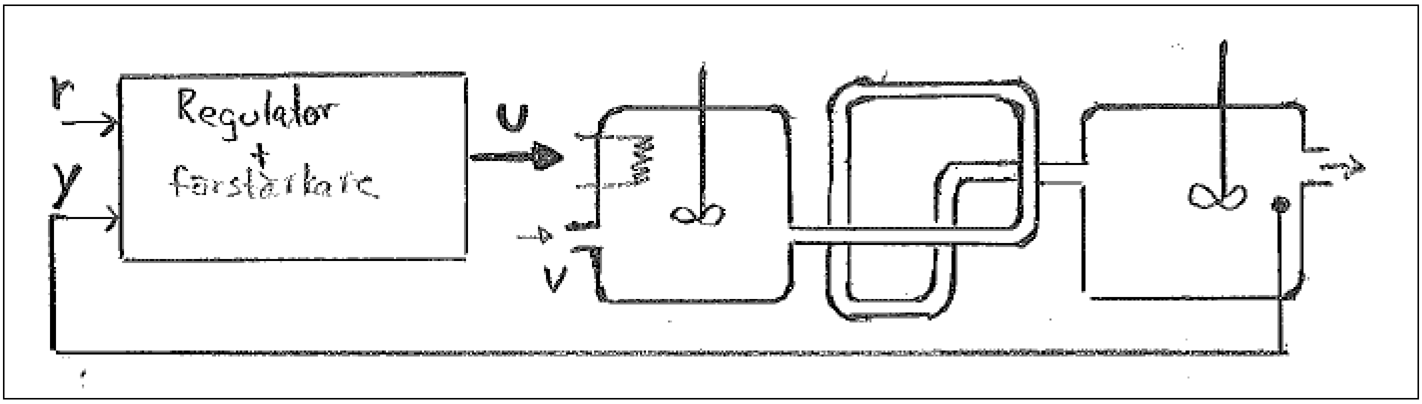
Inledning

Ett vätskeflöde Q pumpas genom två tankar. I vardera tanken förutsätts perfekt omrörning, så att temperaturen i tankarna är konstant över hela volymen V_1 respektive V_2. I tank 1 finns en uppvärmningsanordning som ger en effekt av styrsignalen $u(t)$. De två tankarna samt rörledningen (volym V_r) dem emellan antas vara perfekt isolerade, så att inget värmeutbyte sker genom väggarna. Den huvudsakliga störningen i processen härrör från variationer i inloppstemperaturen $v(t)$.

Det givna problemet är att dimensionera en regulator till den beskrivna dubbeltankprocessen. Målet är att kunna styra temperaturen $y(t)$ på dubbeltankens utflöde, med snabb respons och utan kvarstående fel då inloppstemperaturen varierar. För att kvarstående fel vid processtörning ska elimineras krävs att regulatorn har integrerande verkan. Först gör vi en modellering av processen för att sedan dimensionera och simulera en PID-regulator samt en polplaceringsregulator med hjälp av MATLAB/Simulink. Slutligen diskuterar vi val av regulator



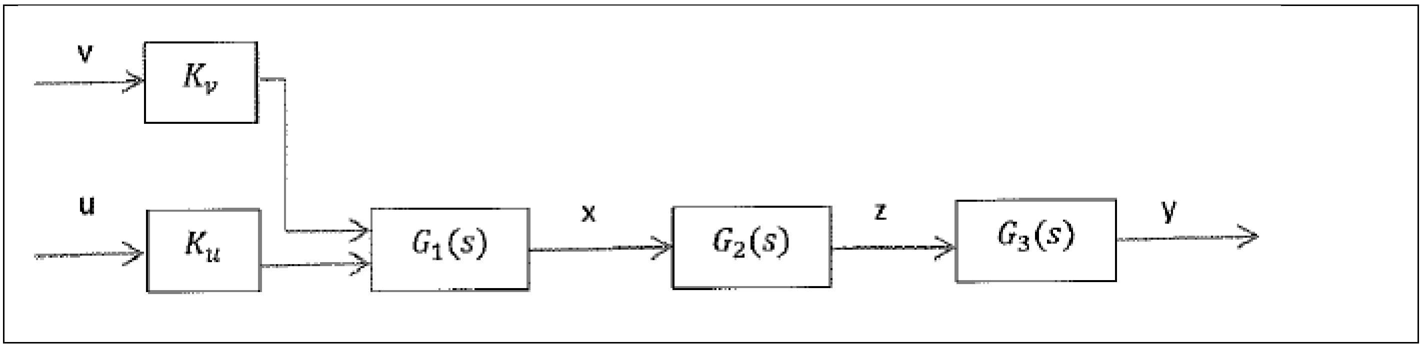
Figur 1 *Dubbeltankprocess med instrumentering*

Modellering av processen

Processen kan sägas bestå av tre delprocesser, som var och en kan beskrivas med överföringsfunktionerna:

$$G_1(s)\text{: Tank 1;}\quad G_2(s)\text{: Rörledningen;}\quad G_3(s)\text{: Tank 2}$$

Ett blockschema som beskriver processen ser då ut så här:



Figur 2 *Blockschema:Modell*

Utsignalen y bestäms alltså av följande samband:

$$Y(s) = U(s) \cdot K_u G_p(s) + V(s) \cdot K_v G_p(s), \qquad G_p(s) = G_1(s) \cdot G_2(s) \cdot G_3(s)$$

Där vi efter inledningens specifikationer beräknar:

$$G_1(s) = \frac{1}{\frac{V_1}{Q}s+1}, \qquad G_2(s) = e^{-\frac{V_r}{Q}s}, \qquad G_3(s) = \frac{1}{\frac{V_2}{Q}s+1}, \qquad K_v = 1, \qquad K_u = \frac{1}{Qc\varrho}$$

Enligt enhetstabellen:

$U(s)$	Styreffekt [W]
$V(s)$	Inloppstemperatur [K]
$Y(s)$	Utloppstemperatur [K]
V_1	Volym tank 1 [m^3]
V_r	Volym rörledningen [m^3]
V_2	Volym tank 2 [m^3]
Q	Flödet [m^3/s]
c	Värmekapacitivitet [J/K·kg]
ϱ	Densiteten [kg/m^3]

***Tabell 1** Enheter*

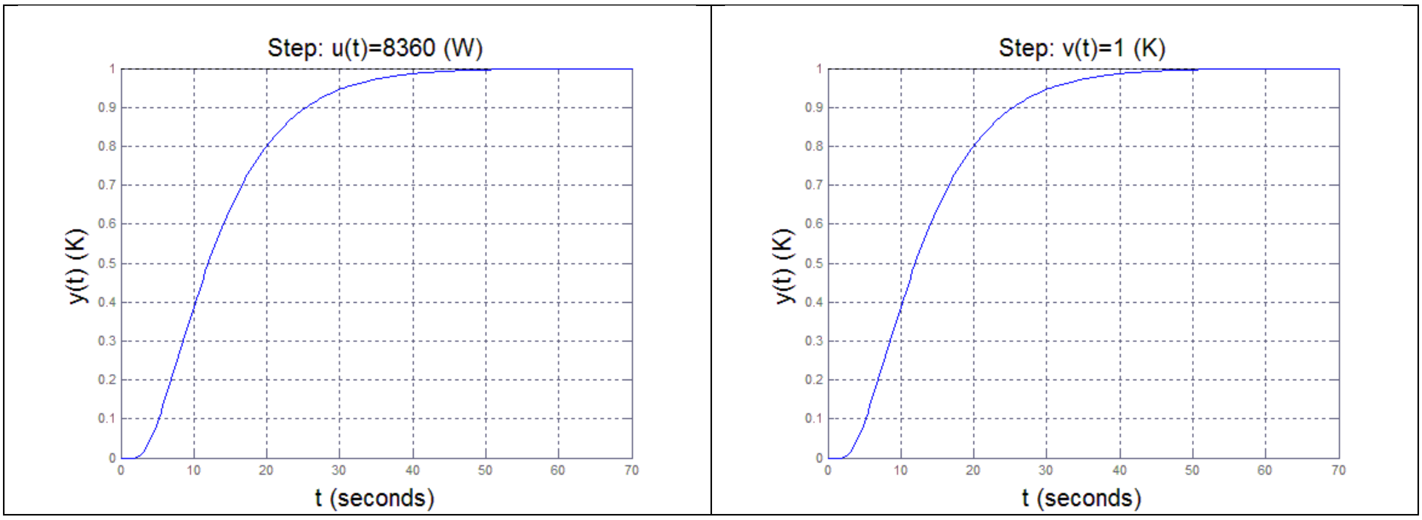
Värden på konstanterna för givna volymer och vätskeflöde av vatten:

V_1	V_r	V_2	Q	c	ϱ
$10 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$14 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	4180	1000

***Tabell 2** Givna värden*

Inledande simulering i MATLAB

Först undersöker vi hur mycket styreffekten måste höjas för att öka temperaturen i tank 2 med 1 °C. Vi uppmäter $u(t) = \frac{1}{K_u} = 8360$ [W]. Sedan undersöker vi hur en ökning på 1 °C inloppstemperaturen (processtörning) påverkar temperaturen i tank 2. Vi uppmäter $y(\infty) = 1$ [K].



***Figur 3** Enhetsstegsvar börvärde och processtörning*

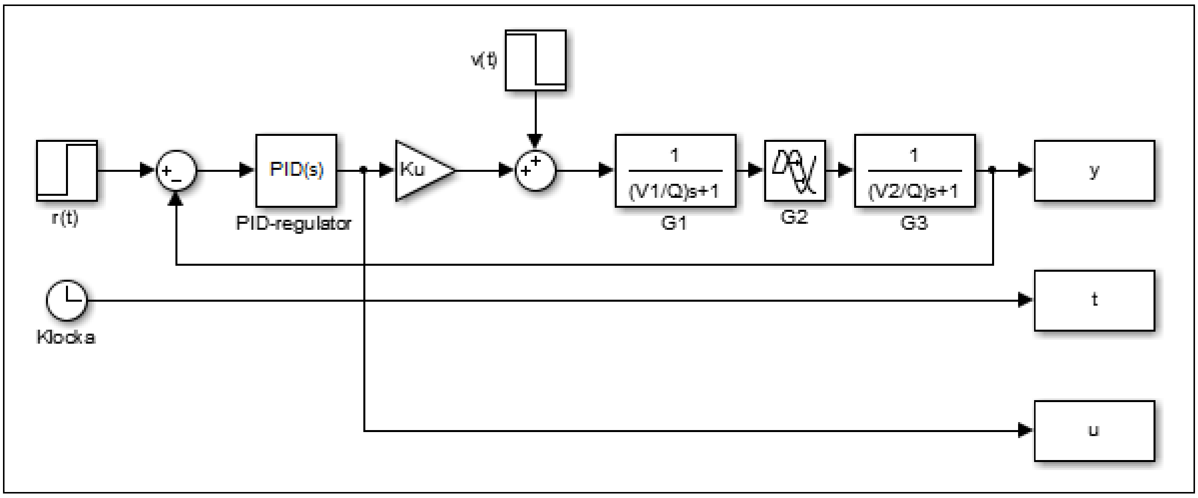
Resultaten anses rimliga och vi noterar att vi utan regulator får ett kvarstående fel vid processtörning.

Dimensioner av PID-regulator

Vi använder först Ziegler-Nichols (Z-N) svängningsmetod för att ställa in regulatorn och analyserar $K_u G_p(s)$ med MATLAB-kommandot "margin" och utläser $K_0 = 10^{\frac{95}{20}}$ och $T_0 = \frac{2\pi}{0.402}$. För PID-regulatorn $G_{PID}(s) = P + I\frac{1}{s} + D\frac{N}{1+N\frac{1}{s}}$ blir då:

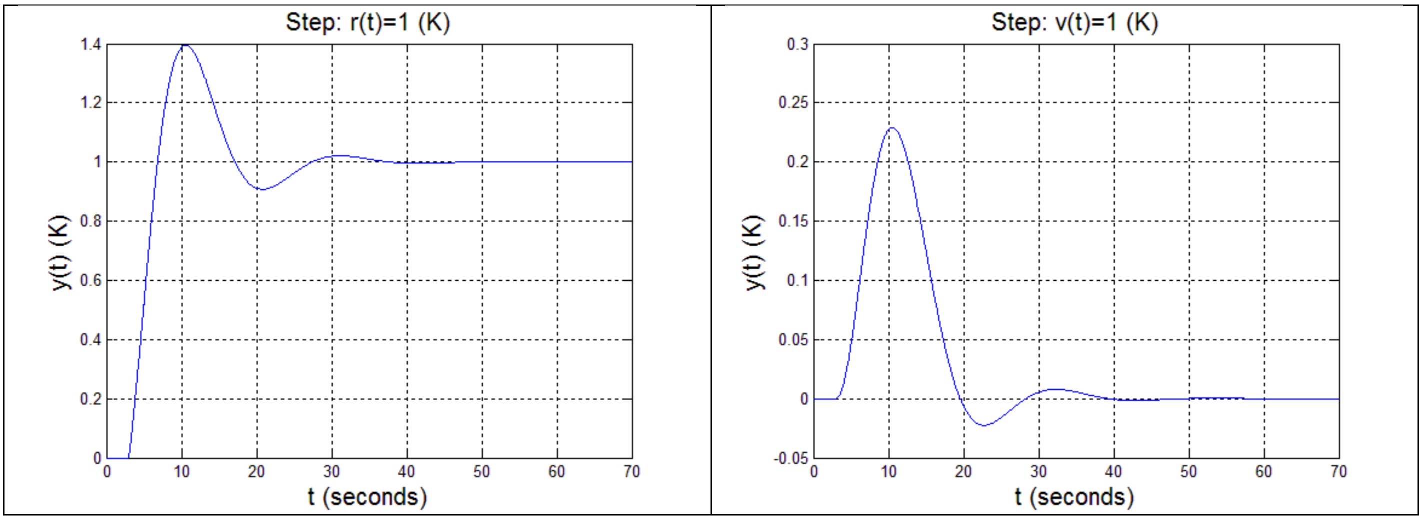
$$P = 0.6 \cdot K_0, \qquad I = \frac{P}{0.5 \cdot T_0}, \qquad D = P \cdot 0.125 \cdot T_0, \qquad N \to \infty$$

Med ovanstående värden (undantaget $N = 100$) simulerar vi vårt återkopplade system i Simulink med både börvärdesändring och processtörning enligt:



Figur 4 *PID: Simulink*

Vi får stegsvaren:



Figur 5 *PID Z-N: Stegsvar börvärde och processtörning*

För att finskjustera PID-regulatorn vid börvärdesändring använder vi nu oss av ”PID Tuner” i Simulink. Erhållen prestanda jämfört med Z-N:

	Finskjusterad	Ziegler-Nichols
Stigtid	3.87 s	2.96 s
Insvängningstid	12.4 s	31.1 s
Översläng	5 %	39.6 %
Topp	1.05	1.4
Amplitudmarginal	10 dB	8.79 dB
Fasmarginal	60°	34.3 °

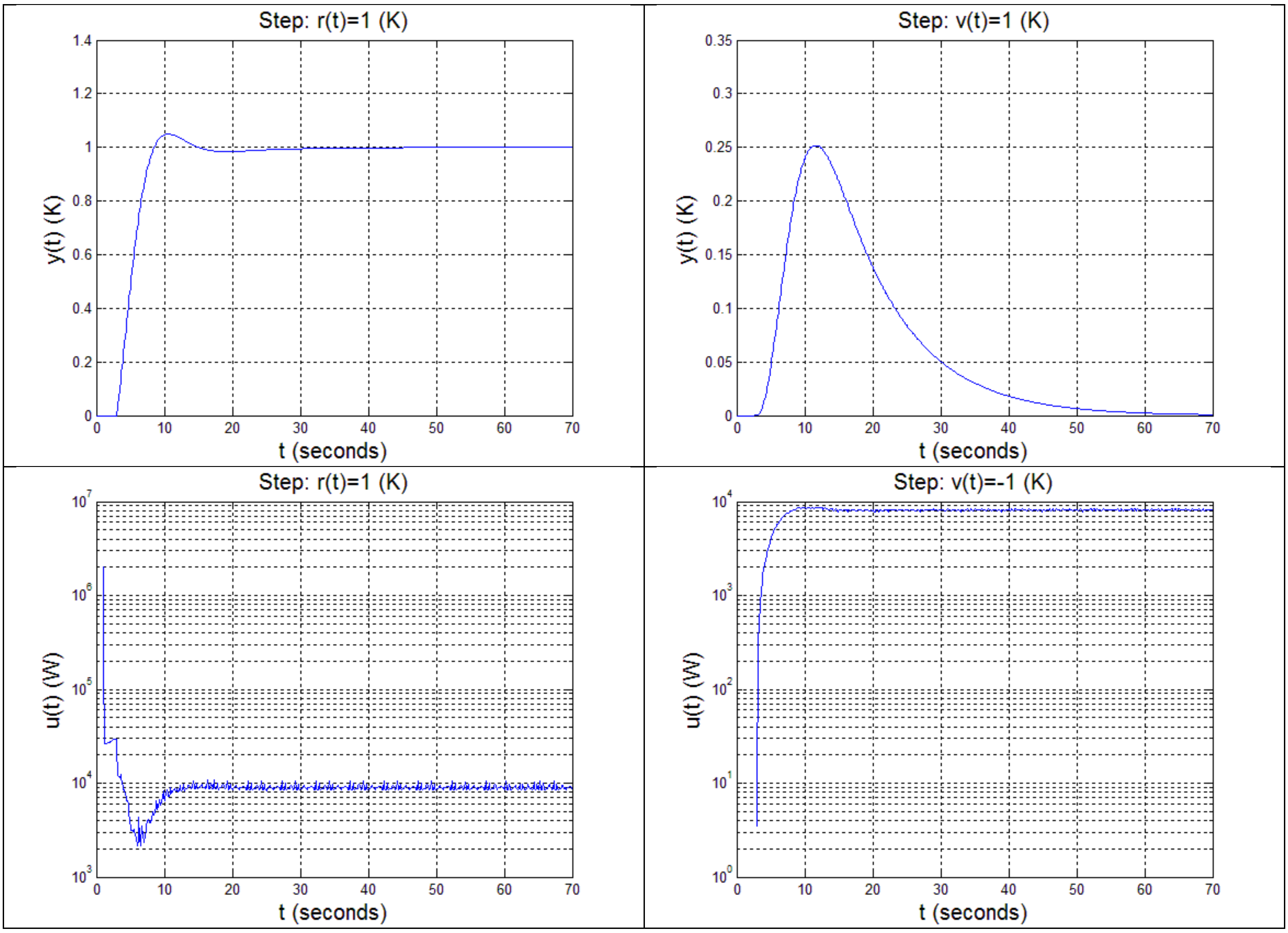
***Tabell 3** PID Tuner: Prestanda*

Tilldelade regulatorparametrar av Simulink för $G_{PID}(s) = P + I\frac{1}{s} + D\frac{N}{1+N\frac{1}{s}}$ blir då:

P	I	D	N
25194.1165104731	1956.14979824376	68803.105267141	28.6213074510803

***Tabell 4** PID finskjusterad: Regulatorparametrar*

För den finskjusterade regulatorn får vi stegsvaren, vi tittar då även på styreffekten vid stegsvar på börvärde och processtörning:



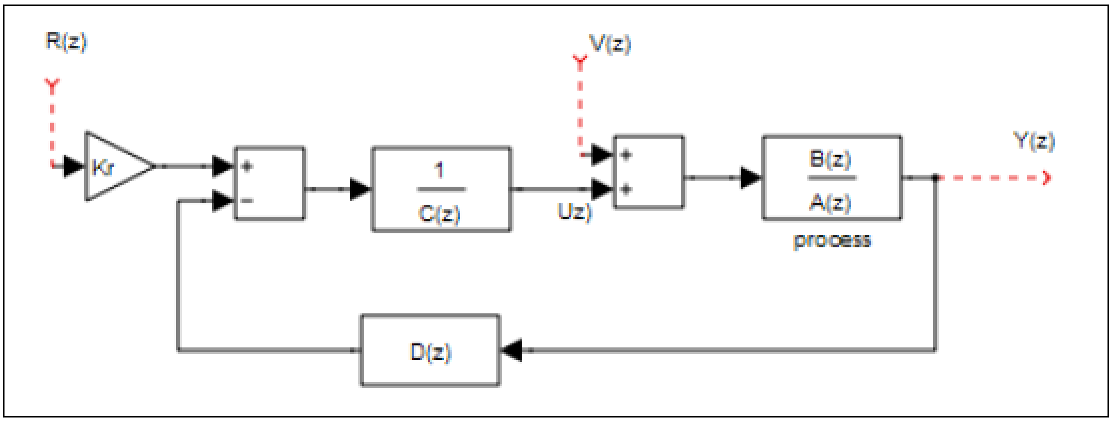
***Figur 6** PID finskjusterad: Stegsvar börvärde och processtörning*

Dimensionering av polplaceringsregulator

Diskretisering av $K_u G_p(s)$ med 1 sampling per sekund ger med MATLAB-kommandot ”c2d” ger:

$$K_u H_p(z) = \frac{1.526 \cdot 10^{-6} z + 1.361 \cdot 10^{-6}}{z^4 - 1.686 z^3 + 0.7097^2} = \frac{1.526 \cdot 10^{-6} z^{-3} + 1.361 \cdot 10^{-6} z^{-4}}{1 - 1.686 z^{-1} + 0.7097^{-2}}$$

Vi ska ta fram en polplaceringsregulator enligt modellen:



***Figur 7** Blockschema: Polplacering*

Där procesen $\frac{B(z)}{A(z)} = K_u H_p(z)$ och polplaceringsekvationen är $P(z) = A'(z) \cdot C'(z) + B(z) \cdot D(z)$ där $A'(z) = A(z) \cdot (1 - z^{-1})$ och $C(z) = C'(z) \cdot (1 - z^{-1})$. Det vill säga integrationen för över till processen $A(z)$ och med denna ”låtsasprocess” dimensioneras regulatorn på vanligt sätt. Vi ansätter gradtalen $n_{c'} = n_b - 1 = 3$ och $n_d = n_{a'} - 1 = 2$ samt $m = n_{a'} = 3$ poler utanför origo. Alltså:

$$C'(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3}, \qquad D(z) = d_0 + d_1 z^{-1} + d_2 z^{-2}, \qquad P(z) = (1 - q z^{-1})^{-3}$$

Där q bestämmer polplacering. Vi löser ekvationen $A'(z) \cdot C'(z) + B(z) \cdot D(z) = (1 - q z^{-1})^{-3}$ och får värden på regulatorparametrarna som funktion av q. För utförliga beräkningar, egen generering av regulatorparametrar samt simulering se bilaga för MATLAB-skript. För valda polplaceringar erhålls:

	K_r	c_1	c_2	c_3	d_0	d_1	d_2
$q = 0.3$	1.1881e+05	1.7860	2.6715	1.4604	1.3887e+06	-2.0315e+06	7.6153e+05
$q = 0.5$	4.3298e+04	1.1860	1.5399	0.8034	7.0523e+05	-1.0809e+06	4.1892e+05
$q = 0.7$	9.3523e+03	0.5860	0.6483	0.3188	2.5254e+05	-4.0941e+05	1.6622e+05

***Tabell 4** Polplaceringsregulator: Regulatorparametrar*

Sedan tar vi fram differensekvationerna för

$Y(z) = U(z) \cdot K_u H_p(z)$ samt $U(z) = \frac{K_r \cdot R(z) - D(z) \cdot Y(z)}{C(z)}$ för att därefter iterativt kunna beräkna ett stegsvar vid börvärdesändring. Framtagna differensekvationer presenteras nedan.

```
y(k)=-a1*y(k-1)-a2*y(k-2)+b3*u(k-3)+b4*u(k-4)
u(k)=Kr*r(k)-(c1-1)*u(k-1)-(c2-c1)*u(k-2)-(c3-c2)*u(k-3)-(-c3)*u(k-4)-d0*y(k)-d1*y(k-1)-d2*y(k-2)
```

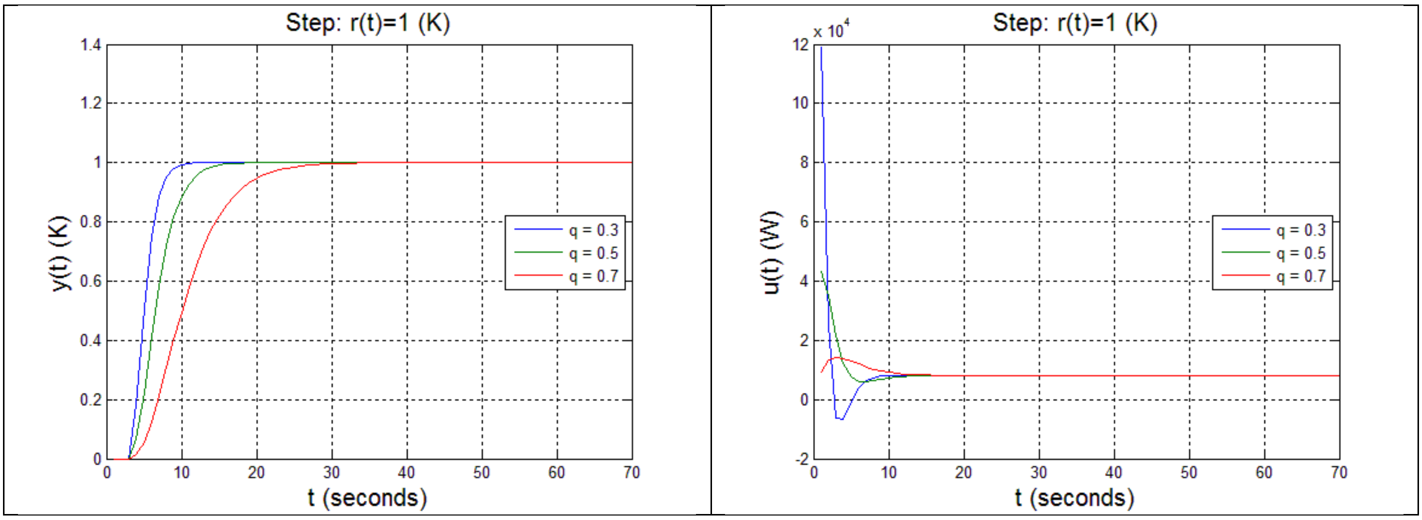
***Tabell 4** Differensekvationer*

Vid simulering av enhetssteg på börvärdet på differensekvationerna erhåller vi från ”stepinfo” för olika värden på q prestandan:

	$q = 0.3$	$q = 0.5$	$q = 0.7$
Stigtid	3.6635 s	6.1375 s	11.8746 s
Insvängningstid	8.9842 s	13.5305 s	23.6540 s
Översläng	0 %	0 %	0 %
Topp	1.0000	1.0000	1.0000

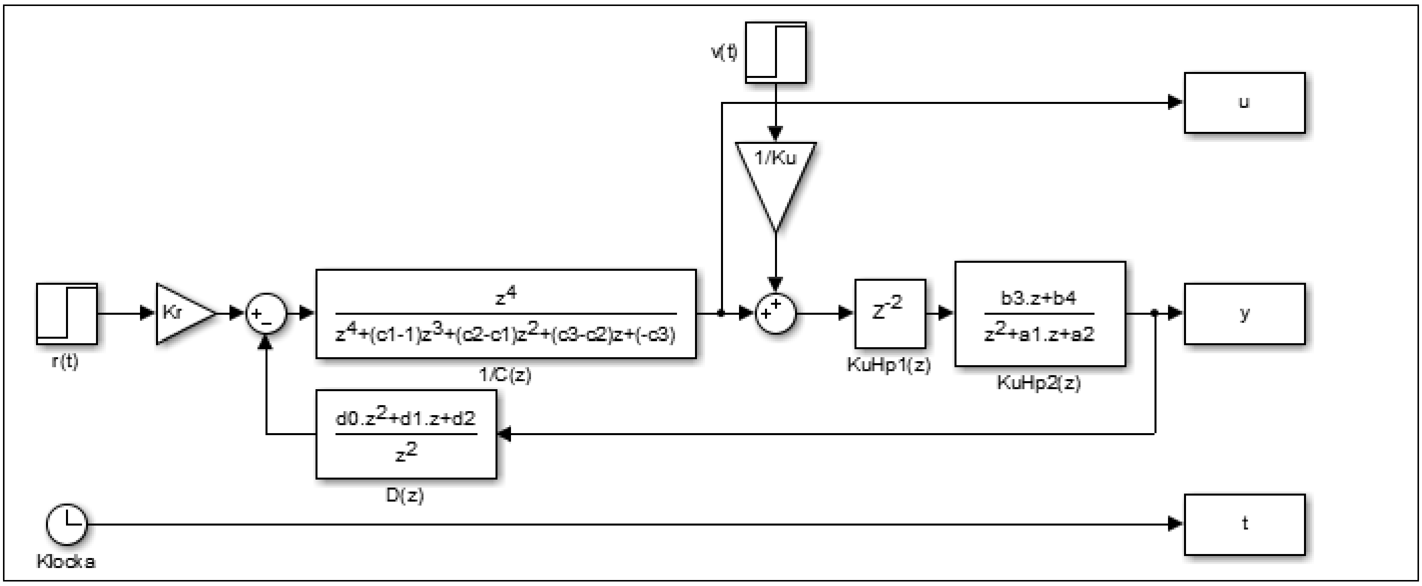
***Tabell 5** Polplaceringsregulator iterativt: prestanda*

Vi plottar stegsvaren för utloppstemperatur $y(t)$ [W] och styreffekt $u(t)$ [K].



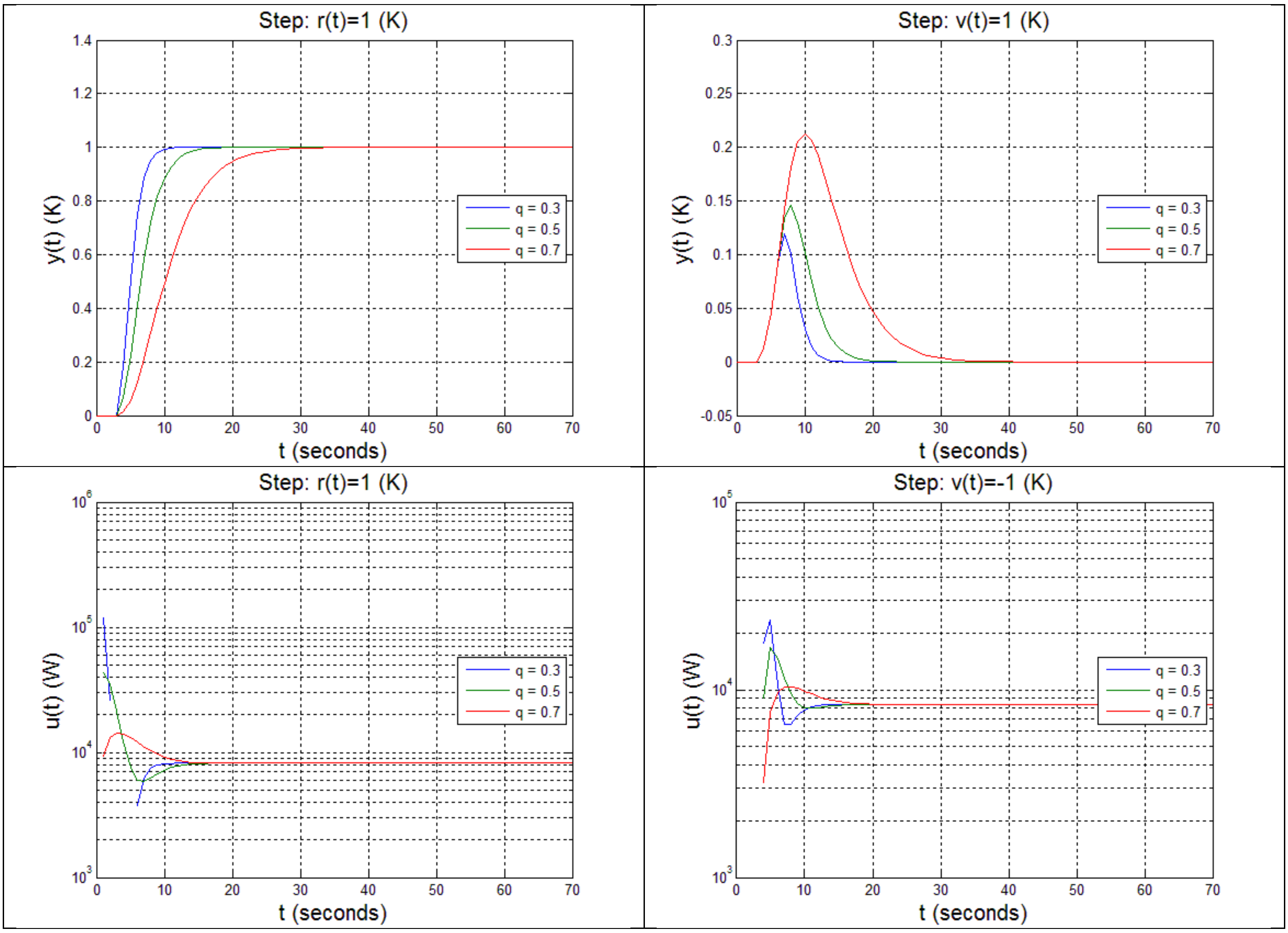
***Figur 8** Polplaceringsregulator iterativ: Stegsvar börvärdesändring*

Slutligen simulerar vi också polplaceringsregulatorn för framtagna regulatorparametrar i Simulink enligt:



***Figur 9** Polplaceringsregulator: Simulink*

Från Simulink erhåller vi stegsvaren:



Figur 10 *Polplaceringsregulator Simulink: Stegsvar börvärde och processtörning*

Vi noterar att resultaten i figur 8 är i princip identiska med de i figur 10. Och att styreffekten när tiden går mot oändligheten vid stegsvaren ligger på samma nivå för figur 6, 8 och 10 nämligen 8360 [W] som förväntat enligt figur 3. I övrigt anses karakteristiken för resultaten lik i jämförelse mellan den finskjusterade PID-regulator (figur 6) och polplaceringsregulatorn (figur 10).

Val av regulator

Valet står mellan den finskjusterade PID-regulatorn och polplaceringsregulatorn med tre olika polplaceringar. Vi jämför prestandan i tabell 3 och 5 samt figur 6 och 10. Vi förslår polplaceringsregulatorn med $q = 0.5$ enligt figur 7, tabell 4. Jämfört med PID-regulatorn har vi ungefär samma insvängningstid men PID-regulatorn ger en översläng som vi slipper med vårt val. Den valda polplaceringsregulatorn ger även en snabbare respons vid processtörningar, med rimlig styreffekt. Polplacering $q = 0.7$ anser vi är för långsam i jämförelse med vårt vall. Sist jämför vid med $q = 0.3$ den är lite snabbare än $q = 0.5$, vi ser dock en orimlig högstyreffekt vid börvärdesändring och en antydan till oscillation i styreffekten. Generellt sett ger polplacering närmare origo lägre stabilitet och med valet $q = 0.5$ får vi även en mindre känslighet för variation i förstärkningen hos regleringen.

```
% Välj först godtycklig polplacering (q_k)
% - Ger väldigt mycket.

q_k = 0.5;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

V1 = 10*10^-3; % [m^3]
Vr =  4*10^-3; % [m^3]
V2 = 14*10^-3; % [m^3]
Q  =  2*10^-3; % [m^3/s]
c = 4180; % [J/K*kg]
r = 1000; % [kg/m^3]

s=tf('s');
G1 = 1/((V1/Q)*s+1);
G2 = exp(-(Vr/Q)*s);
G3 = 1/((V2/Q)*s+1);
Kv = 1;
Ku = 1/(Q*c*r);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

KuHp_tf = c2d(Ku*G1*G2*G3,1); % c2d = kontinuerlig till diskret
% "step(G1*G2*G3,KuHp_tf)" Jämförelse kontinuerlig tidsdiskret modell
% = >
% KuHp(z) = B(z)/A(z) = (b3*z^(-3)+b4*z^(-4)/(1+a1*z^(-1)+a2*z^(-2))
% = >
% P(z) = A'(z)*C'(z) + B(z)*D(z)  där A' = A(z)*(1-z^-1), C(z) = C'(z)*(1-z^(-1))
% nc' = nb -1 = 3
% nd  = na'-1 = 2
% m   = na'   = 3 => P(z) = (1-q*z^(-1))^3
syms z
syms a1 a2 b3 b4
syms c1 c2 c3 d0 d1 d2
syms q
A     = collect(1 + a1*z^-1 + a2*z^-2);
Aprim = collect(A*(1-z^-1));
B     = collect(b3*z^-3 + b4*z^-4);
Cprim = collect(1 + c1*z^-1 + c2*z^-2 + c3*z^-3);
C     = collect(Cprim*(1-z^-1));
D     = collect(d0 + d1*z^-1 + d2*z^-2);
P     = (1-q*z^-1)^3;
Kr    = collect(subs(P/B, z, 1));
% A'(z)*C'(z) + B(z)*D(z) = (1-q*z^(-1))^3
% =>
% z^5: a1 + c1 - 1 = -3*q
% z^4: a2 - a1 - c1 + c2 + a1*c1 = 3*q^2
% z^3: c3 - c2 - a2 - a1*c1 + a1*c2 + a2*c1 + b3*d0 = -q^3
% z^2: a1*c3 - a1*c2 - a2*c1 - c3 + a2*c2 + b3*d1 + b4*d0 = 3*q^2
% z^1: a2*c3 - a2*c2 - a1*c3 + b3*d2 + b4*d1 = 0
% z^0: -a2*c3 + b4*d2 = 0
% AMatrix * XMatrix = BMatrix där XMatrix = [c1; c2; c3; d0; d1; d2]
AMatrix = [1       0       0       0       0       0 ;
           -1+a1   1       0       0       0       0 ;
           -a1+a2  -1+a1   1       b3      0       0 ;
           -a2     -a1+a2  -1+a1   b4      b3      0 ;
```

```
           0       -a2      -a1+a2  0       b4      b3;
           0       0        -a2     0       0       b4;];
BMatrix = [-3*q-(a1-1)   ;
           3*q^2-(a2-a1);
           -q^3-(-a2)    ;
           0             ;
           0             ;
           0             ;];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Insättning av konstanter
a1_k = -1.686;
a2_k = 0.7097;
b3_k = 1.526e-06;
b4_k = 1.361e-06;
% =>
Kr = subs(Kr, [b3 b4], [b3_k b4_k])
Kr = double(subs(Kr, [b3 b4 q], [b3_k b4_k q_k]))
fprintf('\nXMatrix = [c1; c2; c3; d0; d1; d2]\n')
XMatrix = collect(subs(inv(AMatrix)*BMatrix,[a1 a2 b3 b4],[a1_k a2_k b3_k b4_k]))
XMatrix = double(subs(inv(AMatrix)*BMatrix,[a1 a2 b3 b4 q],[a1_k a2_k b3_k b4_k q_k]));
c1_k = XMatrix(1,1);
c2_k = XMatrix(2,1);
c3_k = XMatrix(3,1);
d0_k = XMatrix(4,1);
d1_k = XMatrix(5,1);
d2_k = XMatrix(6,1);
% =>
A = subs(A,[a1 a2],[a1_k a2_k]);
B = subs(B,[b3 b4],[b3_k b4_k]);
C = subs(C,[c1 c2 c3],[c1_k c2_k c3_k]);
D = subs(D,[d0 d1 d2],[d0_k d1_k d2_k]);
P = subs(P,[q],[q_k]);
% =>
q = q_k
a1 = a1_k;
a2 = a2_k;
b3 = b3_k;
b4 = b4_k;
c1 = c1_k
c2 = c2_k
c3 = c3_k
d0 = d0_k
d1 = d1_k
d2 = d2_k
% =>
A_tf = sym2dtf(A,1);
B_tf = sym2dtf(B,1);
C_tf = sym2dtf(C,1);
D_tf = sym2dtf(D,1);
P_tf = sym2dtf(P,1);
% =>
Htot_tf = Kr*B_tf/(A_tf*C_tf+B_tf*D_tf); % Y(z)/R(z)
step(Htot_tf)
grid on
```

```
% Plottar stegsvaret för valda polplaceringar (q1, q2, q,3),
% samt anger presentear prestanda m.h.a. "stepinfo".

q1 = 0.3;
q2 = 0.5;
q3 = 0.7;

[y1 u1 t1] = d_iterativ(q1);
stepinfo(y1,t1)
[y2 u2 t2] = d_iterativ(q2);
stepinfo(y2,t2)
[y3 u3 t3] = d_iterativ(q3);
stepinfo(y3,t3)

subplot(2,1,1)
hlines1 = plot(t1,y1,t2,y2,t3,y3);
title('Step: r(t)=1 (K)','Fontsize',15)
xlabel('t (seconds)','Fontsize',15)
ylabel('y(t) (K)','Fontsize',15)
grid on
set(hlines1(1),'Displayname',['q = ' num2str(q1)])
set(hlines1(2),'Displayname',['q = ' num2str(q2)])
set(hlines1(3),'Displayname',['q = ' num2str(q3)])
legend('Location','east')

subplot(2,1,2)
hlines2 = plot(t1,u1,t2,u2,t3,u3);
title('Step: r(t)=1 (K)','Fontsize',15)
xlabel('t (seconds)','Fontsize',15)
ylabel('u(t) (W)','Fontsize',15)
grid on
set(hlines2(1),'Displayname',['q = ' num2str(q1)])
set(hlines2(2),'Displayname',['q = ' num2str(q2)])
set(hlines2(3),'Displayname',['q = ' num2str(q3)])
legend('Location','east')
```

```
% Välj först godtycklig polplacering (q_k)
% - För given polplacering presenterar funktionen regulatorparametrarna
(Kr,c1,c2,c3,d0,d1,d2).
% - Simulerar det återkopplade systemet för N = 70 samplingar, returnerar i vektorer
%   utloppstemperaturen (y), styreffekten (u) och tiden (t).

function [y u t] = d_iterativ(q_k)
a1 = -1.6860;
a2 = 0.7097;
b3 = 1.5260e-06;
b4 = 1.3610e-06;
q  = q_k
Kr = -(2361183241434822606848*q^3)/6816736018022333 + (7083549724304467820544*q^2)/
6816736018022333 - (7083549724304467820544*q)/6816736018022333 + 2361183241434822606848/
6816736018022333
c1 = 1343/500 - 3*q
c2 = 3*q^2 - (4029*q)/500 + 301181/62500
c3 = -(1327466112874474796439510129852430213846220285015000*q^3)/
10651332194076104339061685961065158384970219288791281 + (
24947146271000164144303648447911839060843430931753479*q^2)/
10651332194076104339061685961065158384970219288791281 - (
27149224677262573319874147130223483924911780487994583797*q)/
5325666097038052169530842980532579192485109644395640500 +
926104841101463488780781550149655958071494426115943031429/
332854131064878260595677686283286199530319352774727531250
d0 = -(6110003985060045229383251247700656165922412773939236306944*q^3)/
10651332194076104339061685961065158384970219288791281 + (
4986999389651464712285845635370638712731929466996305474617344*q^2)/
1331416524259513042382710745133144798121277411098910125 - (
1020688656614397694732976264088216396073921457878295233217691648*q)/
166427065532439130297838843141643099765159676387363765625 +
60475544797479590835593638557318167370326651002239648169168207872/
20803383191554891287229855392705387470644959548420470703125
d1 = (3112805664800708920027824490335592561049122132143974121472*q^3)/
10651332194076104339061685961065158384970219288791281 - (
26147855538443263296679856319499022069032954426628573195403264*q^2)/
6657082621297565211913553725665723990606387055494550625 + (
6450783282633810139992634599826566114833567059036747201813413888*q)/
832135327662195651489194215708215498825798381936818828125 -
417261381520136873746486073947567131931471321624608779333881823232/
104016915957774456436149276963526937353224797742102353515625
d2 = -(692213593171943264470864977050316873216081921657656049664*q^3)/
10651332194076104339061685961065158384970219288791281 + (
8130505927869588936174925144674666901790880648121267513196544*q^2)/
6657082621297565211913553725665723990606387055494550625 - (
2212045916772277816443290404639088934016156464204548747084955648*q)/
832135327662195651489194215708215498825798381936818828125 +
150913070749841259156665261379876495061179595553380443128040783872/
104016915957774456436149276963526937353224797742102353515625

% ITERATIV BERÄKNING
%
% A(z)  = 1 + a1*z^-1 + a2*z^-2
% B(z)  = b3*z^-3 + b4*z^-4
% C(z)  = 1 + (c1-1)*z^-1 + (c2-c1)*z^-2 + (c3-c2)*z^-3 + (-c3)*z^-4
% D(z)  = d0 + d1*z^-1 + d2*z^-2
%
```

```
%                B(z)
% Y(z) = U(z) * ----
%                A(z)
%
%         Kr*R(z)-D(z)*Y(z)
% U(z) = -----------------
%              C(z)
%

N = 70;
r = 1; % Simulera enhetssteg
clear y u;
t = 1:N;

k = 1;
y(k) = 0;
u(k) = Kr*r - d0*y(k);
k = 2;
y(k) = -a1*y(k-1);
u(k) = Kr*r - (c1-1)*u(k-1) - d0*y(k) - d1*y(k-1);
k = 3;
y(k) = -a1*y(k-1) - a2*y(k-2);
u(k) = Kr*r - (c1-1)*u(k-1) - (c2-c1)*u(k-2) - d0*y(k) - d1*y(k-1) - d2*y(k-2);
k = 4;
y(k) = -a1*y(k-1) - a2*y(k-2) + b3*u(k-3);
u(k) = Kr*r - (c1-1)*u(k-1) - (c2-c1)*u(k-2) - (c3-c2)*u(k-3) - d0*y(k) - d1*y(k-1) - d2*y(k-
2);

for k=5:N
    y(k) = -a1*y(k-1) - a2*y(k-2) + b3*u(k-3) + b4*u(k-4); % MÄT via AD-omvandlare
    u(k) = Kr*r - (c1-1)*u(k-1) - (c2-c1)*u(k-2) - (c3-c2)*u(k-3) - (-c3)*u(k-4) - d0*y(k) -
    d1*y(k-1) - d2*y(k-2); % STYR via DA-omvandlare
end

%
% SLUT
end
```

```
function H = sym2dtf(h,Ts)
[num,den]=numden(h);
num_n=sym2poly(num);
den_n=sym2poly(den);
H = tf(num_n,den_n,Ts);
end
```