

Module der Automatisierungstechnik

Prof.em, Dr.-Ing. S. Zacher

Regelungstechnik mit MATLAB/Simulink

Quellen mit insgesamt 62 Simulink-Modellen:

[1] **Zacher, S., Reuter, M.: Regelungstechnik für Ingenieure**

Verlag Springer Vieweg, 14. Auflage, 2014

Seiten: 231, 268, 270, 273, 274, 299, 360, 363-366, 369, 386, 397,
403, 408, 414, 438-440, 459, 461

[2] **Zacher, S.: Übungsbuch Regelungstechnik**

Verlag Springer Vieweg, 5. Auflage, 2014

Seiten: 63, 66, 80, 129, 131, 135, 137, 152, 163, 171, 187-189,
192, 198, 203, 205-207, 216, 217, 234, 235, 237, 238

[3] **Zacher, S.: Regelungstechnik Aufgaben**

Verlag Dr. Zacher, 3. Auflage, 2012

Seiten: 35, 82-84, 87, 89, 90, 94, 100, 105, 108-111, 116, 118

Inhalt:

1 Einführung: Simulation

- 1.1 Datenübergabe MATLAB-Simulink (3)
- 1.2 To Workspace (5)
- 1.3 Scope-Konfiguration
- 1.4 DDE-Kommunikation (8)

2 Standardregler

- 2.1 P-Regler (11)
- 2.2 PI-Regler als Summe P+I (12)
- 2.3 PI-Regler mit PID (13)
- 2.4 To Workspace (14)
- 2.5 Parallelschaltung (16)

3 Identifikation

- 3.1 Wendetangentenverfahren(17)
- 3.2 Aufgabe (18)
- 3.3 Zeit-Prozentkennwert (20)
- 3.4 Aufgabe: Messwerte (22)

4 Modellbasierte Regler

- 4.1 Kompensationsregler (23)
- 4.2 Smith-Prädiktor (26)
- 4.3 ASA-Regler (29)
- 4.4 PFC-Regler (33)
- 4.4 SPFC-Regler (41)

5 Trajektorienfolge-Regelung

- 5.1 Definition (50)
- 5.2 Dead-beat Regler (51)

6 Mehrgrößenregelung

- 6.1 MIMO-Strecke (53)
- 6.2 Entkopplung (53)
- 6.3 Bus-Konzept für MIMO (54)

7 Zweipunktregler

- 7.1 Zweipunktregler ohne Hysterese (55)
- 7.2 Zweipunktregler mit Hysterese (55)

8 Digitale Regler

- 8.1 Digitaler I-Regler (56)
- 8.2 Digitaler P-Regler (56)
- 8.3 Digitaler dead-beat Regler (57)

9 Zustandsrückführung

- 9.1 Stabilität (58)
- 9.2 Zustandsrückführung (59)
- 9.3 Zustandsbeobachter (60)

1 Einführung: Simulation

1.1 Datenübergabe MATLAB - Simulink

MATLAB ist ein befehlsorientiertes Programm, bei dem die Befehle zeilenweise ins Command-Window eingetragen werden.

MATLAB/Simulink ist ein grafisches Programm, in dem die vordefinierten Bausteine aus dem Bibliothek (Library) per Plug-and Play geholt und in eine dafür geöffnete Bedienoberfläche platziert werden. Danach werden die Bausteine vom Benutzer entsprechend der Aufgabenstellung konfiguriert.

MATLAB und MATLAB/Simulink haben eine gemeinsame Zwischenablage, die **Workspace** heißt. Bei einem neu geöffneten Programm ist die Workspace leer. Während der Programmausführung wird die Workspace mit Daten gefüllt. Die Daten sind wie Matrizen dargestellt.

Beispiel:

Command Window

```
EDU>> A=[1 2 3 4 5];
```

```
EDU>> B=[6; 7; 8; 9; 10];
```

```
EDU>> C=A*B
```

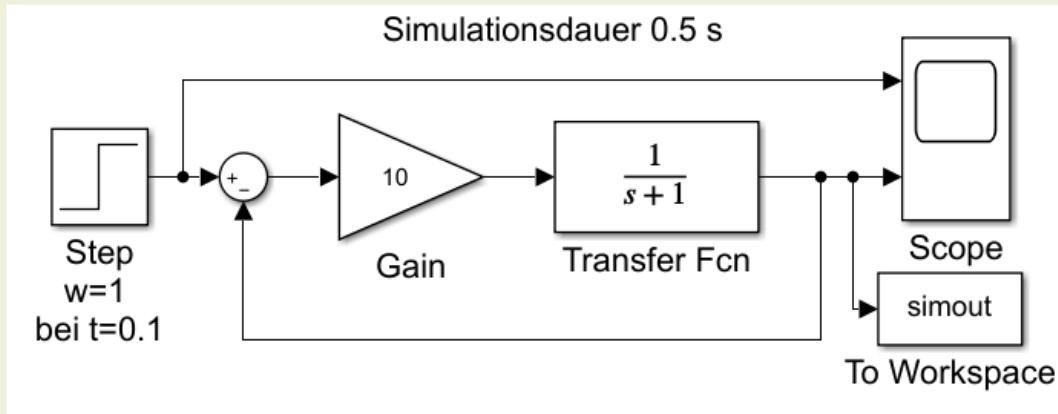
```
C =
```

```
130
```

Workspace

Name ^	Value	Min	Max
A	[1 2 3 4 5]	1	5
B	[6;7;8;9;10]	6	10
C	130	130	130

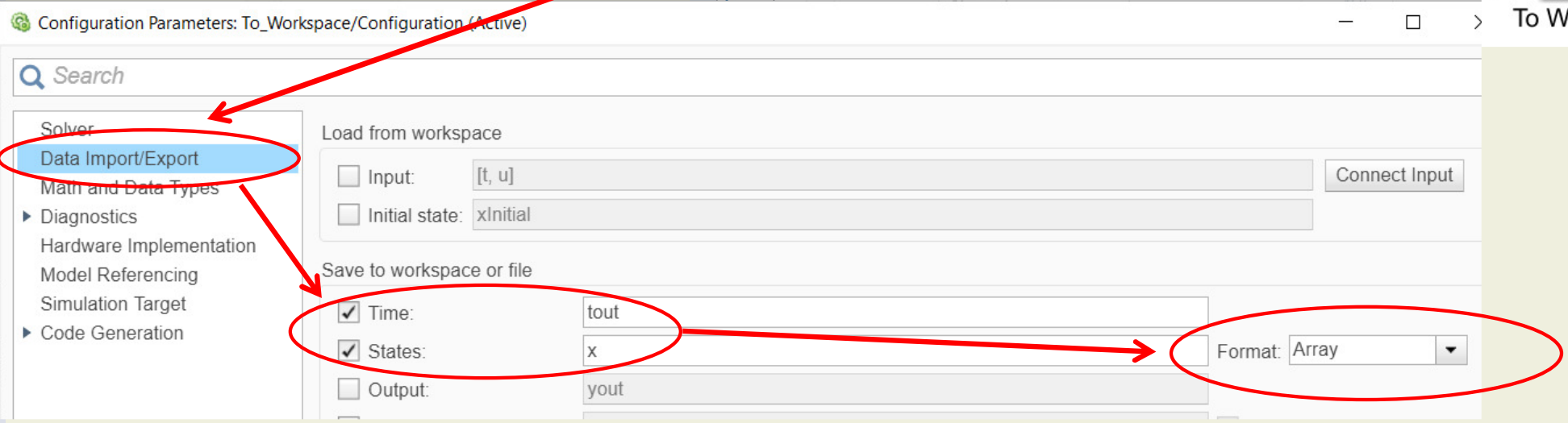
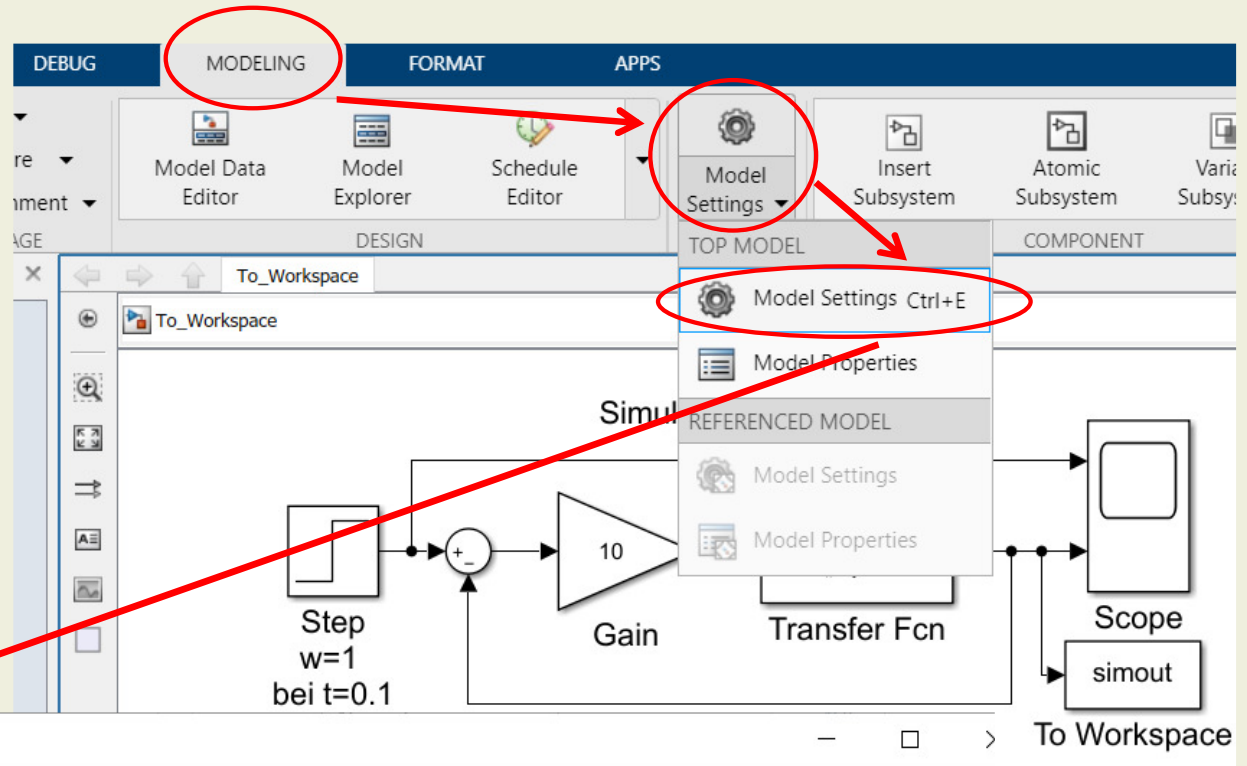
Wird z.B. im Command Window einen Wert $In1 = 0$ eingegeben, wird dieser Wert über Workspace vom Simulink übernommen.



Die Rückgabe von Simulink zu Workspace erfolgt mit dem „To Workspace“-Baustein, z.B. mit „simout“ im Bild links, der mit dem *Data Save-Format* „Array“ konfiguriert werden soll.

Nach jedem RUN wird die Simulationszeit automatisch (auch ohne „To Workspace“) als Vektor „tout“ an Workspace übergeben.

1.2 MATLAB-Konfiguration „Save from workspace or file“:



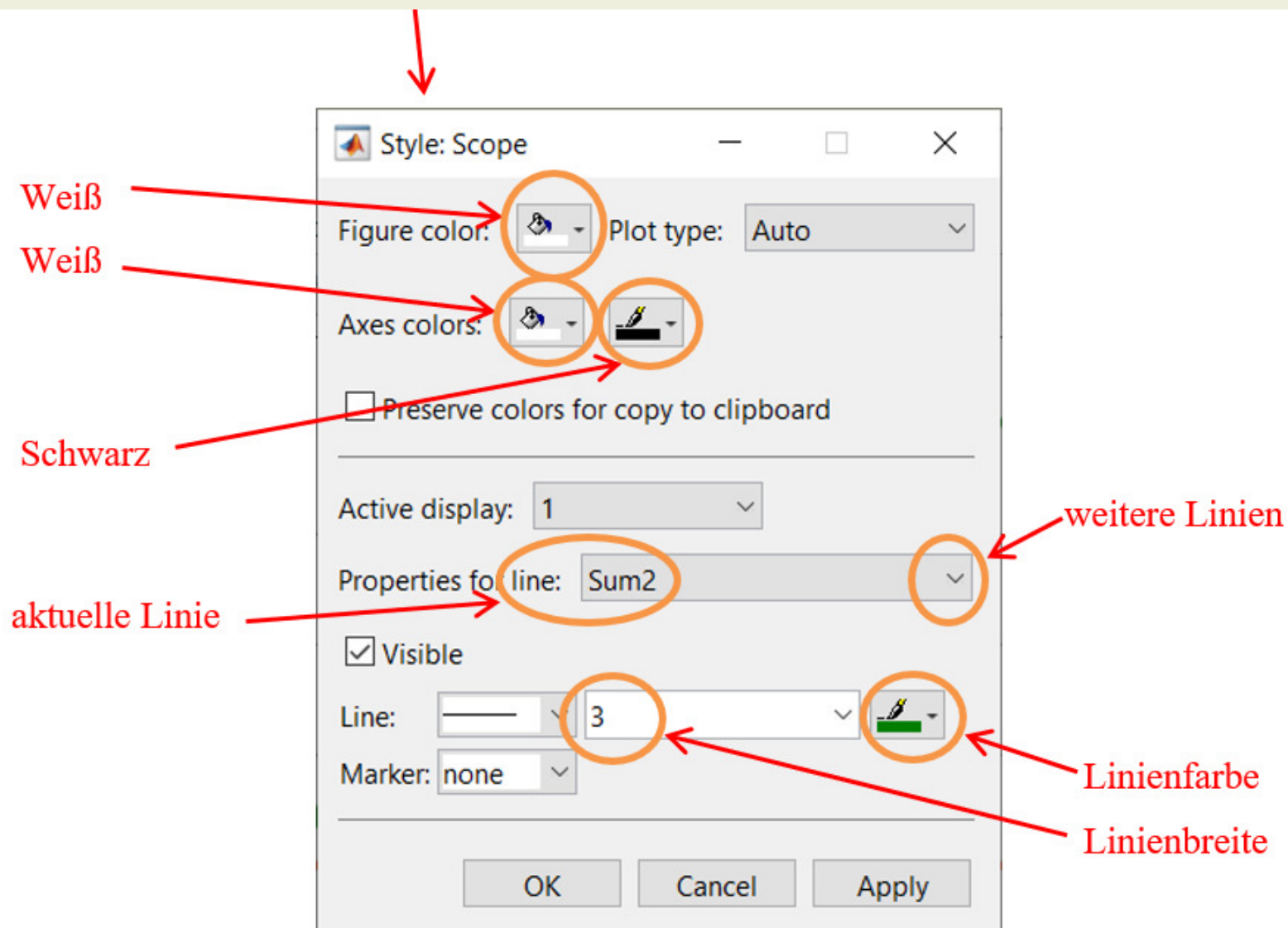
1.3 Scope-Konfiguration

1. Doppelklick

2. Mausklick

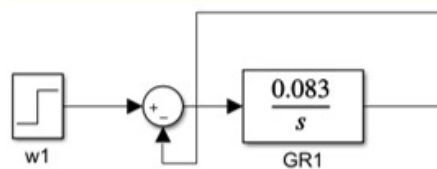
3. Mausklick

The image shows a Simulink Scope window with a plot of three signals over 60 seconds. The y-axis ranges from -1 to 1.5. The x-axis ranges from 0 to 60. The signals are: a green curve starting at 0 and rising to 1.5; an orange curve starting at 0, staying at 0 until t=40, then rising to 1.0; and a blue curve starting at 0, staying at 0 until t=20, then falling to -1.0. The Scope window has a menu bar (File, Edit, View, Simulation, Help) and a toolbar. The 'Scope' icon in the toolbar is circled in orange and labeled '3. Mausklick'. The 'Scope' menu item is circled in orange and labeled '2. Mausklick'. The 'Scope' block in the workspace is circled in orange and labeled '1. Doppelklick'.



mit *Apply* bestätigen und mit *OK* beenden

1.4 sum-Konfiguration



Block Parameters: Sum5

Sum

Add or subtract inputs. Specify one of the following:

- a) character vector containing + or - for each input between ports (e.g. ++|-|++)
- b) scalar, >= 1, specifies the number of input ports. When there is only one input port, add or subtract dimensions or one specified dimension

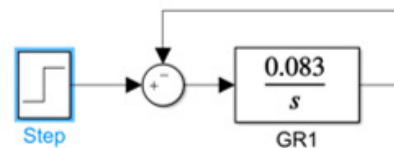
Main | Signal Attributes

Icon shape: round

List of signs:

|+-

OK Cancel



Block Parameters: Sum5

Sum

Add or subtract inputs. Specify one of the following:

- a) character vector containing + or - for each input between ports (e.g. ++|-|++)
- b) scalar, >= 1, specifies the number of input ports. When there is only one input port, add or subtract dimensions or one specified dimension

Main | Signal Attributes

Icon shape: round

List of signs:

+ - |

OK Cancel



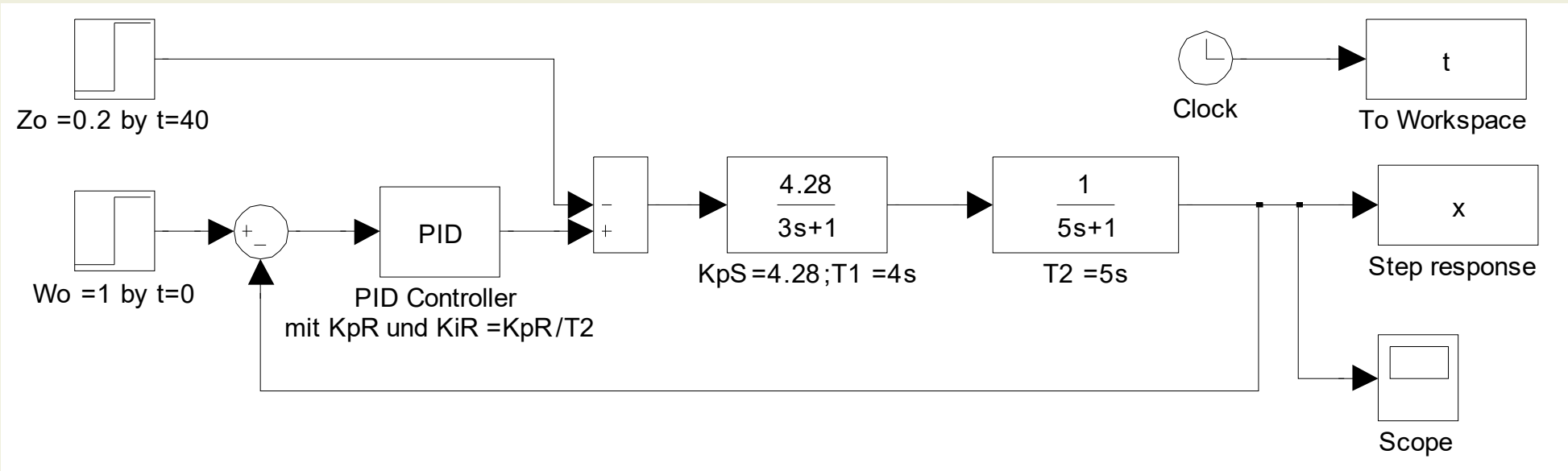
1.4 Datenaustausch zwischen MATLAB und EXCEL

1.4.1 MATLAB als Client

Step 1. Erstellen Sie die Datei **Mappe_Y.xls**

Step 2. Erstellen Sie einen Regelkreis mit dem PI-Regler, Datei **PI_1.mdl**

	A	B	C	D
1	300	30	3	
2	100	10	1	
3				



Step 3. Verbinden Sie die Datei **PI_1.mdl** mit **Mappe_Y.xls** über DDE mit

Kanal=ddeinit('Excel','Mappe_Y.xls')

Es erscheint: Kanal = 4.7836e-299

Step 4. Geben Sie einen Wert K_{pR} aus der EXCEL-Tabelle ein, z.B.

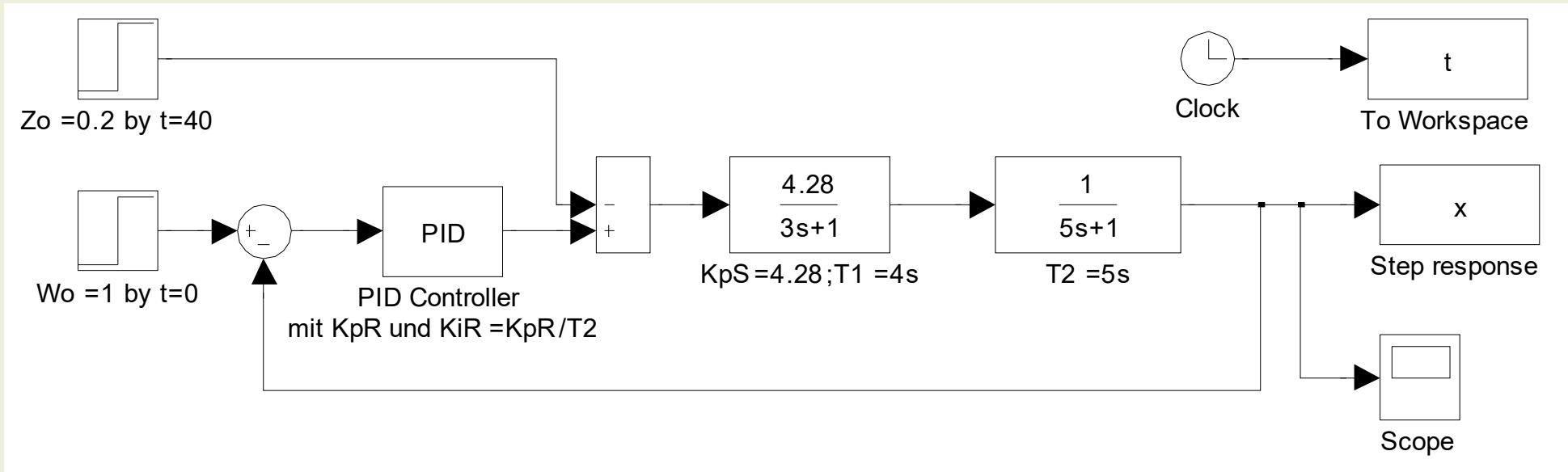
$K_{pR}=\text{ddereq}(\text{Kanal}, 'z1')$

Step 5. Simulieren Sie die Sprungantwort, ggf. nehmen Sie einen anderen Wert K_{pR} aus der Tabelle.

1.4 Datenaustausch zwischen MATLAB und EXCEL

1.4.2 MATLAB als Server

Step 1. Geben Sie **KpR=0.3** und simulieren Sie den Regelkreis.



Step 2. Verbinden Sie die Datei `PI_1.mdl` mit `Mappe_Y.xls`

`Kanal=ddeinit('Excel','Mappe_Y.xls')`

Es erscheint: `Kanal = 4.7836e-299`

	A	B	C	D
1	300	30	3	
2	100	10	1	
3				

Step 3. Senden Sie den Wert K_{pR} von MATLAB zu EXCEL

`rc=ddepoke(Kanal,'z1s4',KpR)`

Bei Erfolg ist der Rückgabewert **rc=1**, sonst **rc=0**

	A	B	C	D
1	300	30	3	300
2	100	10	1	
3				

2 Aufgaben: Standardregler

2.1 Regelkreis mit dem P-Regler

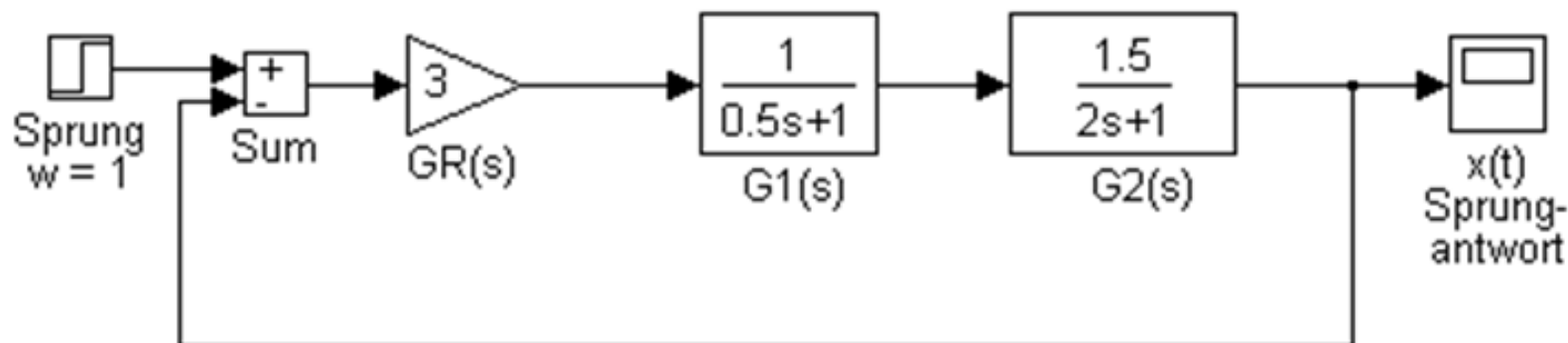
Gegeben ist die Strecke, bestehend aus zwei P-T1-Gliedern mit dem Proportionalbeiwert $K_{ps} = 1.5$.

Die Zeitkonstanten sind: $T_1 = 0.5$ und $T_2 = 2$.

Der P-Regler ist mit $K_{pR} = 3$ eingestellt.

Simulieren Sie den gegebenen Regelkreis. Wie groß ist die bleibende Regeldifferenz $e(\infty)$ nach dem Sprung $w = 1$? Wie ändert sich die bleibende Regeldifferenz, wenn K_{pR} verdoppelt wird, d.h. $K_{pR} = 6$?

Lösung: Quelle [1], Seite 438



2.2 Regelkreis mit dem PI-Regler als Summe von P- und I-Bausteinen

Die Strecke besteht aus zwei P-T1-Gliedern und einem Totzeitglied

$$K_{ps} = 0.8 \quad T_1 = 5 \quad T_2 = 6 \quad T_t = 2$$

Bestimmen Sie die Reglereinstellung nach Ziegler-Nichols-Verfahren.

Dafür stellen Sie zuerst den Regler als P-Regler, d.h. $K_{pR} = 10 \quad T_n = \infty$

Bestimmen Sie die kritischen Werte $K_{PR_krit} = ? \quad T_{n_krit} = ?$

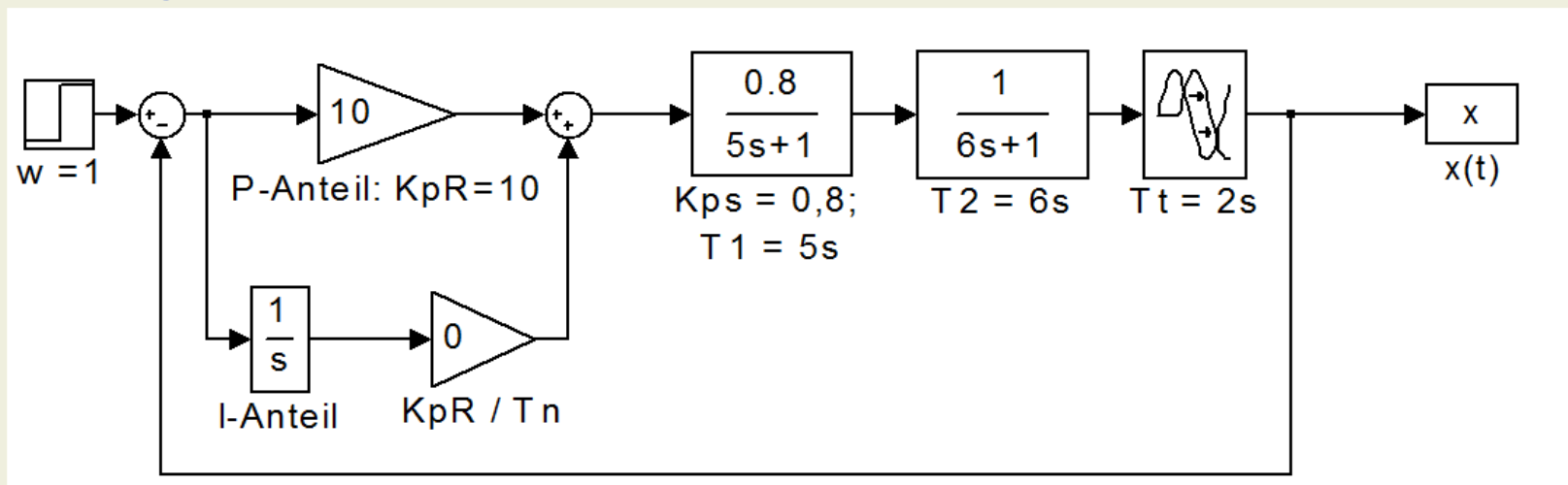
Dann stellen Sie den PI-Regler nach Zi-Ni-Tabelle ein:

$$K_{PR} = 0,4 K_{pR_krit} =$$

$$T_n = 0,85 T_{n_krit} =$$

Wie groß wird dabei die Dämpfung?

Lösung: Quelle [1], Seite 231



2.3 Regelkreis mit dem PI-Regler (PID-Baustein)

Gegeben ist die Strecke, bestehend aus zwei P-T1-Gliedern

$$K_{ps} = 0.5 \quad T_1 = 20 \text{ und } T_2 = 1.$$

Die Kennwerte des PI-Reglers: $K_{PR} = 7.5$ $T_n = 1$

Simulieren Sie den Regelkreis.

Beachten Sie die Konfigurierung des PID-Bausteins:

$$P = K_{PR} = 7.5$$

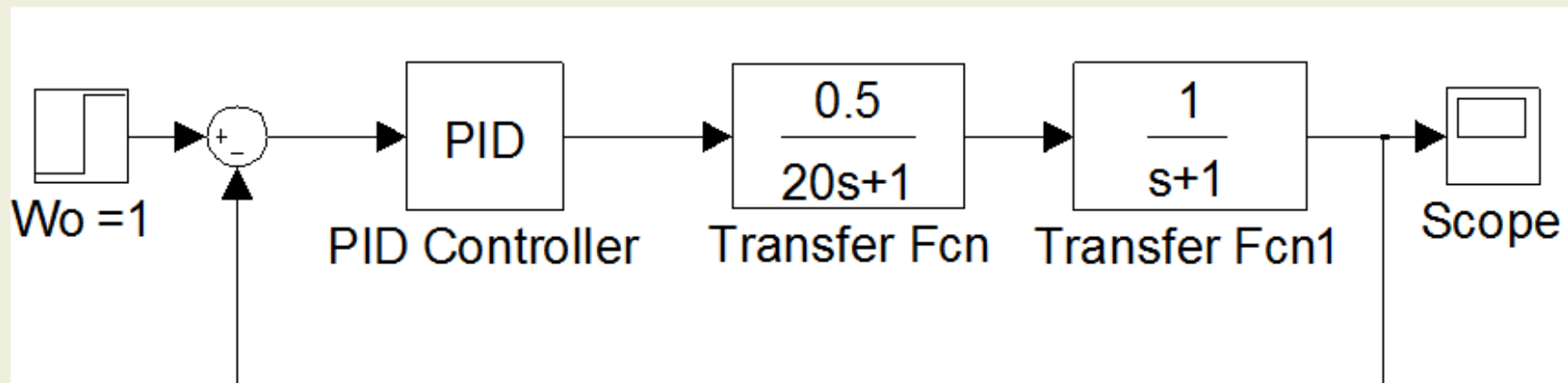
$$I = K_{PR} / T_n = 7.5 / 1 = 7.5$$

$$D = 0$$

Welche Dämpfung hat der Regelkreis?

Nun sind $K_{PR} = 7.5$ und $T_n = 1$. Wie ändert sich die Dämpfung?

Lösung: Quelle [3], Seite 90



2.4 Baustein „To-Workspace“

Erstellen Sie den unten gezeigten Regelkreis (Simulationsdauer 30 s).
Beachten Sie die Konfigurierung des „To_Workspace“-Bausteins:

Variable: x

Save Format: *Array*

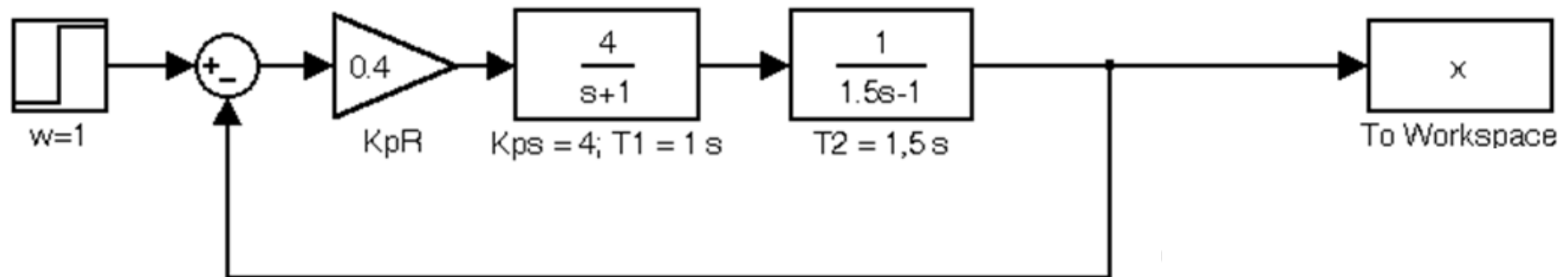
- a) Simulieren Sie die Sprungantwort und geben Sie danach im MATLAB-Command-Window folgende Befehle:

```
plot(tout, x)
```

```
grid % Netzgitter wird ausgegeben
```

```
hold on % Die aktuelle Kurve wird in Figure gehalten
```

Lösung: Quelle [2], Seite 135



2 Aufgaben: Standardregler

2.4 Baustein „To-Workspace“ (Fortsetzung)

b) Ändern Sie K_{pR} von 0.4 auf $K_{pR} = 0.8$ und simulieren Sie die Sprungantwort erneut.

Dann wiederholen Sie den Befehl `plot` um die neue Kurve mit Farbe *rot* anzuzeigen:

```
plot(t, x, `r` )
```

c) Geben Sie nun $K_{pR} = 0.2$ ein und simulieren Sie die Sprungantwort erneut.

Dann wiederholen Sie den Befehl `plot`, um die neue Kurve mit Farbe *grün* anzuzeigen:

```
plot(t, x, `g` )
```

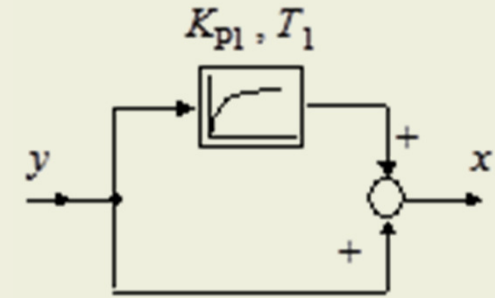
Wie ändert sich die Sprungantwort bei verschiedenen Werten von K_{pR} ?

a) $K_{pR} = 0.4$	Dämpfung $\vartheta =$	stabil	instabil
b) $K_{pR} = 0.8$	$\vartheta =$	stabil	instabil
c) $K_{pR} = 0.2$	$\vartheta =$	stabil	instabil

2.5 Parallelschaltung

Gegeben ist eine Parallelschaltung (Bild rechts) mit $K_{p1}=3$ und $T_1=8$.

Bestimmen Sie die Übertragungsfunktion der Gesamtstrecke mit Eingang y und Ausgang x !



Lösung: Quelle [2], Seite 16

$$G_S(s) = G_{S1}(s) + G_{S2}(s)$$

$$G_{S1}(s) = \frac{K_{P1}}{1 + sT_1} \quad G_{S2}(s) = 1$$

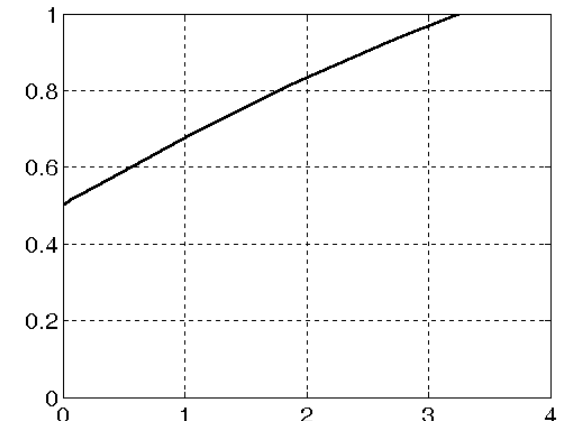
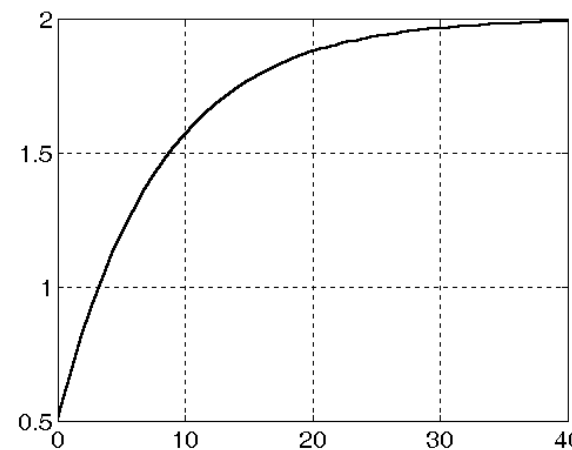
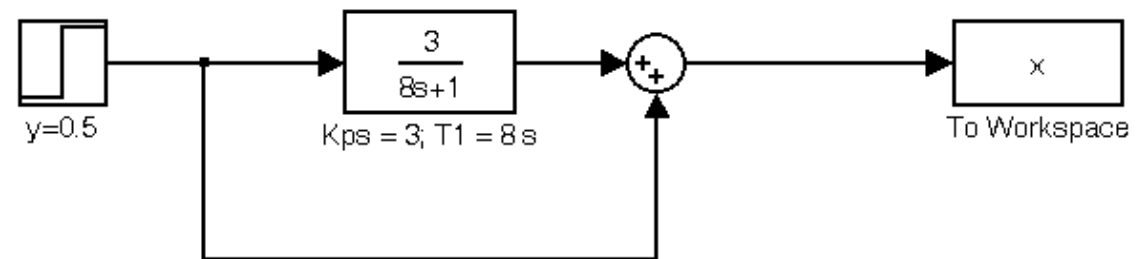
$$G_S(s) = \frac{(1 + K_{P1}) \cdot \left(1 + s \frac{T_1}{1 + K_{P1}} \right)}{1 + sT_1}$$

$$G_S(s) = \frac{K_{PS} (1 + sT_v)}{1 + sT_1}$$

$$K_{PS} = 1 + K_{P1} = 1 + 3 = 4$$

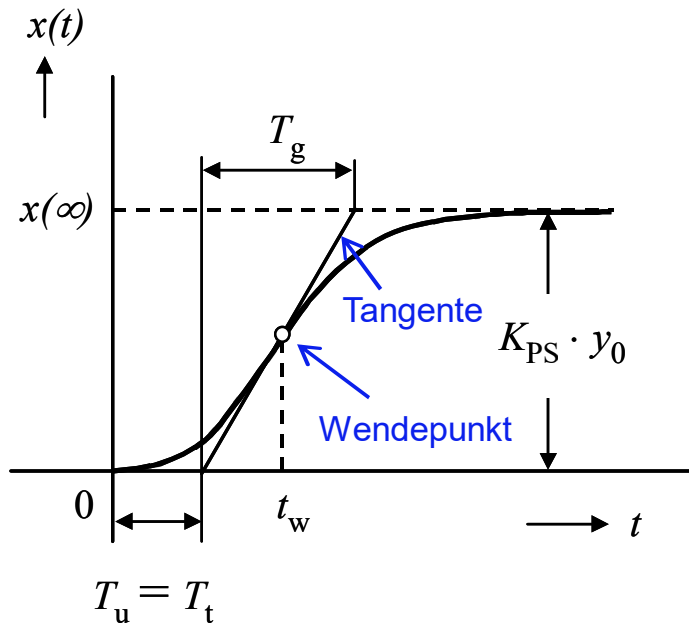
$$T_v = \frac{T_1}{1 + K_{P1}} = 2 \text{ s}$$

Lösung: Quelle [2], Seite 129



3 Identifikation

3.1 Wendetangentenmethode



- a) Die Strecke wird wie ein P-T1-Glied mit der Zeitkonstante T_g und ein Totzeitglied mit der Zeitkonstante T_u vereinfachend beschrieben:

$$G_S(s) = \frac{K_{PS}}{1 + sT_g} \cdot e^{-sT_u}$$

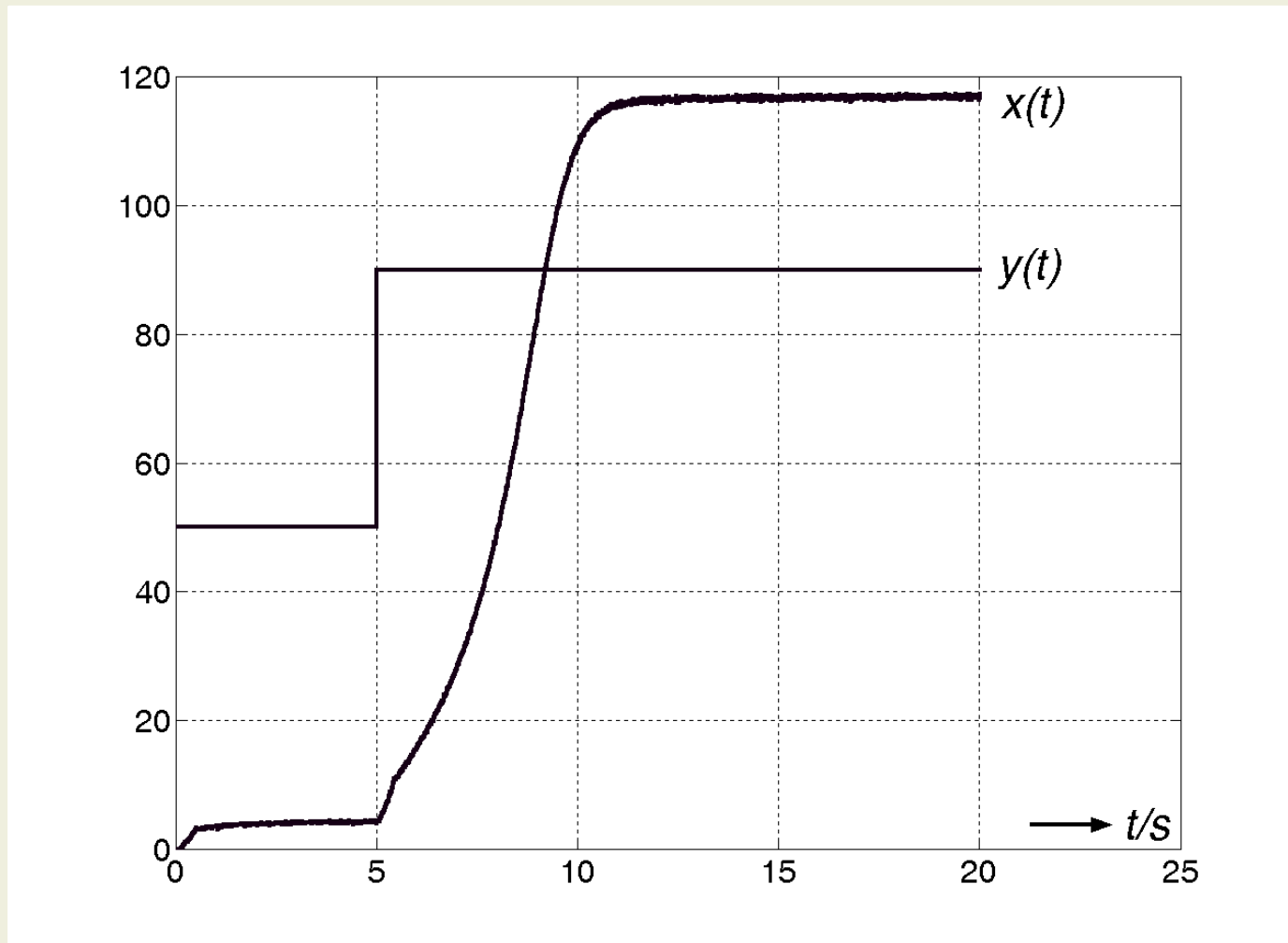
- b) Das Totzeitglied mit der Zeitkonstante T_u wird durch ein P-T1-Glied mit der Zeitkonstante T_u versetzt:

$$G_S(s) = \frac{K_{PS}}{1 + sT_g} \cdot e^{-sT_u} \approx \frac{K_{PS}}{(1 + sT_g)(1 + sT_u)}$$

3.2 Aufgabe: Wendetangentenmethode

Gegeben sind der Eingangssprung und die Sprungantwort einer Strecke (Bild unten).

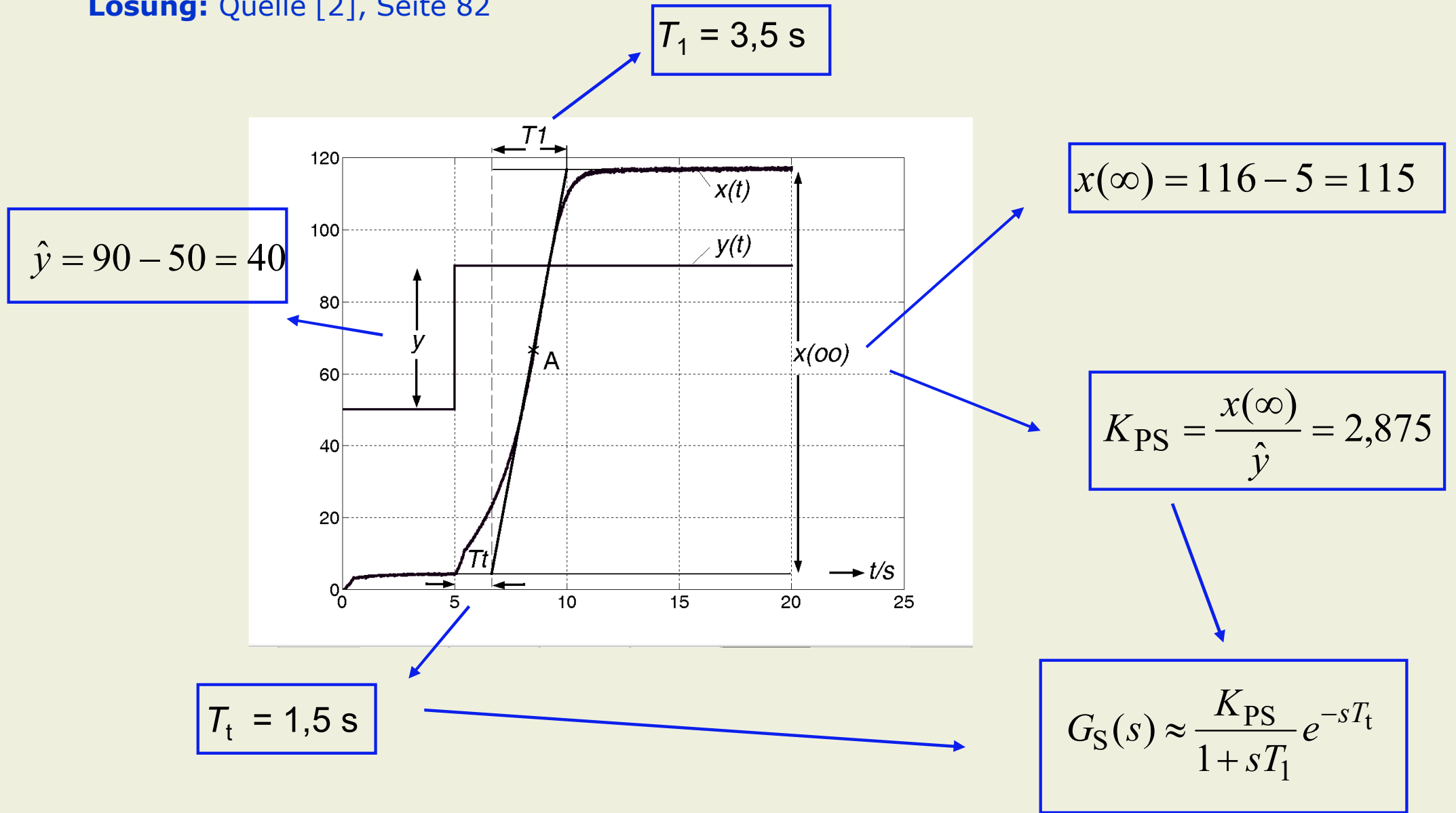
Bestimmen Sie Parameter der Strecke!



3 Aufgaben: Identifikation

3.2 Aufgabe: Wendetangentenmethode

Lösung: Quelle [2], Seite 82



3.3 Zeit-Prozentkennwert-Verfahren

Quelle [1], Seite 229

Die Sprungantwort einer Strecke ist unten im Bild gegeben. Es werden Zeitpunkte t_{10} , t_{50} und t_{90} bestimmt, bei denen $x(t)$

$x(t)$ entsprechend 10%, 50% und 90% des Endwertes erreicht.

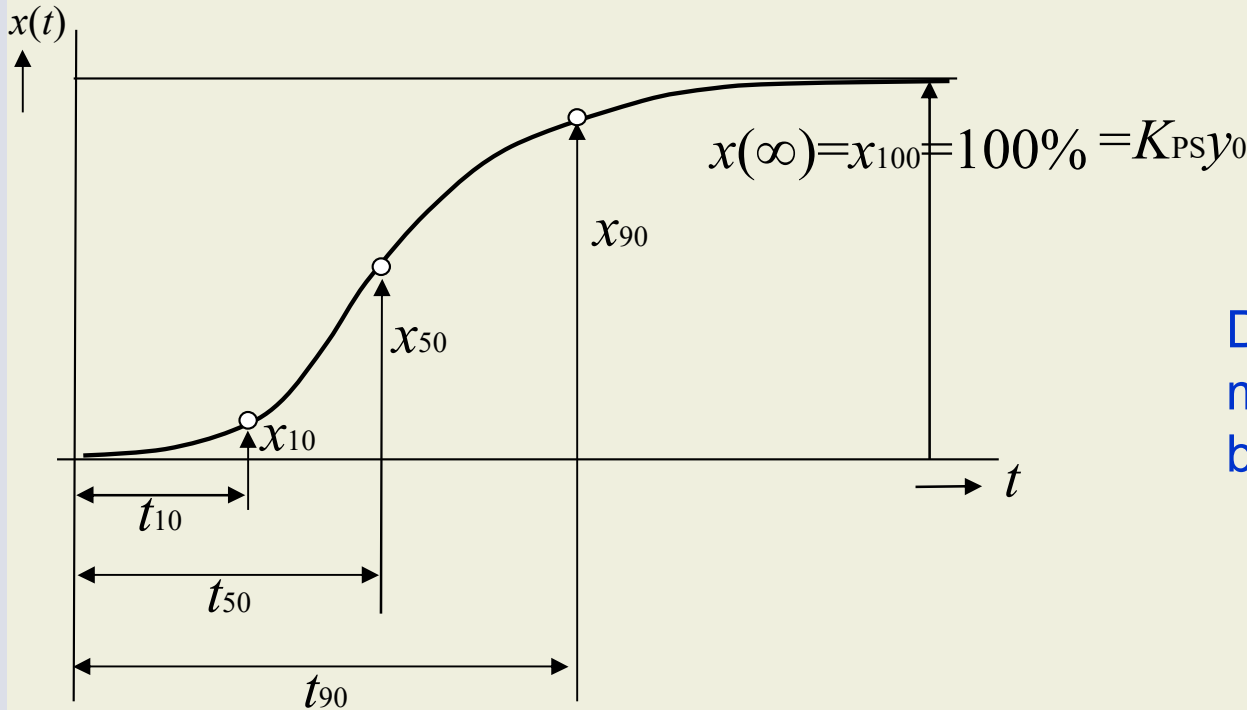
Die Regelstrecke wird wie ein P-Tn-Glied mit n gleichen Zeitkonstanten T beschrieben:

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

Es wird die Kennzahl μ berechnet:

$$\mu = \frac{t_{10}}{t_{90}}$$

Nach der Kennzahl μ wird die Ordnung der Strecke n aus der Tabelle bestimmt:



Ordnung n	μ	τ_{10}	τ_{50}	τ_{90}
1	0,045757	0.105361	0.693147	2.302585
2	0,136722	0.531812	1.678347	3.889720
3	0,207065	1.102065	2.674060	5.322320
4	0,261162	1.744770	3.672061	6.680783
5	0,304318	2.432591	4.670909	7.993590
6	0,339839	3.151898	5.670171	9.274674

3 Aufgaben: Identifikation

3.3 Zeit-Prozentkennwert-Verfahren

Danach werden aus der Tabelle die Werte τ_{10} , τ_{50} und τ_{90} übernommen und die Zeitkonstante T wird berechnet:

$$T = \frac{1}{3} \cdot \left(\frac{t_{10}}{\tau_{10}} + \frac{t_{50}}{\tau_{50}} + \frac{t_{90}}{\tau_{90}} \right)$$

Beispiel:

$$t_{10} = 1,0$$

$$t_{50} = 3,6$$

$$t_{90} = 9,8$$

$$\mu = \frac{t_{10}}{t_{90}} = \frac{1,0}{15,5} = 0,0645$$

Ordnung n	μ	τ_{10}	τ_{50}	τ_{90}
1	0,045757	0.105361	0.693147	2.302585
2	0,136722	0.531812	1.678347	3.889720
3	0,207065	1.102065	2.674060	5.322320
4	0,261162	1.744770	3.672061	6.680783
5	0,304318	2.432591	4.670909	7.993590
6	0,339839	3.151898	5.670171	9.274674

$n = 1$

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

$$T = \frac{1}{3} \cdot \left(\frac{t_{10}}{\tau_{10}} + \frac{t_{50}}{\tau_{50}} + \frac{t_{90}}{\tau_{90}} \right) = \frac{1}{3} \left(\frac{1,0}{0,105} + \frac{3,6}{0,693} + \frac{9,8}{2,302} \right) = 6,3$$

3.4 Aufgabe: Messwerte für Identifikation

Die Messwerte einer Strecke (Sprung und Sprungantwort) sind gegeben:

<http://www.szacher.de/MeineDatei/33247.txt>

t - Zeit

Y – Eingang bzw. Stellgröße X – Ausgang bzw. Regelgröße

Link zu Hinweisen: Passwort ddd

[http://www.szacher.de/BA12/Hinweise Identifikation.pdf](http://www.szacher.de/BA12/Hinweise_Identifikation.pdf)

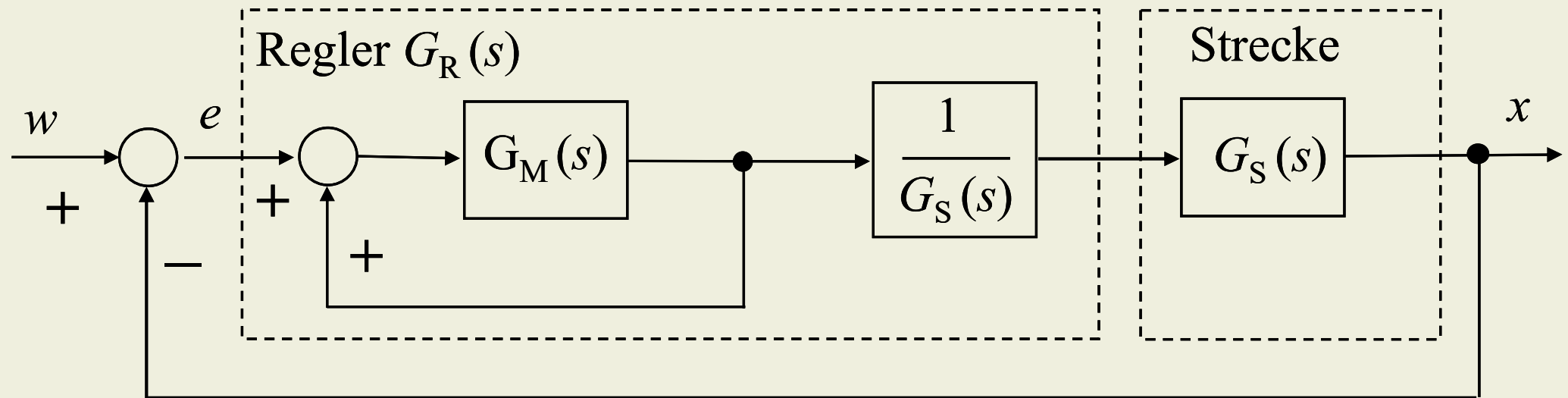
- a) Importieren Sie die Messwerte in MATLAB
- b) Erstellen Sie daraus die Sprungantwort
- c) Bestimmen Sie die Übertragungsfunktion und die Parameter der Strecke:
 - nach Wendetangenten-Verfahren
 - nach Zeit-Prozentkennwerten-Verfahren
- d) Simulieren Sie das Strecken-Modell mit Simulink und vergleichen Sie die Sprungantwort des Modells mit Messwerten

4 Aufgaben: Modellbasierte Regler

4.1 Klassischer Kompensationsregler

Sind die Übertragungsfunktion der Strecke $G_S(s)$ und die gewünschte Übertragungsfunktion $G_M(s)$ des geschlossenen Regelkreises gegeben, wird der Regler nach folgender Übertragungsfunktion eingestellt:

$$G_R(s) = \frac{1}{G_S(s)} \cdot \frac{G_M(s)}{1 - G_M(s)}$$



Quelle [2], Seite 101

4 Aufgaben: Modellbasierte Regler

Aufgabe:

Gegeben sind die Übertragungsfunktion $G_S(s)$ der Strecke und die gewünschte Übertragungsfunktion $G_M(s)$ des geschlossenen Kreises :

$$G_S(s) = \frac{0,5}{(1 + 1,5s)(1 + 0,3s)}$$

$$G_M(s) = \frac{1}{0,02s^2 + 0,2s + 1}$$

Es soll dafür der Kompensationsregler eingesetzt und simuliert werden.

Lösung: Quelle [3], Seite 108

$$G_R(s) = \frac{1}{G_S(s)} \cdot \frac{G_M(s)}{1 - G_M(s)} \rightarrow \frac{G_M(s)}{1 - G_M(s)} = \frac{1}{0,02s^2 + 0,2s} = \frac{5}{s(1 + 0,1s)}$$
$$\frac{1}{G_S(s)} = \frac{(1 + 1,5s)(1 + 0,3s)}{0,5} = 0,9s^2 + 3,6s + 2$$

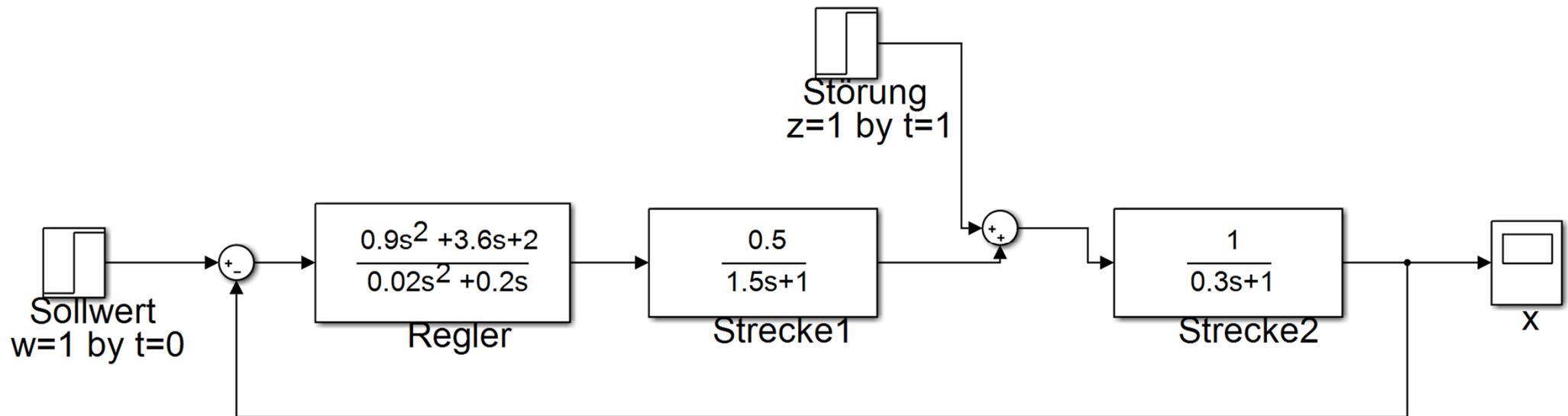
4.1 Klassischer Kompensationsregler

Die Übertragungsfunktion des Kompensationsreglers ist somit:

$$G_R(s) = \frac{1}{G_S(s)} \cdot \frac{G_M(s)}{1 - G_M(s)} = \frac{0,9s^2 + 3,6s + 2}{0,02s^2 + 0,2s}$$

Die Regelung erfolgt nach dem Betragsoptimum, Grundtyp „A“,
Ausregelzeit $T_{\text{aus}} = 1 \text{ s}$.

Soll ein PI-Regler angewendet werden, wird die Ausregelzeit
 $T_{\text{aus}} = 0,3 * 11 = 3,3 \text{ s}$



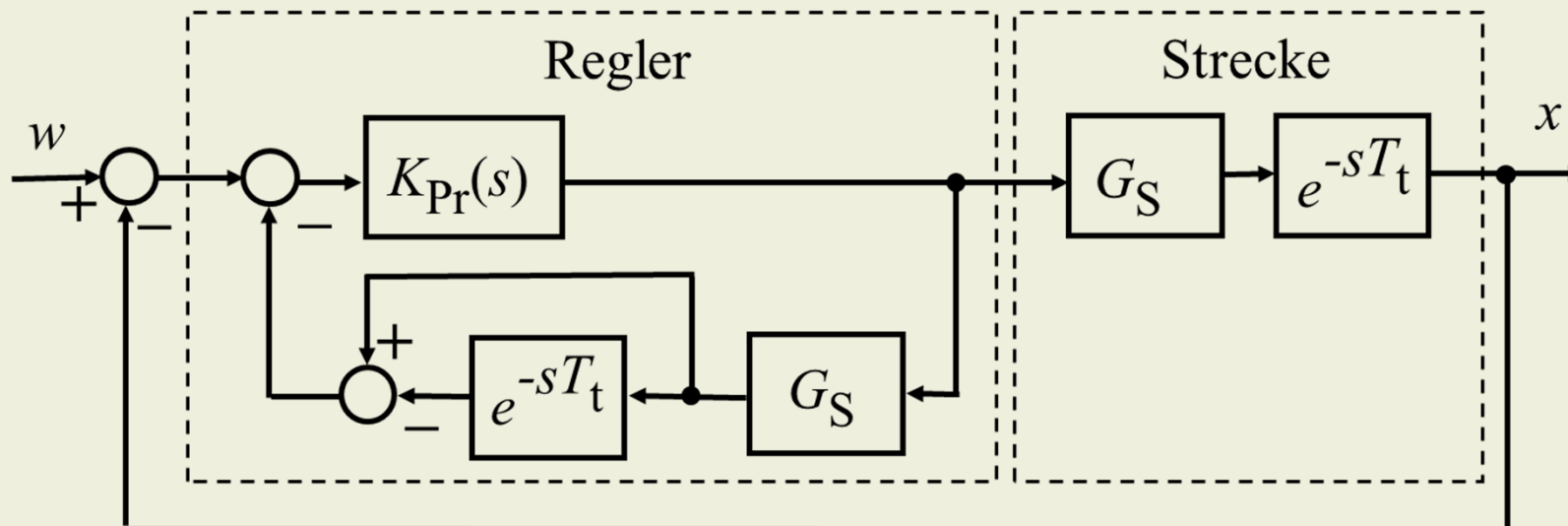
4.2 Smith-Prädiktor

Quelle [1], Seite 360

Für eine industrielle Regelstrecke mit Totzeit ist das Kompensationsprinzip nicht anwendbar, weil die reziproke Übertragungsfunktion der Regelstrecke mit der positiven Potenz vorkommt:

$$G_S(s)e^{-sT_t} = \frac{K_{PS}}{(1+sT_1)(1+sT_2)} e^{-sT_t} \longrightarrow \frac{1}{G_S(s)e^{-sT_t}} = \frac{(1+sT_1)(1+sT_2)}{K_{PS}} e^{sT_t}$$

In diesem Fall kann der *Smith-Prädiktor* angewendet werden, wobei K_{pr} der Kompensationsregler für die Teilstrecke $G(s)$ ohne Totzeit ist.



Aufgabe:

Gegeben sind die Übertragungsfunktion $G_S(s)$ der Strecke und die gewünschte Übertragungsfunktion $G_M(s)$ des geschlossenen Kreises:

$$G_S(s)e^{-sT_t} = \frac{0,8}{(1+2s)(1+s)} e^{-3s} \quad G_M(s)e^{-sT_t} = \frac{1}{0,02s^2 + 0,2s + 1} e^{-3s}$$

Es soll dafür der *Smith*-Prädiktor eingesetzt und simuliert werden.

Lösung: Quelle [2], Seite 237

$$K_{Pr}(s) = \frac{1}{G(s)} \cdot \frac{G_M(s)}{1 - G_M(s)} \rightarrow \frac{G_M(s)}{1 - G_M(s)} = \frac{1}{0,02s^2 + 0,2s} = \frac{5}{s(1 + 0,1s)}$$
$$\frac{1}{G_S(s)} = \frac{(1+2s)(1+s)}{0,8} = \frac{2s^2 + 3s + 1}{0,8}$$

4 Aufgaben: Modellbasierte Regler

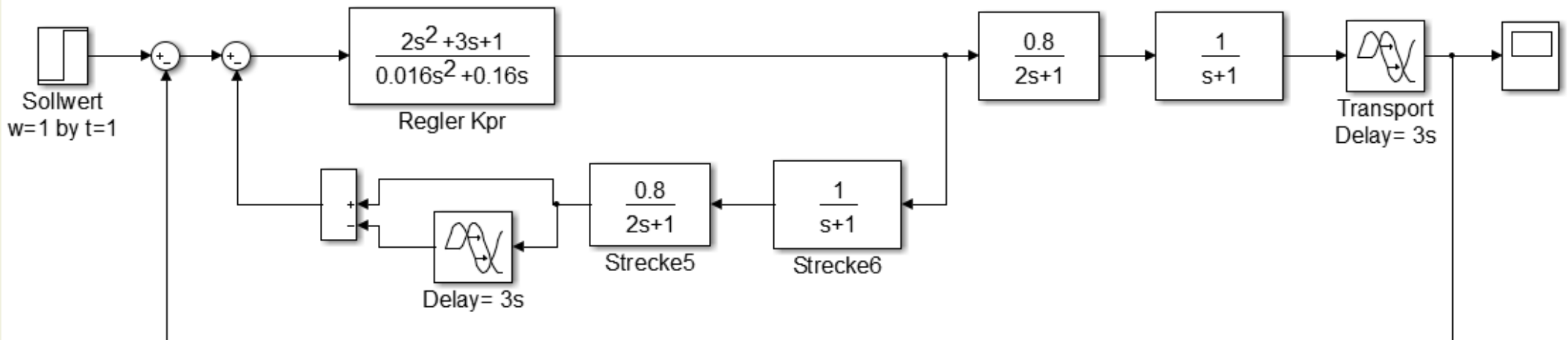
4.2 Smith-Predictor

Die Übertragungsfunktion des Kompensationsreglers ist somit:

$$K_{Pr}(s) = \frac{1}{G(s)} \cdot \frac{G_M(s)}{1 - G_M(s)} = \frac{2s^2 + 3s + 1}{0,8(0,02s^2 + 0,2s)} = \frac{2s^2 + 3s + 1}{0,016s^2 + 0,16s}$$

Simulieren Sie den Regelkreis mit dem *Smith*-Prädiktor und beantworten Sie folgende Fragen:

- Wie groß ist die Ausregelzeit T_{aus} ?
- Wurde das gewünschte Verhalten erreicht?



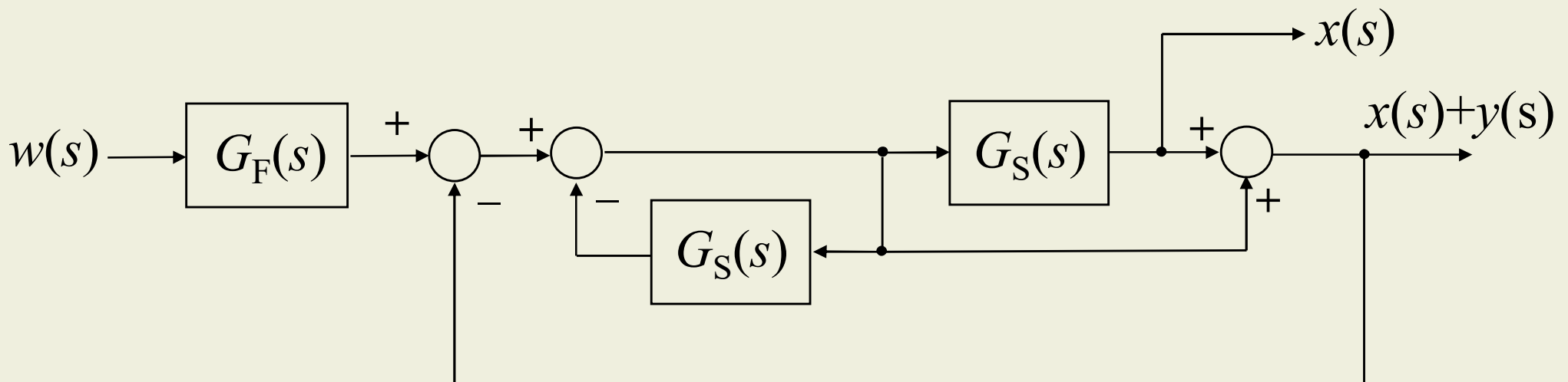
4.3 ASA-Kompensationsregler

Quelle [1], Seite 364

Nach dem von *S.Zacher* entwickelten Antisystem-Approach (ASA) wird anstelle des Reglers eine dynamische Schaltung aus zwei Übertragungsfunktionen $G_S(s)$ der Strecke eingesetzt.

Die Regelung erfolgt nach zwei Regelgrößen: die eigentliche Regelgröße $x(t)$ und die fiktive Größe $[x(t)+y(t)]$, die als Summe von Regelgröße $x(t)$ und Stellgröße $y(t)$ gebildet wird.

Die gewünschte Übertragungsfunktion des geschlossenen Regelkreises $G_M(s)$ wird wie ein Vorfilter $G_F(s)$ zur Bildung des Sollwertes angewendet.

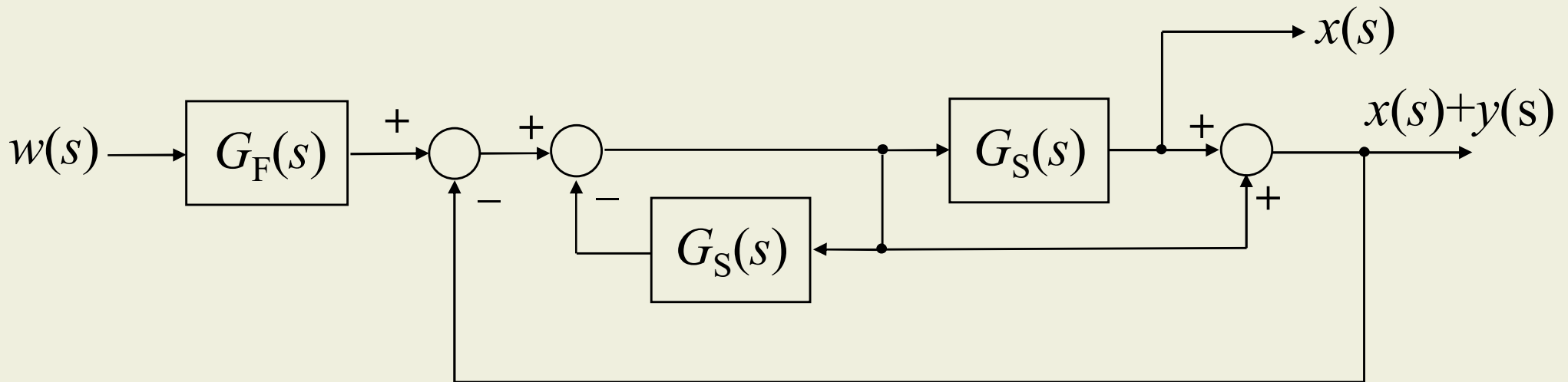


4 Aufgaben: Modellbasierte Regler

4.3 ASA-Kompensationsregler

Es gilt die folgende Einstellregel für die Übertragungsfunktion $G_F(s)$ des Vorfilters, die von der Übertragungsfunktion $G_S(s)$ der Strecke und von der gewünschten Übertragungsfunktion $G_M(s)$ des geschlossenen Kreises definiert wird:

$$G_F(s) = 2G_M(s) \frac{1 + G_S(s)}{G_S(s)}$$



Quelle [1], Seite 364

Aufgabe:

Gegeben sind die Übertragungsfunktion $G_S(s)$ der Strecke und die gewünschte Übertragungsfunktion $G_{M0}(s)$ des offenen Kreises:

$$G_S(s) = \frac{2}{1+s} \quad G_{M0}(s) = \frac{5}{s(0,1s+5)}$$

Die Übertragungsfunktion $G_{M0}(s)$ ist so gewählt, dass die Regelung nach dem Betragsoptimum, Grundtyp „A“ erfolgt.

Die gewünschte Ausregelzeit ist $T_{\text{aus}} = 0,1 \cdot 11 = 1,1 \text{ s}$.

Lösung: Quelle [1], Seite 364

Zuerst wird die gewünschte Übertragungsfunktion $G_M(s)$ des geschlossenen Kreises bestimmt:

$$G_M(s) = \frac{G_{M0}(s)}{1 + G_{M0}(s)} = \frac{5}{0,1s^2 + s + 5}$$

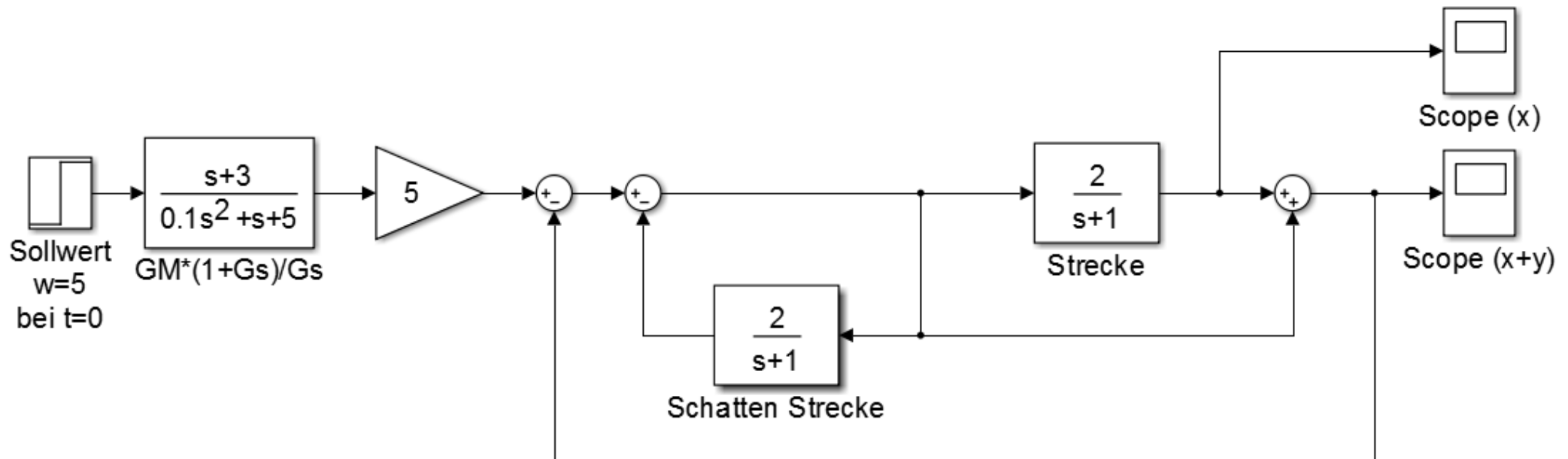
4 Aufgaben: Modellbasierte Regler

4.3 ASA-Kompensationsregler

Danach wird die Übertragungsfunktion $G_F(s)$ des Vorfilters bestimmt:

$$G_F(s) = 2G_M(s) \frac{1 + G_S(s)}{G_S(s)} = 2 \frac{5}{0,1s^2 + s + 5} \cdot \frac{s + 3}{2} = 5 \cdot \frac{s + 3}{0,1s^2 + s + 1}$$

Simulieren Sie den Regelkreis und prüfen Sie, ob das gewünschte Verhalten nach dem Betragsoptimum erreicht wurde.



4.4 PFC-Regler

Die Grundidee des PFC-Verfahrens (*Predictive Functional Control*) besteht darin, dass die Sprungantwort einer Regelstrecke an die Sprungantwort eines Modells bzw. an eine gewünschte Sprungantwort während des Ablaufs angepasst wird. Es wird angenommen, dass ein internes dynamisches Modell des zu optimierenden Prozesses vorliegt und Referenzverfahren liefert. Der Regelkreis wird digitalisiert, die Stellgröße wird anhand der hergeleiteten Rekursionen berechnet.

Das Verfahren wurde in den 70er Jahren von *Jacques Richalet* entwickelt und in folgenden Publikationen veröffentlicht:

- [1] Richalet, J.; Lavielle, G.; Mallet, J.: *La commande prédictive - Mise en oeuvre et applications industrielles*, Editions Eyrolles, 2004
- [2] Richalet, J.; O'Donovan, D.: *Predictive Functional Control : Principles and Industrial Applications (Advances in Industrial Control)*, Springer-Verlag, 2009, ISBN 9781848824928
- [3] Hamm, T.: *Umsetzung eines Modell-prädiktiven Regel-Konzeptes in einer prozessnahen Komponente eines Prozessleitsystems*. Diplomarbeit der Fachhochschule Wiesbaden, 2006, (Betreuer: Prof. Dr. Serge Zacher und Dr. Martin Altebockwinkel).

Der Modellausgang $x_M(t)$ wird für die gesamte Ausregelzeit T_{aus} berechnet. Bei $t = T_{\text{aus}}$ erreicht der Modellausgang $x_M(t)$ 95% des Wertes im Beharrungszustand. Daraus werden die Regeldifferenzen e_{Mk} und e_{Mk+h} zum Beginn t_k und zum Ende t_{k+h} jedes vordefinierten Zeitabschnitts T_h (Prädiktionshorizont) bestimmt.

Zu den gleichen Zeitpunkten werden die Regeldifferenzen e_k und e_{k+h} ermittelt, wobei sich die Regeldifferenz e_k zum Beginn des Prädiktionshorizonts auf den tatsächlichen Istwert x_k bezieht. Dagegen wird e_{k+h} zum Ende des Prädiktionshorizonts aus einer Referenztrajektorie $x_{\text{ref}}(t)$ errechnet.

4.4 PFC-Regler

Die Grundidee des PFC-Verfahrens (*Predictive Functional Control*) besteht darin, dass die Sprungantwort einer Regelstrecke an die Sprungantwort eines Modells bzw. an eine gewünschte Sprungantwort während des Ablaufs angepasst wird. Es wird angenommen, dass ein internes dynamisches Modell des zu optimierenden Prozesses vorliegt und Referenzverfahren liefert. Der Regelkreis wird digitalisiert, die Stellgröße wird anhand der hergeleiteten Rekursionen berechnet.

Das Verfahren wurde in den 70er Jahren von *Jacques Richalet* entwickelt und in folgenden Publikationen veröffentlicht:

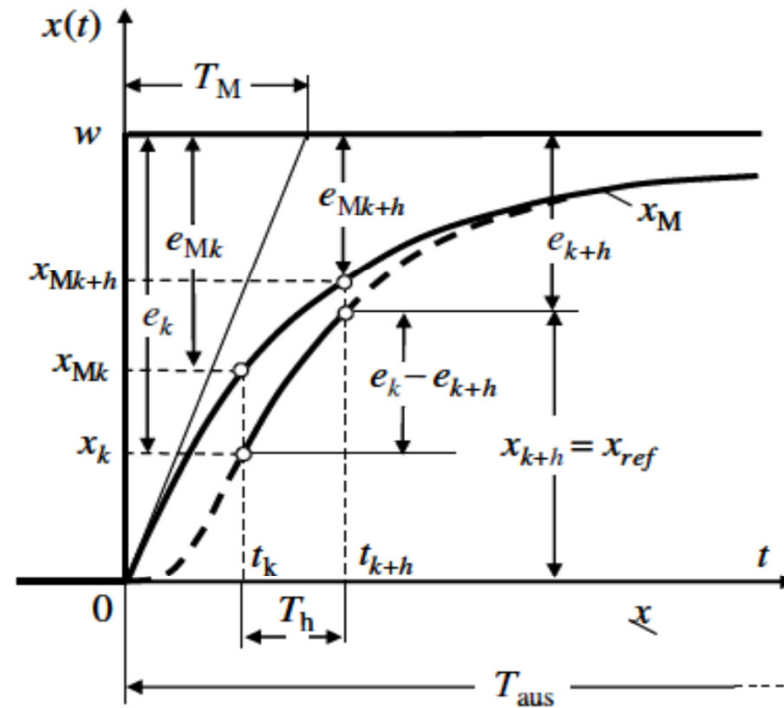
- [1] Richalet, J.; Lavielle, G.; Mallet, J.: *La commande prédictive - Mise en oeuvre et applications industrielles*, Editions Eyrolles, 2004
- [2] Richalet, J.; O'Donovan, D.: *Predictive Functional Control : Principles and Industrial Applications (Advances in Industrial Control)*, Springer-Verlag, 2009, ISBN 9781848824928
- [3] Hamm, T.: *Umsetzung eines Modell-prädiktiven Regel-Konzeptes in einer prozessnahen Komponente eines Prozessleitsystems*. Diplomarbeit der Fachhochschule Wiesbaden, 2006, (Betreuer: Prof. Dr. Serge Zacher und Dr. Martin Altebockwinkel).

Der Modellausgang $x_M(t)$ wird für die gesamte Ausregelzeit T_{aus} berechnet. Bei $t = T_{\text{aus}}$ erreicht der Modellausgang $x_M(t)$ 95% des Wertes im Beharrungszustand. Daraus werden die Regeldifferenzen e_{Mk} und e_{Mk+h} zum Beginn t_k und zum Ende t_{k+h} jedes vordefinierten Zeitabschnitts T_h (Prädiktionshorizont) bestimmt.

Zu den gleichen Zeitpunkten werden die Regeldifferenzen e_k und e_{k+h} ermittelt, wobei sich die Regeldifferenz e_k zum Beginn des Prädiktionshorizonts auf den tatsächlichen Istwert x_k bezieht. Dagegen wird e_{k+h} zum Ende des Prädiktionshorizonts aus einer Referenztrajektorie $x_{\text{ref}}(t)$ errechnet.

4 Modellbasierter Reglerentwurf

4.4 PFC-Regler



Die gewünschte Sprungantwort $x_M(t)$ und die Referenztrajektorie $x_{ref}(t)$

Der Vorteil dieses Verfahrens besteht also darin, dass die Regelgröße nicht nur auf den Sollwert w , sondern auf die Referenztrajektorie $x_{ref}(t)$ geregelt wird. Dadurch soll die Regelung schneller und ohne große Überschwinger erfolgen.

Das Verfahren wird nachfolgend am Beispiel einer P-T₁-Regelstrecke mit der Zeitkonstante T_1 und dem Proportionalbeiwert K_{PS} erklärt.

4 Modellbasierter Reglerentwurf

4.4 PFC-Regler

$$G_P(s) = \frac{K_{PS}}{1 + sT_P}$$

In diesem Fall wird das Modell der Regelstrecke durch die Übertragungsfunktion

$$G_M(s) = \frac{K_{PM}}{1 + sT_M}$$

bzw. durch die Lösung einer inhomogenen DGL der 1. Ordnung mit der Zeitkonstante $T_M = T_P$ und dem Proportionalbeiwert $K_{PS} = K_{PM}$ beschrieben:

$$T_M \frac{dx_M(t)}{dt} + x_M(t) = K_{PM} \hat{y}$$

Die Lösung dieser DGL bei einem Eingangssprung der Stellgröße \hat{y} und einer Anfangsbedingung $x_M(0) = 0$ wird bekanntlich mit folgender Formel ermittelt:

$$x_M(t) = \underbrace{x_M(0)}_{x_e} e^{-\frac{t}{T_M}} + \underbrace{K_{PM}(1 - e^{-\frac{t}{T_M}})}_{x_a} \hat{y}$$

Hier sind: x_a – Lösungsansatz der homogenen DGL, x_e – die Partikulärlösung des mit der Sprungfunktion behafteten Eingangssignals.

Nach dem PFC-Verfahren wird die gesamte Ausregelzeit T_{aus} in n Zeitabschnitten (Prädiktionshorizonten) der Länge T_h aufgeteilt:

$$T_h = \frac{T_{aus}}{n}$$

4 Modellbasierter Reglerentwurf

4.4 PFC-Regler

Bezeichnet man bei gleichen Werten der Zeitkonstanten $T_M = T_P$

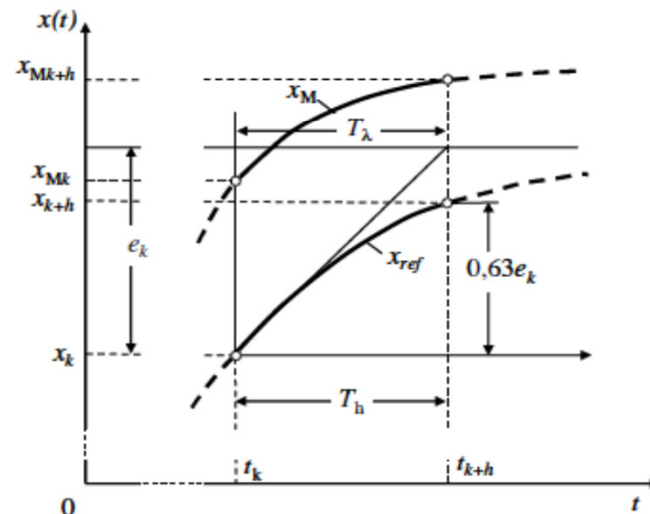
$$e^{-\frac{T_h}{T_P}} = \alpha \quad \text{bzw.} \quad e^{-\frac{T_h}{T_M}} = \alpha,$$

so ergibt sich daraus die folgende rekursive Formel zur Berechnung der Regelgröße des Modells (Modellgleichung):

$$x_{Mk+h} = \alpha \cdot x_{Mk} + (1 - \alpha) \cdot K_{PM} \cdot y_k$$

Nun soll die Reglergleichung erstellt werden. Der modellbasierte Regler soll zum Beginn jedes Prädiktionshorizonts k die Stellgröße y_k so berechnen, dass die aktuelle Sprungantwort x_k an die gewünschte Sprungantwort x_{refk} angepasst wird. In anderen Worten, zum Abschluss jedes vorherigen Prädiktionshorizonts soll die aus der obigen Modellgleichung ausgerechnete Regelgröße x_{Mk} mit dem tatsächlichen Istwert x_k verglichen und daraus eine neue Stellgröße y_k für den nächsten Prädiktionshorizont berechnet werden.

Nachfolgend wird die Herleitung des Regleralgorithmus kurz beschrieben (Bild unten).



4 Modellbasierter Reglerentwurf

4.4 PFC-Regler

Beispiel: (erstellt von Projektgruppe *Sebastian Heim, Alexander Koller, Jürgen Staub*)

Strecke $G_S(s) = \frac{K_{PS}}{1 + sT_1}$ mit $K_{PS} = 2$ und $T_P = 30$ s Dauer: $T_{aus} = 200$ s

Modell $G_M(s) = \frac{K_{PM}}{1 + sT_M}$ mit $K_{PM} = 2$ und $T_M = 25$ s Abtastzeit $T = 0,1$ s

```
clear
clc

ts = 0.1;                               % Abtastzeit
T=200;                                   % Dauer
time=0:ts:(T-ts);                        % Zeitstrahl

%-----Prozess (PT1)-----
Kp=2
Tp=30

%-----Modell-----
Km=2;
Tm=25;

%-----Faktoren für die Differenzgleichung (später aus G(z))
alp=exp(-ts/Tp);
blp=1-alp;

alm=exp(-ts/Tm);
blm=1-alm;
```

4 Modellbasierter Reglerentwurf

4.4 PFC-Regler

```
%-----Reglerparameter-----
CLTR=60 % Wenn Referenztrajektorie (e-Funktion) 95% erreicht hat
h=1; % Prädiktionshorizont
lh=1-exp(-ts*h/(CLTR/3));

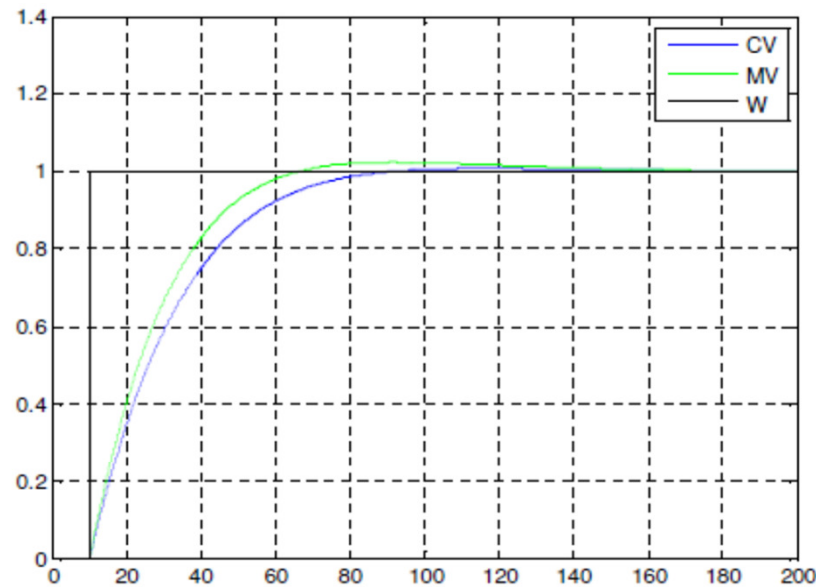
a1mh=a1m^h;
b1mh=1-a1mh;

%-----Variablen-----
MV=zeros(1,T/ts); % Steuern
CV=MV; % ClosedLoop
sm=MV; % Modell Ausgang
OL=MV; % OpenLoop
start=10; % Nach 10 Schritten gehts los
sp =1; % Sprunghöhe
W=[zeros(1,start/ts) sp*ones(1,T/ts-start/ts)]; % Ansteuerung

% W=[zeros(1,start/ts) ts/100:ts/100:sp-ts/100 sp*ones(1,T/ts-start/ts-99)];
for i=10:1:length(time) % erst bei 10 Anfängen damit es Vergangenheitswerte gibt
    OL(i)=OL(i-1)*a1p+b1p*Kp*W(i-1);
    CV(i)=CV(i-1)*a1p+b1p*Kp*MV(i-1);
    sm(i)=sm(i-1)*a1m+b1m*Km*MV(i-1);
    MV(i)=((W(i)-CV(i))*lh+sm(i)*b1mh)/(Km*b1mh);
end
plot(time, CV, 'b', time, sm, 'g', time, W, 'k')
h = legend('CV', 'MV', 'W', 5);
set(h, 'Interpreter', 'none');
grid
```


4 Modellbasierter Reglerentwurf

4.4 PFC-Regler



Bei gleichen Werten von $T_M = T_P$ sind die Sprungantworten des Strecke und des Modells nicht zu unterscheiden. Um die beiden Kurven doch ersichtlich zu machen, sind hier zwei verschiedene Werte $T_P = 30$ s und $T_M = 25$ s gewählt.

Im obigen Beispiel sind folgende PFC-Parametern gesetzt:

$h = 1$ die Länge des Prädiktionshorizonts

$$\alpha_p = e^{-\frac{T_A}{T_P}} = 0,9967 \quad \beta_p = 1 - \alpha_p = 0,0033$$

$$\alpha_M = e^{-\frac{T_A}{T_M}} = 0,9960 \quad \beta_M = 1 - \alpha_M = 0,0040$$

$$\alpha_{Mh} = 0,9960 \quad \beta_{Mh} = 1 - \alpha_{Mh} = 0,0040$$

In [3] wurde gezeigt, dass man durch die entsprechende Wahl des Parameters λ die Anpassung des Istwertes an die Referenztrajektorie optimieren kann. Nimmt man, z. B.

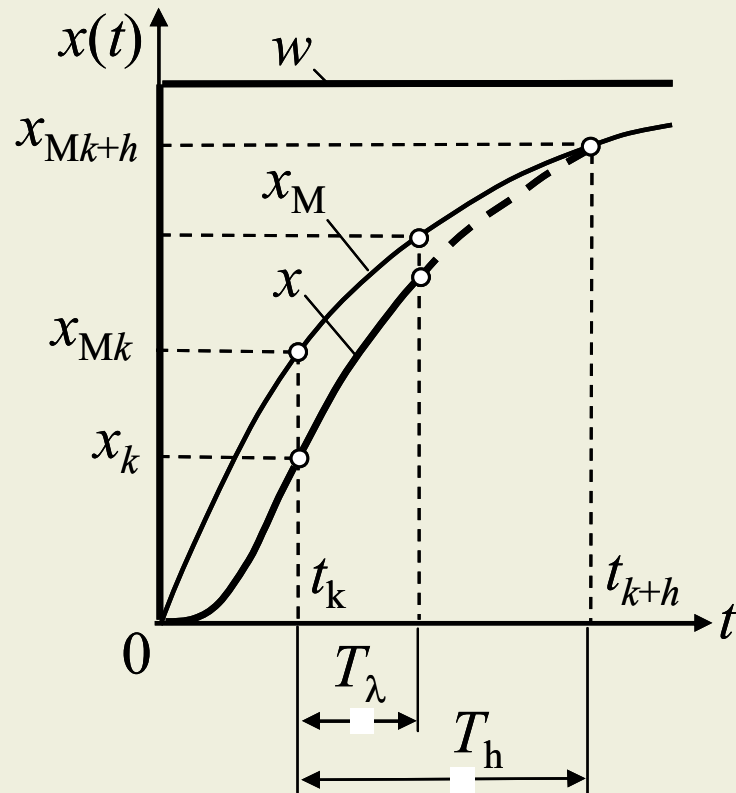
$$T_\lambda = \frac{1}{5} T_h$$

erreicht der Istwert des Kreises schneller den Referenzwert von ca. 95 % des Fehlers e_k , jedoch wird der Übergang zwischen Prädiktionshorizonten nicht stoßfrei erfolgen bzw. die Regelgüte wird verschlechtert.

4.5 SPFC-Regler

Quelle [1], Seite 362

Der PFC-Algorithmus (*Predictive Function Control*) wurde von *Jacques Richalet* entwickelt. Die aktuelle Sprungantwort der Regelstrecke $x(t)$ wird während der Regelung periodisch an die Sprungantwort eines vorher gegebenen dynamischen Modells $x_M(t)$ angepasst.



Die von *S.Zacher* daraus hergeleitete Vereinfachung heißt SPFC-Algorithmus (*Simplified Predictive Function Control*).

Der Original-PFC-Algorithmus

$$y_k = \frac{1}{K_{PM}} \left[x_{Mk} + \frac{1-\lambda}{1-\alpha} (w - x_k) \right]$$

wurde für SPFC unter Annahme $\alpha = \lambda$ zu folgender Form vereinfacht:

$$y_k = \frac{1}{K_{PM}} \left[x_{Mk} + (w - x_k) \right]$$

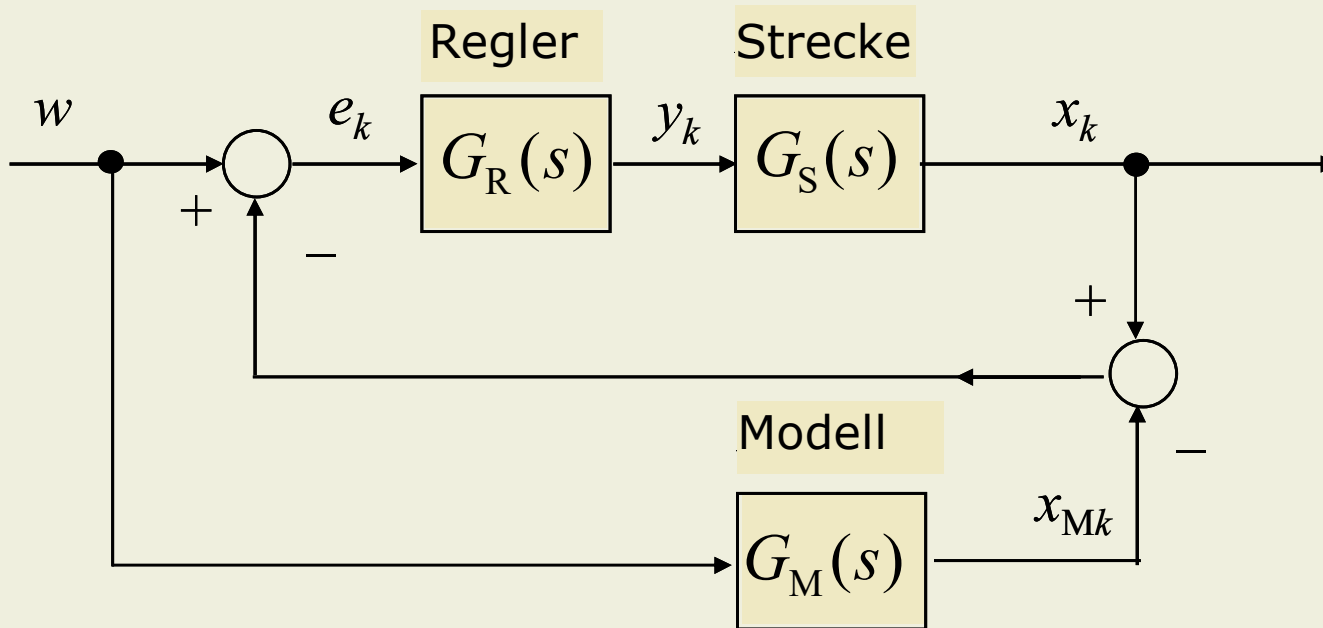
4.5 SPFC-Regler

Quelle [1], Seite 362

Somit kann ein Regelkreis nach dem SPFC-Algorithmus

$$y_k = \frac{1}{K_{PM}} [x_{Mk} + (w - x_k)] = \frac{1}{K_{PM}} [w - (x_k - x_{Mk})]$$

folgendermaßen dargestellt werden (Bild unten). Dann wurde die oben gegebene Gleichung für s-Ebene modifiziert:



$$y_k \Rightarrow y(s)$$

$$x_k \Rightarrow x(s)$$

$$x_{Mk} \Rightarrow x_M(s)$$

bzw.

$$y(s) = \frac{1}{G_S(s)} x(s)$$

$$x_M(s) = G_M(s)w(s)$$

4 Aufgaben: Modellbasierte Regler

4.5 SPFC-Regler

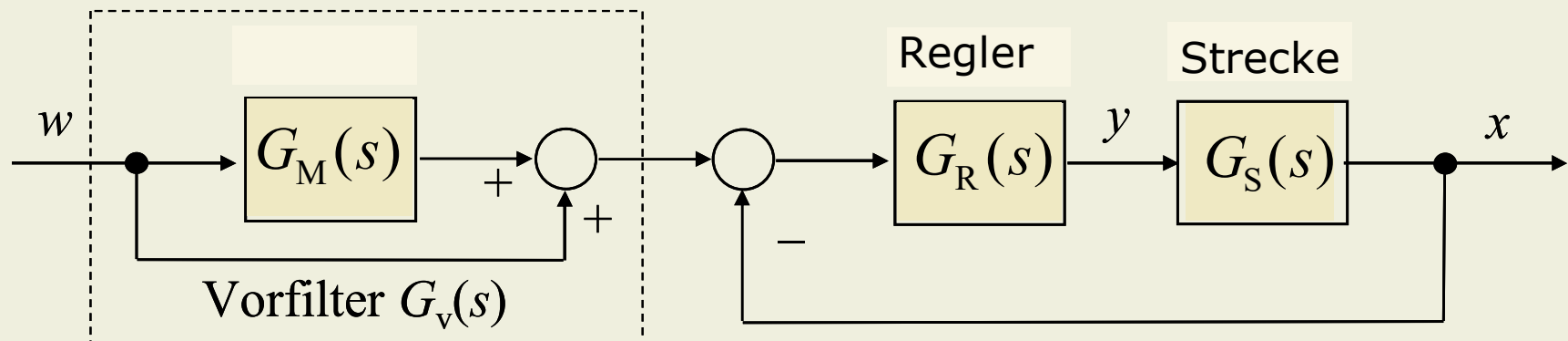
Aus dem SPFC-Algorithmus
$$y_k = \frac{1}{K_{PM}} [w - (x_k - x_{Mk})]$$

entstand in der s -Ebene folgende Gleichung:

$$y(s) = \frac{1}{G_S(s)} x(s) = w(s) - [x(s) - G_M(s)w(s)]$$

Unter Beachtung $\frac{x(s)}{w(s)} = G_W(s)$ ergibt sich (siehe Wirkungsplan unten):

$$G_W(s) = [1 + G_M(s)] \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}$$



4.5 SPFC-Regler

Aufgabe: P-T1-Strecke mit P-Regler

Gegeben sind:

P-T1-Strecke $G_S(s) = \frac{K_{PS}}{1 + sT_1}$ $K_{PS} = 0,5$
 $T_1 = 2$

P-Regler: $G_R(s) = K_{PR}$

P-T1-Modell
der Strecke: $G_M(s) = \frac{K_{PM}}{1 + sT_M}$

Gewünschtes
P-T1-Verhalten: $G_w(s) = \frac{1}{1 + sT_w}$ $T_w = 0,1$

Gesucht sind: K_{PR} , T_M , K_{PM}

4 Aufgaben: Modellbasierte Regler

4.5 SPFC-Regler

Lösung, P-Regler:

Quelle [1], Seite 363

SPFC-Gleichung mit drei unbekanntenen Größen, wovon zwei beliebig gewählt werden, z.B.:

$$G_w(s) = \frac{1}{1 + sT_w} \quad G_M(s) = \frac{K_{PM}}{1 + sT_M} \quad G_S(s) = \frac{K_{PS}}{1 + sT_1}$$
$$G_R(s) = K_{PR}$$

$$G_w(s) = [1 + G_M(s)] \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}$$

$$K_{PR} = \frac{1}{K_{PS}} = 2$$

$$T_M = T_w = 0,1$$

$$\frac{1}{1 + sT_w} = \left(1 + \frac{K_{PM}}{1 + sT_M} \right) \cdot \frac{K_{PR} K_{PS}}{1 + sT_1 + K_{PR} K_{PS}}$$

$$\frac{1}{1 + 0,1s} = \left(1 + \frac{K_{PM}}{1 + 0,1s} \right) \cdot \frac{1}{1 + 2s + 1}$$

$$K_{PM}(s) = 1 + s(T_1 - T_M)$$

$$G_M(s) = \frac{K_{PM}}{1 + sT_M} = \frac{1 + s(T_1 - T_M)}{1 + sT_M} = \frac{1 + 1,9s}{1 + 0,1s}$$

4 Aufgaben: Modellbasierte Regler

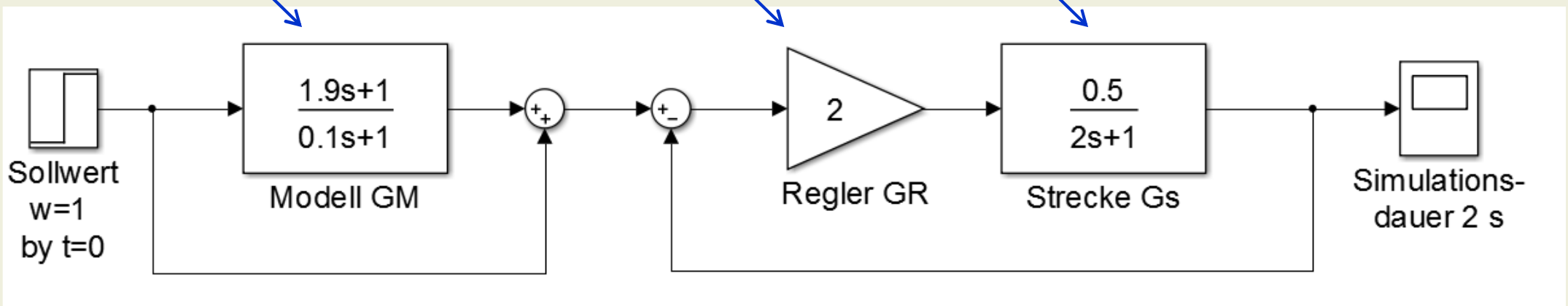
4.5 SPFC-Regler

Simulieren Sie den Regelkreis nach dem SPFC-Verfahren mit gegebenen und oben bestimmten Werten:

$$G_M(s) = \frac{1 + 1,9s}{1 + 0,1s}$$

$$G_R(s) = 0,5$$

$$G_S(s) = \frac{0,5}{1 + 2s}$$



Prüfen Sie, ob das gewünschte Verhalten

$$G_w(s) = \frac{1}{1 + 0,1s}$$

erreicht wird!

4.5 SPFC-Regler

Aufgabe: P-T1-Strecke mit PI-Regler

Gegeben sind:

P-T1-Strecke $G_S(s) = \frac{K_{PS}}{1 + sT_1}$ $K_{PS} = 0,5$
 $T_1 = 2$

PI-Regler: $G_R(s) = \frac{K_{PR}(1 + sT_n)}{sT_n}$

P-T1-Modell
der Strecke: $G_M(s) = \frac{K_{PM}}{1 + sT_M}$

Gewünschtes
P-T1-Verhalten: $G_w(s) = \frac{1}{1 + sT_w}$ $T_w = 0,1$

Gesucht sind: K_{PR} , T_M , K_{PM}

$$G_w(s) = \frac{1}{1 + sT_w}$$

$$G_M(s) = \frac{K_{PM}}{1 + sT_M}$$

$$G_R(s) = \frac{K_{PR}(1 + sT_n)}{sT_n}$$

$$G_S(s) = \frac{K_{PS}}{1 + sT_1}$$

$$G_0(s) = G_R(s)G_S(s) = \frac{K_{PR}(1 + sT_n)K_{PS}}{sT_n(1 + sT_1)}$$

Kompensation:

$$T_n = T_1 = 2$$

Annahmen:

$$K_{PR} = \frac{1}{K_{PS}} = 2$$

$$T_M = T_w = 0,1$$

$$G_w(s) = [1 + G_M(s)] \frac{G_0(s)}{1 + G_0(s)}$$

$$\frac{1}{1 + sT_w} = \left(1 + \frac{K_{PM}}{1 + sT_M} \right) \cdot \frac{K_{PR}K_{PS}}{sT_n + K_{PR}K_{PS}}$$

$$\frac{1}{1 + 0,1s} = \left(1 + \frac{K_{PM}}{1 + 0,1s} \right) \cdot \frac{1}{2s + 1}$$

4 Aufgaben: Modellbasierte Regler

4.5 SPFC-Regler

Simulieren Sie den Regelkreis nach dem SPFC-Verfahren und prüfen Sie, ob das gewünschte Verhalten $G_w(s)$ erreicht wurde!

Wie reagiert der Regler auf die Störung?

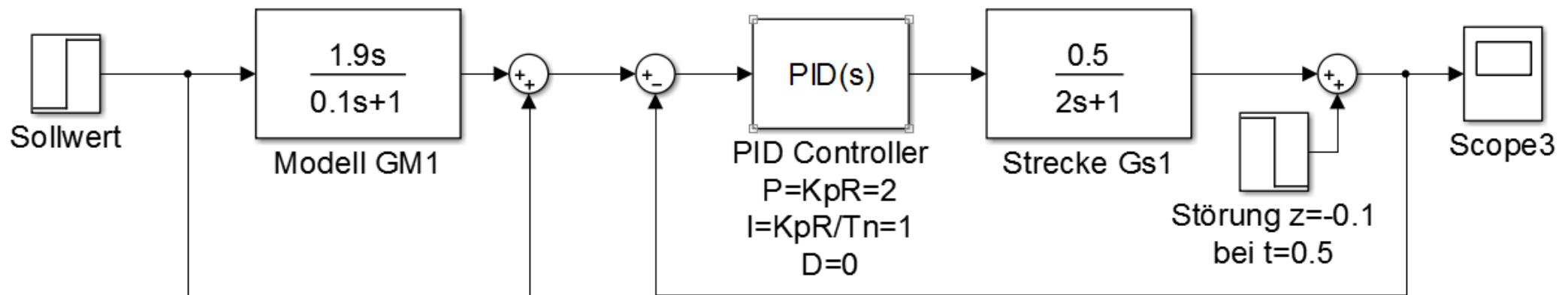
Wie kann man die Regelung beim Störverhalten verbessern?

$$\frac{1}{1 + 0,1s} = \left(1 + \frac{K_{PM}}{1 + 0,1s} \right) \cdot \frac{1}{2s + 1}$$

$$K_{PM}(s) = s(T_1 - T_M)$$

$$G_M(s) = \frac{K_{PM}}{1 + sT_M} = \frac{s(T_1 - T_M)}{1 + sT_M} = \frac{1,9s}{1 + 0,1s}$$

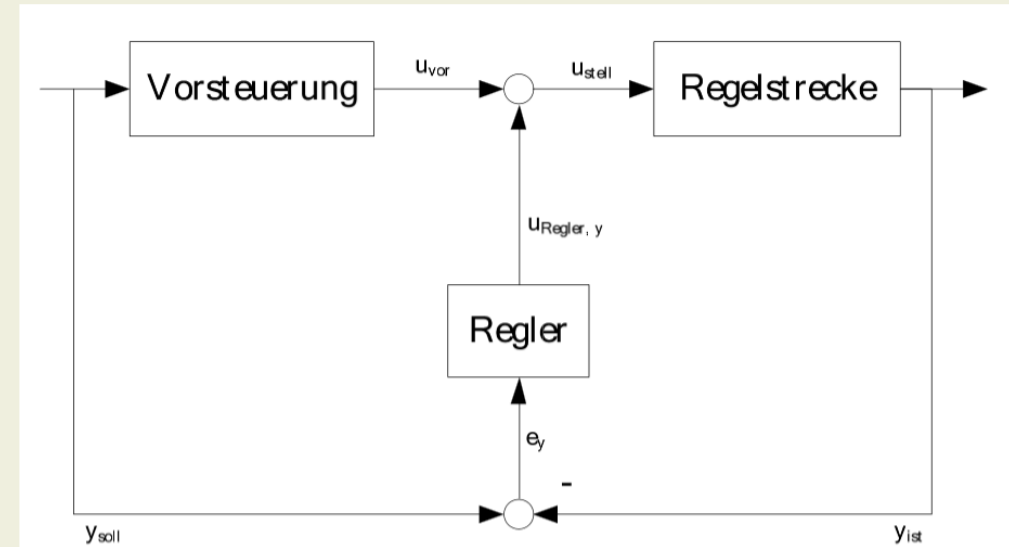
$$G_F(s) = 1 + G_M(s) = 1 + \frac{1,9s}{1 + 0,1s} = \frac{1 + 2s}{1 + 0,1s}$$



5 Trajektorienfolgeregelung

5.1 Definition

Unter Trajektorienfolgeregelung versteht man die Regelung nach einer vorgegebenen Referenztrajektorie des Sollwertes.



Quelle: https://www.mw.tum.de/fileadmin/w00bpv/www/Die_Fakultaet/TdF2012/Grasser_Zusammenfassung.pdf

Nachfolgend wird die Trajektorienfolgeregelung ohne Rückführung behandelt, die man auch als *Open-loop Control* bezeichnet.

Die *Control*-Aufgabe besteht darin, die Regelgröße aus dem gegebenen Anfangspunkt $x(0)=0$ bei $t = 0$ in einen definierten Endpunkt $x(t_{aus})$ bei $t = t_{aus}$ nach einer definierten Referenztrajektorie zu überführen.

Das Control-Signal wird von positiven $+y_{max}$ auf negativen $-y_{max}$ umgeschaltet. Die Umschaltpunkte werden optimal wie $t_{aus}/2$ gewählt.

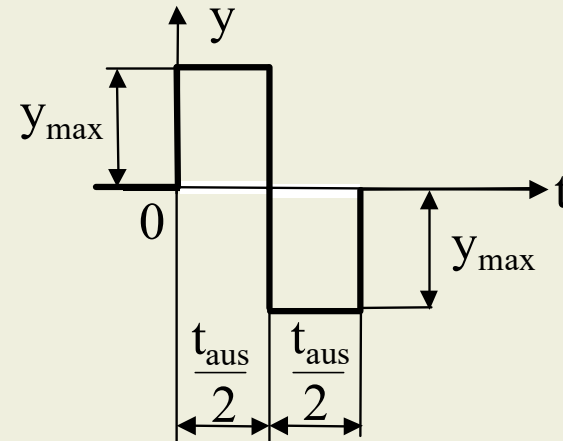
5 Trajektorienfolgeregelung

5.2 Dead-beat Control

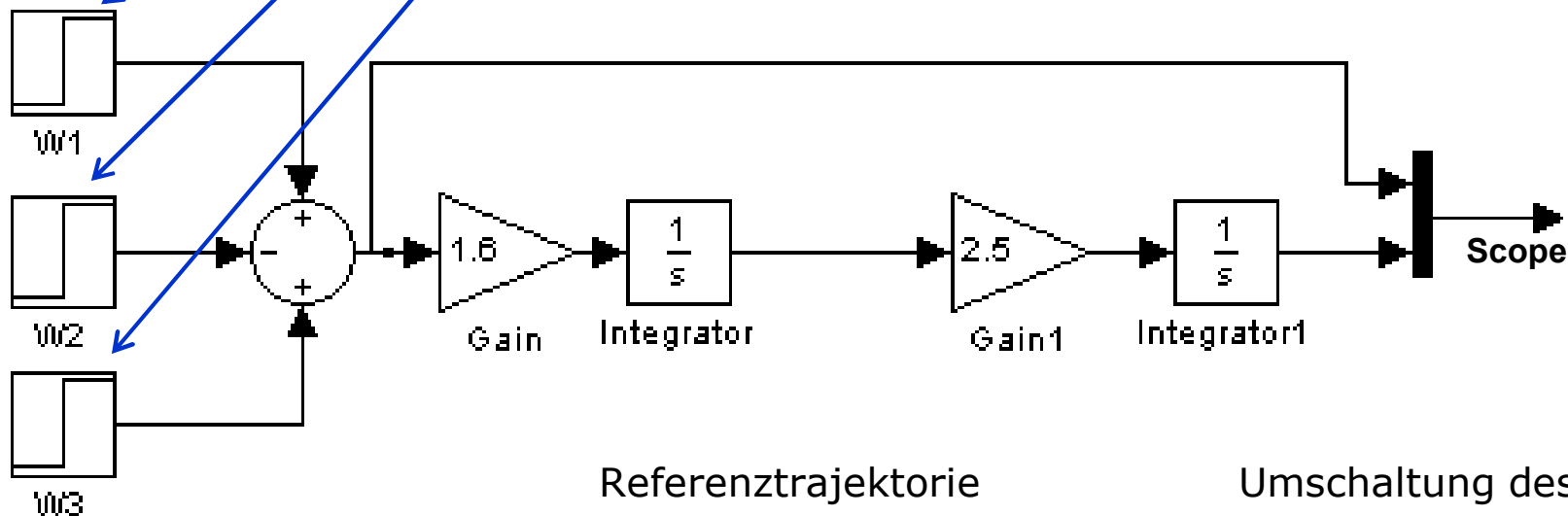
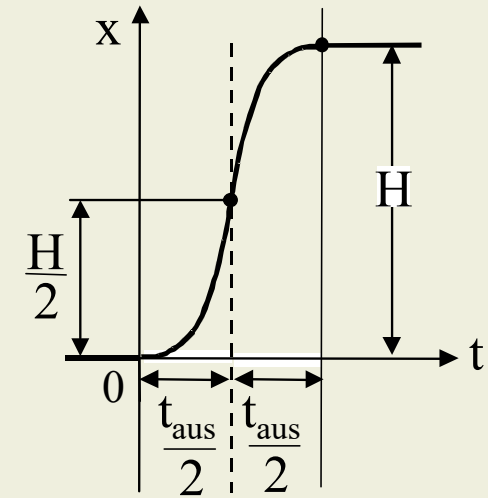
Control-Signal

Step input	W1	W2	W2
Step time	0	0.4	0.8
Initial value	0	0	0
Final value	3.125	6.250	3.125
Sample time	0	0	0

Control-Signal



Referenztrajektorie



Gegeben:
 $K_{IS}=4$
 $H=2$
 $t_{aus}=0.8$
 $y_{max}=6.25$

Referenztrajektorie

$$x(t) = \frac{1}{2} K_{IS} t^2 y_{max}$$

Umschaltung des Control-Signals

$$x\left(\frac{t_{aus}}{2}\right) = \frac{1}{2} \cdot K_{IS} \cdot \left(\frac{t_{aus}}{2}\right)^2 \cdot y_{max}$$

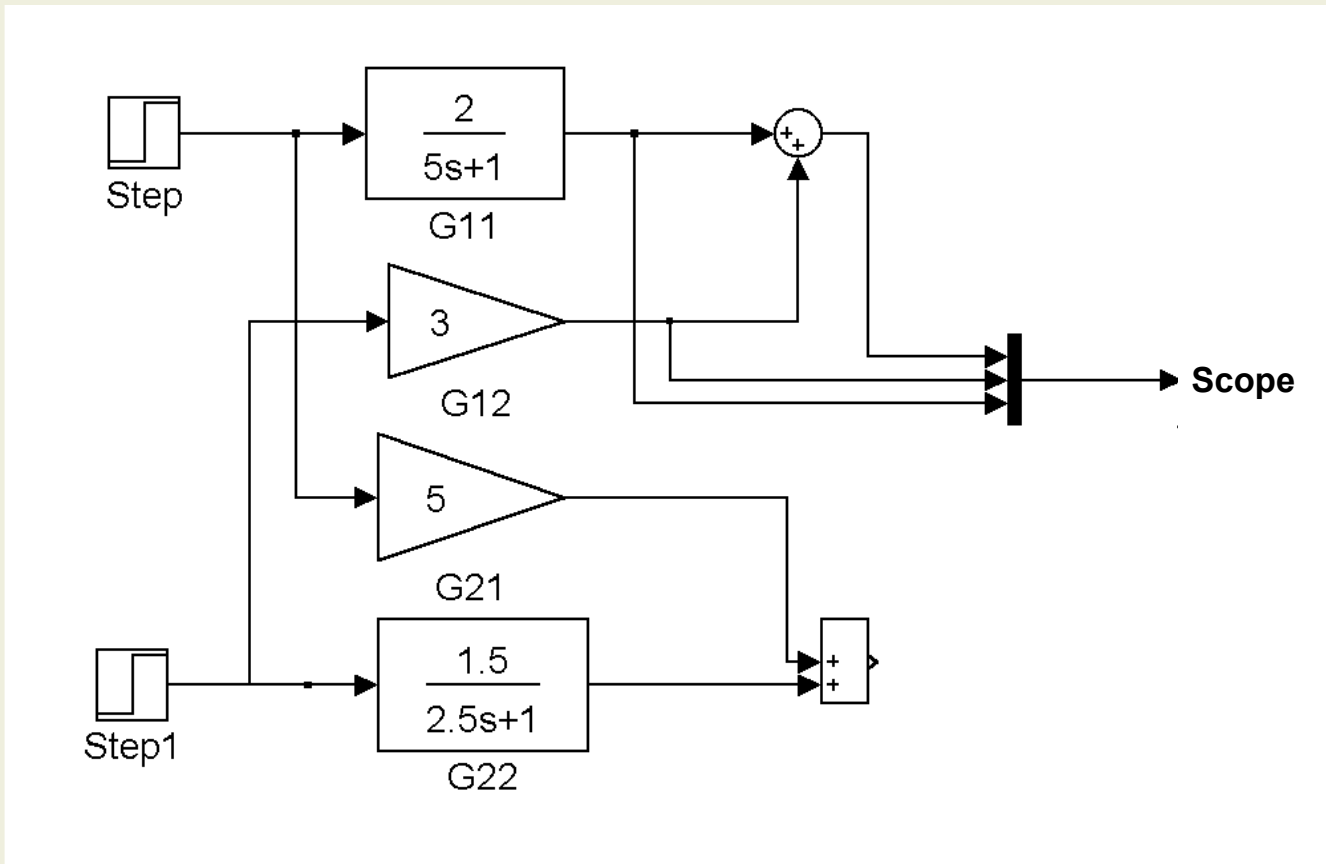
Quelle [3], Seite 116

6 Aufgaben: Mehrgrößenregelung

6.1 MIMO-Strecke in P-kanonische Form

Gegeben ist das Simulink-Modell einer MIMO-Strecke.

Quelle [2], Seite 171

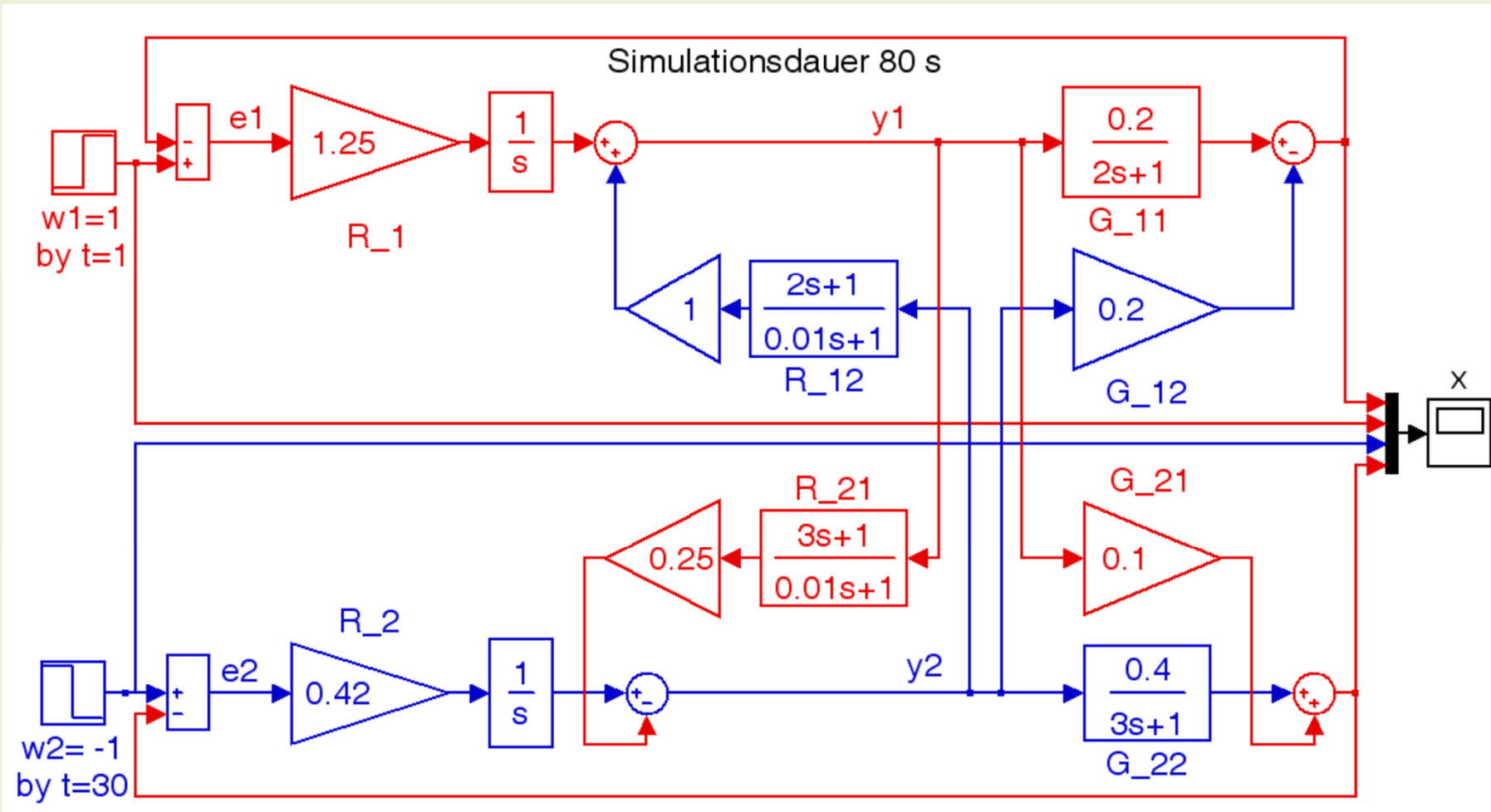


Simulieren Sie die Sprungantworten und bestimmen Sie die Übertragungsfunktionen der MIMO-Strecke in P-kanonische Form!

6.2 Entkopplungsregelung

Simulieren Sie die Sprungantworten der unten gegebenen MIMO-Regelkreises und prüfen Sie, ob der Kreis entkoppelt ist!

Quelle [1], Seite 268

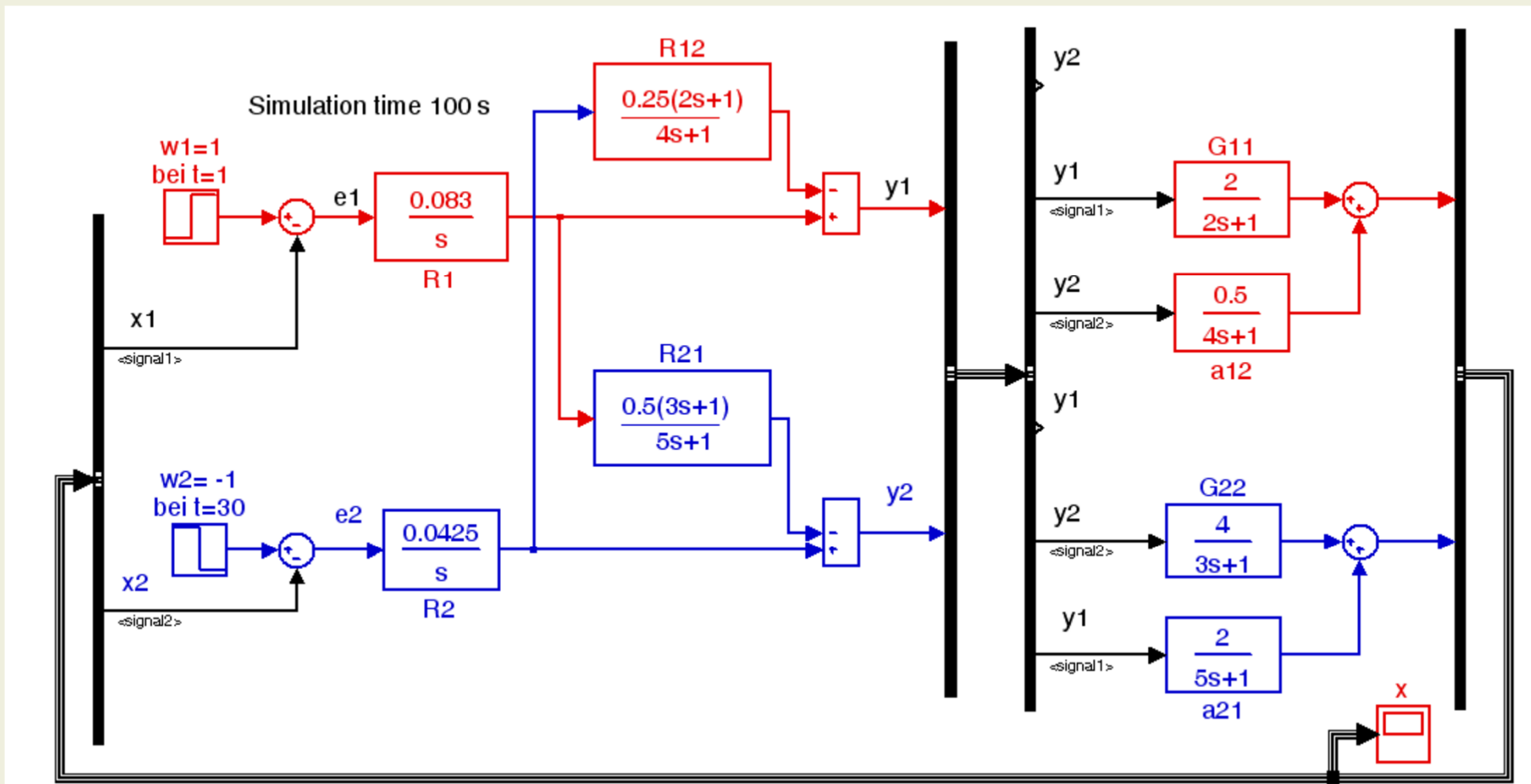


6.3 Bus-Konzept für MIMO-Regelung

Simulieren Sie die Sprungantworten der unten gegebenen MIMO-Regelkreises und prüfen Sie, ob der Kreis entkoppelt ist!

Quelle: S.Zacher. Bus-Approach for Feedback MIMO-Control, 2014, Seite 82

<http://zacher-automation.de/bus-approach.html>



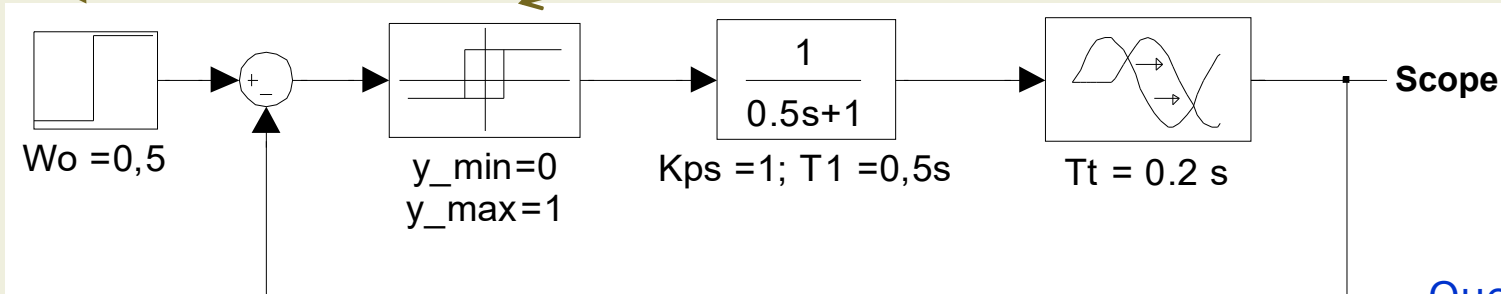
7 Zweipunktregler

7.1 Zweipunktregler ohne Schaltdifferenz

Keine Schaltdifferenz
Stellgröße $y_{max}=1$
 $y_{min}=0$

Sollwert $w=0.5$

Switch on point	0.0
Switch off point	0.0
Output when on	1
Output when	0



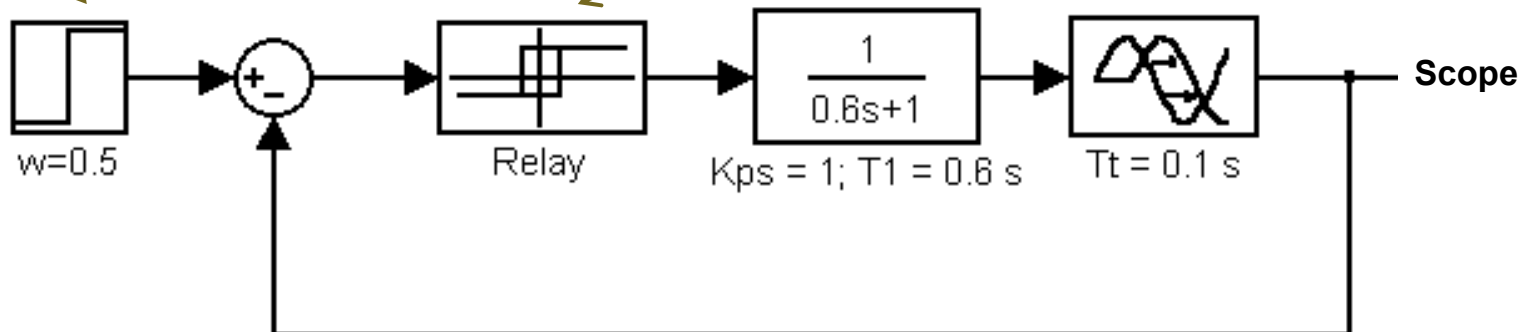
Quelle [1], Seite 299

7.2 Zweipunktregler mit Schaltdifferenz

Schaltdifferenz $x_d = 0.1$
Stellgröße $y_{max}=1$
 $y_{min}=0$

Sollwert $w=0.5$

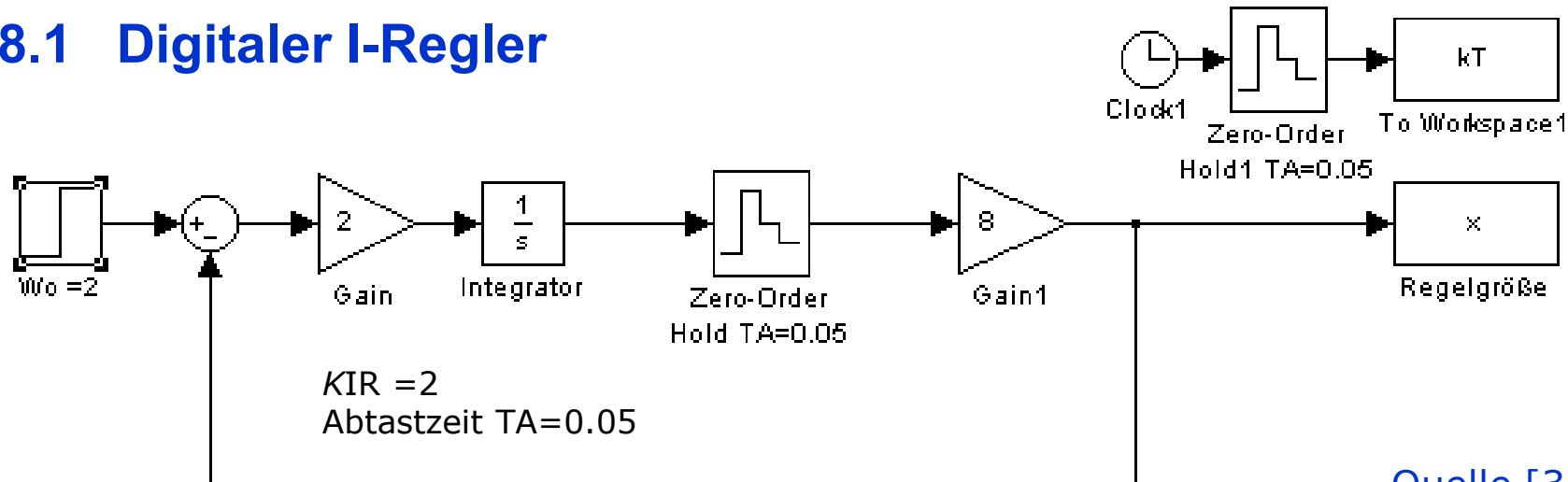
Switch on point	0.1
Switch off point	- 0.1
Output when on	1
Output when	0



Quelle [3], Seite 94

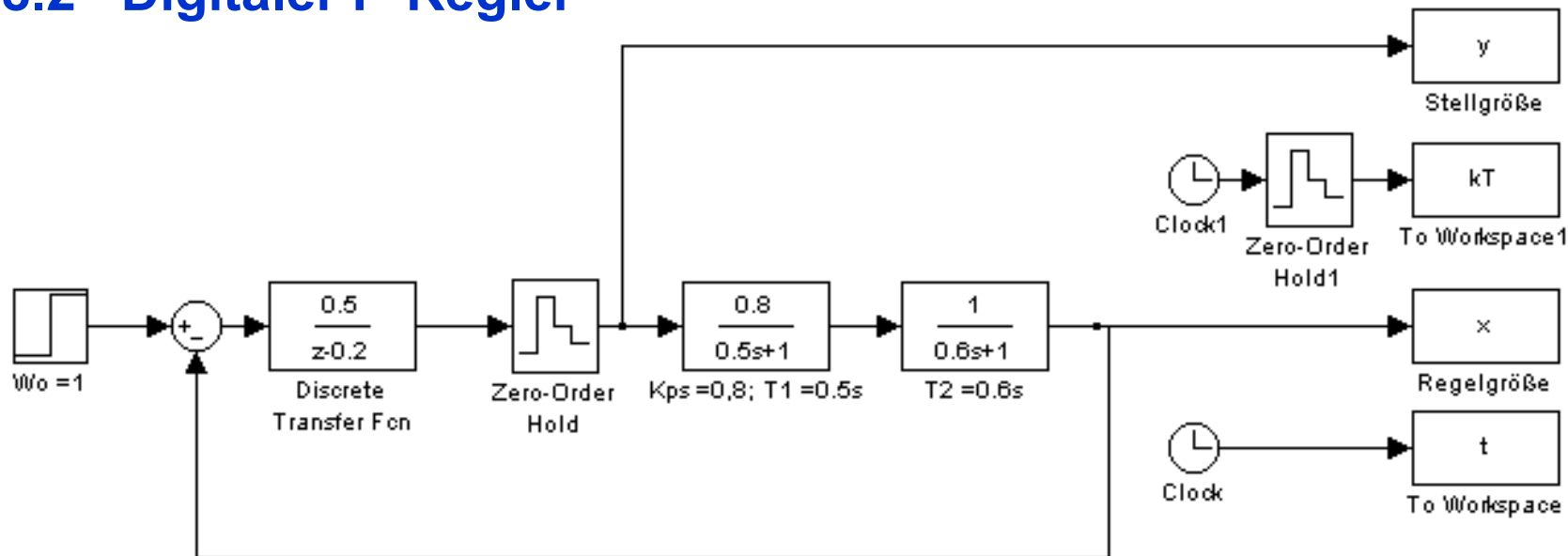
8 Digitale Regler

8.1 Digitaler I-Regler



Quelle [3], Seite 100

8.2 Digitaler P-Regler



Quelle [1], Seite 439

8.3 Digitaler Dead-beat Regler

Gegeben ist die Regelstrecke:

$$G_S(s) = \frac{0,8}{s(1 + 1,5s)}$$

$$G_{HS}(z) = \frac{0,0516}{z(z - 0,9352)} = \frac{Z_z(z)}{z^l N_z(z)}$$

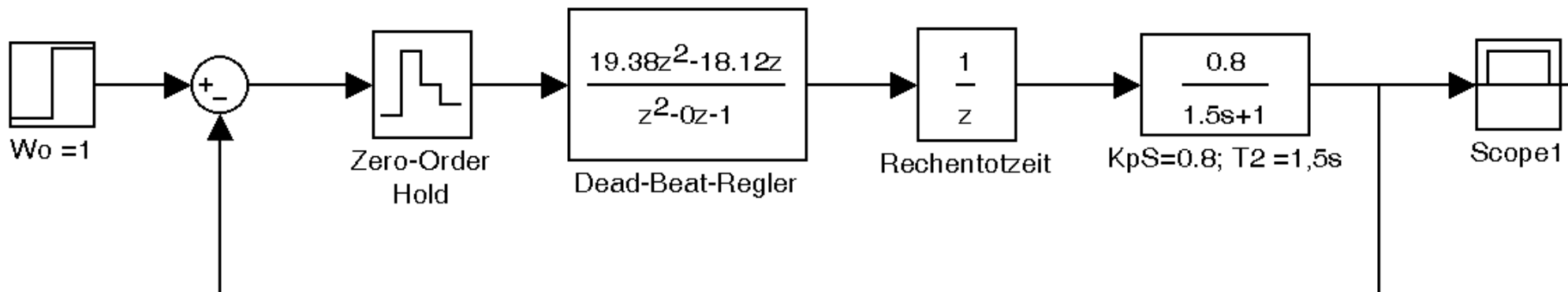
mit $Z_z(z) = 0.0516$ $N_z(z) = z - 0.9352$
 $Z_z(1) = 0.0516$ $l = 1$ $m = 1$

Der Dead-beat Regler wird bestimmt:

$$R_z(z) = \frac{z^l N_z(z)}{Z_z(1) \cdot z^{l+m} - Z_z(z)}$$

$$R_z(z) = \frac{z(z - 0,9352)}{0,0516 z^2 - 0,0516}$$

$$R_z(z) = \frac{19,38 z^2 - 18,12 z}{z^2 - 1}$$



Quelle [3], Seite 118

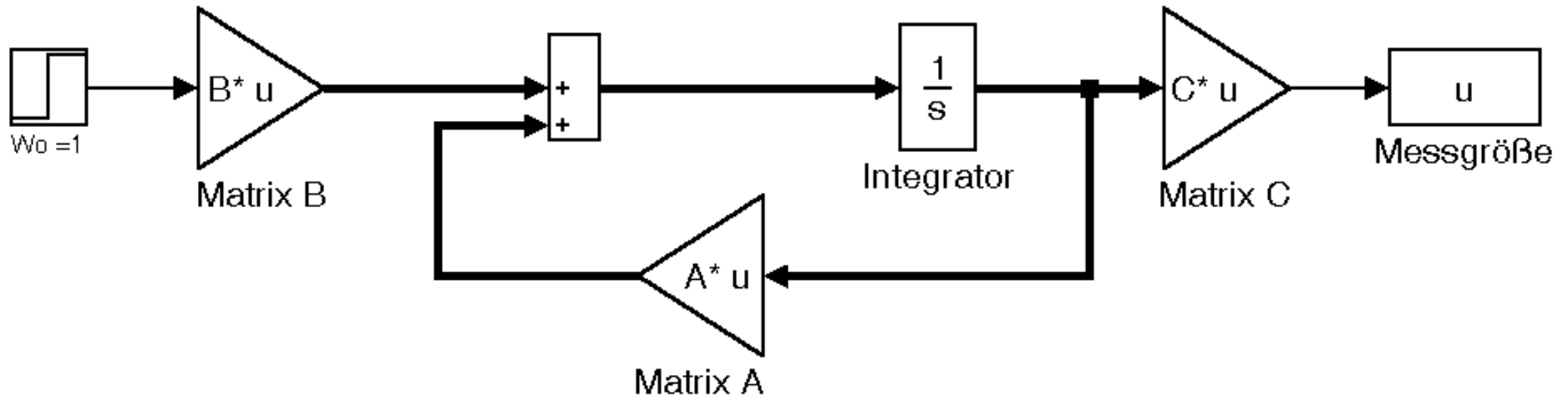
9 Zustandsrückführung

9.1 Aufgabe: Stabilität

Gegeben ist ein System:

$$A = \begin{bmatrix} 1 & 1 \\ 5 & -6 \end{bmatrix};$$
$$B = \begin{bmatrix} 2 \\ 5.5 \end{bmatrix};$$
$$C = \begin{bmatrix} 0 & 1 \end{bmatrix};$$
$$D = 0;$$

Prüfen Sie rechnerisch und mit Hilfe einer Simulation (unten), ob das System stabil ist!



Quelle [2], Seite 188

9.2 Aufgabe: Zustandsrückführung

Gegeben ist ein System
und die gewünschten Polstellen:

$$p1 = -2 + j$$

$$p2 = -2 - j$$

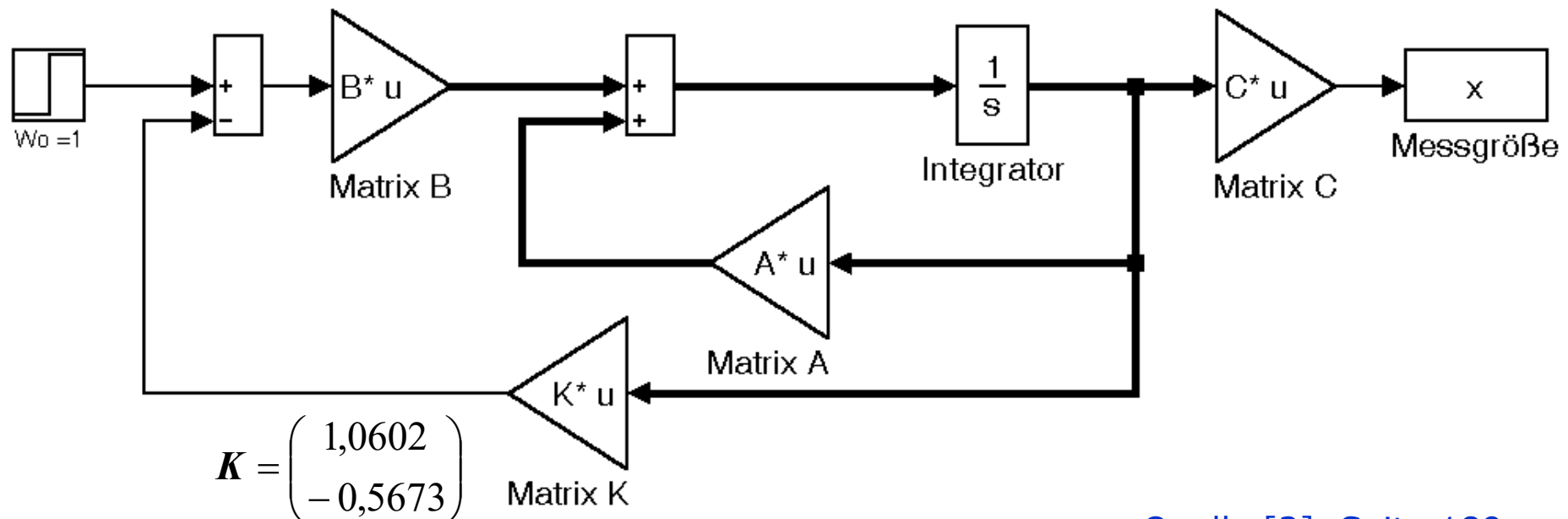
$$A = \begin{bmatrix} 1 & 1 \\ 5 & -6 \end{bmatrix};$$

$$B = \begin{bmatrix} 2 \\ 5.5 \end{bmatrix};$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix};$$

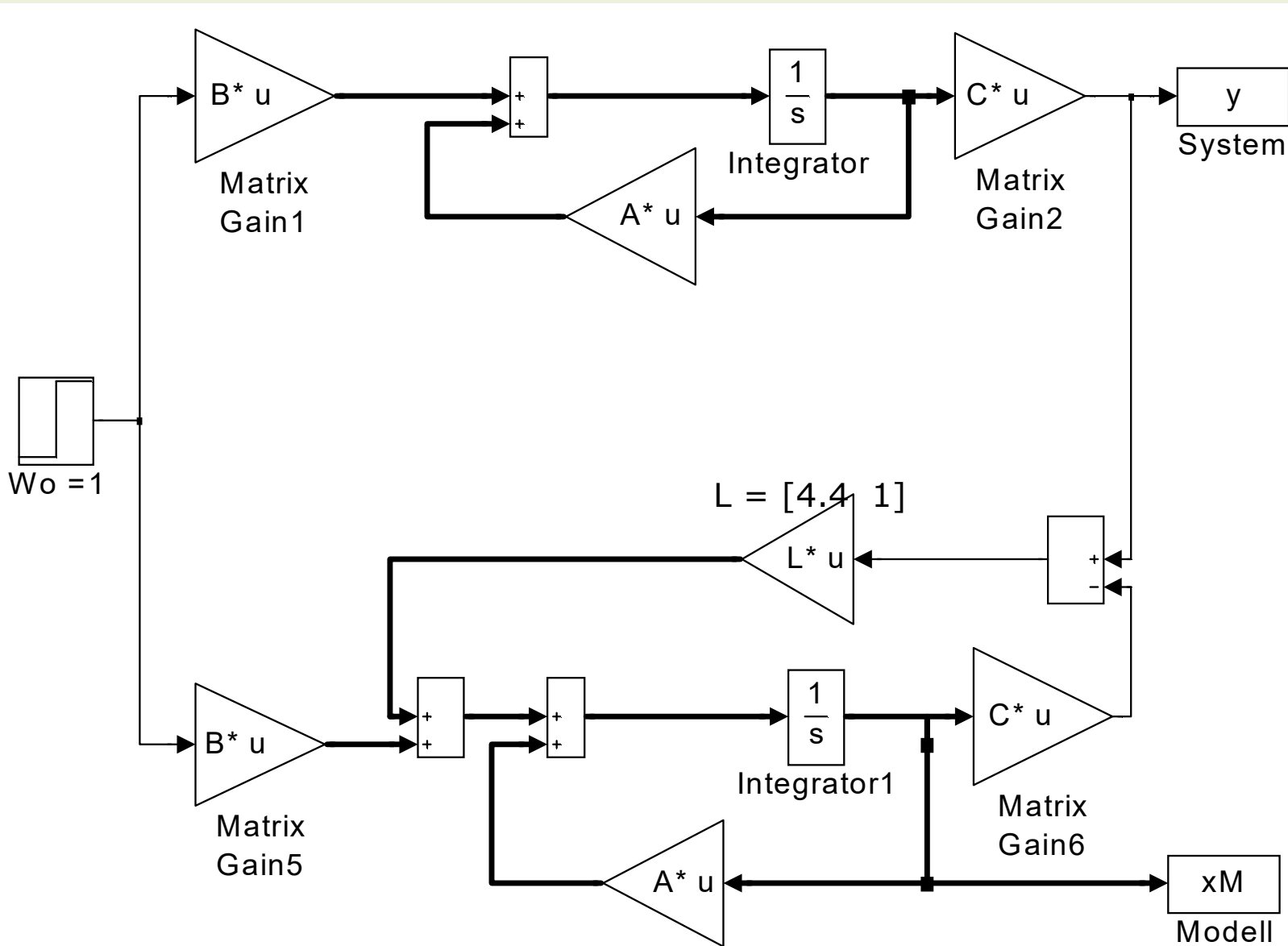
$$D = 0;$$

Bestimmen Sie dafür die Rückführmatrix und prüfen Sie das Ergebnis mittels einer Simulation (unten)!



Quelle [2], Seite 189

9.3 Zustandsbeobachter



```

A = [2, 1; 5, -5];
B = [ 2; 5.5];
C = [ 1, 0];
p1 = -2+i ;
p2 = -conj (p1);
P = [p1; p2] ;
Lob = acker(A', C', P);

```

$L = Lob'$

$AM = A - L * C$

```

subplot(311);
plot(tout, x);
subplot(312);
plot(tout, xM);
subplot(313);
plot(tout, xe);

```

Quelle [1], Seite 413

Module der Automatisierungstechnik

Prof. Dr.-Ing. S. Zacher

Ende der Präsentation

Regelungstechnik mit MATLAB/Simulink

Quellen mit insgesamt 62 Simulink-Modellen:

[1] **Zacher, S., Reuter, M.: Regelungstechnik für Ingenieure**

Verlag Springer Vieweg, 14. Auflage, 2014

Seiten: 231, 268, 270, 273, 274, 299, 360, 363-366, 369, 386, 397,
403, 408, 414, 438-440, 459, 461

[2] **Zacher, S.: Übungsbuch Regelungstechnik**

Verlag Springer Vieweg, 5. Auflage, 2014

Seiten: 63, 66, 80, 129, 131, 135, 137, 152, 163, 171, 187-189,
192, 198, 203, 205-207, 216, 217, 234, 235, 237, 238

[3] **Zacher, S.: Regelungstechnik Aufgaben**

Verlag Dr. Zacher, 3. Auflage, 2012

Seiten: 35, 82-84, 87, 89, 90, 94, 100, 105, 108-111, 116, 118