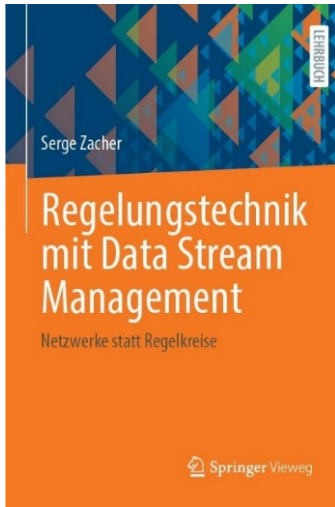


Prof. em, Dr.-Ing. S. Zacher



# Schubert-Terminator

Beseitigung von Störgrößen bei Regelstrecken  
mit Totzeiten und D-Anteilen

„Die bekannten Verfahren der Störgrößenunterdrückung sind nur dann wirksam, wenn die Störungen bekannt und messbar sind, was eher selten vorkommt. In diesem Kapitel ist das Störverhalten nach einem neuen Konzept behandelt, wonach ein neues Modul, so genanntes Data Stream Manager (DSM) entwickelt ist. Das DSM ist „Terminator“ genannt, weil damit alle Störungen, ohne gemessen zu werden und unabhängig von Applikationsstelle, komplett beseitigt werden.“

**S. Zacher: *Regelungstechnik mit Data Stream Management.***

Kapitel 6, Springer Vieweg Verlag, 2021 <https://www.springer.com/gp/book/9783658308599>

# Abstract, Urheberrechts- und Haftungshinweis

Im vorliegenden Artikel wird der Einsatz des Terminators für PD-T2-Strecken behandelt, d.h. für Strecken mit Zeitverzögerung 2. Ordnung und einem differenzierenden D-Anteil. Die Besonderheit der betrachteten Strecken besteht auch darin, dass sie außerdem eine Totzeit besitzen, vor oder nach der Stelle, wo die Störung wirkt.

Die Größe und die dynamischen Eigenschaften der Störgröße (d.h. deren Übertragungsfunktion und deren Applikationsstelle) sind unbekannt, so dass sich die konventionellen Verfahren der Störgrößenkompensation, wie z.B. die Störgrößenaufschaltung, nicht anwenden lassen. „Eine Beseitigung der Auswirkung von Störgrößen durch eine Regelung hat den Nachteil, dass der Regler immer erst korrigierend eingreifen kann, wenn eine Regeldifferenz vorliegt. Wegen der Verzögerungen in der Strecke erscheint die Störung erst verspätet am Eingang des Reglers.“ (Zitat, Quelle [2], Seite 260)

An MATLAB®-Beispielen ist gezeigt, wie man einen Data Stream Manager „Terminator“ (siehe [1], [2], [3]) für solche PD-T2-Strecken mit Totzeiten realisieren kann und wie die externen Störgrößen damit - wenn auch nicht komplett, jedoch viel schneller als mit klassischen Verfahren - beseitigt werden.

Ausschlaggebend für die Störkompensation mit dem Terminator ist das neue Konzept der Regelung (siehe [1]), nach dem das Führungsverhalten „ANFAHREN“ (rise) und das Störverhalten „HALTEN“ (hold) bezeichnet werden. Dementsprechend werden beide Verhalten auch unterschiedlich behandelt, nämlich: beim ANFAHREN wird die Regelgröße mit einem klassischen Standardregler zur gegebenen Führungsgröße  $W$  bzw. zu dem gegebenen Sollwert  $X_{\text{soll}} = W$  „angefahren“. Nach dem ANFAHREN-Betrieb beendet wird, fängt der HALTEN-Betrieb an. Dabei übernimmt der Terminator die Überwachung nicht des Sollwertes  $X_{\text{soll}}$ , sondern des Sollwertes der Stellgröße  $Y_{\text{soll}}$ . Dadurch wird es ermöglicht, entweder die Störung zu erkennen, wie es bei einem *Luneberger*-Beobachter der Fall ist (siehe z.B. [2]), oder direkt die Störung zu beseitigen.

Für Fragen, Anmerkungen und Anregungen oder Hilfestellung gilt die Mail-Adresse des Autors [info@szacher.de](mailto:info@szacher.de).

---

Die vorliegende Publikation unterliegt der Urheberrecht. Alle Rechte sind bei Dr. S. Zacher vorbehalten.

**All rights are by the author, Dr. S. Zacher, reserved.** Die Weiterentwicklung oder Nutzung der Publikation ohne Referenz auf Urheber ist nicht zugelassen. **No use of this publication without references on the author.**

Für die Anwendung der vorliegenden Publikation in der Industrie, im Laborbetrieb und in anderen praktischen Fällen sowie für eventuelle Schäden, die aus unvollständigen oder fehlerhaften Angaben über das dynamische Systeme ergeben können, übernimmt der Autor keine Haftung. **For the practical use of the results of this publication takes the author no responsibility.**

# INHALT:

Vorwort .....	4
1. Konzept des Terminators .....	6
2. Mathematische Herleitung des Terminator-Algorithmus .....	9
3. Terminator für Regelstrecken ohne Totzeiten .....	11
3.1 MATLAB-Skript: PT2-Strecke ohne Totzeit .....	11
3.2 Simulink: PT2-Strecke ohne Totzeit.....	13
4. Terminator für Regelstrecken mit D-Anteilen und Totzeiten ....	14
4.1 Simulink: PT2-Strecke mit Totzeit .....	14
4.2 Simulink: PDT2-Strecke ohne Totzeit.....	15
4.3 ASA-Regler mit Bypass für PDT2 ohne Totzeit.....	16
4.4 Terminator-Algorithmus für Strecken mit Totzeiten .....	18
4.3 Simulink: PDT-2-Strecke mit Totzeit.....	20
5. Schubert-Terminator für PDT2-Strecken mit Totzeiten.....	21
5.1 Steuerung des Soll-Stellwertes $y_{\text{soll}}$ .....	21
5.2 Regelung des Soll-Stellwertes $y_{\text{soll}}$ .....	23
5.3 MATLAB-Skript: PDT2-Strecke mit Totzeit .....	25
5.4 Simulink: PDT2-Strecke mit Totzeit .....	29
6. Schubert-Terminator mit RISE- und HOLD .....	30
3.1 MATLAB-Skript: R&H für PDT2-Strecke mit Totzeit.....	31
3.2 Simulink: R&H für PDT2-Strecke mit Totzeit.....	37
7. Zusammenfassung .....	40
8. Literaturverzeichnis .....	41

## Vorwort

Ein Baustein des Regelkreises, mit dem alle Störungen, ohne gemessen zu werden und unabhängig von Applikationsstelle, komplett beseitigt werden, wurde zum ersten mal 2021 in meinem Buch „*Regelungstechnik mit Data Stream Management*“ angeboten und „Terminator“ genannt. Das ist die Weiterentwicklung des „Router“-Konzepts, das noch 2014 auch in meinem Buch „*Bus-Approach for Feedback MIMO-Control*“ veröffentlicht wurde.

Beide Bausteine, wie auch die ganze Reihe von ASA-Reglern (siehe *Automation-Letters* 04, 06-08, 16, 25, 32, 33, 38 <https://www.szacher.de/Automation-Letters/>) basieren auf einer, mit dem Jahr 2000 datierten Publikation:

„Die Idee ... basiert auf der Antisymmetrie, die die gleichen Gruppenoperationen wie die Symmetrie benutzt, jedoch in Gegenrichtung. Dieses Prinzip ist zwar seit 1929 bekannt, in technischen Bereichen liegen aber noch keine antisymmetrischen Darstellungen vor. Dabei liefert die Physik überzeugende Erfolge der Antisymmetrie, wo mehrere Anwendungsbeispiele wie Kraft und Gegenkraft, Elektron und Positron, Proton und Neutron zu finden sind.“ (Zitat, Quelle: Buch S. Zacher „*SPS-Programmierung mi Funktionsbausteinsprache*“, 2000, VDE-Verlag, Seite 182).

Der Ansatz von zwei gleichen Systemen, die in Gegenrichtung wirken, habe ich später für dynamische Systeme als duale Operationen der Symmetrie und Antisymmetrie in Buch „*Duale Regelungstechnik*“ (2003) verallgemeinert, *Antisystem-Approach (ASA)* genannt und ab den 14. Auflage (2014) ins Buch „*Regelungstechnik für Ingenieure*“ eingefügt.

Die ausführliche Beschreibung der Antisymmetrie, sowie die daraus hergeleiteten universellen Stabilitätskriterien ZBV (Zwei-Bode-Plots-Verfahren) und DBV (Drei-Bode-Plots-Verfahren) findet man in meinem Buch „*Drei-Bode-Plots-Verfahren*“. Ein Zitat aus diesem Buch, Seite 164, ist unten gebracht:

„Nach dem *Antisystem-Approach (ASA)* kann zu jedem System ein Anti-System ermittelt werden, wodurch zwischen diesen beiden Systemen eine Bilanz entsteht. Diese Bilanz kann auf verschiedene technische Systeme angewandt werden. Die Bilanz zwischen dem eigentlichen System bzw. der Regelstrecke und dem dazugehörigen Antisystem (genannt auch *Schattenstrecke*), ist für das Antisystem-Approach von großer Bedeutung. Der Präfix „Anti“ deutet darauf hin, dass die beiden Systeme gegeneinander wirken“.

In vorliegender Publikation ist zum ersten mal eine neue Version des Terminators präsentiert, in der die regelungstechnischen Grundglieder mit Steuerungsbausteinen zusammengesetzt sind, was dem Konzept „Data Stream Management“ [1] entspricht.

## 1 Vorwort

Ausschlaggebend für die vorliegende Publikation war die Mail eines Studierenden aus München:

„Ich schreibe gerade meine Masterarbeit ... und untersuche hierbei Regler bzw. Reglerstrukturen unter dem Einfluss von Totzeiten. Dabei bin ich auf ihre Artikel zur ASA-Regelung und auf den ASA-Prädiktor bzw. das Buch Data-Stream-Management gestoßen. Im Anschluss habe ich meine Strecke mit PDT2-Verhalten (Strecke ist stabil, minimalphasiges Verhalten ist gegeben) unter dem Einfluss von Totzeiten in MATLAB und Simulink untersucht. Mir fiel dabei das schlechte Störverhalten auf. Dieses gilt es nun zu kompensieren bzw. zu minimieren.

Dazu habe ich das Thema mit dem Terminator in ihrem Buch gefunden. Diesen habe ich um die Strecke herum gebaut. Allerdings bringen die D-Anteile ziemlich viel Unruhe in den Regelkreis, wenn sowieso schon eine Latenz in der Strecke vorhanden ist. Der DSM-Turbo Baustein brachte auch kein Erfolg. Ich bekomme leider nicht die gewünschte Störunterdrückung hin. Haben Sie dazu eine Idee oder können mir an gegebener Stelle weiterhelfen?

Vielen lieben Dank im Voraus und vor allem bedanke ich mich für die Literatur von Ihnen und die guten Denkanstöße.“

Die Anfrage wollte ich nicht unbeachtet nicht lassen und begann die besagte Strecke simulieren. Jedoch war das Störverhalten wirklich so „unregelbar“, dass ich mehr als eine Woche für die Lösung benötigte. Während dieser Zeit erinnerte ich mich an einer Anekdote über Arbeitseifer des Komponisten Franz Schubert, „dass er nachts stets seine gewohnten Augengläser aufbehielt, damit er am Morgen gleich ohne Zeitverlust zu komponieren beginnen konnte.“ (Zitat, Quelle: *Franz Schubert*, <https://de.wikipedia.org/wiki/> besucht am 02.12.2021). Unter ähnlichen Umständen entstand auch der neue, hier präsentierte Terminator, den ich *Schubert-Terminator* genannt hatte.

Zum Schluss möchte ich den Studierenden aus München nicht anonymisiert, sondern namentlich bedanken. Nachdem er dafür seine freundliche Zustimmung erteilt hatte, beende ich das Vorwort mit folgendem Satz:

Mein besonderer Dank gilt Herrn Patrick Schubert für seine anregende Mail !

S. Zacher

Stuttgart, den 30. Dezember 2021

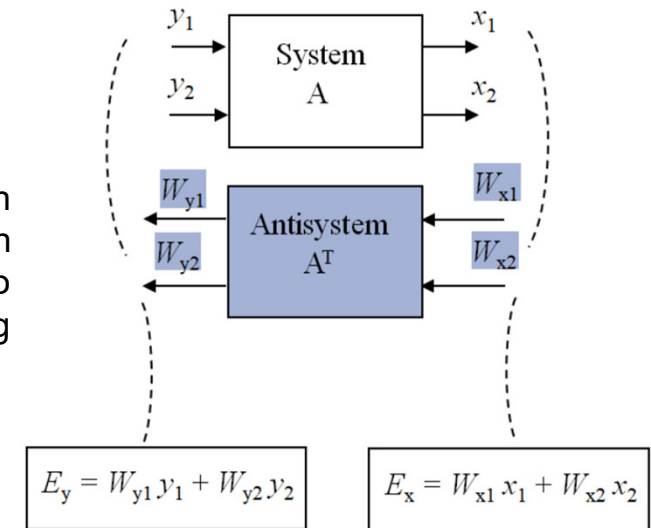
# 1 Konzept des Terminators

Die Idee des Terminators wurde zum ersten mal 2014 in [8] für die Entkopplung bei Mehrgrößenregelung mittels eines Routers veröffentlicht. Die Beschreibung des Data Stream Managers „Terminator“ findet man in [1], [3], [4].

Der Terminator entstand aus zwei Konzepten:

- *Antisystem-Approach* (siehe <https://youtu.be/UhUrWrx24Ag>)
- *Data Stream Management* (siehe <https://youtu.be/HyKQQOU1Low>)

Nach dem *Antisystem-Approach* [11] (siehe Bild rechts) kann man zu jedem beliebigen System mit der Übertragungsfunktion  $G(s)$  ein Antisystem mit der gleichen Übertragungsfunktion  $G(s)$ , jedoch mit der Signalübertragung in Gegenrichtung, so bilden, dass sich beide Systeme bei beliebigen Eingangssignalen  $e(s)$  gegenseitig kompensieren. Es gilt immer die Bilanz  $E_y = E_x$ .

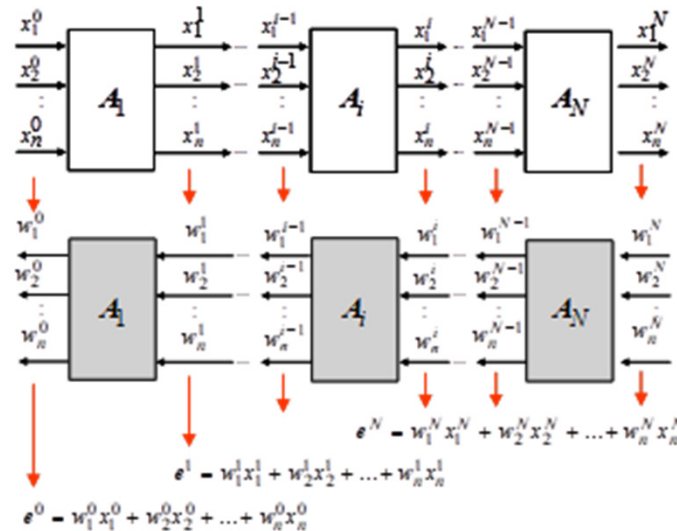


Die gesamte Übertragungsfunktion  $G(s)$

$$G(s) = G_{\text{system}} G_{\text{antisystem}}$$

eines Systems, das aus dem Originalsystem  $G_{\text{system}}$  und dem Antisystem  $G_{\text{antisystem}}$  gebildet ist, wird bei beliebigen Eingangssignalen immer  $G(s) = 1$  sein.

Ein Beispiel ist in [12] gegeben:



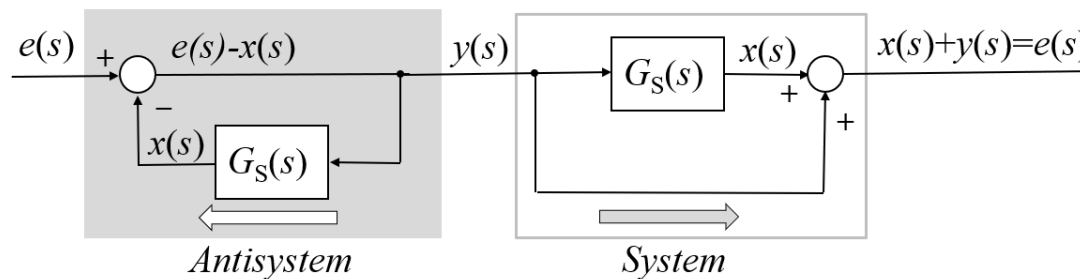
## 1 Konzept des Terminators

Es gibt natürliche Antisysteme, wie z. B. eine Strecke  $G_S(s)$  im geschlossenen Regelkreis mit einem Regler  $G_R(s)$ . Es gilt für die charakteristische Gleichung:

$$1 + G_0(s) = 0 \quad \text{bzw.} \quad G_S(s)G_R(s) = -1$$

Das Originalsystem ist  $G_{\text{system}} = G_S(s)$  und das Antisystem ist die negativ inverse Übertragungsfunktion des Reglers  $G_{\text{antisystem}} = -(1/G_R(s))$ , so dass es bei beliebigen Eingangssignalen immer die Bilanz  $G_{\text{system}}G_{\text{antisystem}} = 1$  erfüllt ist.

Andererseits kann man Antisysteme als Reihenschaltung einer Rückführung mit einer Parallelschaltung bilden (Bild unten). Die Übertragungsfunktion  $G(s)$  des Originalsystems ist üblicherweise die Übertragungsfunktion der realen physikalischen Regelstrecke. Die Übertragungsfunktion  $G(s)$  des Antisystems ist das Modell der Strecke und wird dadurch *Schatten-Strecke* genannt.



Die Schatten-Strecke muss nicht unbedingt ein mathematisches Modell der realen Strecke sein. Eine zweite Strecke, falls sie verfügbar ist, kann auch physikalisch als *Schatten-Strecke* eingesetzt werden [13].

Das zweite Konzept, nach dem der Terminator angeboten wurde, ist *Data Stream Managements*. Die Regelung wird in zwei Betriebsphasen eingeteilt (siehe <https://youtu.be/F5Had3sjo>):

- ANFAHREN: die Ist-Regelgröße  $x(t)$  wird mit dem Regler zum Sollwert  $w = x_{\text{soll}}$  gebracht. Es wird angenommen, dass während dieser Phase keine Störung  $z(t)$  wirkt, was auch in der klassischen Regelungstechnik beim Führungsverhalten der Fall ist. Nach dem ANFAHREN-Abschluss bei  $t = \infty$  (wie man die Ausregelzeit in der Regelungstechnik mathematisch bezeichnet) erreicht die Ist-Regelgröße  $x(\infty)$  den Sollwert  $w = x_{\text{soll}}$ . Dabei wird sich die Stellgröße  $y(\infty)$  zu einem Sollwert  $y_{\text{soll}}$  stabilisieren.
- HALTEN: es wird angenommen, dass während dieser Betriebsphase keine Änderung des Sollwertes  $w = x_{\text{soll}}$  passiert, was dem Begriff *Störverhalten* aus der Regelungstechnik wiederum entspricht. Wirkt nun eine Störgröße  $z(t)$ , wird entgegen der klassischen Regelungstechnik keinen Regler, sondern der Terminator die *Regelung* übernehmen.



## 1 Konzept des Terminators

Der Wirkungsplan eines Regelkreises mit dem Terminator ist unten gezeigt. Es gilt:  $G_S(s) = G_{S1}(s)G_{S2}(s)$

Die Übertragungsfunktion des geschlossenen Regelkreises **ohne Terminator** wird bekanntlich wie folgt ermittelt:

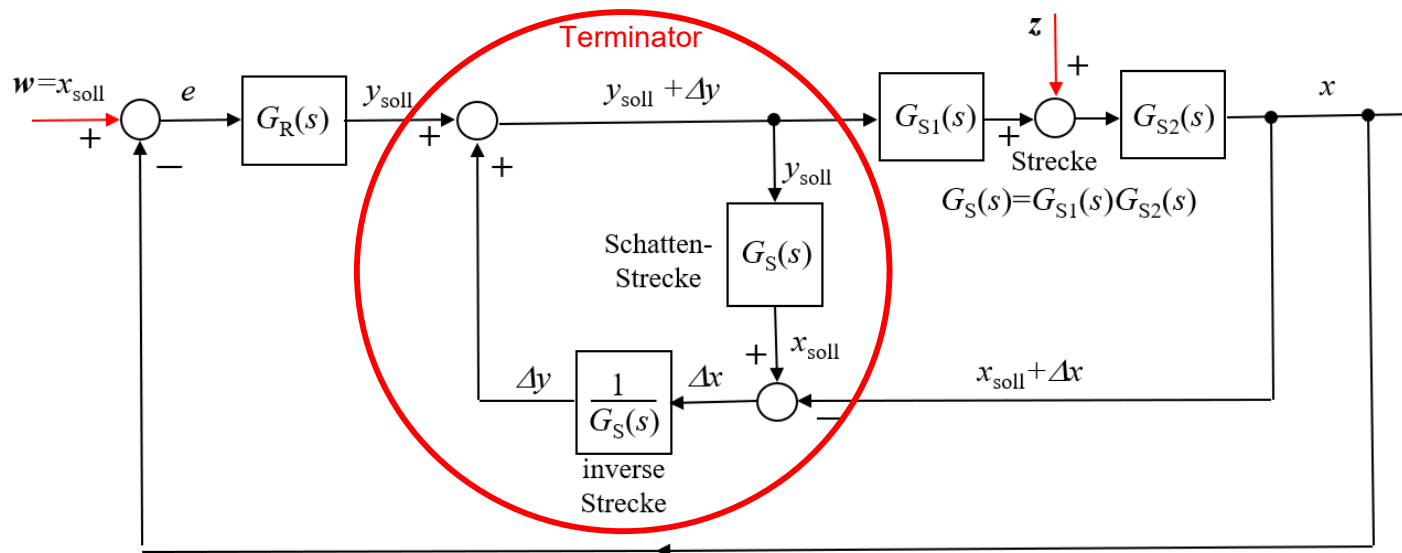
ANFAHREN: Führungsgröße  $w = x_{\text{soll}}$ ; Störgröße  $z = 0$

$$G_w(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}$$

HALTEN: Führungsgröße  $w = 0$ ; Störgröße  $z \neq 0$

$$G_w(s) = \frac{G_{vz}(s)}{1 + G_0(s)} = \frac{G_{S2}(s)}{1 + G_R(s)G_S(s)}$$

Der Terminator-Algorithmus wird unten im Allgemeinen ohne Mathematik beschrieben, die mathematischen Hintergründe folgen im nächsten Kapitel.



Durch den Vergleich der realen „gestörten“ Strecke  $G_S(s)$  mit einem „ungestörten“ Strecken-Modell (*Schatten-Strecke*) wird der „Zuwachs“ der Regelgröße  $\Delta x$  als die Differenz zwischen der „gestörten“ Regelgröße  $x_{\text{soll}} + \Delta x$  und dem Sollwert  $x_{\text{soll}}$  bestimmt.

Danach wird der Regelgrößen-„Zuwachs“  $\Delta x$  in die Stellgrößen-„Zuwachs“  $\Delta y$  vom Terminator automatisch umgewandelt:

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

Der Stellgrößen-„Zuwachs“  $\Delta y$  wird vom Terminator zum Stellgrößen-Sollwert  $y_{\text{soll}}$  addiert bzw. subtrahiert, so dass der Regelgrößen-„Zuwachs“  $\Delta x$  eliminiert wird.

Die Wirkung des Terminators erfolgt viel schneller, als die Wirkung des klassischen Reglers beim Störverhalten, weil die inverse Regelstrecke  $1/G_S$  hohe D-Anteile besitzt. Solche Störungskompensation wirkt unabhängig davon, an welcher Stelle die Störgröße wirkt, weil die  $G_S(s)$  die Übertragungsfunktion der gesamten Strecke ist.



## 2 Mathematische Herleitung des Terminator-Algorithmus

Nehmen wir an, dass die Übertragungsfunktionen der Regelstrecke  $G_S(s)$  und die des Reglers  $G_R(s)$  gegeben sind. Unbekannt ist die Höhe der Störgröße  $z(s)$  sowie der Stelle, zu welcher die Störung eingegeben wird. In anderen Worten, bekannt ist, dass es

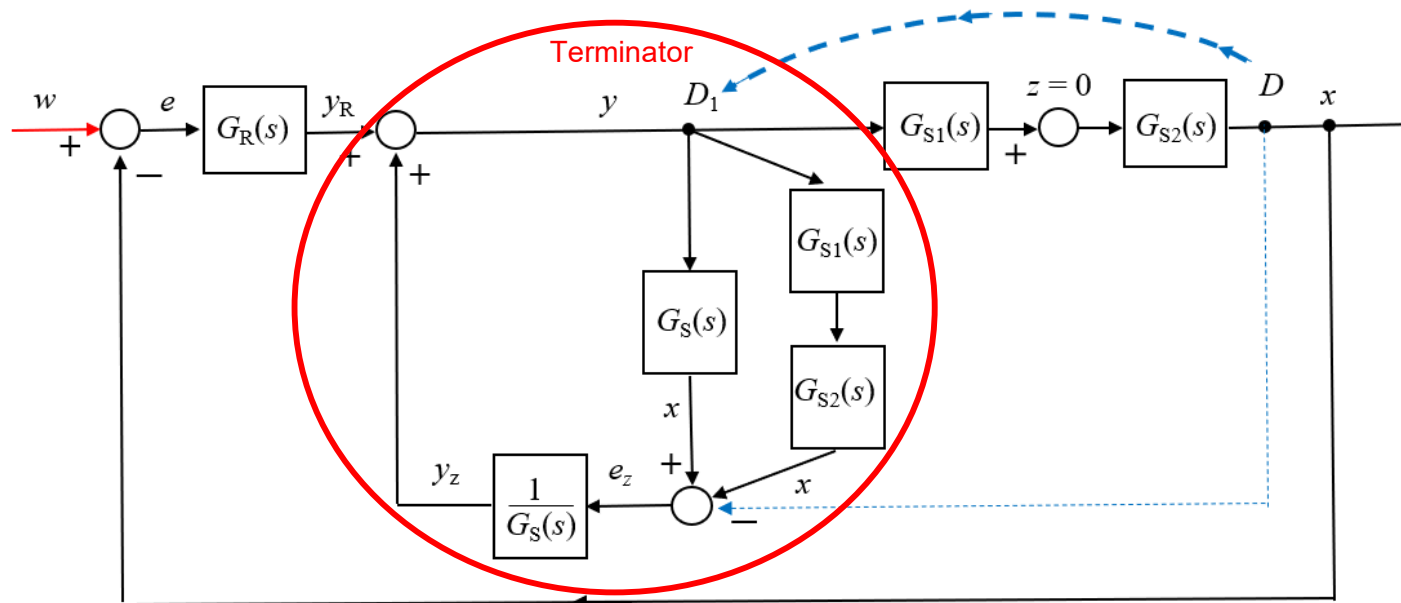
$$G_S(s) = G_{S1}(s)G_{S2}(s)$$

gilt, aber die Übertragungsfunktionen der Teilstrecken vor der Störung  $G_{S1}(s)$  und nach der Störung  $G_{S2}(s)$  sind nicht gegeben.

**ANFAHREN mit Terminator:** Führungsgröße  $w = x_{\text{soll}}$ , Störgröße  $z = 0$ .

Um die Übertragungsfunktion  $G_w(s)$  des Regelkreises mit dem Terminator zu bestimmen, überführen wir die Abzweigungsstelle aus dem Punkt D in Punkt D1 (Bild unten). Daraus folgt:

$$G_0(s) = G_R(s) \cdot \frac{1}{1 + [G_S(s) - G_{S1}(s)G_{S2}(s)] \cdot \frac{1}{G_S(s)}} \cdot G_{S1}(s)G_{S2}(s) = G_R(s)G_S(s) \quad G_w(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}$$



Die Übertragungsfunktionen  $G_0(s)$  und  $G_w(s)$  des Regelkreises mit dem Terminator unterscheiden sich nicht von Übertragungsfunktionen des Regelkreises ohne Terminator.

Das Verhalten des Regelkreises beim ANFAHREN wird vom Terminator nicht beeinflusst, weil es für die Übertragungsfunktion  $G_T(s)$  des Terminators beim ANFAHREN gilt:

$$G_T(s) = 1$$

## 2 Mathematische Herleitung des Terminator-Algorithmus

**HALTEN mit Terminator:** Führungsgröße  $w = x_{\text{soll}}$ ; Störgröße  $z = z_0$

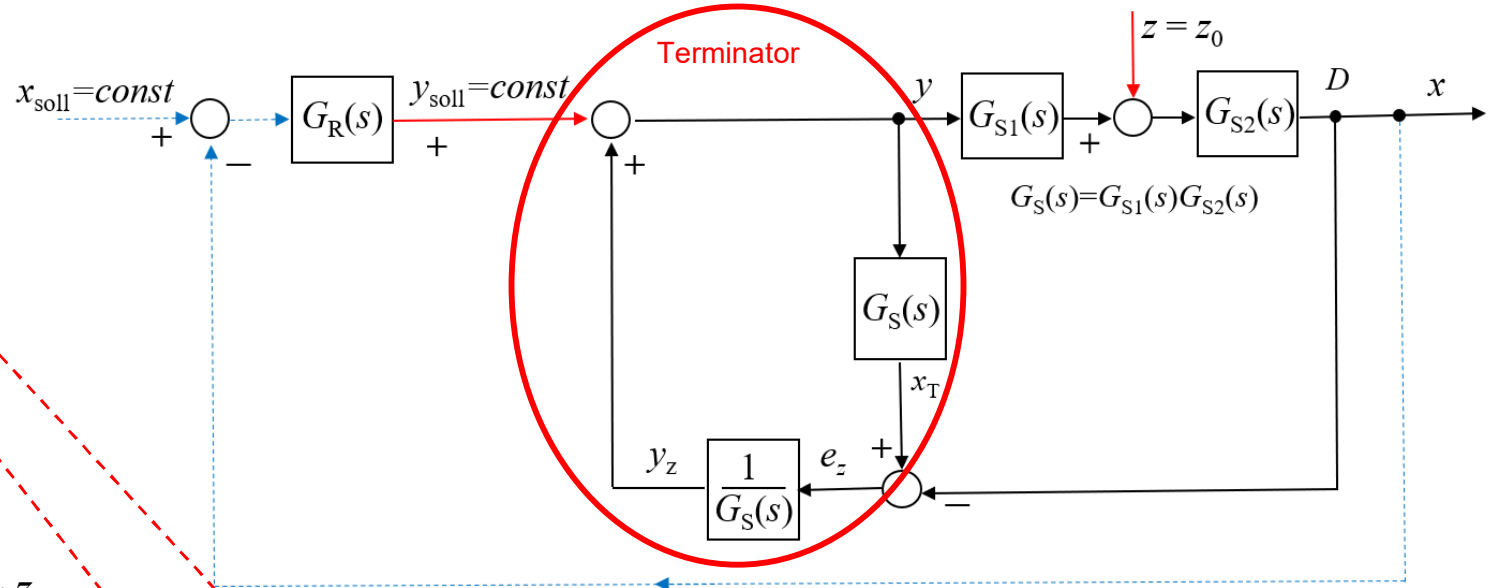
$$(1) \quad y_R = G_R \cdot e = y_{\text{soll}}$$

$$(2) \quad y = y_R + y_z$$

$$(3) \quad y_z = \frac{1}{G_S} \cdot e_z$$

$$(4) \quad x = G_S \cdot y + G_{S2} \cdot z_0$$

$$(5) \quad e_z = G_S \cdot y - x$$



$$y = \frac{e_z + x}{G_S}$$

$$y = \frac{x - G_{S2} \cdot z_0}{G_S}$$

$$\frac{e_z + x}{G_S} = \frac{x - G_{S2} \cdot z_0}{G_S}$$

$$e_z = -G_{S2} \cdot z_0$$

$$y_z = \frac{1}{G_S} \cdot e_z = -\frac{1}{G_{S1} \cdot G_{S2}} \cdot G_{S2} \cdot z_0 = -\frac{1}{G_{S1}} \cdot z_0$$

$$y = y_R - \frac{1}{G_{S1}} \cdot z_0$$

$$x = G_S \cdot y + G_{S2} \cdot z_0 = G_S \cdot \left( y_R - \frac{1}{G_{S1}} \cdot z_0 \right) + G_{S2} \cdot z_0$$

$$x = G_S \cdot y_R - \frac{G_S \cdot G_{S2}}{G_{S1}} \cdot z_0 + G_{S2} \cdot z_0$$

$$x = G_S \cdot y_R$$

**Die Störung  $z_0$  verschwindet!**

Der Regelkreis mit der Störung  $z = z_0$

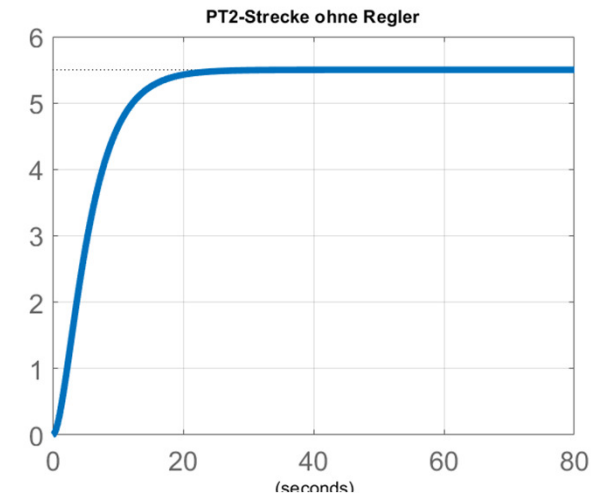
verhält sich genauso, wie der Kreis ohne Störung:  $x = G_S \cdot y_{\text{soll}} = x_{\text{soll}}$

# 3 Terminator für Regelstrecken ohne Totzeiten

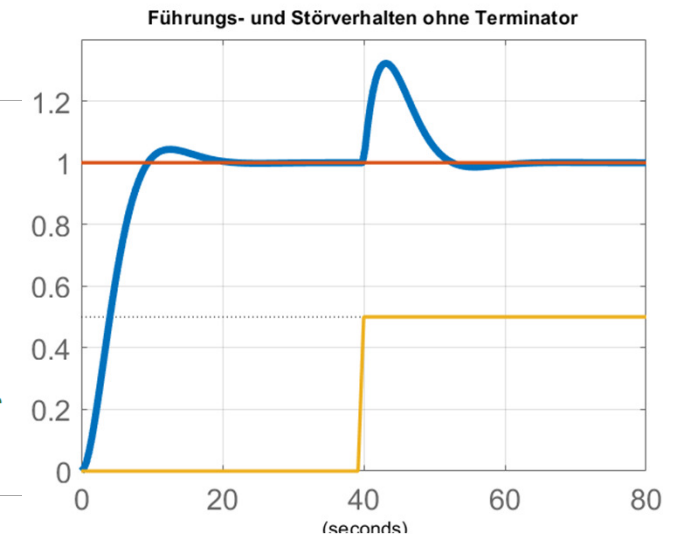
## 3.1 MATLAB-Skript: PI-Regler für PT2-Strecke ohne Tt

Terminator\_PT2\_ohne\_Tt.m

```
1 % Terminator für PT2-Strecke, Stand 30.12.2021
2 % Quelle: S.Zacher: "Regelungstechnik mit Data Stream Management", 2021
3 % 2021 Copyright Dr.Zacher, www.szacher.de
4 %*****
5 %% 1. Regelstrecke und PI-Regler
6 clearvars; clc; % Bildschirm bzw. Workspace säubern
7 s = tf('s'); % Laplace-Operator
8 Kps=5.5;T1=4; T2=2; % Streckenparameter
9 Gs1=Kps/(1+s*T1); % Gs1:Teilstrecke vor der Störung
10 Gs2=1/(1+s*T2); % Gs2:Teilstrecke nach der Störung
11 Gs=Gs1*Gs2; % Gs: gesamte Strecke
12 step(Gs,80);title('PT2-Strecke ohne Regler'); grid
13 Tn=T1; KpR=Tn/(2*Kps*T2) % Tn, KpR nach Betragsoptimum
14 GR=KpR*(1+s*Tn)/(s*Tn); % GR: PI-Regler
```



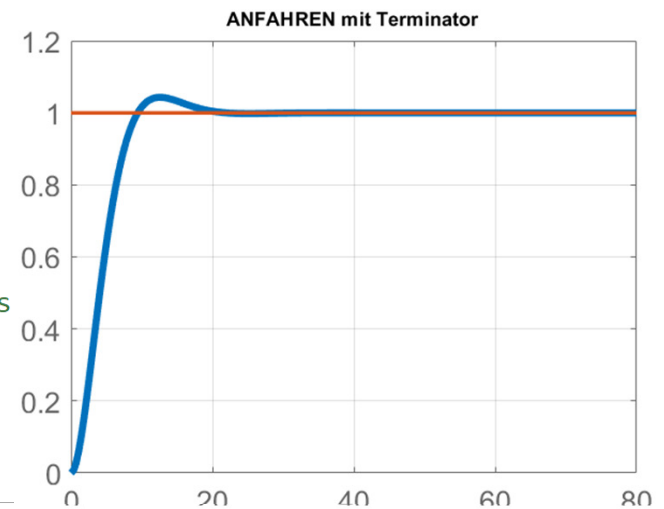
```
15 %% 2. Regelung: Führungs- und Störverhalten ohne Terminator
16 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
17 z=0.5*exp(-40*s); % z: Störung z=0.5 bei t=40 sec
18 Gvw=GR*Gs; % Gvw: Vorwärts-Übertragungsfunktion für w
19 Gvz=Gs2; % Gvz: Vorwärts-Übertragungsfunktion für z
20 G0=GR*Gs; % G0: offener Regelkreis ohne Terminator
21 Gw=Gvw/(1+G0); % Gw: Führungsverhalten ohne Terminator
22 Gz=Gvz/(1+G0); % Gz: Störverhalten ohne Terminator
23 G=w*Gw+z*Gz; % G: Führungs- und Störverhalten ohne Terminator
24 figure
25 step(G,w,z,80); title('Führungs- und Störverhalten ohne Terminator'); grid
```



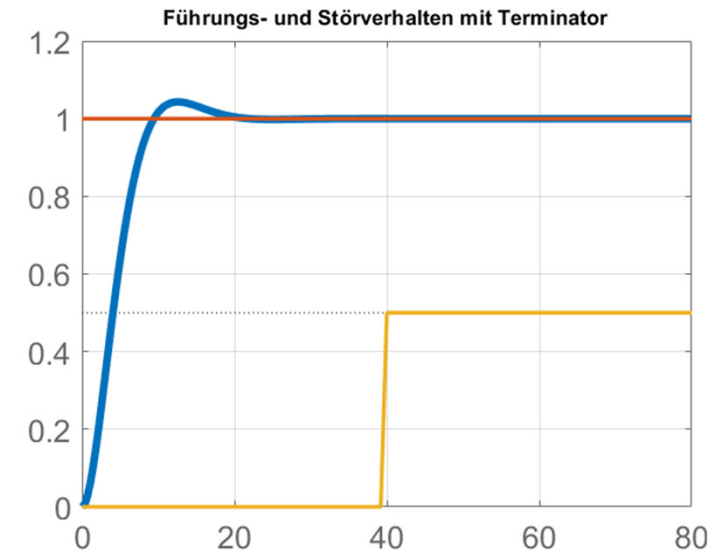
### 3 Terminator für Regelstrecken ohne Totzeiten

#### 3.1 MATLAB-Skript: PI-Regler für PT2-Strecke ohne Tt

```
26 %% 3. Regelung: Führungsverhalten mit Terminator ohne Störung
27 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
28 Gs_inv=1/Gs; % Gs_inv: ideale inverse Strecke
29 % Gs_inv=(0.01*s+1)*(0.01*s+1)/(Kps*(T1*s+1)*(T2*s+1)); % reale inverse Gs
30 GT=1/(1+Gs_inv*(Gs-Gs)); % GT: Terminator GT=1;
31 Gvw=GR*GT*Gs; % Gvw: Vorwärts-Übertragungsfunktion für w
32 G0=GR*GT*Gs; % G0: offener Regelkreis mit Terminator
33 Gw=Gvw/(1+G0); % Gw: geschlossener Regelkreis mit Terminator
34 step(w*Gw,w,80); title('ANFAHREN mit Terminator'); grid
```

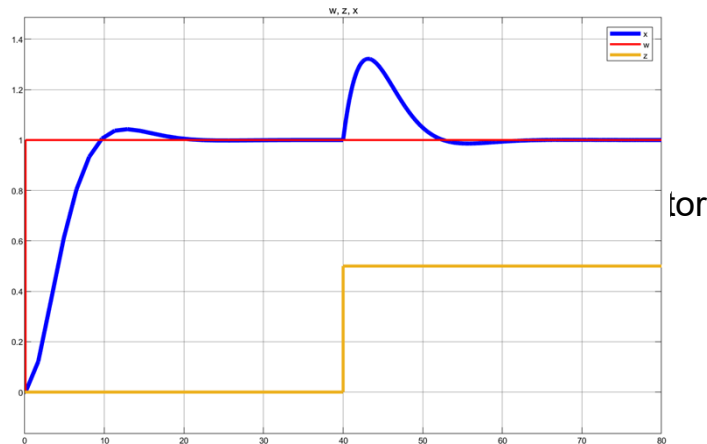
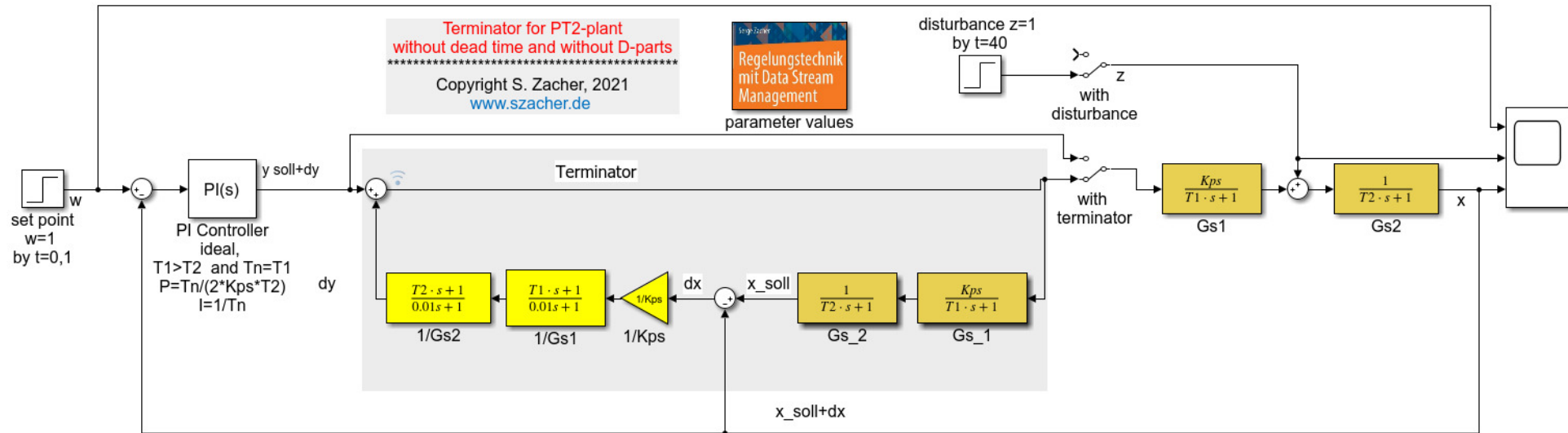


```
35 %% 4. Regelung: Führungs- und Störverhalten mit Terminator
36 hold off
37 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
38 z=0.5*exp(-40*s); % z: Störung z=5 bei t=40 sec
39 Gs_inv=1/Gs; % Gs_inv: ideale inverse Strecke
40 % Gs_inv=(0.01*s+1)*(0.01*s+1)/(Kps*(T1*s+1)*(T2*s+1)); % reale inverse Gs
41 GT=1/(1+Gs_inv*(Gs-Gs)); % GT: Terminator GT=1;
42 Gvw=GR*GT*Gs; % Gvw: Vorwärts-Übertragungsfunktion für w
43 GvzT=Gvz*(1-Gs*Gs_inv); % Gvz: Vorwärts-Übertragungsfunktion für z
44 G0=GR*GT*Gs; % G0: offener Regelkreis mit Terminator
45 Gw=Gvw/(1+G0); % Gw: Führungsverhalten mit Terminator
46 GzT=GvzT/(1+G0); % Gz: Störverhalten mit Terminator
47 GT=w*Gw+z*GzT; % G: Führungs- und Störverhalten mit Terminator
48 step(GT,w,z,80); title('Führungs- und Störverhalten mit Terminator'); grid
```

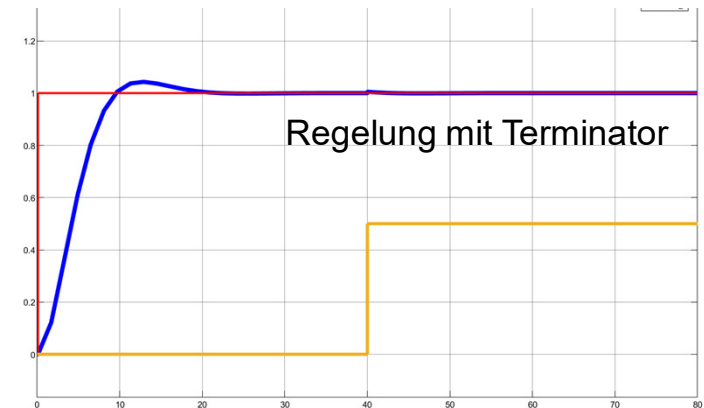
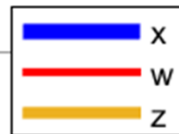


Im obigen Beispiel wurde eine ideale inverse Strecke  $G_{S\_inv}(s)$  benutzt (Zeile 39 des Skriptes). Dabei wird  $G_{vzT}=0$  sein (Zeile 43) und die Störung  $z$  wird in der Zeile 46 komplett vernichtet: es wird  $G_{zT}=0$ . Jedoch sind solche Übertragungsfunktionen wie  $G_{S\_inv}(s)$ , bei denen die Ordnung des Zählerpolynoms größer als die Ordnung des Nennerpolynoms ist, bekanntlich nicht realisierbar. Es soll die reale inverse Strecke nach Zeile 40 simuliert werden. Trotzdem kann es mit der Realisierung des Skriptes nach Zeile 40 nachgewiesen werden (hier nicht gezeigt), dass die Störgröße – wenn auch nicht komplett – verschwindet. Bei Regelstrecken mit Totzeiten und D-Anteilen (siehe Abschnitt 4) sollen doch zusätzliche Maßnahmen vorgenommen werden, um die Störungen besser zu beseitigen.

### 3.2 Simulink: PI-Regler für eine PT2-Strecke ohne Tt



$Kps = 5.5;$   
 $T1 = 4;$   
 $T2 = 2;$   
 $Tn = T1;$   
 $KpR = 0.1818;$



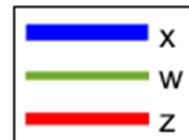
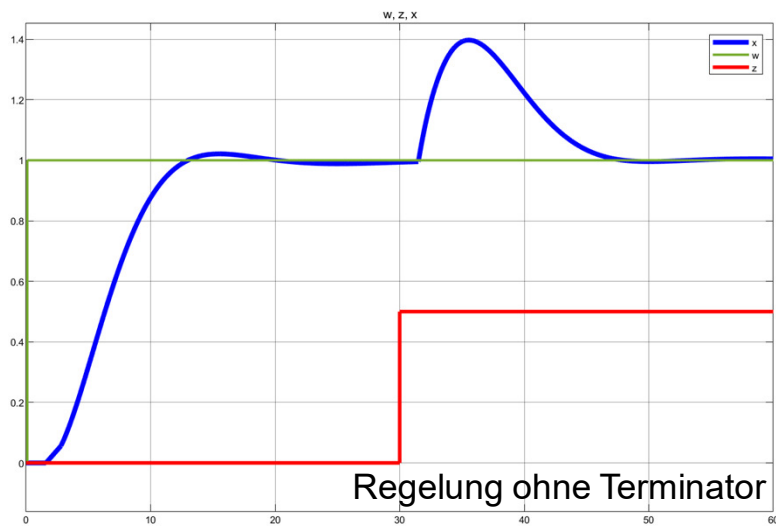
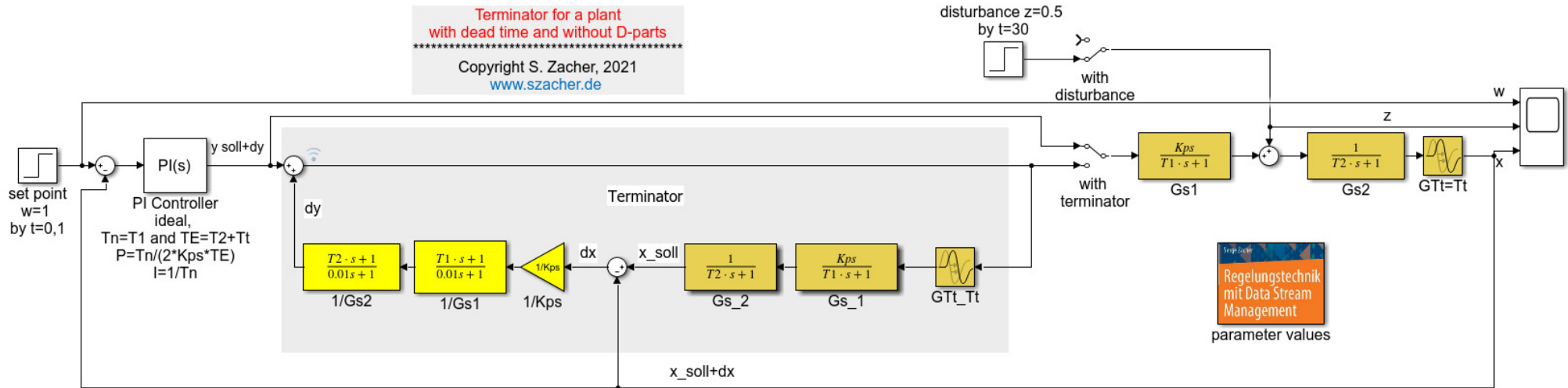
Oben wurden Beispiele von „ruhigen“ Industriestrecken, wie PT2-Glieder ohne D-Anteilen und ohne Totzeiten, mit großen Zeitkonstanten betrachtet, bei denen ein Terminator komplett aus regelungstechnischen Gliedern aufgebaut werden und dabei auch komplett die Störgrößen beseitigen kann (vgl. [1] - [4]).

# 4 Terminator für Regelstrecken mit D-Anteilen und Totzeiten

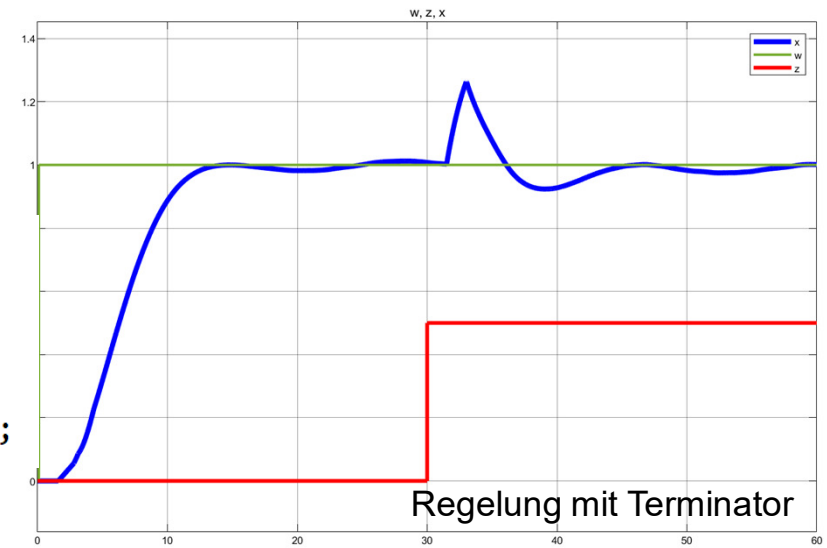
Bei „aggressiven“ Regelstrecken mit eigenen D-Anteilen, Totzeiten und komplexen Polen der charakteristischen Gleichung reichen allein die regelungstechnischen Gliedern nicht aus, wie unten an einigen Beispielen gezeigt wird.

## 4.1 Simulink: PI-Regler für PT2-Strecke mit Totzeit

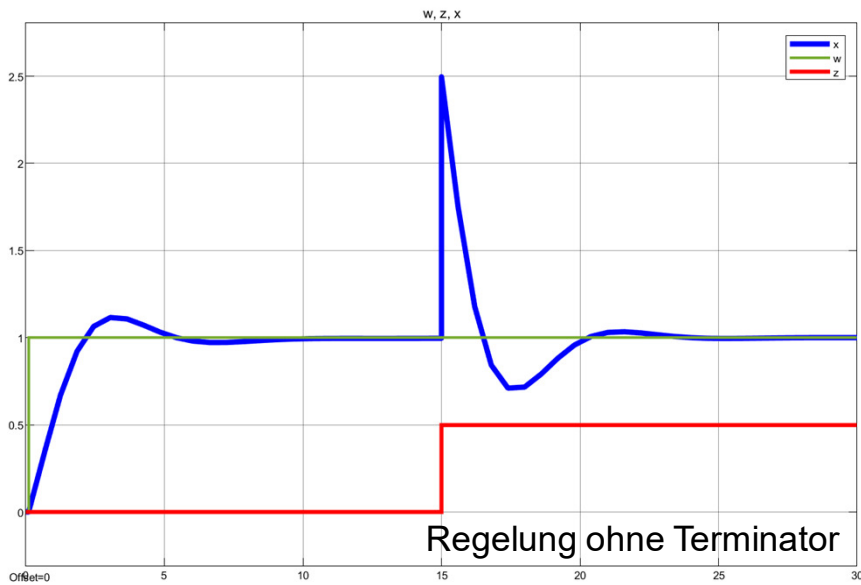
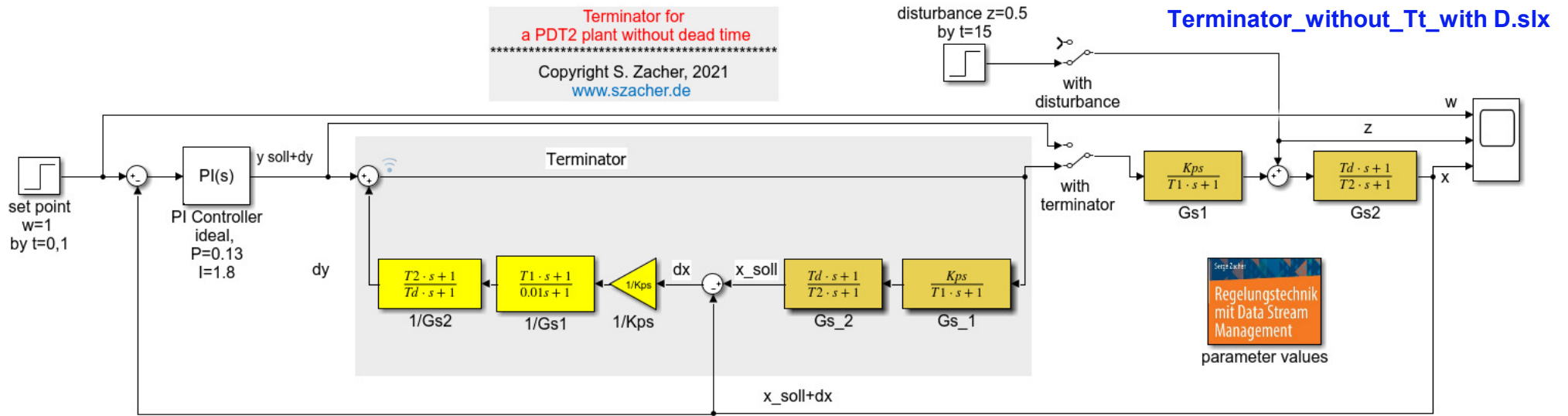
Terminator\_with\_Tt\_wirhout D.slx



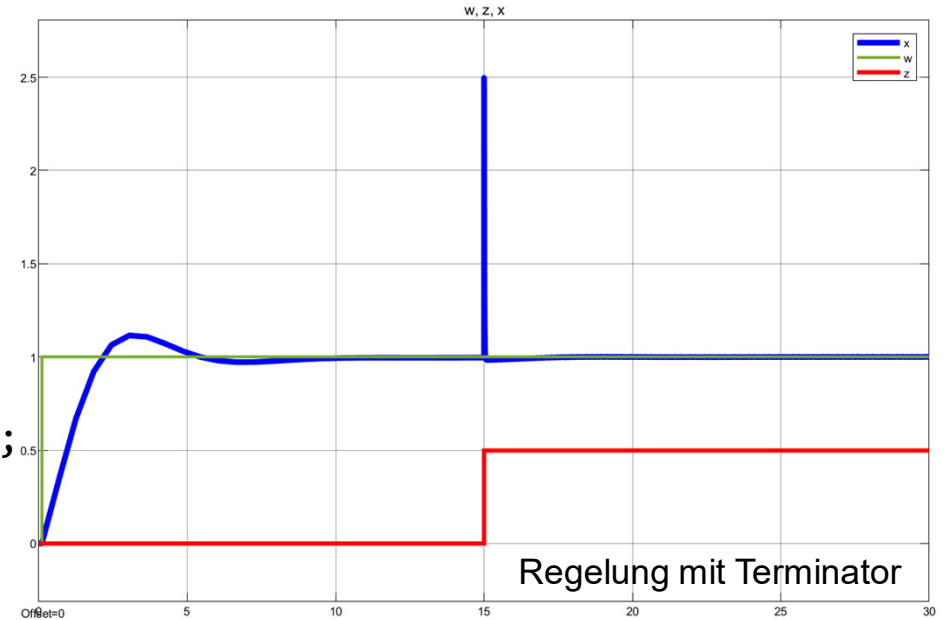
$K_{ps}=5.5;$   
 $T_1=4;$   
 $T_2=2;$   
 $T_t=1.5;$   
 $T_n=5;$   
 $K_{pR}=0.14;$



## 4.2 Simulink: PI-Regler für eine „aggressive“ PDT2-Strecke ohne Tt



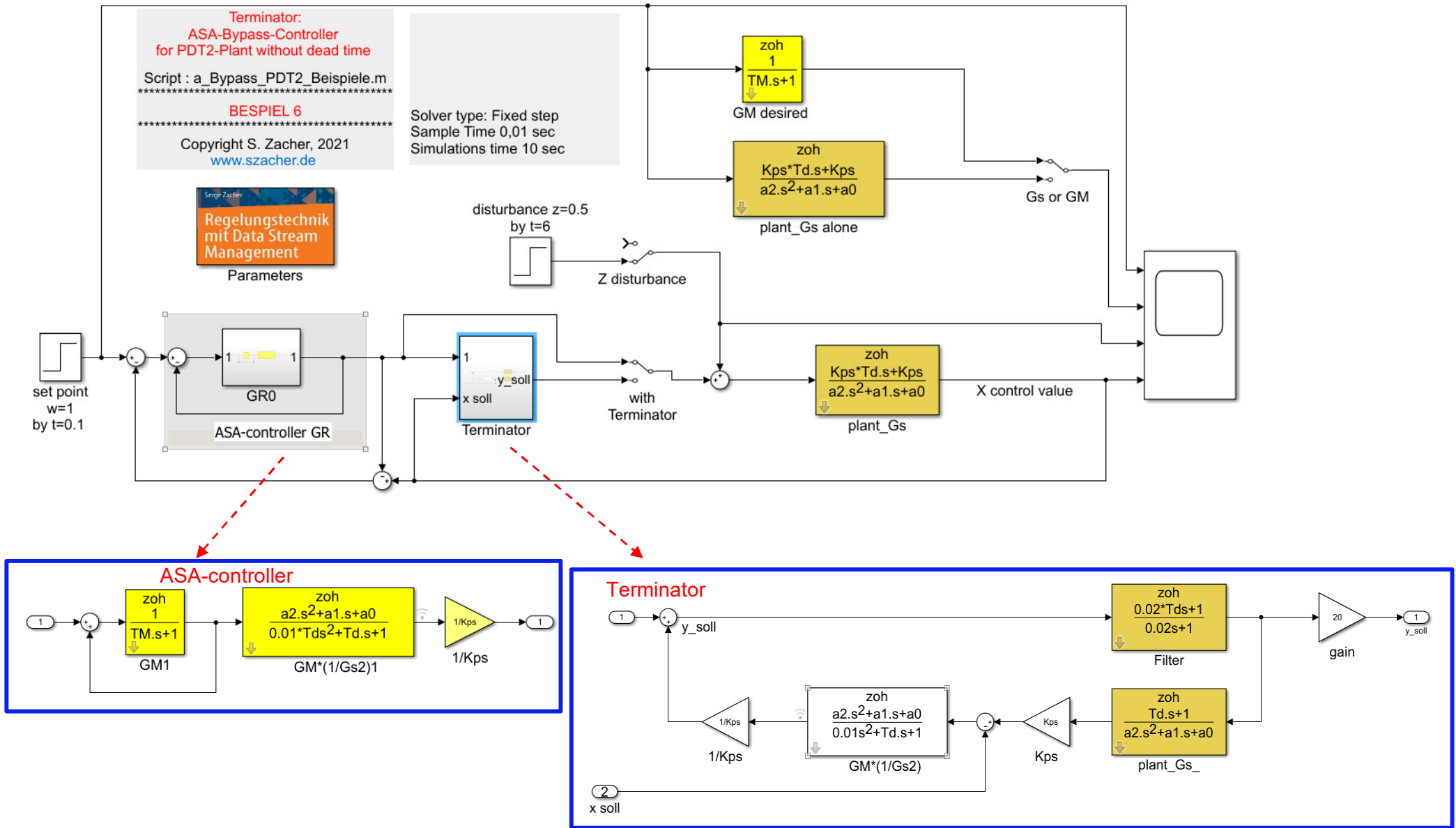
$K_{ps}=5.5;$   
 $Td=6;$   
 $T1=4;$   
 $T2=2;$   
 $Tn=T1;$   
 $KpR=0.18;$





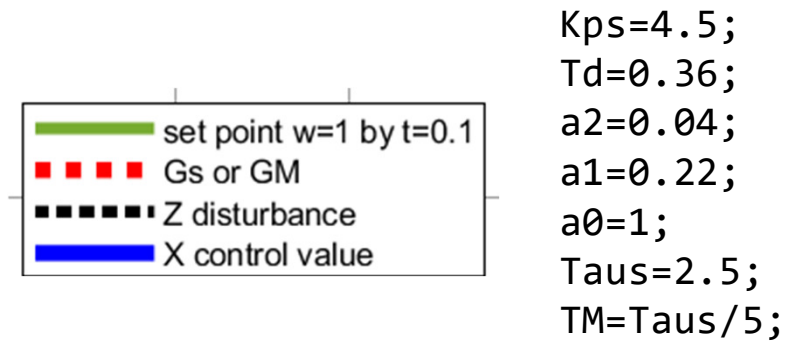
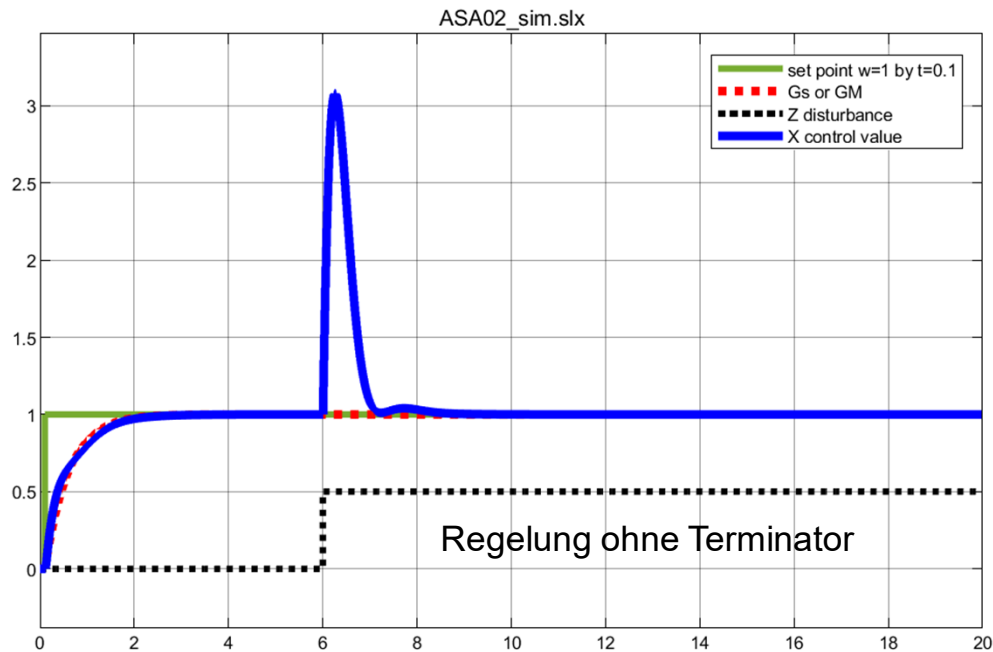
### 4.3 Simulink: ASA-Regler mit Bypass für eine PDT2-Strecke ohne Tt

Terminator\_06\_ASA\_PDT2\_ohneTt\_schwing.slx



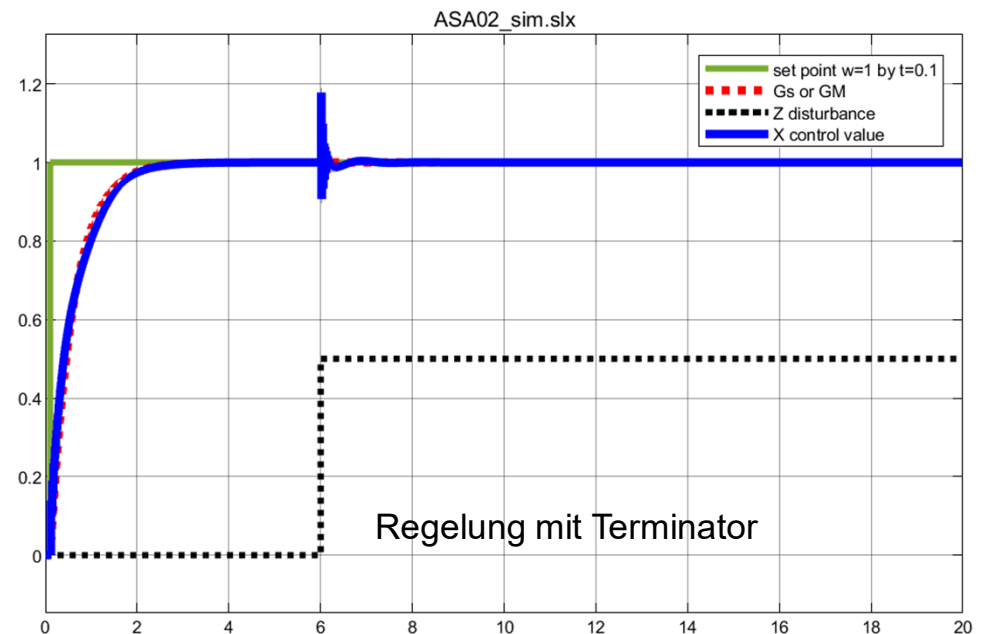
### 3 Terminator für Regelstrecken ohne Totzeiten

#### 4.3 Simulink: ASA-Regler mit Bypass für PDT2-Strecke ohne Tt



Aus obigen Beispielen von „aggressiven“ Regelstrecken, d.h. von Strecken mit eigenen D-Anteilen, Totzeiten und komplexen Polen der charakteristischen Gleichung, ist es ersichtlich, dass die unvorhergesehenen Störungen allein mit regelungstechnischen Gliedern komplett nicht beseitigt werden können.

Es sollen die Steuerungs-Elemente einbezogen werden, wie nachfolgend behandelt wird. Trotzdem kann die am Eingang der PD-Strecke plötzlich wirkende Störung auch damit nicht komplett beseitigt werden.



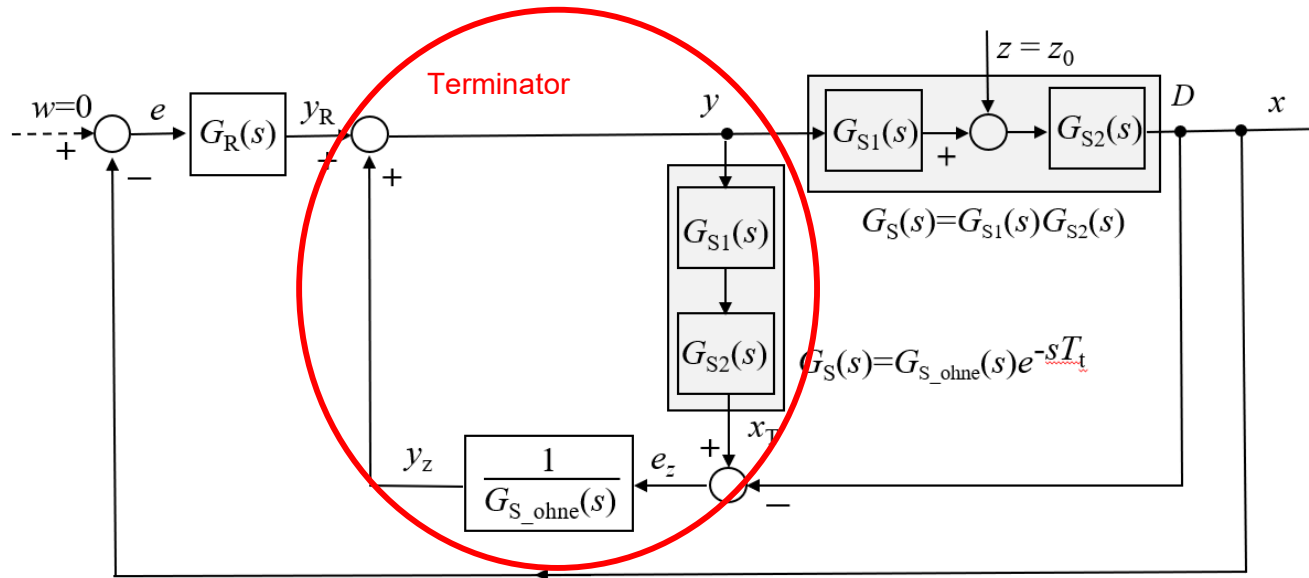
## 4.4 Terminator-Algorithmus für Strecken mit Totzeiten

Nehmen wir an, dass die Übertragungsfunktion der Regelstrecke  $G_S(s)$  bestehend aus einer Teilstrecke ohne Totzeit  $G_{S\_ohne}(s)$  und einer Totzeit  $T_t$  gegeben ist:  $G_S(s) = G_{S\_ohne}(s) \cdot e^{-sT_t}$

Unbekannt ist die Höhe der Störgröße  $z = z_0$  sowie der Stelle, zu welcher die Störung eingegeben wird. In anderen Worten, bekannt ist, dass es

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

gilt (Bild unten), aber die Übertragungsfunktionen der Teilstrecken vor der Störung  $G_{S1}(s)$  und nach der Störung  $G_{S2}(s)$  sind nicht gegeben.



$$G_{S1}(s) = e^{-sT_{t1}}$$

$$G_{S2}(s) = G_{S2\_ohne}(s)e^{-sT_{t2}}$$

$$T_t = T_{t1} + T_{t2}$$

$$G_{S\_ohne}(s) = G_{S2\_ohne}(s)$$

**ANFAHREN mit Terminator:** Führungsgröße  $w = x_{soll}$ , Störgröße  $z = 0$ .

Da beim ANFAHREN keine Störgröße wirkt, sind die Übertragungsfunktion des Terminators  $G_T(s)$  und die Übertragungsfunktion  $G_w(s)$  des Regelkreises bei Strecken mit oder ohne Totzeiten identisch bzw. wie im Abschnitt 3 bereits hergeleitet wurde:

$$G_T(s) = 1$$

$$G_0(s) = G_R(s)G_T(s)G_S(s) = G_R(s)G_S(s)$$

$$G_w(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{G_R(s)G_S(s)}{1 + G_R(s)G_S(s)}$$

**HALTEN mit Terminator:** Führungsgröße  $w = 0$ ; Störgröße  $z = z_0$ .

Unten ist gezeigt, dass die Störgröße vom Terminator nicht kompensiert wird, weil es nicht möglich ist, eine Inverse für die Strecke mit Totzeit zu bilden:

$$\frac{1}{G_S} = \frac{1}{G_{S\_ohne} \cdot e^{-sT_t}} = \frac{1}{G_{S\_ohne}} \cdot e^{+sT_t}$$

(1)  $y_R = G_R \cdot e = y_{soll}$

(2)  $y = y_R + y_z = y_z$

(3)  $y_z = \frac{1}{G_{S\_ohne}} \cdot e_z$

(4)  $x = G_S \cdot y + G_{S2} \cdot z_0$

(5)  $e_z = G_S \cdot y - x$

$$y = \frac{e_z + x}{G_S}$$

$$y = \frac{x - G_{S2} \cdot z_0}{G_S}$$

$$\frac{e_z + x}{G_S} = \frac{x - G_{S2} \cdot z_0}{G_S}$$

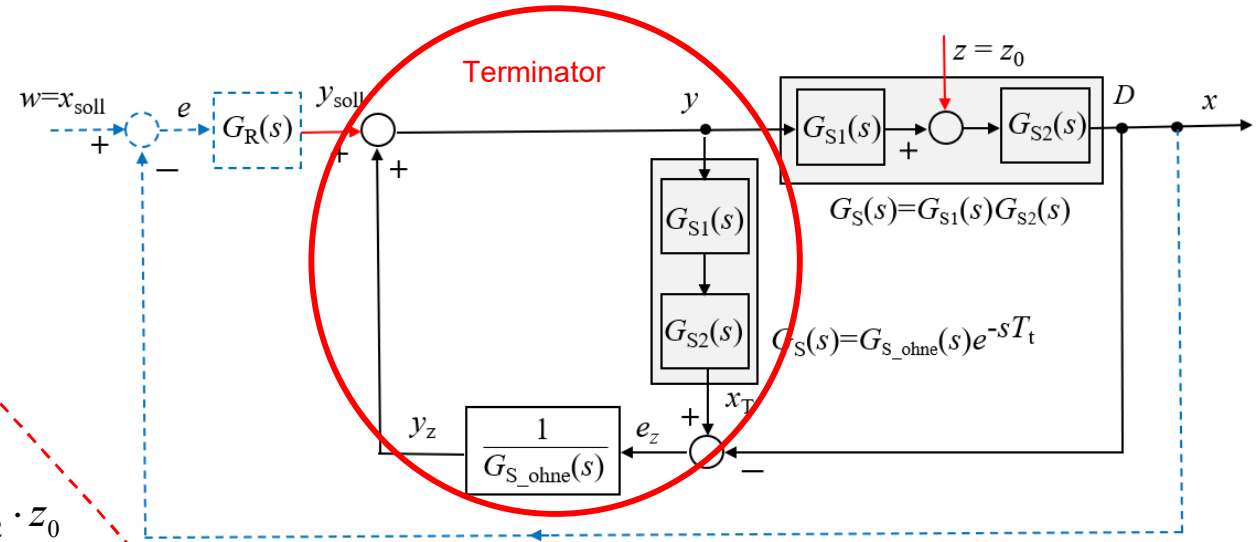
$$e_z = -G_{S2} \cdot z_0$$

$$y_z = \frac{1}{G_{S\_ohne}} \cdot e_z = -\frac{1}{G_{S2\_ohne}} \cdot G_{S2} \cdot z_0$$

$$y_z = -\frac{G_{S2\_ohne} \cdot e^{-sT_{t2}}}{G_{S2\_ohne}} \cdot z_0$$

$$y_z = -e^{-sT_{t2}} \cdot z_0$$

$$y = y_R - e^{-sT_{t2}} \cdot z_0$$



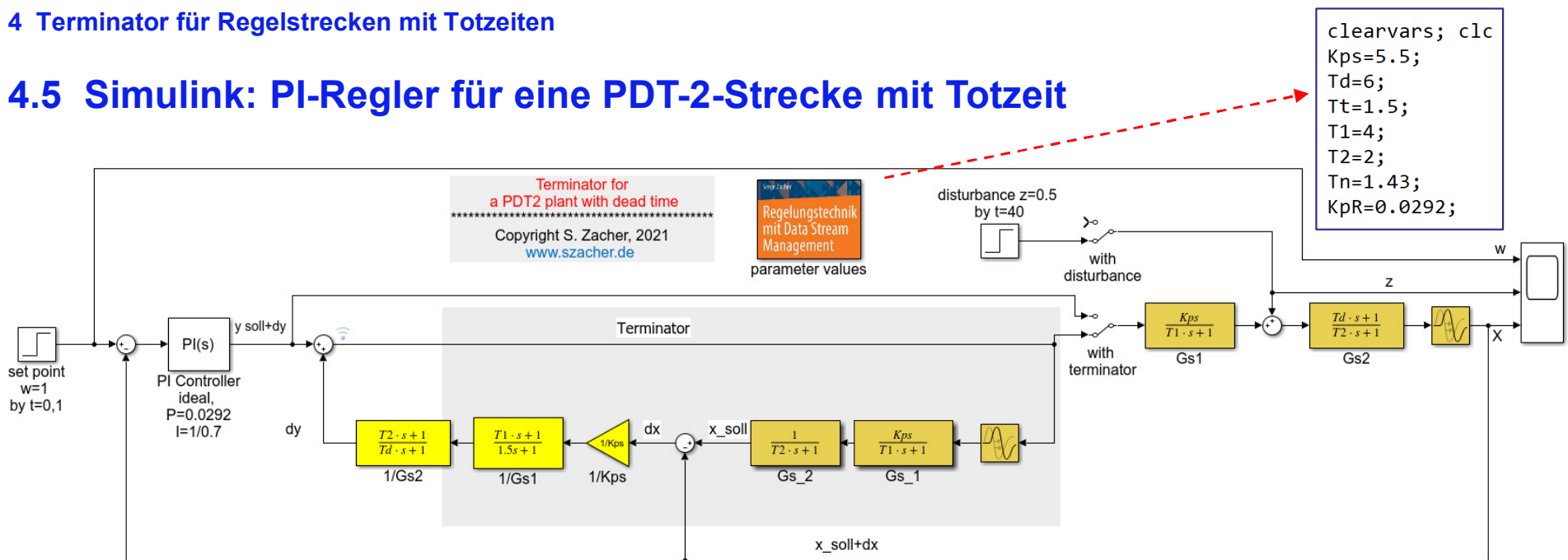
$$x = G_S \cdot y + G_{S2} \cdot z_0 = G_S \cdot (y_R - e^{-sT_{t2}} \cdot z_0) + G_{S2} \cdot z_0$$

$$x = G_S \cdot y_{soll} - G_S \cdot e^{-sT_{t2}} \cdot z_0 + G_{S2} \cdot z_0$$

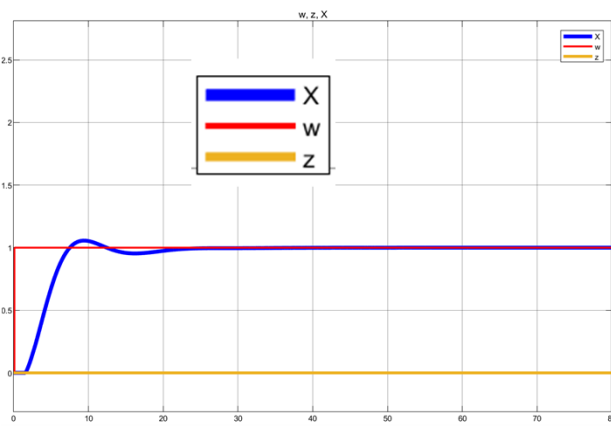
$$x = G_S \cdot y_R - G_{S2} (G_{S1} \cdot e^{-sT_{t2}} + 1) \cdot z_0$$

Die Störung  $z_0$  verschwindet nicht bzw. der Terminator ist wirkungslos, so dass der Terminator-Algorithmus modifiziert werden soll (siehe Abschnitt 5).

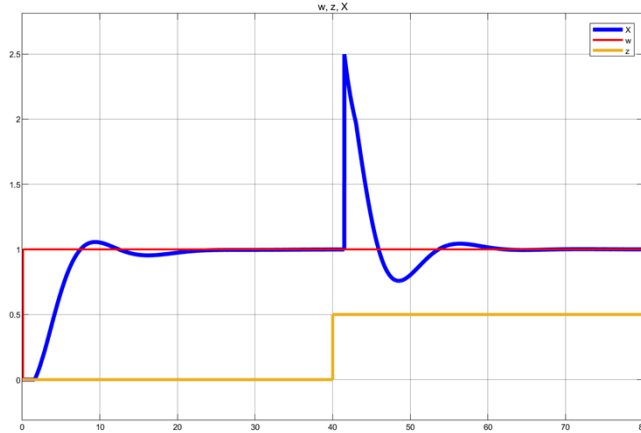
### 4.5 Simulink: PI-Regler für eine PDT-2-Strecke mit Totzeit



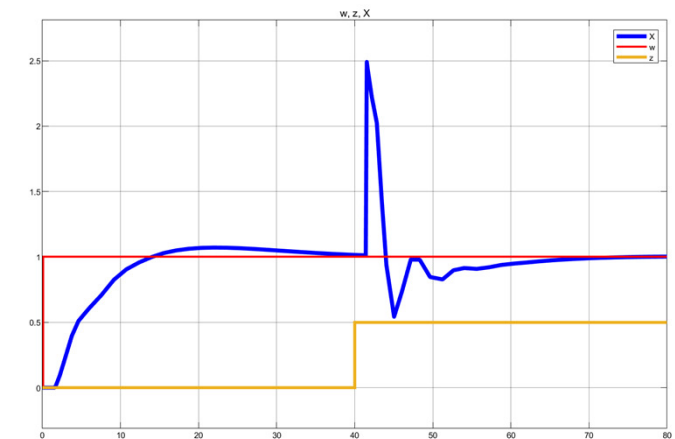
Terminator\_with\_Tt\_with D.slx



Regelung ohne Terminator  
ohne Störung



Regelung ohne Terminator  
mit Störung



Regelung mit Terminator  
mit Störung

Der Terminator ist wirkungslos, so dass der Terminator-Algorithmus für PDT2-Strecke mit Totzeit modifiziert werden soll (siehe nächste Seite).

## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

Ausschlaggebend für die nachfolgend angebotene Terminator-Option, die *Schubert-Terminator* genannt wird (siehe Vorwort), ist der im vorherigen Abschnitt festgestellte Zusammenhang:

$$e_z = -G_{S2} \cdot z_0$$

Die Variable  $e_z$  liefert die Information über die Größe der bislang unbekanntenen Störung  $z_0$  und über den Zeitpunkt, wenn die Störgröße wirkt, sowie über die Steckenteil  $G_{S2}(s)$ , die vor der Störgröße liegt.

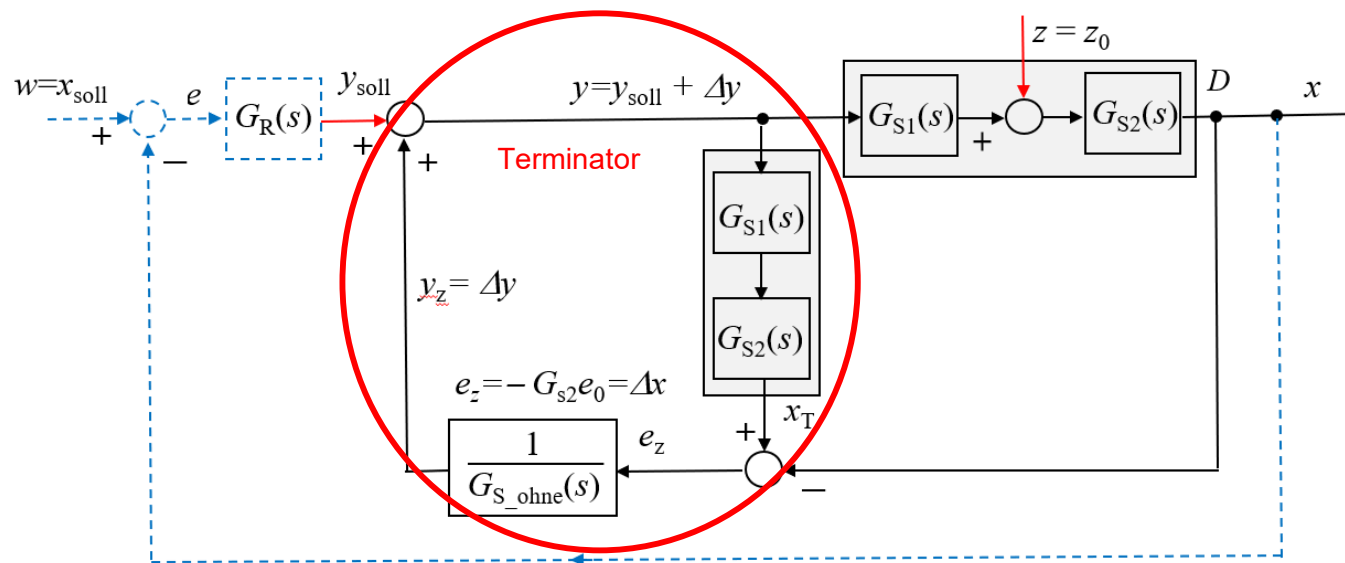
Vergessen wir nicht, dass die Übertragungsfunktion  $G_S(s)$  der gesamten Strecke und die gesamte Totzeit  $T_t = T_{t1} + T_{t2}$  gegeben sind:

$$G_S = G_{S1} \cdot G_{S2} \quad G_{S1}(s) = e^{-sT_{t1}} \quad G_{S2} = G_{S\_ohne} \cdot e^{-sT_{t2}}$$

Die Variable  $e_z$  ist somit nichts anderes als ein Beobachter der Störung und wird nachfolgend Z-Observer genannt. Man kann den Z-Observer für die Störgrößenkompensation unterschiedlich nutzen.

### 5.1 Steuerung des Soll-Stellwertes $y_{soll}$ (open loop control)

Nachdem der Sollwert  $x_{soll}$  beim ANFAHREN-Phase erreicht wird, soll der Regler bei der nachfolgenden HALTEN-Phase von „Automatik“ auf „Halt“-Betrieb umgeschaltet werden. Der letzte  $y$ -Stellwert bzw.  $y_{soll}$  wird konstant gehalten (Bild unten).



## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.1 Steuerung des Soll-Stellwertes $y_{\text{soll}}$ (open loop control)

Es gilt beim HALTEN ohne Störung bzw. bei  $z_0 = 0$  im geregelten stabilen Zustand bei  $t = \infty$  :

$$x_{\text{soll}}(\infty) = K_{\text{PS1}} \cdot K_{\text{PS2}} \cdot y_{\text{soll}}(\infty) \implies y_{\text{soll}}(\infty) = \frac{x_{\text{soll}}(\infty)}{K_{\text{PS1}} \cdot K_{\text{PS2}}}$$

Mit der Störung  $z_0$  gilt es im geregelten stabilen Zustand bei  $t = \infty$  :

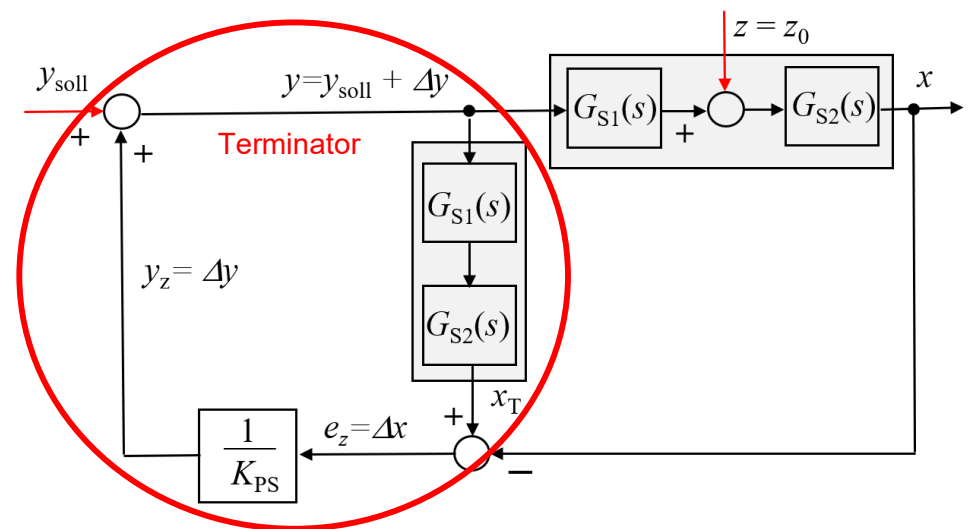
$$x_{\text{soll}}(\infty) = K_{\text{PS1}} \cdot K_{\text{PS2}} \cdot (y_{\text{soll}} + \Delta y) + K_{\text{PS2}} \cdot z_0$$

Der Zuwachs  $\Delta x$  wird über den Z-Observer bestimmt:

$$\Delta x = e_z = K_{\text{PS2}} \cdot z_0$$

Daraus resultiert der Steuerung-Algorithmus:

$$x_{\text{soll}}(\infty) - e_z = K_{\text{PS1}} \cdot K_{\text{PS2}} \cdot y_{\text{soll}}(\infty) + K_{\text{PS1}} \cdot K_{\text{PS2}} \cdot \Delta y \implies \Delta y = \frac{-e_z}{K_{\text{PS1}} \cdot K_{\text{PS2}}}$$



#### Beispiel:

Gegeben:

$$x_{\text{soll}}(\infty) = 1$$

$$K_{\text{PS1}} = 5,5$$

$$K_{\text{PS2}} = 1$$

$$y_{\text{soll}} = \frac{x_{\text{soll}}(\infty)}{K_{\text{PS1}} \cdot K_{\text{PS2}}} = 0,18$$

Nach dem Störsprung  $z_0 = 0,5$  wird  $e_z$  vom Z-Observer abgelesen:  $e_z = K_{\text{PS2}} \cdot z_0 = 0,5$

Der geforderte Zuwachs der Stellgröße :

$$\Delta y = \frac{-e_z}{K_{\text{PS1}} \cdot K_{\text{PS2}}} = \frac{-0,5}{5,5} = -0,0909$$

Wird die folgende Stellgröße

$$y_{\text{soll}} + \Delta y = 0,18 - 0,0909 = 0,0891$$

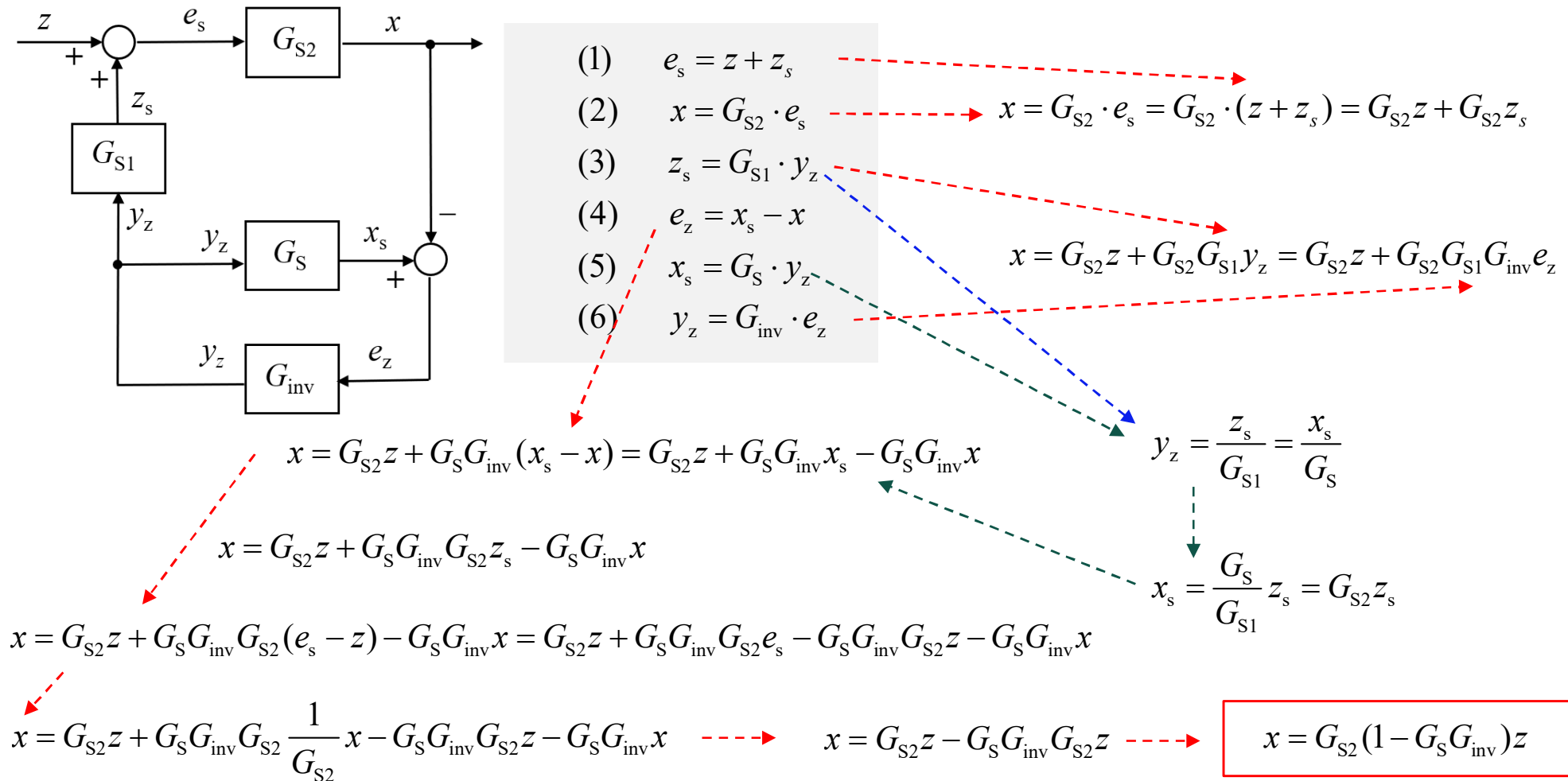
zum Streckeneingang eingegeben, wird die gestörte Regelgröße wieder den Sollwert  $x_{\text{soll}} = 1$  erreichen.



## 5.2 Regelung des Soll-Stellwertes $y_{\text{soll}}$ (closed loop control)

Wie bereit oben erwähnt wurde, wird der Regler bei der HALTEN-Phase von „Automatik“ auf „Halt“-Betrieb umgeschaltet. Der letzte  $y$ -Stellwert bzw.  $y_{\text{soll}}$  wird konstant gehalten.

Im Abschnitt 5.1 wurde das statische Verhalten des Regekreises bei  $t = \infty$  diskutiert. Nachfolgen wird das dynamische Verhalten des Regekreises betrachtet. Da der konstante Stellwert  $y_{\text{soll}}$  bei abgeschaltetem Regler keine Wirkung auf Dynamik hat, wird der Regelkreis beim HALTEN nur durch ein Eingangssignal beeinflusst, nämlich durch die Störung  $z_0$  (siehe Bild unten).



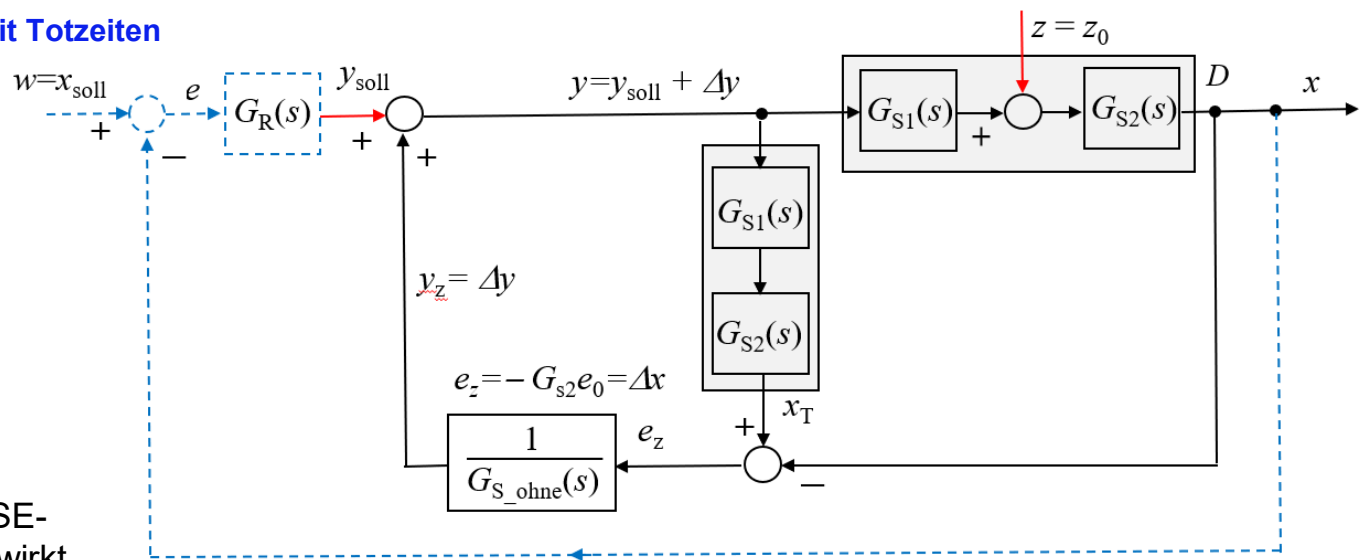
## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.2 Regelung des Soll-Stellwertes $y_{\text{soll}}$

Der Wirkungsplan des geschlossenen Regelkreises beim HOLD-Betrieb (bzw. HALTEN) wurde im Abschnitt 5.1 eingeführt (Bild rechts). Es gilt:

$$G_{\text{inv}}(s) = \frac{1}{G_{\text{S\_ohne}}(s)}$$

Nach dem Abschluss des RISE-Betriebes (bzw. ANFAHREN) wird der Regler  $G_{\text{R}}(s)$  blockiert und der letzte  $y$ -Stellwert des RISE-Betriebes  $y_{\text{soll}}$  wird konstant gehalten. Es wirkt sprungförmig nur die Störung  $z=z_0$ .



Im RISE-Betrieb wird die Stör-Übertragungsfunktion  $G_z(s)$  bekanntlich nach folgender Formel bestimmt:

$$G_z(s) = \frac{G_{\text{vz}}(s)}{1 + G_0(s)}$$

wobei die Übertragungsfunktionen des Terminators  $G_{\text{T}}(s)$  und des offenen Kreises  $G_0(s)$  sind:

$$G_{\text{T}}(s) = \frac{1}{1 + G_{\text{inv}}(s) \cdot [G_{\text{S}}(s) - G_{\text{S}}(s)]} \quad G_0(s) = G_{\text{R}}(s) G_{\text{T}}(s) G_{\text{S}}(s)$$

Im HOLD-Betrieb wird die Stellgröße des Reglers konstant gehalten bzw.  $G_0(s) = 0$ . Die Vorwärts-Übertragungsfunktion  $G_z(s)$ , die auf der vorherigen Seite bestimmt wurde, gilt als die Stör-Übertragungsfunktion  $G_z(s)$ :

$$G_{\text{vz}} = G_z = G_{\text{S}2} (1 - G_{\text{S}} G_{\text{inv}})$$

Daraus sind die Vorteile des Schubert-Terminators ersichtlich, nämlich:

- wenn  $G_{\text{S}}(s) = G_{\text{inv}}(s)$  ist, wird  $G_z(s) = 0$  bzw. die Störung hat keine Wirkung;
- wenn  $G_{\text{S}}(s) \neq G_{\text{inv}}(s)$  ist, wird  $G_z(s) \neq 0$ . Die Störung kann komplett nicht beseitigt werden. Jedoch wird die Störung wegen D-Anteilen in  $G_{\text{inv}}(s)$  schnell ausgegelt.

## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.2 Regelung des Soll-Stellwertes $y_{\text{soll}}$

Als Beispiel ist unten das MATLAB-Skript für die Strecke mit Totzeiten und D-Anteil gegeben:

$$G_{S1}(s) = \frac{K_{PS}}{1 + sT_1}$$

$$G_{S2}(s) = \frac{1 + sT_d}{1 + sT_2}$$

$$G_{Tt}(s) = e^{-sT_t}$$

$$K_{PS} = 5,5 \quad T_d = 6$$

$$T_1 = 4,0 \quad T_2 = 2$$

$$T_t = 1,5$$

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

$$G_{S_{\text{ohne}}}(s) = G_{S1}(s)G_{S2}(s)$$

$$G_{vz} = G_z = G_{S2}(1 - G_S G_{\text{inv}})$$

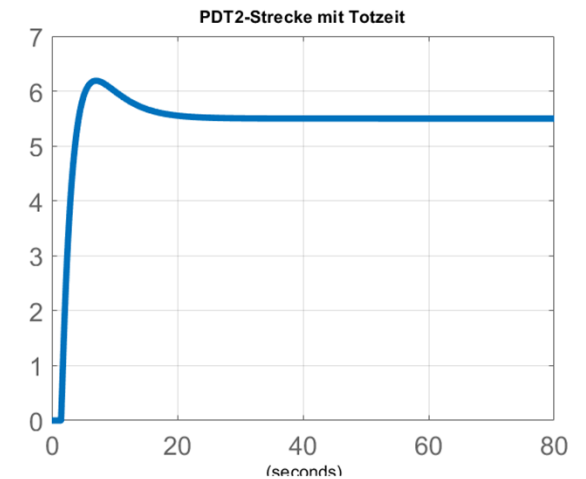
$$G_{\text{inv}}(s) = \frac{1}{G_{S_{\text{ohne}}}(s)}$$

Die Bedingung  $G_{\text{inv}}(s)=G_S(s)$  ist wegen Totzeit nicht erfüllt, d.h.  $G_z(s) \neq 0$ . Trotzdem wird die Störung mit dem Schubert-Terminator schneller beseitigt, als es mit dem Terminator der vorherigen Abschnitten 3 und 4 der Fall war.

### 5.3 MATLAB-Skript: PI-Regler für PDT2-Strecke mit Tt

Rise\_hold\_PDT2\_mit\_Tt.m

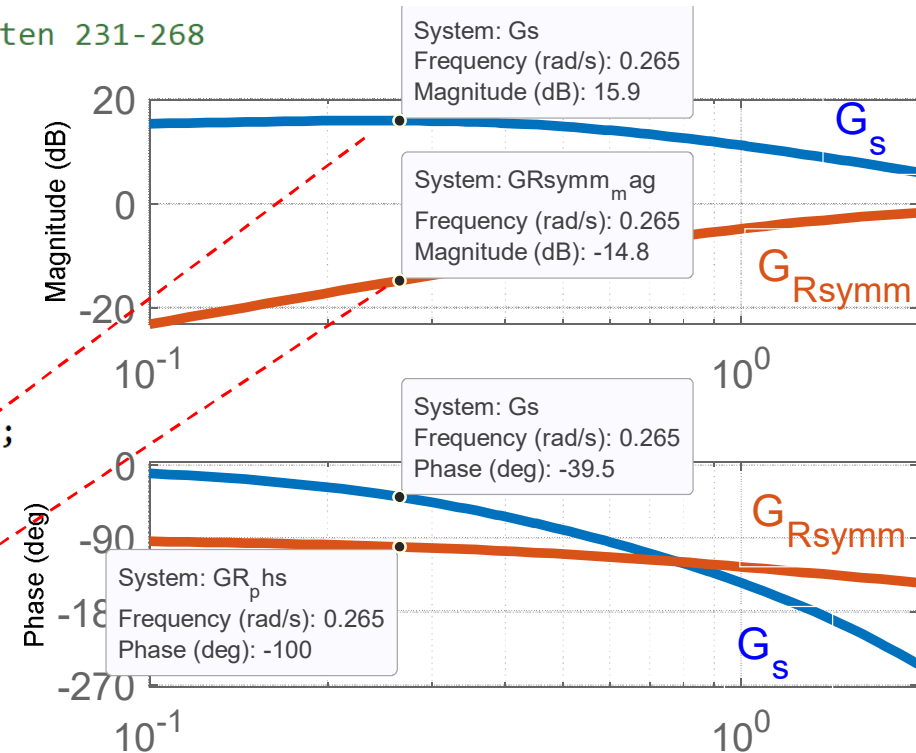
```
8 %% 1. Regelstrecke und PI-Regler
9 clearvars; clc; % Bildschirm bzw. Workspace säubern
10 s = tf('s'); % Laplace-Operator
11 Kps=5.5;T1=4; T2=2;Td=6; % Streckenparameter
12 Tt=1.5; % Tt: Totzeit
13 Gs1=Kps/(1+s*T1); % Gs1: Teilstrecke vor der Störung PT1
14 Gs2=(1+s*Td)/(1+s*T2); % Gs2: Teilstrecke nach der Störung DT1
15 GTt=s*exp(-s*Tt)/s; % GTt: Teilstrecke nach der Gs2 (Totzeit)
16 Gs_ohne=Gs1*Gs2; % Gs_ohne: gesamte Strecke ohne Totzeit
17 Gs=Gs1*Gs2*GTt; % Gs: gesamte Strecke ohne Totzeit
18 G_inv=(1/Kps)*(1+s*T1)*(1+s*T2)/((1+s*Td)*(0.01*s+1));% inverse ohne Tt
19 step(Gs,80);title('PDT2-Strecke mit Totzeit'); grid
20 KpR=1;Tn=0.7; % KpR,Tn: angenommene Parameter
21 GR=KpR*(1+s*Tn)/(s*Tn); % GR: PI-Regler
```



## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.3 MATLAB-Skript: PI-Regler für PDT2-Strecke mit Tt

```
22 %% 2. Reglereinstellung nach DBV
23 % gewünschte Phasenreserve 60°, Quelle [1], Seiten 231-268
24 GR=KpR*(1+s*Tn)/(s*Tn);
25 GRsymm_mag=1/GR;
26 GR_phs=1/(s^2*GR);
27 wmin=0.1;wmax=2;
28 bode(Gs,{wmin,wmax}); hold on;grid
29 subplot(211);
30 bodemag(Gs,{wmin,wmax}); hold on
31 opts_mag=bodeoptions('cstprefs');
32 opts_mag.PhaseVisible='off';
33 h_mag=bodeplot(GRsymm_mag,{wmin,wmax},opts_mag);
34 hold on;grid
35 subplot(212);
36 bode(Gs,{wmin,wmax}); hold on
37 opts_ph=bodeoptions('cstprefs');
38 opts_ph.MagVisible='off';
39 h_ph=bodeplot(GR_phs,{wmin,wmax},opts_ph);
40 hold on;grid
```

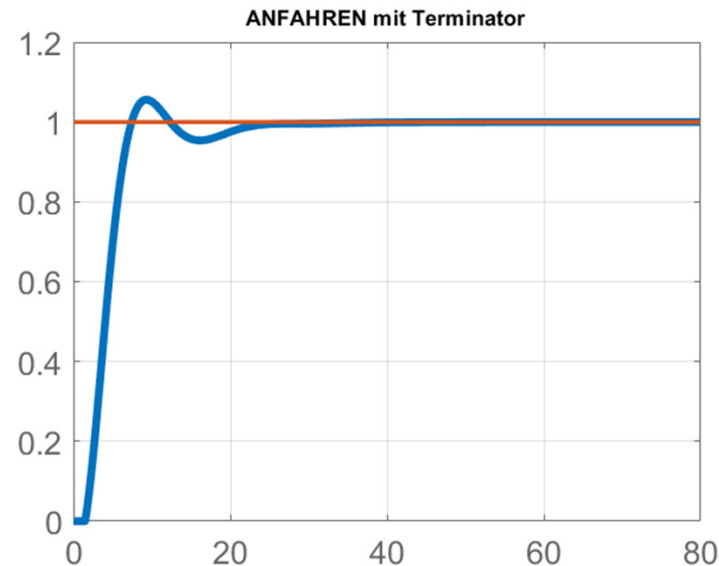
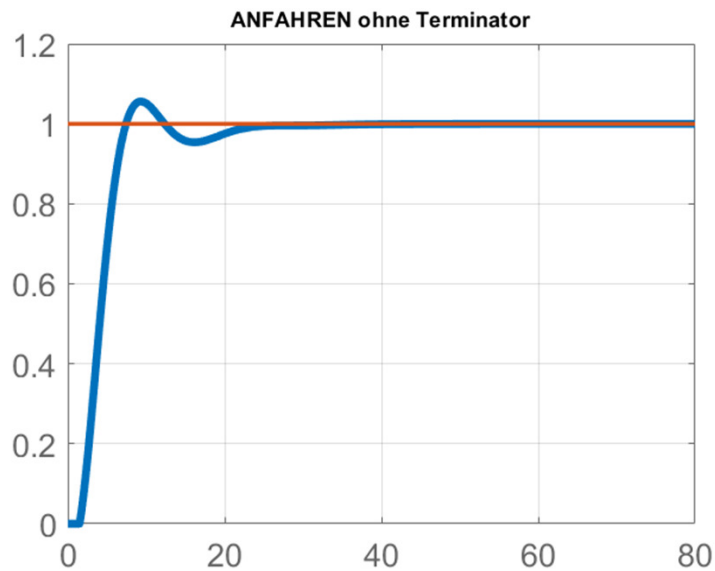


```
41 %% 3. Reglereinstellung:
42 % Verschiebung des Amplitudengangs in gewünschten Arbeitspunkt
43 delta_K=15.9+14.8; % Verschiebung im Bode-Plot bei omega 0.265 ablesen
44 KpR=10^(-delta_K/20); % delta_K von dB umrechnen -----> K_PR=0,0292
45 GR=KpR*(1+s*Tn)/(s*Tn); % Der PI-Regler mit dem neuen KpR-Wert
46 % falls nötig, die Sektion 2 mit dem neuen KpR-Wert wiederholen
```

## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.3 MATLAB-Skript: PI-Regler für PDT2-Strecke mit Tt

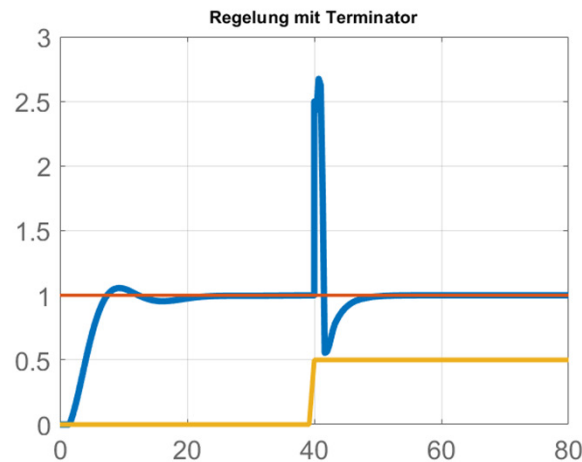
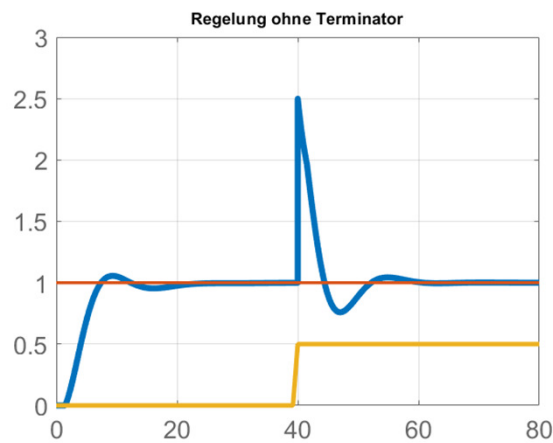
```
47 %% 4. Führungsverhalten ohne Terminator
48 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
49 G0=GR*Gs; % G0: offener Regelkreis ohne Terminator
50 Gw=G0/(1+G0); % Gw: geschlossener Regelkreis ohne Terminator
51 % Das Bild mit dem Bode-Plot vorher schließen
52 figure % das neue Fenster für Sprungantwort wird geöffnet
53 step(w*Gw,80); title('ANFAHREN ohne Terminator'); grid
54 %% 5. Führungsverhalten mit Terminator
55 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
56 GT=1/(1+G_inv*(Gs-Gs)); % GT: Terminator GT=1;
57 Gvw=GR*GT*Gs; % Gvw: Vorwärts-Übertragungsfunktion des Kreises für w
58 G0=GR*GT*Gs; % G0: offener Regelkreis mit Terminator
59 Gw=Gvw/(1+G0); % Gw: geschlossener Regelkreis mit Terminator
60 figure
61 step(w*Gw,80); title('ANFAHREN mit Terminator'); grid
```



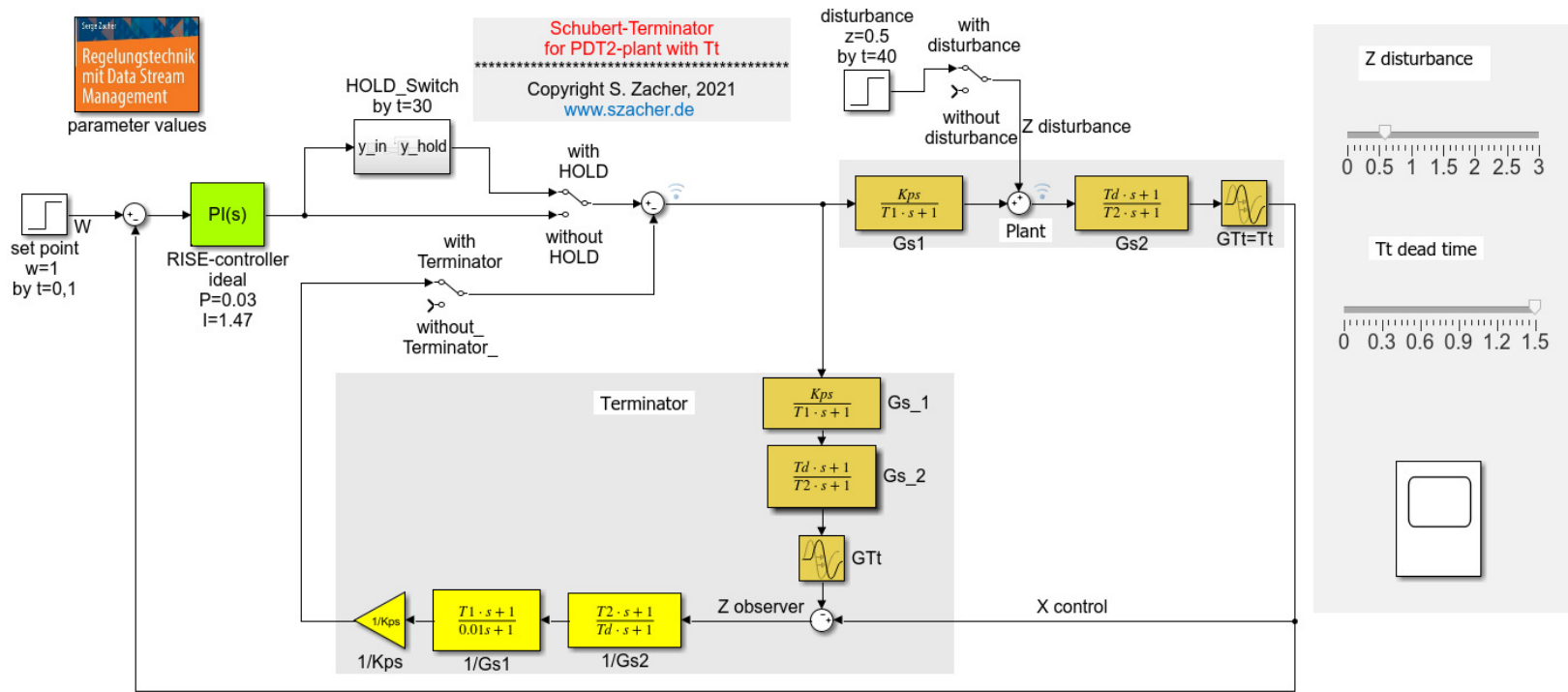
## 5 Schubert-Terminator für PDT2-Strecken mit Totzeiten

### 5.3 MATLAB-Skript: PI-Regler für PDT2-Strecke mit Tt

```
62 %% 6. Führungs- und Störverhalten ohne Terminator
63 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
64 z=0.5*exp(-40*s)*s/s; % z: Störsprung z=0.5 bei t=6 sec
65 Gvz=Gv2; % Gvz: Vorwärts-Übertragungsfunktion des Kreises für z
66 Gz=Gvz/(1+G0); % Gvw: Übertragungsfunktion des Kreises beim Störverhalten
67 G_oT=Gw*w+Gz*z; % G_oT:geschlossener Regelkreis ohne Terminator
68 figure
69 step(G_oT,w,z,80);title('Regelung ohne Terminator');
70 grid
71 %% 7. Führungs- und Störverhalten mit Terminator
72 w=1*s/s; % w=1: Sollwert-Sprung bei t=0
73 z=0.5*exp(-40*s)*s/s; % z: Störsprung z=0.5 bei t=6 sec
74 GT=1/(1+G_inv*(Gs-Gs)); % GT: Terminator GT=1;
75 Gvw=GR*GT*Gs; % Gvw: Vorwärts-Übertragungsfunktion für w
76 GvzT=Gs2*(1-Gs*G_inv); % GvzT:Vorwärts-Übertragungsfunktion für z
77 G0=GR*GT*Gs; % G0: offener Regelkreis ohne Terminator (RISE)
78 Gw=Gvw/(1+G0); % Gw: Führungsverhalten mit Terminator
79 GzT=GvzT; % GzT: Störverhalten mit Terminator bei HOLD
80 G_mT=w*Gw+z*GzT; % G_mT:Führungs-und Störverhalten mit Terminator
81 figure
82 step(G_mT,w,z,80); title('Regelung mit Terminator'); grid
```



# 5.4 Simulink: PI-Regler für PDT2-Strecke mit Totzeit

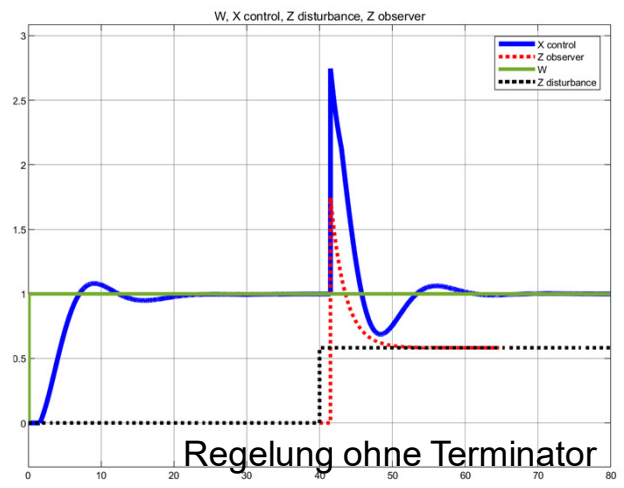


**Z disturbance**

0 0.5 1 1.5 2 2.5 3

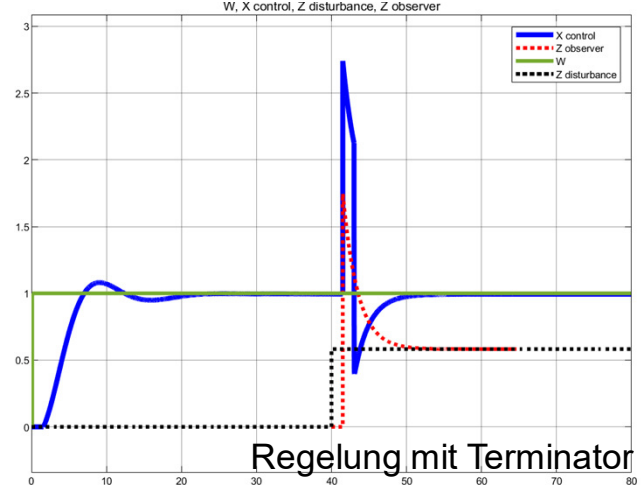
**Tt dead time**

0 0.3 0.6 0.9 1.2 1.5



$Kps=5.5;$   
 $T1=4;$   
 $T2=2;$   
 $Td=6;$   
 $Tt=1.5;$   
 $P=0.03;$   
 $I=1.47;$

— X  
— W  
- - - Z  
- - - Z





## 6 Schubert-Terminator mit RISE- und HOLD-Reglern

Um die Regelung von Strecken mit Totzeiten und D-Anteilen mit dem *Schubert-Terminator*, die im vorherigen Abschnitt 5 betrachtet wurde, zu verbessern, wird in diesem Abschnitt ein zusätzlicher Regler speziell für HOLD-Betrieb eingeführt. Somit soll die gesamte Regelung mit zwei Reglern erfolgen:

### RISE-Betrieb (ANFAHREN)

- Der Sollwertsprung wird eingegeben:

$$w = w_0$$

- Die Regelgröße  $x_{\text{ist}}$  wird mit einem Regler mit I-Anteil (I-, PI-, PID) ohne statischen Fehler zum Sollwert gebracht:

$$x_{\text{ist}} = x_{\text{soll}} = w_0$$

- Die Stellgröße  $y$  erreicht dabei einen Wert

$$y = y_{\text{soll}}$$

- Der RISE-Regler wird blockiert, der Stellwert  $y_{\text{soll}}$  wird gehalten-
- Der RISE-Betrieb wird damit beendet, es fängt die Phase HOLD an.

### HOLD-Betrieb (HALTEN)

- Der Stellwert wird gehalten:

$$y = y_{\text{soll}}$$

- Die Wirkung der Störgröße  $z$  wird mit dem  $Z_{\text{observer}}$  erkannt (siehe Abschnitt 5).
- Der HOLD-Regler wirkt, um die Regelgröße  $x_{\text{soll}}$  konstant zu halten:

- Die aktuelle Abweichung der Regelgröße  $x_{\text{ist}}$  von  $x_{\text{soll}}$  wird bestimmt:

$$\Delta x = x_{\text{soll}} - x_{\text{st}}$$

- Die Regeldifferenz  $\Delta x$  wird mit einem HOLD-Regler mit D-Anteil (DT1-, PD-T1) möglichst schnell ausgeregelt.
- Dabei wird die Stellgröße auf einen neuen Wert gebracht, bei dem wieder  $w = w_0$  gilt:

$$y = y_{\text{soll}} + \Delta y$$

## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

Rise\_and\_hold\_controller.m

```

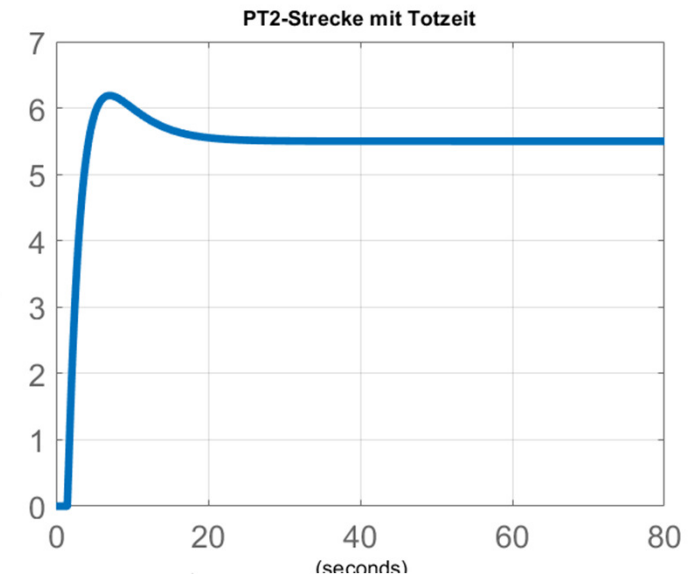
1 % Schubert-Terminator mit R&H-controller, Stand 30.12.2021
2 % RISE-controller (PI) für Führungsverhalten (Anfahren, RISE)
3 % HOLD-controller (PDT1) für Störverhalten (Halten, HOLD)
4 % Quellen: [1] S.Zacher: "Drei-Bode-Plots-Verfahren für Regelungstechnik", 2020
5 %           [2] S.Zacher: "Regelungstechnik mit Data Stream Management", 2021
6 % 2021 Copyright Dr.Zacher, www.szacher.de
7 %*****

```

```

8 %% 1. Regelstrecke und RISE-PI-Regler
9 clearvars; clc; % Bildschirm bzw. Workspace säubern
10 s = tf('s'); % Laplace-Operator
11 w=1*s/s; % w: Sollwert-Sprung bei t=0 sec
12 z=0.5*exp(-40*s)*s/s; % z: Störsprung bei t=40 sec
13 Kps=5.5;T1=4; T2=2;Td=6; % Streckenparameter
14 Tt=1.5; % Tt: Totzeit
15 Gs1=Kps/(1+s*T1); % Gs1: Teilstrecke vor der Störung PT1
16 Gs2=(1+s*Td)/(1+s*T2); % Gs2: Teilstrecke nach der Störung DT1
17 GTt=s*exp(-s*Tt)/s; % GTt: Teilstrecke nach der Gs2 (Totzeit)
18 Gs_ohne=Gs1*Gs2; % Gs_ohne: gesamte Strecke ohne Totzeit
19 Gs=Gs1*Gs2*GTt; % Gs: gesamte Strecke ohne Totzeit
20 G_inv=(1/Kps)*(1+s*T1)*(1+s*T2)/((1+s*Td)*(0.01*s+1)); % G_inv: inverse ohne Tt
21 step(Gs,80);title('PT2-Strecke mit Totzeit'); grid
22 KpR=1;Tn=0.7; % KpR,Tn: angenommene Parameter
23 GR=KpR*(1+s*Tn)/(s*Tn); % GR: RISE-Regler(für Führungsverhalten)

```



## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

```

24 %% 2. RISE-PI-Reglereinstellung nach DBV,
25 % gewünschte Phasenreserve ca.60°,
26 % Quelle [1], Seiten 231-268
27 GR=KpR*(1+s*Tn)/(s*Tn);
28 GRsymm_mag=1/GR;
29 GR_phs=1/(s^2*GR);
30 wmin=0.1;wmax=2;
31 bode(Gs,{wmin,wmax}); hold on;grid
32 subplot(211);
33 bodemag(Gs,{wmin,wmax}); hold on
34 opts_mag=bodeoptions('cstprefs');
35 opts_mag.PhaseVisible='off';
36 h_mag=bodeplot(GRsymm_mag,{wmin,wmax},opts_mag);
37 hold on;grid
38 subplot(212);
39 bode(Gs,{wmin,wmax}); hold on
40 opts_ph=bodeoptions('cstprefs');
41 opts_ph.MagVisible='off';
42 h_ph=bodeplot(GR_phs,{wmin,wmax},opts_ph);
43 hold on;grid

```

```

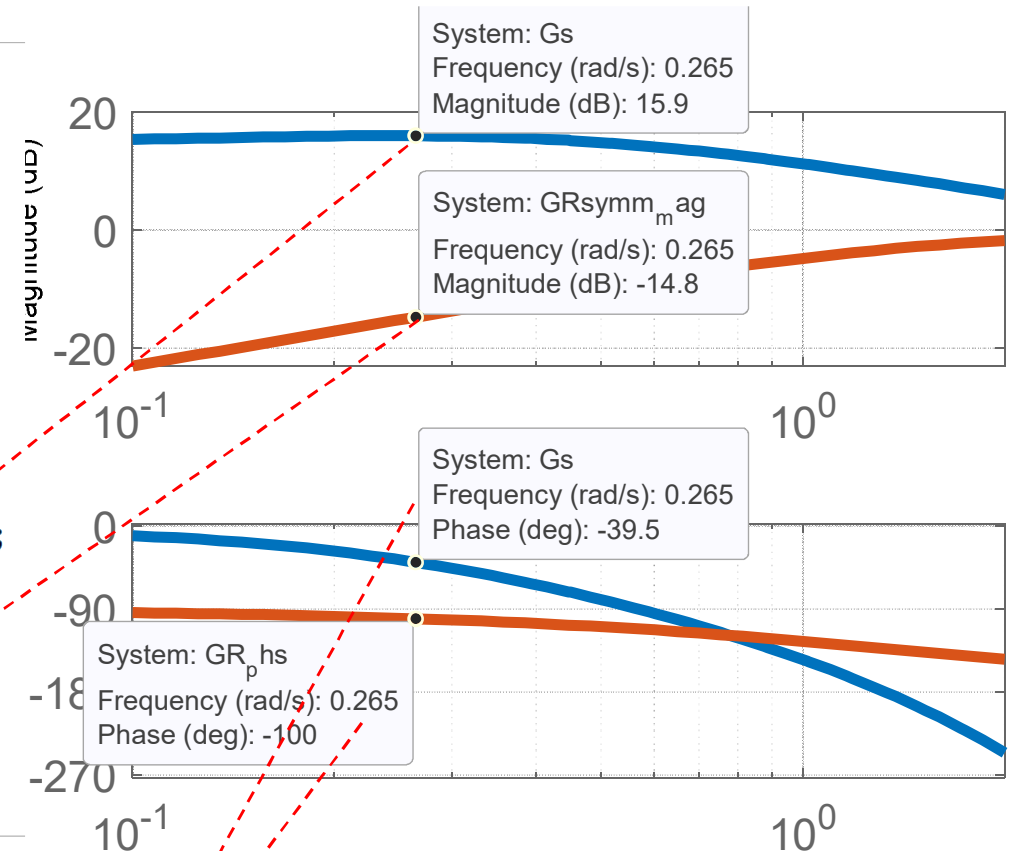
44 %% 3. RISE-PI-Reglereinstellung:

```

```

45 % Verschiebung des Amplitudengangs in gewünschten Arbeitspunkt
46 delta_K=15.9+14.8; % im Bode-Plot bei omega 0.265 ablesen
47 KpR=10^(-delta_K/20); % delta_K von dB umrechnen
48 GR=KpR*(1+s*Tn)/(s*Tn); % Der PI-Regler mit dem neuen KpR-Wert
49 % falls nötig, die Sektion 2 mit dem neuen KpR-Wert wiederholen

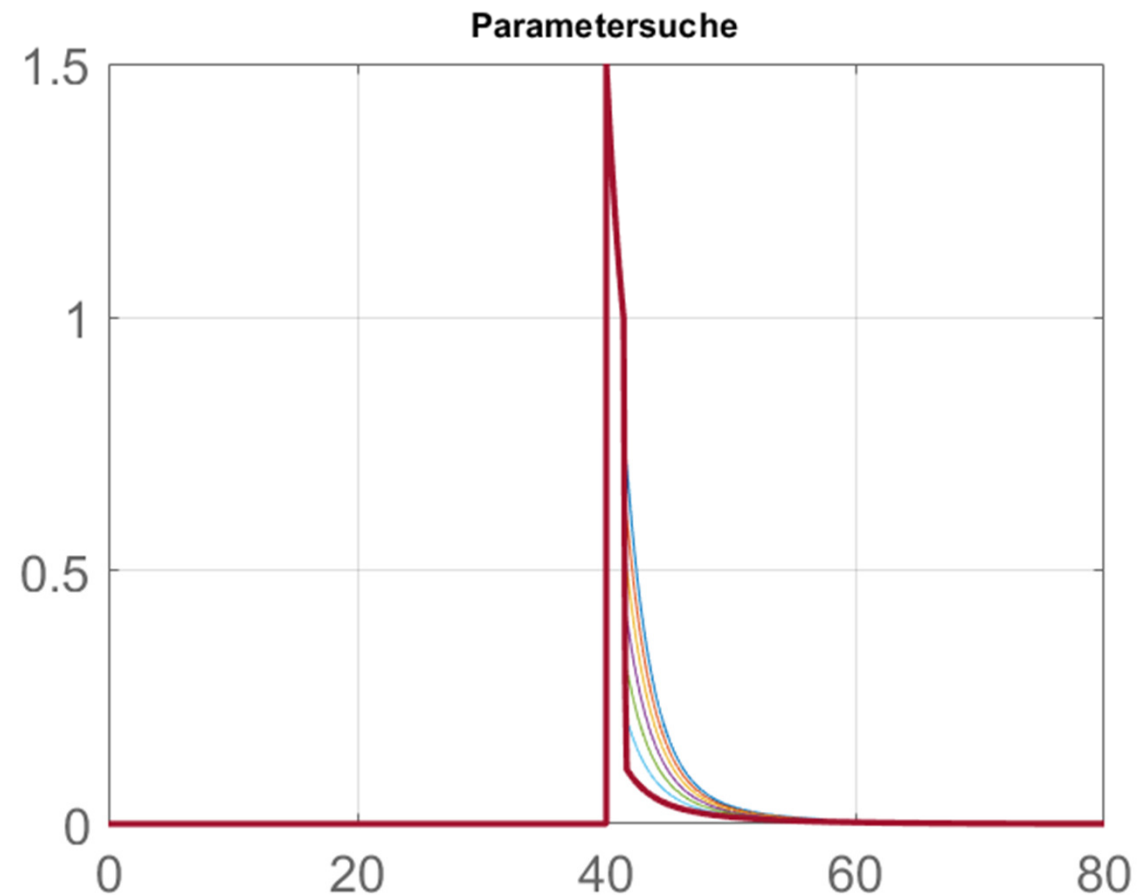
```



$$K_{PR}=0,0292$$

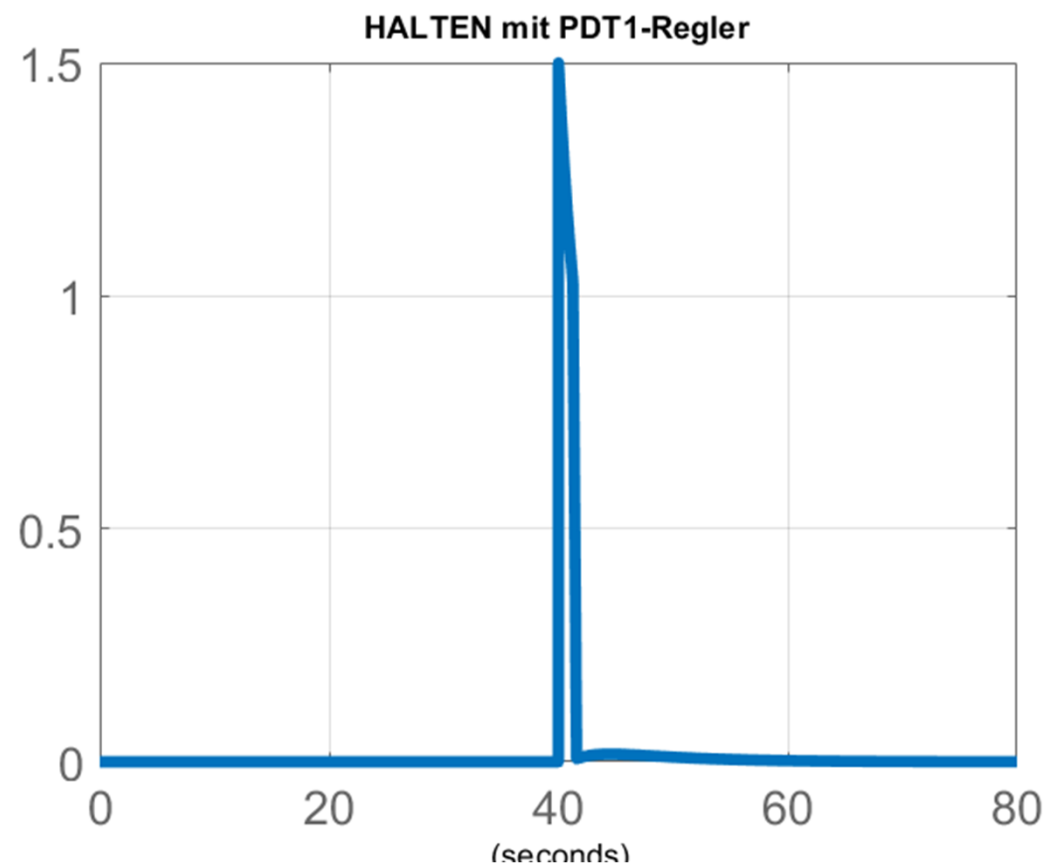
## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

```
50 %% 4. HOLD-PDT1-Regler: Parametersuche
51 Tv=1;TR=7; % KpR=1, um stat.Fehler e(oo)zu vermeiden
52 for k=1:1:7 % Schleife für Parameter-Suche
53 GH=(1+s*Tv)/(1+s*TR); % HOLD-PDT1-Regler
54 GvzH=Gz2*(1-GH*Gs*G_inv); % GvzH:Vorwärts-Übertragungsfunktion für z
55 GzH=GvzH; % GzH: Störverhalten mit Terminator bei HOLD
56 % bitte vorher das Bild mit Bode-Diagramm der Sektionen 3, 4 löschen
57 step(z*GzH,80); title ('Parametersuche');
58 hold on
59 Tv=Tv+0.5
60 end
61 grid
```



## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

```
62 %% 5. HOLD-Kreis: Regelung mit Terminator nach den TV und TR der Sektion 6
63 hold off
64 %Tv=4.5;TR=7; % HOLD-Regler-Parameter nach Sektion 4
65 GH=(1+s*Tv)/(1+s*TR); % HOLD-Regler
66 GvzH=G_s2*(1-GH*G_s*G_inv); % GvzH:Vorwärts-Übertragungsfunktion für z
67 GzH=GvzH; % GzH: Störverhalten mit Terminator bei HOLD
68 step(z*GzH,80);title('HALTEN mit PDT1-Regler'); grid
```

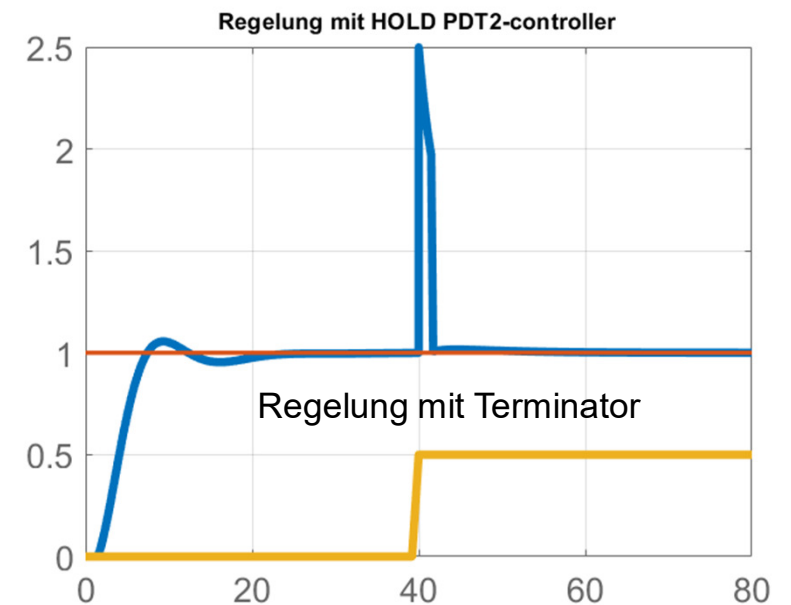
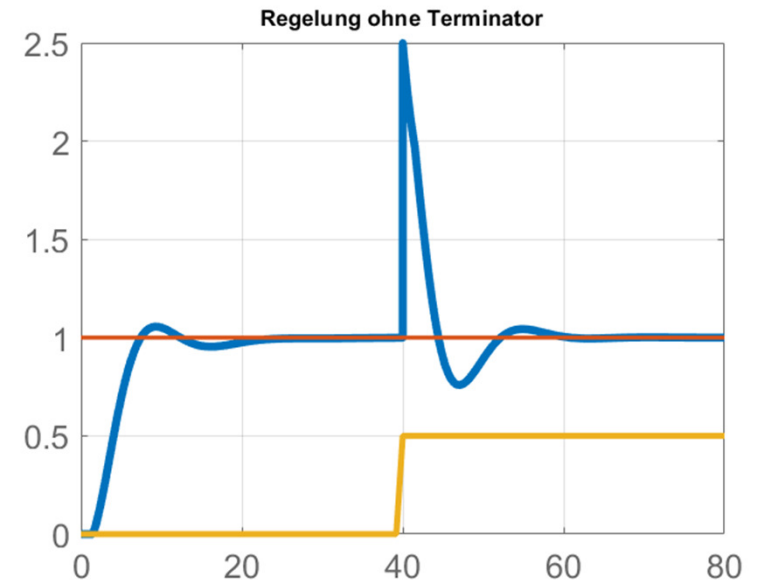


## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

```

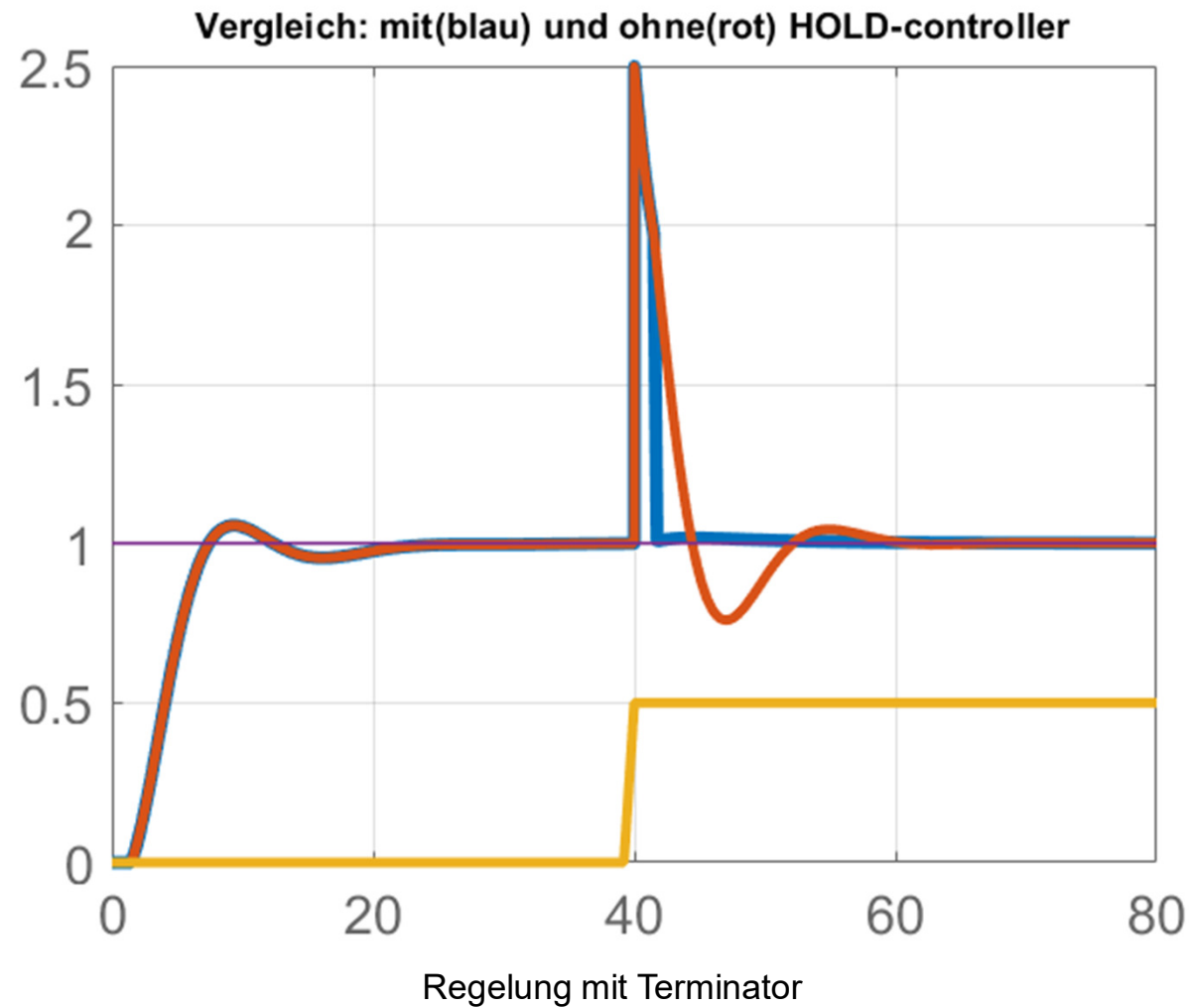
69  %% 6. R&H-Regelung ohne Terminator
70  Gvw=GR*Gs;           % Gvw: Vorwärts-Übertragungsfunktion
71  G0=GR*Gs;           % G0: offener Regelkreis ohne Termi
72  Gw=Gvw/(1+G0);      % Gw: Führungsverhalten ohne Termin
73  Gvz=Gs2;           % Gvz: Vorwärts-Übertragungsfunktion
74  Gz=Gvz/(1+G0);      % Gvz: Störverhalten ohne Terminator
75  G_oT=Gw*w+Gz*z;     % G_oT: geschlossener Regelkreis ohne
76  figure
77  step(G_oT,w,z,80);title('Regelung ohne Terminator');
78  grid
79  %% 7. R&H-Regelung mit Terminator und mit HOLD DT1-controll
80  % Sektionen 4,5,6 ausführen (Tv,TR,GH,G0_H,GzH bestimmen),
81  GT=1/(1+G_inv*(Gs-Gs)); % GT: Terminator GT=1;
82  G0=GR*GT*Gs;        % G0: offener Regelkreis mit Termi
83  Gvw=GR*Gs;          % Gvw: Vorwärts-Übertragungsfunktio
84  Gw=Gvw/(1+G0);      % Gw: Führungsverhalten mit Termin
85  G_mT=Gw*w+GzH*z;    % G_mT: geschlossener Regelkreis mi
86  figure
87  step(G_mT,w,z,80);title('Regelung mit HOLD PDT2-controller')

```



## 6.1 MATLAB-Skript: R&amp;H für PDT2-Strecke mit Tt

```
88 %% 8. Vergleich Regelung mit und ohne HOLD-PDT1-controller
89 hold off
90 step(G_mT, G_oT,z,w,80);title('Vergleich: mit(blau) und ohne(rot) HOLD-controller');grid
```





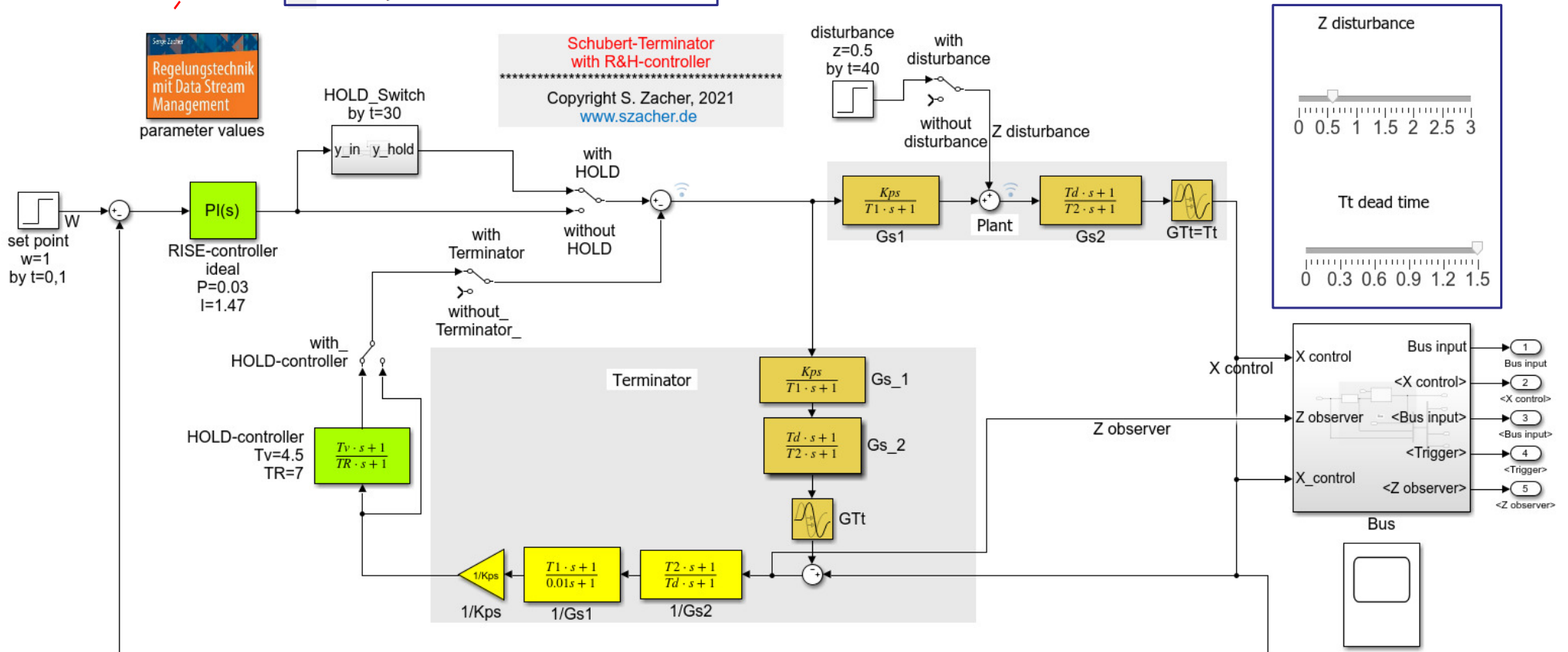
## 6.2 Simulink: R&H für PDT2-Strecke mit Tt

Rise\_and\_hold\_with\_Tt\_with\_D.slx

```

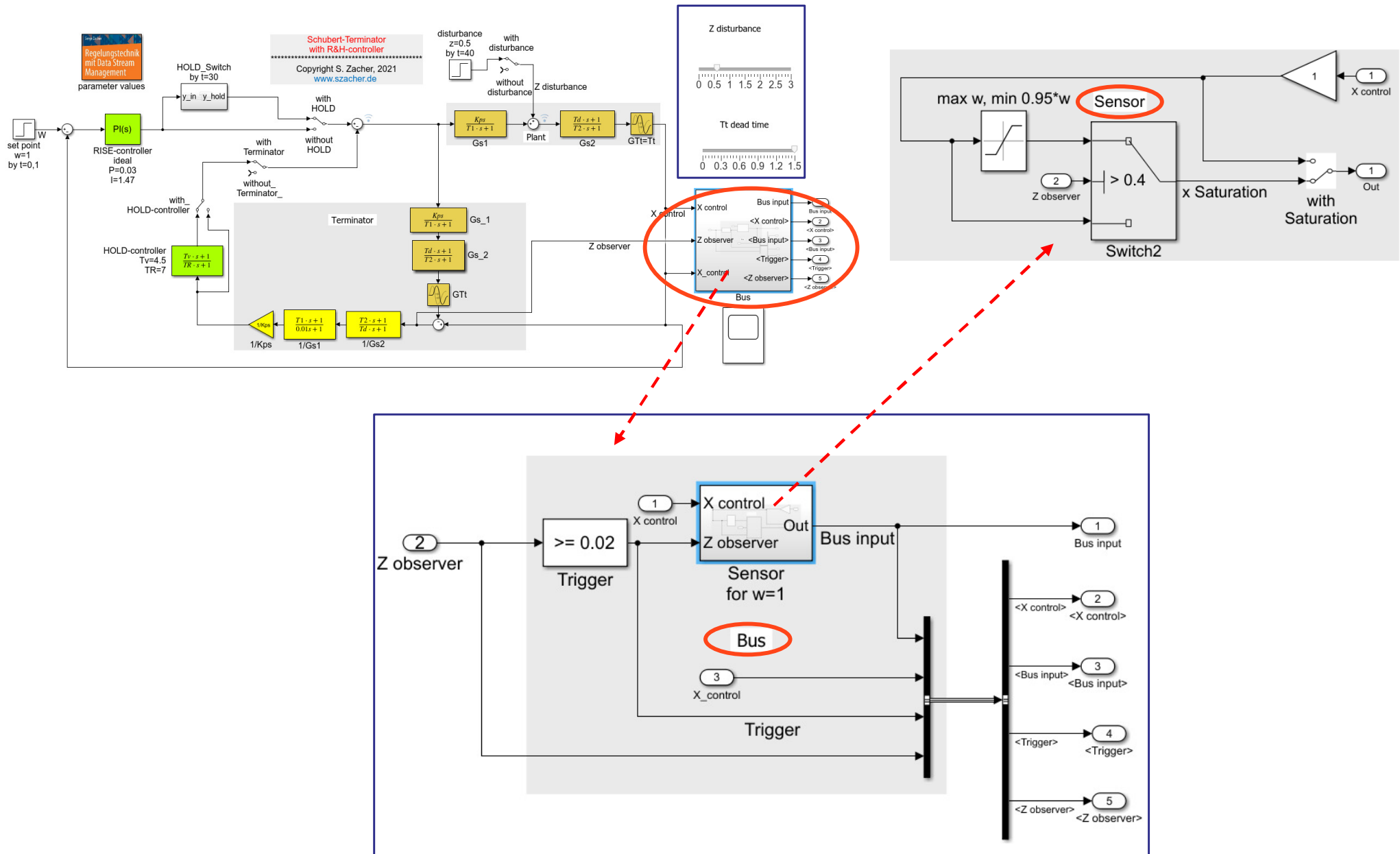
6 clearvars; clc
7 Kps=5.5; % Plant
8 T1=4;
9 T2=2;
10 Td=6;
11 Tt=1.5;
12 P=0.03; % Rise-controller
13 I=1.47;
14 Tv=4.5; % Hold-controller
15 TR=7;
    
```

Wie bereits anhand des MATLAB-Skriptes gezeigt wurde, kann man den ersten Impuls von der Höhe 2,5 bei t=40 sec nicht beseitigen, weil der  $Z_{observer}$  erst nach der Totzeit  $Tt = 1,5$  sec die Störung erkennen kann. Falls jedoch die Überwachung der Regelgröße nicht direkt über die Stellgröße  $y$  erfolgen kann, wäre es möglich, den Impuls über eine Bus-Schnittstelle begrenzen (siehe nächste Seite mit dem Baustein „Saturation“).



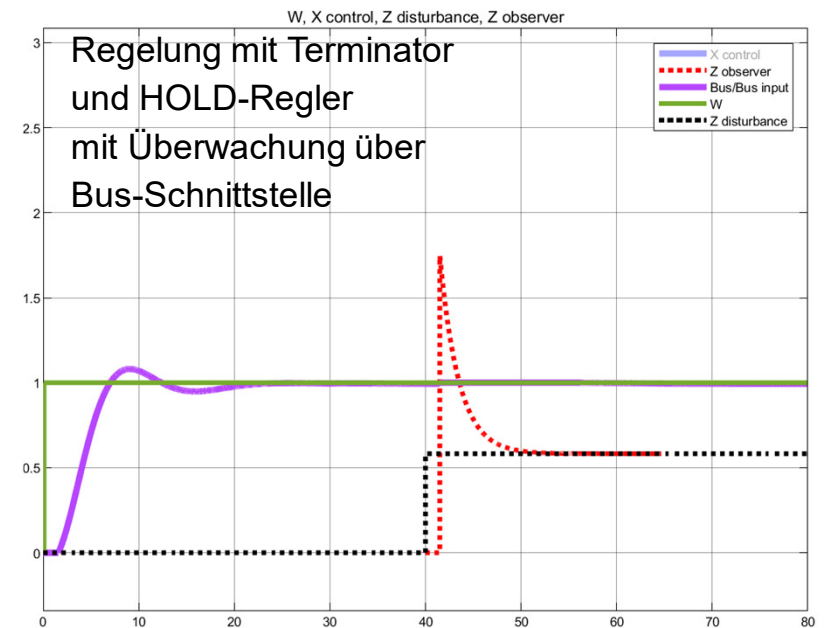
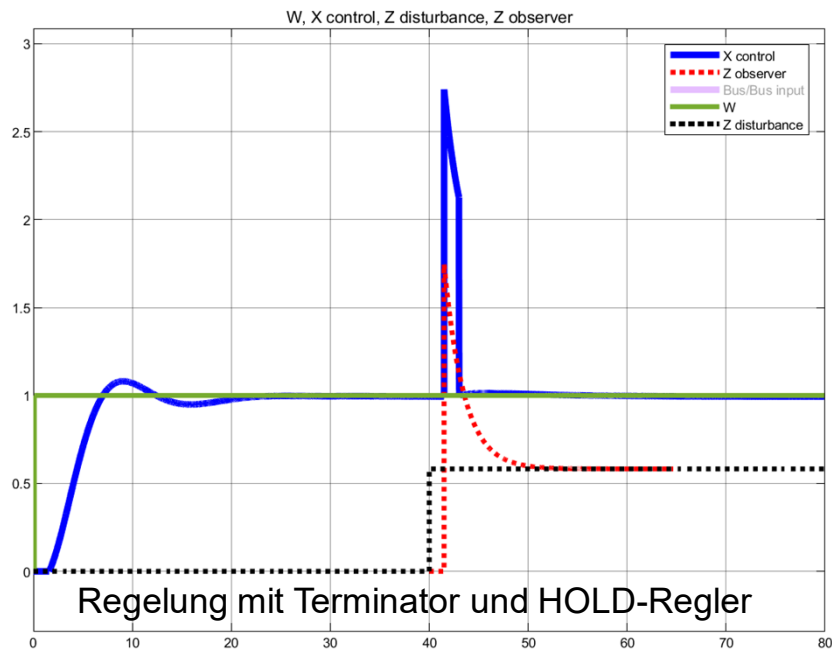
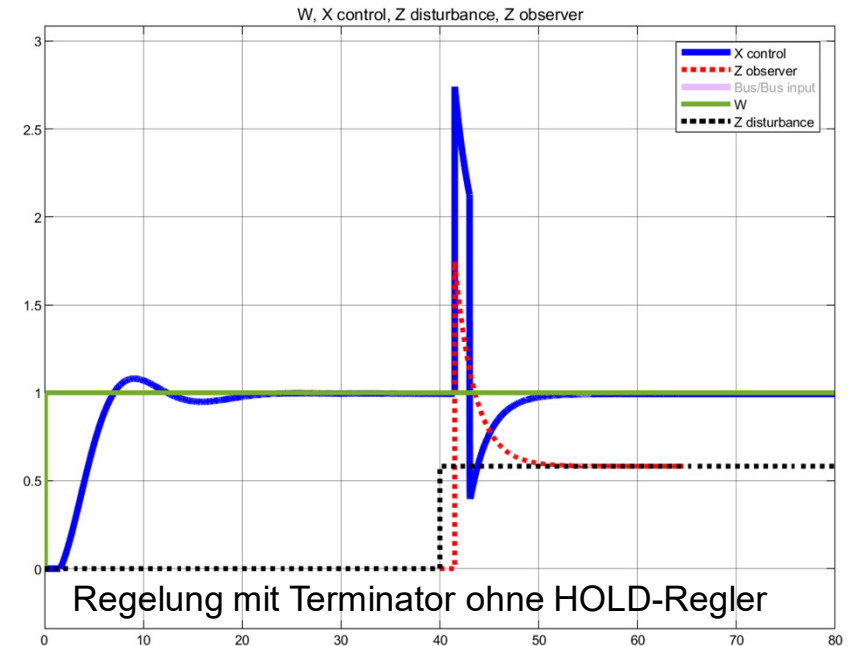
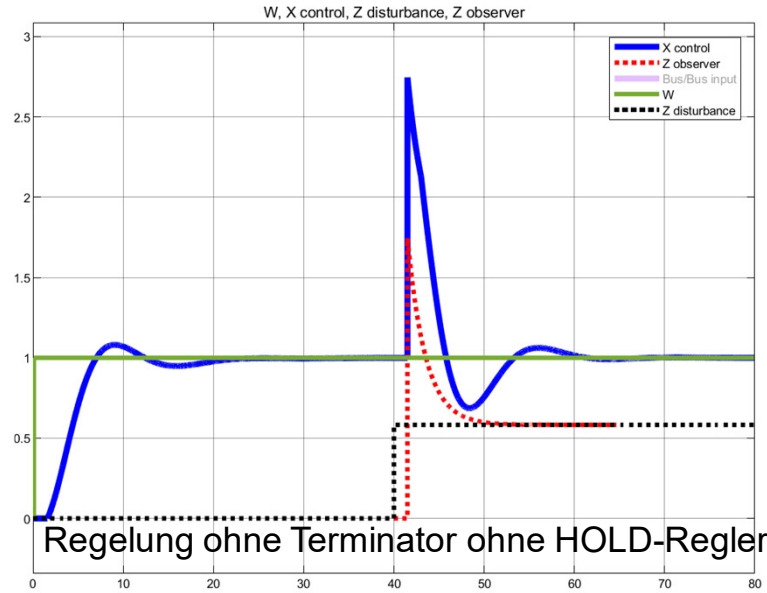
# 6 Schubert-Terminator mit RISE- und HOLD-Reglern

## 6.2 Simulink: R&H für PDT2-Strecke mit Tt



## 6 Schubert-Terminator mit RISE- und HOLD-Reglern

### 6.2 Simulink: R&H für PDT2-Strecke mit Tt



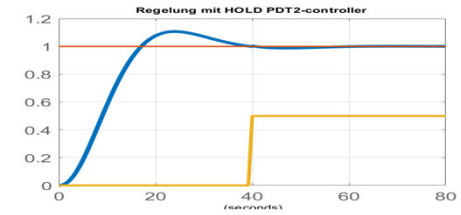
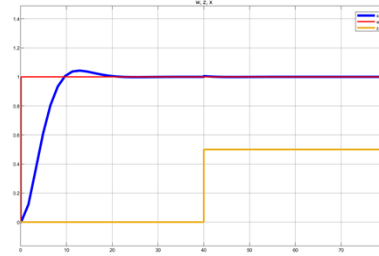
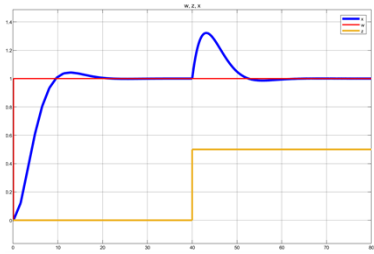
# 8 Zusammenfassung

ohne Terminator,  
ohne HOLD-Regler

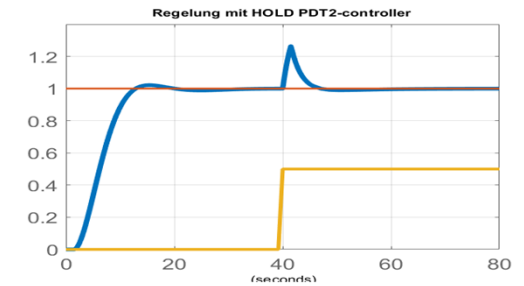
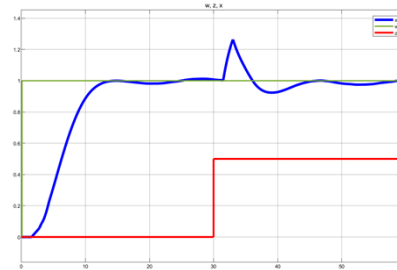
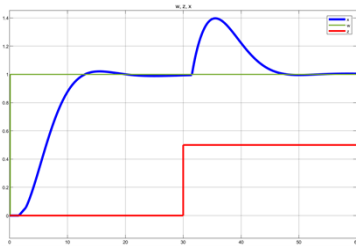
mit Terminator,  
ohne HOLD-Regler

mit Terminator,  
mit HOLD-Regler

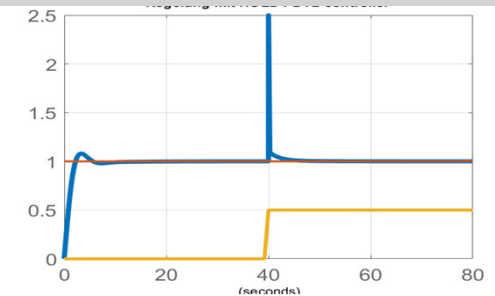
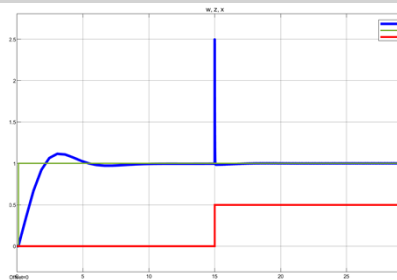
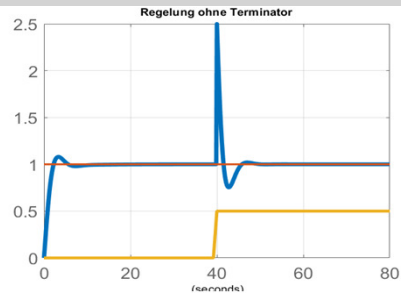
PT2-Strecke



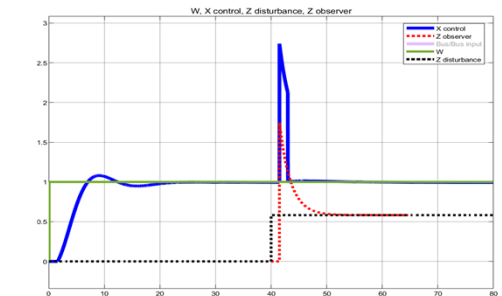
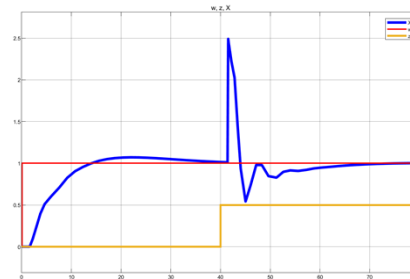
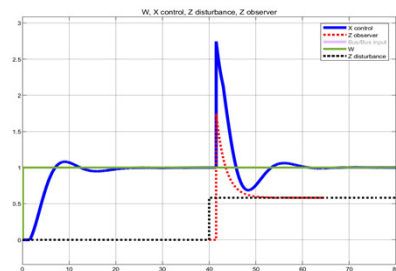
PT2-Tt-Strecke



PDT2-Strecke



PDT2-Tt-Strecke



## 8 Literaturverzeichnis (Zugegriffen 02.12.2021)

- [1] Zacher; S: *Regelungstechnik mit Data Stream Management*. Springer-Vieweg Verlag, 2021  
<https://www.springer.com/gp/book/9783658308599>
- [2] Zacher, S. Reuter, M: *Regelungstechnik für Ingenieure*. 15. Auflage, Springer-Vieweg Verlag, 2017  
<https://www.springer.com/gp/book/9783658176310>
- [3] Zacher, S.: *Terminator im Regelkreis*. Automation-Letter Nr. 43, 2021  
[https://www.zacher-international.com/Automation\\_Letters/43\\_Terminator.pdf](https://www.zacher-international.com/Automation_Letters/43_Terminator.pdf)
- [4] Zacher, S. : *Terminator*, 2021. <https://youtu.be/EVHLY8RtSHQ>
- [5] Zacher, S.: *Drei-Bode-Plots-Verfahren und BAD*, 2021.  
[https://www.zacher-international.com/Automation\\_Letters/11\\_Fingerprint\\_DBV.pdf](https://www.zacher-international.com/Automation_Letters/11_Fingerprint_DBV.pdf)
- [6] Zacher; S: *Drei-Bode-Plots-Verfahren für Regelungstechnik*. Springer-Vieweg Verlag, 2020  
<https://www.springer.com/gp/book/9783658292195>
- [7] Zacher, S.: *Bode-Aided-Design*, 2021. <https://youtu.be/iil-tRnY3J8>
- [8] Zacher, S.: *Bus-Approach for Feedback MIMO-Control*. Verlag Dr. S. Zacher, Stuttgart, 2014  
<https://www.szacher.de/my-Books/Bus/>
- [9] Zacher, S.: *PID-Gesichte*, Automation-Letter Nr. 09, 2016.  
[https://www.zacher-international.com/Automation\\_Letters/09\\_PID\\_Gesichte.pdf](https://www.zacher-international.com/Automation_Letters/09_PID_Gesichte.pdf)
- [10] Zacher, S.: *Reglereinstellung*, Automation-Letter Nr. 01, 2015-2021  
[https://www.zacher-international.com/Automation\\_Letters/01\\_FAQ\\_Reglereinstellung.pdf](https://www.zacher-international.com/Automation_Letters/01_FAQ_Reglereinstellung.pdf)

- [11] Zacher, S.: *Antisystem-Approach (ASA) for Engineering of Wide Range of Dynamic Systems*. IJONEST, Vol. 3, No.1, 2021, pp. 52-66, ISSN: 2642-4088  
<https://ijonest.net/index.php/ijonest/issue/view/4>
- [12] Zacher; S; Saeed, W.: *Design of multivariable control systems using Antisystem-Approach*. - In: 7. Fachkolloquium AALE, Wien, 2010, pp. 201-209  
[https://www.zacher-international.com/wiss\\_Publikationen/ASA\\_Zacher\\_Saeed.pdf](https://www.zacher-international.com/wiss_Publikationen/ASA_Zacher_Saeed.pdf)
- [13] Zacher, S.: *Asa-Implementierung*, Automation-Letter Nr. 08, 2016.  
[https://www.zacher-international.com/Automation\\_Letters/08\\_ASA\\_Implementierung.pdf](https://www.zacher-international.com/Automation_Letters/08_ASA_Implementierung.pdf)