Chapter 12
Discrete Regulator Design for Stable Processes

12.1 Design of a Youla Parameterized Regulator

The Youla parameterized control can be applied also to sampled systems. The regulator design is similar to the procedure discussed in Chap. 7. In sampled data systems, the realization of the regulator for processes with dead-time does not cause any problems.

The sampled system is given by its pulse transfer function G. The pulse transfer function of the process has to be separated into the cancellable G_+ and the non-cancellable G_- components. The discrete dead-time is denoted by d.

$$G = G_+ G_- z^{-d}$$

In the case of lag elements there is always a z^{-1} term, so the discrete dead-time d is calculated as the ratio of the real, physical dead-time T_d and the sampling time T_s plus one, i.e., $d = \mathrm{entier}(T_\mathrm{d}/T_\mathrm{s}) + 1$. Preferably choose the sampling time so that the ratio $T_\mathrm{d}/T_\mathrm{s}$ is an integer.

R_r is the pulse transfer function of the reference model (reference filter) and R_n is the pulse transfer function of the disturbance filter.

The expression of the Youla parameter is: $Q = R_\mathrm{n} G_+^{-1}$.

The block diagram of the Youla parameterized discrete control system in *IMC* form is given in Fig. 12.1. (An equivalent circuit is shown in Fig. 12.3 of the textbook [1].)

If the disturbance is zero and the process and its model are the same, then the feedback signal is zero, and the reference signal tracking is realized according to

$$y = R_\mathrm{r} G_- z^{-d}\, r.$$

L. Keviczky et al., *Control Engineering: MATLAB Exercises*,
Advanced Textbooks in Control and Signal Processing,
https://doi.org/10.1007/978-981-10-8321-1_12

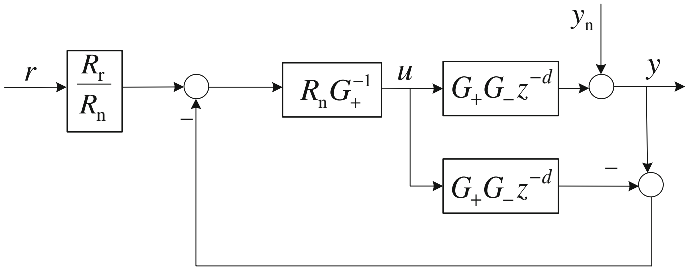


Fig. 12.1 Block diagram of the Youla parameterized discrete control system

For reference signal tracking without static error, the static gain of G_- and R_r should be 1, i.e. $G_-(z=1)=1$ and $R_r(z=1)=1$. The resulting transfer function for the disturbance input is

$$y = \left(1 - R_n G_- z^{-d}\right) y_n.$$

To ensure a zero steady output value in case of a step disturbance (total disturbance rejection) the static gain of R_n should be also 1, i.e. $R_n(z=1)=1$.

The zero of the process which is outside of the unit circle should not be cancelled, as this would result in an unstable pole in the regulator. Also it is not expedient to cancel zeros which are on the left side of the unit circle, as this would introduce a pole in the regulator which would cause intersampling oscillations in the control system. (This phenomenon will be analysed in more detail in the discussion of dead-beat control.)

Example 12.1 Consider the dead-time process analysed in Example 7.3.

The transfer function of the process is

$$P(s) = \frac{1}{(1+5s)(1+10s)} e^{-30s}.$$

The sampling time is $T_s = 1$ s.

Design a Youla parameterized regulator for this process. First the filters are chosen to provide a delay of one sampling time. Then analyse how is the behaviour of the control system modified if the filters are obtained by sampling the continuous filters given by the transfer function $R_r(s) = R_n(s) = \frac{1}{(1+s)^2}$.

Write a MATLAB™ program whose input data are the invertible and non-invertible factors G_+ and G_- of the pulse transfer function of the process, and the filters are R_r and R_n. The program has to calculate and plot the output and control signals of the control system for unit step reference signal and zero output disturbance, then the output and the control signals for zero reference signal and unit step output disturbance.

Save the program with the name Youla_discrete.

```
% Youla_discrete: Youla discrete basic program
display('.....Q='),Q=minreal(Rn/Gp,0.0001)
display('.....C='),C=minreal(Q/(1-Q*G),0.0001)
display('.....L='),L=minreal(C*G,0.0001)
display('.....Tr='),Tr=minreal((Rr/Rn)*Q*G,0.0001)
display('.....Ur='),Ur=minreal((Rr/Rn)*Q,0.0001)
pause
t=0:Ts:50;
figure(1)
yr=step(Tr,t);
subplot(211), plot(t,yr,'*'),grid
ur=step(Ur,t);
subplot(212), stairs(t,ur),grid
pause;
display('.....Sn='),Sn=minreal((1-Q*G),0.0001)
display('.....Un='),Un=minreal(-C*(1-Q*G),0.0001)
pause
figure(2)
yn=step(Sn,t);
subplot(211),plot(t,yn,'*'),grid;
un=step(Un,t);
subplot(212),stairs(t,un),grid;
```

Give the process in MATLAB™.

```
clear; clc; s=zpk('s')
P=1/((1+5*s)*(1+10*s))
Ts=1; z=zpk('z',Ts); G1=c2d(P,Ts)
G=G1*z^(-30)
```

The pulse transfer function $G(z)$ considering also the dead-time is

```
0.0090559 (z+0.9048)
-------------------- z^-30
(z-0.9048) (z-0.8187)
```

Separate the pulse transfer function of the process into cancellable and non-cancellable factors. First suppose that the whole dynamics can be cancelled.

```
Gm=1;          %G-
Gp=G1/Gm       %G+
```

Set the filters as

```
Rr=1/z; Rn=1/z;
```

Call the Youla_discrete program.

Youla_discrete

Figure 12.2 gives the output signal (upper figure) and the control signal (lower figure) for unit step reference signal. Figure 12.3 shows the output and the control signals for unit step output disturbance. It is a strange phenomenon that the oscillating control signal (on the lower figures) results in a calm output signal. The explanation is that the simulation is executed only at the sampling points. Simulating the real continuous process decreasing oscillations can be observed between the sampling points.

Build the SIMULINK™ block diagram corresponding to Fig. 12.1. Let the process be considered with its continuous model (Fig. 12.4). Running the simulation it can be seen that really there are oscillations between the sampling points (Fig. 12.5).

Fig. 12.2 Output and control signals for step reference signal

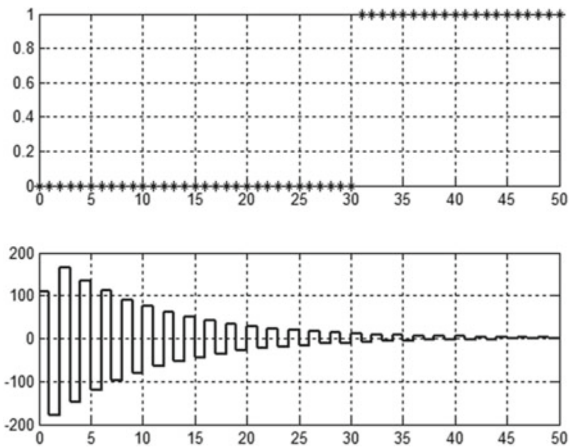
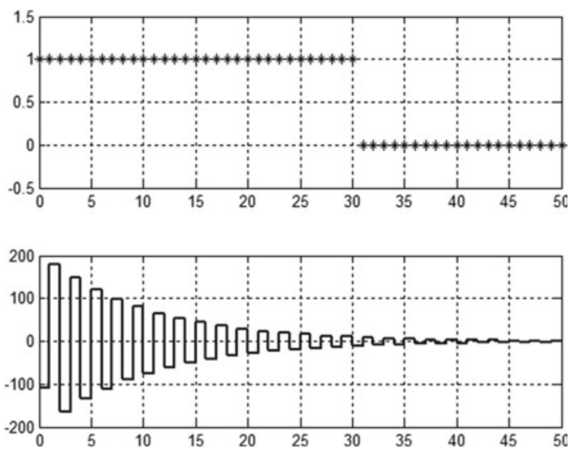


Fig. 12.3 Output and control signals for step output disturbance signal



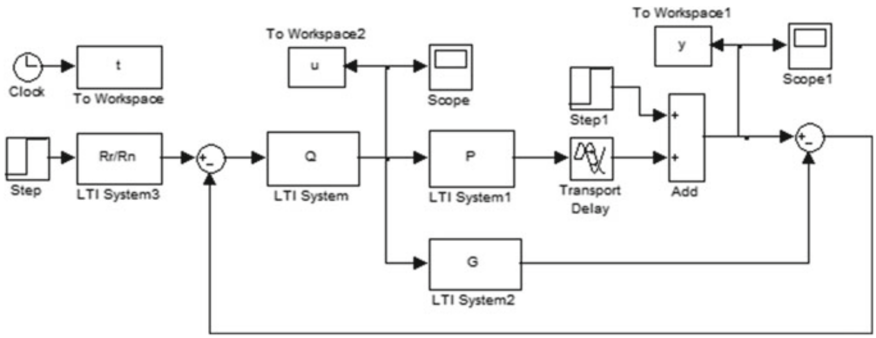
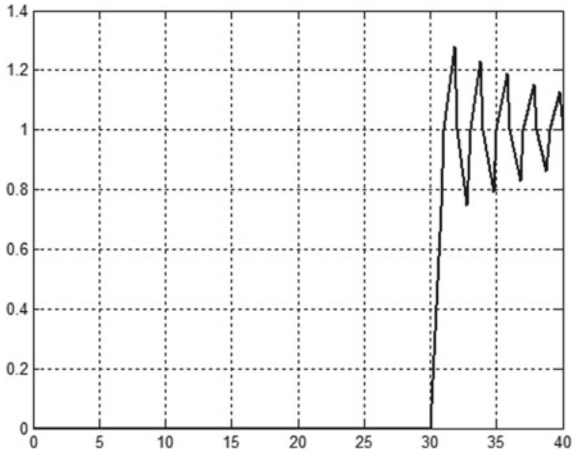


Fig. 12.4 SIMULINK™ diagram for the Youla parameterized control system

Fig. 12.5 Oscillations between the sampling points



In the following the non-cancellable part contains a zero on the left side of the unit circle.

The separation is as follows:

$$G_- = \frac{1 + 0.9048z^{-1}}{1.9048}; \quad G_+ = \frac{0.0090559 \cdot 1.9048}{(1 - 0.9048z^{-1})(1 - 0.8187z^{-1})}.$$

With MATLAB commands:

```
Gm =(1+0.9048*z^(-1))/1.9048
Gp = minreal(G1/Gm,0.0001)
```

Then call the program:

```
Youla_discrete
```

Figure 12.6 shows the output and the control signal for unit step reference signal. The signals for a disturbance input are shown in Fig. 12.7. The control signal reaches a steady value after two jumps, and the output signal shows calm behaviour. Running the SIMULINK™ model it can be seen that at the output of the continuous process there are no oscillations between the sampling points.

Investigate the operation of the control system with second order filters.

```
Rr=c2d(1/(1+s)^2,Ts);Rn=Rr;
```

Then run the program again.

```
Youla_discrete
```

In Figs. 12.8 and 12.9 it can be seen that the dynamics of the settling process has changed, it became a bit slower.

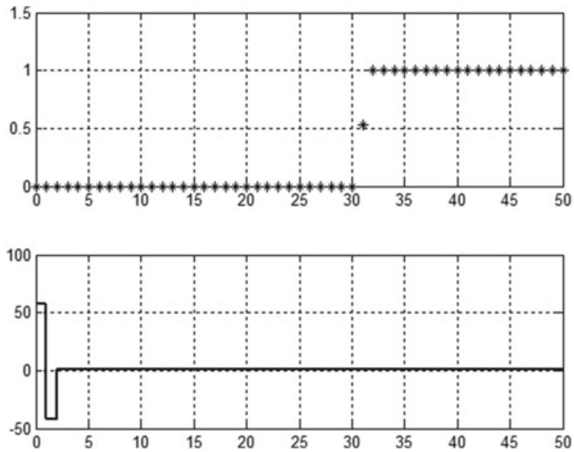


Fig. 12.6 Output and control signals for step reference signal with appropriate separation of *G*

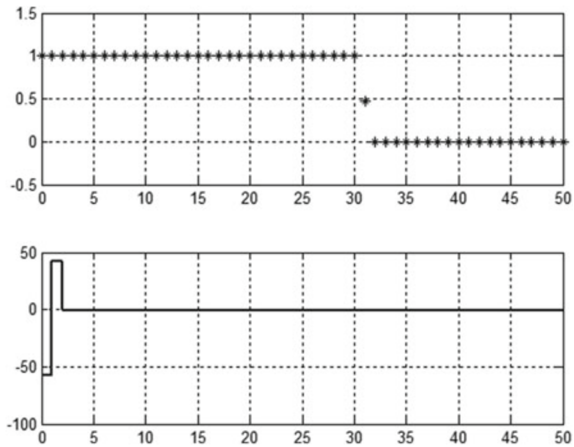


Fig. 12.7 Output and control signals for step output disturbance with appropriate separation of *G*

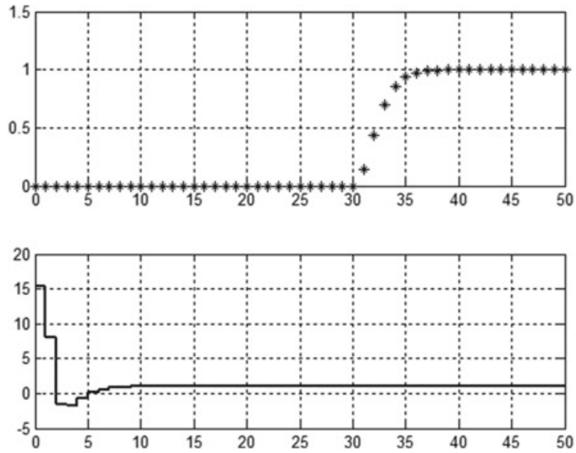


Fig. 12.8 Output and control signals for step reference signal with filters

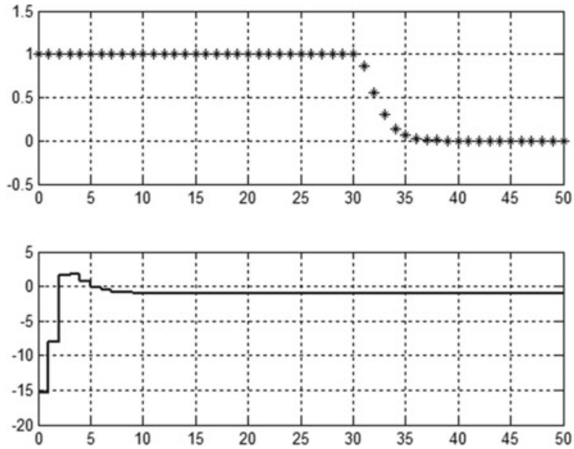


Fig. 12.9 Output and control signals for step output disturbance with filters

The role of filters on the one hand is to influence the dynamic behaviour of the control system. If the two filters are different, the dynamics of reference signal tracking and that of disturbance rejection will be different. In this case it is called *two-degree-of-freedom* (*2DOF*) control. On the other hand, applying the filters, the maximum value of the control signal becomes smaller, as can be seen in Figs. 12.6, 12.7, 12.8 and 12.9. The filters also influence the robustness of the control system. If the process and its model are not exactly the same, i.e. there is a plant-model mismatch, by choosing the appropriate filters generally robust behaviour of the control system can be achieved, i.e. the behaviour of the control system can be acceptable even if the model is not accurate.

Problem Analyse the reference signal tracking and disturbance rejection properties of the control system if the dynamics of the two filters differs.

Execute the simulation with the SIMULINK™ model, both for the case where the process and its model are the same and also if there is a mismatch between them. In the latter case let the dead-time of the system be 40 s, while the dead-time of the model is 30 s. Find filters which ensure acceptable behaviour even in this case.

Example 12.2 Consider Example 12.1 in the text book [1]. Simulate the behaviour of the control system in MATLAB™.

The process is a sampled first order lag element with dead-time.

The pulse transfer function of the process is: $G(z) = \dfrac{0.2}{z - 0.8} z^{-3} = G_+ G_- z^{-d}$.

The pulse transfer functions of the filters are: $R_r(z) = \dfrac{0.8z^{-1}}{1 - 0.2z^{-1}}$ and $R_n(z) = \dfrac{0.5z^{-1}}{1 - 0.5z^{-1}}$.

Embedded filters: $G_r = G_n = 1$.

Separation of the process: $G_+(z) = \dfrac{0.2z^{-1}}{1 - 0.8z^{-1}}$ and $G_-(z) = 1$.

The YOULA parameter: $Q = R_n G_+^{-1} = \dfrac{2.5(1 - 0.8z^{-1})}{1 - 0.5z^{-1}}$.

The series regulator: $C = \dfrac{R_n G_+^{-1}}{1 - R_n G_- z^{-3}} = \dfrac{2.5(1 - 0.8z^{-1})}{1 - 0.5z^{-1} - 0.5z^{-4}}$.

Specify the data of the process and the filters in MATLAB™.

```
z=zpk('z');
Gp=0.2*z^(-1)/(1-0.8*z^(-1));          % G+
Gm=1;                                   % G-
d=3;
G1=minreal(Gp*Gm,0.0001);
G=G1*z^(-d);
Rr=0.8*z^(-1)/(1-0.2*z^(-1));
Rn=0.5*z^(-1)/(1-0.5*z^(-1));
```

Give a fictive sampling time: **Ts=1;**

Then call the program Youla_discrete:

```
Youla_discrete
```

Figure 12.10 shows the dynamics of the reference signal tracking, while Fig. 12.11 gives the dynamics of the disturbance rejection.

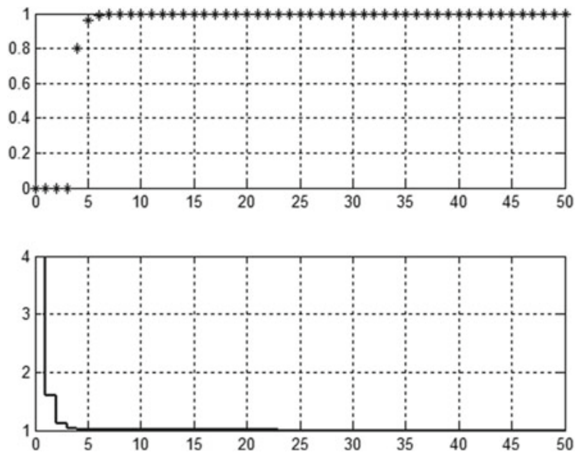


Fig. 12.10 The dynamics of reference signal tracking

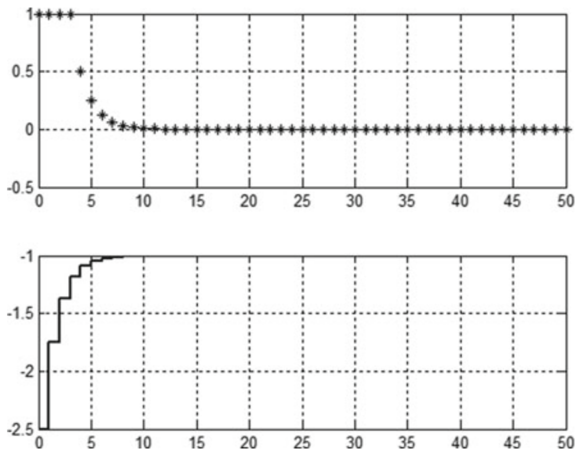


Fig. 12.11 The dynamics of disturbance rejection

12.2 Control of a Dead-Time System with a SMITH Predictor

If the process contains a large dead-time, applying a *PID* regulator, the control system will be slow (this behaviour was demonstrated in the design of a continuous regulator in Chap. 8). To accelerate the control system, a YOULA parameterized regulator can be used, or the SMITH predictor regulator developed earlier, around 1950. The design of a SMITH predictor brings up the question, whether the usual control system of a process with dead-time could be made equivalent to a control system where the dead-time appears outside of the closed loop (Fig. 12.12).

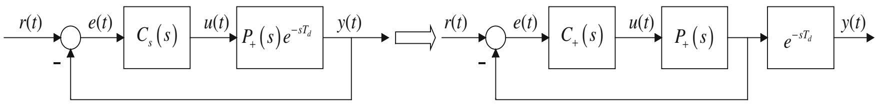


Fig. 12.12 The idea of SMITH predictor

Writing the equivalence of the resulting transfer functions of the two systems yields

$$\frac{C_s(s)P_+(s)e^{-sT_d}}{1+C_s(s)P_+(s)e^{-sT_d}} = \frac{C_+(s)P_+(s)}{1+C_+(s)P_+(s)}e^{-sT_d}$$

$$C_s(s)[1+C_+(s)P_+(s)] = C_+(s)[1+C_s(s)P_+(s)e^{-sT_d}]$$

whence

$$C_s(s) = \frac{C_+(s)}{1+(1-e^{-sT_d})C_+(s)P_+(s)}$$

is the transfer function of the SMITH predictor.

The regulator $C_+(s)$ is designed for the process without dead-time. This control system will be fast. Then the practically applicable regulator is calculated according to the relation above, giving the transfer function of the SMITH predictor. (Let us remark that the signal of the process output before the dead-time is not available.)

It is seen that the dead-time appears in the expression of the regulator.

Theoretically, a SMITH predictor can be used both for continuous and sampled systems, but as the realization of dead-time in continuous systems is difficult and can be solved only approximately, therefore this algorithm generally is applied only in discrete systems. In sampled systems, dead-time means the shift of the signal and therefore it is simply realizable.

The pulse transfer function of the discrete SMITH predictor is

$$C_s(z) = \frac{C_+(z)}{1+(1-z^{-d})C_+(z)G_+(z)}$$

where $G_+(z)$ is the pulse transfer function of the process without the dead-time, and d is the discrete dead-time.

Example 12.3 Design a SMITH predictor regulator for a proportional process containing three time lags and dead-time. The transfer function of the continuous process is

$$P(s) = P_+(s)\,e^{-sT_d} = \frac{K\,e^{-sT_d}}{(1+sT_1)(1+sT_2)(1+sT_3)},$$

with parameters $K = 5$, $T_1 = 10$, $T_2 = 4$, $T_3 = 1$ and $T_d = 10$.

Design the regulator $C_+(s)$ to ensure a phase margin of about 60°. The maximum value of the control signal should not exceed the value umax = 10.

```
K=5; T1=10; T2=4; T3=1; Td=10; ft=60;
```

Define the variable ‘s' by

```
s=zpk('s');
```

Use the *LTI* structures of the *Control System Toolbox* ($P_+(s)$ = Ps)

```
Ps=1/((1+s*T1)*(1+s*T2)*(1+s*T3))
```

First Step: Design of the $C_+(z)$ Discrete Regulator

A good practical rule for the choice of the sampling time is that it should be smaller than the smallest time constant of the process. (For simulation tasks it could be chosen to be about one tenth of the smallest time constant, for control purposes it is appropriate if the sampling time is chosen around one half or one third of the smallest time constant; at most it can be equal to it. Besides, it is expedient to choose the sampling time in such a way that the ratio of the dead-time and the sampling time is an integer.)

```
Ts=0.5;
```

The regulator $C_+(z)$ is designed in the frequency domain according to the low frequency approximation method described in Chap. 13.

The pulse transfer function of the sampled continuous process with a zero-order holding ($G_+(z)$ = Gz) is

```
Gz=c2d(Ps,Ts)
```

The resulting pulse transfer function is

$$G_+(z) = 0.0004416\frac{(z+0.2254)\,(z+3.167)}{(z-0.9512)\,(z-0.8825)\,(z-0.6065)}.$$

The *PID* regulator based on pole cancellation is designed according to Chap. 13.

The pole of $G_+(z)$ belonging to the biggest time constant, $p_1 = e^{-T_s/T_1} = 0.9512$ is cancelled and instead an integrator is introduced (*PI*), and the pole belonging to the second biggest time constant $p_2 = -0.8825$ is also cancelled, and instead an ideal differentiating effect is introduced (*PD*). So the pulse transfer function of the regulator is

$$C_+(z) = k_c \frac{z - 0.9512}{z - 1} \frac{z - 0.8825}{z}.$$

The constant k_c in the regulator is designed to set the prescribed phase margin. In the first step, let its value be 1.

```
p1=exp(-Ts/T1)
p2=exp(-Ts/T2)
kc=1;
Cz=zpk([p1 p2],[0 1],kc,Ts)
```

The pulse transfer function of the open loop is

```
Lz=Cz*Gz
```

Executing the simplifications:

```
Lz=minreal(Lz)
```

The value of k_c can be read from the graphical surface of *ltiview* viewer. Its value will be the reciprocal of the gain belonging to the phase angle $-120°$.

```
ltiview('bode',Lz);
```

or k_c can be calculated directly with command margin,

```
[mag,phase,w]=bode(Lz);
kc=margin(mag,phase-60,w);
```

The gain is $k_c = 33.46$.
Calculate again the transfer functions:

```
Cz=kc*Cz;
Lz=kc*Lz;
```

Check the phase margin:

```
margin(Lz)
```

Calculate and plot the output and control signals of the discrete system.

```
Hz=Lz/(1+Lz)
Uz=Cz/(1+Lz)
subplot(211); step(Hz)
subplot(212); step(Uz)
```

Second Step: Calculation of the Pulse Transfer Function $C_{\mathrm{sm}}(z)$ of the SMITH Predictor

$$C_{\mathrm{sm}}(z) = \frac{C_+(z)}{1 + (1 - z^{-d})C_+(z)G_+(z)}$$

where $d = T_{\mathrm{d}}/T_{\mathrm{s}}$

```
z=tf('z',Ts);
d=Td/Ts
Csm=Cz/(1+(1-z^(-d))*Cz*Gz)
Csm=minreal(Csm,0.0001)
```

The behaviour of a hybrid (discrete–continuous) system can be analysed in the SIMULINK™ environment. Build the SIMULINK™ model (Fig. 12.13.). The model takes the parameters Csm, Ps from the MATLAB™ surface. The sampling time is set at the zero order hold element. The hold element can be left out of the circuit, as when connecting the discrete and continuous blocks SIMULINK™ automatically employes a zero order hold. Set the value of the dead-time. Execute the simulation till 30 s. The result of the simulation can be seen on the scopes, but the simulation data can be reached also in MATLAB™. Double clicking on the *Scope* blocks, choose the option *Parameters* (the second icon from the left). Then on the option *Data History* mark *Save data to workspace*, then set the *Variable name* to ty and tu, respectively on the scopes with format *Array*. After running the SIMULINK™ model, the signals can be plotted from MATLAB™ with the following commands:

```
subplot(211);plot(ty(:,1),ty(:,2));grid;
subplot(212);stairs(tu(:,1),tu(:,2));grid;
```

Determine the overshoot and the settling time of the output signal and check the maximum value of the control signal. The output and the control signals are shown in Fig. 12.14. The control is much faster than it would be with a series *PID* regulator designed for the process with dead-time.

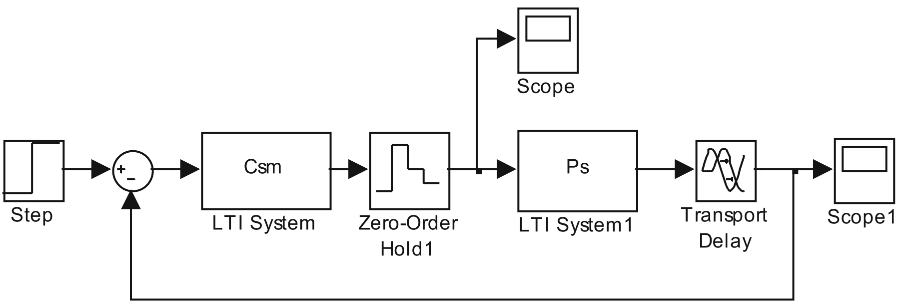


Fig. 12.13 SIMULINK™ diagram of SMITH predictor

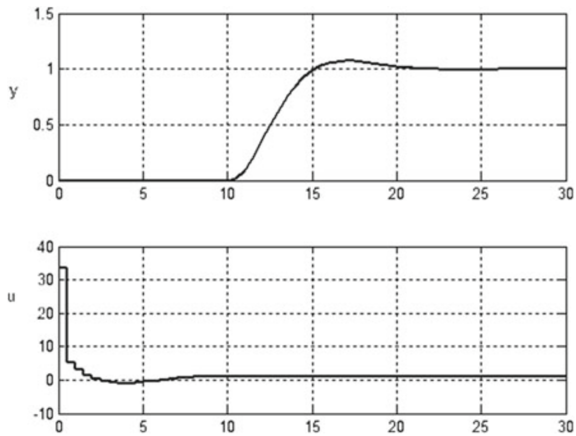


Fig. 12.14 The output and control signals in Smith predictor

Observe that the control signal is the same as the control signal obtained in the first step of the design for the case of the process without dead-time. The output signal is the same as the output signal of the control without the dead-time, but shifted by the dead-time.

Problem Design a Smith predictor regulator for the second order process with dead-time given in Example 12.1. For the process without dead-time design a *PID* regulator with phase margin of 60°.

Remark The Smith prediktor is a special case of the Youla parameterization. It does not apply filters.

Problem Determine the expression of the Youla parameter corresponding to the Smith predictor.

Problem With the SIMULINK™ model execute simulations for the case of a plant-model mismatch. Change the parameters of the continuous process (static gain, time constants, dead-time one by one and also together) by ±10%. The regulator was designed for the original, nominal process. Evaluate the simulation results.

12.3 Design of a Dead-Beat Regulator

The structure of the control system is shown in Fig. 12.15.

In sampled-data (discrete) control systems, the regulator is an intelligent device, typically it is a microprocessor based *Programmable Logic Regulator* (*PLC*) device. The task of the *PLC* is the realization of a control algorithm, and handling of signals related to the operation of the regulator (filtering, A/D and D/A converters, interfaces). With software realization, besides *PID* control there is the possibility of applying several special control algorithms. Such algorithms are the

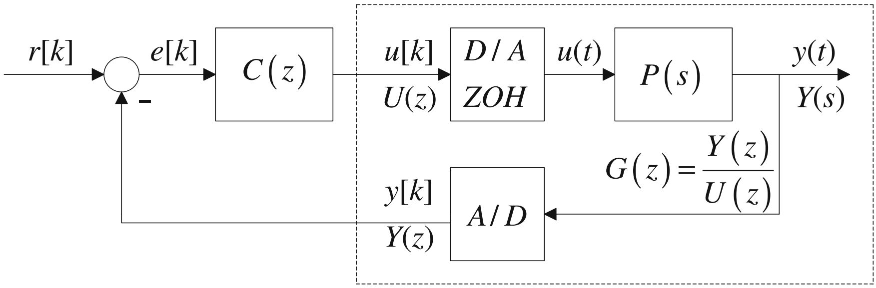


Fig. 12.15 Block diagram of a sampled control system

discrete *PID* regulator, the Youla parameterized regulator and the Smith predictor. Another discrete control algorithm is the dead beat regulator, which ensures the accurate settling of the output signal within a given finite number of the sampling periods. First analyse the method applying it to stable processes without dead-time ($T_d = 0$), and the reference signal is supposed to be a unit step. Let us remark that dead beat regulator can be designed also for cases when these conditions are not fulfilled.

In the sequel the solution is shown in three steps. In the first step the fastest control is designed, the requirement is to settle the process output to the desired value in one sampling step. It will be seen that settling in one step can be ensured, but the control signal values can be extremely high, furthermore, in most cases intersampling oscillations do appear. In the second step, the cancellable and non-cancellable zeros will be separated to avoid oscillations in the pulse transfer of the process. It will be seen that the oscillations can be avoided by increasing the prescribed settling time. If after that the design process still results in control signals that are too high, then in the third step, the design procedure can be refined by introducing a design polynomial, increasing the settling time, but still keeping it finite.

The design is executed in the z domain. An interesting feature of the method is that it eliminated some non-desired time domain properties of the system (oscillations, high overexcitation) taking into consideration properties in the z domain. The basic task is the design of the discrete regulator.

First the hybrid (continuous-discrete) problem is converted to a pure discrete problem by determining the pulse transfer function $G(z)$, which is the discrete equivalent of the connected D/A converter and holding element and continuous process given by its transfer function $P(s)$. The sampling time T_s also has to be given. Then the pulse transfer function $C(z)$ of the regulator is designed and the behaviour of the closed loop control system is analysed.

The main point of the design is that we prescribe the behaviour of the closed loop giving its closed loop pulse transfer function $T(z)$. In the case of dead beat control the resulting transfer function ensures the settling after a unit step reference signal during a given number of sampling steps, i.e. $T(z) = z^{-d}$, where d is the discrete dead-time, namely $d = \text{entier}(T_\text{d}/T_\text{s}) + 1$.

$$\frac{C(z)\,G(z)}{1 + C(z)\,G(z)} = T(z)$$

Hence $C(z)$, the pulse transfer function of the regulator, is

$$C(z) = \frac{T(z)}{G(z)\,[1 - T(z)]}.$$

Example 12.4 Consider the continuous process given by the transfer function

$$P(s) = \frac{1}{(1 + 5s)\,(1 + 10s)}$$

The sampling time is $T_\text{s} = 1$ s.

First define the s and z variables.

```
Ts=1;
s=zpk('s');
z= zpk ('z',Ts);
Ps=1/((1+5*s)*(1+10*s))
```

To have an idea what the requirement of settling in one sampling period means draw the step response of the process.

```
step(Ps);
```

Calculate the pulse transfer function of the process:

```
Gz=c2d(Ps,Ts)
```

The obtained pulse transfer function is

$$G(z) = \frac{\mathcal{B}(z)}{\mathcal{A}(z)} = 0.0090559\,\frac{(z + 0.9048)}{(z - 0.8187)\,(z - 0.9048)}$$

The discrete poles are transformed from the continuous poles according to the relationship $z_i = \text{e}^{p_i T_\text{s}} = \text{e}^{-T_\text{s}/T_i}$. A discrete zero also appeared.

First design a one step dead-beat regulator. The resulting transfer function between the output signal and the reference signal should be

$$T(z) = z^{-1}.$$

The regulator is obtained as

$$C(z) = \frac{1}{G(z)\,(z-1)}.$$

```
Tz=1/z
Cz=Tz/(Gz*(1-Tz))
Cz=minreal(Cz)
```

The pulse transfer function of the regulator is

$$C(z) = 110.425\,\frac{(z-0.8187)\,(z-0.9048)}{(z+0.9048)\,(z-1)}.$$

Analyse the behaviour of the control system in discrete time. The step response really shows a one step delay.

```
step(Cz*Gz/(1+Cz*Gz))
```

If we want to get an accurate picture of the behaviour of the system, the behaviour of the continuous process should be analysed considering also the output signal values between the sampling points. The simulation is not so straightforward in the MATLAB™ environment, but it is easy with SIMULINK™. Start SIMULINK™ and build the model shown in Fig. 12.16.

```
simulink
```

Create a new model file (with extension *.mdl*) and copy the individual blocks from the block libraries. Change the values of the parameters to the required values. From the menu set the *Simulation –> Parameters –> Stop* time parameter to 8. SIMULINK™ uses the parameters defined in MATLAB™.

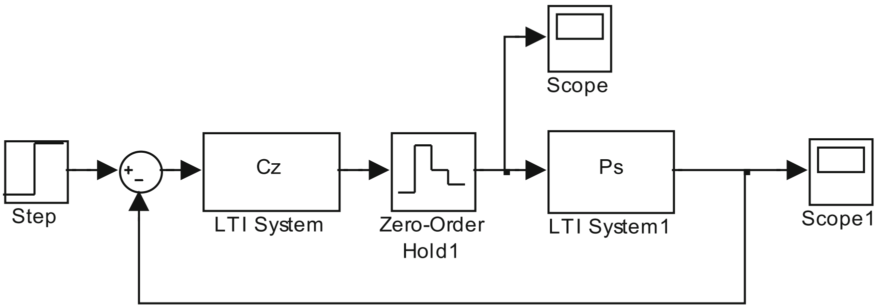


Fig. 12.16 SIMULINK™ diagram of a sampled control system

The individual blocks are taken from the following block libraries:

$C(z)$, $P(s)$:	*Control System Toolbox –> LTI system*
Zero order hold:	*Simulink –> Discrete –> Zero-Order-Hold*
Subtraction:	*Simulink –> Math –> Sum*
Step input:	*Simulink –> Sources –> Step*
Scope:	*Simulink –> Sinks –> Scope*.

The zero order hold element can be left out of the block diagram. The results can be seen on the *Scope* blocks.

The results can be transformed to MATLAB™ by double clicking on the *Scope* blocks, choosing the option *Parameters*, and then indicating on *Data History* the option *Save data to workspace*. Set *Variable name* ty and tu, respectively, with *Array* format.

After running the SIMULINK™ model, the signals can be plotted from the MATLAB™ environment by the following commands

```
subplot(211);plot(ty(:,1), ty(:,2));grid;
subplot(212);stairs(tu(:,1), tu(:,2));grid;
```

The simulation result shown in Fig. 12.17 demonstrates that there are oscillations between the sampling points. The reason for the oscillations is that the regulator compensated the numerator $\mathcal{B}(z)$ of the process, and therefore it has a pole which causes oscillations in the signal $u[k]$. If the pulse transfer of the process is

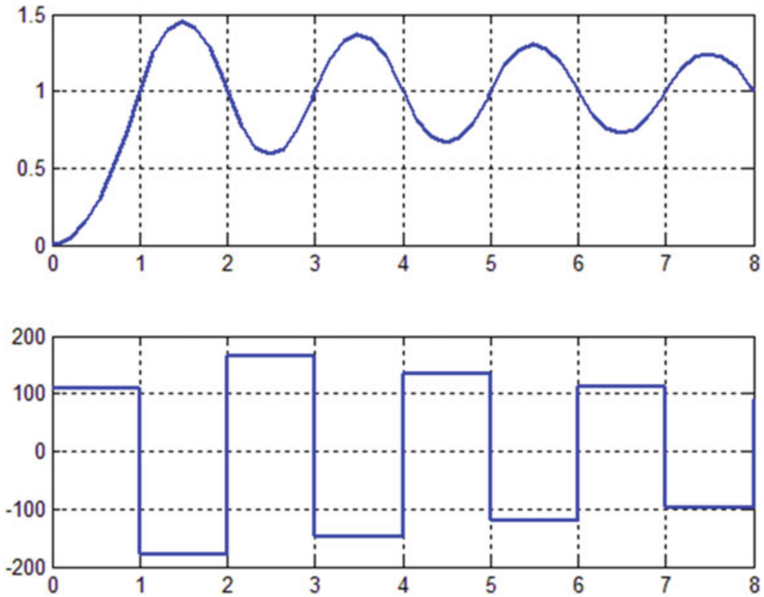


Fig. 12.17 Output and control signals in dead-beat control

written in the form $G(z) = \mathcal{B}(z)/\mathcal{A}(z)$, then the pulse transfer function of the regulator is $C(z) = \frac{\mathcal{A}(z)}{\mathcal{B}(z)(z-1)}$, and the loop transfer function is the pulse transfer function of an integrator.

$$L(z) = C(z)G(z) = \frac{\mathcal{A}(z)}{\mathcal{B}(z)(z-1)} \cdot \frac{\mathcal{B}(z)}{\mathcal{A}(z)} = \frac{1}{z-1}$$

It can be seen that the zeros of the process appear in the regulator as poles. The pulse transfer function $G(z)$ contains one zero, $z_1 = -0.9048$, which appears in the regulator as a pole. Analyse this in more detail.

```
C1z=1/(z+0.9048)
step(C1z)
```

It is seen that this component causes the oscillations. In the continuous domain as $z_1 = \mathrm{e}^{-sT_s}$, $s_1 = -\ln(\mathrm{z})/T_s$

```
s1=log(-0.9048)
  s1 = -0.1000 + 3.1416i
```

This corresponds to complex conjugate pole pairs with a small damping factor, the oscillation frequency is $\pi/T_s = 3.14$, just the twice the sampling frequency. Obviously this is not a real zero, it appeared due to sampling, therefore it does not have to be compensated. The Appendix demonstrates how the complex conjugate pole pairs are transformed to the z domain.

(As a demonstration let us analyse the step responses of the following pulse transfer functions.

```
P1z=0.5/(z-0.5); step(P1z);
P2z=1.5/(z+0.5); step(P2z);
P3z=2.5/(z+1.5); step(P3z);
```

It can be seen that the output of the first system shows an aperiodic settling process corresponding to the response of a first order lag element. The output of the second system presents decreasing oscillations, while the third system is unstable. The constants were chosen to ensure unit static gain. The regulator should not cancel the "bad" zeros of the process, i.e. those which are outside of the unit circle or lie in the unfavourable area of the unit circle. The unfavourable area, as will be seen in the Appendix, is the area outside of a "heart shape curve", which includes also the negative real zeros inside the unit circle.)

Let us separate the numerator of the pulse transfer function of the process into cancellable and non-cancellable components.

$$\mathcal{B}(z) = \mathcal{B}_+(z)\,\mathcal{B}_-(z)$$

$\mathcal{B}_+(z)$ contains the cancellable roots and $\mathcal{B}_-(z)$ those which are not cancellable. If a zero in $\mathcal{B}(z)$ is not cancelled, than it will appear in the resulting pulse transfer of the control system between the output signal and the reference signal. A requirement is to track the step reference signal without steady error. Therefore the static gain of $\mathcal{B}_-(z)$ should be 1, i.e. $\mathcal{B}_-(z)|_{z=1} = 1$.

In the next step a regulator is designed which does not compensate the zero which would cause the oscillation. In the design, the dead-time d of the process is also considered. In the sequel, the pulse transfer functions are given with the z^{-1} shift operator.

The required pulse transfer function is given in the following form:

$$T\left(z^{-1}\right) = \mathcal{B}_-\left(z^{-1}\right) z^{-d}$$

The pulse transfer function of the process is

$$G\left(z^{-1}\right) = \frac{\mathcal{B}_+(z^{-1})\mathcal{B}_-(z^{-1})}{\mathcal{A}(z^{-1})} z^{-d}.$$

The resulting transfer function between the output and the reference signal is

$$\frac{C(z^{-1})G(z^{-1})}{1 + C(z^{-1})G(z^{-1})} = T\left(z^{-1}\right) = \mathcal{B}_-\left(z^{-1}\right) \cdot z^{-d}$$

Hence the pulse transfer function of the regulator:

$$C\left(z^{-1}\right) = \frac{\mathcal{B}_-(z^{-1}) \cdot z^{-d}}{G(z^{-1})[1 - \mathcal{B}_-(z^{-1}) \cdot z^{-d}]} = \frac{T(z^{-1})}{G(z^{-1})[1 - T(z^{-1})]},$$

or, considering the decomposition of $G(z)$,

$$C\left(z^{-1}\right) = \frac{\mathcal{A}(z^{-1})}{\mathcal{B}_+(z^{-1})[1 - \mathcal{B}_-(z^{-1}) \cdot z^{-d}]}$$

Apply the regulator design for our example avoiding intersampling oscillations.

The separation of the numerator is

$\mathcal{B}_-(z^{-1}) = (1 + 0.9048z^{-1})/(1 + 0.9048)$; $B_+(z^{-1}) = 0.0090559 \cdot (1 + 0.9048)$ and $d = 1$.

This is calculated by

```
Bm =1+0.9048*z^(-1)
Bmn=Bm/dcgain(Bm)
Bpn=Gz.k*dcgain(Bm)
Tz=Bmn/z
```

```
Cz=Tz/(Gz*(1-Tz))
Cz=minreal(Cz,0.001)
```

The regulator pulse transfer function is then

```
57.972(z-0.9048)(z-0.8187)
--------------------------
      (z+0.475)(z-1)
```

Simulate again the behaviour with the SIMULINK™ model. The output and the control signal are shown in Fig. 12.18.

It can be seen that there are no oscillations and at the same time the control signal became more moderate, with smaller maximum value. The control system became slower, now the output signal reaches the steady value in two sampling periods. As the overexcitation is still about 50, this value may exceed the possibilities of the applied actuator. Therefore a method has to be sought to decrease further the overexcitation while keeping the finite settling time.

Supplement the control algorithm with a *design filter polynomial*, which calmly "guides" the finite time settling process. For example, choosing the design polynomial

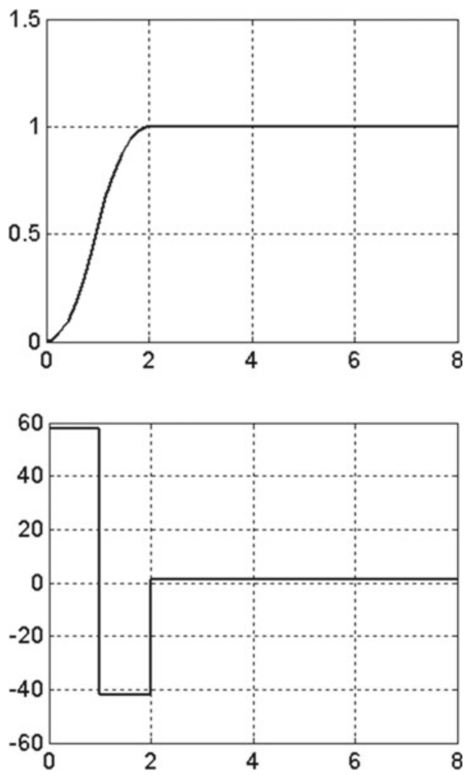


Fig. 12.18 Output and control signals in dead-beat control avoiding inter-sampling oscillations

$$\mathcal{F}(z) = \frac{1}{3} + \frac{z^{-1}}{3} + \frac{z^{-2}}{3}$$

its smoothing effect can be observed looking at its step response.

```
Fz=(1+z^(-1)+z^(-2))/3
step(Fz)
```

With the design polynomial the design criterion is

$$T\left(z^{-1}\right) = \mathcal{F}(z) \cdot \mathcal{B}_{-}\left(z^{-1}\right) \cdot z^{-d}$$

For zero static error, the static gain of the design polynomial should be 1, i.e. $\mathcal{F}(z = 1) = 1$.

The regulator is

$$C\left(z^{-1}\right) = \frac{\mathcal{F}(z^{-1})\mathcal{B}_{-}(z^{-1}) \cdot z^{-d}}{G(z^{-1})[1 - \mathcal{F}(z^{-1})\mathcal{B}_{-}(z^{-1}) \cdot z^{-d}]} = \frac{\mathcal{A}(z^{-1})\mathcal{F}(z^{-1})}{\mathcal{B}_{+}(z^{-1})[1 - \mathcal{F}(z^{-1})\mathcal{B}_{-}(z^{-1}) \cdot z^{-d}]}$$

```
Tz=Fz*Bmn/z
Cz=Tz/(Gz*(1-Tz))
Cz=minreal(Cz,0.001)
```

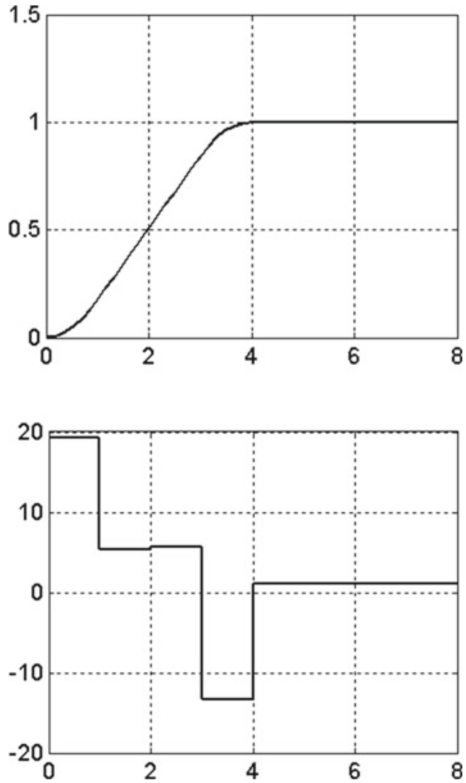


Fig. 12.19 With a design filter the settling process is modified

Running the SIMULINK™ diagram it can be seen that the settling time has been increased and the maximum value of the control signal has become lower (Fig. 12.19).

Add a dead-time $T_d = 1$ to the process and simulate the behaviour of the control system

```
Td=1
d=Td/Ts+1
Tz=Bmn/(z^d)
Cz=Tz/(Gz*(1-Tz))
Cz=minreal(Cz,0.001)
```

Put a *delay* block (Simulink –> Continuous –> Transport Delay) in the SIMULINK™ model and set its parameter to Td. Similarly to the previous discussion with a different $\mathcal{F}(z)$ design polynomial the control system could fulfill more flexible specifications.

Appendix

Let us assume that a continuous system can be characterized by a dominant pole pair, so it can be considered as a second order oscillating system. For a given damping factor ξ for different time constants the poles are located on straight lines which start from the origin of the complex plain and close angle $\cos\varphi = \xi$ with the negative real axis. With sampling these straight lines are mapped into "heart shape curves" on the complex plain. These mapping is illustrated by the following program. The location of the poles ensures acceptable transients if the damping factor is above a given value (e.g. $\xi \geq 0.6$).

The conjugate complex poles with a given damping factor in the continuous domain. In the s domain with constant damping factor ξ we get $s = \sigma + j\omega$ straight lines going through the origin, as σ takes different values. For a given value of σ, $\omega = \frac{\sigma}{\xi}\sqrt{1-\xi^2}$. In the discrete domain according to mapping $z = e^{sT_s}$ 'heart shape curves' are obtained. To demonstrate this, write the following MATLAB code:

```
szigma=0:0.01:1.6;
kszi=0.4;
Ts=1;
z=exp(Ts*(-szigma+j*sqrt(1-kszi*kszi)*szigma/kszi));
plot(real(z),imag(z),real(z),-imag(z)),grid;
```

Those roots of $\mathcal{B}(z)$ which lie inside the closed curve (where the damping is bigger than on the contour) are the roots of $\mathcal{B}_+(z)$, while those roots which lie on the curve or outside of it are the roots of $\mathcal{B}_-(z)$.