

```
%%  
%% ESERCIZIO 1  
%%  
%% Punto 1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
>>A1=[-4,-3,0,3;0,-3,-1,1;0,-2,-3,2;0,-2,-1,0]
```

L'inserimento di una matrice avviene editando tra parentesi [] i valori separati da spazi o virgole. Il ";" segnala la terminazione della riga nella matrice.

```
A1 =  
    -4    -3     0     3  
     0    -3    -1     1  
     0    -2    -3     2  
     0    -2    -1     0
```

```
>>B1=[0;0;0;1]
```

```
B1 =  
     0  
     0  
     0  
     1
```

```
>>C1=[1,1,0,0]
```

```
C1 =  
     1     1     0     0
```

```
>>s1=ss(A1,B1,C1,0)
```

Tramite questo assegnamento, Matlab alloca nella variabile s1 le informazioni circa il sistema dinamico definito dalle matrici A, B, C, D.

```
a =  
      x1  x2  x3  x4  
x1  -4  -3   0   3  
x2   0  -3  -1   1  
x3   0  -2  -3   2  
x4   0  -2  -1   0
```

```
b =  
      u1  
x1   0  
x2   0  
x3   0  
x4   1
```

```
c =  
      x1  x2  x3  x4  
y1   1   1   0   0
```

```
d =  
      u1  
y1   0
```

Continuous-time model.

```
>> eig(s1)
```

```
ans =
```

```
-4.0000  
-3.0000  
-1.0000  
-2.0000
```

La parte reale degli autovalori del sistema è strettamente minore di 0. Il sistema risulta quindi asintoticamente stabile.

```
%%% Punto 2. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
>>x0=[1;0;1;2]
```

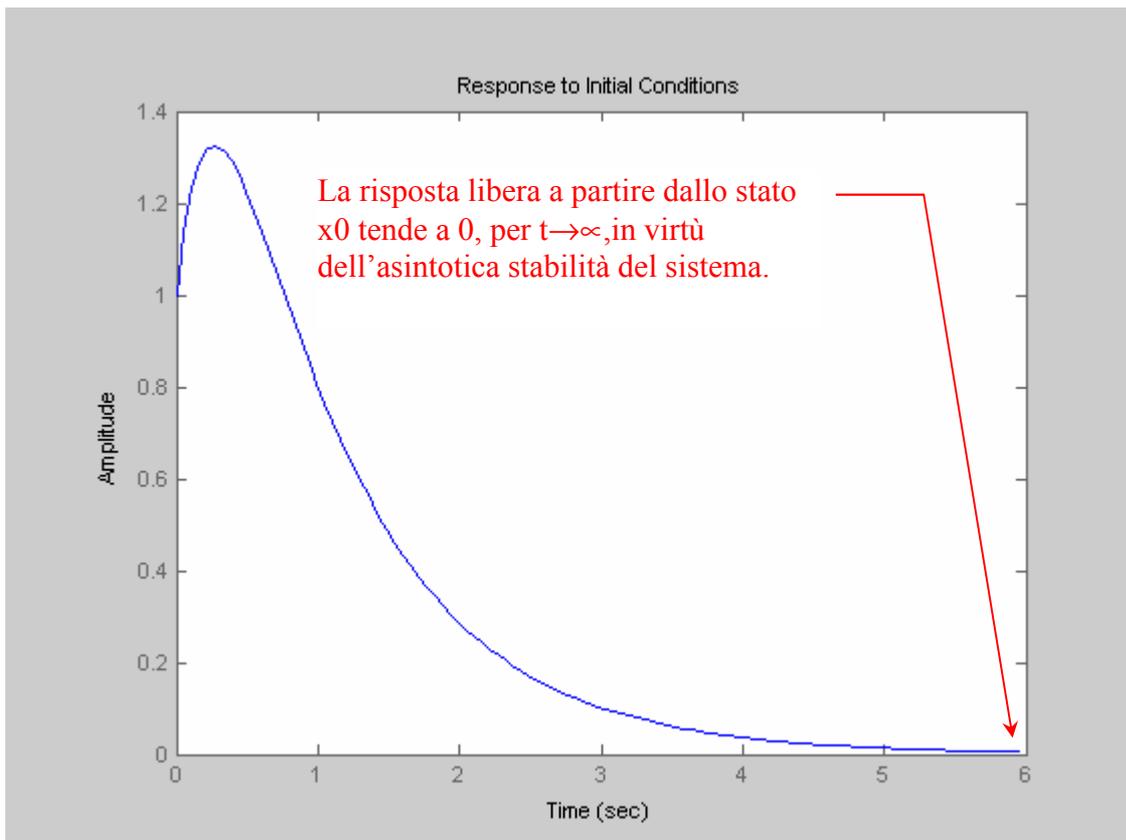
```
x0 =
```

```
1  
0  
1  
2
```

La funzione initial viene chiamata passandole come parametri s1, ossia la variabile che contiene la descrizione del sistema, ed il vettore delle condizioni iniziali x0. Il risultato della simulazione può essere visualizzato in figura, oppure può essere memorizzato per manipolazioni successive, assegnando i vettori y (uscita), t (tempo simulato) ed x (stato) con il comando :

```
>>initial(s1,x0)
```

```
[y,t,x] = initial(sist,x0)
```



```
%%% Punto 3. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
>>dcgain(s1)
```

```
ans =
```

```
0.9167
```

```
>>-C1*inv(A1)*B1
```



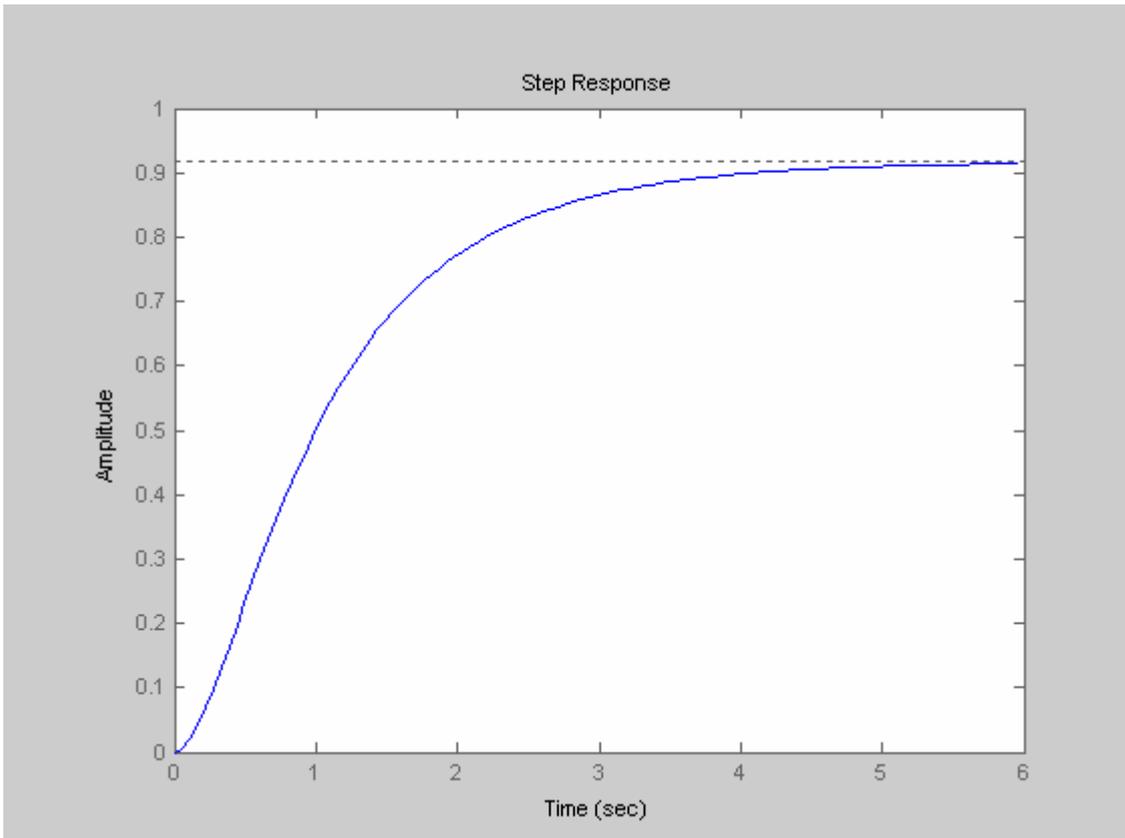
Dalla teoria:  
 $Y(s)=G(s)U(s)$  con  
 $G(s)=C(sI_n-A)^{-1}B+D$   
Per  $s \rightarrow 0$  ( $t \rightarrow \infty$ ),  
 $G(0)=-C(A)^{-1}B+D$   
 $Y(0)=G(0)U(0)$

```
ans =
```

```
0.9167
```

```
%%% Punto 4. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
>>step(s1)
```



```
%%%
%%% ESERCIZIO 2
%%%
%%% Punto 1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

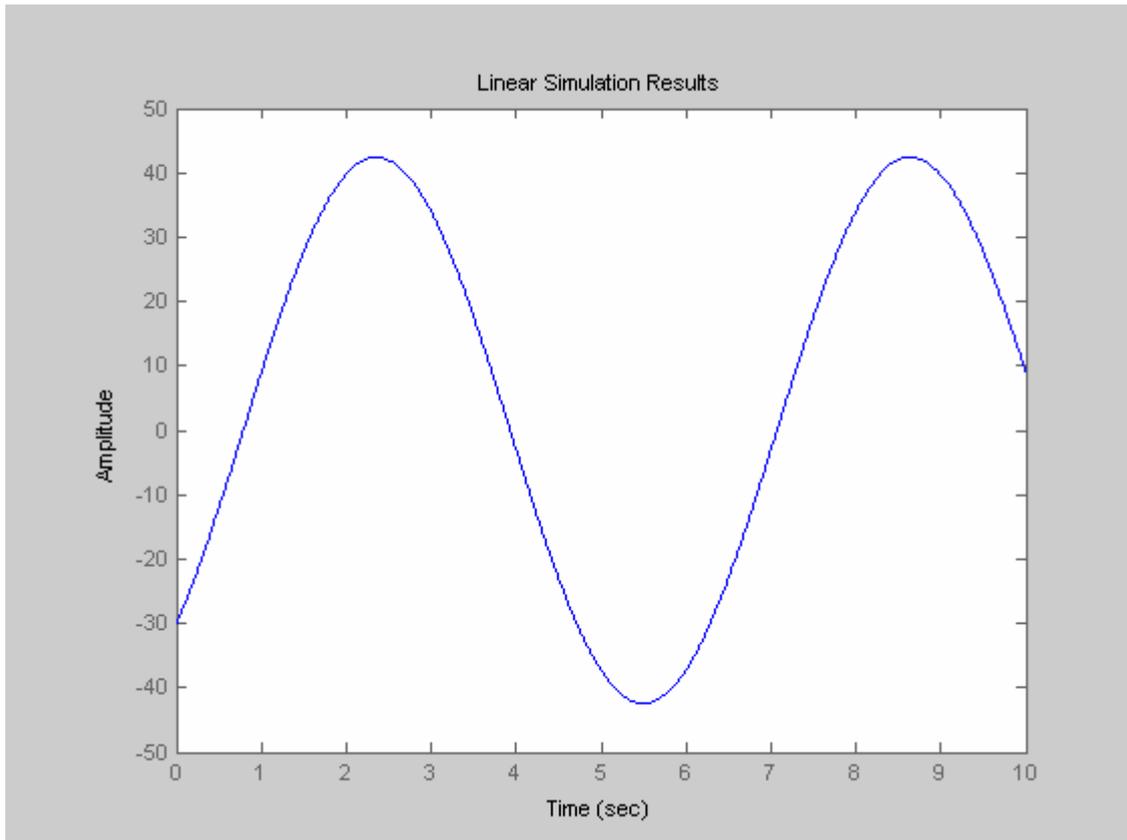
```
>>s2=ss(-1,5,3,0)
```

```
a =
    x1
x1 -1
```

```
b =
    u1
x1  5
```

```
c =
    x1
```





```

%%
%%
%% ESERCIZIO 3
%%
%% Punto 1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
>>s31=tf(3*[1] , [2 1])
```

```

Transfer function:
      3
-----
 2 s + 1

```

La funzione tf ha come parametri due vettori che contengono i coefficienti dei polinomi N(s) e D(s) in ordine decrescente. N(s) e D(s) rappresentano numeratore e denominatore della f.d.t.

```
>>s32=tf(3*[1 1] , [2 1])
```

```

Transfer function:
 3 s + 3
-----
 2 s + 1

```

I sistemi considerati sono del primo ordine, con il medesimo polo. Applicando il teorema del valore iniziale e del valore finale si vede come il guadagno statico coincida, mentre i valori iniziali differiscono (rispettivamente 0, 3/2 e -3/2).

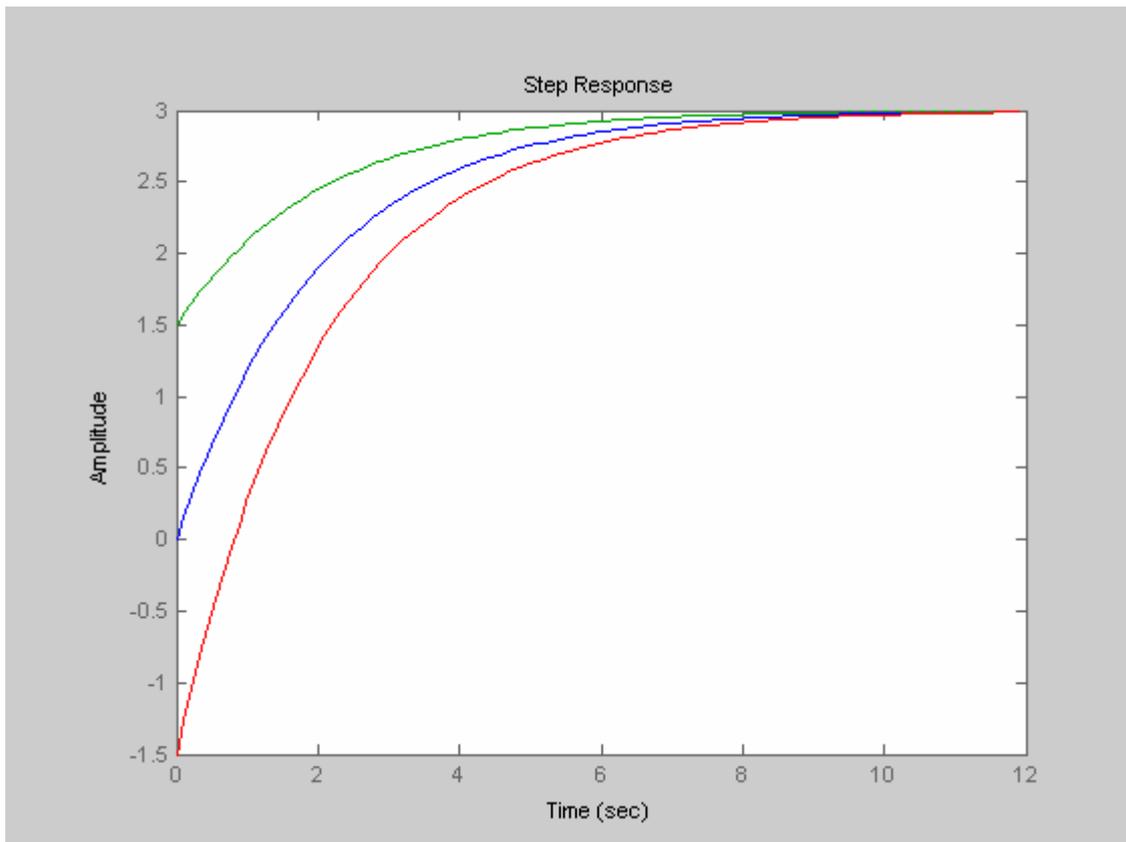
```
>>s33=tf(3*[-1 1] , [2 1])
```

```

Transfer function:
-3 s + 3
-----
 2 s + 1

```

```
>>step(s31,s32,s33)
```



%%% Punto 2. %%

```
>>s34=tf(3,[10 11 1])
```

Transfer function:  
3

$$10 s^2 + 11 s + 1$$

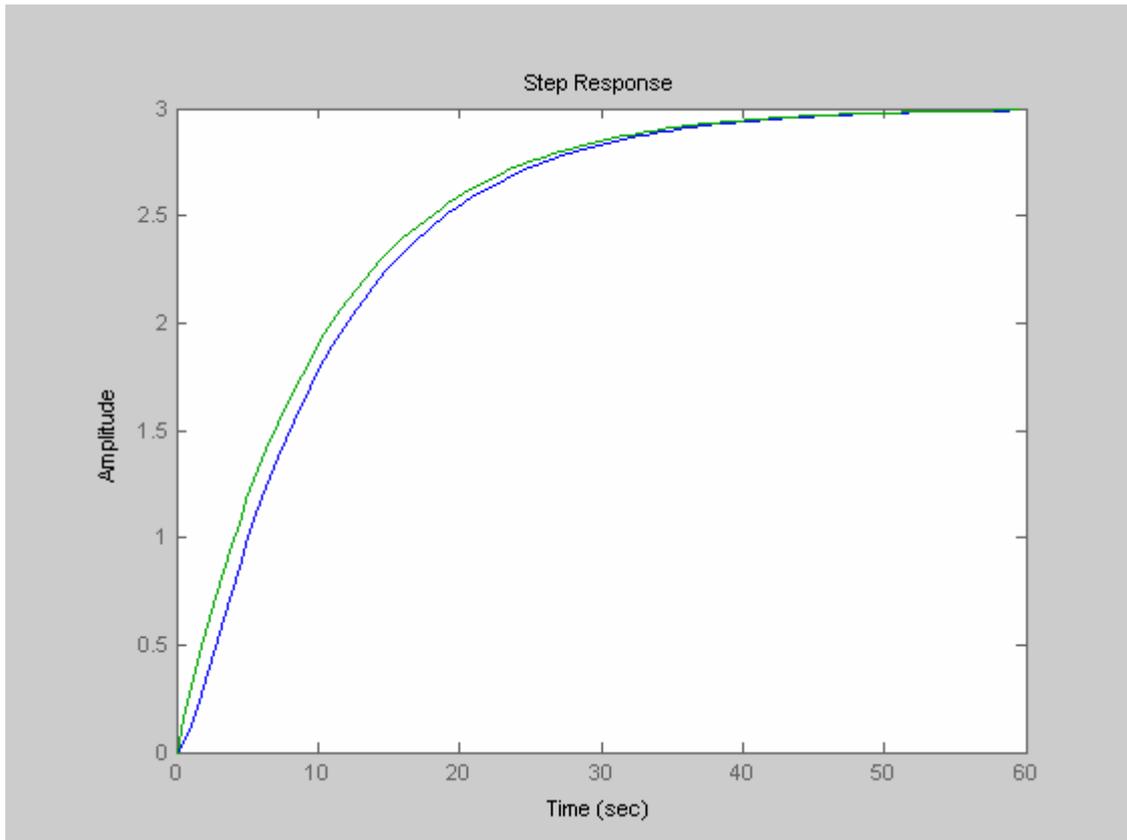
```
>>s35=tf(3,[10 1])
```

Transfer function:  
3

$$10 s + 1$$

```
>>step(s34,s35)
```

I sistemi considerati hanno un polo in comune. Applicando il teorema del valore iniziale e del valore finale si determinano guadagno statico (3) ed il valore iniziale (0). Si osservi la dinamica iniziale più lenta del sistema s34, dovuta alla costante di tempo  $T1=1 \ll T2=10$ .



%%% Punto 3. %%

```
>>s36=tf(3,[100 20 1])
```

Transfer function:  
3

-----  
 $100 s^2 + 20 s + 1$

```
>>s37=tf(3*[30 1],[100 20 1])
```

Transfer function:  
90 s + 3

-----  
 $100 s^2 + 20 s + 1$

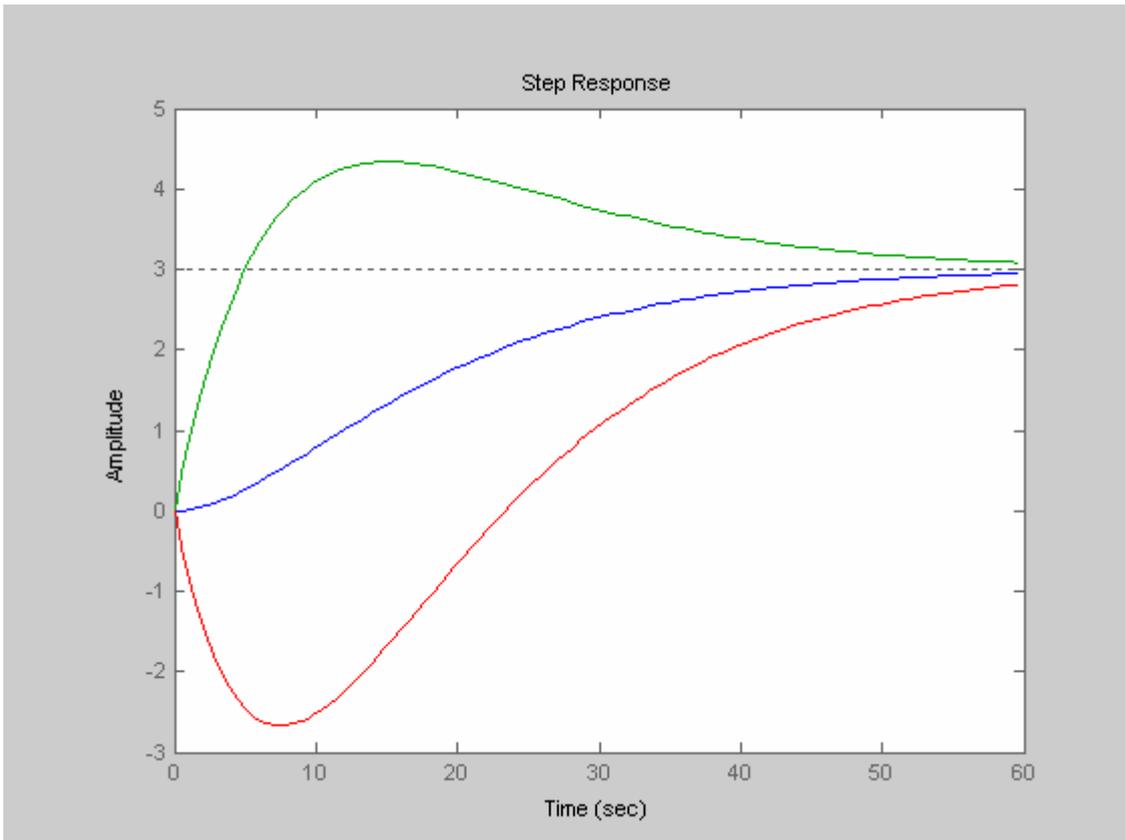
```
>>s38=tf(3*[-30 1],[100 20 1])
```

Transfer function:  
-90 s + 3

-----  
 $100 s^2 + 20 s + 1$

```
>>step(s36,s37,s38)
```

I sistemi considerati hanno gli stessi poli. Mentre il primo sistema non ha zeri, il secondo ed il terzo presentano uno zero, con ugual modulo e di segno opposto. Si osservi che il transitorio, nella parte iniziale, è di segno opposto alla sollecitazione in ingresso (il tratto di risposta inversa è dovuto allo zero nel semipiano destro).



```

%%
%% ESERCIZIO 4
%%
%% Punto 1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

>>wn=log(100)/(10*0.2)

```

←  $T_a = \ln 100 / (\zeta \omega_n)$   
 Affinché il tempo di assestamento della  
 risposta allo scalino al 99% valga circa 10 s è  
 necessario che:  
 $\omega_n = \ln 100 / (\zeta T_a)$

```

wn =
    2.3026

```

```

%% Punto 2. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

>>s4=tf(3*wn^2,[1 0.4*wn wn^2])

```

Transfer function:  
 15.91  
 -----  
 $s^2 + 0.921 s + 5.302$

```

%% Punto 3. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

>>step(s4)

```

