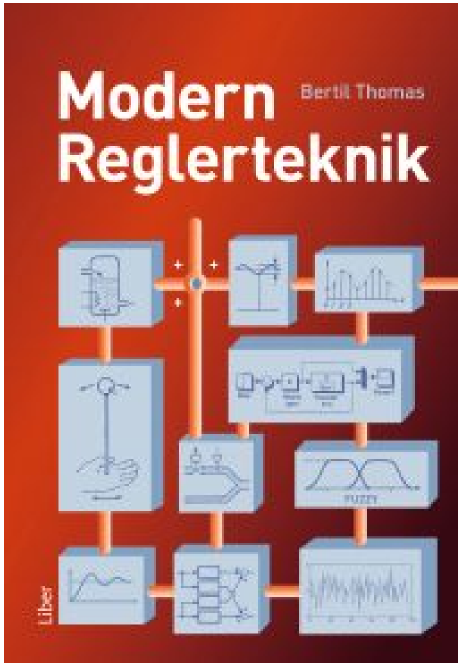
Reglerteknik

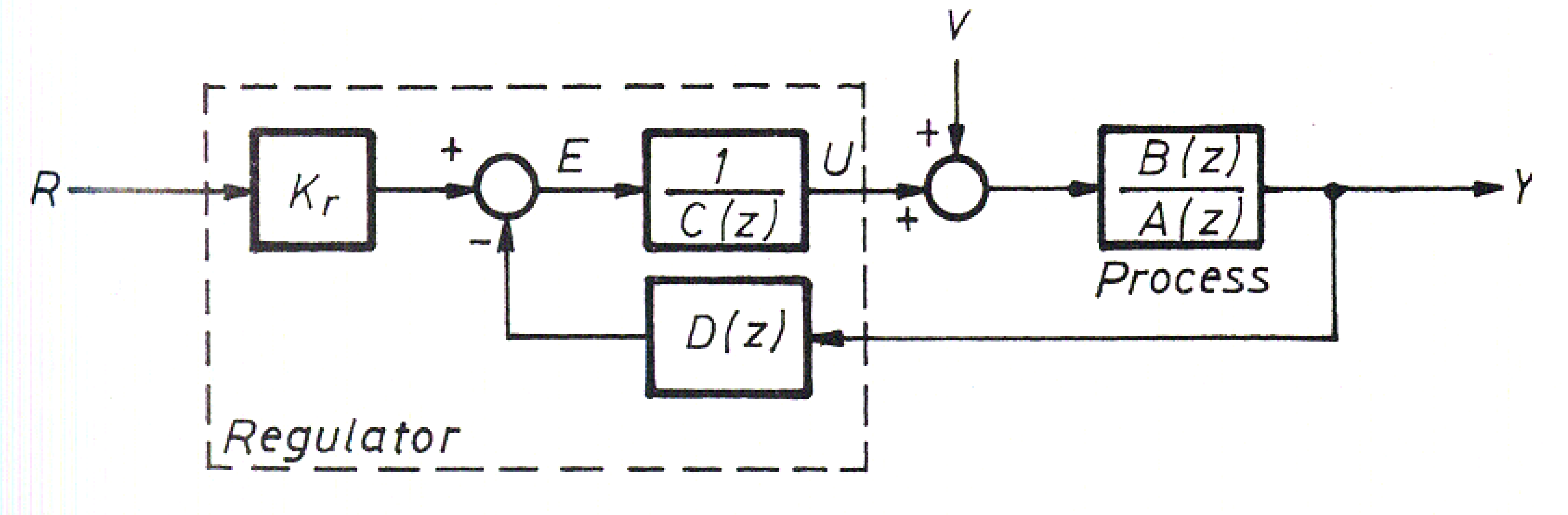
Polplaceringsregulator



Köp bok och övningshäfte på kårbokhandeln

William Sandqvist william@kth.se

Polplaceringsregulator



$$H_{tot} = \frac{Y(z)}{R(z)} = \frac{K_r \cdot B(z)}{A(z)C(z) + B(z)D(z)}$$

$$P(z) = A(z)C(z) + B(z)D(z)$$

Polplaceringsekvation

$$A(z)C(z)+B(z)D(z)=P(z)$$

Välj gradtal: $n_p = n_a + n_b - 1 \quad [gradtal]$

$$n_c = n_b - 1 \quad n_d = n_a - 1$$

Ansätt: $C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{nc} z^{-n}$

$$D(z) = d_0 + d_1 z^{-1} + d_2 z^{-2} + \ldots + d_{nd} z^{-n}$$

$$K_r = \frac{P(z=1)}{B(z=1)}$$

Bokens polplaceringsexempel

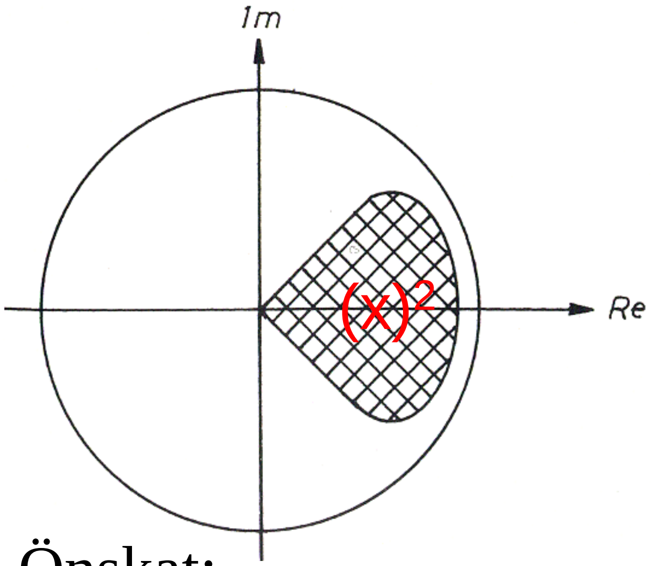
$$y(k) = 0{,}82\,y_{k-1} + 0{,}5u_{k-1} + 0{,}5u_{k-2} \quad \{Z:\}$$

$$Y = 0{,}82Yz^{-1} + 0{,}5Uz^{-1} + 0{,}5Uz^{-2}$$

$$H(z) = \frac{U}{Y} = \frac{0{,}5z^{-1} + 0{,}5z^{-2}}{1 - 0{,}82z^{-1}}$$

$$B(z) = 0{,}5z^{-1} + 0{,}5z^{-2}$$

$$A(z) = 1 - 0{,}82z^{-1}$$



Önskat:
dubbelpol i $z = 0{,}5$

Polplaceringsexempel

$$B(z) = 0{,}5z^{-1} + 0{,}5z^{-2}$$

$$A(z) = 1 - 0{,}82z^{-1}$$

Önskat:
dubbelpol i $z = 0{,}5$

$$P(z) = (1 - 0{,}5z^{-1})^2 = 1 - z^{-1} + 0{,}5z^{-2} \quad \Rightarrow \quad n_p = 2$$

$$n_c = n_b - 1 = 2 - 1 = 1 \quad \Rightarrow \quad C(z) = 1 + c_1 z^{-1}$$

$$n_d = n_a - 1 = 1 - 1 = 0 \quad \Rightarrow \quad D(z) = d_0$$

$$AC + BD = P:$$

$$(1 - 0{,}82z^{-1})(1 + c_1 z^{-1}) + (0{,}5z^{-1} + 0{,}5z^{-2})(d_0) = 1 - z^{-1} + 0{,}25z^{-2}$$

Polplaceringsexempel

$$(1-0{,}82z^{-1})(1+c_1z^{-1})+(0{,}5z^{-1}+0{,}5z^{-2})(d_0)=$$

$$=1+c_1z^{-1}-0{,}82z^{-1}-0{,}82c_1z^{-2}+0{,}5d_0z^{-1}+0{,}5d_0z^{-2}=$$

$$=1+\boxed{(c_1-0{,}82+0{,}85d_0)}\,z^{-1}+\boxed{(0{,}5d_0-0{,}82)}\,z^{-2}$$

$$=1\boxed{-}z^{-1}+\boxed{0{,}25}\,z^{-2}$$

AC+BD = P

$$c_1-0{,}82+0{,}5d_0=-1 \quad\Leftrightarrow\quad c_1+0{,}5d_0=-0{,}18$$

$$0{,}5d_0-0{,}82c_1=0{,}25 \quad\Leftrightarrow\quad -0{,}82c_1+0{,}5d_0=0{,}25$$

Polplaceringsexempel

$$c_1 - 0{,}82 + 0{,}5d_0 = -1 \quad \Leftrightarrow \quad c_1 + 0{,}5d_0 = -0{,}18$$

$$0{,}5d_0 - 0{,}82c_1 = 0{,}25 \quad \Leftrightarrow \quad -0{,}82c_1 + 0{,}5d_0 = 0{,}25$$

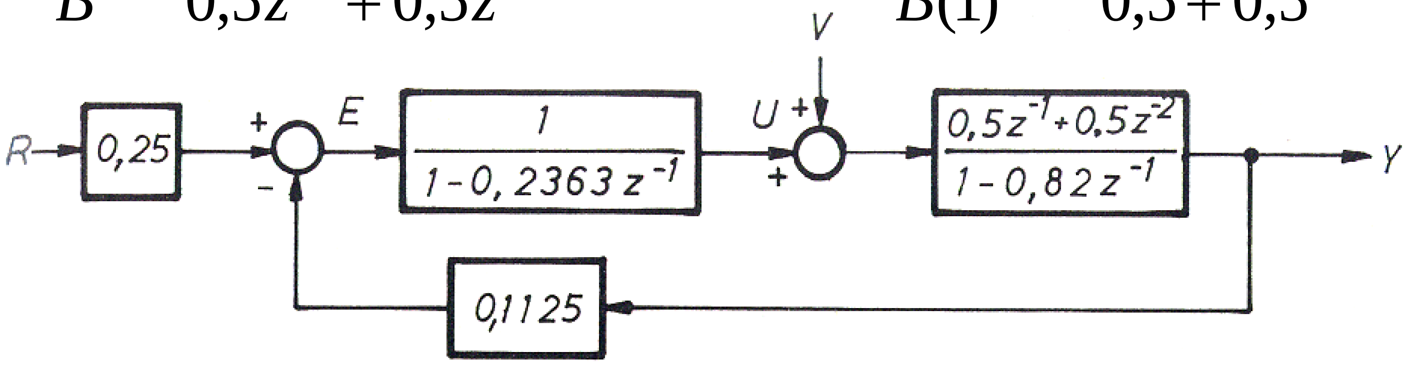
$$\begin{pmatrix} 1 & 0{,}5 \\ -0{,}82 & 0{,}5 \end{pmatrix} \bullet \begin{pmatrix} c_1 \\ d_0 \end{pmatrix} = \begin{pmatrix} -0{,}18 \\ 0{,}25 \end{pmatrix}$$

$$c_1 = -0{,}236 \quad d_0 = 0{,}113$$

```
» X=[1, 0.5; -0.82, 0.5]\[-0.18, 0.25]'
X = -0.2363   c1
     0.1125   d0
»
```

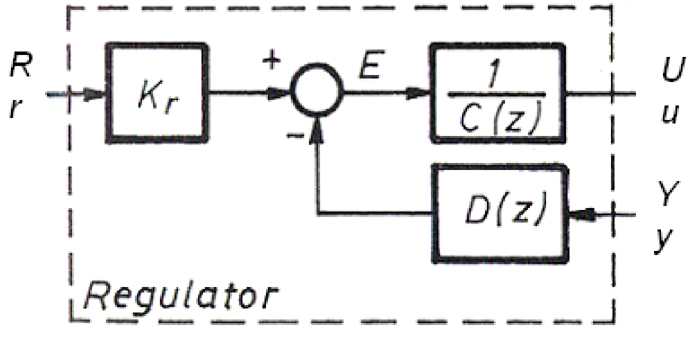
Polplaceringsexempel

$$\frac{P}{B} = \frac{1 - z^{-1} + 0{,}25z^{-2}}{0{,}5z^{-1} + 0{,}5z^{-2}} \quad \Rightarrow \quad K_r = \frac{P(1)}{B(1)} = \frac{1 - 1 + 0{,}25}{0{,}5 + 0{,}5} = 0{,}25$$



$$E = R \cdot K_r - Y \cdot D$$

$$U = \frac{E}{C}$$

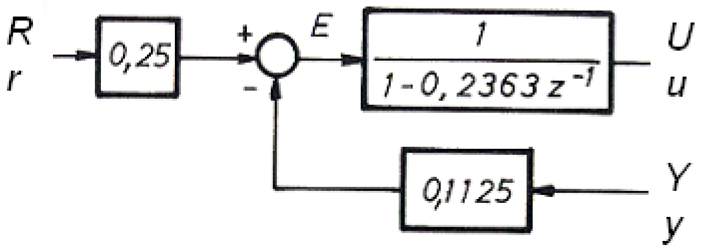


William Sandqvist william@kth.se

Polplaceringsexempel

$$E = R \cdot K_r - Y \cdot D$$

$$U = \frac{E}{C}$$



$$E = 0{,}25\,R - 0{,}1125\,Y$$

$$U = \frac{E}{1 - 0{,}2363 z^{-1}} \quad \Rightarrow \quad U(1 - 0{,}2363 z^{-1}) = E$$

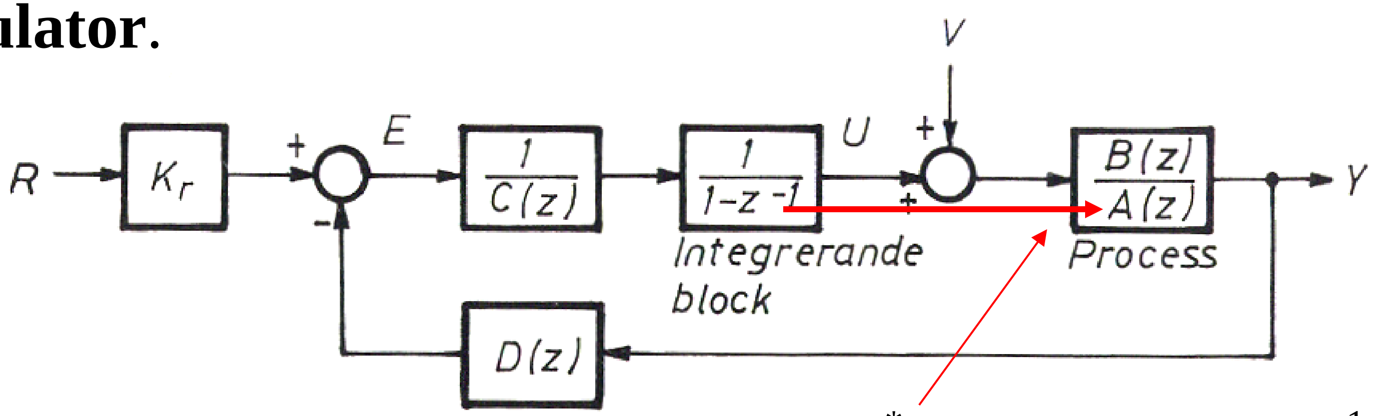
$$\Rightarrow \quad U = E - 0{,}2363 U\, z^{-1}$$

- Differensekvationer:

$$e_k = 0{,}25 r_k - 0{,}1125 y_k \qquad u_k = e_k - 0{,}2363 u_{k-1}$$

Integrerande block

För att undvika kvarvarande fel behöver man **integrerande regulator**.



$$A^{*}(z) = A(z)\cdot(1-z^{-1})$$

Beräkningsmässigt hänför man integreringen till $A(z)$.

(= enklast så)

med integrerande block

Önskat $P(z)$ dubbelpol z = 0,5 och *extra* pol placerad i origo.

$$B(z) = 0{,}5z^{-1} + 0{,}5z^{-2}$$

$$A(z) = 1 - 0{,}82z^{-1} \Rightarrow A^*(z) = (1 - 0{,}82z^{-1})(1 - z^{-1})$$

$$P(z) = (1 - 0{,}5z^{-1})^2 \cdot (1 - 0 \cdot z^{-1}) = 1 - z^{-1} + 0{,}5z^{-2} \quad \Rightarrow \quad n_p = 2$$

$$n_c = n_b - 1 = 2 - 1 = 1 \quad \Rightarrow \quad C(z) = 1 + c_1 z^{-1}$$

$$n_d = n_a - 1 = 2 - 1 = 1 \quad \Rightarrow \quad D(z) = d_0 + d_1 z^{-1}$$

$$A^* C + BD = P:$$

$$(1 - 0{,}82z^{-1})(1 - z^{-1})(1 + c_1 z^{-1}) + (0{,}5z^{-1} + 0{,}5z^{-2})(d_0) = 1 - z^{-1} + 0{,}25z^{-2}$$

med integrerande block

$$(1-0,82z^{-1})(1-z^{-1})(1+c_1 z^{-1})+(0,5z^{-1}+0,5z^{-2})(d_0)=$$

$$=1+(c_1-1,82+0,5d_0)\,z^{-1}+(0,82-1,82c_1+0,5d_1+0,5d_0)\,z^{-2}+$$

$$+(0,82c_1+0,5d_1)z^{-3}= \quad 1-z^{-1}+0,25\,z^{-2}$$

$$(c_1-1,82+0,5d_0)=-1$$

$$(0,82-1,82c_1+0,5d_1+0,5d_0)=0,25$$

$$(0,82c_1+0,5d_1)=0$$

$$c_1+0,5d_0+0d_1=0,82$$

$$-1,82c_1+0,5d_0+0,5d_1=-0,57$$

$$0,82c_1+0d_0+0,5d_1=0$$

```
» X=[1,0.5,0;-1.82,0.5,0.5;0.82,0,0.5]\[0.82,-0.57,0]'
X = 0.382
     0.876
    -0.626
»
```

c_1

d_0

d_1

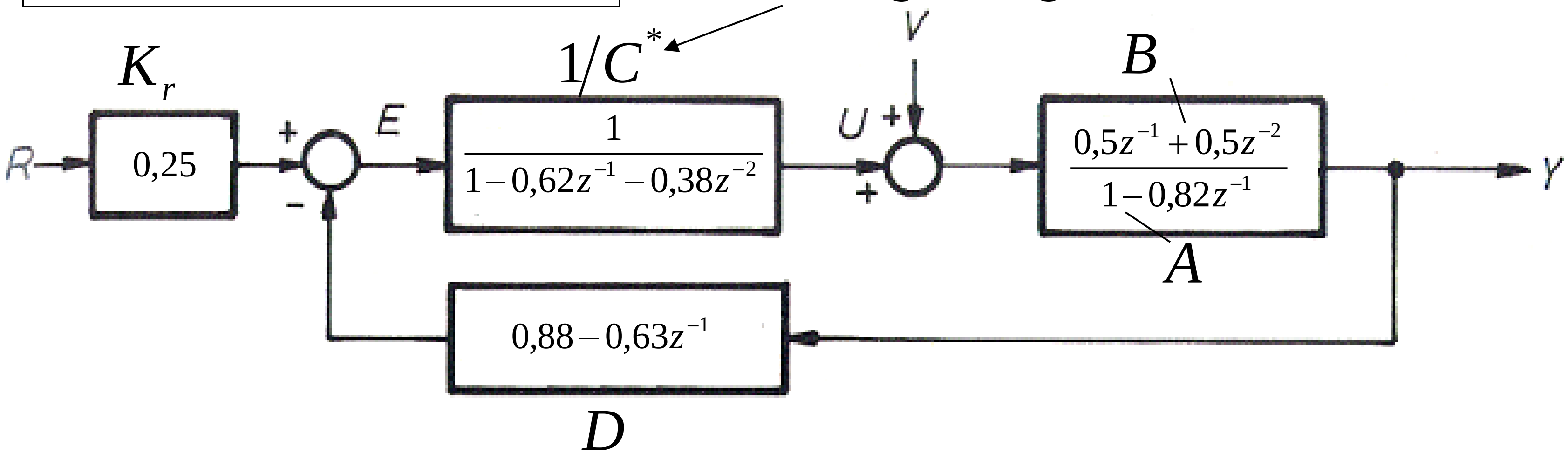
med integrerande block

$$\frac{P}{B} = \frac{1 - z^{-1} + 0{,}25z^{-2}}{0{,}5z^{-1} + 0{,}5z^{-2}} \quad \Rightarrow \quad K_r = \frac{P(1)}{B(1)} = \frac{1 - 1 + 0{,}25}{0{,}5 + 0{,}5} = 0{,}25$$

$$\boxed{C^* = C(z) \cdot (1 - z^{-1})} = (1 - z^{-1})(1 + 0{,}382z^{-1}) = \boxed{1 - 0{,}618z^{-1} - 0{,}382z^{-2}}$$

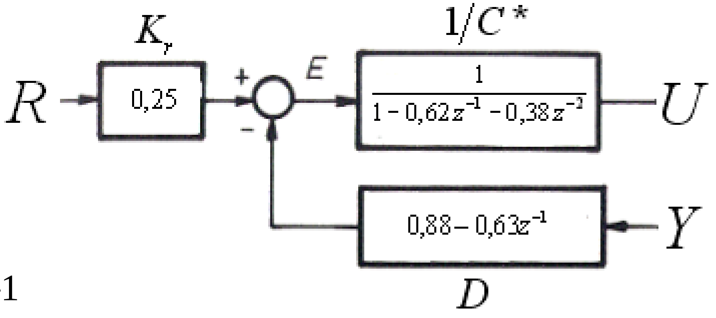
$$\boxed{D(z) = 0{,}876 - 0{,}626z^{-1}}$$

Nu integrering i C-blocket!



Regulatorns differensekvationer

$$\boxed{\begin{aligned} E &= R \cdot K_r - Y \cdot D \\ U &= \frac{E}{C^*} \end{aligned}}$$



$$E = 0,25R - 0,88Y + 0,63Y\,z^{-1}$$

$$U = \frac{E}{1 - 0,62\,z^{-1} - 0,38\,z^{-2}} \quad \Leftrightarrow \quad U - 0,62U\,z^{-1} - 0,38U\,z^{-2} = E$$

$$U = E + 0,62U\,z^{-1} + 0,38U\,z^{-2}$$

- Differensekvationer:

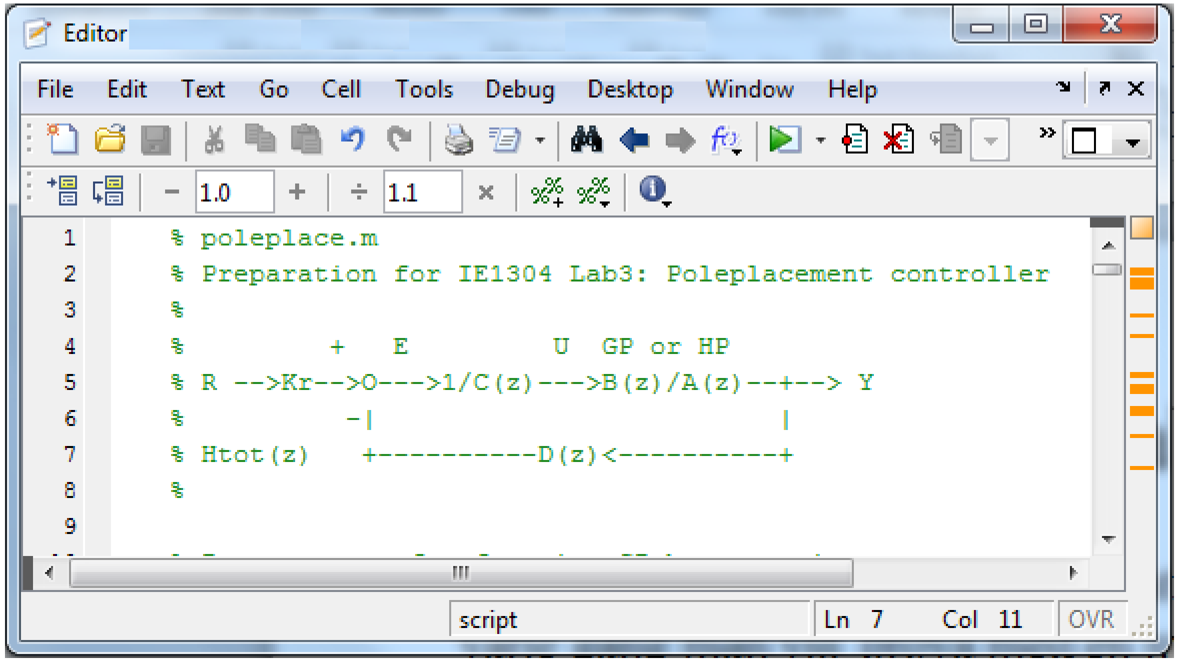
$$\boxed{e_k = 0,25r_k - 0,88\,y_k + 0,63\,y_{k-1} \qquad u_k = e_k + 0,62u_{k-1} + 0,38u_{k-2}}$$

Matlabscript till laborationen

Polplaceringsmetoden kräver omfattande beräkningar varje gång man vill prova med en ny polplacering. Detta är något man ***inte hinner med*** att göra vid en laboration – man måste automatisera beräkningarna med hjälp av ett Matlab-script.

När Du gör labförberedelserna ska Du spara beräkning-arna i form av ett Matlabscript för att snabbt kunna göra om dom vid laborationen med de aktuella och verkliga värdena.

Matlabscript till laborationen



Man börjar scriptet direkt med en kommentar (%). Den kommentaren skrivs ut om man skriver **`help poleplace`** i Matlabs kommandofönster – bra ”komihåghjälp” om vad scriptet gör för något.

Matlabscript till laborationen

Inför laborationen antar vi att processens förstärkning, och två tidkonstanter, har följande preliminära värden:

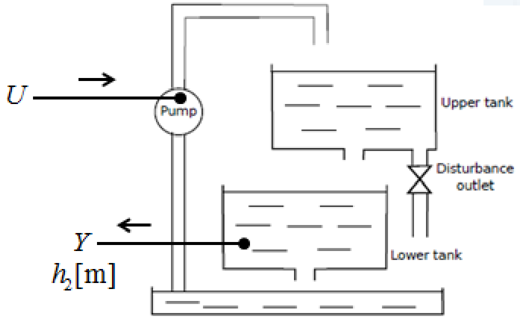


Figure 1: The controlled process

```
% Process transfer function GP has two time constants

Kp =3   % Change process gain if needed
T1=6    % Change first time constant if needed
T2=21   % Change second time constant if needed

GP = series(tf(Kp,[T1 1]),tf(1,[T2 1]))
```

Efter gjorda mätningar under laborationen vet Du mer …

Matlabscript till laborationen

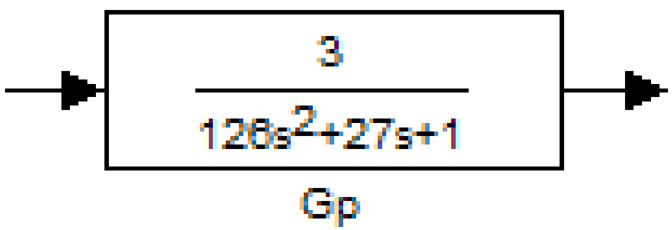
För att göra scriptet mer läsbart brukar man kopiera resultatet från kommandofönstret och klistra in dem som kommentarer i scriptet!

```
GP = series(tf(Kp,[T1 1]),tf(1,[T2 1]))

% Transfer function:
%          3
% ------------------
% 126 s^2 + 27 s + 1
```

Simuleringsmodell i Simulink

Simulink kan simulera tidskontinuerliga och tidsdiskreta system tillsammans. För processen är det lämpligt att använda den tidskontinuerliga modellen.

```
% Transfer function:
%         3
% ------------------
% 126 s^2 + 27 s + 1
```



Matlabscript till laborationen

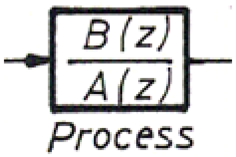
Till polplaceringsberäkningarna kommer vi behöva processens tidsdiskreta modell. Matlab kan skriva ut modellen med positiv (z^n) *eller* negativ (z^{-n}) notation.

```
% Discretize the process
disp('Now discretizing')
h = 0.6 % Change if needed

HP=c2d(GP,h);
fprintf('\nHP Negative notation')
HP.variable = 'z^-1'
fprintf('\nHP Positive notation')
HP.variable = 'z'
```

```
% Discrete transfer function:
% HP Negative notation
% Transfer function:
% 0.004107 z^-1 + 0.003935 z^-2
% -----------------------------
% 1 - 1.877 z^-1 + 0.8794 z^-2
% Sampling time (seconds): 0.6
%
% HP Positive notation
% Transfer function:
% 0.004107 z + 0.003935
% ----------------------
% z^2 - 1.877 z + 0.8794
% Sampling time (seconds): 0.6
```

Matlabscript till laborationen

Vi behöver processens överförings-funktion på negativ form uppdelad i nämnare (A) och täljare (B).




```
% HP Negative notation
% Transfer function:
% 0.004107 z^-1 + 0.003935 z^-2
% ------------------------------
% 1 - 1.877 z^-1 + 0.8794 z^-2
```

```
% HP=B/A
[B,A]=tfdata(HP,'v')

% B = [ 0 0.0041 0.0039]
% A = [ 1.0000 -1.8767  0.8794]
```

Matlabscript till laborationen

Processens A och B har gradtalet två och inför laborationen antar vi följande *preliminära* poler till polplaceringspolynomet:
$z = 0{,}7 \pm 0{,}2\mathrm{i}$ och $z = 0$.

```
% 3 poles suggestion:
p1=0.7+0.2i;  % Change if needed
p2=0.7-0.2i;  % Change if needed
p3=0;

% Prompt user for pole values
% p1 = input('First pole (a+bi): ');
% p2 = input('Second pole (a-bi): ');
% display('Third pole at origo 0+0i');

% Poleplacement polynomial
P = poly([p1,p2,p3])

% P = [1.0000 -1.4000 0.5300 0]
```

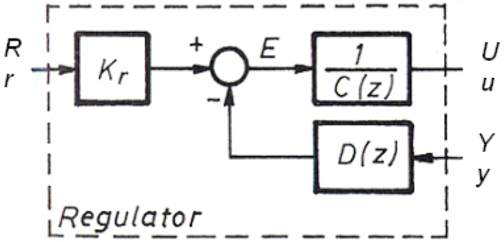
$$\boxed{P(z) =}\; A(z)C(z) + B(z)D(z)$$

Ska man pröva många värden på polerna kan det vara bekvämt om scriptet ”promptar” användaren!

Matlabscript till laborationen

Ansats av regulatorpolynomen C och D:

```
% Regulator polynomes C and D
% C = 1 + c_1*z^-1
% D = d_0 + d_1*z^-1
```

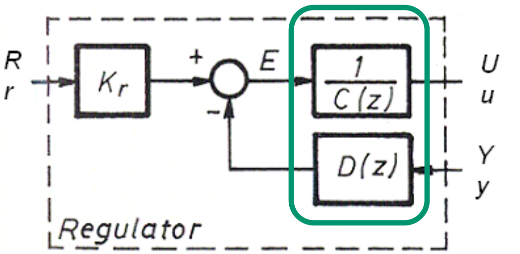


$$A(z)C(z)+B(z)D(z)=P(z)$$

Parametrarna **`c_1 d_0`** och **`d_1`** beräknas efter identifiering mot regulatorpolynomets motsvarande potenstermer. Algebran kan *inte* utföras med Matlab (om modulen Maple saknas) utan får göras med ”papper och penna” eller med Mathematica. När algebran är gjord kan Matlab lösa ekvationssystemet numeriskt. ***Du måste redovisa de algebraiska beräkningarna vid laborationen!***

Matlabscript till laborationen

Redovisa dina algebraiska beräkningar som leder till detta ekvationssystem vid laborationen!

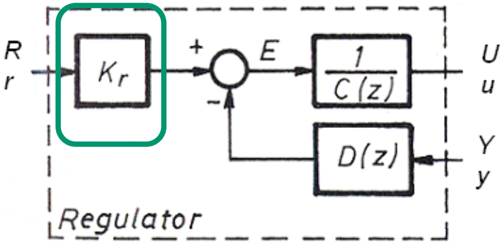



```
[ 1    B(2) 0
  A(2) B(3) B(2)
  A(3) 0    B(3) ]
              \
[P(2)-A(2), P(3)-A(3), P(4)]' ;
c_1=ans(1),d_0=ans(2),d_1=ans(3)

%  c_1 = 0.2148
%  d_0 = 63.7665
%  d_1 = -48.0050
```

Nu är parametrarna **c_1 d_0 d_1** beräknade!

Matlabscript till laborationen

$$K_r = \frac{P(1)}{B(1)}$$

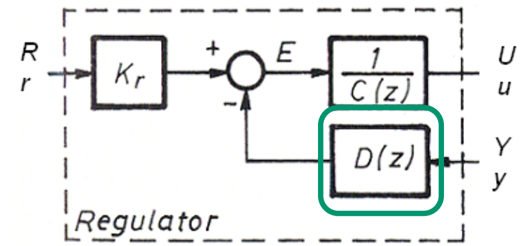



```
% Kr=P(1)/B(1)

Kr = polyval(P,1) / polyval(B,1)

% Kr=16.1664
```

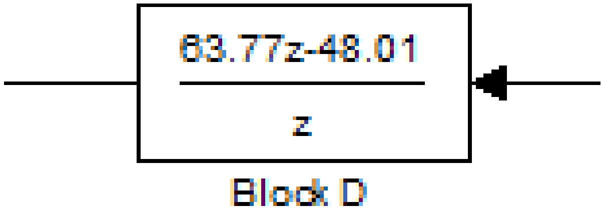
Nu är ***alla*** parametrar beräknade!

Simuleringsmodell i Simulink

```
% Controller blocks

D = [d_0  d_1 ]
D_block = tf( D,[1 0],h);
fprintf('\nBlock D positive notation')
D_block.variable = 'z'
```

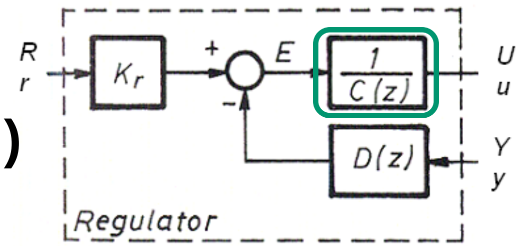



```
% D =  63.7665  -48.0050
%
% Block D positive notation
% Transfer function:
% 63.77 z - 48.01
% ---------------
%         z
% Sampling time (seconds): 0.6
```

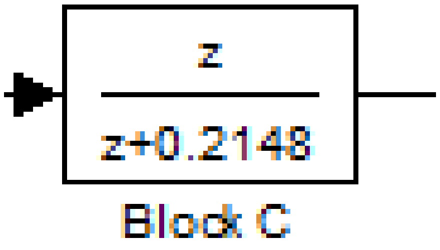


Simuleringsmodell i Simulink

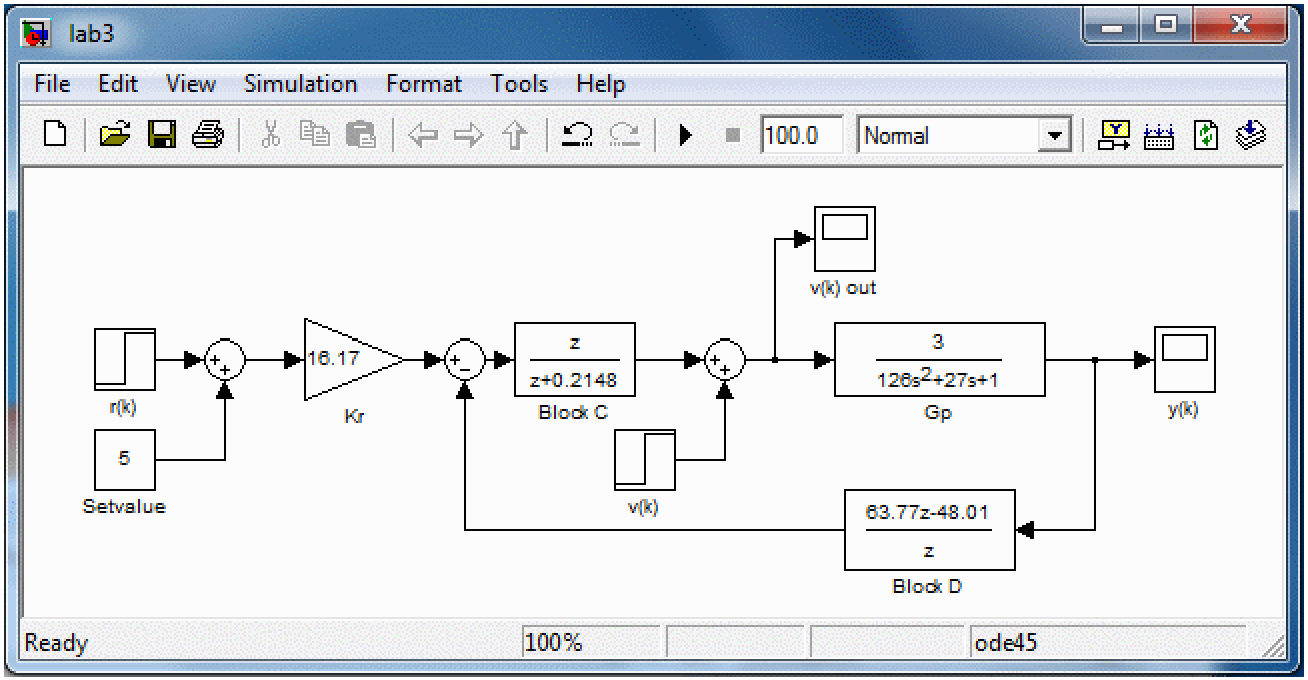
```
C = [1  c_1]
C_block = tf([1 0],C,h );
fprintf('\nBlock C positive notation')
C_block.variable = 'z'
```




```
% C =  1.0000    0.2148
%
% Block C positive notation
% Transfer function:
%    z
% ----------
% z + 0.2148
% Sampling time (seconds): 0.6
```

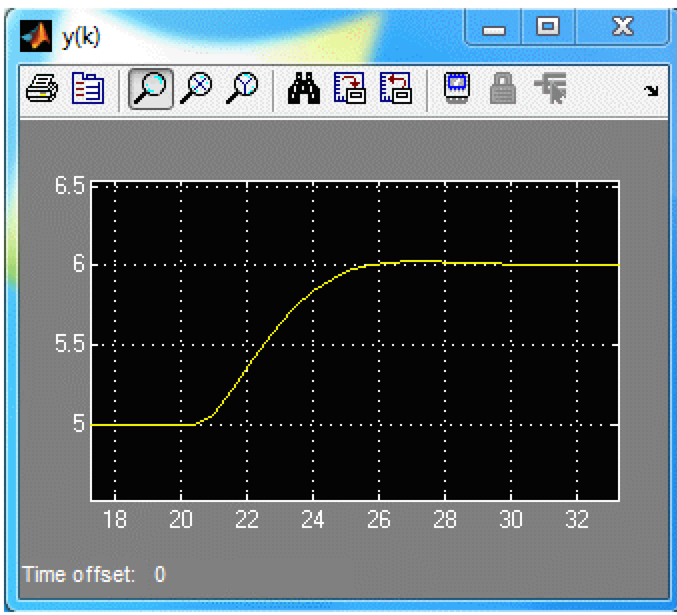


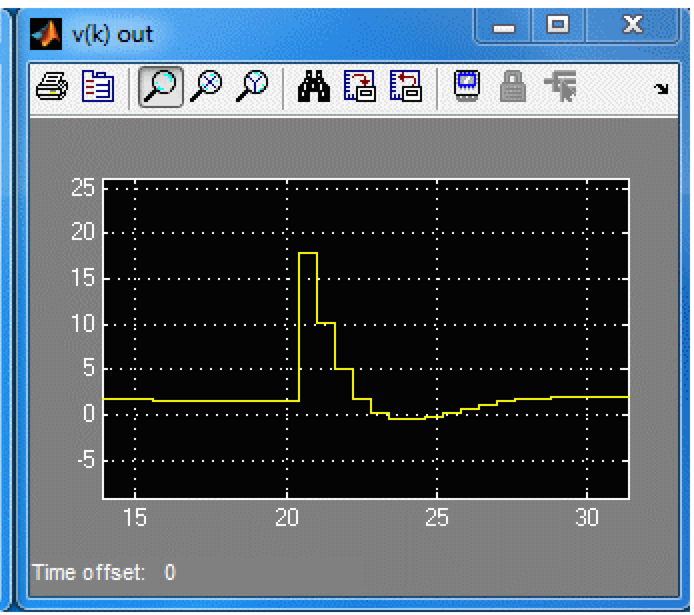
Simuleringsmodell i Simulink



Simuleringsmodell i Simulink

Simulerat stegsvar när börvärdet förändras från 5 till 6 efter 20 sek. Processen utstorhet *y*(*k*) är kontinuerlig, medan regulatorns *v*(*k*) är diskret.

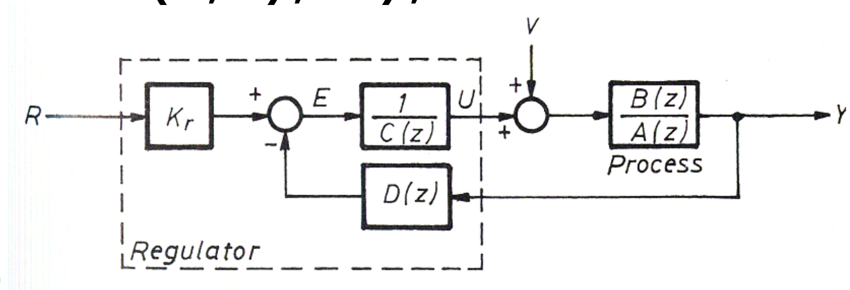




Matlabscript till laborationen

$$H_{tot} = \frac{K_r \cdot B(z)}{A(z)C(z) + B(z)D(z)}$$

```
% Total transfer function
%             Kr*B
% Htot = ---------
%           A*C + B*D
% Transfer function of controlled system:

fprintf('\nTotal system, positive notation')
Htot = tf( Kr*B, conv(A,C) + conv(B,D),h );
Htot.variable = 'z’


% Transfer function:
%         0.06639 z + 0.06361
% ------------------------------------
% z^3 - 1.4 z^2 + 0.53 z + 2.776e-017
```



William Sandqvist william@kth.se

Lämpliga kommandon i Matlab

Efter det att man kört sitt script en gång finns det totala systemets överföringsfunktion kvar i i Matlabs minne. Man kan nu undersöka systemet med fler Matlab-kommandon:

```
step(GP)     % step response of process model

step(Htot)   % step response of controlled process

margin(Htot) % Gain and phase margins of controlled system

[Y,T]=step(Htot); stepinfo(Y,T,'SettlingTimeThreshold', 0,05)
```

Är scriptet korrekt?

Hur kontrollerar man att scriptet är rätt tänkt och att det ger riktiga svar?

```
[P,Z]=pzmap(Htot)

% P =   0
%       0.7000+ 0.2000i
%       0.7000- 0.2000i
% Z = -0.9580
```

Polerna blev placerade exakt där de skulle – därför är scriptet korrekt!

William Sandqvist william@kth.se

Matlabscript för C-kod

Matlab kan skriva **text och siffervärden** till en fil!
Till exempel kan polplaceringsregulatorn skrivas ut som en C-funktion, som man sedan kan ta med när man kompilerar ett C-projekt.

```
fid=fopen('ppc.c','w');
fprintf(fid,'float PolePlaceController(float r, float y)\n');
fprintf(fid,'{\n');
fprintf(fid,'   static float pre_u = 0;\n');
fprintf(fid,'   static float pre_y = 0;\n');
fprintf(fid,'   float e;\n');
fprintf(fid,'   float u;\n\n');
fprintf(fid,'   e = %6.3f*r - %6.3f*y - %6.3f*pre_y;\n',Kr,d_0,d_1);
fprintf(fid,'   u = e -  %6.3f*pre_u;\n\n',c_1);
fprintf(fid,'   return u;\n}\n');
fclose(fid);
```

C-funktion (ingår ej i kursen)