

# Automatica (Laboratorio)

Corso di Laurea in Ingegneria Elettronica IV anno (V.O.)  
Dipartimento di Ingegneria dell'Università di Ferrara. A.A. 2001/2002

## Relazioni di Laboratorio

di

Tarin Gamberini

## Indice delle esercitazioni

1. Matrice di Hankel	pag. 3
2. Raggiungibilità ed Osservabilità	pag. 5
3. Lotka-Volterra	pag. 11
4. Sistema ibrido	pag. 19
5. Motore in corrente continua	pag. 23
6. Osservatori e retroazione uscita-stato-ingresso	pag. 29
7. Controllo di un motore in corrente continua	pag. 35
8. Sintonizzazione di controllori PID	pag. 41

# Matrice di Hankel

## Testo dell'esperienza

1. Scrivere la funzione  $H = \text{my\_hankel}(X, \text{NrH}, \text{NcH}, \text{shift})$ , in cui  $X$  è un vettore di  $L$  elementi,  $\text{NrH}$  è il numero di righe di  $H$ ,  $\text{NcH}$  è il numero di colonne di  $H$ , e  $\text{shift}$  è un intero maggiore od uguale a 0. La matrice  $H$  deve essere costruita in modo tale che:

$$H = \begin{bmatrix} X(1 + \text{shift}) & \cdots & X(\text{shift} + \text{NcH}) \\ \vdots & \ddots & \vdots \\ X(\text{shift} + \text{NrH}) & \cdots & X(\text{shift} + \text{NcH} + \text{NrH} - 1) \end{bmatrix}$$

con l'ipotesi che  $L = \text{shift} + \text{NcH} + \text{NrH} - 1$ .

## Codice dei sorgenti

Nel file `Vriga.mat` abbiamo salvato il vettore  $W=[1, 2, 3, 4, 5, 6, 7, 8]$  utilizzato come esempio.

Il file `my_hankel.m` è un function file che ricevuto in ingresso i parametri specificati nel testo genera in uscita la matrice  $H$ . La funzione controlla che la dimensione del vettore riga  $X$  verifichi l'ipotesi richiesta  $L = \text{shift} + \text{NcH} + \text{NrH} - 1$  e successivamente, tramite due for nidificati, genera la matrice  $H$ .

```
function H=my_hankel(X,nrH,ncH,shift);

% Dimensione del vettore riga X
L=size(X,2);

% Verifica dell'ipotesi L>=nrH+ncH-1
if (nrH+ncH-1>L),
    error('attenzione: parametri non validi')
end

% Costruzione della matrice H
for r=0 : nrH-1,
    for c=1 : ncH,
        H(r+1,c)= X(shift+c+r);
    end
end

return
```

## Simulazioni

Riportiamo di seguito alcune chiamate alla funzione `my_hankel`:

```
>> load Vriga
>> W
```

```
W =  
1      2      3      4      5      6      7      8
```

```
>> my_hankel(W,4,4,0)
```

```
ans =
```

```
1      2      3      4  
2      3      4      5  
3      4      5      6  
4      5      6      7
```

```
>> my_hankel(W,4,4,1)
```

```
ans =
```

```
2      3      4      5  
3      4      5      6  
4      5      6      7  
5      6      7      8
```

```
>> my_hankel(W,2,4,1)
```

```
ans =
```

```
2      3      4      5  
3      4      5      6
```

```
>> my_hankel(W,4,2,1)
```

```
ans =
```

```
2      3  
3      4  
4      5  
5      6
```

```
>>
```

# Raggiungibilità ed Osservabilità

## Testo dell'esperienza

1. Scrivere un programma che, date le matrici  $A_{n \times n}$  e  $B_{n \times r}$ , costruisca la matrice  $P = [B, A*B, \dots, A^{n-1}*B]$ . Successivamente effettuare il test del rango.
2. Scrivere un programma che, date le matrici  $A_{n \times n}$  e  $C_{m \times n}$ , costruisca la matrice  $Q = [C^T, A^T * C^T, \dots, A^{n-1} * C^T]^T$ . Successivamente effettuare il test del rango.
3. Data la terna  $(A_{n \times n}, B_{n \times 1}, C_{1 \times n})$ , eseguire un cambiamento di base per determinare la parte raggiungibile (controllabile) del sistema. Successivamente determinare la forma minima.
4. Data la terna  $(A_{n \times n}, B_{n \times 1}, C_{1 \times n})$ , calcolare la matrice  $P = [B, A*B, \dots, A^{n-1}*B]$ . Successivamente calcolare le matrici  $T1 = \text{im}(P)$  e  $T2$ , con  $T2$  tale che  $T = [T1, T2]$  sia quadrata e invertibile. Si esegua la trasformazione  $A_c = \text{inv}(T)*A*T$ ,  $B_c = \text{inv}(T)*B$  e  $C_c = C*T$ . Infine, detto  $n_c$  il numero di colonne di  $T1$ , estrarre le matrici  $A_{c1}$ , avente le prime  $n_c$  righe e colonne di  $A_c$ ,  $B_{c1}$  dalle prime  $n_c$  righe di  $B_c$  e  $C_{c1}$ , le prime  $n_c$  colonne di  $C_c$ . In maniera analoga, calcolare la matrice  $Q = [C^T, A^T * C^T, \dots, A^{n-1} * C^T]^T$  ed effettuare la trasformazione  $T$  ricavata, come in precedenza, dall'immagine di  $Q'$  e dal suo complemento ortogonale. Si estraggano quindi le matrici  $(A_o, B_o, C_o)$ .  
Si esegua l'esercizio con le matrici seguenti:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & -2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad C = [0 \ 0 \ 0 \ 0 \ 1]$$

## Codice dei sorgenti

Nel file ABC.mat abbiamo salvato le matrici A, B e C indicate.

Il file my\_ctrb.m è un function file che ricevuto in ingresso le matrici A e B genera in uscita la matrice  $P = [B, A*B, \dots, A^{n-1}*B]$  necessaria per determinare la parte raggiungibile e controllabile del sistema.

```
function P=my_ctrb(A,B)

% Righe di A
n=size(A,1);

P=B;
for index=1 : n-1,
    P=[B,A*P];
end

return
```

Il file my\_obsv.m è un function file che ricevuto in ingresso le matrici A e C genera in uscita la matrice  $Q = [C^T, A^T * C^T, \dots, A^{n-1} * C^T]^T$  necessaria per determinare la parte osservabile e ricostruibile del sistema.

```
function Q=my_obsv(A,C)
```

```

% Righe di A
n=size(A,1);

Q=C';
for index=1 : n-1,
    Q=[Q,(A')^index*C'];
end
Q=Q';

return

```

Il file RCOR.m è uno script file che richiamando my\_ctrb.m e my\_obsv.m calcola la parte raggiungibile e controllabile, osservabile e ricostruibile del sistema specificato dalle matrici A, B e C.

```

% Data la terna (An*n,Bn*1,C1*n), determiniamo la parte raggiungibile e
% controllabile e la forma minima del sistema

% A è di dimensione nA x nA
nA=size(A,1);
% Colonne di B
r=size(B,2);
% Righe di C
m=size(C,1);

% Costruisco P=[ B, A*B, ... , A^(n-1) * B ]
P=my_ctrb(A,B)
% Numero di colonne linearmente indipendenti di P
ncP=rank(P);
% Righe di P
nrP=size(P,1);
% Test di rango
if (ncP ~= nA)
    disp('Il sistema non è completamente raggiungibile e controllabile')
end

% Costruisco T per l'estrazione della parte raggiungibile e controllabile
T=orth([P,eye(nA)])
% Calcolo la parte raggiungibile e controllabile
Ac=inv(T)*A*T
Bc=inv(T)*B
Cc=C*T
% Visualizzo
disp('La parte raggiungibile e controllabile è:')
Ac1=Ac(1:ncP,1:ncP)
Bc1=Bc(1:ncP,1:r)
Cc1=Cc(1:m,1:ncP)

% Costruisco Q=[ C', A'*C', ... , A^(n-1) * C' ]
Q=my_obsv(A,C)
% Numero di colonne linearmente indipendenti di Q'
ncQt=rank(Q');
% Test di rango
if (ncQt ~= nA)
    disp('Il sistema non è completamente osservabile e ricostruibile')
end

```

```
% Costruisco TT per l'estrazione della parte osservabile e ricostruibile
TT=orth([Q',eye(nA)])
% Calcolo la forma minima
Ao=inv(TT)*A*TT
Bo=inv(TT)*B
Co=C*TT
% Visualizzo
disp('La forma minima è:')
Ao1=Ao(1:ncQt,1:ncQt)
Bo1=Bo(1:ncQt,1:r)
Co=Co(1:m,1:ncQt)

end
```

## Simulazioni

Riportiamo di seguito alcune chiamate alla funzione.

```
>> load ABC
>> A
```

A =

```
1    0    0    0    0
2   -2    1    0    0
3    0    1    0    0
0    0    0    1    0
0    0    0    2   -1
```

```
>> B
```

B =

```
0
0
0
1
0
```

```
>> C
```

C =

```
0    0    0    0    1
```

```
>> RCOR
```

P =

```
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
1    1    1    1    1
```

0     2     0     2     0

Il sistema non è completamente raggiungibile e controllabile

T =

0	0	1.0000	0	0	
0	0	0	1.0000	0	
0	0	0	0	1.0000	
-0.5696	-0.8219		0	0	0
-0.8219	0.5696		0	0	0

Ac =

0.5852	2.2875	0	0	0
0.2875	-0.5852	0	0	0
0	0	1.0000	0	0
0	0	2.0000	-2.0000	1.0000
0	0	3.0000	0	1.0000

Bc =

-0.5696  
-0.8219  
0  
0  
0

Cc =

-0.8219     0.5696     0     0     0

La parte raggiungibile e controllabile è:

Ac1 =

0.5852     2.2875  
0.2875     -0.5852

Bc1 =

-0.5696  
-0.8219

Cc1 =

-0.8219     0.5696

Q =

0     0     0     0     1



```

0    0    0    2    -1
0    0    0    0    1
0    0    0    2    -1
0    0    0    0    1

```

Il sistema non è completamente osservabile e ricostruibile

TT =

```

0      0    1.0000      0      0
0      0      0    1.0000      0
0      0      0      0    1.0000
-0.8219  0.5696      0      0      0
0.5696  0.8219      0      0      0

```

Ao =

```

-0.5852  -0.2875      0      0      0
-2.2875   0.5852      0      0      0
0      0    1.0000      0      0
0      0    2.0000  -2.0000  1.0000
0      0    3.0000      0    1.0000

```

Bo =

```

-0.8219
0.5696
0
0
0

```

Co =

```

0.5696  0.8219      0      0      0

```

La forma minima è:

Ao1 =

```

-0.5852  -0.2875
-2.2875   0.5852

```

Bo1 =

```

-0.8219
0.5696

```

Co =

```

0.5696  0.8219

```



# Lotka-Volterra

## Testo dell'esperienza

Si consideri il modello matematico di Lotka-Volterra che descrive la dinamica di due popolazioni interagenti:

$$\dot{x}_1(t) = a_1 \left( 1 - \frac{x_1(t)}{k} \right) x_1(t) - a_2 x_1(t) x_2(t) + u(t) \quad (\text{prede})$$

$$\dot{x}_2(t) = -a_3 x_2(t) + a_4 x_1(t) x_2(t) \quad (\text{predatori})$$

dove  $x_1(t)$  e  $x_2(t)$  rappresentano rispettivamente il numero di prede e il numero di predatori presenti nell'ecosistema all'istante  $t$  ed  $u(t)$  l'apporto esterno di cibo per le prede introdotto nell'unità di tempo. Il coefficiente  $k$  rappresenta il numero massimo di prede presenti nell'ecosistema in assenza di predatori e in assenza di apporto di cibo esterno ( $u(t) = 0$ ). Il parametro  $a_3$  ( $> 0$ ) è il tasso di crescita del predatore, in assenza di prede, mentre  $a_1$  ( $> 0$ ) il tasso di crescita delle prede. Il termine  $-a_2 x_1(t) x_2(t)$  modella il decremento nella popolazione delle prede per la presenza dei predatori, con  $a_2 > 0$ , mentre il termine  $a_4 x_1(t) x_2(t)$  rappresenta l'incremento della popolazione dei predatori dovuto alla presenza delle prede.

Nelle ipotesi di assegnare ai parametri i valori  $a_1 = 20$ ,  $a_2 = 1$ ,  $a_3 = 7$ ,  $a_4 = 0.5$  e  $k = 30$ , si determinino:

1. L'andamento nel tempo del numero di prede e predatori, supponendo nullo l'ingresso  $u(t)$  e nelle ipotesi di partire da un ecosistema contenente 10 prede e 10 predatori. Si calcoli anche la traiettoria percorsa dal sistema nello spazio degli stati.
2. Gli stati di equilibrio del sistema in assenza di ingresso.
3. I valori di regime raggiunti dal numero di prede e predatori nelle ipotesi che  $u(t)$  sia un gradino di ampiezza  $u(t) = 20$  e a partire dalle stesse condizioni proposte al punto 1). Si determini per tentativi l'ampiezza del gradino che consente di mantenere a regime un numero di predatori pari a 15.

## Quesito 1.

Nel function file `ecosistema.m` abbiamo salvato la descrizione del modello matematico nello spazio degli stati che verrà poi utilizzato dall'`ODEsolver`.

```
function xd=ecosistema(t,x,flag,param)
% Modello matematico di Lotka - Volterra di un ecosistema preda predatore

a1=param(1);          % >0 Tasso di crescita delle prede
a2=param(2);          % >0 Tasso di decremento prede a causa dei predatori
a3=param(3);          % >0 Tasso di crescita dei predatori
a4=param(4);          % >0 Tasso crescita di predatori dovuto alle prede
k =param(5);          % >0 Max n. predatori senza prede e cibo esterno
tStart=param(6);      % istante iniziale di applicazione del gradino
tStop =param(7);      % istante finale di applicazione dell'ingresso
amplitude=param(8);  % ampiezza dell'ingresso per tStart < t < tStop

% Disattiva l'ingresso a gradino per t < tStart o t > tStop
if ((t>=tStart) & (t<=tStop))
    udt=amplitude;
else
```

```

    udt=0.0;
end

% Modello matematico
x1d=a1*(1-x(1)/k)*x(1)-a2*x(1)*x(2)+udt; % prede
x2d=-a3*x(2)+a4*x(1)*x(2);           % predatori

% Setto il vettore di stato di ritorno della funzione
% Gli ODEsolver lo vogliono rigorosamente colonna
xd=[x1d ; x2d];

return

```

Lo script file Lotka\_Volterra\_libero.m setta i parametri da passare all'ODEsolver e grafica la traiettoria ed il moto risultanti nell'intervallo di tempo analizzato.

```

% Script file per l'integrazione del sistema differenziale Lotka -
Volterra
% prede predatori, descritto nel function file ecosistema.m, graficandone
% i risultati

% Definizione dell'accuratezza
options=odeset('RelTol',1e-6);

% Parametri fisici del modello
a1=20;      % >0 Tasso di crescita delle prede
a2=1;      % >0 Tasso di decremento prede a causa dei predatori
a3=7;      % >0 Tasso di crescita dei predatori
a4=0.5;    % >0 Tasso crescita dei predatori dovuto alle prede
k =30;     % >0 Max n. predatori in assenza di prede e cibo esterno
tStart=0;  % istante iniziale di applicazione del gradino
tStop =1;  % istante finale di applicazione del gradino
amplitude=0; % ampiezza gradino per tStart < t < tStop

param=[a1,a2,a3,a4,k,tStart,tStop,amplitude];

% Intervallo di integrazione
t_evoluzione=[0,5];

% Stato iniziale Xo[1,1] sono prede, Xo[1,2] sono predatori
xo=[10,10];

% Integra
[t,x]=ode45('ecosistema',t_evoluzione,xo,options,param)

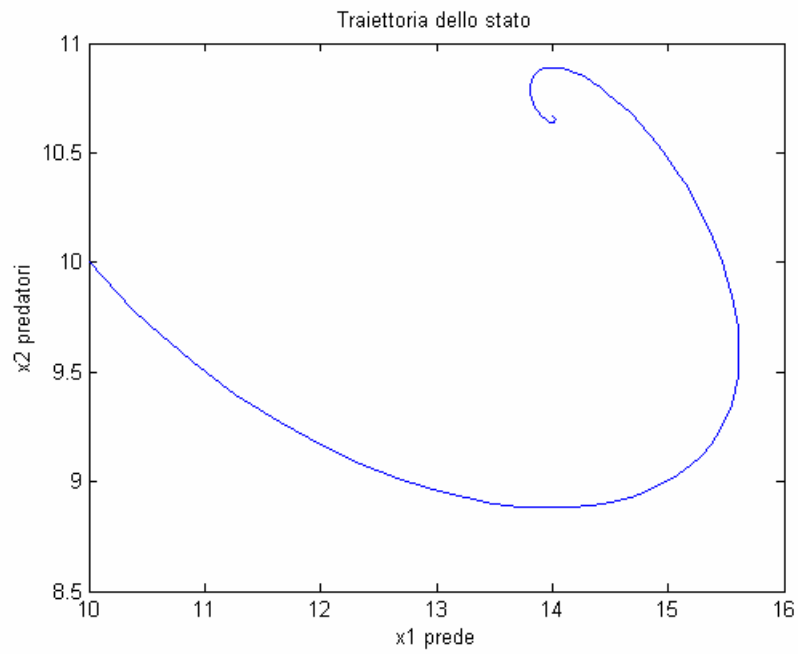
% Grafico del moto dello stato
figure
plot(t,x(:,1),'-',t,x(:,2),'--')
title('Moto degli stati');
xlabel('tempo')
ylabel('x1 prede -          x2 predatori --')

% Grafico della traiettoria dello stato
figure
plot(x(:,1),x(:,2),'-')
title('Traiettoria dello stato');
xlabel('x1 prede')

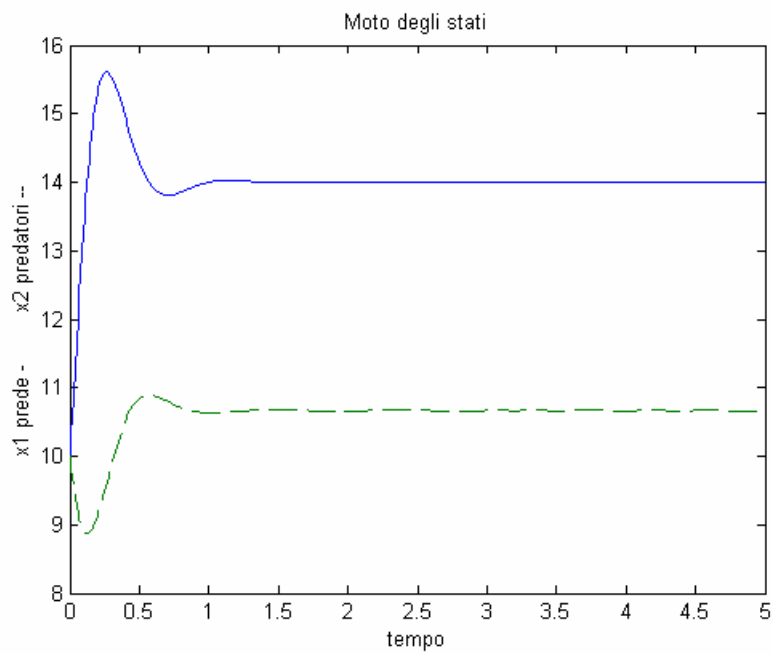
```

`ylabel('x2 predatori')`

Il risultato dell'integrazione è riportato in Figura 1: Traiettorie dello stato e Figura 2:



**Figura 1: Traiettorie dello stato**



**Figura 2: Moto libero**

**Quesito 2.**

Gli stati di equilibrio del sistema si calcolano risolvendo:

$$\frac{dx}{dt} = 0$$

L'equazione relativa ai predatori:

$$-a_3x_2 + a_4x_1x_2 = 0 \qquad x_1 = \frac{a_3}{a_4} = 14$$

Sostituendo nell'equazione delle prede:

$$a_1\left(1 - \frac{x_1}{k}\right)x_1 - a_2x_1x_2 = 0 \qquad x_2 = \frac{a_1\left(1 - \frac{x_1}{k}\right)}{a_2} = 10.6667$$

Risultati riscontrabili dai grafici precedenti.

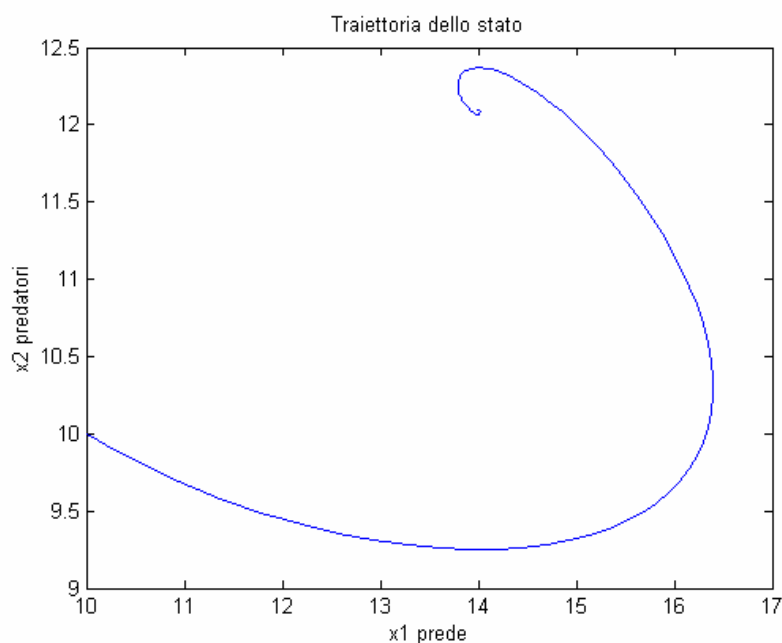
**Quesito 3.**

Si tratta di determinare il moto forzato con ingresso un gradino di ampiezza 20. Modificato il sorgente con:

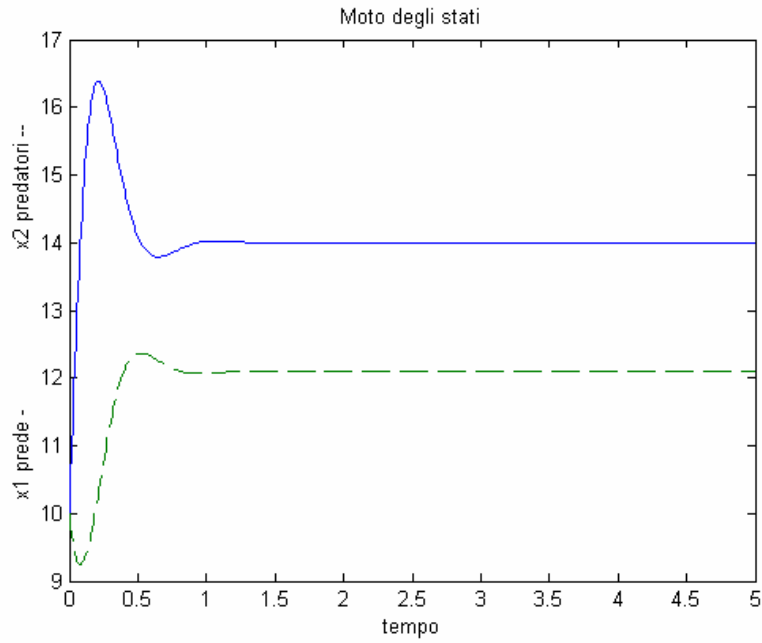
```
tStart=0;      % istante iniziale di applicazione del gradino
tStop =5;     % istante finale di applicazione del gradino
amplitude=20; % ampiezza gradino per tStart < t < tStop

% Intervallo di integrazione
t_evoluzione=[0,5];
```

otteniamo i risultati di Figura 3 e Figura 4:

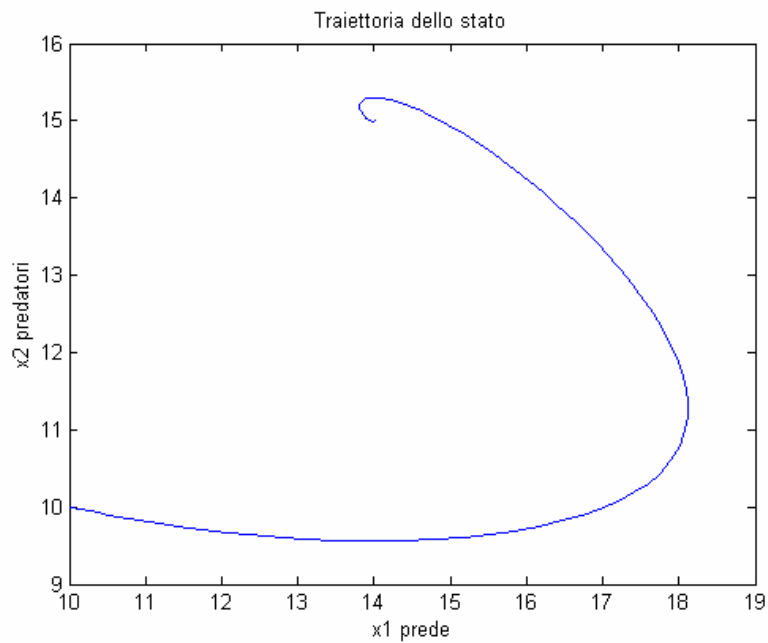


**Figura 3: Traiettoria dello stato con  $u(t)=20$**

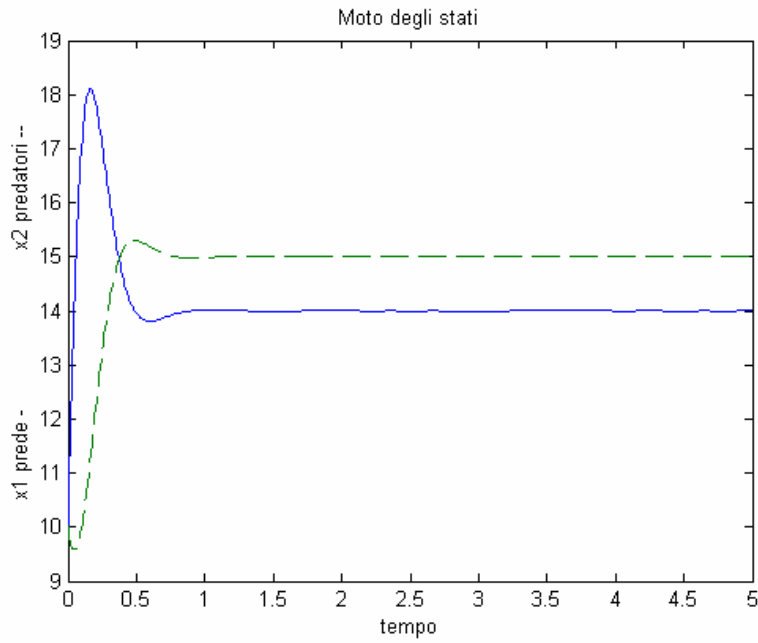


**Figura 4: Moto forzato con  $u(t)=20$**

Infine procedendo per tentativi abbiamo determinato l'ampiezza del gradino necessaria a mantenere a regime un numero di predatori pari a 15. Applicando infatti  $u(t)= 60.667$  otteniamo gli andamenti di Figura 5 e Figura 6:



**Figura 5: Traiettoria con  $u(t)=60.667$**



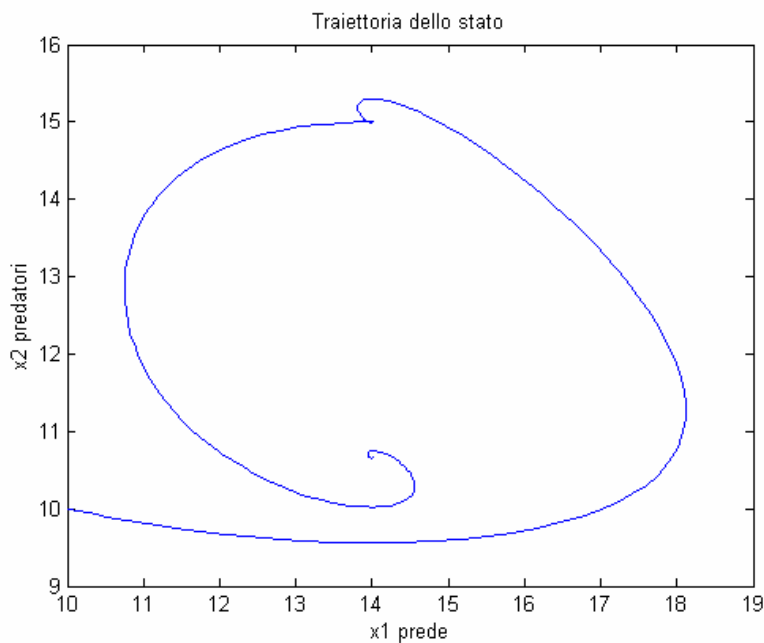
**Figura 6: Moto forzato con  $u(t)=60.667$**

Abbiamo infine effettuato una simulazione di moto forzato con  $u(t)=60.667$  nell'intervallo  $[0,5]$  lasciando poi il sistema libero nell'intervallo  $[5,10]$ :

```
tStart=0;           % istante iniziale di applicazione del gradino
tStop =5;           % istante finale di applicazione del gradino
amplitude=60.667; % ampiezza gradino per tStart < t < tStop

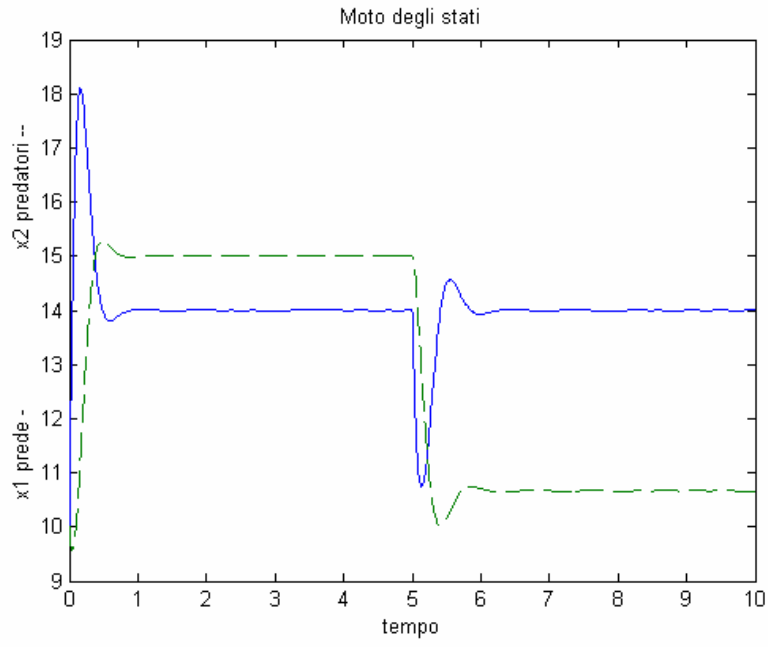
% Intervallo di integrazione
tevoluzione=[0,10];
```

ottenendo gli andamenti Figura 7 e Figura 8:



**Figura 7:  $u(t)=60.667$  in  $[0,5]$  poi  $u(t)=0$  in  $[5,10]$**





**Figura 8: Moto forzato con  $u(t)=60.667$  in  $[0,5]$  e moto libero con  $u(t)=0$  in  $[5,10]$**



# Sistema ibrido

## Testo dell'esperienza

Si definisce modello ibrido un sistema composto da diversi modelli, ciascuno valido in una particolare condizione di funzionamento del sistema stesso. Tale condizione dipende dallo stato del sistema e dai suoi ingressi.

Si consideri quindi il sistema ibrido descritto dal seguente modello matematico:

$$\dot{x}(t) = \begin{cases} A1x(t) & \text{se } x1(t) * x2(t) < 0 \\ A2x(t) & \text{se } x1(t) * x2(t) \geq 0 \end{cases}$$

in cui :

$$x(t) = \begin{bmatrix} x1(t) \\ x2(t) \end{bmatrix} \quad A1 = \begin{bmatrix} -0.1 & 1.0 \\ -10.0 & -0.1 \end{bmatrix} \quad A2 = \begin{bmatrix} -0.1 & 10 \\ -1.0 & -0.1 \end{bmatrix}$$

1. Si discuta la stabilità dei sistemi singoli.
2. Si disegnino le traiettorie dello stato per i sistemi singoli e per il sistema completo considerando un tempo di simulazione di 10s e le condizioni iniziali (1, 0), (0, 1) e (10<sup>-6</sup>, 10<sup>-6</sup>).
3. Si disegni l'andamento nel tempo delle variabili di stato dei sistemi singoli e di quello ibrido nelle stesse condizioni.

## Quesito 1.

Gli autovalori della matrice di transizione di stato A1 del primo sistema singolo sono:

$$\text{eig}(A1) \quad -0.1000 + 3.1623i \quad -0.1000 - 3.1623i$$

Poiché hanno parte reale negativa il primo sistema è asintoticamente stabile.

Gli autovalori della matrice di transizione di stato A2 del secondo sistema singolo sono:

$$\text{eig}(A2) \quad -0.1000 + 3.1623i \quad -0.1000 - 3.1623i$$

Poiché hanno parte reale negativa il secondo sistema è asintoticamente stabile.

## Quesito 2.

Riportiamo per brevità solo i sorgenti del sistema ibrido completo, quelli dei sistemi singoli sono facilmente ricavabili da esso.

Nel function file ibrido.m abbiamo salvato la descrizione del modello matematico nello spazio degli stati che verrà poi utilizzato dall'ODEsolver.

```
function xd=ibrido(t,x,flag,a1,a2)
% Funzione che modella un sistema ibrido
```

```

% Modello matematico
if (x(1)*x(2)<0)
    xd=a1*[x(1);x(2)];
else
    xd=a2*[x(1);x(2)];
end

return

```

Lo script file ibrido\_libero.m setta i parametri da passare all'ODEsolver e grafica la traiettoria ed il moto risultanti nell'intervallo di tempo analizzato.

Abbiamo effettuato tre simulazioni di moto libero a partire dagli stati iniziali  $(x_1(t), x_2(t))$  pari a  $(1,0)$ ,  $(0,1)$ ,  $(1e-6, 1e-6)$ .

```

% Script file per l'integrazione del sistema differenziale ibrido

```

```

% Definizione dell'accuratezza
options=odeset('RelTol',1e-6);

% Parametri fisici del modello
a1=[-0.1,1.0;-10.0,-0.1];
a2=[-0.1,10.0;-1.0,-0.1];

% Intervallo di integrazione
tevoluzione=[0,10];

% Stato iniziale
%xo=[1,0];           prima simulazione
%xo=[0,1];           seconda simulazione
xo=[1e-6,1e-6];

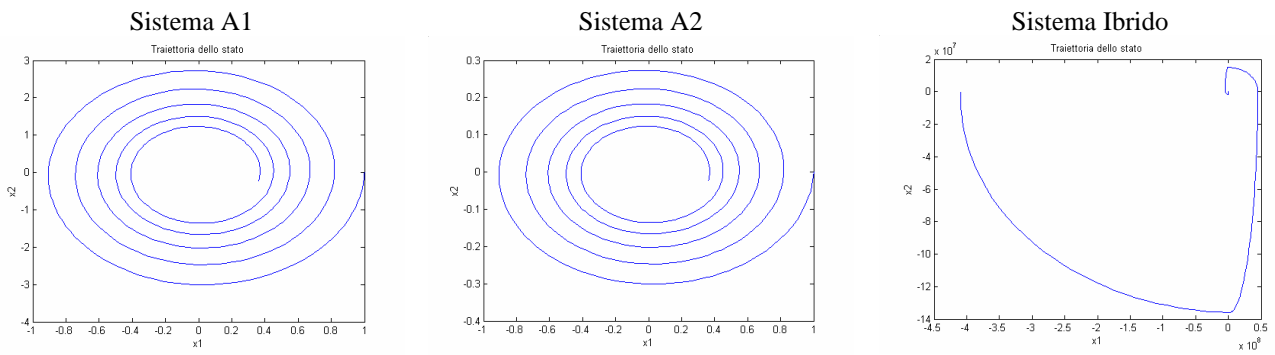
% Integra
[t,x]=ode45('ibrido',tevoluzione,xo,options,a1,a2)

% Grafico del moto dello stato
figure
plot(t,x(:,1),'-',t,x(:,2),'--')
title('Moto degli stati');
xlabel('tempo')
ylabel('x1 -                x2 --')

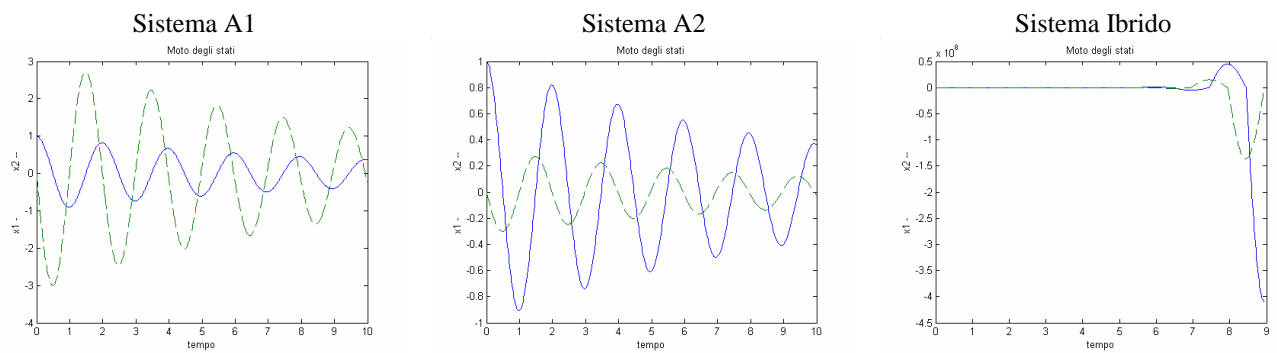
% Grafico della traiettoria dello stato
figure
plot(x(:,1),x(:,2),'-')
title('Traiettoria dello stato');
xlabel('x1')
ylabel('x2')

```

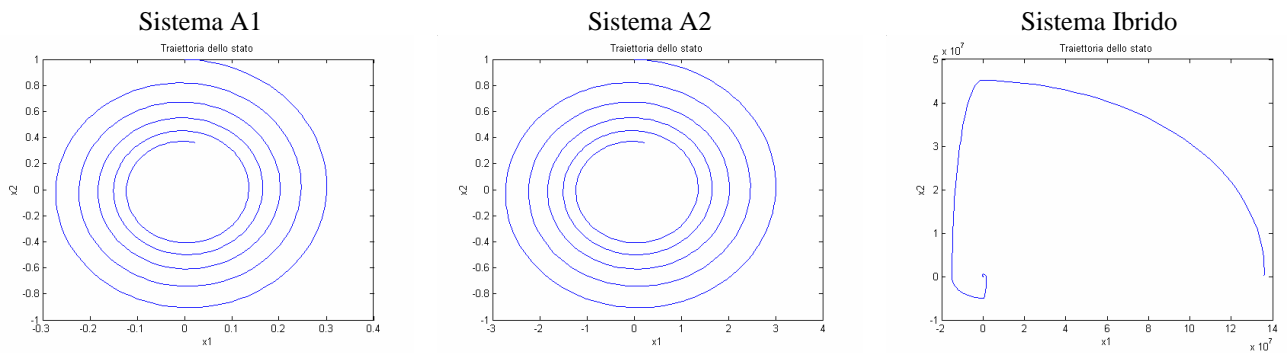
Riportiamo di seguito i risultati dell'integrazione dei singoli sistemi e di quello ibrido:



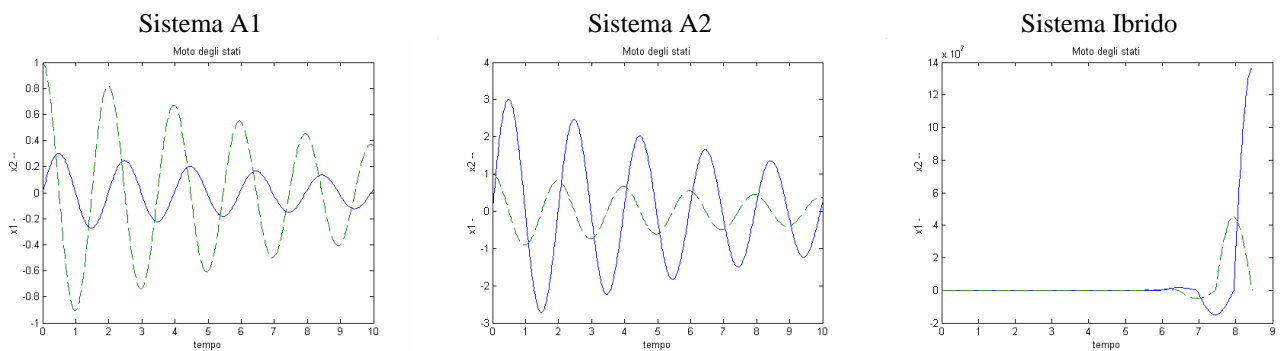
**Figura 9: Traiettoria con stato iniziale  $x_1(t)=1, x_2(t)=0$**



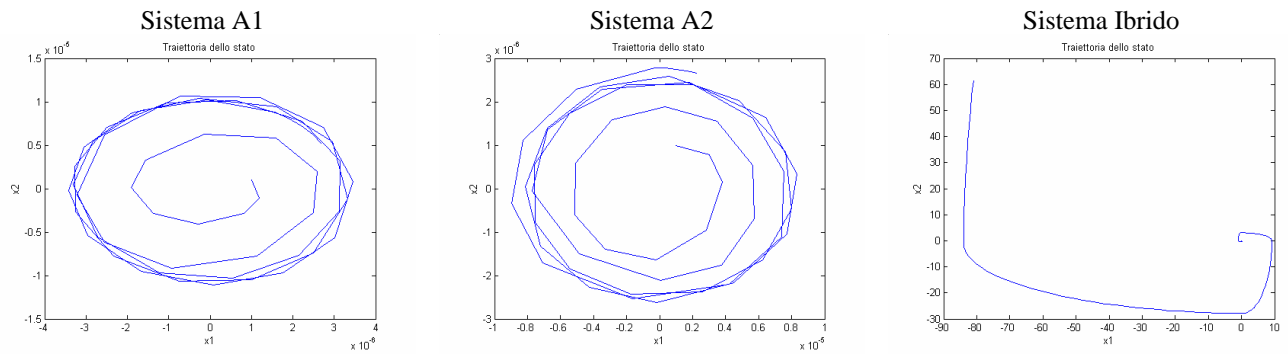
**Figura 10: Moto libero con stato iniziale  $x_1(t)=1, x_2(t)=0$**



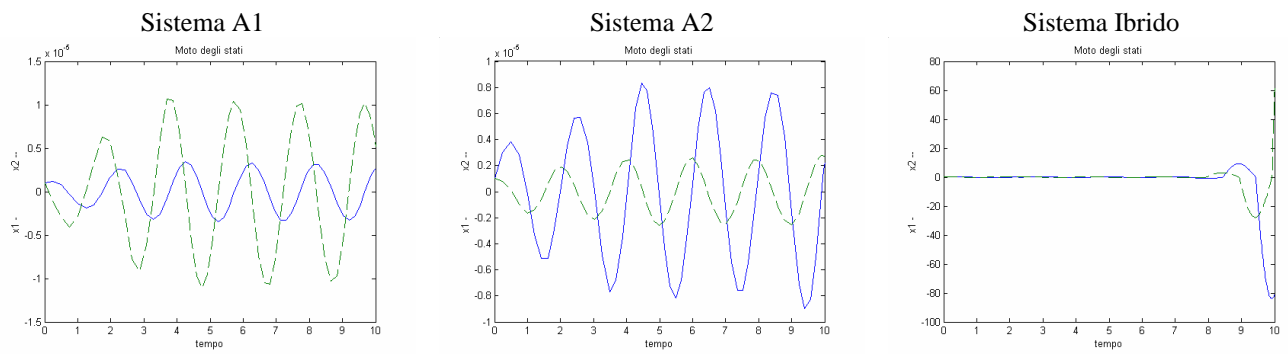
**Figura 11: Traiettoria con stato iniziale  $x_1(t)=0, x_2(t)=1$**



**Figura 12: Moto libero con stato iniziale  $x_1(t)=0, x_2(t)=1$**



**Figura 13: Traiettoria con stato iniziale  $x_1(t)=x_2(t)=10^{-6}$**



**Figura 14: Moto libero con stato iniziale  $x_1(t)=x_2(t)=10^{-6}$**

La simulazione evidenzia come una “combinazione” di sistemi stabili possa originare un sistema ibrido instabile.

# Motore in corrente continua

## Testo dell'esperienza

1. Si realizzi in ambiente simulink il sistema di equazioni differenziali relative al modello del motore in corrente continua descritto da:

$$\begin{aligned}\dot{i}_a(t) &= -\frac{Ra}{La}i_a(t) - \frac{Km}{La}\omega(t) + \frac{1}{La}Va(t) \\ \dot{\omega}(t) &= \frac{Km}{J}i_a(t) - \frac{f}{J}\omega(t) - \frac{1}{J}Cc(t)\end{aligned}$$

Se ne verifichi successivamente la correttezza confrontandolo con le realizzazioni equivalenti nello spazio degli stati.

2. Utilizzando gli stessi valori dei parametri del motore in corrente continua, determinare l'ampiezza del gradino  $V_a(t)$  necessaria a raggiungere una velocità angolare di regime pari a  $\omega(t) = 10\text{rad/s}$ , nelle ipotesi di assenza del carico  $C_c = 0$  e con il modello del motore del secondo ordine. Si verifichi analiticamente il risultato ottenuto.
3. Fissata l'ampiezza della tensione di armatura a  $V_a = 10\text{V}$ , progettare la durata dell'impulso  $t$  in modo da raggiungere una posizione assegnata  $a = 20\text{rad}$ , sempre nelle ipotesi di assenza di carico.
4. Fissato  $t$ , graficare l'andamento temporale della posizione del rotore per una tensione pari alla metà e al doppio della tensione fissata al punto precedente.

## Quesito 1.

Nello script file `Dati_Motore_cc.m` abbiamo salvato i parametri ed il modello del motore in corrente continua nello spazio degli stati che verrà poi utilizzato dall'`ODEsolver` (stiff) invocato da simulink.

```
% Dati motore in corrente continua

Ra=3;      %Ohm      Resistenza di armatura
La=30e-3; %H        Induttanza di armatura
Km=2;     %Nm/A
J=3;      %Kg m^2
f=5e-3;   %Nms/rad

A=[-Ra/La -Km/La; Km/J -f/J];
B=[1/La 0; 0 -1/J];
C=[0 1];
D=[0 0];
```

Abbiamo descritto in ambiente grafico simulink il sistema di equazioni differenziali del motore in corrente continua insieme al suo modello nello spazio degli stati, quest'ultimo rappresentato dall'apposito blocco funzionale. Il sistema di equazioni differenziali è stato rappresentato trasformandolo secondo Laplace e quindi descrivendolo con schema a blocchi. In particolare  $i_a$  e  $\omega_a$  sono state ottenute da  $di_a/dt$  e  $d\omega_a/dt$  integrando, ossia moltiplicando per il blocco  $1/s$ .

Per evidenziare l'equivalenza dei modelli descrittivi abbiamo effettuato la simulazione applicando gli stessi identici ingressi come appare in Figura 15:

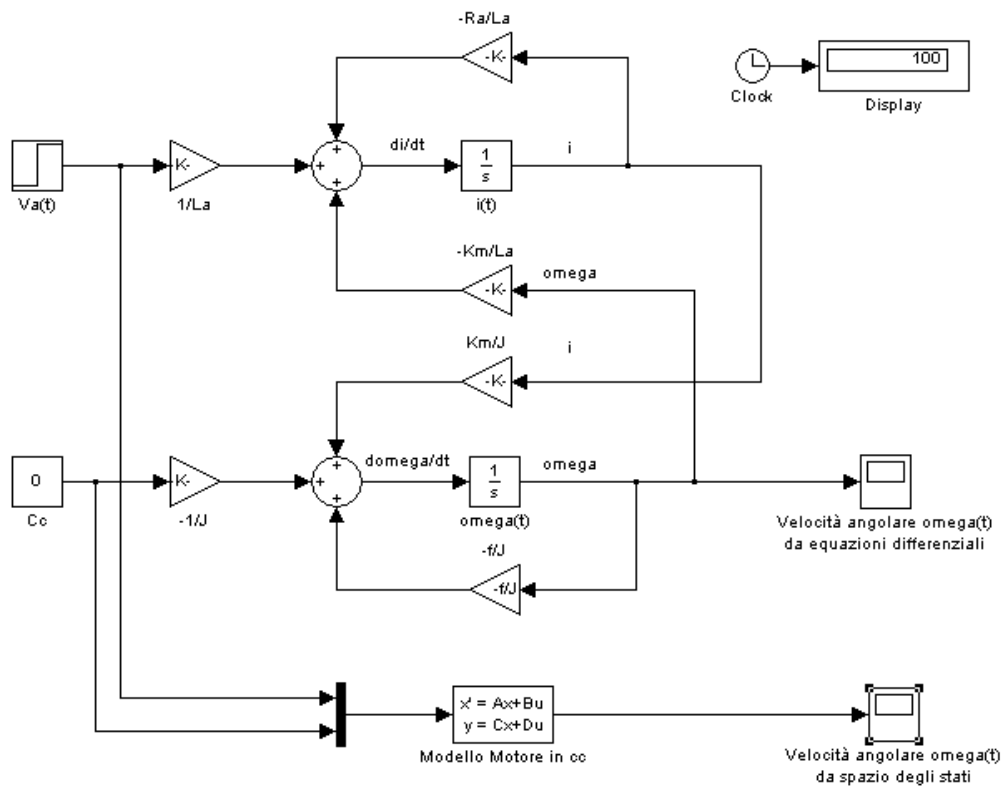
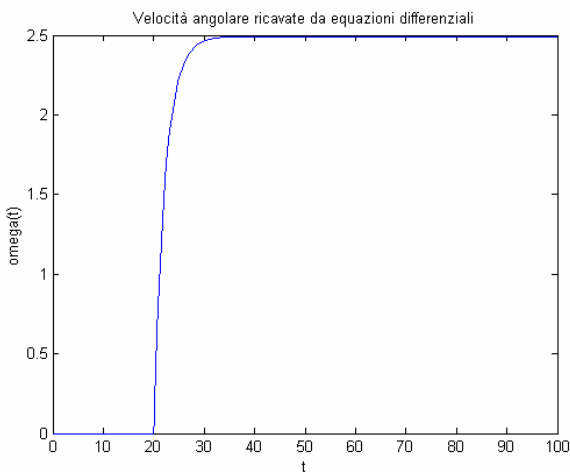


Figura 15: Modello in ambiente simulink del motore in corrente continua

Ottenendo due risposte identiche (Figura 16):

Risposta dal sistema di equazioni differenziali



Risposta dal modello nello spazio degli stati

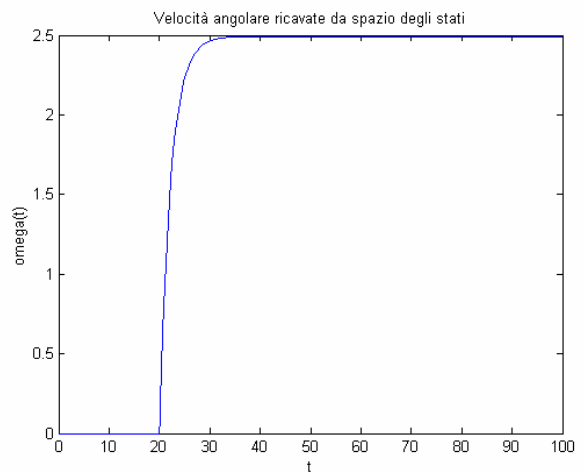


Figura 16: Risposta ad  $V_a(t)=5V$  applicato all'istante  $t=20sec$



**Quesito 2.**

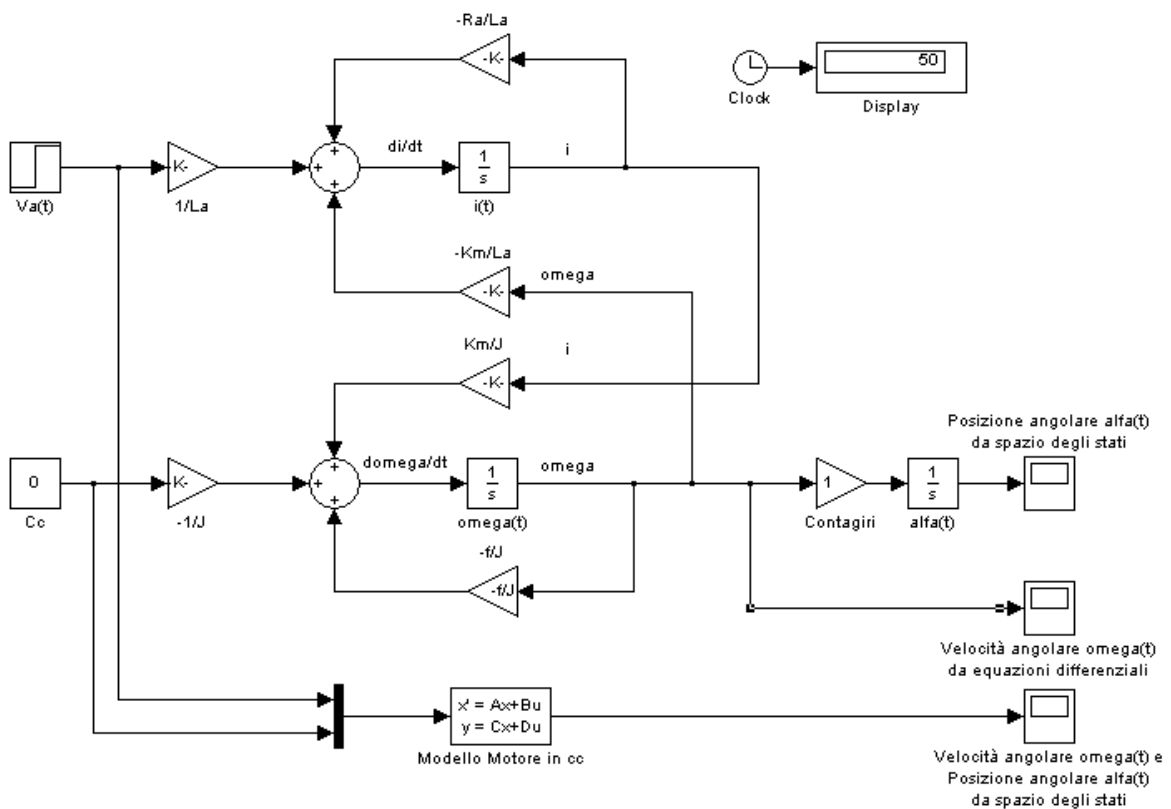
Nello script file Dati\_Motore\_cc\_2ord.m abbiamo scritto il modello del secondo ordine nello spazio degli stati aggiungendo la posizione del rotore  $\alpha(t)$ :

```
% Dati motore in corrente continua, modello del II ordine

Ra=3;      %Ohm      Resistenza di armatura
La=30e-3; %H        Induttanza di armatura
Km=2;      %Nm/A
J=3;       %Kg m^2
f=5e-3;    %Nms/rad

A=[-Ra/La -Km/La 0; Km/J -f/J 0; 0 1 0];
B=[1/La 0; 0 -1/J; 0 0];
C=[0 1 0; 0 0 1];
D=[0 0; 0 0];
```

Caricando nel workspace Dati\_Motore\_cc\_2ord.m possiamo riutilizzare il precedente schema simulink con una piccola modifica alla parte differenziale: integrando nel tempo la velocità angolare ottengo la posizione angolare del rotore (Figura 17). La parte dello schema relativa allo spazio degli stati, da già uno stato che comprende anche la posizione:



**Figura 17: Modello in ambiente simulink del motore in corrente continua del II ordine**

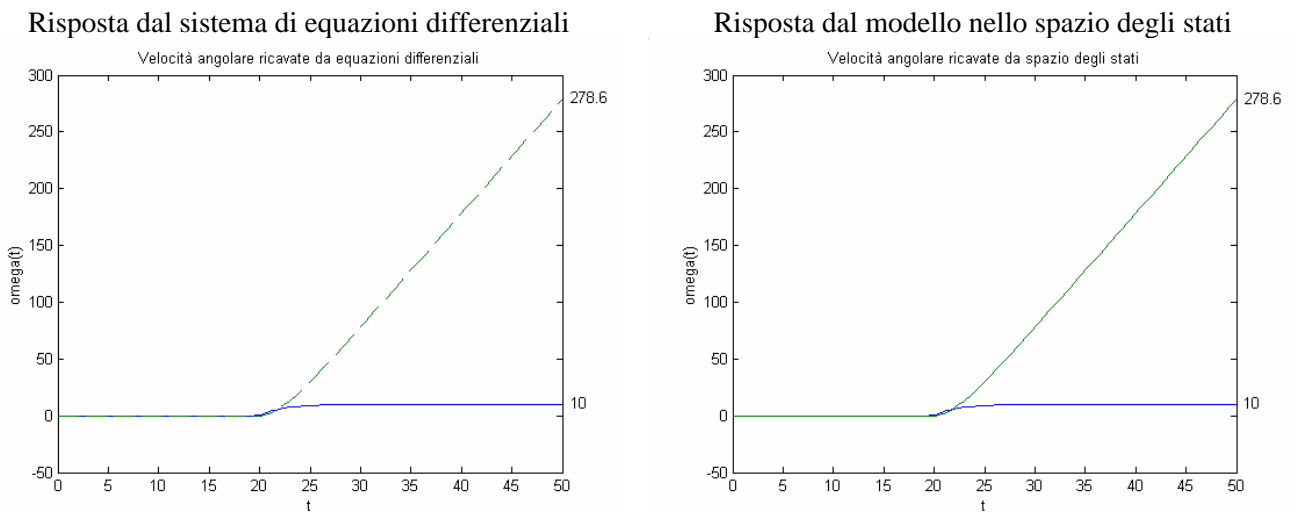
Per rispondere al secondo quesito dobbiamo risolvere rispetto a  $V_a$  il sistema di equazioni ponendo:

$$\frac{d\omega}{dt} = 0 \qquad \frac{Km}{J} i_a - \frac{f}{J} 10 = 0 \qquad i_a = 10 \frac{f}{Km}$$

che sostituita nell'altra equazione:

$$\frac{di_a}{dt} = 0 \qquad V_a = Km10 + Ra10 \frac{f}{Km} = 20.15$$

Ripetendo la simulazione avendo posto  $V_a(t)=20.15$  si osservano i grafici di Figura 18:



**Figura 18: Risposta ad  $V_a(t)=20.15V$  applicato all'istante  $t=20sec$**

**Quesito 3.**

Applicando un ingresso a gradino di ampiezza  $V_a=20V$  riusciamo a raggiungere la posizione angolare  $\alpha=20rad$  se imponiamo una durata di  $\tau=4sec$ , come in Figura 19:

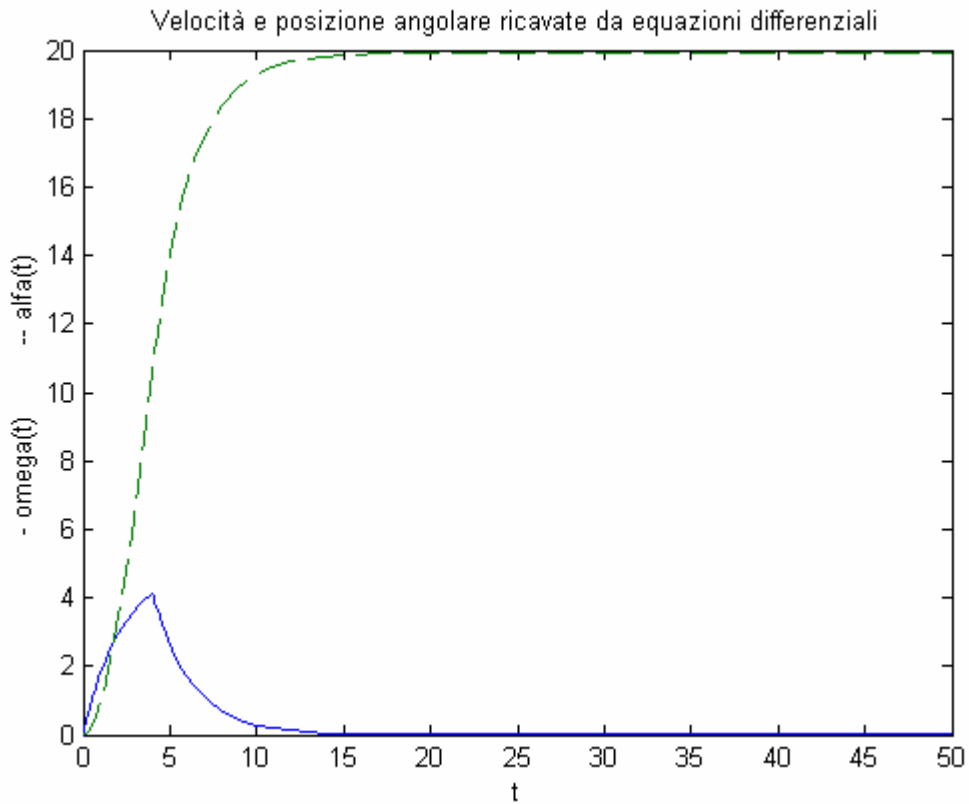


Figura 19: Risposta all'impulso  $V_a(t)=10V$  per  $t \in [0,t]$  con  $t=4sec$

**Quesito 4.**

Riportiamo in Figura 20 l'andamento temporale della posizione angolare del rotore  $\alpha(t)$  e della sua velocità angolare  $\dot{\alpha}(t)$  in risposta a gradini di durata di  $\tau=4sec$  ma di differente ampiezza:

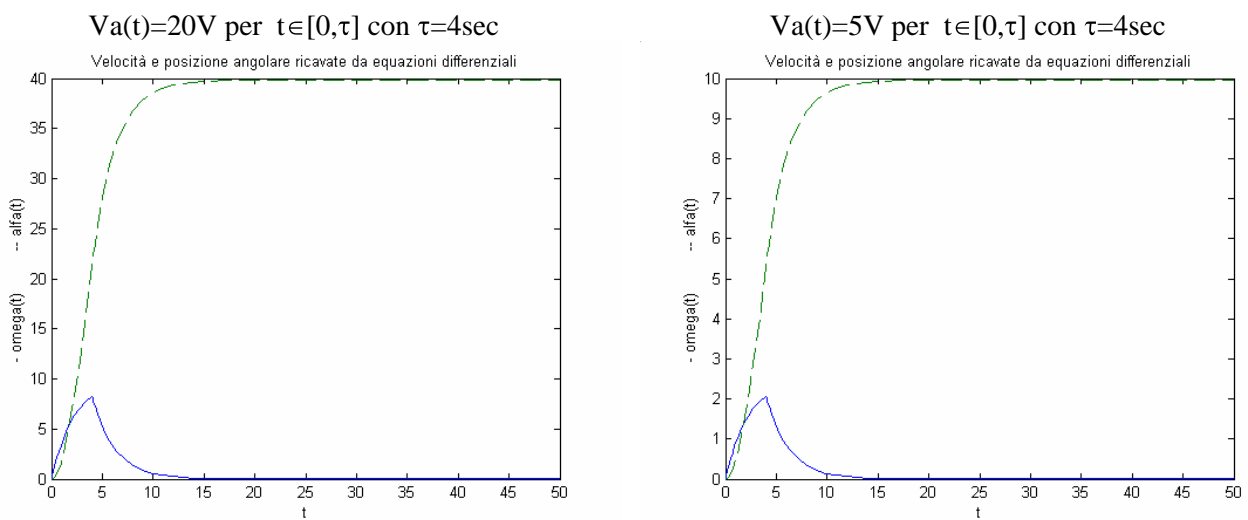


Figura 20: Risposte ad impulsi di stessa durata ma di diversa ampiezza



# Osservatori e retroazione uscita-stato-ingresso

É dato il sistema dinamico descritto dal modello:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

con  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^r$  e  $y(t) \in \mathbb{R}^m$  e  $n = 4$ ,  $r = 1$ ,  $m = 1$ . Le matrici del sistema siano:

$$A = \begin{bmatrix} -13 & -46 & -48 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad C = [0 \quad 0 \quad 1 \quad 0 \quad 0]$$

In riferimento al sistema dato:

1. Effettuare una realizzazione del sistema in ambiente Simulink e calcolarne gli autovalori;
2. determinare in ambiente Matlab il valore della matrice di retroazione  $K_s$  per una retroazione uscita-ingresso che assegni ad un autovalore il valore  $-9$ . Successivamente realizzare lo schema Simulink per il sistema in retroazione eccitato da un gradino di ampiezza 10; graficare la risposta al gradino  $y(t)$  e l'errore  $e(t)$ . Calcolare graficamente il tempo di assestamento e la massima sovraelongazione.
3. Dato in ingresso un gradino di ampiezza 10, determinare una matrice di retroazione uscita-ingresso  $K_r$  affinché il tempo di assestamento sia  $T_a=3s$ . In tali condizioni, in seguito determinare graficamente la massima sovraelongazione.
4. Determinare per il sistema di partenza il guadagno  $K_o$  di un osservatore dello stato che assegni gli autovalori  $[-3, -3, -4]$ . Si grafichi l'errore di stima nelle ipotesi che lo stato iniziale del sistema parta da  $[12, -20, -30]$ .

## Quesito 1.

Il modello in ambiente simulink è:

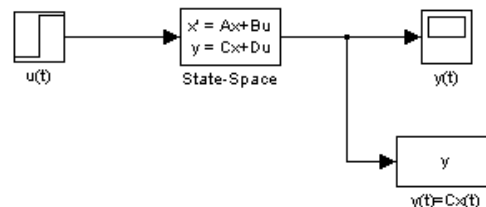


Figura 21: Sistema in ambiente simulink

Caratterizzato dagli autovalori calcolati con  $\text{eig}(A)$ :

$$\text{AUTOVAL} = [ -8.0000, 4.6458, -3.0000, -2.0000, -0.6458 ]$$

### Quesito 2.

Per assegnare l'autovalore  $-9$  ho scritto lo script file `sistema_assegn_autoval.m` in cui sfrutto la funzione `rlocfind`:

```
% Assegnamento di autovalore al sist. descritto in descrizione_sistema.m
% retroazionato uscita ingresso.

% Carico la descrizione del sistema
descrizione_sistema

% Calcolo del guadagno Ks dell'anello di retroazione in modo da assegnare
% un autovalore pari a -9
disp('Guadagno Ks della retroazione uscita ingresso')
disp('in modo da assegnare l''autovalore -9')

[Ks,POLI]=rlocfind(SYS,-9)
```

Il risultato dell'esecuzione da:

$$Ks=42.0000 \quad \text{POLI}=[ -9.0000, -2.0000+2.4495i, -2.0000-2.4495i ]$$

Il sistema retroazionato uscita ingresso è riportato in Figura 22:

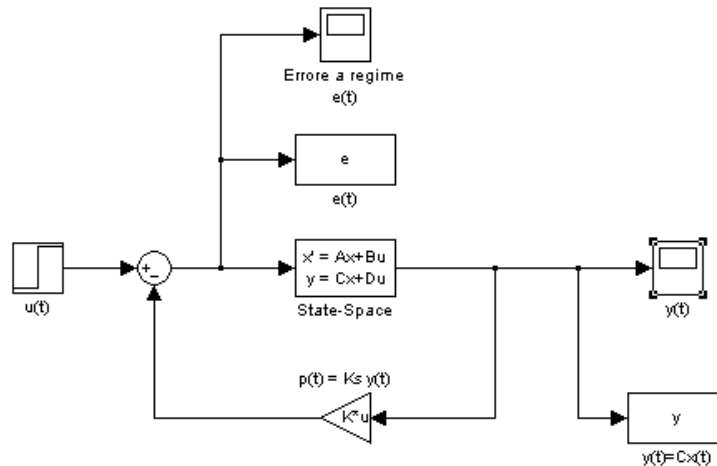
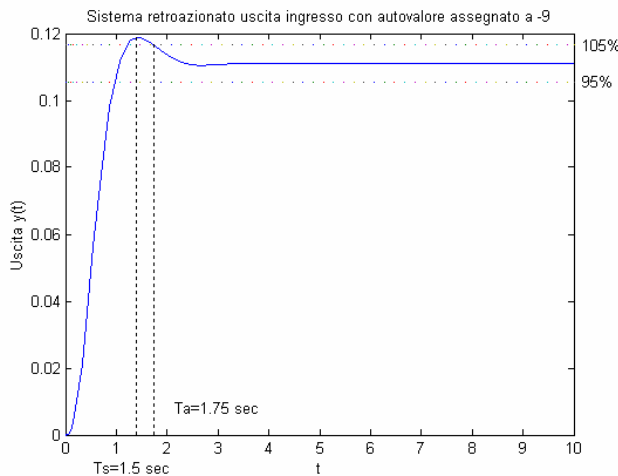
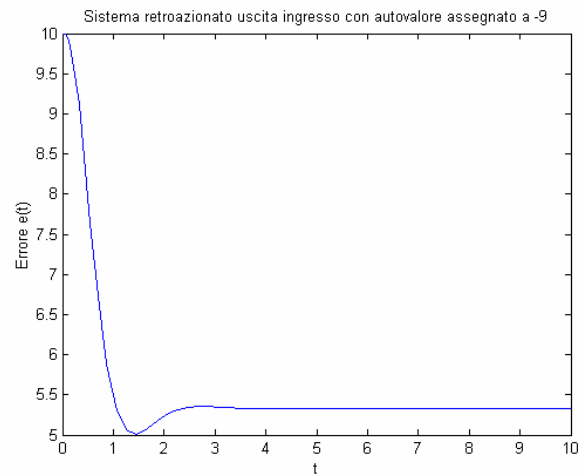


Figura 22: Sistema retroazionato uscita ingresso

La simulazione calcola l'uscita (Figura 23) caratterizzata da massima sovraelongazione percentuale  $S=7.00\%$  e l'errore a regime (Figura 24).



**Figura 23: Uscita  $y(t)$  con  $u(t)=10$**



**Figura 24: Errore  $e(t)$  con  $u(t)=10$**

**Quesito 3.**

Anche se il sistema descritto è di ordine superiore al secondo possiamo soddisfare le specifiche con una certa precisione osservando che i poli più negativi danno modi che si assestano velocemente, mentre il sistema sarà maggiormente influenzato dai 2 poli, a parte reale negativa, più vicini all'origine. In questo modo stiamo approssimando il sistema ad uno del secondo ordine con:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \qquad s_{1,2} = -\delta\omega_n \pm \omega_n \sqrt{1 - \delta^2}$$

In cui il tempo di assestamento  $T_a$  e la massima sovraelongazione  $S$  sono dati da:

$$T_a = \frac{3}{\delta\omega_n} \qquad S = 100e^{-\frac{\pi\delta}{\sqrt{1-\delta^2}}}$$

Poiché è richiesto un  $T_a=3$  sec si ricava facilmente  $\delta\omega_n=1$ . Con `rlocus(SYS)` visualizziamo il luogo delle radici del sistema con blocco di retroazione  $K_r$ ; ora con `rlocfind(SYS)` cerchiamo graficamente sul luogo delle radici il polo di parte reale  $-\delta\omega_n=-1$  appena calcolato, ottenendo:  $-1.02+4.72i$ .

Tramite lo script file:

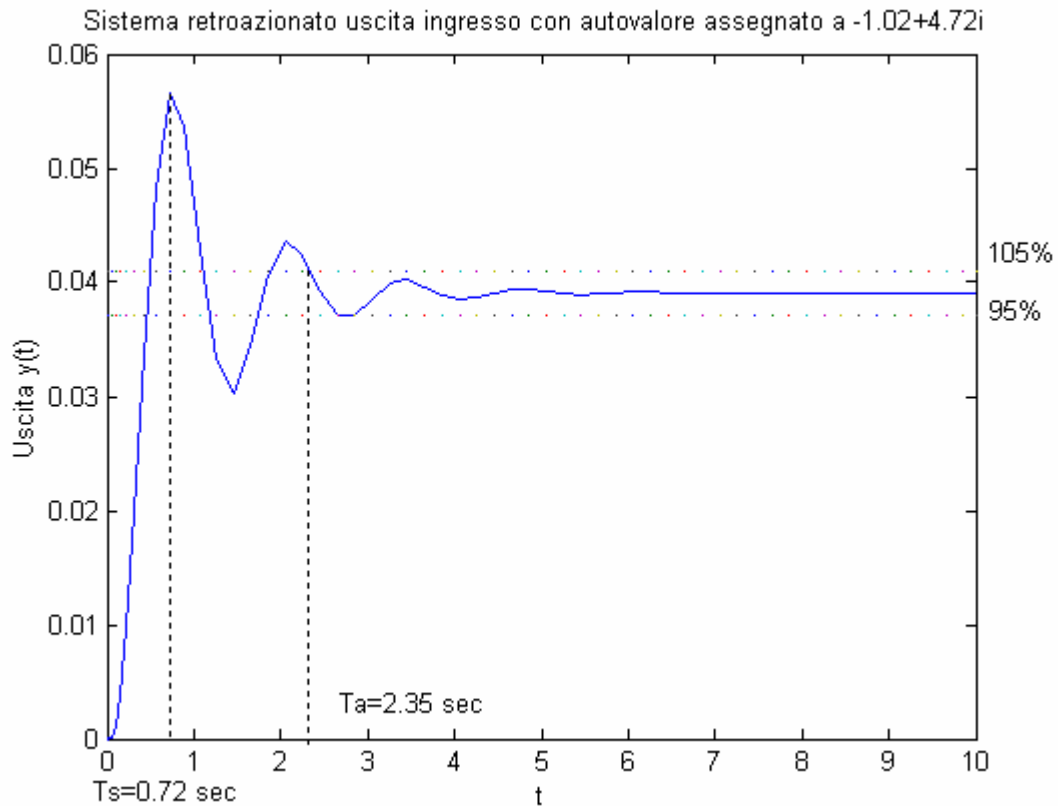
```
% Specifica su Ta=3sec garantita tramite assegnamento di un autovalore
% al sistema descritto in descrizione_sistema.m retroaz. uscita ingresso.

disp('Guadagno Ks della retroazione uscita ingresso')
disp('in modo da assegnare l''autovalore -1.02+4.72i che garantisce
Ta=3s')
[Kr,POLI]=rlocfind(SYS, -1.02 + 4.72i )
```

determiniamo il  $K_r$  necessario a garantire il  $T_a$  richiesto:

$$K_r = 207.9089 \qquad POLI = [ -10.9335 \quad -1.0332+4.7263i \quad -1.0332-4.7263i ]$$

Applicando in ingresso un gradino di ampiezza 10 abbiamo ottenuto l'andamento di Figura 25:



**Figura 25: Risposta ad  $u(t)=10$**

Verificando il vincolo imposto su  $T_a$  ed osservando una massima sovraelongazione percentuale 44.72%

#### Quesito 4.

Poiché il sistema non è completamente raggiungibile e controllabile ne abbiamo determinato la forma minima, tramite lo script file:

```
% Descrizione del sistema
A=[-13 -46 -48 0 0; 1 0 0 0 0; 0 1 0 0 0; 0 1 0 0 1; 0 0 1 3 4];
B=[1; 0; 0; 0; 0];
C=[0 0 1 0 0];
D=[0];

% Determino la forma minima
RCOR
% tale script file ritorna proprio le matrici Ao1 Bo1 Co1

% So che Ao1 è 3x3, per cui riesco a piazzare i 3 seguenti autovalori
disp('Determino Ko dell''osservatore che assegna gli autovalori:')
AUTOVALORI=[-3, -3.001, -4]

% Determino il Ko per costruire l'osservatore identità della forma minima
```



```

disp('Il guadagno dell''osservatore è:')
Ko=place(Ao1',Co1',AUTOVALORI)'

% Costruisco l'osservatore per adattarmi al blocco funzionale di simulink
disp('Sono disponibili per simulink le seguenti matrici dell''osservatore')
Aos=Ao1-Ko*Co1
Bos=[Bo1 Ko]
Cos=eye(3)
Dos=zeros(3,2)
    
```

caratterizzata dalle matrici:

```

Ao1 =

    -9.3295    -0.1893     0.0371
    50.4564   -2.9951     0.6026
    44.0368   -3.0085    -0.6754
    
```

```

Bo1 =

    -0.1466
     0.7421
     0.6540
    
```

```

Co1 =

    -0.7419    -0.5198     0.4235
    
```

Lo schema adottato per la simulazione è:

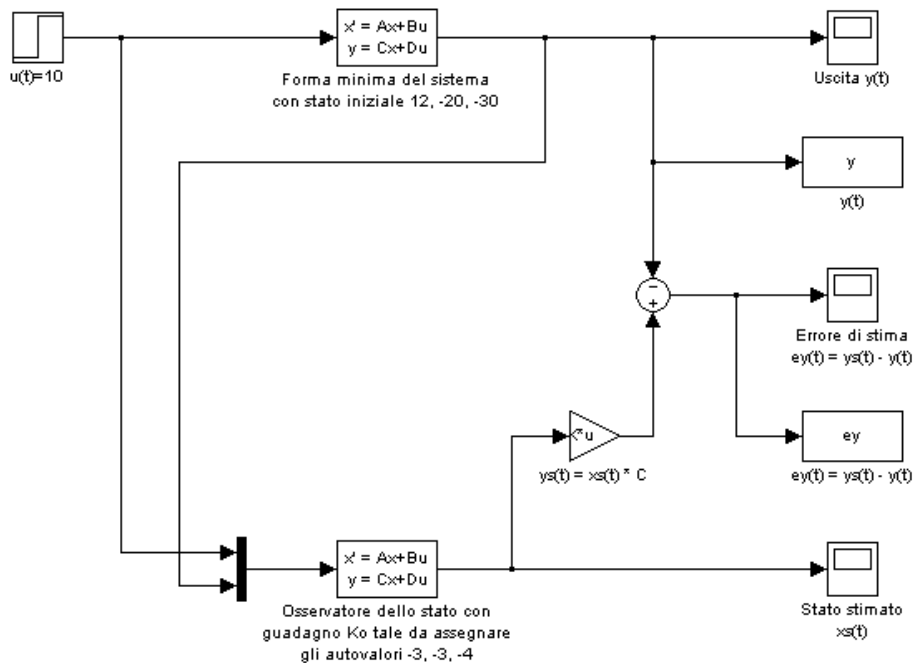
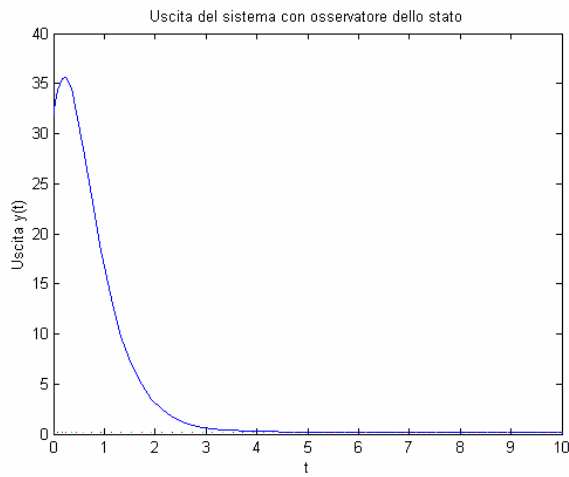
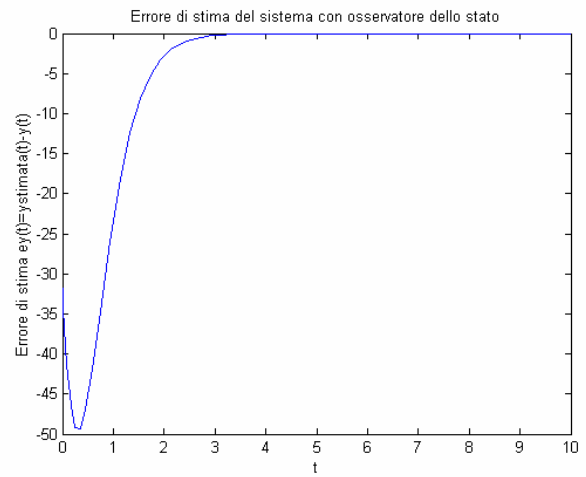


Figura 26: Sistema con osservatore dello stato

A fronte di un gradino di ampiezza 10, la forma minima del sistema risponde come in Figura 27, mentre l'uscita stimata tramite l'osservatore dello stato si differenzia da quella reale per un errore il cui andamento è riportato in Figura 28:



**Figura 27: Uscita  $y(t)$  con stato iniziale  $x_0=[12, -20, -30]$**



**Figura 28: Errore di stima  $e_y(t)=y_{stimata}(t) - y(t)$**

# Rete correttrice per un motore in corrente continua

## Testo dell'esperienza

Un motore in corrente continua è rappresentato dalla seguente funzione di trasferimento:

$$G_m(s) = \frac{6.087 \times 10^{10}}{s(s^3 + 423.42s^2 + 2.6667 \times 10^6 s + 4.2342 \times 10^8)}$$

A causa dell'accoppiamento con l'albero motore, tale funzione contiene due poli poco smorzati che generano oscillazioni nella risposta. Devono essere soddisfatte le seguenti specifiche:

1. Massima elongazione < 1%,
2. Tempo di salita < 0.15 secondi
3. Tempo di assestamento < 0.15 secondi
4. La risposta non deve presentare oscillazioni

Progettare una rete anticipatrice con funzione di trasferimento:

$$G_c(s) = \frac{1 + \alpha \tau s}{1 + \tau s}$$

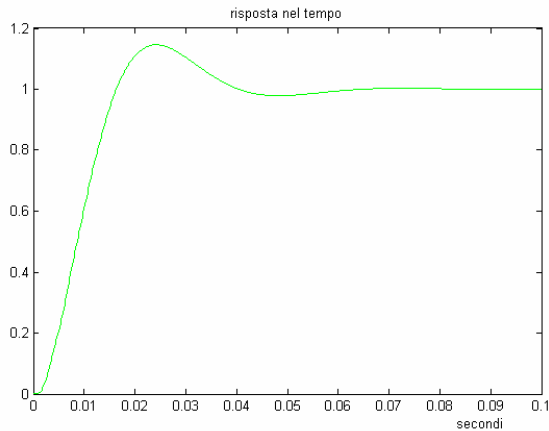
affinché risultino verificate le specifiche relative alla risposta al gradino.

## Progetto

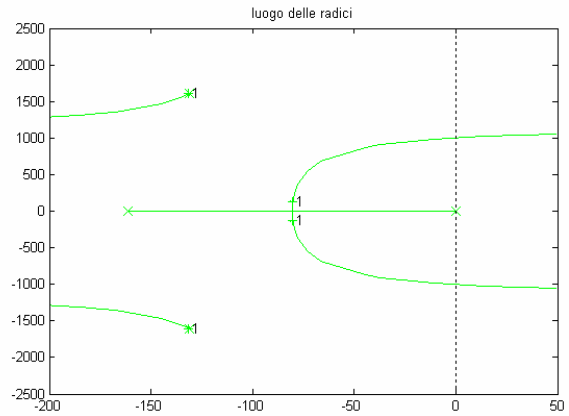
Entrati in ambiente tfi inseriamo la funzione di trasferimento del motore in corrente continua:

$$G_m = \frac{6.087e+010}{s (s^3 + 423.4*s^2 + 2.667e+006*s + 4.234e+008)}$$

Chiudendo il sistema in retroazione unitaria ed applicando in ingresso un gradino otteniamo una risposta come in Figura 29, in cui si evidenzia un'oscillazione dovuta ai due poli poco smorzati in Figura 30:



**Figura 29: Risposta al gradino**



**Figura 30: Luogo delle radici**

Il sistema da correggere ha:

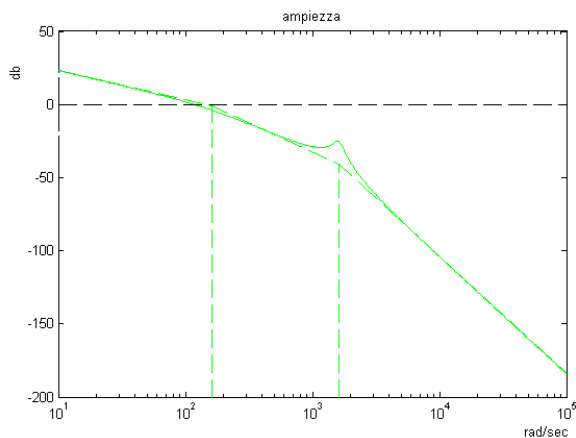
RISPOSTA AL GRADINO :

massima sovraelongazione: 14.61 percento per t=0.02404 sec  
 tempo di ritardo (al 50 percento): 0.008639 sec  
 tempo di salita (dal 10 al 90 percento): 0.01096 sec  
 tempo di assestamento (al più/meno 5 percento): 0.03471 sec

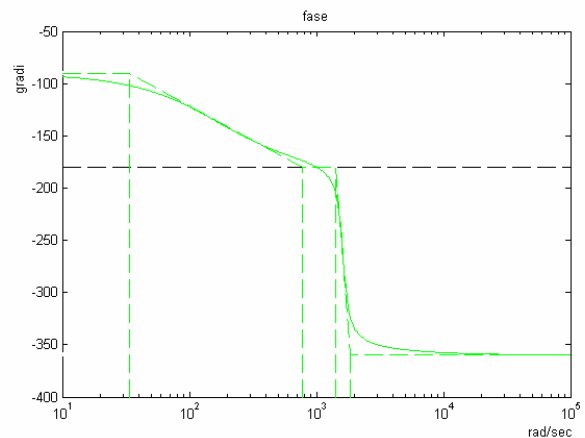
ERRORI A REGIME (SOLO AD ANELLO CHIUSO) :

errore a regime in risposta al gradino: 0  
 errore a regime in risposta alla rampa: 0.006956

Per completezza riportiamo anche i diagrammi di Bode di Gm dell'ampiezza (Figura 31) e della fase (Figura 32):



**Figura 31: Diagramma di Bode dell'ampiezza di Gm**



**Figura 32: Diagramma di Bode della fase di Gm**

Così pure di Gm chiusa in anello unitario:

$$G_{mr1} = G_m / (1 + G_m)$$

$$G_{mr1} = \frac{6.087e+010}{(s^4 + 423.4*s^3 + 2.667e+006*s^2 + 4.234e+008*s + 6.087e+010)}$$

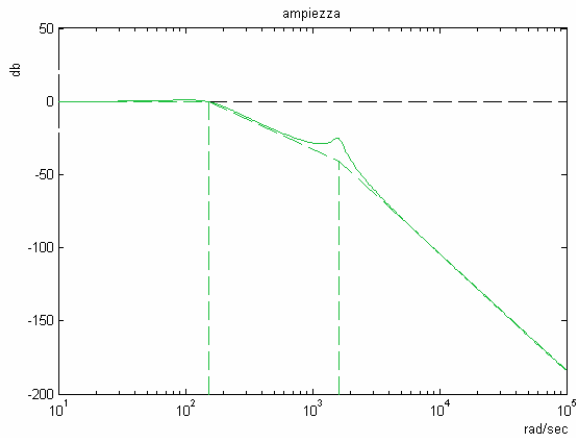


Figura 33: Diagramma di Bode dell'ampiezza di Gmr1

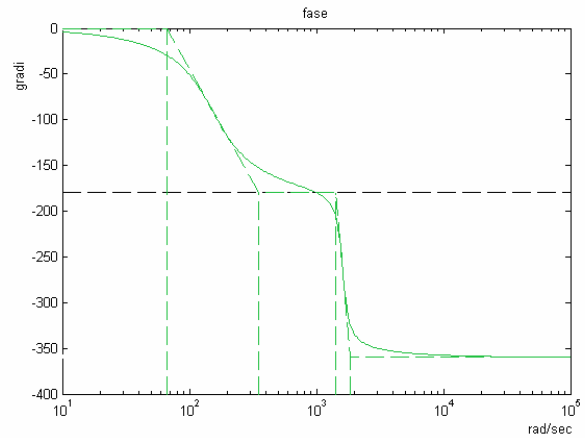


Figura 34: Diagramma di Bode della fase di Gmr1

Poiché dal luogo delle radici di Gm (Figura 30) si osservano i poli:

POLI	MODULO	DELTA
-131.5 +1608*j	+1613	+0.08153
-131.5 -1608*j	+1613	+0.08153
-80.2 +130.3*j	+153	+0.5243
-80.2 -130.3*j	+153	+0.5243

costruisco la rete anticipatrice in modo che compensi la parte reale del polo non nullo più vicino all'asse immaginario. Pongo lo zero della retocorrettrice in prossimità del polo da compensare, mentre pongo il polo della rete a valori più negativi in modo da assicurare una dinamica veloce:

$$G_c = (s+80) / (s+2*80)$$

$$G_c = \frac{1 (s + 80)}{(s + 160)}$$

La funzione di trasferimento G della rete e del motore è:

$$G = G_c * G_m$$

$$G = \frac{6.087e+010 (s + 80)}{s (s + 160) (s^3 + 423.4*s^2 + 2.667e+006*s + 4.234e+008)}$$

Il cui luogo delle radici (Figura 35) evidenzia l'effetto "attraattivo" dello zero:

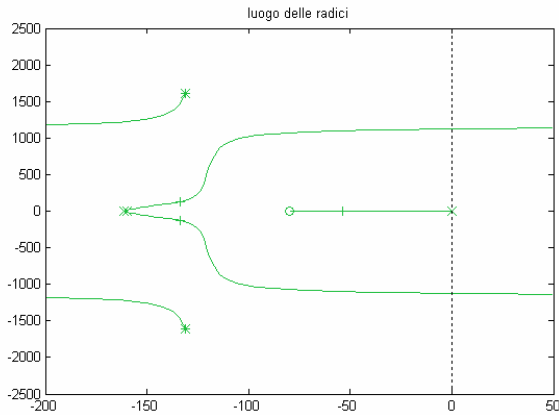


Figura 35: Luogo delle radici di G

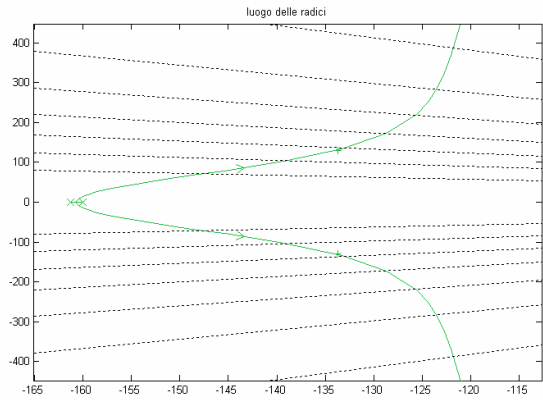


Figura 36: Zoom per la scelta  $0.85 < \delta < 1$

Come primo tentativo per il calcolo di K assumiamo che il sistema sia del secondo ordine, per cui riusciamo a garantire una massima sovraelongazione  $S < 1\%$  a patto di scegliere  $0.85 < \delta < 1$  (Figura 36).

Sceglia un  $\delta=0.8559$ , cui corrisponde un  $K=0.52$ , otteniamo una  $G1=K*G$  di primo tentativo:

$$G1 = 0.52 * G$$

$$G1 = \frac{3.165e+010 (s + 80)}{s (s + 160) (s^3 + 423.4*s^2 + 2.667e+006*s + 4.234e+008)}$$

caratterizzata dalla risposta al gradino di Figura 37, e dal luogo di Figura 38:

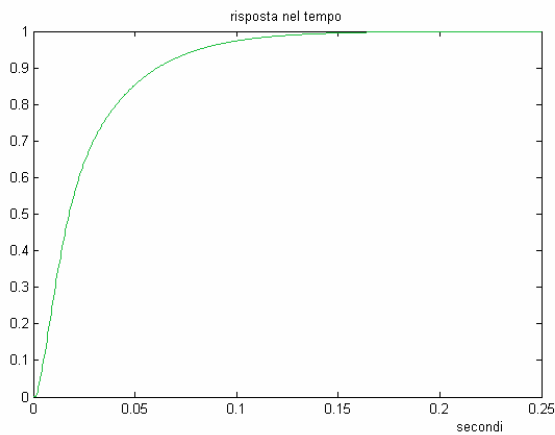


Figura 37: Risposta al gradino di G1

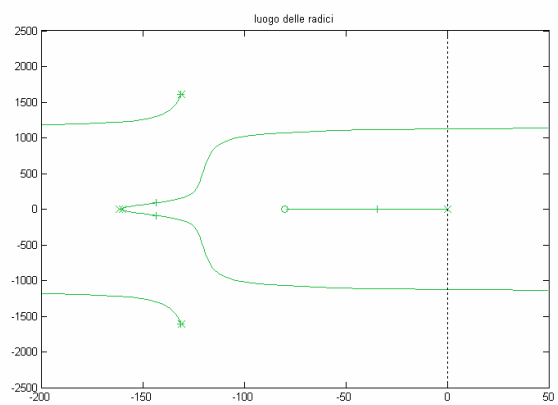


Figura 38: Luogo delle radici di G1

e che soddisfa i vincoli richiesti:

RISPOSTA AL GRADINO :

nessuna sovraelongazione  
tempo di ritardo (al 50 per cento): 0.01765 sec  
tempo di salita (dal 10 al 90 per cento): 0.05599 sec  
tempo di assestamento (al più/meno 5 per cento): 0.08116 sec

ERRORI A REGIME (SOLO AD ANELLO CHIUSO) :

errore a regime in risposta al gradino: 0  
errore a regime in risposta alla rampa: 0.02675





# Sintonizzazione di controllori PID

## Testo dell'esperienza

1. Data la funzione di trasferimento:

$$G_s = \frac{1}{(1+s)^3}$$

progettare un regolatore PID utilizzando le formule di Ziegler-Nichols (funzioni del valore del guadagno in retroazione  $K_o$  che porta il sistema al limite della stabilità e del periodo delle oscillazioni ottenute  $T_o$ ).

I parametri  $K_o$  e  $T_o$  possono essere ottenuti dal luogo delle radici del sistema chiuso in retroazione unitaria con un regolatore di tipo P secondo un guadagno proporzionale  $K_o$ . Con le formule di Ziegler-Nichols progettare quindi un PI, un PID ed un PID con limitazione della derivata ( $N=10$ ) e confrontare i risultati ottenuti con i diversi regolatori.

2. Riprendendo il sistema descritto dall' Equazione 6.8 realizzare un PID (con  $N=10$ ) con limitazione dell'azione derivativa (cioè che effettui la derivata solo dell'uscita) e si confronti la risposta con quella ottenuta con PID in forma reale realizzato precedentemente.
3. Riprendendo lo schema che utilizza il PID con derivata reale della sola uscita, si introduca un disturbo sull'uscita stessa (rumore bianco con varianza 0.01) e si calcoli l'andamento nel tempo del segnale di controllo per  $N = 10$  e  $N = 30$ . Fare prove con  $N$  diverso e giustificare teoricamente i risultati ottenuti.

## Quesito 1.

Abbiamo inserito la funzione di trasferimento in tfi:

$$G_s = 1 / (1+s)^3$$

$$G_s = \frac{1}{(s + 1)^3}$$

Tramite rootl,  $G_s$  abbiamo calcolato il luogo delle radici (Figura 39):

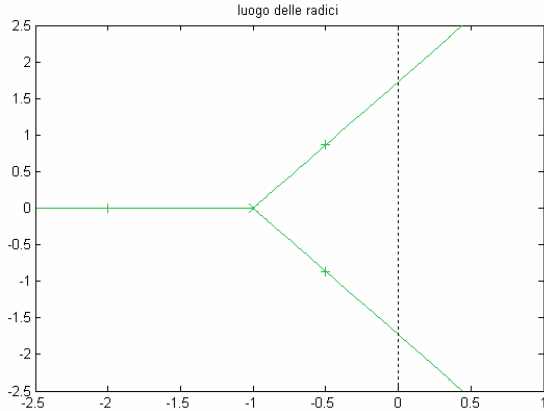


Figura 39: Luogo delle radici di Gs

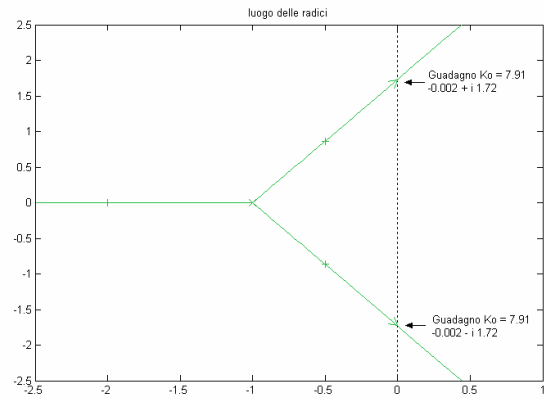


Figura 40: Condizioni di stabilità limite

Con la sola azione di controllo del termine proporzionale si può portare il sistema in condizioni di stabilità limite. Tramite rootl,Gs determiniamo i poli al limite della stabilità (Figura 40). Il guadagno Ko è calcolato per via grafica clickando sul luogo delle radici in corrispondenza dell'asse immaginario, mentre To è determinato tenendo conto del legame con la parte immaginaria del polo con parte reale nulla. Ricaviamo:

$$P_{1,2} = -0.002 \pm 1.72 \quad \rightarrow \quad T_o = \frac{2\pi}{\omega_o} = \frac{2\pi}{1.72} = 3.65 \quad \text{cui corrisponde un guadagno} \quad K_o = 7.91$$

In condizioni di stabilità limite implementiamo il regolatore PI scegliendo i parametri in base alle relazioni di Ziegler e Nichols calcolate dal seguente:

```
function [Kp,Ti]=PI_lim(Ko,To)
% Calcolo dei parametri del regolatore PI in condizioni limite
% di stabilita' secondo le formule di Ziegler e Nichols, dove:
% Ko guadagno al limite della stabilita
% To periodo delle oscillazioni al limite della stabilita

Kp=0.45*Ko;    % Sensibilita proporzionale
Ti=0.85*To;    % Costante di tempo dell'azione integrale

return
```

La cui chiamata coi parametri determinati dal luogo delle radici ritorna:

```
[Kp,Ti]=PI_lim(7.91,3.65)

Kp = 3.5595          Ti = 3.1025
```

Sempre in condizioni di stabilità limite implementiamo il regolatore PID scegliendo i parametri in base alle relazioni di Ziegler e Nichols calcolate dal seguente:

```
function [Kp,Ti,Td]=PID_lim(Ko,To)
% Calcolo dei parametri del regolatore PID in condizioni limite
% di stabilita' secondo le formule di Ziegler e Nichols, dove:
% Ko guadagno al limite della stabilita
% To periodo delle oscillazioni al limite della stabilita

Kp=0.6*Ko;    % Sensibilita proporzionale
```

```
Ti=0.12*To; % Costante di tempo dell'azione integrale
Td=0.5*To; % Costante di tempo dell'azione derivativa

return
```

La cui chiamata coi parametri determinati dal luogo delle radici ritorna:

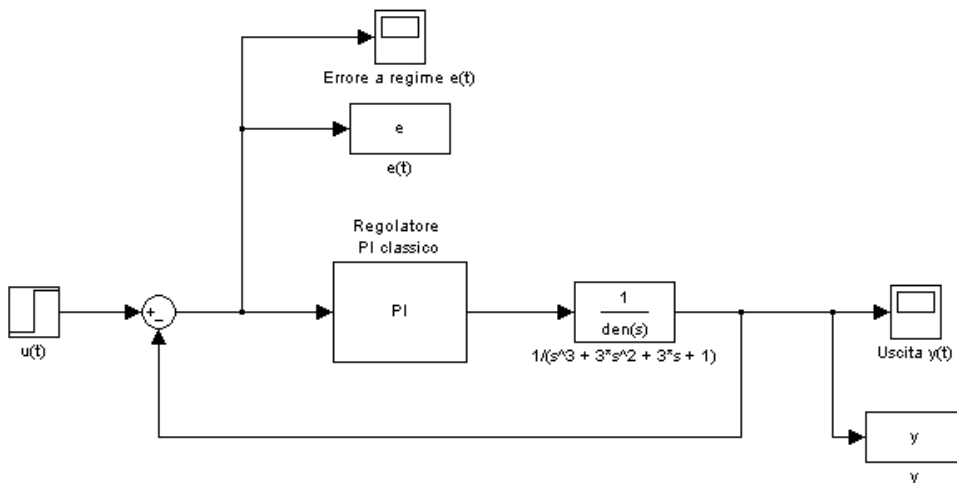
```
[Kp,Ti,Td]=PID_lim(7.91,3.65)
```

Kp = 4.7460                      Ti = 0.4380                      Td = 1.8250

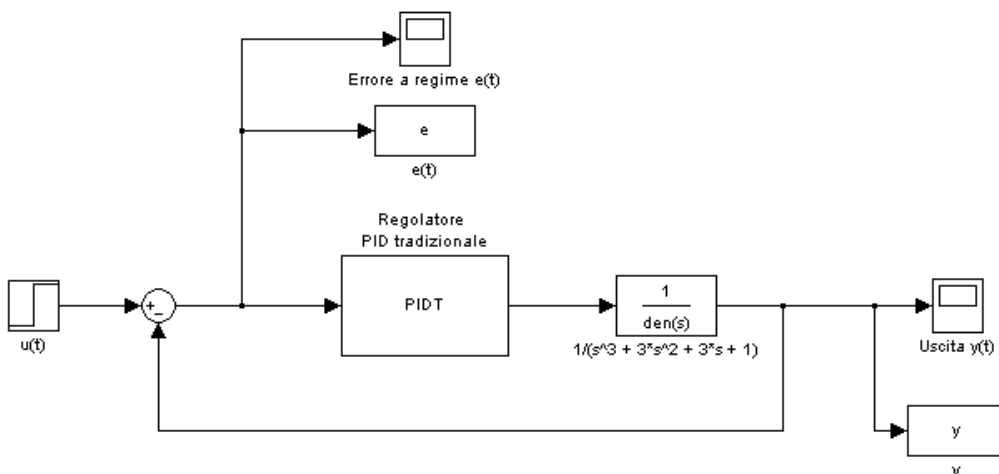
Sempre in condizioni di stabilità limite implementiamo ora il regolatore PID con limitazione sulla derivata (N=10). I parametri sono identici a quelli del regolatore PID tradizionale tranne per il termine derivativo Td che va diviso per la limitazione desiderata N=10.

Nella realizzazione dei sistemi in ambiente simulink abbiamo utilizzato la libreria PIDlib.mdl che mette a disposizione blocchi funzionali per i regolatori più comuni.

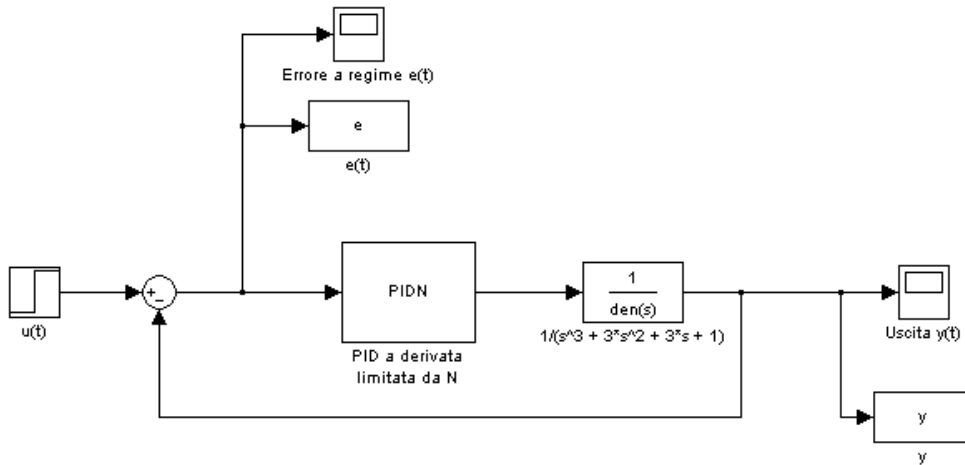
Riportiamo di seguito gli schemi realizzati in ambiente grafico simulink:



**Figura 41: Sistema con regolatore PI**

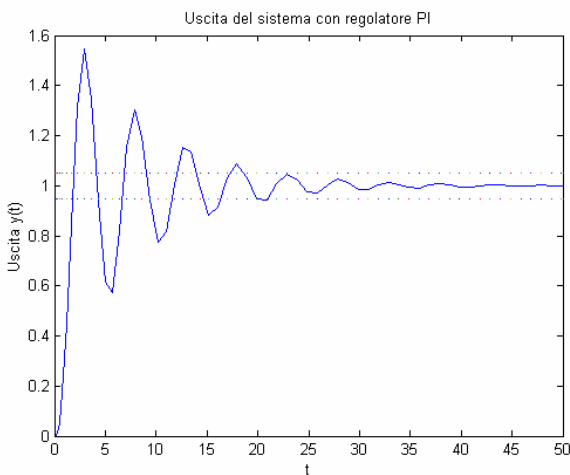


**Figura 42: Sistema con regolatore PID**

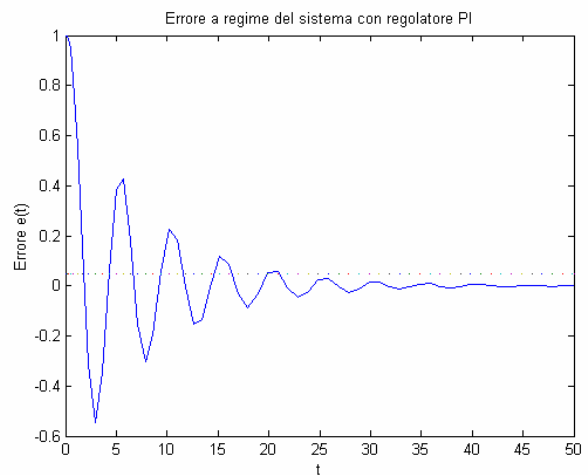


**Figura 43: Sistema con regolatore PID con limitazione della derivata**

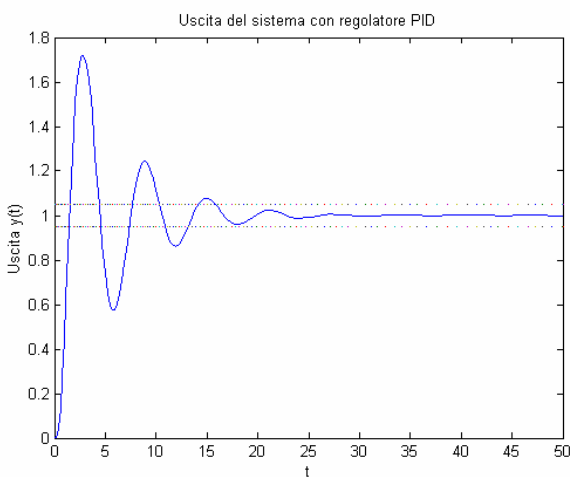
Applicando in ingresso un gradino di ampiezza unitaria ad ognuno dei sistemi costruiti, otteniamo le varie risposte (Figura 44, Figura 46, Figura 48) caratterizzate dai relativi errori a regime (Figura 45, Figura 47, Figura 49):



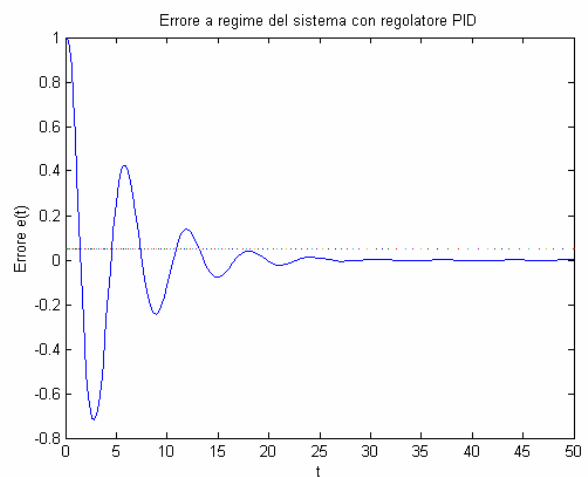
**Figura 44: y(t) del sistema regolato con PI**



**Figura 45: e(t) del sistema regolato con PI**



**Figura 46: y(t) del sistema regolato con PID**



**Figura 47: e(t) del sistema regolato con PID**

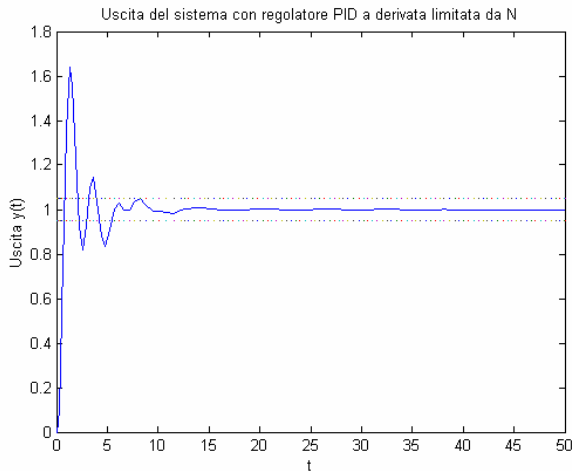


Figura 48:  $y(t)$  del sistema regolato con PID limitato sulla derivata,  $N=10$

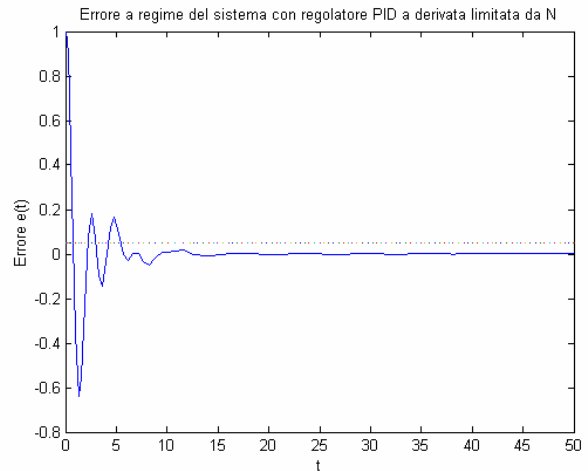


Figura 49:  $e(t)$  del sistema regolato con PID limitato sulla derivata,  $N=10$

Come si può notare nel passaggio da regolatore PI a regolatore PID la velocità del sistema aumenta, ma aumenta anche la sovraelongazione dell'uscita; comunque si ha un sistema più pronto. Nella stessa maniera nel passare al regolatore PID con derivata limitata la risposta del sistema è ancora più veloce, mantenendo circa la stessa sovraelongazione. Comportamento analogo si ha negli errori a regime.

**Quesito 2.**

Abbiamo costruito lo schema di Figura 50 per confrontare il PID in forma reale con un PID con limitazione

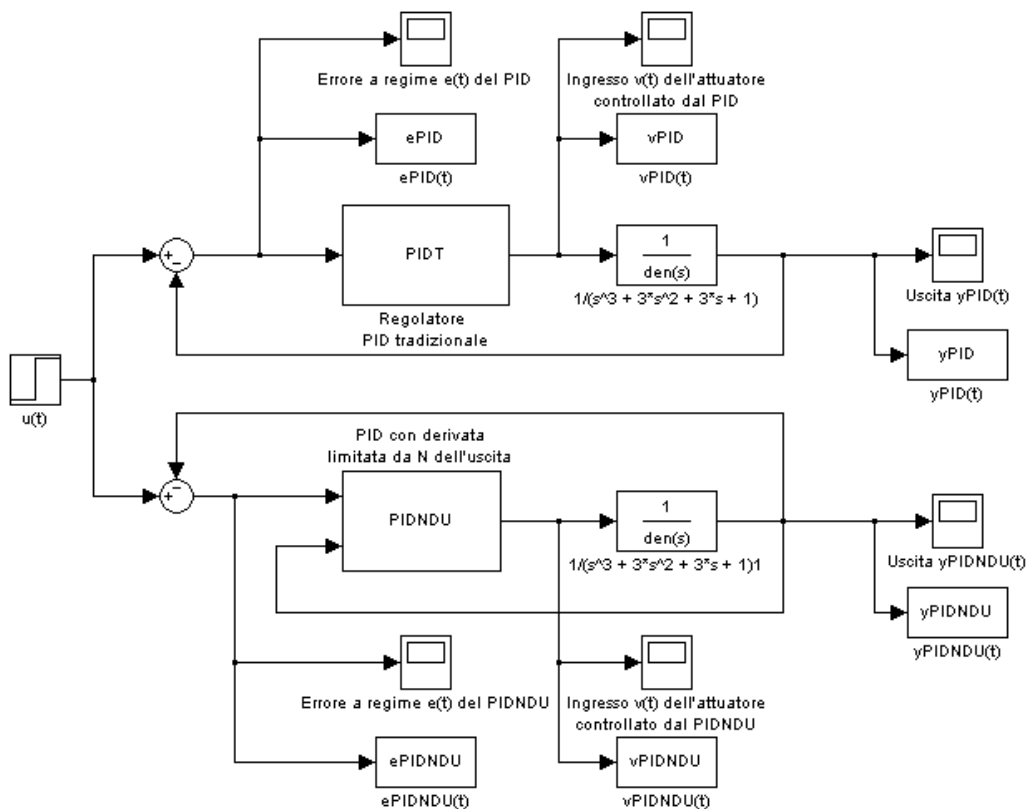
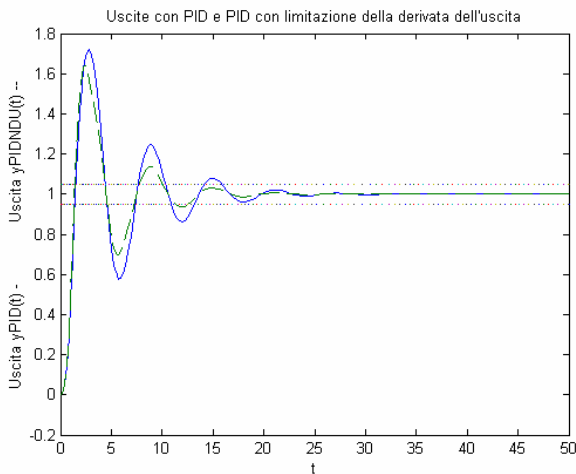
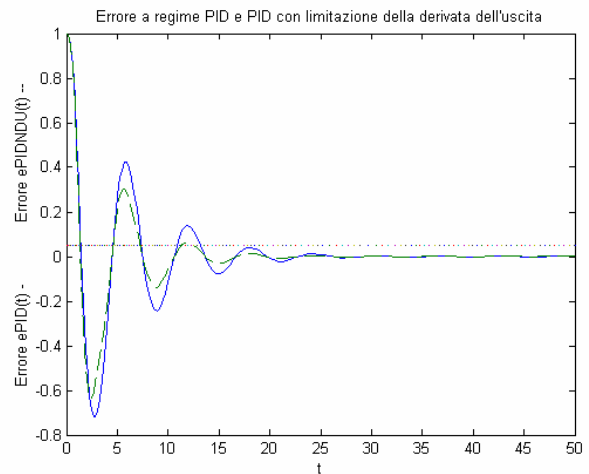


Figura 50: Confronto fra PID e PID con limitazione dell'azione derivativa solo sull'uscita

dell'azione derivativa della sola uscita, ottenendo le risposte  $y(t)$  di Figura 51 e gli errori a regime  $e(t)$  di Figura 52.

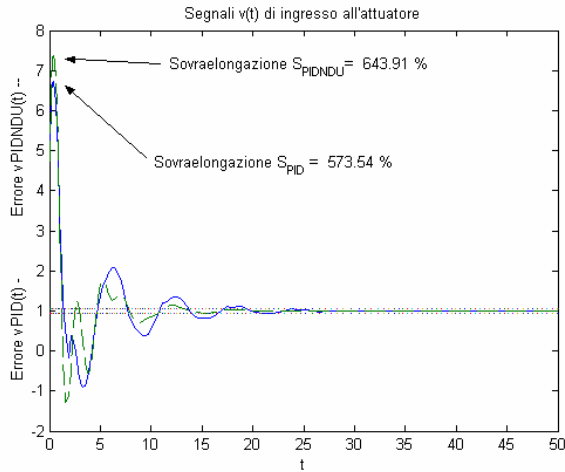


**Figura 51: Confronto  $y(t)$ PID con  $y(t)$ PIDNDU**

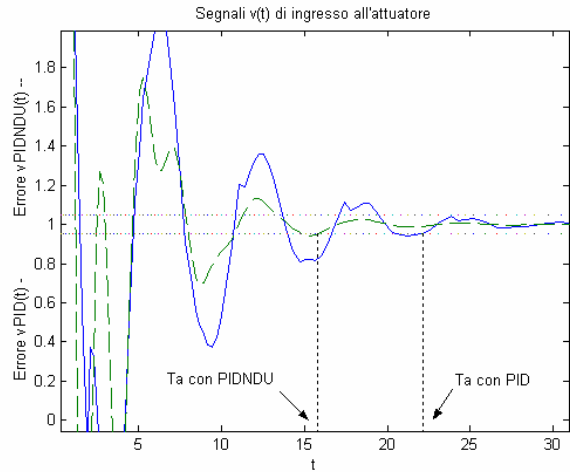


**Figura 52: Confronto  $e(t)$ PID con  $y(t)$ PIDNDU**

Di particolare interesse il segnale  $v(t)$  in ingresso all'attuatore. Sebbene il PID con limitazione dell'azione derivativa dell'uscita risulti più pronto, nel seguire il gradino unitario di ingresso, rispetto al PID reale (Figura 54/Figura 53), esso paga con una superiore sovralongazione (Figura 53).



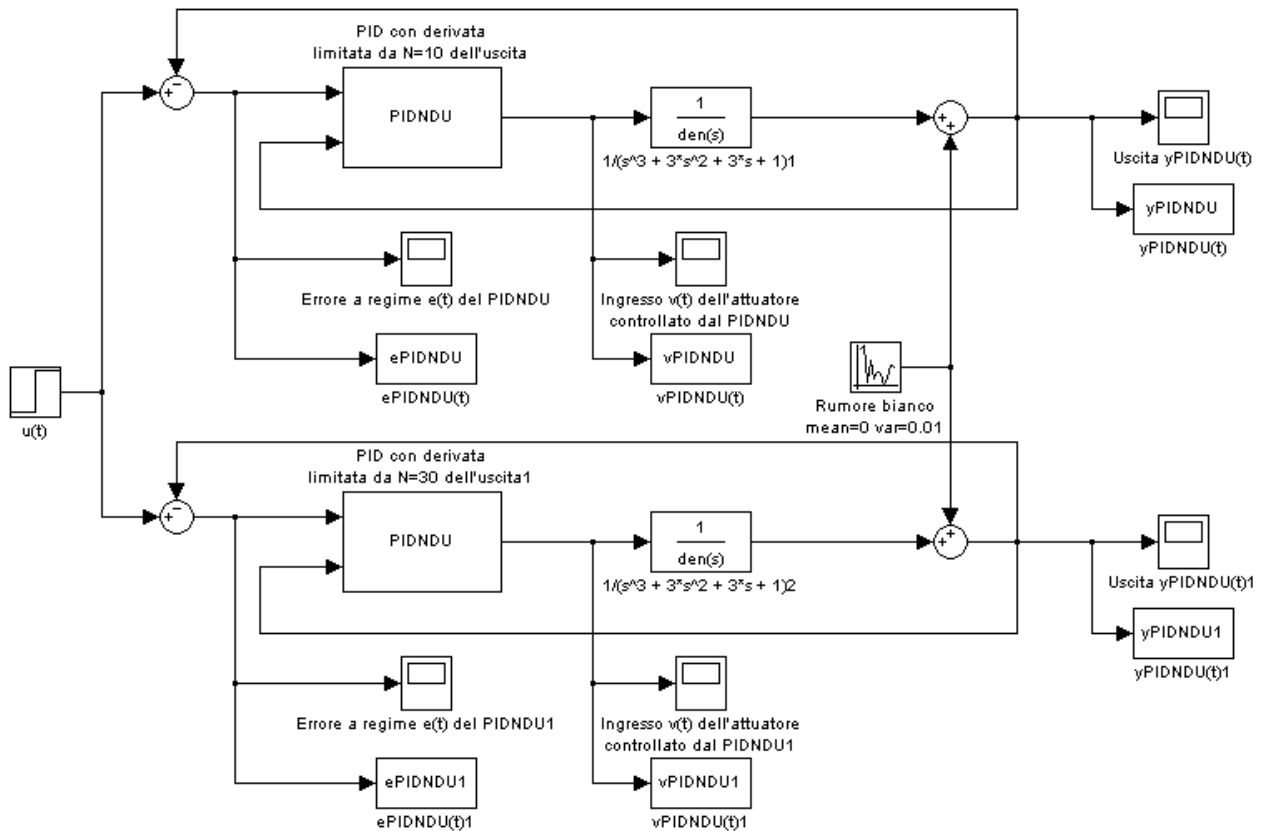
**Figura 53: Confronto  $v(t)$ PID con  $v(t)$ PIDNDU**



**Figura 54: Zoom del confronto  $e(t)$ PID con  $y(t)$ PIDNDU**

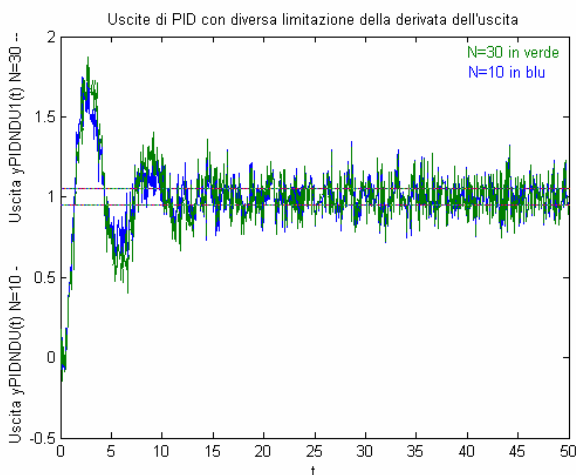
**Quesito 3.**

Col circuito di Figura 55 confrontiamo due controllori PID con diversa limitazione dell'azione derivativa sulla sola uscita:  $N=10$  e  $N=30$ . Inoltre introduciamo, prima della retroazione, del rumore bianco con varianza 0.01.

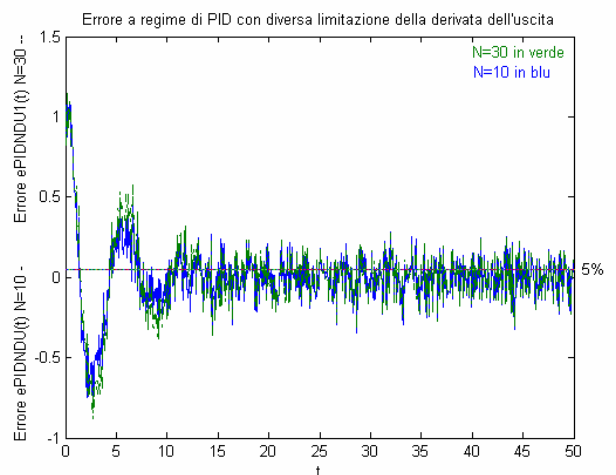


**Figura 55: Confronto di diverse attenuazioni,  $N=10$  e  $N=30$ , in presenza di rumore**

Otteniamo le risposte  $y(t)$  di Figura 56 caratterizzate dall'errore a regime  $e(t)$  di Figura 57:

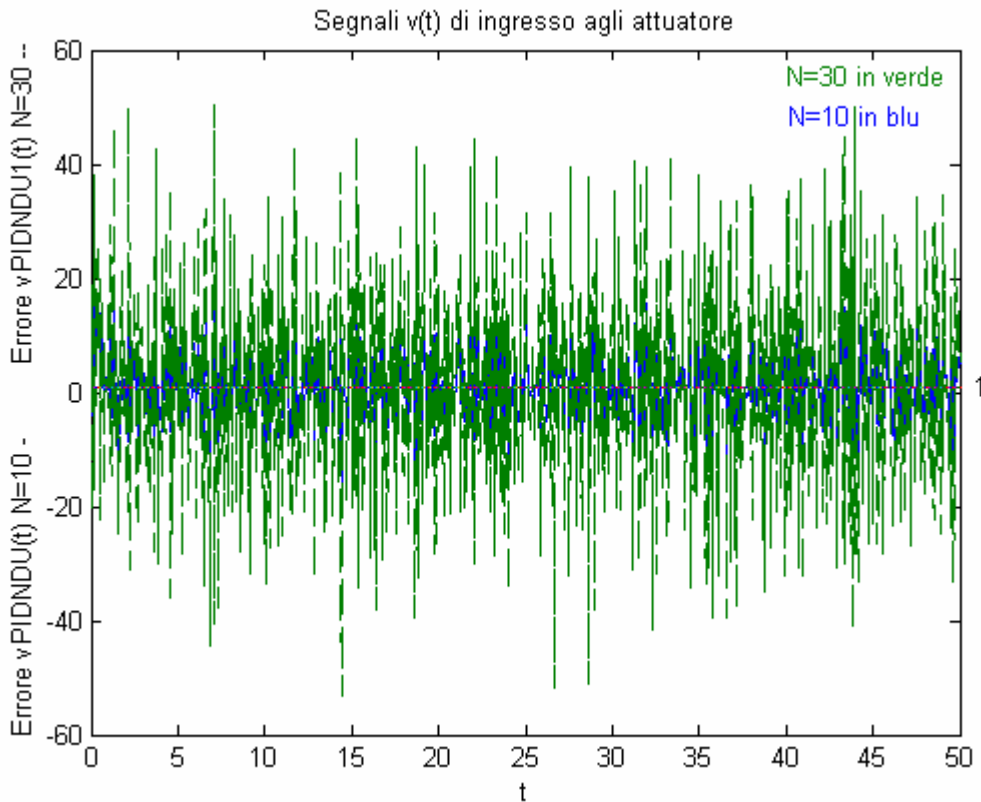


**Figura 56:  $y(t)$ PIDNDU con  $N=30$  ed  $N=10$**



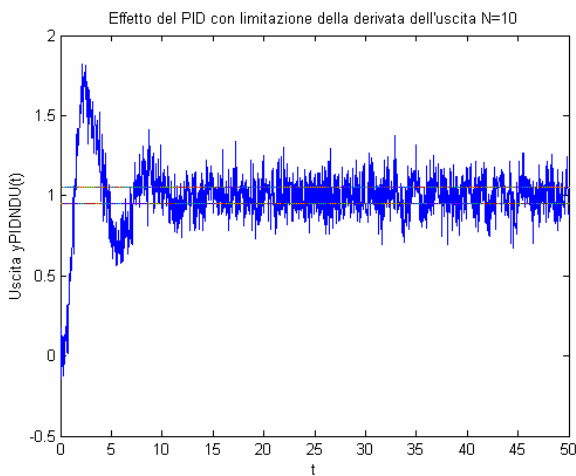
**Figura 57:  $e(t)$ PIDNDU con  $N=30$  ed  $N=10$**

Interessante è l'andamento del segnale  $v(t)$  in ingresso all'attuatore. Come si può osservare (Figura 58) il contributo del termine derivativo, in seguito all'applicazione del gradino in ingresso, introduce sollecitazioni potenzialmente dannose per gli organi di attuazione:

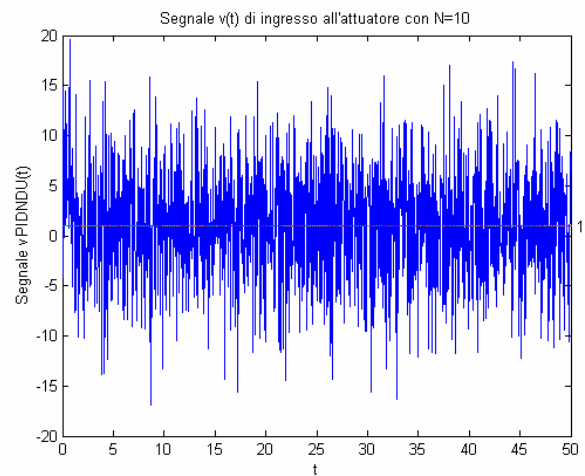


**Figura 58:  $v(t)$ PIDNDU con  $N=30$  ed  $N=10$**

In particolare considerati differenti regolatori con attenuazioni  $N=10$ ,  $N=50$ ,  $N=100$ , otteniamo le risposte di Figura 59, Figura 61, Figura 63 in seguito all'intervento dell'attuatore pilotato dai relativi ingressi in Figura 60, Figura 62, Figura 64:

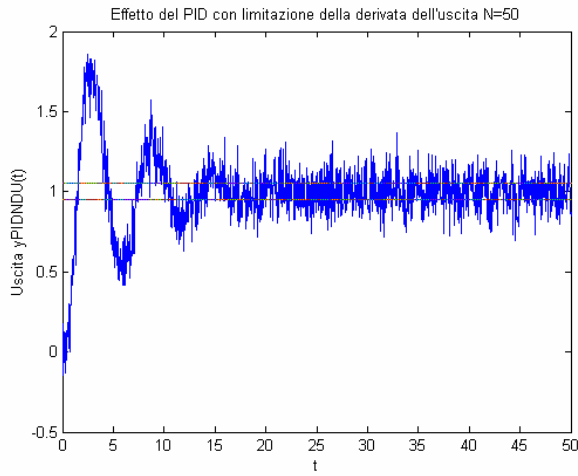


**Figura 59: Uscita  $y(t)$ PIDNDU con  $N=10$**

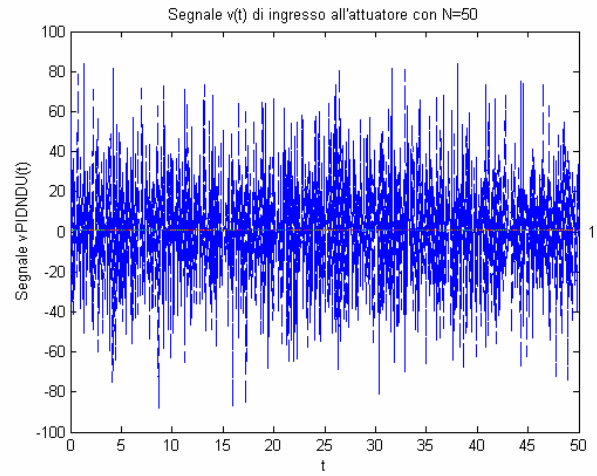


**Figura 60: Ingresso attuatore  $v(t)$ PIDNDU con  $N=10$**

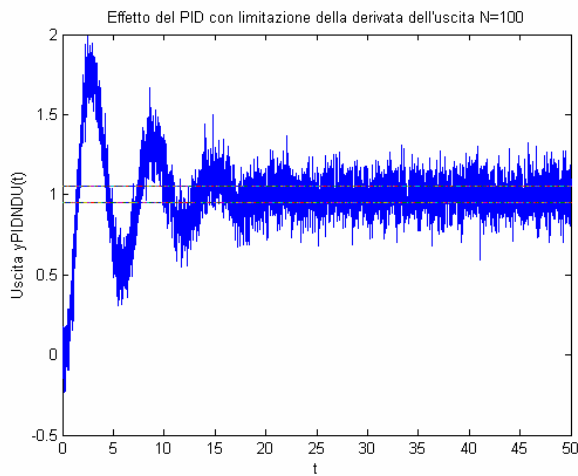




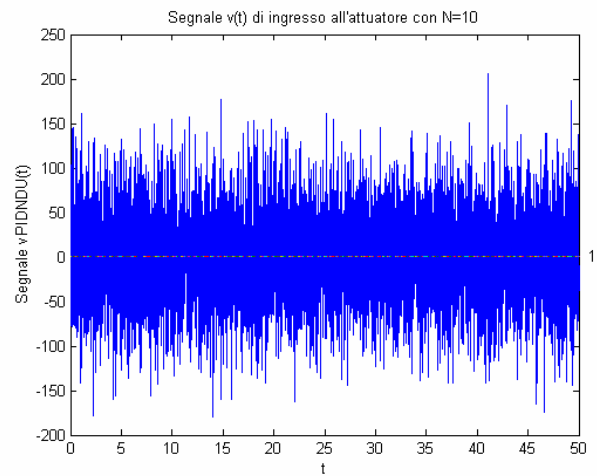
**Figura 61: Uscita  $y(t)$ PIDNDU con  $N=50$**



**Figura 62: Ingresso attuatore  $v(t)$ PIDNDU con  $N=50$**



**Figura 63: Uscita  $y(t)$ PIDNDU con  $N=100$**



**Figura 64: Ingresso attuatore  $v(t)$ PIDNDU con  $N=100$**

Il sistema al crescere di  $N$  diviene più pronto, ma presenta un errore che cresce come ampiezza, rendendo problematico il progetto dell'attuatore.



This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.