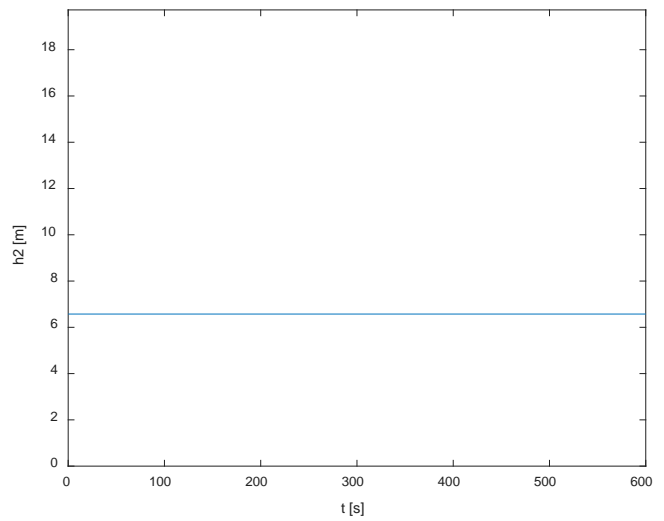
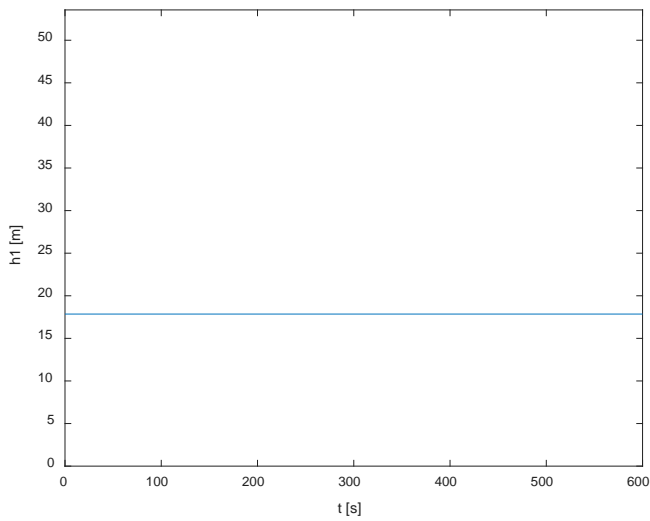


Soluzione Ese 12 – Dimensionamento controllori con Cohen-Coon

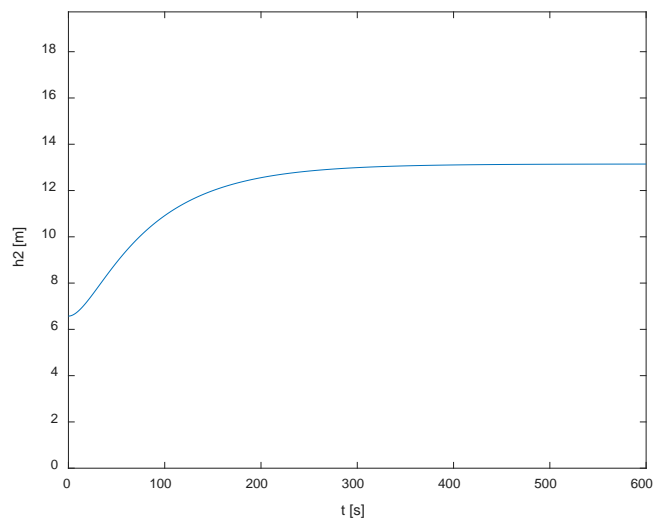
Strumentazione e Controllo di Impianti Chimici

Tutor: Giuseppe Pesenti

Condizioni stazionarie iniziali



Disturbo sul sistema in anello aperto



Codice – Simulazione del sistema in anello aperto con disturbo

```
A1=30; % m^2
A2=50; % m^2
r1=1.2; % s/m^2

% Steady-state iniziale
Fiss=9.4; % m^3/s
h2ss=Fiss/1.43; % m
h1ss=h2ss+r1*Fiss; % m

% Disturbo
A=Fiss; % m^3/s

opz=odeset('RelTol',1e-12,'AbsTol',1e-12);
[tout,hout]=ode45(@(t,h)openloop(t,h,A1,A2,r1,Fiss,A),[0 600],[h1ss h2ss],opz);

figure(1)
plot(tout,hout(:,1))
xlabel('t [s]')
ylabel('h1 [m]')
ylim([0 3*h1ss])

figure(2)
plot(tout,hout(:,2))
ylabel('h2 [m]')
xlabel('t [s]')
ylim([0 3*h2ss])

function dhdt=openloop(t,h,A1,A2,r1,Fiss,A)

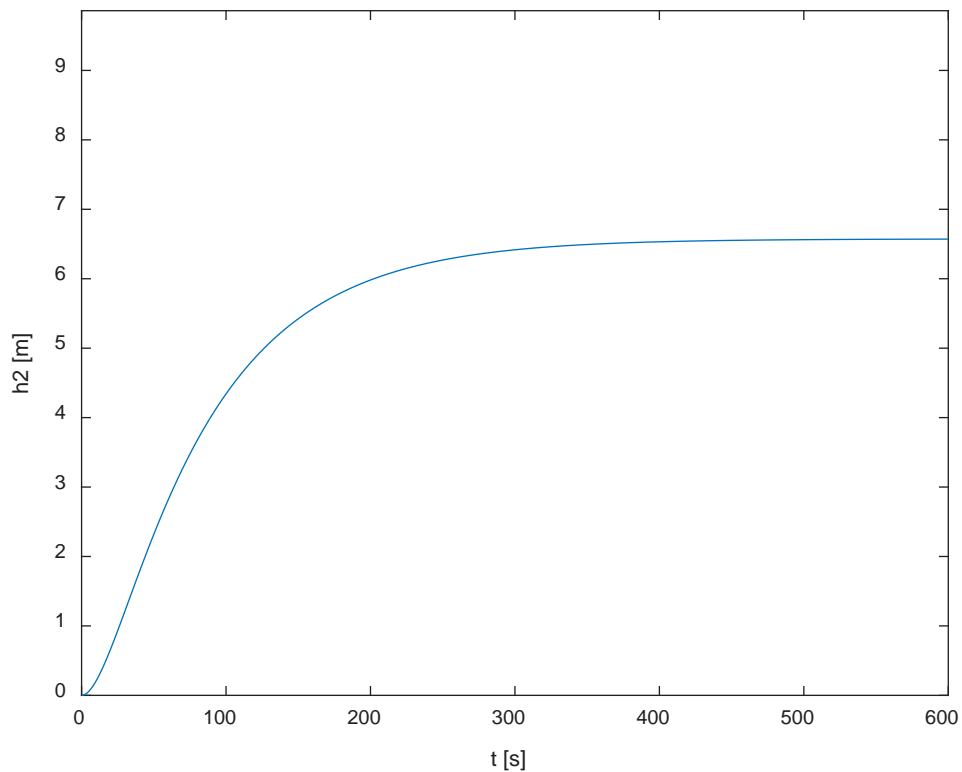
Fi=Fiss; % m^3/s
if t>0
    Fi=Fiss+A; % m^3/s
end
Fo=1.43*h(2); % m^3/s
F1=(h(1)-h(2))/r1; % m^3/s

dhdt=zeros(2,1);
dhdt(1)=(Fi-F1)/A1; % m/s
dhdt(2)=(F1-Fo)/A2; % m/s

end
```

Stima di t_d , K , τ con metodo grafico

```
figure(2)
plot(tout,hout(:,2)-h2ss)
ylabel('h2 [m]')
xlabel('t [s]')
ylim([0 1.5*h2ss])
```



```
% Costanti del sistema del primo ordine con tempo di ritardo
% Metodo grafico
```

```
A=Fiss; % m^3/s
B=6.568; % m - stima
td=8; % s - stima
S=(2.938-0)/(63.11-td); % m/s - stima
```

```
K=B/A; % s/m^2
tau=B/S; % s
```

Stima di td , K , τ con metodo numerico

```
% Costanti del sistema del primo ordine con tempo di ritardo
% Metodo numerico

h2=hout(:,2);

% Calcolo delle derivate prime
der1=zeros(1,length(tout)-1);
for i=1:length(tout)-1
    der1(i)=(h2(i+1)-h2(i))/(tout(i+1)-tout(i)); % m/s
end

% Calcolo delle derivate seconde
der2=zeros(1,length(tout)-2);
for i=1:length(tout)-2
    der2(i)=(der1(i+1)-der1(i))/(tout(i+1)-tout(i)); % m/s^2
end

% Ricerca della posizione in cui la derivata seconda cambia segno
pos=1;
while der2(pos)*der2(pos+1)>=0
    pos=pos+1;
end

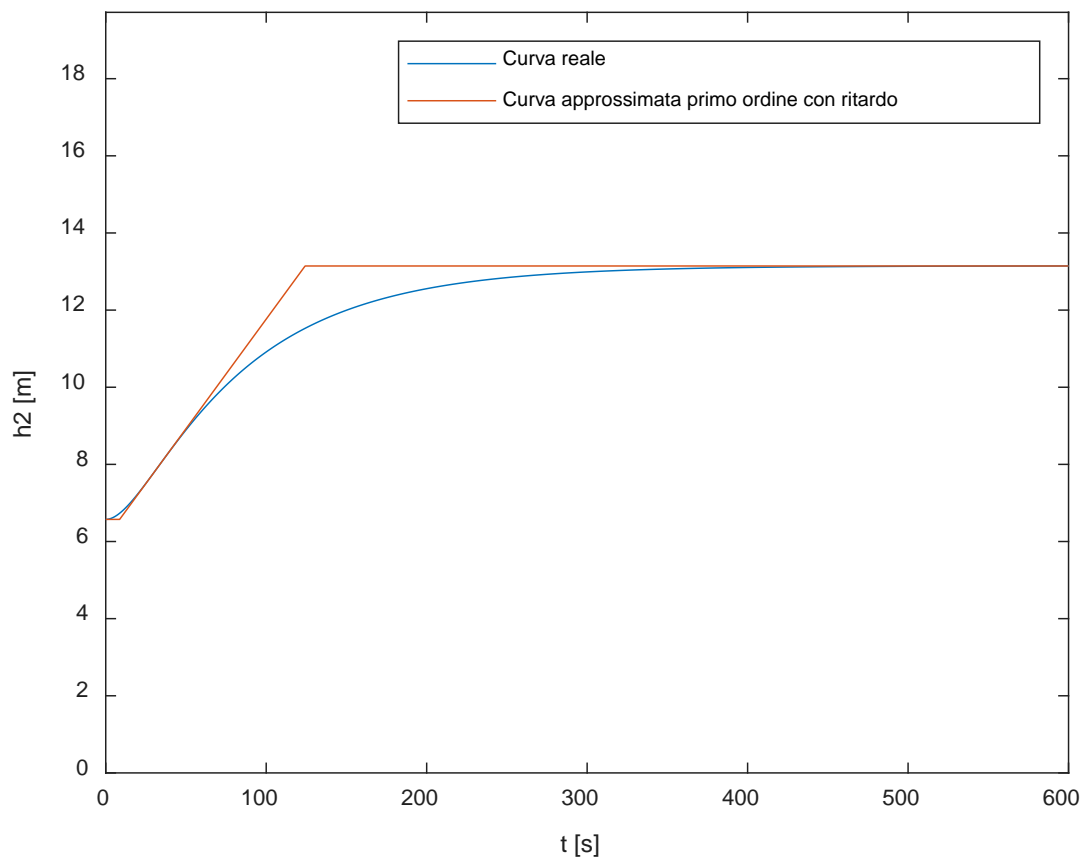
% In alternativa, cercare il massimo della derivata prima
m=max(der1);
pos2=find(der1==m);

B=h2(end)-h2(1); % m
S=der1(pos); % m/s
td=tout(pos)-(h2(pos)-h2(1))/S; % s

K=B/A; % s/m^2
tau=B/S; % s
```

Approssimazione con curva del primo ordine con tempo di ritardo

```
h2approx=zeros(1,length(tout));  
for i=1:length(tout)  
    h=S*(tout(i)-td);  
    h=min(h,B);  
    h=max(h,0);  
    h2approx(i)=h+h2(1);  
end  
  
hold on  
plot(tout,h2approx)  
legend('Curva reale','Curva approssimata primo ordine con ritardo')
```



Calcolo dei parametri dei controllori tramite il metodo di Cohen-Coon

```
% Calcolo parametri controllori

% Controllore P
P_Kc=1/K*tau/td*(1+td/3/tau); % m^2/s

% Controllore PI
PI_Kc=1/K*tau/td*(0.9+td/12/tau); % m^2/s
PI_tauI=td*(30+3*td/tau)/(9+20*td/tau); % s

% Controllore PID
PID_Kc=1/K*tau/td*(4/3+td/4/tau); % m^2/s
PID_tauI=td*(32+6*td/tau)/(13+8*td/tau); % s
PID_tauD=td*4/(11+2*td/tau); % s
```

Controllore P

$K_c=19.558 \text{ m}^2/\text{s}$

Controllore PI

$K_c=17.292 \text{ m}^2/\text{s}$

$\tau_{I1}=24.945 \text{ s}$

Controllore PID

$K_c=25.799 \text{ m}^2/\text{s}$

$\tau_{I1}=20.676 \text{ s}$

$\tau_D=3.109 \text{ s}$

Simulazione del sistema con controllore P

```
A1=30; % m^2
A2=50; % m^2
r1=1.2; % s/m^2

% Steady-state iniziale
Fiss=9.4; % m^3/s
h2ss=Fiss/1.43; % m
h1ss=h2ss+r1*Fiss; % m

% Costante controllore P
Kc=19.558; % m^2/s
bias=Fiss; % m^3/s
h2sp=6.6; % m

opz=odeset('RelTol',1e-12,'AbsTol',1e-12);
[toutP,houtP]=ode45(@(t,h)closedloopP(t,h,A1,A2,r1,Kc,bias,h2sp),[0 600],[h1ss
h2ss],opz);

figure(1)
subplot(1,2,1)
plot(toutP,houtP(:,1))
xlabel('t [s]')
ylabel('h1 [m]')
ylim([0 1.5*h1ss])

subplot(1,2,2)
plot(toutP,houtP(:,2))
ylabel('h2 [m]')
xlabel('t [s]')
ylim([0 1.5*h2ss])

function dhdt=closedloopP(t,h,A1,A2,r1,Kc,bias,h2sp)

if t<200
    Fo=1.43*h(2); % m^3/s
else
    Fo=1.9*h(2); % m^3/s
end
F1=(h(1)-h(2))/r1; % m^3/s

% Controllore P
epsi=h2sp-h(2); % m
Fi=bias+Kc*epsi; % m^3/s

dhdt=zeros(2,1);
dhdt(1)=(Fi-F1)/A1; % m/s
dhdt(2)=(F1-Fo)/A2; % m/s

end
```

Simulazione del sistema con controllore PI

```
A1=30; % m^2
A2=50; % m^2
r1=1.2; % s/m^2

% Steady-state iniziale
Fiss=9.4; % m^3/s
h2ss=Fiss/1.43; % m
h1ss=h2ss+r1*Fiss; % m

% Costante controllore PI
Kc=17.292; % m^2/s
tauI=24.945; % s
bias=Fiss; % m^3/s
h2sp=6.6; % m

opz=odeset('RelTol',1e-12,'AbsTol',1e-
12,'OutputFcn',@(t,h,flag)gestioneintegrale(t,h,flag,h2sp));
[toutPI,houtPI]=ode45(@(t,h)closedloopPI(t,h,A1,A2,r1,Kc,tauI,bias,h2sp),[0
600],[h1ss h2ss],opz);

figure(1)
subplot(1,2,1)
plot(toutPI,houtPI(:,1))
xlabel('t [s]')
ylabel('h1 [m]')
ylim([0 1.5*h1ss])

subplot(1,2,2)
plot(toutPI,houtPI(:,2))
ylabel('h2 [m]')
xlabel('t [s]')
ylim([0 1.5*h2ss])
```



```

function dhdt=closedloopPI(t,h,A1,A2,r1,Kc,tauI,bias,h2sp)

global integrale told epsiold

if t<200
    Fo=1.43*h(2); % m^3/s
else
    Fo=1.9*h(2); % m^3/s
end
F1=(h(1)-h(2))/r1; % m^3/s

% Controllore PI
epsi=h2sp-h(2); % m

if isempty(integrale)
    integralelocale=0;
else
    integralelocale=integrale+(epsi+epsiold)/2*(t-told);
end

Fi=bias+Kc*epsi+Kc/tauI*integralelocale; % m^3/s

dhdt=zeros(2,1);
dhdt(1)=(Fi-F1)/A1; % m/s
dhdt(2)=(F1-Fo)/A2; % m/s

end

```

```

function fine=gestioneintegrale(t,h,flag,h2sp)

global integrale told epsiold

fine=0;

if flag=="init"
    integrale=0;
    told=0;
    epsiold=0;
elseif flag==" "
    for i=1:length(t)
        epsi=h2sp-h(2,i);
        integrale=integrale+(epsi+epsiold)/2*(t(i)-told);
        told=t(i);
        epsiold=epsi;
    end
end

end

```

Simulazione del sistema con controllore PID

```
A1=30; % m^2
A2=50; % m^2
r1=1.2; % s/m^2

% Steady-state iniziale
Fiss=9.4; % m^3/s
h2ss=Fiss/1.43; % m
h1ss=h2ss+r1*Fiss; % m

% Costante controllore PID
Kc=25.799; % m^2/s
tauI=20.676; % s
tauD=3.109; % s
bias=Fiss; % m^3/s
h2sp=6.6; % m

opz=odeset('RelTol',1e-5,'AbsTol',1e-
6,'OutputFcn',@(t,h,flag)gestioneintegrale(t,h,flag,h2sp));
[toutPID,houtPID]=ode45(@(t,h)closedloopPID(t,h,A1,A2,r1,Kc,tauI,tauD,bias,h2sp),[
0 600],[h1ss h2ss],opz);

figure(1)
subplot(1,2,1)
plot(toutPID,houtPID(:,1))
xlabel('t [s]')
ylabel('h1 [m]')
ylim([0 1.5*h1ss])

subplot(1,2,2)
plot(toutPID,houtPID(:,2))
ylabel('h2 [m]')
xlabel('t [s]')
ylim([0 1.5*h2ss])
```

```

function dhdt=closedloopPID(t,h,A1,A2,r1,Kc,tauI,tauD,bias,h2sp)

global integrale told epsiold

if t<200
    Fo=1.43*h(2); % m^3/s
else
    Fo=1.9*h(2); % m^3/s
end
F1=(h(1)-h(2))/r1; % m^3/s

% Controllore PID
epsi=h2sp-h(2); % m

if isempty(integrale) || told==0
    integralelocale=0;
    derivata=0;
else
    integralelocale=integrale+(epsi+epsiold)/2*(t-told);
    derivata=(epsi-epsiold)/(t-told);
end

Fi=bias+Kc*epsi+Kc/tauI*integralelocale+Kc*tauD*derivata; % m^3/s

dhdt=zeros(2,1);
dhdt(1)=(Fi-F1)/A1; % m/s
dhdt(2)=(F1-Fo)/A2; % m/s

end

function fine=gestioneintegrale(t,h,flag,h2sp)

global integrale told epsiold

fine=0;

if flag=="init"
    integrale=0;
    told=0;
    epsiold=0;
elseif flag==" "
    for i=1:length(t)
        epsi=h2sp-h(2,i);
        integrale=integrale+(epsi+epsiold)/2*(t(i)-told);
        told=t(i);
        epsiold=epsi;
    end
end

end
end

```

Confronto performance controllori P, PI, PID

