Appendix A: Plots of selected controlled process models

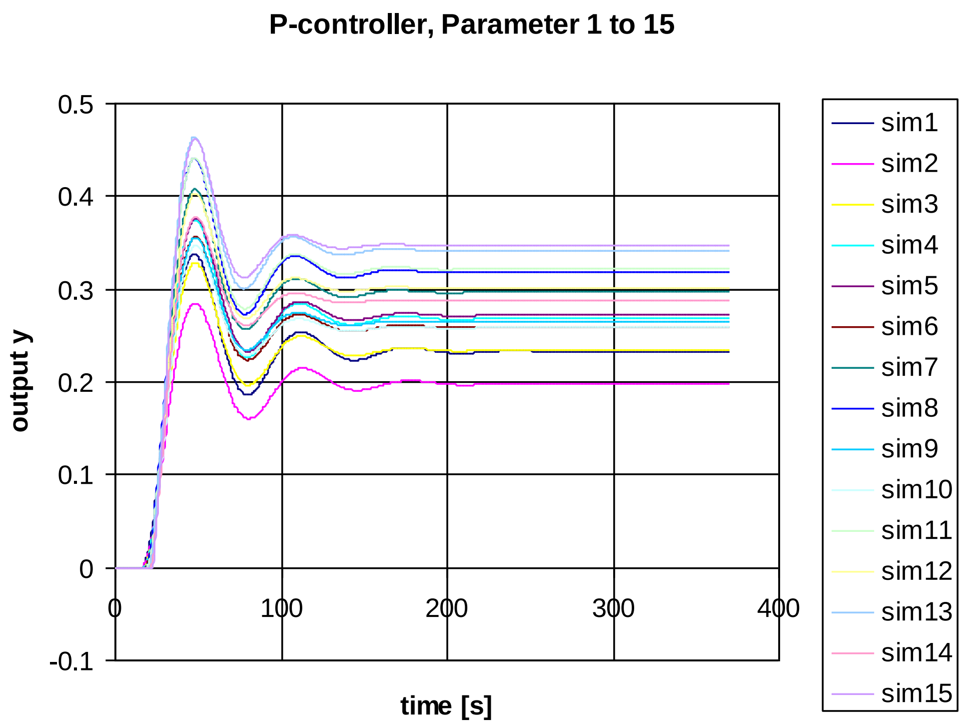

Figure I Results of Process 4 simulation with P-controller.

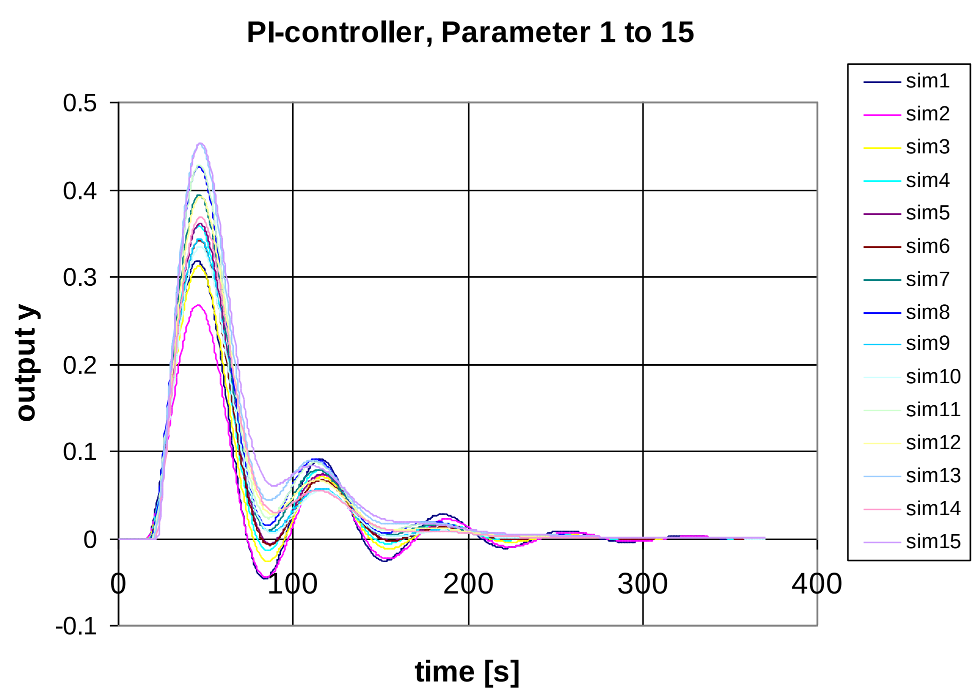

Figure II Results of Process 4 simulation with PI-controller.

Appendix B: Mathlab code for calculation of IAE and TV

The processes were simulated in Simulink, and IAE and TV were calculated in Mathlab by using the script IAEandTV.m

```
% beregninger av IAE og TV for settpunktsendringer
s = [1 0]; % 1 for settpunktendring og 0 for load change
step=ys(2,1)-ys(1,1); % Benyttet konstant steglengde, krav for Simpsons
% y output er lagret i ys- for settpunktsendring
for i=2:length(ys(1,:)) % antall y rekkevektorer (-1 for tid-vektor)
   e=abs(ys(:,i)-s(1)); % avviket fra settpunktet
   m=[1:2:N-2]; % indexer for IAE beregninger
   % Simpson`s 1/3 regelen anvendes. Den kan anvendes ved odde antall målinger
   %evt partall antall subintervaller
   IAE = sum(step/3*(e(m)+4*e(m+1)+e(m+2)));%beregner
   IAE_set((i-1),:)=IAE;
end
   % Beregner Total Variasjonen(TV) i output(u) fra
   % controlleren for settpunktsendring
for i=2:length(us(1,:))
   N=length(yd); % antall funksjonsevalueringer
   u_s=us(:,i);
   k=[2:1:N]; % indexer for TV beregninger
   TV = sum(abs(u_s(k)-u_s(k-1))); %TV=sum(abs[x
   TV_set(i-1,:)=TV;
end
% beregninger av IAE og TV for loadchange
step=y_load(2,1)-y_load(1,1); % Benyttet konstant steglengde, krav for Simpsons
% y output er lagret i yload- for settpunktsendring
for i=2:length(y_load(1,:)) % antall y rekkevektorer (-1 for tid-vektor)
   e=abs(y_load(:,i)-s(2)); % avviket fra settpunktet
   m=[1:2:N-2]; % indexer for IAE beregninger
   % Simpson`s 1/3 regelen anvendes. Den kan anvendes ved odde antall målinger
   %evt partall antall subintervaller
   IAE = sum(step/3*(e(m)+4*e(m+1)+e(m+2)));%beregner
   IAE_load(i-1,:)=IAE;
end
   % Beregner Total Variasjonen(TV) i output(u) fra
   % controlleren for load change
for i=2:length(u_load(1,:))
   N=length(u_load); % antall funksjonsevalueringer
   uload=u_load(:,i);
   k=[2:1:N]; % indexer for TV beregninger
   TV = sum(abs(uload(k)-uload(k-1)));
   TV_load(i-1,:)=TV;
end
IAE_set %Resultater
TV_set
IAE_load
TV_load
```

Appendix C: Plots of simulated models for analysis of IAE and TV

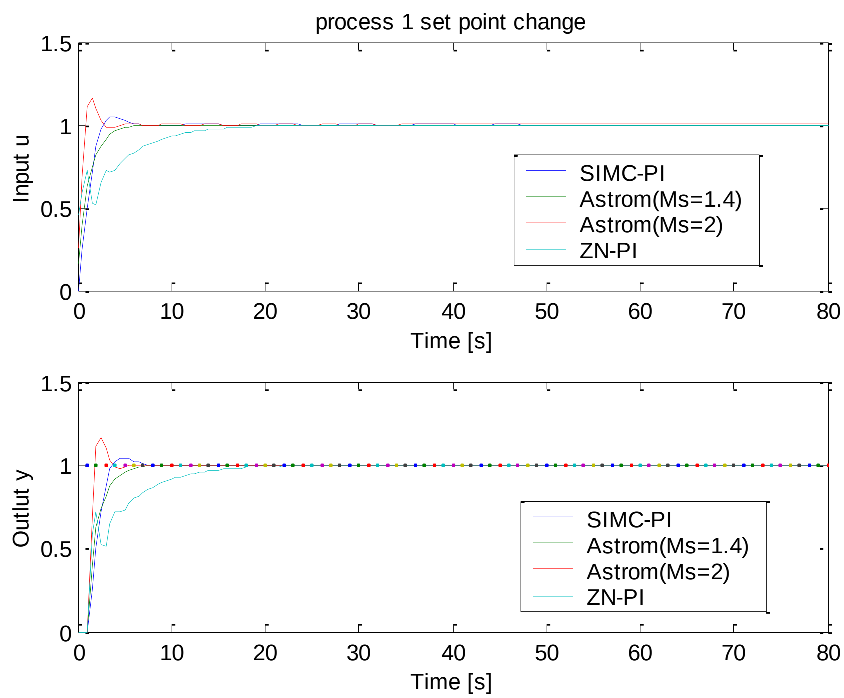

Figure III Pure dead time process, set point change.

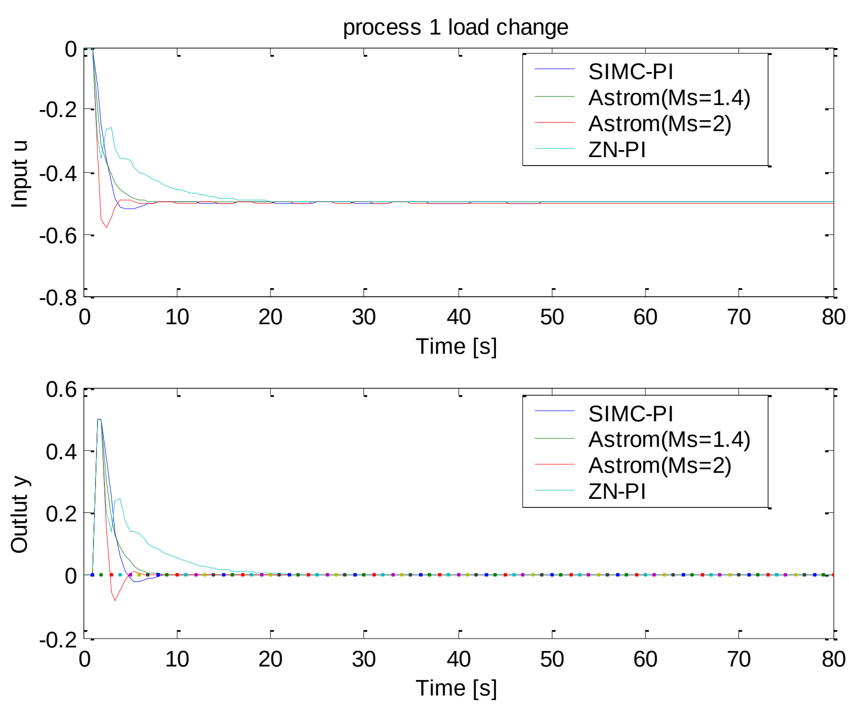

Figure IV Pure dead time process, load change.

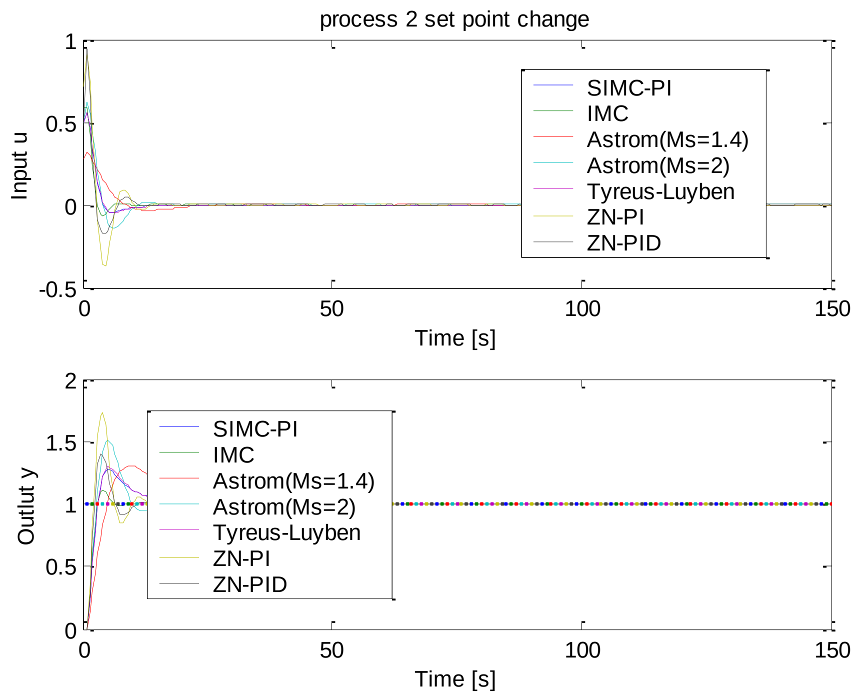

Figure V Integrating process, set point change.

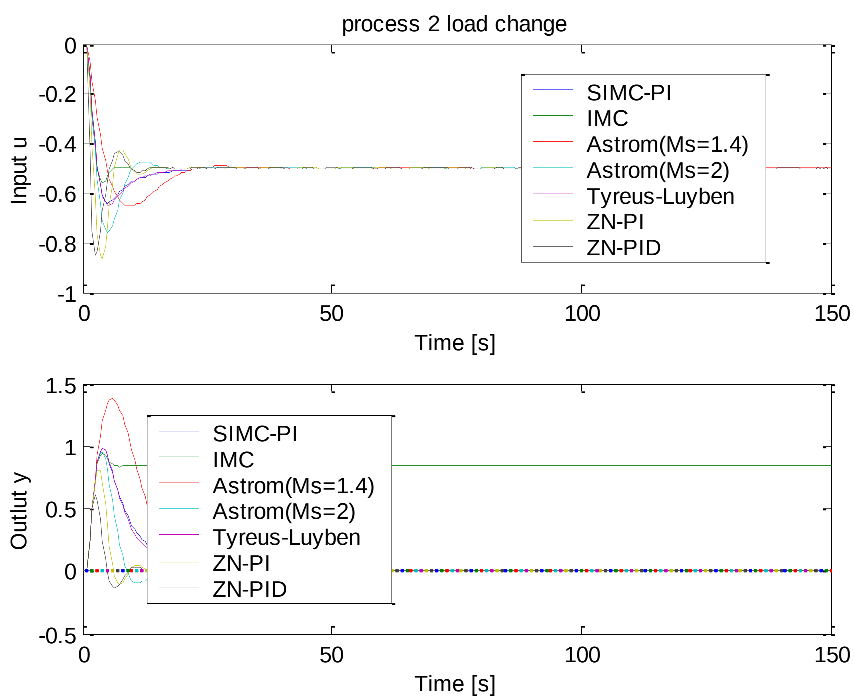

Figure VI Integrating process, load change.

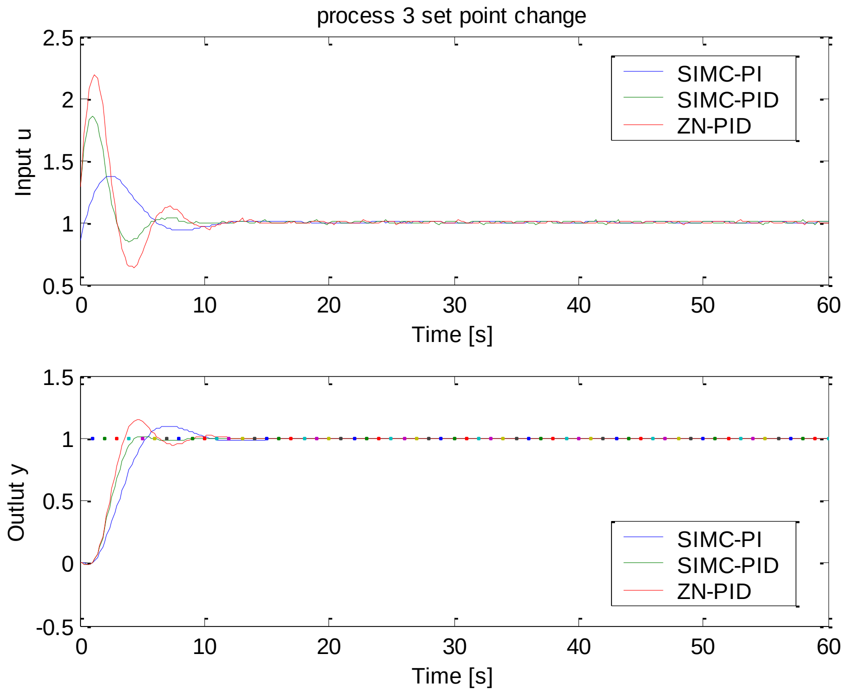

Figure VII Process example G_3, set point change.

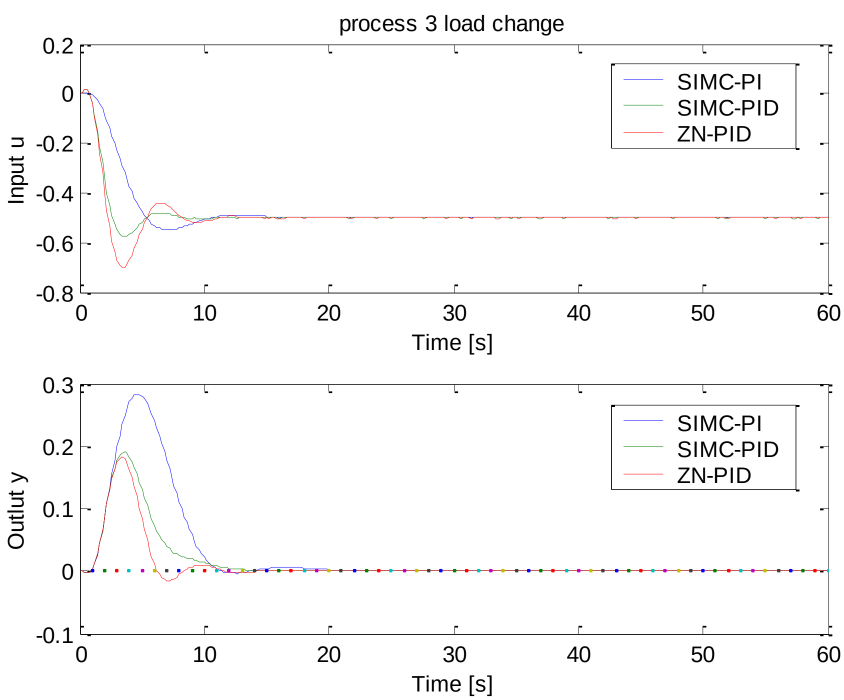

Figure VIII Process example G_3, load change.

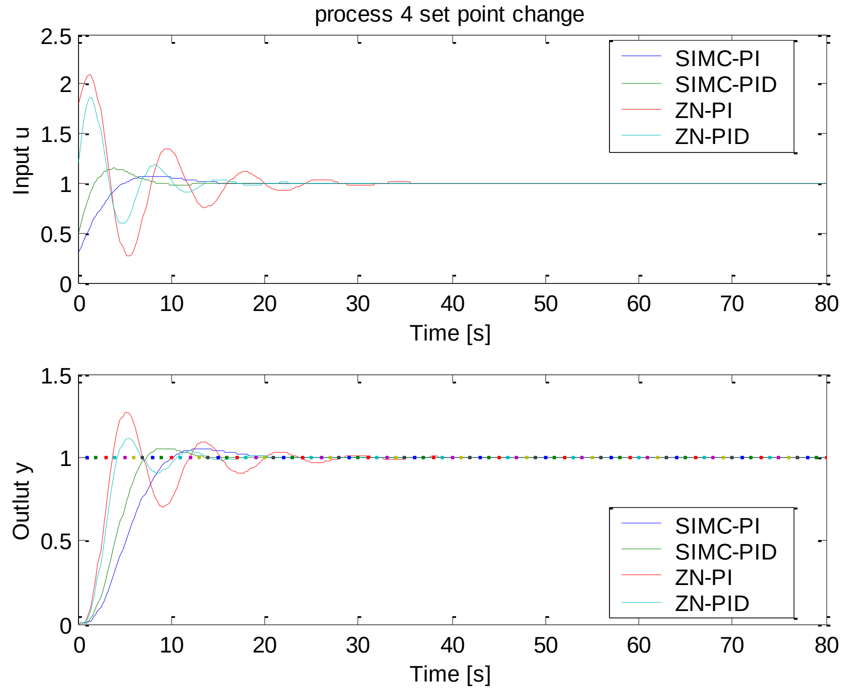

Figure IX Process example G_4, set point change.

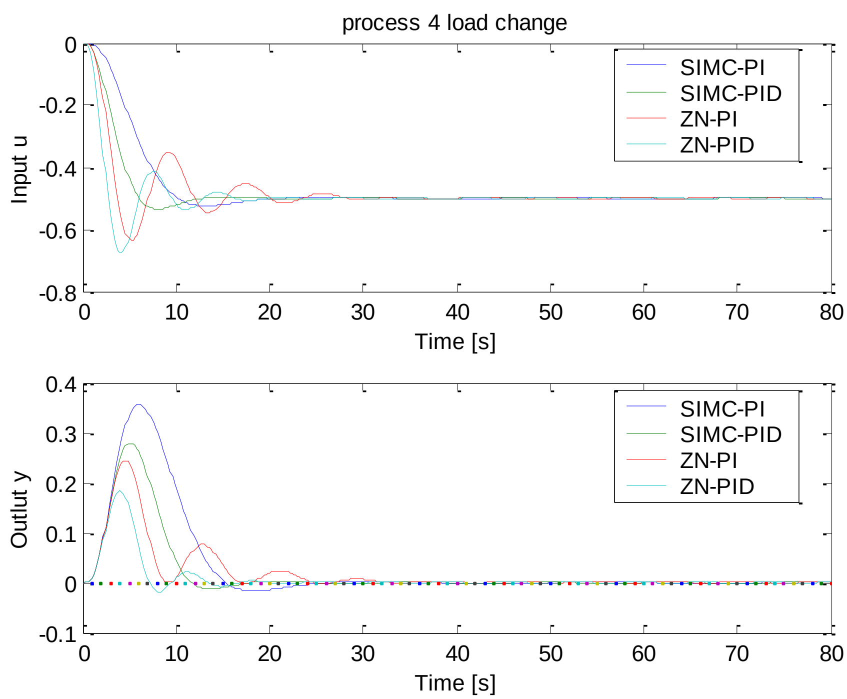

Figure X Process example G_4, load change.

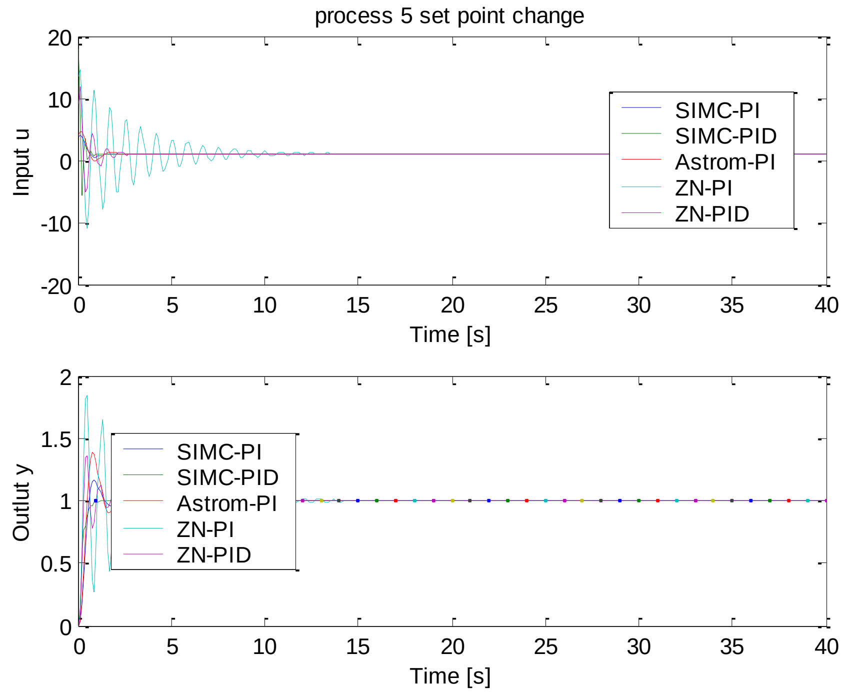

Figure XI Process example G_5, set point change.

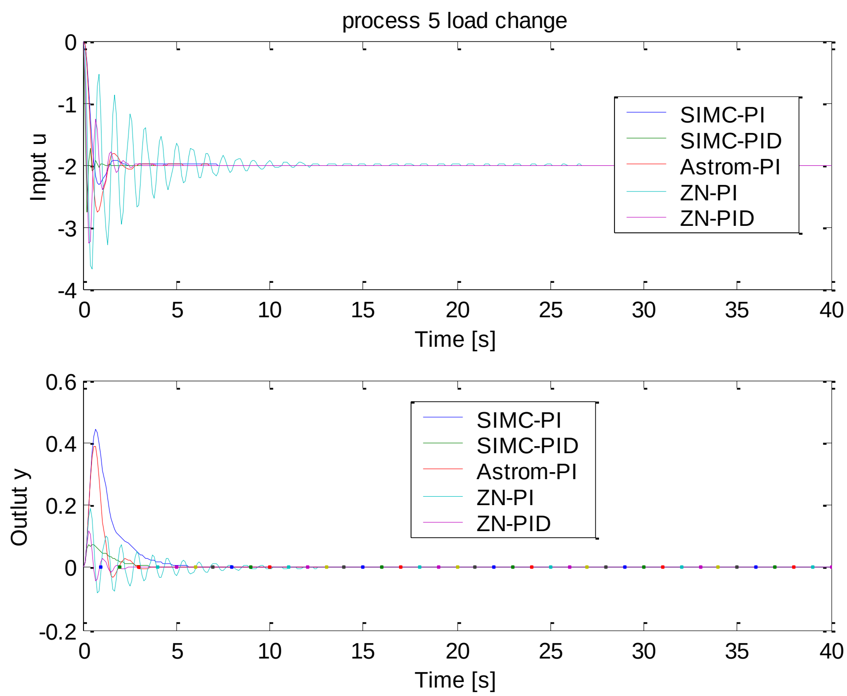

Figure XII Process example G_5, load change.

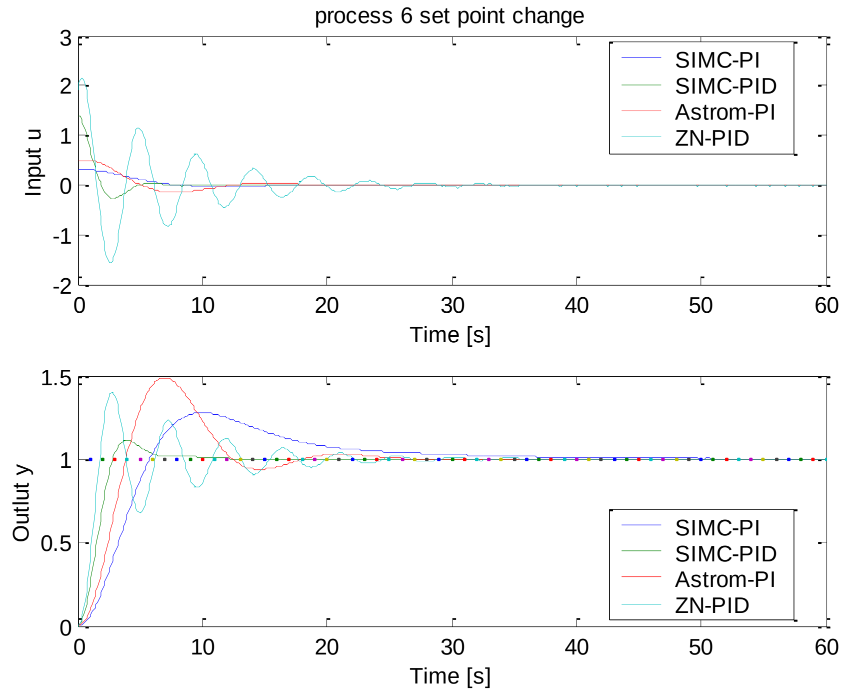

Figure XIII Process example G_5, set point change.

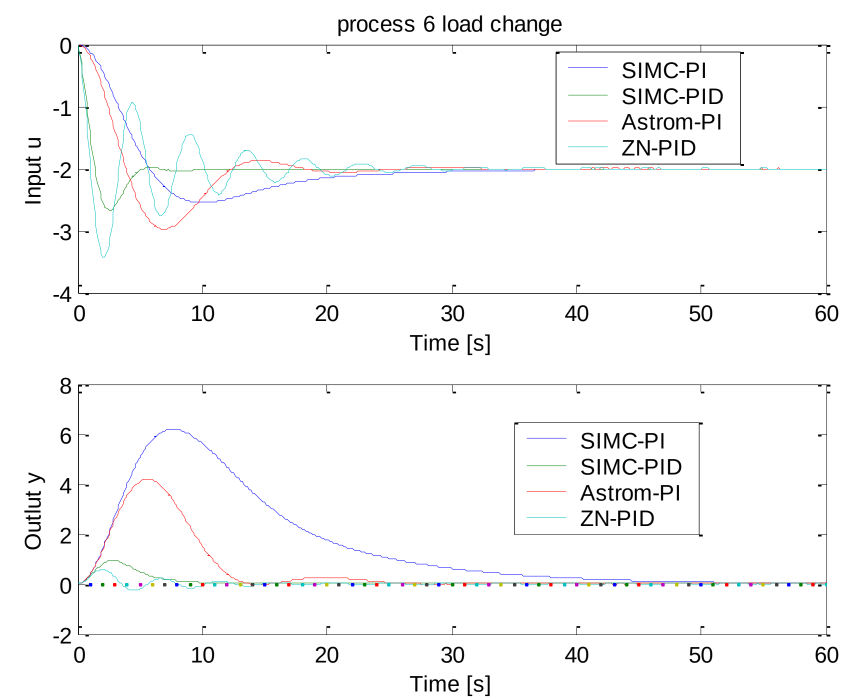

Figure XIV Process example G_6, load change.

Appendix D: Mathlab code for frequency analysis routine

minbode.m routine

```
function [A,vA,w180,wc,GM,PM,Ms,Mt]=minbode_L(num,den,gain,deadtime,w)
%Frequency analysis of SISO system L= gain*deadtime*num/den
%By Truls Larsson, Trondheim 15.09.98 and 21.09.98
%
%Kall:  minbode(num,den,gain,deadtime,w)
%gir:   Bode-plott av L samt frekvensplott av |S| og |T| (S=1/1+L; T=L/1+L)
%
%eller: [A,vA,w180,wc,GM,PM,Ms,Mt]=minbode(num,den,gain,deadtime,w)
%gir:   amplitude (A) og fase (vA) av L som funksjon av frekvens,
%       w180 (der va=-180 grader), wc (der A=1), GM, PM,
%       samt Ms (peak for |S|) og Mt (peak for |T|)
%
%Input:
%num      : er teller polynomet
%den      : er nevner polynomet
%gain     : er forsterkning
%deadtime : er dodtid
%w        : er frekvensene (NB! omraadet maa vaere stort nok)
%Eksempel:
%num=[2 1]; den=[25 10 1 0];  gain=0.1; deadtime=3; w= logspace(-2,2,100);
%
if nargin==0
  disp('usage:  minbode(num,den,gain,deadtime,w)')
  disp('or:  [A,vA,w180,wc,GM,PM,Ms,Mt]=minbode(num,den,gain,deadtime,w)')
  return
end

%Frekvensresponsen
teller=polyval(num,w*i);
nevner=polyval(den,w*i);
Dod   =exp(-deadtime*w*i);
Frek=gain*teller./nevner.*Dod;  % korrigert av SiS

%Amplitude og fase
A=abs(Frek);
vA=angle(Frek);

%Test for stor sprang
vA1=vA(1:length(vA)-1);
vA2=vA(2:length(vA));
DvA=vA2-vA1;

for i=1:length(DvA)
  if DvA(i) > 3
    vA(i+1:length(vA))=vA(i+1:length(vA))-2*pi;
  end
end
```

```
vA=vA*180/pi;

%Beregning av w180 og gain margin
if min(vA)>-180
  disp(['Ingen fase kryssover frekvens i det aktuelle frekvens' ...
              ' intervallet'])
    w180=[];
    GM=[];
else
  for i=1:length(vA)
    if vA(i)>-180
      posisjon=i;
    end
  end
  %Secant
  w180=w(posisjon)-(vA(posisjon)+180)* ...
      (w(posisjon+1)-w(posisjon))/(vA(posisjon+1)-vA(posisjon));
  %Gain margin ved interpolasjon
  Lw180=A(posisjon)+(w180-w(posisjon))*(A(posisjon+1)-A(posisjon))...
      /(w(posisjon+1)-w(posisjon));
  GM=1/Lw180;
end



%Beregning av wc
if max(A)<1
  %disp('Maks amplitude mindre enn ein.');  skrudd av
  wc=[];
  vAwc=[];
  PM=[];
else
  for i=1:length(A)
    %Kryssing oven fra
    if A(i)>1
      posisjon=i;
    end
  end
  %Secant
  wc=w(posisjon)-(A(posisjon)-1)* ...
      (w(posisjon+1)-w(posisjon))/(A(posisjon+1)-A(posisjon));
  %Fasen ved
  vAwc=vA(posisjon)+(wc-w(posisjon))*(vA(posisjon+1)-vA(posisjon))...
      /(w(posisjon+1)-w(posisjon));
  PM=vAwc+180;
end

%Beregning av S og T
S=1./abs(1+Frek);
Ms=max(S);
T=abs(Frek./(1+Frek));
Mt=max(T);

if nargout==0
 figure
 title('Bode plot')
```

```
subplot(211)
loglog(w,A)
hold
loglog([w180,w180],[1,Lw180])
loglog(w,ones(length(w)),':')
ylabel('Amplitude')
title('Bode plot')
hold
subplot(212)
semilogx(w,vA)
hold
semilogx(w,-ones(length(w))*180,':')
semilogx([wc,wc],[vAwc,-180])
axis([min(w) max(w) max([min(vA),-360]) 0])
ylabel('Fase forskyvning')
xlabel('Frekvens')
hold

figure
loglog(w,S,w,T,'.-')
legend('S','T')
xlabel('Frekvens')
ylabel('Amplitude')
end
```