

Inserimento in Matlab della funzione di trasferimento:

$$G(s) = 0.2 \frac{(1-2s)}{s(1+10s)(1+0.1s)}$$

```
>> s=tf('s')
```

```
Transfer function:
```

```
s
```

```
>> Gs= 0.2*(1-2*s)/(s*(1+0.1*s)*(1+10*s))
```

```
Transfer function:
```

```
  -0.4 s + 0.2
```

```
-----  
s^3 + 10.1 s^2 + s
```

```
%%%  
%%% Definizione dei vettori usati in Simulink  
%%%  
%%%  
%%%
```

```
>> [numGs,denGs]=tfdata(Gs,'v')
```

```
numGs =
```

```
      0      0  -0.4000   0.2000
```

```
denGs =
```

```
  1.0000  10.1000   1.0000      0
```

```
%%%  
%%% calcolo della funzione di trasferimento in retroazione  
%%% unitaria  
%%%  
%%%  
%%%
```

```
>> Gr=Gs/(1+Gs)
```

```
Transfer function:
```

```
  -0.4 s^4 - 3.84 s^3 + 1.62 s^2 + 0.2 s
```

```
-----  
s^6 + 20.2 s^5 + 103.6 s^4 + 16.36 s^3 + 2.62 s^2 + 0.2 s
```

```
>> [numGr,denGr]=tfdata(Gr,'v')
```

```
numGr =
```

```
      0      0  -0.4000  -3.8400   1.6200   0.2000
```

```
0
```

```
denGr =
```

```
0      1.0000    20.2000    103.6100    16.3600    2.6200    0.2000
```

```
%%%  
%%% Verifica della stabilità del sistema in retroazione unitaria  
%%%
```

```
>> roots(denGr)
```

```
ans =
```

```
      0  
-10.0422  
-10.0000  
-0.0289 + 0.1381i  
-0.0289 - 0.1381i  
-0.1000
```

```
%%%  
%%% A parte un polo nell'origine, che garantisce l'errore a regime  
%%% nullo in risposta al gradino, tutti gli altri poli hanno parte  
%%% reale negativa, implicando così la stabilità del sistema  
%%% in retroazione unitaria  
%%%
```

```
>>
```

```
%%%  
%%% calcolo del tempo di assestamento e massima sovraelongazione  
%%% e tempo di assestamento, una volta generati in Simulink i  
%%% vettori ync e t  
%%%
```

```
>> lsiminfo(ync,t)
```

Introduzione della rete correttiva:

$$R(s) = \frac{(1+10s)}{(1+0.1s)}$$

```
>> numRs=[10 1]
```

```
numRs =
```

```
    10     1
```

```
>> denRs=[0.1 1]
```

```
denRs =
```

```

    0.1000    1.0000
>> Rs=tf(numRs,denRs)

Transfer function:
10 s + 1
-----
0.1 s + 1

>>

%%%
%%% Calcolo della funzione di trasferimento del sistema
%%% compenstato in retroazione
%%%

>> Grcs=(Rs*Gs)/(1+Rs*Gs)

Transfer function:
      -0.4 s^6 - 7.88 s^5 - 37.56 s^4 + 12.72 s^3 + 3.64 s^2
+ 0.2 s
-----
0.01 s^8 + 0.402 s^7 + 5.68 s^6 + 33.32 s^5 + 70.5 s^4 + 33.12 s^3
+ 4.64 s^2 + 0.2 s

>>

%%%
%%% Verifica della stabilità della funzione di trasferimento del
%%% sistema compenstato in retroazione
%%%

>> [numGrcs,denGrcs]=tfdata(Grcs,'v')

numGrcs =
    3.6400    0.2000    0.0000   -0.4000   -7.8800  -37.5640   12.7220
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

denGrcs =
    4.6400    0.2000    0.0100    0.4020    5.6801   33.3240   70.4960   33.1220
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

>>

>> roots(denGrcs)

ans =
     0
 -16.4201
 -10.0000
 -10.0000
  -3.1991
  -0.3807
  -0.1000

```

-0.1000

>>

%%%

%% A parte un polo nell'origine, che garantisce l'errore a regime
%% nullo in risposta al gradino, tutti gli altri poli hanno parte
%% reale negativa, implicando così la stabilità del sistema
%% compensato in retroazione unitaria

%%%

%%%

%% Calcolo del tempo di assestamento e massima sovraelongazione
%% e tempo di assestamento, una volta generati in Simulink i
%% vettori yc e t del sistema compensato in retroazione unitaria

%%%

>> lsiminfo(yc,t)