entwurf des Visualisierungsprogramms behandelt wird, soll eine Darlegung der wesentlichen Eigenschaften von Lookout den Einstieg in das Programmieren erleichtern. Mehr darüber kann man in [14] lesen.

### 5.6.1 Konzept und Funktionsweise von Lookout

Lookout zeichnet sich besonders durch einfache Handhabung aus, die die Entwicklung von Visualisierungsprogrammen innerhalb kürzester Zeit ermöglicht. Durch die Online-Konfiguration kann man die bestehenden Applikationen und die Verbindungen zur SPS während eines laufenden Prozesses verändern.

Lookout basiert auf einer objektorientierten Architektur, die den Prozess durch Objekte beschreibt und die Objekte miteinander verknüpft. Unter Objekten versteht man komplexe Einheiten mit grafischer Darstellung, Funktionen und Datenhaltung. Durch verschiedene Steuerungsobjekte wie PID-Regler kann der Prozess nicht nur überwacht sondern auch gesteuert werden. Ein gewünschtes Objekt wird aus Lookout-Bibliotheken mit Objektklassen ausgewählt, mit eigenem Namen (Tagname) und mit Parametern versehen. Um den Zugriff auf Objekte einzuschränken oder gar zu verweigern, können bis zu zehn Sicherheitsebenen eingerichtet werden.

Jedes Objekt besteht aus:

- Objektdaten, die der Kommunikation zwischen den Objekten dienen und die über die Funktion *Edit Connections* miteinander verbunden werden können.
- Objektparametern, die das grafische Aussehen eines Prozesselements (z.B. eines Schieberegisters) oder die Einstellungen für ein serielles Datenprotokoll beschreiben, z.B. die Datenrate, die Parität und die Stoppbits.

Die Objektdaten und Objektparameter bilden eine eigenständige Einheit mit eigener Datenbank, die automatisch beim Kreieren eines Objekts angelegt wird. In einer Datenbank können bis zu mehrere tausend Objektdaten gespeichert werden. Der Lookout-Objektkatalog beinhaltet Objekte unterschiedlicher Kategorien wie Kalkulation, Steuerung, Anzeige, Protokolltreiber oder Networking. Nach ihren Funktionen werden Objekte in Objektklassen zusammengefasst. Damit wird eine großen Anzahl von Objekten auf wenige Objektklassen komprimiert, z.B. statt

Das Erstellen einer neuen Applikation erfolgt nach folgenden Schritten:

### 1. Schritt: Starten

Man startet Lookout und wählt ein neues Projekt im Menüpunkt *File/ New* an. Es wird ein neues Bedienfenster *New Control Panel* geöffnet, dem man einen neuen, mit *OK* zu bestätigenden Namen geben soll.

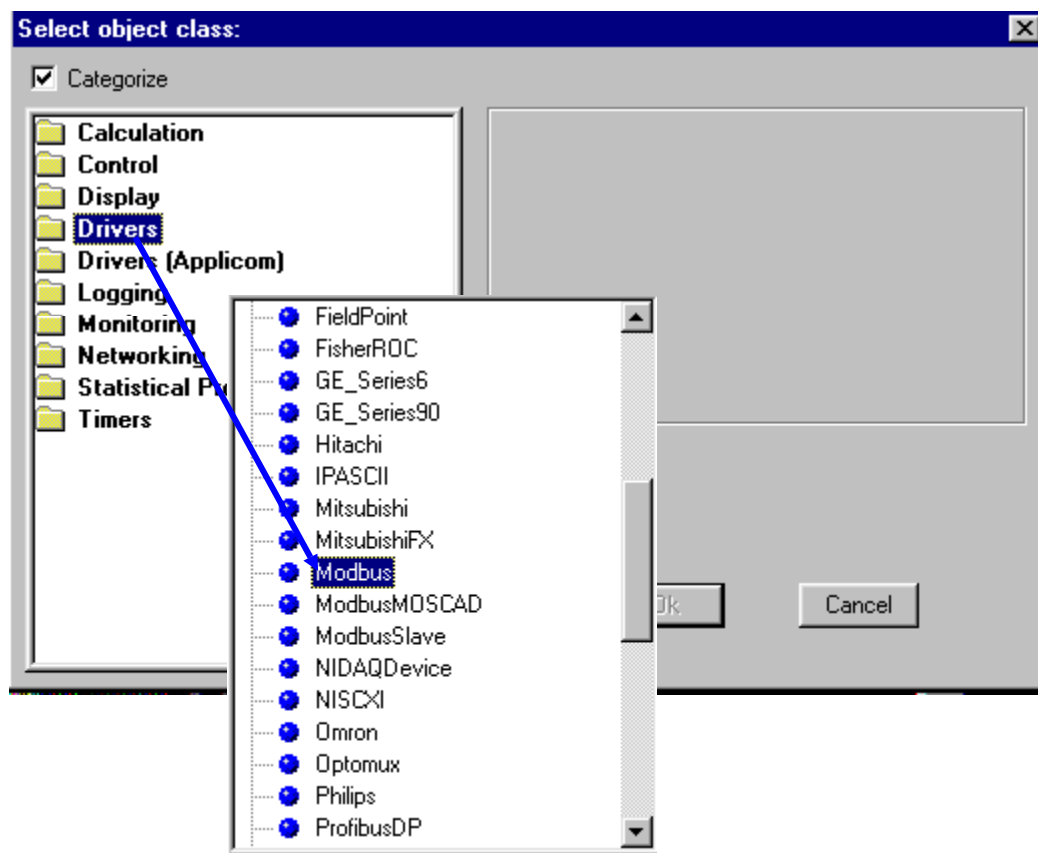
### 2. Schritt: Modbusobjekt erzeugen

Um in Lookout mit der SPS zu kommunizieren, muss zuerst ein neues Modbusobjekt erzeugt werden. Dafür ruft man das Treiberverzeichnis unter Menüpunkt *Objekt/Create...* und wählt daraus ein Modbusobjekt aus, wie im **Bild 5.16** abgebildet ist.

### 3. Schritt: Konfiguration eines Objektes

Die Konfiguration des Objektes wird geändert, wie beispielsweise im **Bild 5.17** dargestellt ist, und mit einem *OK* bestätigt.

Damit ist Lookout an den Prozess angekoppelt, eine Datenbank des Objektes kann kreierte werden.



**Bild 5.16** Erzeugen eines Modbus-Objektes

### 5. Schritt: Visualisierung des Objektes Tank

Mit *File/ Open* startet man die Applikation *Lookout\Übungen\Übung1.Ikp* aus dem vorherigen Schritt und wechselt in den Modus *Edit*. Umschalten ist über *Edit/ Edit Mode* oder über die Tastenkombination *Ctrl-Leertaste* möglich.

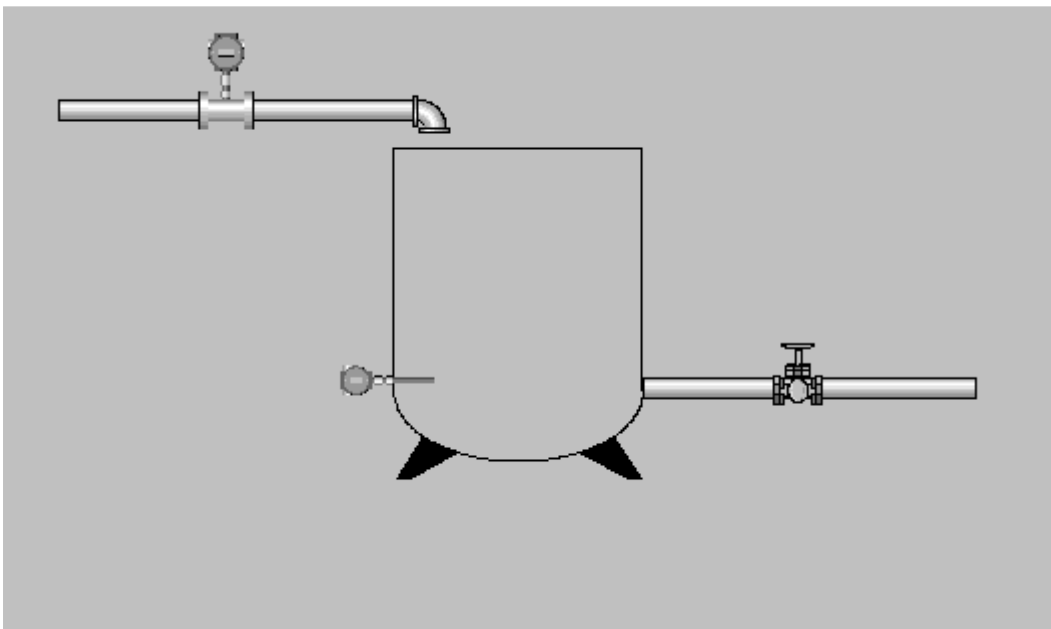
Das Erreichen des *Edit*-Modus wird durch einen gelben Balken am unteren Bildschirmrand gekennzeichnet.

Mit dem Menüpunkt *Insert/Graphic...* werden die grafischen Elemente

- Tanks\Tank5\_A.wmf
- Pipes\Pipe\_H.wmf
- Pipes\Elbo\_UR.wmf
- Valves\VLV07U.wmf
- Meters\Inline1U.wmf
- Meters\Probe1L.wmf

eingefügt und mit der Maus oder mit der Tastenkombination *Leertaste* und *Pfeiltasten* zu einer grafischen Darstellung des Behälters zusammengefasst (**Bild 5.19**). Damit entsteht nicht nur die grafische Abbildung des Prozesses sondern auch eine objektorientierte Applikation, die – ergänzt mit Anzeige-Elementen – auf die Ereignisse des Prozesses entsprechend vorgegebener Funktionalitäten reagieren wird.

Die Applikation wird unter *Lookout\Übungen\Übung2.Ikp* gespeichert.



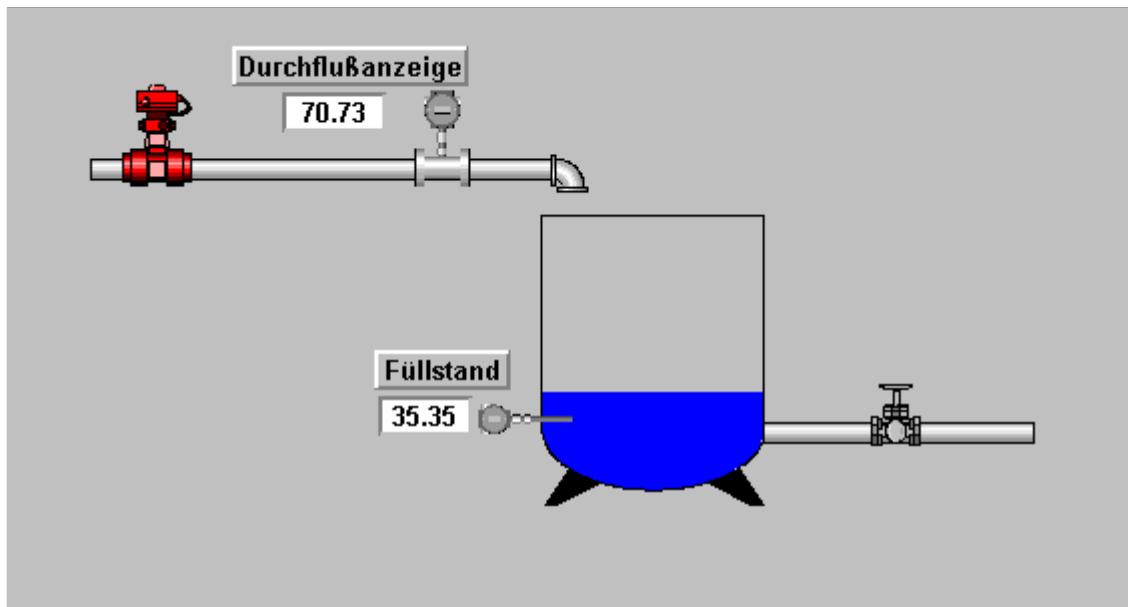
**Bild 5.19** Visualisieren des Objektes *Tank* mit Zufluss und Abfluss, einem Abflussventil und zwei Messgeräten (Füllstand und Durchfluss beim Füllen des Reaktors)



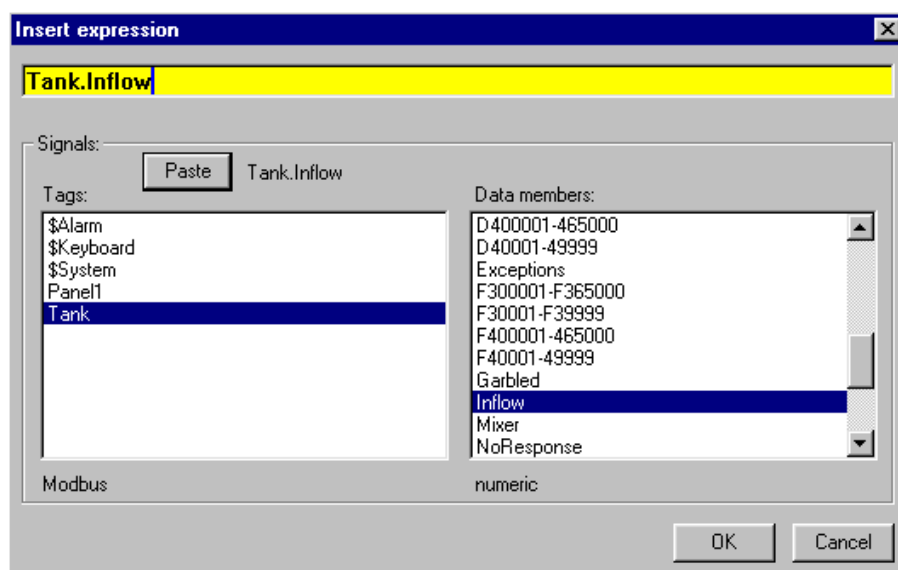
Die digitalen und analogen Signale der SPS sollen nun in Form von grafischen und digitalen Anzeigen dargestellt werden (**Bild 5.20**). Dafür werden entsprechenden *Expressions* erstellt und eingefügt.

#### 6. Schritt: Digitale Anzeige von Messwerten

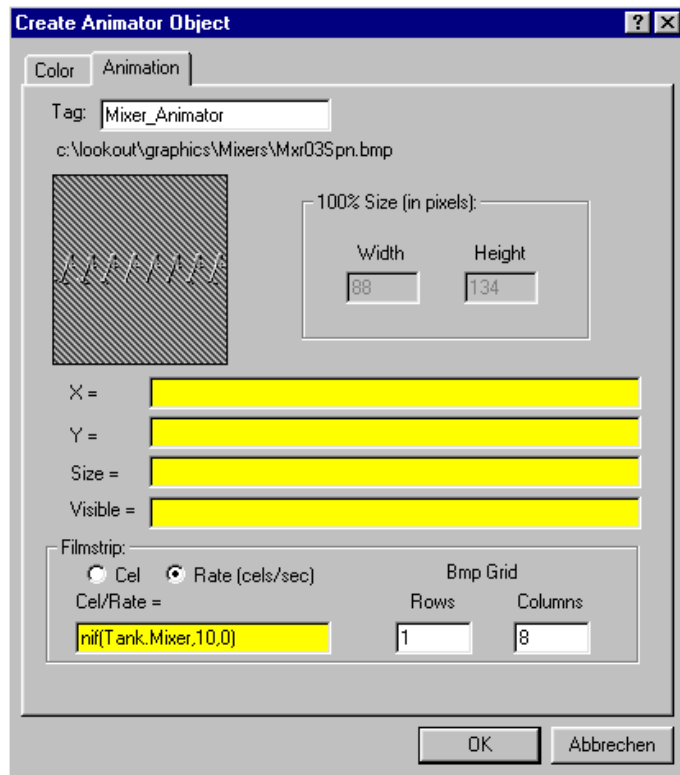
Man wählt zunächst den Menüpunkt *Insert/Expression*, um das Konfigurationsfenster zu öffnen, in dem die gewünschte Messstelle aus der Objektliste *Tank*, z.B. *Tank.Inflow* (**Bild 5.21**), markiert wird.



**Bild 5.20** Die gewünschte Abbildung des Prozesses mit Füllstands- und Durchflussanzeigen

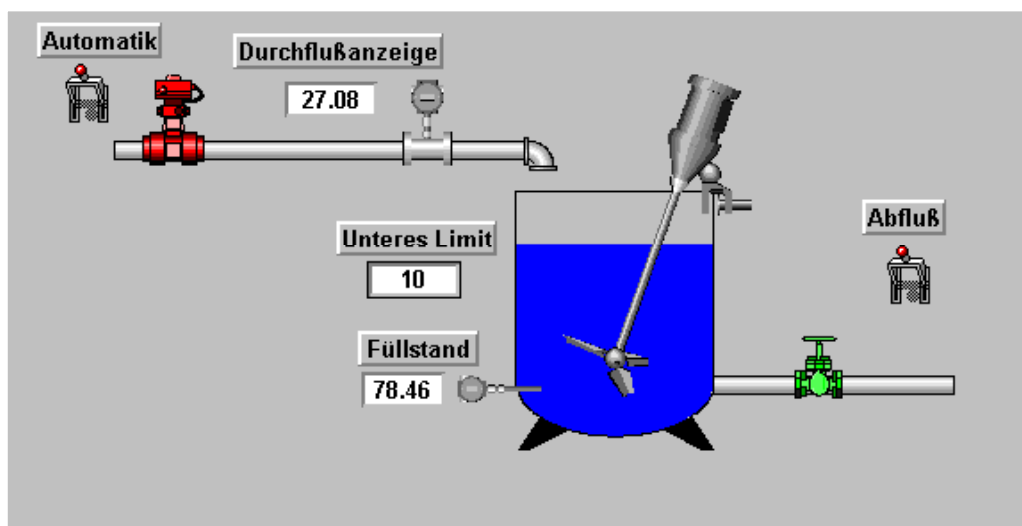


**Bild 5.21** Konfigurationsfenster der *Expressions* für Anzeige des Durchflusses (*Inflow*)



**Bild 5.33** Konfigurierungsfenster für Prozessanimation eines Mixers mit der Geschwindigkeit von 10 Bilder pro Sekunde

Für Ein-/Ausschalten des Objekts *Tank.Mixer* wird ein neues Objekt *Schalter6* erstellt. Beide Objekte werden über einen NIF-Ausdruck (numerische IF-Funktion) verknüpft und unter *Lookout\Übungen\Übung5.Ikp* gespeichert (**Bild 5.34**).



**Bild 5.34** Prozessvisualisierung des Behälters mit einem animierten Mixer. Die Animations-Geschwindigkeit (Bilder pro Sekunde) wird vorgegeben, die Drehrichtung ist nicht wählbar.

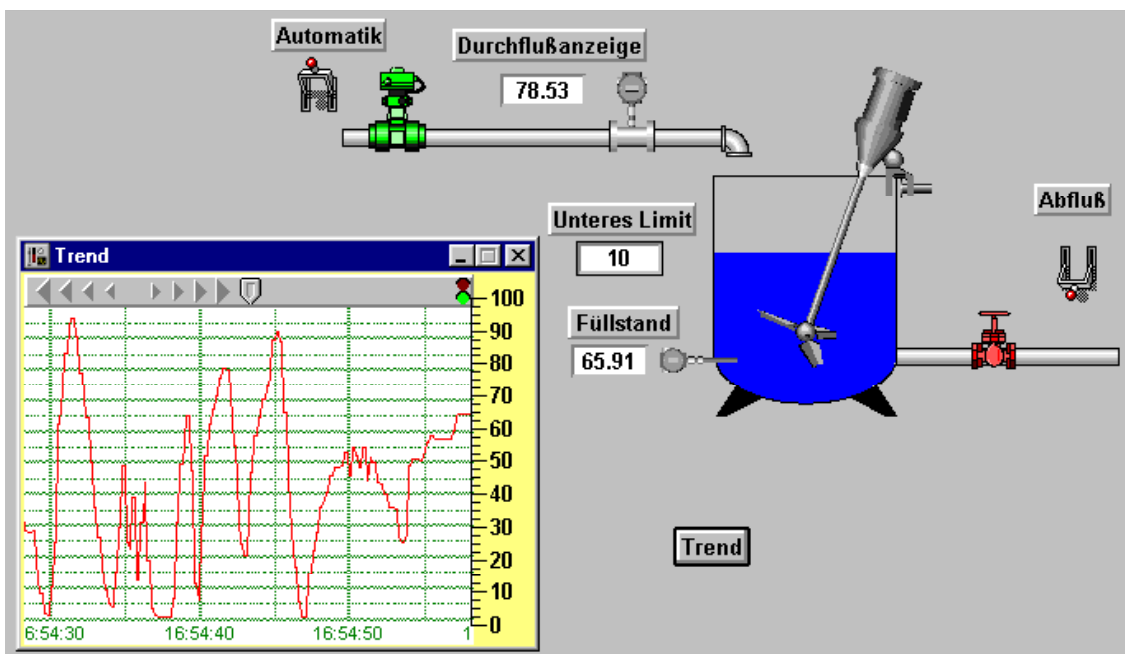
### 5.6.6 Erstellen einer Trenddarstellung

Unter Trend versteht man die grafische Darstellung von Messwerten (Tag-Werten) in einem Zeitdiagramm, in dem die Werte über eine Trendperiode wiedergegeben werden und die einzelnen Datenpunkte zu Trendlinien verbunden sind. Der Trend kann nicht nur über die Zeit sondern auch über Messgrößen wie Temperatur oder Druck mit einem *X-Y-Chart* dargestellt werden. In einer Applikation kann ein Trend als neues Element auf der Bildoberfläche oder als neues Fenster angelegt werden (**Bild 5.35**).

Es gibt zwei Arten der Trendermittlung und -darstellung:

- Echtzeit-Darstellung, wenn die Tag-Werte in Echtzeit über eine kurze Zeitspanne gezeigt werden
- Hystorische Darstellung, wenn die Tag-Werte aus einem Datenträger aufgerufen und gezeigt werden.

In Lookout kann man einen HyperTrend aus beiden Darstellungen in einem einzigen Diagramm kombinieren. Die Anzahl von HyperTrend-Objekten ist praktisch unbegrenzt und nur von der Größe der Applikation abhängig. Die Attribute wie Dimension oder Zeitspanne sowie die Skalierung von einzelnen Kurven sind frei wählbar. Mit Hilfe von Scrollfunktionen kann man die Trendkurven in andere Zeitabschnitte übertragen. Nachfolgend wird die bereits erstellte Applikation des Wasserbehälters durch ein neues Trendfenster erweitert.



**Bild 5.35** Echtzeit-Trenddarstellung des Füllstandes

## 5.7 Aufgaben zum Kapitel 5

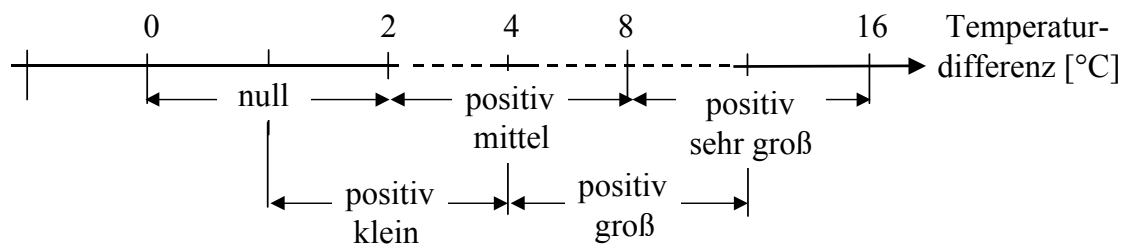
### Aufgabe 5.1

Die Temperatur eines Prozesses, z.B. der Prozesstafel ET722, soll mit vier Ventilen gesteuert werden. Mit dem Stellventil *Heiz\_zulauf* unter der analogen SPS-Adresse %4:00001 wird die Heizung mit einem Fuzzy-Regler geregelt. Mit dem Ventil *Kühlmittel\_zulauf* (digitale SPS-Adresse %0:00004) wird die Kühlung angesteuert. Die Ablaufventile für die Heizung *Heiz\_ablauf* und für die Kühlung *Kühlmittel\_ablauf* dienen dazu, die Heiz- und Kühlmittelkreisläufe ein- und auszuschalten.

Falls die Prozesstafel ET722 nicht zu Verfügung steht, kann eine beliebige Reihenschaltung zweier P-T1-Glieder als Regelstrecke benutzt werden. Im betrachteten Beispiel sind die Parameter der Regelstrecke:  $K_{PS} = 0,8$  und  $T_1 = 6$  s,  $T_1 = 94$  s, was dem 50%-Füllstand der Prozesstafel ET722 entspricht.

Die Prozessautomatisierung ist unter folgenden Bedingungen zu entwerfen:

- Das Produkt soll auf einer konstanten Soll-Temperatur  $T_{Soll} = 50$  °C (analoge Adresse %2:00003) gehalten werden.
- Als Regelgröße steht die Ist-Temperatur  $T_{Ist}$  am Eingang (analoge Adresse %3:00003 ) zur Verfügung.
- Der Fuzzy-Regler hat zwei Eingangsgrößen: Regeldifferenz (°C) und Ableitung (°C/min).
- Als linguistischen Variablen fungieren *Temperaturdifferenz*, *Temperaturableitung* am Eingang sowie *Stellgrad Heizventil* am Ausgang.
- Die Eingangs- und Ausgangsvariablen werden in fünf Wertebereiche unterteilt.
- Die Stellbereiche werden so gewählt, dass die Temperaturdifferenz zwischen 0 °C und 8 °C und die Temperaturableitung zwischen  $\pm 1,5$  °C/min liegt.
- Der Stellbereich des Heizventils liegt zwischen 0 % und 100 %.
- Die linguistische Terme sind im **Bild 5.42** und **Bild 5.43** definiert.



**Bild 5.42** Linguistische Variablen und Terme der Temperaturdifferenz