

Soluzione Ese 09 – Dinamica e Controllo di due serbatoi

Strumentazione e Controllo di Impianti Chimici

Tutor: Giuseppe Pesenti

Soluzione senza 'OutputFcn' (dal codice dell'esercitazione del 01-Giu-2021) – usando var global

```
clear all
close all
clc
global Fi A1 A2 R1 hSP Kc tauI

%serbatoio 1
Fi=9.4; %m3/s
A1=30; %m2
R1=1.2; %s/m2
h01=3; %m
%serbatoio 2
A2=50; %m2
h02=h01;
% impostazioni controllore
hSP=6.6; %m
Kc=30;
tauI=10;

options=odeset('AbsTol',1e-12,'RelTol',1e-10);
tspan=[0 600]; % s
[t,h]=ode23s(@altezza,tspan,[h01 h02 0],options);

figure(1) % il subplot successivo suddivide la figura corrente in 3 righe, 1
colonna
subplot(3,1,1) % selezioniamo la posizione "1" delle tre celle del subplot
plot(t,h(:,1),'LineWidth',1.5)
hold on
plot(t,h(:,2),'LineWidth',1.5)
LivelloSP=zeros(1,length(t));
for i=1:length(t)
    LivelloSP(i)=hSP;
end
hold on
plot(t,LivelloSP,'k--','LineWidth',1.5)
legend('h1','h2','h2_{SP}')

subplot(3,1,2) % selezioniamo la posizione "2" delle tre celle del subplot
plot(t,h(:,3),'LineWidth',1.5)
legend('termine I')
```

```

subplot(3,1,3) % selezioniamo la posizione "3" delle tre celle del subplot
vetFo2=zeros(1,length(t));
for i=2:length(t)
    epsi=h(i,2)-hSP;
    integrale=h(i,3);
    Fo2=Fi+Kc*epsi+Kc/tauI*integrale;
    Fo2=max(0,Fo2); % evitare Fo2 negativi
end
plot(t(2:end),vetFo2(2:end),'LineWidth',1.5) % escludiamo vetFo2 a t=0, cioè
t(1)
hold on
vetFi=zeros(1,length(t));
for i=1:length(t)
    if t(i)<300
        vetFi(i)=9.4;
    else
        vetFi(i)=2*9.4;
    end
end
plot(t,vetFi,'LineWidth',1.5)
legend('Fo2','Fin')

```

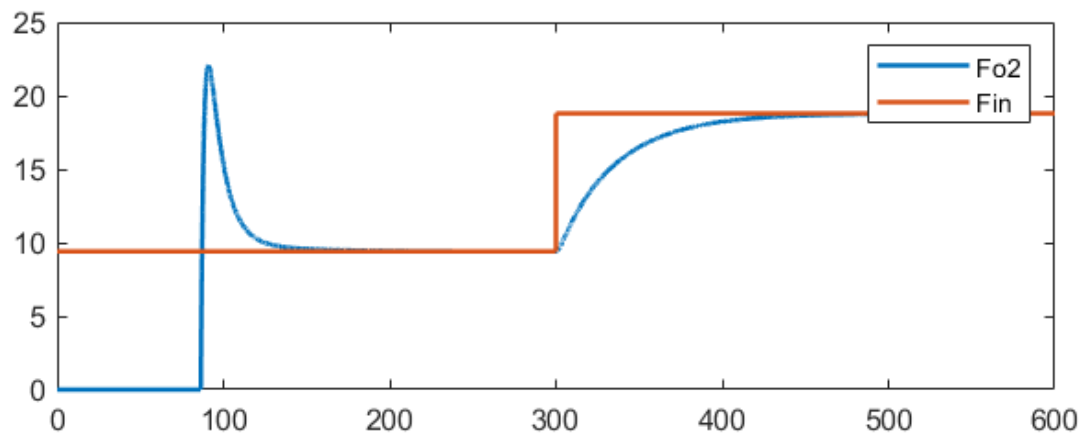
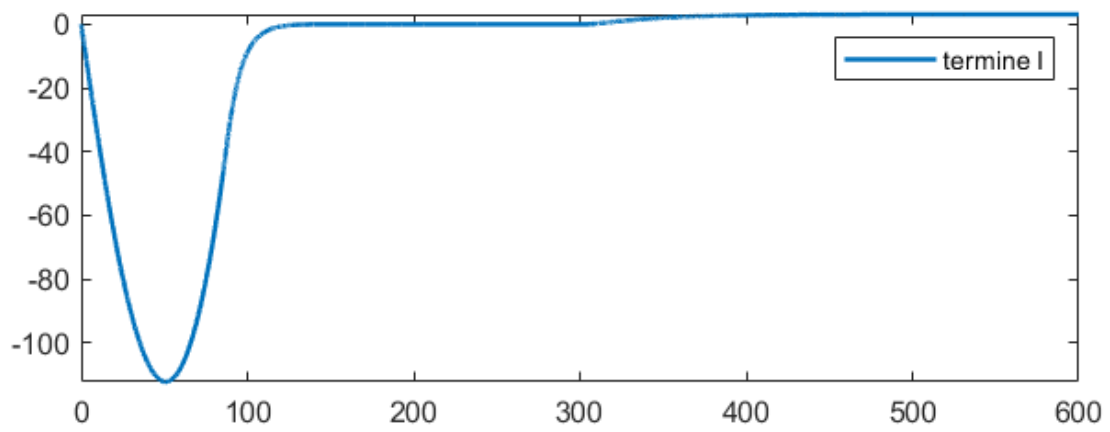
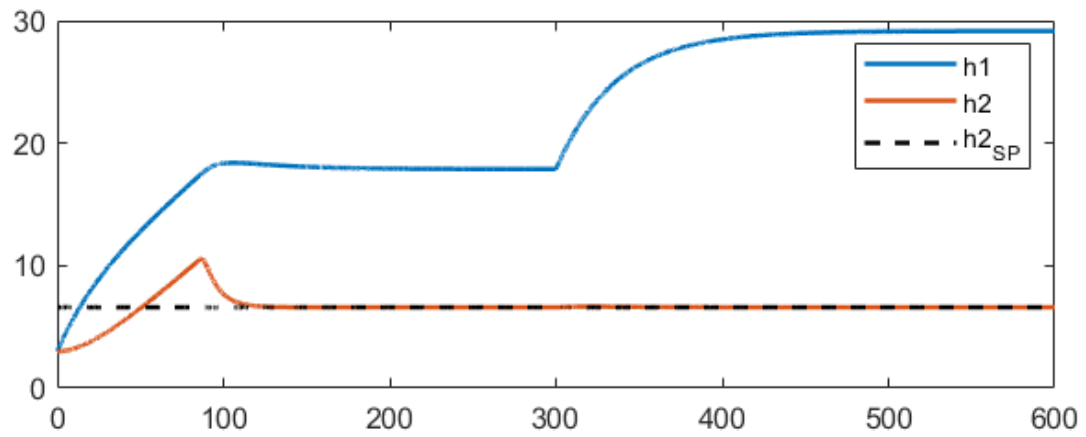
```

function dhdt=altezza(t,h)
global Fi A1 A2 R1 hSP Kc tauI
integrale=h(3);
Fo12=(h(1)-h(2))/R1;
% Fo2=1.43*h(2);
epsi=h(2)-hSP;
bias=Fi;

Fo2=bias+Kc*epsi+Kc/tauI*integrale;
Fo2=max(0,Fo2); % evitare Fo2 negativi

if t<300 %s
    Fin=Fi;
else
    Fin=2*Fi;
end
dhdt=zeros(2,1);
dhdt(1)=(Fin-Fo12)/A1; % m/s
dhdt(2)=(Fo12-Fo2)/A2; % m/s
dhdt(3)=epsi;
end

```



Soluzione con 'OutputFcn' (dal codice dell'esercitazione del 27-Mag-2021) – usando var global

```
clear all
close all
clc
global Fi A1 A2 R1 hSP Kc tauI vetFo2

%serbatoio 1
Fi=9.4; %m3/s
A1=30; %m2
R1=1.2; %s/m2
h01=3; %m
%serbatoio 2
A2=50; %m2
h02=h01;
% impostazioni controllore
hSP=6.6; %m
Kc=30;
tauI=10;

options=odeset('AbsTol',1e-12,'RelTol',1e-10,'OutputFcn',@Printo);
tspan=[0 600]; % s
[t,h]=ode23s(@altezza,tspan,[h01 h02],options);

figure(1)
plot(t,h,'LineWidth',1.5)
ylim([0 50])
xlim([0 600])
% Aggiunta: confronto con il setpoint
LivelloSP=zeros(1,length(t));
for i=1:length(t)
    LivelloSP(i)=hSP;
end
hold on
plot(t,LivelloSP,'k--','LineWidth',1.5)
legend('h1','h2','h2_{SP}')

% Aggiunta: plot di Fo2 nel caso PI, e confronto con Fi
figure(2)
plot(t(2:end),vetFo2,'LineWidth',1.5) % non abbiamo definito vetFo2 a t=0, cioè
t(1)
vetFi=zeros(1,length(t));
for i=1:length(t)
    if t(i)<300
        vetFi(i)=9.4;
    else
        vetFi(i)=2*9.4;
    end
end
hold on
plot(t,vetFi,'LineWidth',1.5)
legend('Fo2','Fin')
```

```

function dhdt=altezza(t,h)
global Fi A1 A2 R1 hSP Kc tauI integrale told epsiold
Fo12=(h(1)-h(2))/R1;
% Fo2=1.43*h(2);
epsi=h(2)-hSP;
bias=Fi;
if isempty(integrale)
    % Il risolutore ode fa qualche chiamata a questa funzione prima di
    % chiamare l'output function con flag 'init': la variabile "integrale"
    % perciò non è ancora stata inizializzata a zero ma è [] (cioè empty).
    % Gestiamo questo caso come un'eccezione:
    integralelocale=0;
else
    % integralelocale=integrale;
    integralelocale=integrale+(epsiold+epsi)/2*(t-told);
end
Fo2=bias+Kc*epsi+Kc/tauI*integralelocale;
Fo2=max(0,Fo2); % evitare Fo2 negativi

if t<300 %s
    Fin=Fi;
else
    Fin=2*Fi;
end
dhdt=zeros(2,1);
dhdt(1)=(Fin-Fo12)/A1; % m/s
dhdt(2)=(Fo12-Fo2)/A2; % m/s

end

function finire=Printo(tstep,hstep,flag)
finire=0; % non vogliamo che l'outputfcn termini l'integrazione

global hSP epsiold told integrale Fi Kc tauI vetFo2

if strcmp(flag,'init')
    integrale=0;
    told=0;
    epsiold=0;

    % Aggiunta: salviamo i valori di Fo2
    vetFo2=[]; % (non facciamo nessuna ipotesi sul valore iniziale a t=0)

elseif strcmp(flag,'')
    epsi=hstep(2)-hSP;

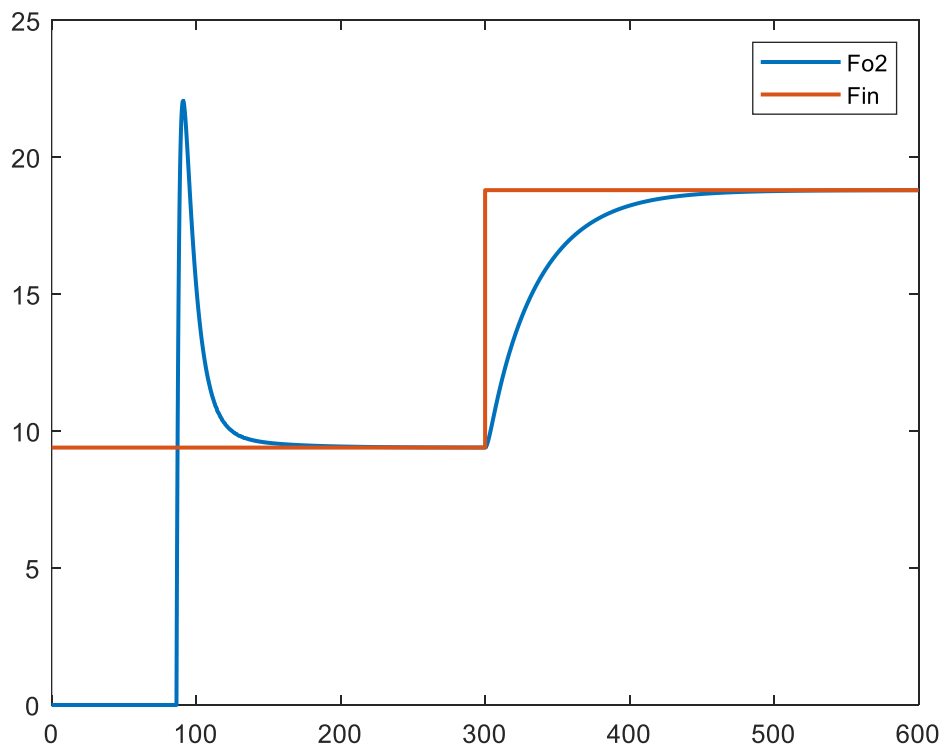
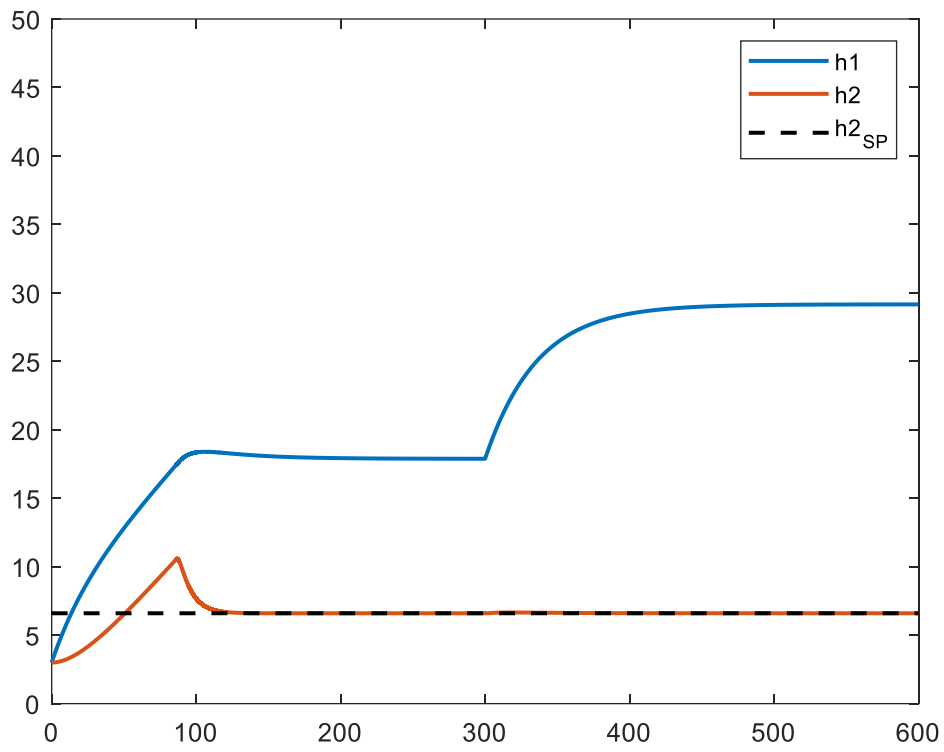
    integrale=integrale+(epsiold+epsi)/2*(tstep-told);
    epsiold=epsi;
    told=tstep;

    % Aggiunta: salviamo i valori di Fo2
    bias=Fi;
    Fo2=bias+Kc*epsi+Kc/tauI*integrale;
    Fo2=max(0,Fo2); % evitare Fo2 negativi
    vetFo2=[vetFo2,Fo2];

elseif strcmp(flag,'done')
end

end

```



Soluzione con 'OutputFcn' e confronto open-loop, P, PI – senza usare var global

```
clear all

Aserb1=30; % m^2
Aserb2=50; % m^2
r1=1.2; % s/m^2
h_0=[3;3]; % m

Fi_0=9.4; % m^3/s
h2_sp=6.6; % m
F0bias=Fi_0; % m^3/s
Kc=30; % m^2/s
tauI=10; % s

handleopen=@(t,h)serbatoi_openloop(t,h,r1,Aserb1,Aserb2,Fi_0);
[tout,hout]=ode45(handleopen,[0 10*60],h_0,odeset('RelTol',1e-8));
figure(1)
subplot(1,3,1)
plot(tout,hout)

handleclosed=@(t,h)serbatoi_closedloop(t,h,r1,Aserb1,Aserb2,Fi_0,F0bias,Kc,h2_sp);
[tout2,hout2]=ode45(handleclosed,[0 10*60],h_0,odeset('RelTol',1e-8));
subplot(1,3,2)
plot(tout2,hout2)
hold on
plot(tout2,ones(1,length(tout2))*h2_sp,'k')
legend('h1','h2','h2 sp')
xlabel('t [s]')
title('Altezza del livello di liquido [m] - Closed loop P')

handleOutputPI=@(t,h,flag)outputPI(t,h,flag,h2_sp);
opzPI=odeset('RelTol',1e-8,'OutputFcn',handleOutputPI,'Refine',1);
handlePI=@(t,h)serbatoi_PI(t,h,r1,Aserb1,Aserb2,Fi_0,F0bias,Kc,h2_sp,tauI);
[tout3,hout3]=ode45(handlePI,[0 10*60],h_0,opzPI);
subplot(1,3,3)
plot(tout3,hout3)
hold on
plot(tout3,ones(1,length(tout3))*h2_sp,'k')
legend('h1','h2','h2 sp')
xlabel('t [s]')
title('Altezza del livello di liquido [m] - Closed loop PI')
```

```
function dhdt=serbatoi_openloop(t,h,r1,Aserb1,Aserb2,Fi_0)
```

```
if t<5*60
```

```
    Fi=Fi_0; % m^3/s
```

```
else
```

```
    Fi=2*Fi_0; % m^3/s
```

```
end
```

```
F1=(h(1)-h(2))/r1; % m^3/s
```

```
F0=1.43*h(2); % m^3/s
```

```
dhdt=zeros(2,1);
```

```
dhdt(1)=1/Aserb1*(Fi-F1); % m/s
```

```
dhdt(2)=1/Aserb2*(F1-F0); % m/s
```

```
end
```

```
function dhdt=serbatoi_closedloop(t,h,r1,Aserb1,Aserb2,Fi_0,F0bias,Kc,h2_sp)
```

```
if t<5*60
```

```
    Fi=Fi_0; % m^3/s
```

```
else
```

```
    Fi=2*Fi_0; % m^3/s
```

```
end
```

```
F1=(h(1)-h(2))/r1; % m^3/s
```

```
epsi=h(2)-h2_sp; % m
```

```
F0=F0bias+Kc*epsi; % m^3/s
```

```
F0=max(F0,0);
```

```
dhdt=zeros(2,1);
```

```
dhdt(1)=1/Aserb1*(Fi-F1); % m/s
```

```
dhdt(2)=1/Aserb2*(F1-F0); % m/s
```

```
end
```



```

function dhdt=serbatoi_PI(t,h,r1,Aserb1,Aserb2,Fi_0,F0bias,Kc,h2_sp,tauI)
global integrale told

if t<5*60
    Fi=Fi_0; % m^3/s
else
    Fi=2*Fi_0; % m^3/s
end

F1=(h(1)-h(2))/r1; % m^3/s

epsi=h(2)-h2_sp; % m

if isempty(integrale)
    integraleusaetta=0;
else
    integraleusaetta=integrale+epsi*(t-told);
end

F0=F0bias+Kc*epsi+Kc/tauI*integraleusaetta; % m^3/s

F0=max(F0,0);

dhdt=zeros(2,1);
dhdt(1)=1/Aserb1*(Fi-F1); % m/s
dhdt(2)=1/Aserb2*(F1-F0); % m/s

end

```

```

function finire = outputPI(tstep,hstep,flag,h2sp)
global integrale told epsiold

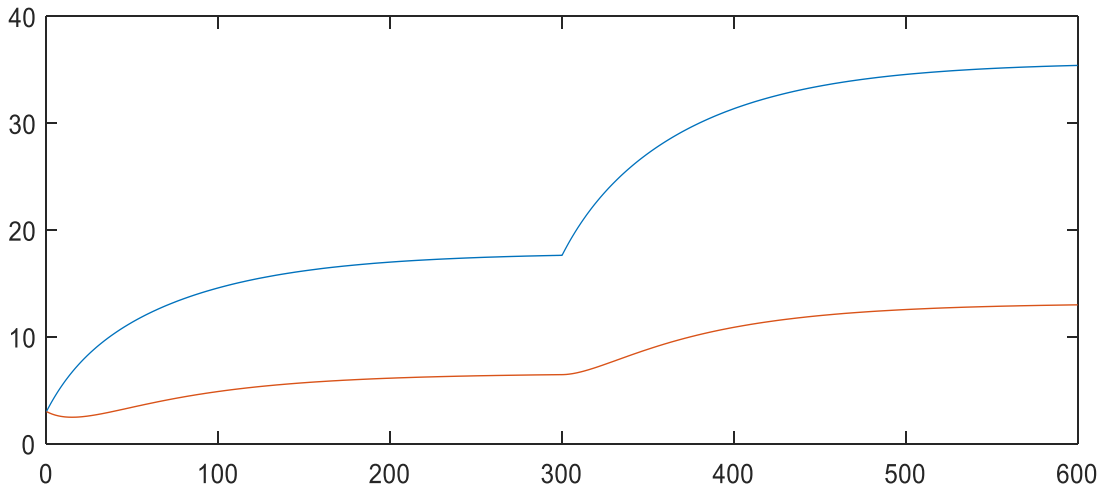
finire=0;

if strcmp(flag,'init')
    integrale=0;
    told=0;
    epsiold=0;
elseif strcmp(flag,'')
    deltat=tstep-told;
    epsistep=hstep(2)-h2sp;
    integrale=integrale+(epsistep+epsiold)/2*deltat;

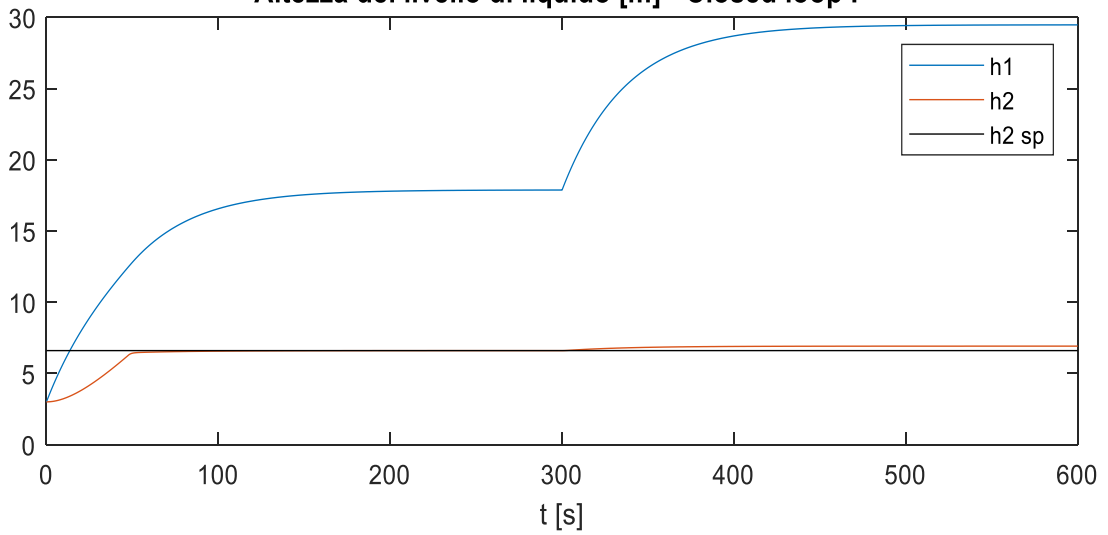
    told=tstep;
    epsiold=epsistep;
elseif strcmp(flag,'done')
    disp('')
end

end

```



Altezza del livello di liquido [m] - Closed loop P



Altezza del livello di liquido [m] - Closed loop PI

