Dr. Serge ZACHER VERLAG
Stuttgart
D-70188

Automation-Letter Nr. 5
06.12.2015

Prof. Dr. S. Zacher

Model Discretizer von MATLAB

Hinweise zur Digitalisierung von analogen Simulink-Modellen

"Model Discretizer selectively replaces continuous Simulink® blocks with discrete equivalents. Discretization is a critical step in digital controller design and for hardware in-the-loop simulations."

MathWorks, Model Diskretizer http://de.mathworks.com/help/simulink/ug/model-discretizer.html

Abstract, Urheberrechts- und Haftungshinweis

Ab Versionen R2014 und R2015 ermöglicht MATLAB/Simulink die Digitalisierung von Simulink-Modellen mit den Menü-Befehlen

Analysis → *Control Design* → *Model Discretizer*

automatisch durchführen.

Ein Simulink-Modell wird damit automatisch von Laplace-Transformierten Übertragungsfunktionen *G*(*s*) in die z-Transformatierte Übertragungsfunktionen *G*(*z*) umgewandelt. Dies erkennt man sofort durch den Schriftzug *zoh*, der auf allen digitalisierten Bausteinen nach der Ausführung des *Model Discretizer* erscheint.

Nachfolgend werden an einem Beispiel zwei Wege der Digitalisierung beschrieben: mit dem o.g. *Model Discretizer* und nach der z-Transformation.

I N H A L T

Einführung

Die vorliegende Publikation ist meine Antwort auf die folgende Frage eines Studenten zum Wahlfach MBSE „Modellbasierte Softwareentwicklung“:

Sehr geehrter Herr Zacher,

Laut Ihrem Skript ist die z-Transformation notwendig, um einen zeitdiskreten Baustein zu erhalten, welchen ich anschließend in Form eines C-Programmes auf den Board *Arduino* laden kann.

So muss ich nun die Strecke bzw. den Regler entsprechend diskretisieren.

Die Übertragungsfunktionen lauten: Strecke: $G_S(s) = 17{,}1667/((1+0{,}1s)(1+0{,}2s))$

Regler: $G_R(s) = (0{,}0146s+0{,}0728)/0{,}2s$

Die besten Versuche sahen wie folgt aus:

Als erstes habe ich die Übertragungsfunktion in eine Differentialgleichung umgewandelt.

Da diese 2. Ableitungen enthielt, formte ich daraus mittels einfacher Integration eine Integro-Differentialgleichung, welche ich zeitdiskret als Differenzengleichung dargestellt habe.

Dabei galt

int(y(t)dt) = Ta*y[k-1]+Ta/2*delta(y[k])

für delta(y[k]) = y[k]-y[k-1]

Mit den Vorschriften

x[k] --> X(z)

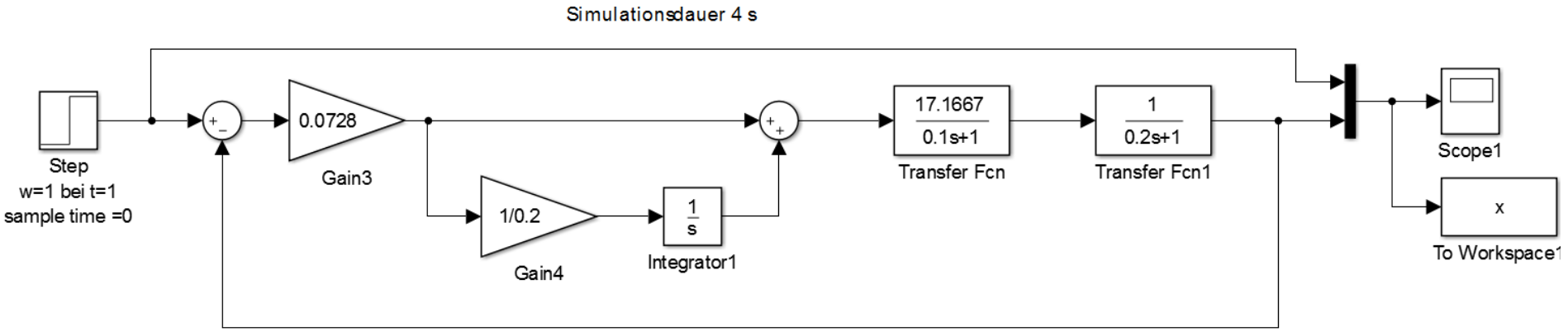
x[k-1] --> 1/z *X(z)

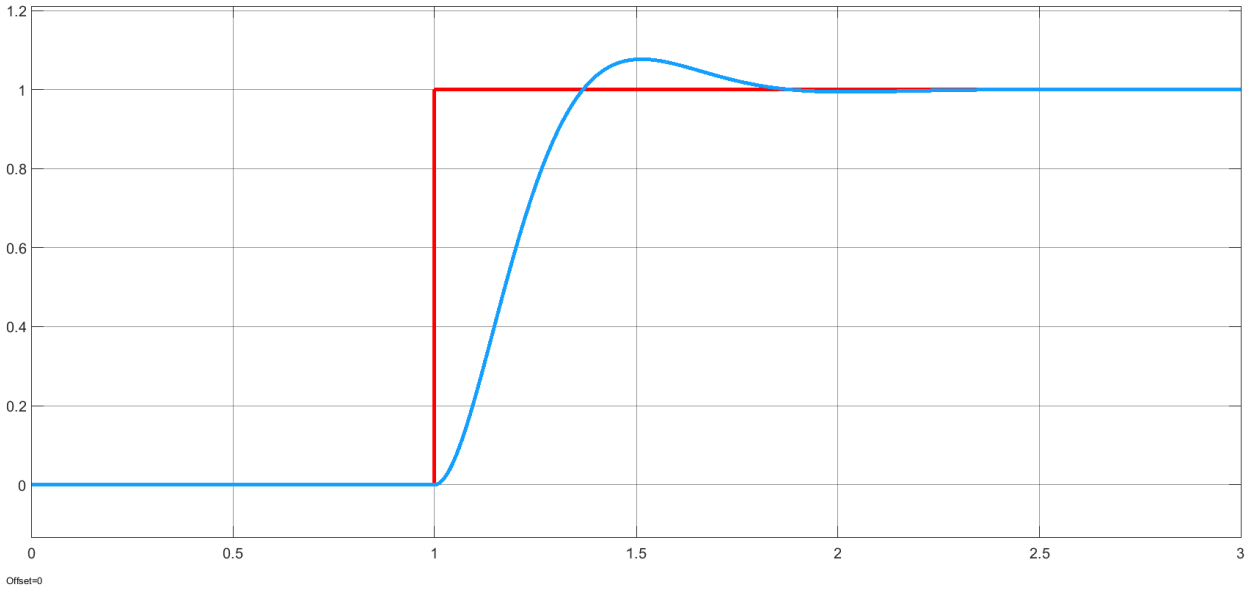
erhielt ich eine Übertragungsfunktion im z-Bereich wie folgt:

$Gs(z) = (17{,}1667*Ta^2*z+17{,}1667*Ta^2)/((0{,}04+0{,}6*Ta+Ta^2)*z+(Ta^2-0{,}04))$

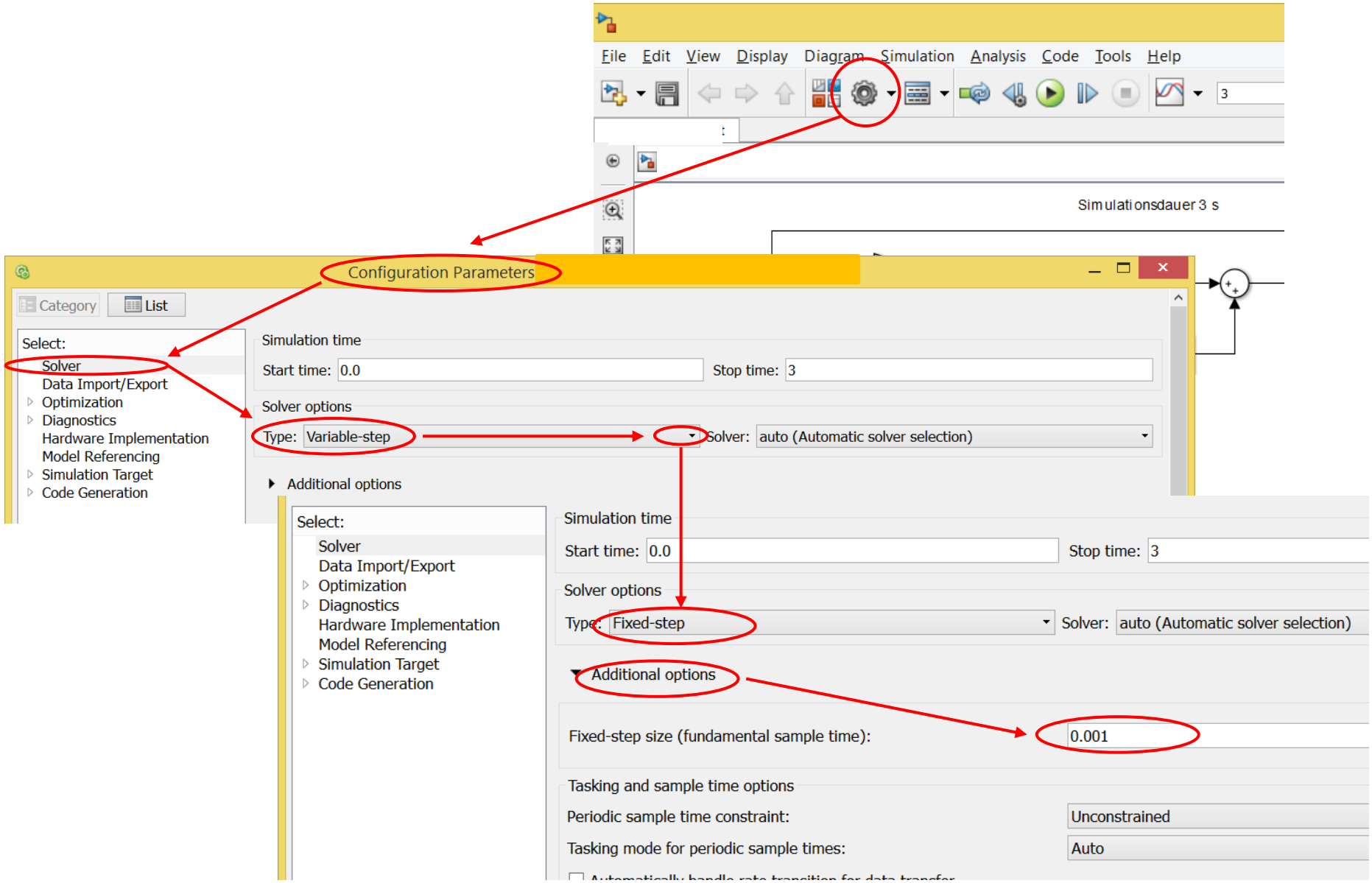
Vergleichende Simulation mit den originalen Blöcken erwies allerdings, dass keine Übereinstimmung zu sehen war.

1. Analoges Simulationsmodell eines Regelkreises

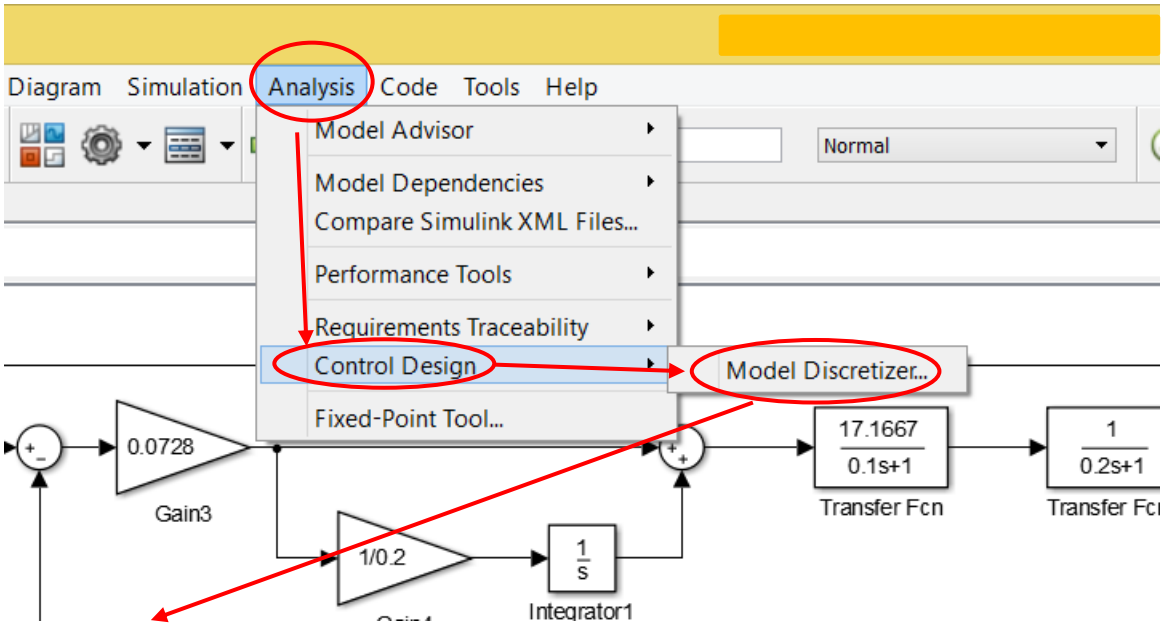


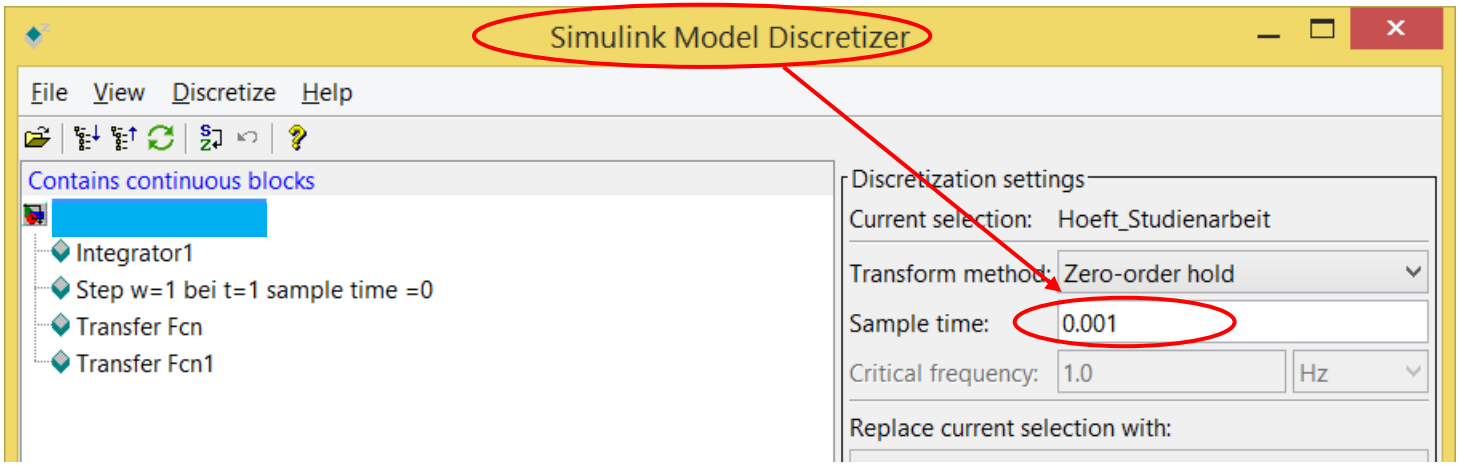


2. *Solver*-Einstellungen für Code-Generierung

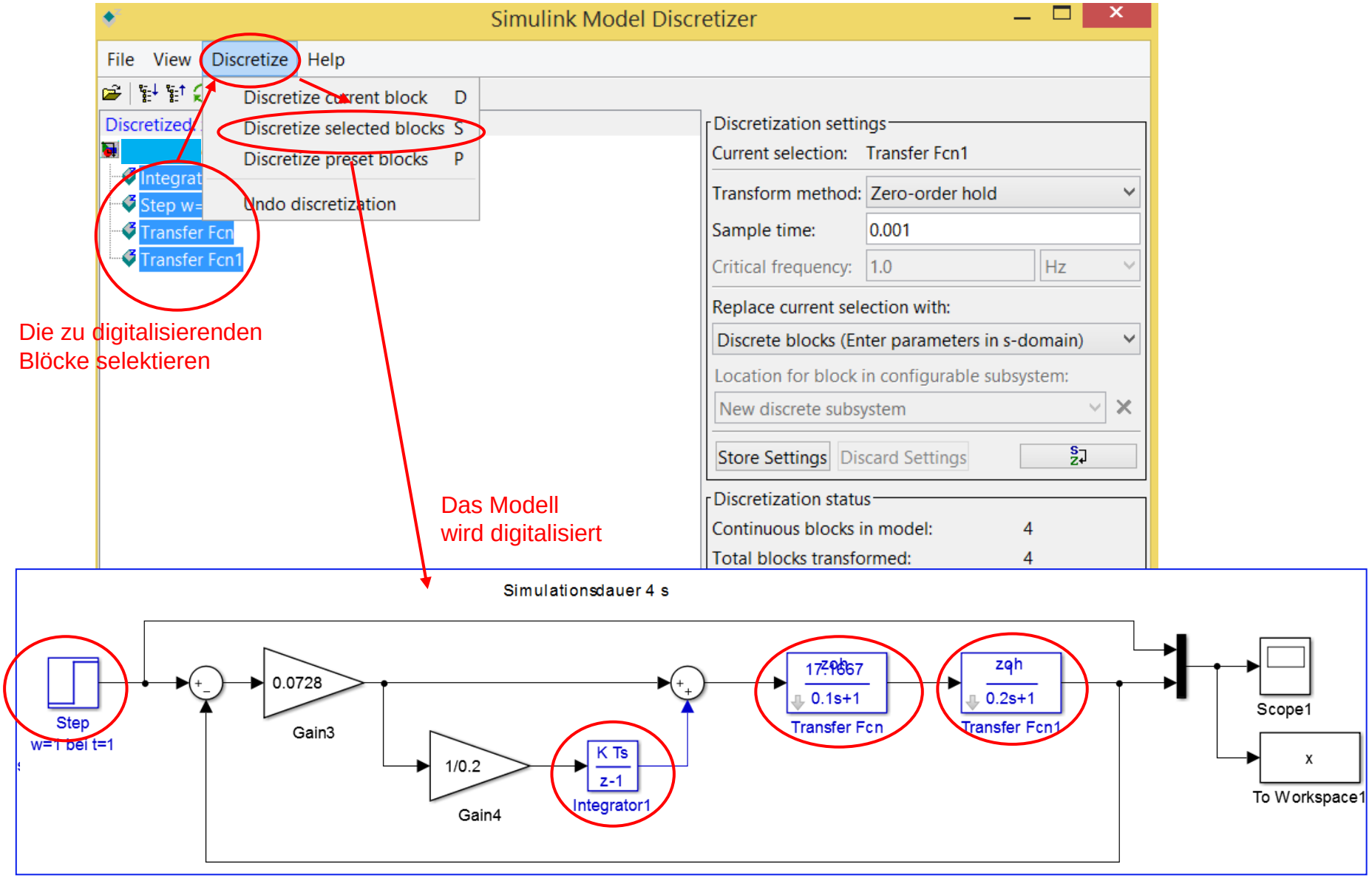


3.1 Digitalisierung mit *Model Discretizer*

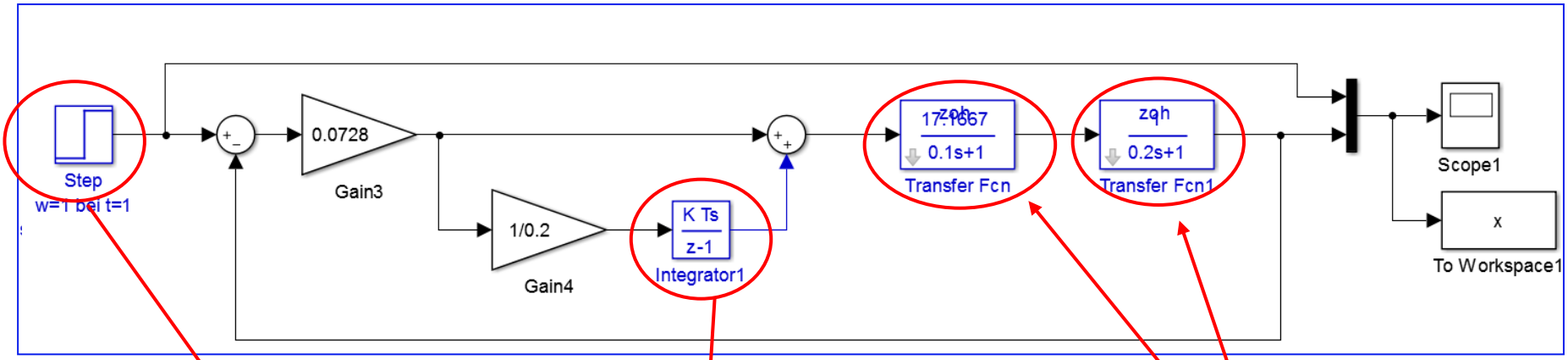


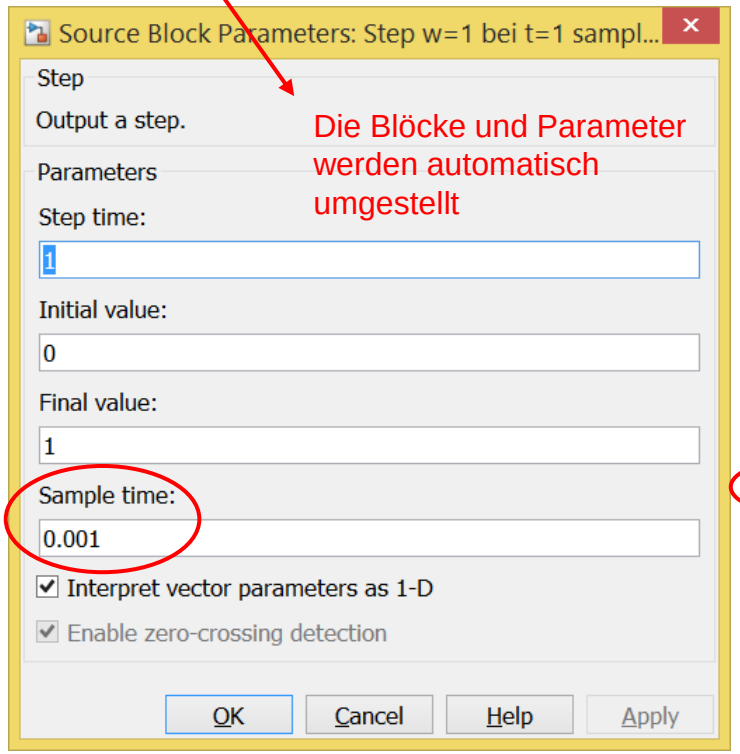


3.1 Digitalisierung mit *Model Discretizer*

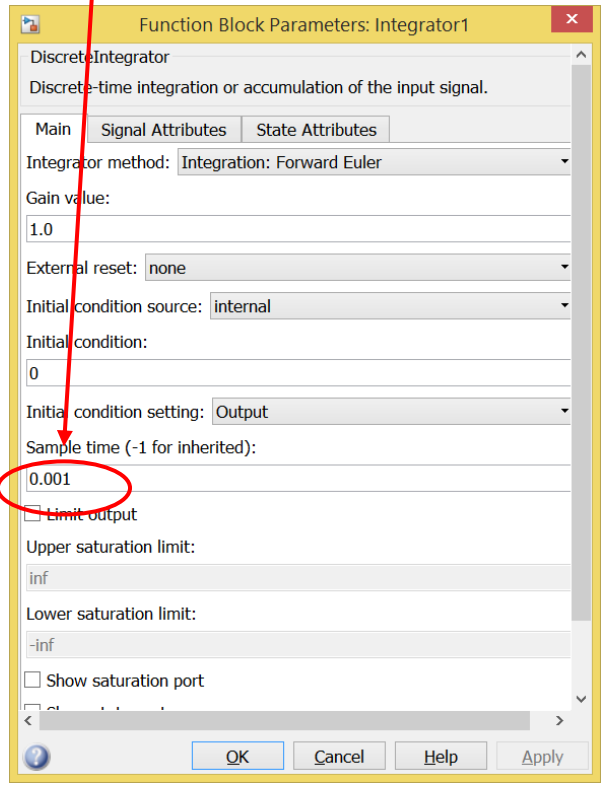


3.1 Digitalisierung mit *Model Discretizer*





Die Blöcke und Parameter werden automatisch umgestellt



Die Parameter der Strecke werden automatisch nach *zoh*-Methode umgerechnet

3.2 Digitalisierung nach z-Transformation

Beispiel: PI-Regler mit P-T2-Strecke

Gegeben ist die P-T2-Strecke mit $K_{PS} = 0{,}5 \qquad T_1 = 1{,}5 \qquad T_2 = 0{,}3$

$$G_S(s) = \frac{K_{PS}}{(1+sT_1)(1+sT_2)}$$

Die Strecke wird mit einem PI-Regler nach dem Betragsoptimum geregelt:

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

Die Übertragungsfunktion des aufgeschnittenen Regelkreises:

$$G_0(s) = G_R(s)G_S(s) = \frac{K_{PR}(1+sT_n)}{sT_n} \cdot \frac{K_{PS}}{(1+sT_1)(1+sT_2)}$$

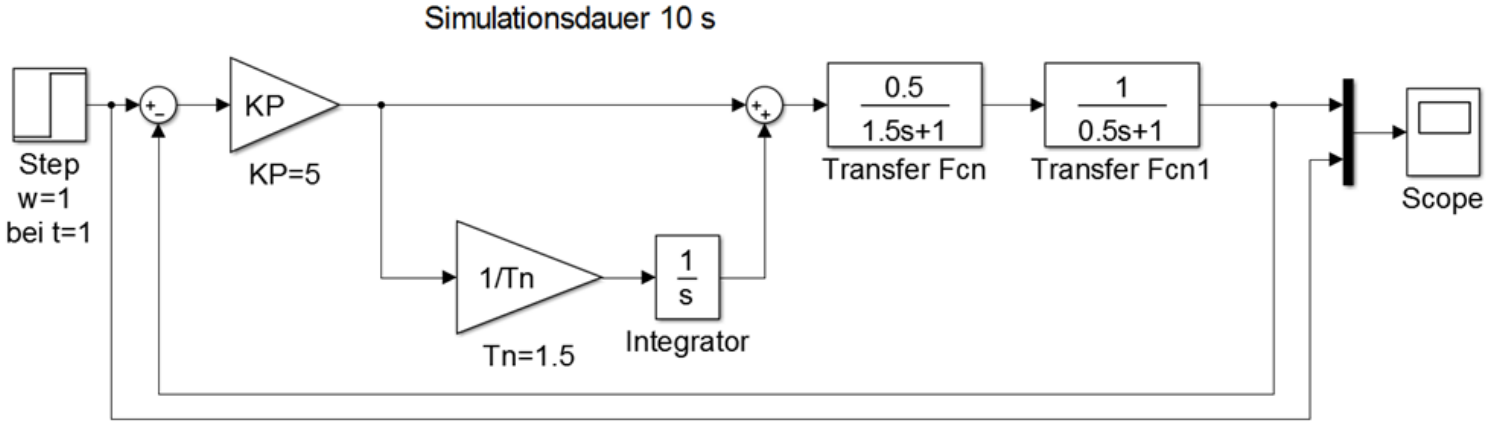
Nach der Kompensation mit $T_n = T_{größte} = T_1 = 1{,}5$

ergibt sich:

$$G_0(s) = \frac{K_{PR}K_{PS}}{sT_n(1+sT_2)}$$

Daraus folgt nach dem Betragsoptimum, Grundtyp „A“:

$$K_{PR} = \frac{T_n}{2K_{PS}T_2} \qquad K_{PR} = \frac{1{,}5}{2 \cdot 0{,}5 \cdot 0{,}3} = 5$$



Da das SIL nur für digitale Regelkreise funktioniert, wird zuerst die Strecke für Abtastzeit $T = 0{,}1$ s digitalisiert (siehe Buch Zacher, „Regelungstechnik-Aufgaben“, 3. Auflage, 2012, Aufgabe 4.2.11, Seite 112):

$$G_S(s) = \frac{K_{PS}}{(1+sT_1)(1+sT_2)} \quad \Longrightarrow \quad G_{HS}(z) = 0{,}187\,\frac{0{,}0275z + 0{,}0157}{(z-0{,}7189)(z-0{,}9352)}$$

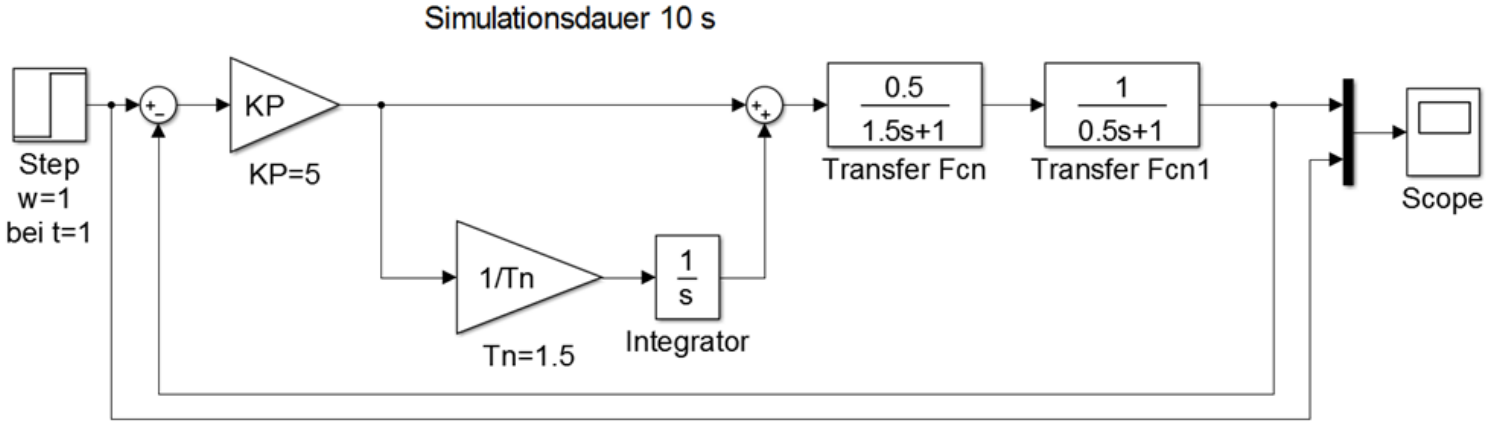
Auch die z-Übertragungsfunktion des PI-Reglers mit gegebenen Kennwerten

$K_{PR} = 5$

$T_n = 1{,}5$ s

wird aus der Tabelle Tabelle 1.3.4, Seite 12, des Buches S. Zacher „Regelungstechnik-Aufgaben“, 3. Auflage, 2012, übernommen:

$$G_R(s) = \frac{K_{PR}(1+sT_n)}{sT_n} \quad \Longrightarrow \quad G_R(z) = \frac{c_1 z - c_2}{z-1} = K_{PR}\,\frac{z-0{,}93}{z-1}$$



Da das SIL nur für digitale Regelkreise funktioniert, wird zuerst die Strecke für Abtastzeit $T = 0{,}1$ s digitalisiert (siehe Buch Zacher, „Regelungstechnik-Aufgaben“, 3. Auflage, 2012, Aufgabe 4.2.11, Seite 112):

$$G_S(s) = \frac{K_{PS}}{(1+sT_1)(1+sT_2)} \quad \Longrightarrow \quad G_{HS}(z) = 0{,}187\,\frac{0{,}0275z + 0{,}0157}{(z-0{,}7189)(z-0{,}9352)}$$

Auch die z-Übertragungsfunktion des PI-Reglers mit gegebenen Kennwerten

$K_{PR} = 5$

$T_n = 1{,}5$ s

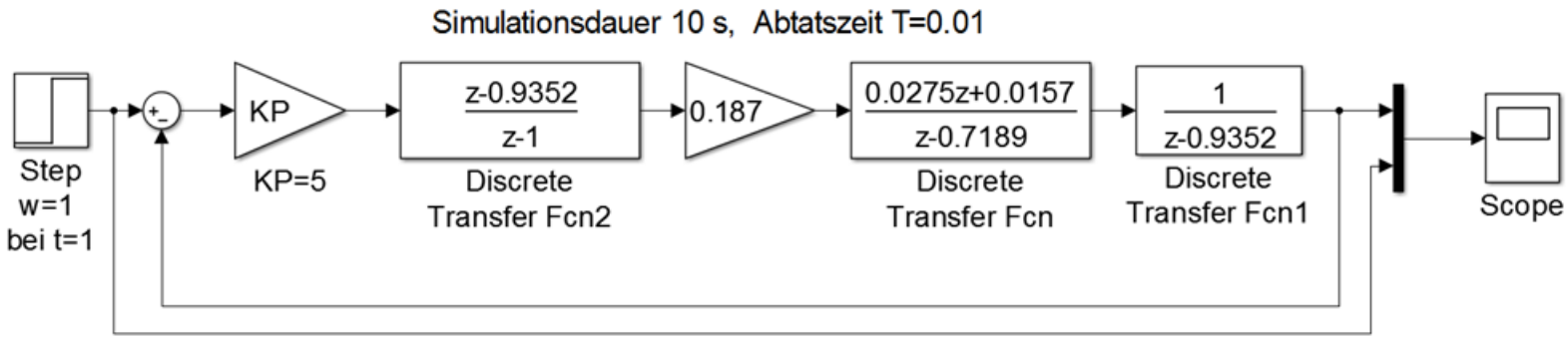
wird aus der Tabelle Tabelle 1.3.4, Seite 12, des Buches S. Zacher „Regelungstechnik-Aufgaben“, 3. Auflage, 2012, übernommen:

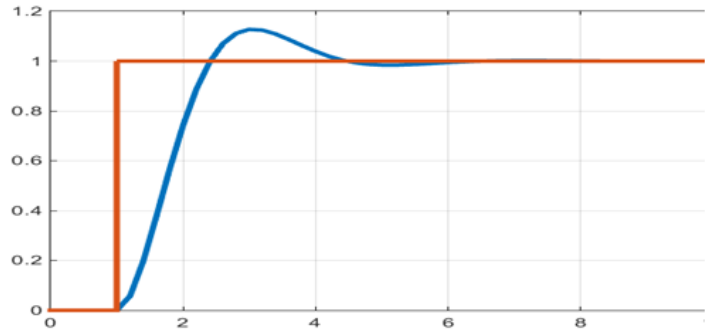
$$G_R(s) = \frac{K_{PR}(1+sT_n)}{sT_n} \quad \Longrightarrow \quad G_R(z) = \frac{c_1 z - c_2}{z-1} = K_{PR}\,\frac{z-0{,}93}{z-1}$$

mit

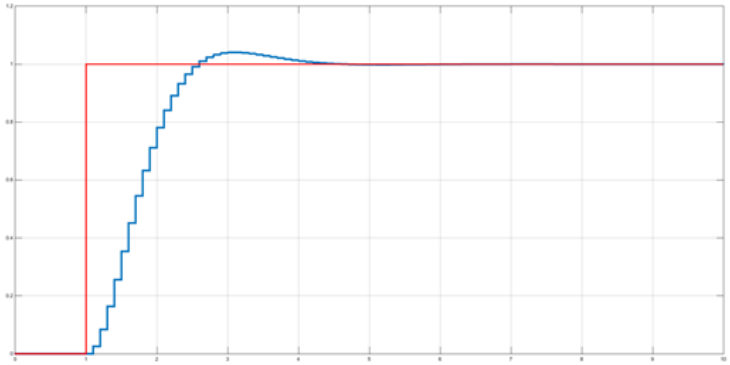
$$c_1 = K_{PR} = 5$$

$$c_2 = K_{PR}\left(1 - \frac{T}{T_n}\right) = K_{PR}\left(1 - \frac{0{,}1}{1{,}5}\right) = 0{,}93K_{PR}$$





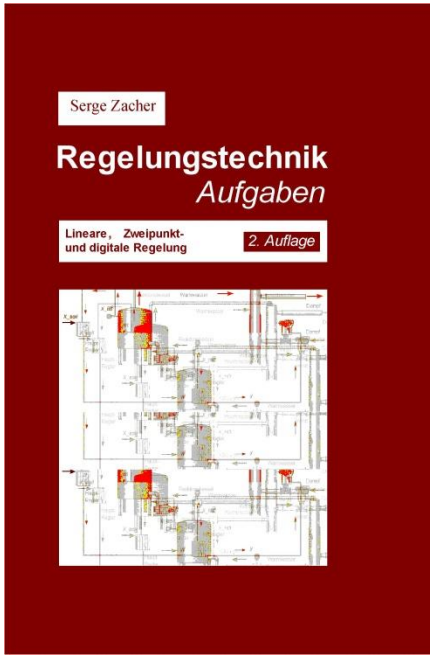
Sprungantwort des analogen Kreises



Sprungantwort des digitalisierten Kreises

4. Anhang: z-Transformation einer P-T2 Strecke

nach dem Buch S, Zacher, „Regelungstechnik-Aufgaben“,
3. Auflage, 2012, Aufgabe 4.2.11, Seite 112



$$G_S(s) = \frac{K_{PS}}{(1+sT_1)(1+sT_2)}$$

$K_{PS} = 0{,}5$

$T_1 = 1{,}5\,\text{s}$

$T_2 = 0{,}5\,\text{s}$

Abtastzeit $T = 0{,}1$ s

$$G_{HS}(z) = \frac{z-1}{z} \cdot Z\left[\frac{G_S(s)}{s}\right] = \frac{z-1}{z} \cdot Z\left[\frac{0{,}5}{s(1+sT_1)(1+sT_2)}\right]$$

$$G_{HS}(z) = \frac{0{,}5}{T_1 T_2}\frac{z-1}{z} \cdot Z\left[\frac{ab(a-b)}{s(s+a)(s+b)}\right] = 1{,}11\frac{z-1}{z} \cdot Z\left[\frac{ab(a-b)}{s(s+a)(s+b)}\right]$$

$$Z\left[\frac{ab(a-b)}{s(s+a)(s+b)}\right] = \frac{N_z(z)z}{(z-1)(z-a_1)(z-b_1)}$$

$$a_1 = e^{-aT} = e^{-\frac{T}{T_2}} = e^{-0{,}33} = 0{,}7189$$

$$b_1 = e^{-bT} = e^{-\frac{T}{T_1}} = e^{-0{,}067} = 0{,}9352$$

$$N_z(z) = (a - b - ae^{-bT} + be^{-aT})z + [(a-b)e^{-(a+b)T} - ae^{-aT} + be^{-bT}]$$

$$N_z(z) = 0{,}0274 \cdot z + 0{,}0157$$

$$Z\left[\frac{ab(a-b)}{s(s+a)(s+b)}\right] = \frac{N_z(z)z}{(z-1)(z-a_1)(z-b_1)}$$

$$G_{HS}(z) = 1{,}11\frac{z-1}{z} \cdot Z\left[\frac{ab(a-b)}{s(s+a)(s+b)}\right]$$

$$G_{HS}(z) = \frac{1{,}11}{5{,}94}\frac{z-1}{z} \cdot Z\left[\frac{5{,}94}{s(s+a)(s+b)}\right]$$

$$G_{HS}(z) = 0{,}187\frac{z-1}{z} \cdot \frac{(0{,}0275z + 0{,}0157)z}{(z-1)(z-0{,}7189)(z-0{,}9352)}$$

$$G_{HS}(z) = 0{,}187\frac{(0{,}0275z + 0{,}0157)}{(z-0{,}7189)(z-0{,}9352)}$$

5. Zusammenfassung

Die vorliegende Publikation ist in erster Linie für Teilnehmerinnen und Teilnehmer des Unterrichts „MBSE: Modellbasierte Softwareentwicklung“ geeignet.

Die Simulation mit MATLAB/Simulink ist sowohl für analoge, als auch für digitale Regelkreise möglich. Dagegen erfolgt die automaische C-Code Generierung mit *MATLAB Coder*, *Simulink Coder* und *PLC Coder* nur für digitale Kreise. Ab Versionen R2014 und R2015 ermöglicht MATLAB/Simulink auch die Digitalisierung von analogen Simulink-Modellen mit den Menü-Befehlen

Analysis → *Control Design* → *Model Discretizer*

automatisch durchführen. Jedoch zuerst soll die Simulation im *Solver* von *Variable-Step* in *Fixed-Step* umgestellt werden.

In der vorliegende Publikation wurde Schritt für Schritt beschrieben, wie man an *Solver* und *Model Discretizer* gelangt und wie die Umstellungen gemacht werden.

Ein Simulink-Modell wird damit automatisch von Laplace-Transformierten Übertragungsfunktionen $G(s)$ in die z-transformitierte Übertragungsfunktionen $G(z)$ umgewandelt. Dies erkennt man sofort durch den Schriftzug *zoh*, der auf allen digitalisierten Bausteinen nach der Ausführung des *Model Discretizer* erscheint.

Obwohl dieser Weg für die Code-Generierung zu empfehlen ist, wurde auch die Digitalisierung mittels z-Transformation beschrieben und an einem Beispiel verdeutlicht.