



TU Berlin Fakultät IV  
Fachgebiet Regelungstechnik  
Praktikum Grundlagen der Regelungstechnik

# **Praktikum Regelungstechnik**

## **Versuch 3**

Dirk Barbendererde (321 836)  
Boris Henckell (325 779)

9. Juli 2012

Gruppe: G1 Di 12-14

Betreuer: Markus Valtin

# Inhaltsverzeichnis

<b>1</b>	<b>Reglerentwurf</b>	<b>1</b>
1.1	PID-Regler . . . . .	1
1.2	Padé-Approximation . . . . .	2
1.3	Reglerentwurf für die Strecke $\hat{G}$ . . . . .	2
1.4	Smith-Prädiktor . . . . .	4
<b>2</b>	<b>Simulation</b>	<b>4</b>
2.1	Stabilität mit steigender Totzeit . . . . .	4
2.2	Regelkreis mit PID-Regler . . . . .	6
2.3	Smith-Prädiktor . . . . .	6
2.4	Padé-Approximation . . . . .	6
2.5	Führungsverhalten beim Störfall . . . . .	7
<b>3</b>	<b>Durchführung</b>	<b>7</b>
3.1	Arbeitspunkttemperatur . . . . .	7
3.2	Führungssprung . . . . .	7
<b>4</b>	<b>Auswertung</b>	<b>9</b>
4.1	Auswirkungen von Totzeiten . . . . .	9
4.2	Reglerentwurf . . . . .	9
<b>5</b>	<b>Scilabcode</b>	<b>9</b>

# 1 Reglerentwurf

## 1.1 PID-Regler

Aufgabe:

Entwerfen Sie für das totzeitfreie System

$$\tilde{G}(s) = \frac{V_0}{(s - s_{\infty 1})(s - s_{\infty 2})}$$

einen (realen) PID-Regler  $K_{PID}$ , indem sie beide Polstellen der Strecke kürzen. Die übrigen Parameter sollen so bestimmt werden, dass die Dynamik des Führungsverhaltens des resultierenden Regelkreises mit dem des folgenden Polpaares übereinstimmt:

$$P(s) = \frac{1}{s^2 \omega^2 + s \frac{2d}{\omega} + 1}$$

Der Regelkreis soll also relativ schnell sein, jedoch ohne dass dabei Überschwingen auftritt.

Zu Beginn des Reglerentwurfs definieren wir die uns vorgegebene linearisierte Strecke  $\tilde{G}$ .

Anschließend erstellen wir folgendermaßen unseren PID-Regler:

$$\tilde{K} = K_{PID} \frac{1}{1 + T_s} \frac{1}{s} (s^2 T_d T_i + T_i s + 1)$$

mit

$$T_i = \frac{-(s_1 + s_2)}{s_1 s_2}$$
$$T_d = \frac{1}{-(s_1 + s_2)}$$

In diesem Fall stellen die beiden Variablen  $s_1$  und  $s_2$  die Polstellen der Strecke  $\tilde{G}$  dar. Auf diese Weise kürzen wir diese schon mal durch den Regler. Verbleiben noch die Position des Realisierbarkeitspols sowie die Verstärkung des Reglers als Parameter um die Dynamik des Führungsverhaltens des Reglers zu beeinflussen.

Als Vergleich erstellen wir die Führungssprungantwort des gegebenen Polpaares.

Mit einer Verstärkung von  $K_{PID} = 1.6$  und einem Realisierbarkeitspol bei  $-20$  erhalten wir folgende Führungssprünge:

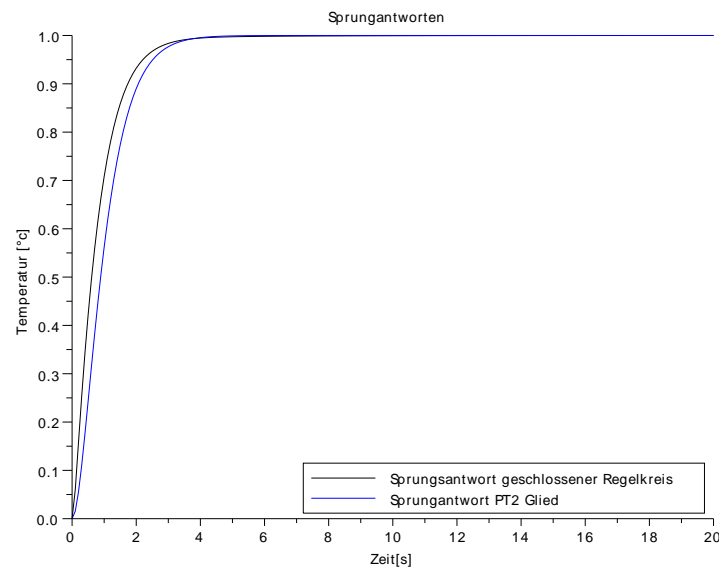


Abb. 1: Führungssprungantwort

## 1.2 Padé-Approximation

Aufgabe:

Approximieren Sie den Term  $e^{-sT_d}$ , welcher für die Totzeit verantwortlich ist, mithilfe einer Padé-Approximation erster Ordnung durch eine gebrochen-rationale Funktion und stellen Sie das approximierte Streckenmodell  $\tilde{G}$  auf, indem Sie die Totzeit in  $G$  durch ihre Approximation ersetzen.

Als nächstes haben wir die Totzeit mit einer Padé-Approximation erster Ordnung durch eine gebrochen-rationale Funktion ersetzt und multiplizieren sie an unsere Strecke. Für diese Näherung des Systems ( $\hat{G}$ ) entwerfen wir nun einen Regler.

Die Approximation lautet:

$$p_1(s) = \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s}$$

## 1.3 Reglerentwurf für die Strecke $\hat{G}$

Aufgabe:

Es soll ein Regler für das approximierte Streckenmodell  $\hat{G}$  mittels Polvorgabe entworfen werden, der folgende Eigenschaften des geschlossenen Kreises ermöglicht:

- kein Überschwingen der Regelgröße bei sprungförmigen Führungs- oder Störsignalen
- ungefähr gleiche Anstiegszeit der Regelgröße wie bei der Verwendung des PID-Regler aus 1.1
- Regelfehler  $\rightarrow 0$  für  $t \rightarrow \infty$  unter sprungförmigen Referenzen

1. Stellen Sie den Regleransatz mit kleinstmöglicher Nennerordnung und Integratoranteil auf. Wie viele Pole müssen Sie vorgeben?
2. Stellen Sie das Sollpolynom auf; verwenden sie hierzu die Pole des Polpaares aus Aufgabe 1.1 und die Pole der Strecke  $G$ , wählen sie einen weiteren Pol bei ( $s_\infty = -2$ ).
3. Stellen Sie die Sylvester Matrix durch Koeffizientenvergleich des Polpolynoms des geschlossenen Kreises mit Ihrem Sollpolynom auf und berechnen sie die Reglerparameter mithilfe von Scilab.

Mit dem Regleransatz

$$K(s) = \frac{\beta_0 + \beta_1 s + \beta_2 s^2 + \beta_3 s^3}{s(\alpha_0 + \alpha_1 s + s^2)}$$

Das approximierte Streckenmodell  $\hat{G}$  allein hat die Ordnung  $n = 3$ . Da der Regler einen Integratoranteil besitzen soll und wir den Reglerentwurf wählen bei dem dieser Integrator im nachhinein an einen Regler multipliziert wird, müssen wir die Strecke anpassen und erhalten eine neue Streckenordnung von  $n = 4$ . Daraus folgt, dass die zu berechnende Sylvestermatrix  $8 \cdot 8$  Einträge und das Sollpolynom 8 Einträge besitzen muss. Um das zu realisieren müssen wir insgesamt 7 Pole vorgeben, da der letzte Eintrag des Sollpolynoms der Koeffizient vor dem  $s^0$  ist.

Neben dem Pol bei  $s_\infty = -2$  und den zwei Polen des Polpaares benötigen wir noch genau 4 Pole aus der veränderten Strecke. Unsere Strecke ergibt sich aus:

$$\hat{G}_I = \hat{G} \frac{1}{s}$$

und hat genau 4 Pole.

Insgesamt ergibt sich daraus das folgende Sollpolynom:

$$q_{soll} = 32.148065s + 177.77307s^2 + 285.03595s^3 + 208.51884s^4 + 77.572633s^5 + 14.179911s^6 + s^7$$

Anschließend haben wir die Sylvestermatrix aufgestellt und mithilfe des Sollpolynoms die Koeffizienten des Reglers errechnet. Unser Regler sieht folgendermaßen aus:

$$K_{posI} = \frac{14.231104 + 56.480246s + 26.455258s^2 + 3.1456906s^3}{62.570515s + 28.849556s^2 + 5s^3}$$

Damit entspricht er dem geforderten Regleransatz. Die Kontrolle der Polstellen des resultierenden geschlossenen Regelkreises ergibt, dass sie mit denen des Sollpolynoms übereinstimmen.

## 1.4 Smith-Prädiktor

Aufgabe:

Entwerfen Sie für die totzeitbehaftete Strecke einen Smith-Prädiktor unter Verwendung des Reglers aus 1.1

Der Smith-Prädiktor für die totzeitbehaftete Strecke unter Verwendung des Reglers aus 1.1 ergab sich wie folgt:

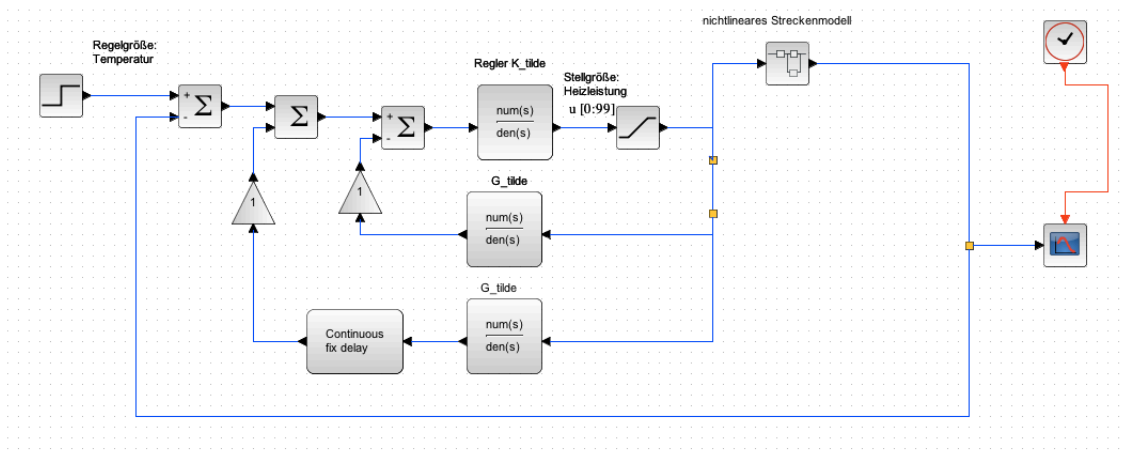


Abb. 2: Smith-Prädiktor

## 2 Simulation

### 2.1 Stabilität mit steigender Totzeit

Aufgabe:

Simulieren Sie den PID Regler zunächst ohne Totzeit mit einem Führungssprung der Amplitude  $+30^{\circ}\text{C}$  mit dem idealen  $PT_2$ -Modell! Fügen Sie dem Modell solange größer werdende Totzeiten ( $T_d = 0.4, 0.8, \dots$ ) hinzu, bis der Regelkreis instabil wird! Beschreiben Sie kurz, welchen Einfluss die Totzeit auf das Regelkreisverhalten hat!

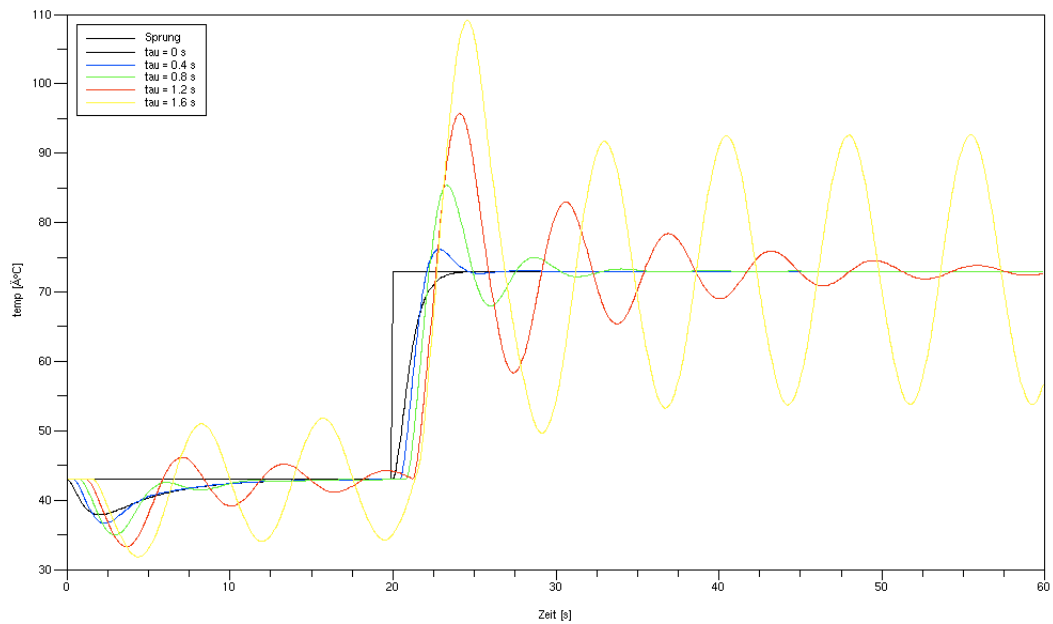


Abb. 3: Simulationsergebnisse

Anhand der fünf Simulationen mit steigender Totzeit lässt sich erkennen, dass die Sprungantwort zunehmend überschwingt. Bei dieser Simulation weiß der Regler nichts von der Totzeit und erwartet eine sofortige Reaktion auf seine Regelung. Da diese jedoch erst verspätet beim Regler ankommt übersteuert er. Je größer die Totzeit ist umso mehr fällt dieser Effekt auf. In unserem Fall schafft es der Regler für eine Totzeit von 1.6 s nicht mehr, aus dem Überwigen rauszukommen und wir instabil.

Die folgenden 5 Simulationen sind im nächsten Bild zusammengefasst und werden nachfolgend betrachtet.

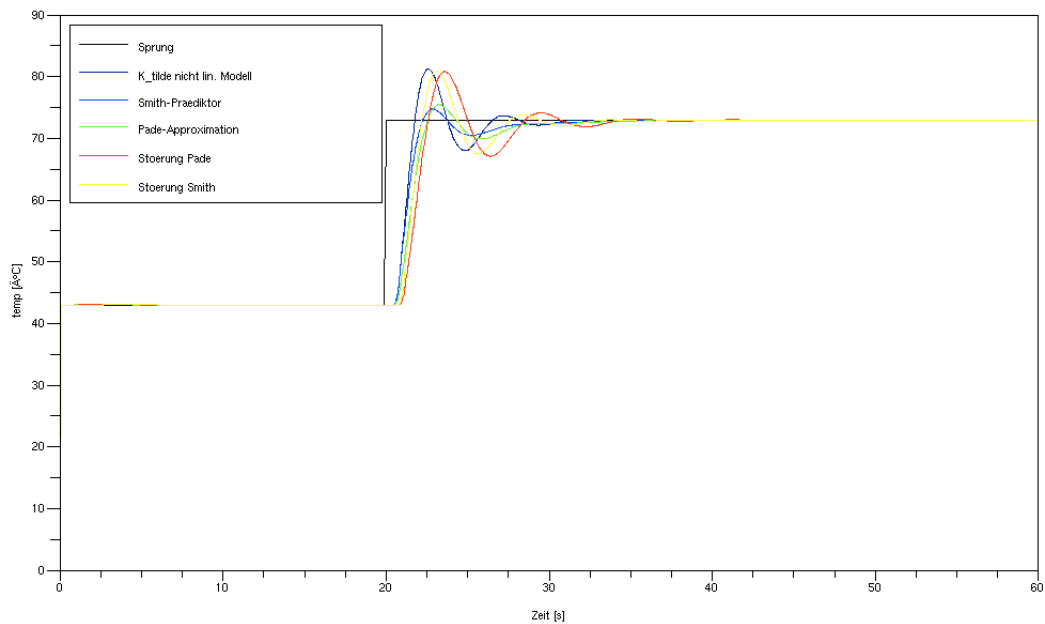


Abb. 4: Simulation rest

### 2.2 Regelkreis mit PID-Regler

Aufgabe:

Simulieren Sie den Regelkreis mit PID-Regler (ab hier immer mit Totzeit und dem vorgegebenen Scicos-Modell von der Webseite) unter einer sprungförmigen Referenz der Amplitude  $+30^{\circ}\text{C}$  ! Kommentieren Sie kurz Ihre Beobachtungen!

Bei dieser Simulation wurde anstatt des linearisierten Modells der Strecke, ein nichtlinearisiertes Modell verwandt. Es fällt auf, dass bei diesem Modell die Temperatur des Arbeitspunktes von Anfang an konstant ist. Die Sprungantwort reagiert mit auffallendem Überschwingen als das lineare Modelle mit der selben Totzeit. Das liegt unserer Meinung nach an dem genaueren Modell der Strecke. Bisher wurde noch kein Versuch unternommen die Totzeit zu kompensieren

### 2.3 Smith-Prädiktor

Aufgabe:

Implementieren Sie ihren Smith-Prädiktor und simulieren Sie erneut! Was ändert sich, was bleibt gleich?

Bei dieser Simulation wurde ein Smith-Prädiktor hinzugefügt. Es ist ein auffallender Unterschied zu erkenne. zwar schwingt die Antwort noch ein wenig über jedoch bei weitem nicht soviel wie ohne Smith-Prädiktor.

### 2.4 Padé-Approximation

Aufgabe:

Implementieren Sie nun auch den Regler auf Basis der Padé-Approximation und simulieren Sie einen Führungssprung  $+30^{\circ}\text{C}$  ! Wie ist die Regelgüte im Vergleich zum PID-Regler mit Smith-Prädiktor?



Bei dem Regler mit Padé Approximation verschlechtert sich das Sprungverhalten wieder. Das Überspringen nimmt zu und auch die reaktion verlangsamt sich. Das ist auch nicht verwunderlich, da unter anderem die Totzeit bei diesem Regler nur angenähert wurde, während die Totzeit beim Smith-Prädiktor durch ein "Delay"-Block dargestellt wurde, der genauer arbeitet als die Approximation.

## 2.5 Führungsverhalten beim Störfall

Aufgabe:

Erproben sie nun einen Störfall: Erhöhen Sie die Totzeit (nur die des Modells, die Regler bleiben die gleichen) auf  $T_d = 0.7s$  und simulieren Sie erneut das Führungsverhalten beider Regelkreise! Kommentieren Sie kurz das Ergebnis!

Bei der verwendeten Störung sind beide Sprungantworten wieder von einem Starken überspringen geprägt. Jedoch schwingt der Regler mit Smith-Prädiktor weniger über, ist also deutlich stailier. Wir erwarten daher, dass er in jeder hinsicht besser ist als der Regler mit Padé-Approximation.

## 3 Durchführung

### 3.1 Arbeitspunkttemperatur

Aufgabe:

Ermitteln Sie die Temperatur, die sich bei einer Heizleistung von 10 einstellt und verwenden Sie diese als Arbeitspunkttemperatur!

TODO:

SICHER??? simulation: start bei  $30^\circ$  sprung um  $31^\circ$

Der Arbeitspunkt mit einer Heizleistung von 10 lag bei  $31^\circ C$ . Da die Außentemperatur sehr hoch war hat sich das System jedoch nicht immer wieder vollständig auf den Arbeitspunkt abgekühlt, weshalb einige Messungen bei etwas über dem Arbeitspunkt starten.

### 3.2 Führungssprung

Aufgabe:

Implementieren Sie den PID-Regler (zunächst ohne Smith-Prädiktor) am Versuchsstand und führen Sie einen Führungssprung um  $+30^\circ C$  aus! Die Stellgrößen als auch die Temperatur sind jeweils für jeden Versuch aufzuzeichnen.

Fügen Sie nun den Smith-Prädiktor hinzu und starten Sie wiederum das Experiment!

Erproben Sie ebenfalls den Regler auf Approximationsbasis!

### 3 DURCHFÜHRUNG

Nun haben wir alle drei Regler nacheinander implementiert und getestet. Wir haben, um die Führungssprungantwort zu messen, einen Sprung von  $31^{\circ}\text{C}$  nach  $61^{\circ}\text{C}$  auf die Regelstrecke gegeben.

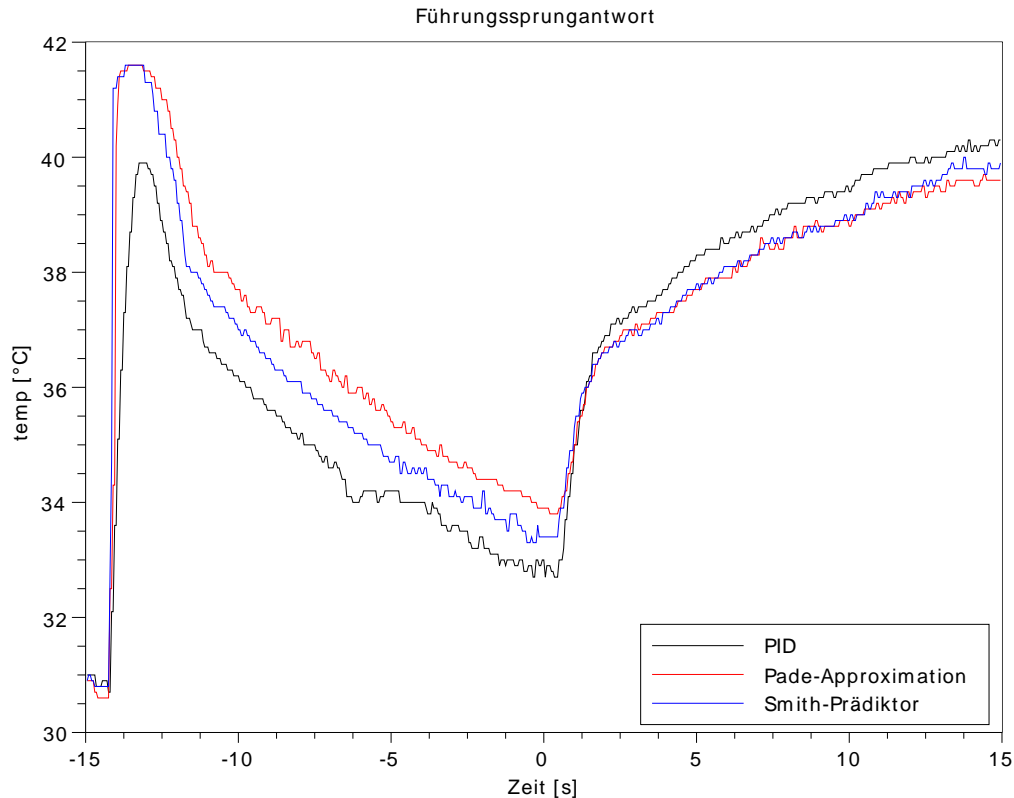


Abb. 5: Gemessene  
Führungssprungantwort

Unerwarteter Weise verhielt sich das System überhaupt nicht wie Simuliert. Nähere Nachforschungen ergaben, dass sich die Parameter der echten Strecke zu stark von denen des vorgegeben Modells unterscheiden hatten. Die Einstellungen des Systems waren beim Versuchsaufbau also deutlich anders als wir beim Entwurf der Regler angenommen hatten.

Vor allem war die Streckenverstärkung deutlich geringer als wir annahmen, weshalb die Verstärkung unserer Regler nicht ausreichend war. In Abb. 5 ist deutlich zu erkennen, dass es alle unsere Regler nach 15 Sekunden gerade einmal geschafft haben auf  $40^{\circ}$  zu heizen obwohl sie schon nach ca. 5 Sekunden bei  $60^{\circ}$  hätten sein sollen.

Somit sind die Messergebnisse offensichtlich nicht mit den Simulierten vergleichbar. Aufgrund des falschen Reglerentwurfs sind leider auch keine rückschlüsse auf die Eignung bzw. Qualität der Regler möglich.

## 4 Auswertung

### 4.1 Auswirkungen von Totzeiten

Anhand der Simulationen haben wir beispielhaft herausgefunden wie sich Totzeiten auf ein Regelsystem auswirken.

Falls die Totzeit eines Systems beim Reglerentwurf nicht berücksichtigt wird kommt es, je nach Länge der Verzögerung, zu Überschwingen der Sprungantwort. Genügend große Totzeiten können den Regler sogar instabil werden lassen, falls der Regler während der Totzeit so doll weiter regelt, dass der Integrator über die Begrenzung hinweg voll läuft und so eine Art Windup-Effekt auftritt.

### 4.2 Reglerentwurf

Für den Reglerentwurf bei Strecken mit Totzeit hat sich in unseren Simulationen der Smith-Prädiktor durchgesetzt. Sowohl das Führungssprung- als auch das Störverhalten waren besser als beim Regler mit Padé Approximation. Hierbei spielt es wahrscheinlich auch eine Rolle, dass bei letzterem für die Totzeit nur eine Näherung verwendet wird, wohingegen der Smith-Prädiktor eine genauere Verzögerungskomponente hat.

Schlussendlich hat der Smith-Prädiktor zwar eine höhere Reglergüte, der Reglerentwurf mittels einer Approximierten Totzeit ist aber einfacher und mit weniger Aufwand um zu setzen..

## 5 Scilabcode

Listing 1: Scilab-script

```
1 // Regelungstechnik Versuch 3
2
3
4 // Boris: cd "/Users/borishenckell/Documents/eclipse workspace/PR_Regelungsrechnik_-_Versuch_3/PR_Regelungsrechnik
   // - Versuch 3"
5 // Dirk:
6 // cd "/media/daten/workspace/PR_Regelungsrechnik_-_Versuch_3/PR_Regelungsrechnik - Versuch 3/Scilab/"
7
8
9
10 s = poly(0, 's');
11
12
13 Kpid = 1.355; // Verstärkung
14 s1 = -0.29; // Polstellen der Strecke
15 s2 = -3.12; // Polstellen der Strecke
16 p = -3.6 // Realisierbarkeitspole
17 T = 1/-p;
18 Ti = -(s1+s2)/(s1*s2);
19 Td = 1/-(s1+s2);
20
21 G_tilde = 2.259 / ((s - s1) * (s - s2));
22
```

```

23 K_tilde = Kpid * (1/(1+T*s))*(1/Ti)*(1/s)*(s^2*Td*Ti+Ti*s+1);
24
25
26 // zum Vergleichen
27 w = 2 * %pi * 0.3;
28 d = 1;
29
30 P = 1/ (s^2/w^2+s*2*d/w + 1);
31
32 polpaar = roots(P.den);
33
34
35 // Sprungantwort
36 GKgeschlossen_tilde = (G_tilde*K_tilde)/(1+G_tilde*K_tilde);
37 GKgeschlossen_tilde = syslin('c', clean(real(GKgeschlossen_tilde.num)),clean(real(GKgeschlossen_tilde.den)));
38
39 t1=[0:0.1:100];
40 // u=ones(1,length(t1))*0.1;
41 h1=csim('step',t1,GKgeschlossen_tilde);
42 h2=csim('step',t1,P);
43
44 // clf(15);scf(15);
45 // plot2d(t1,h1)
46 // plot2d(t1,h2,2)
47 // legend('Sprungsantwort geschlossener Regelkreis', 'Sprungantwort PT2 Glied',4);
48 xtitle('Sprungantworten', 'Zeit[s]', 'Temperatur_[c]')
49
50
51 // Pade-Approximation
52
53 tau = 0.4; // 0.4
54
55 p1 = (1-tau/2*s)/(1+tau/2*s);
56 G_dach=G_tilde*p1;
57
58
59 // Reglerentwurf mit Polvorgabe
60
61
62 // erstellen der Streckenmatrix
63 G_dachI = G_dach*1/s;
64 nl=length(roots(G_dachI.den));
65 AsI=zeros(2*nl,2*nl);
66 // As(Zeile,Spalte)
67 for k = 1:2*nl+1
68     for l = 1:2*nl
69         if k<nl+2 then
70             if l < nl+1 then
71                 AsI(k+l-1,l) = coeff(G_dachI.den,nl+1-k);
72             end
73             if l > nl then
74                 AsI(k+l-1-nl,l) = coeff(G_dachI.num,nl+1-k);
75             end
76         end
77     end
78 end
79
80 pol_G_dachI = roots(G_dachI.den);
81 qsoll = (s-pol_G_dachI(1))*(s-pol_G_dachI(2))*(s-pol_G_dachI(3))*(s-pol_G_dachI(4))*(s-polpaar(1))*(s-polpaar(2))*(s+2);

```

```

82 qsoll = clean(qsoll);
83
84 cvekl = coeff(qsoll)';
85 cvekl = cvekl([1 8 7 6 5 4 3 2]);
86
87 kcoeffl=clean(inv(Asl)*cvekl);
88 Kposl = syslin('c', kcoeffl(nl+1)*s^(nl-1)+kcoeffl(nl+2)*s^(nl-2) + kcoeffl(nl+3)*s^(nl-3)+kcoeffl(nl+4)*s^(nl-4),s*(kcoeffl
    (1)*s^(nl-1)+kcoeffl(2)*s^(nl-2)+kcoeffl(3)*s^(nl-3)+kcoeffl(4)*s^(nl-4)));
89
90 Kposl = clean(Kposl);
91
92 // Führungssprung
93 // GKgeschlossen = Kposl*G_dach/(1+Kposl*G_dach)
94
95 GKgeschlossen = syslin('c', clean(Kposl.num*G_dach.num),clean(Kposl.den*G_dach.den + Kposl.num*G_dach.num))
96
97 // clean(GKgeschlossen);
98
99 // GKgeschlossen = syslin('c', GKgeschlossen.num,real(GKgeschlossen.den));
100
101 S = syslin('c', clean(Kposl.den*G_dach.den)/(clean(Kposl.den*G_dach.den) + clean(Kposl.num*G_dach.num)))
102 S = syslin('c', S.num,real(S.den));
103 S = clean(S);
104
105
106 h3=csim('step',t1,GKgeschlossen);
107 h4=csim('step',t1,S);
108
109 // scf(16); clf(16)
110 // plot2d(t1,h3);
111
112 // scf(17); clf(17);
113 // plot2d(t1,h4);
114
115
116 // clf(18);scf(18);
117 // plot2d(t1,h3)
118 // plot2d(t1,h4,2)
119 // legend('T', 'S',4);
120 // xtitle ('Sensitivitäts- und komplementäre Sensitivitätsfunktion', 'Zeit[s]', 'K.A.')
121
122 // ---- pdf abspeichern ---- //
123
124 // xs2pdf(15,'./ Bilder/Sprungantwort.pdf');
125 // xs2pdf(16,'./ Bilder/GKgeschlossen.pdf');
126 // xs2pdf(17,'./ Bilder/S.pdf');
127 // xs2pdf(18,'./ Bilder/S_T_Vergleich.pdf');

```