

## Esercizio in Lab. Info Grande

10 aprile 2017

Assegnato il seguente modello dell'impianto:

$$G(s) = \frac{1}{s(s+10)(s+50)}$$

usando il luogo delle radici, si progetti un regolatore proporzionale per il sistema chiuso in retroazione, affinché il sistema complessivo presenti le seguenti caratteristiche in risposta al gradino unitario:

$$\begin{cases} S\% \leq 7.5\% \quad (\delta \geq 0.63) \\ T_a \leq 0.4s. \end{cases}$$

Nel caso in cui tali requisiti non dovessero essere rispettati, si determini il valore del guadagno  $K$  della seguente rete correttiva (anticipatrice) in modo da soddisfare le specifiche sopra:

$$R(s) = K \frac{1+s/15}{1+s/100}$$

In Matlab si introduce il modello del sistema da compensare:

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

Transfer function:

s

```
>> Gs=1/(s*(s+10)*(s+50))
```

Transfer function:

```
1
-----
s^3 + 60 s^2 + 500 s
```

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

numGs =

```
0 0 0 1
```

denGs =

```
1 60 500 0
```

e si disegna il luogo delle radici:

```
>> rlocus(Gs)
```

Può essere utile determinare il guadagno critico del sistema chiuso in retroazione unitaria negativa usando la funzione Matlab:

```
>> Ku=rlocfind(Gs)
```

Select a point in the graphics window

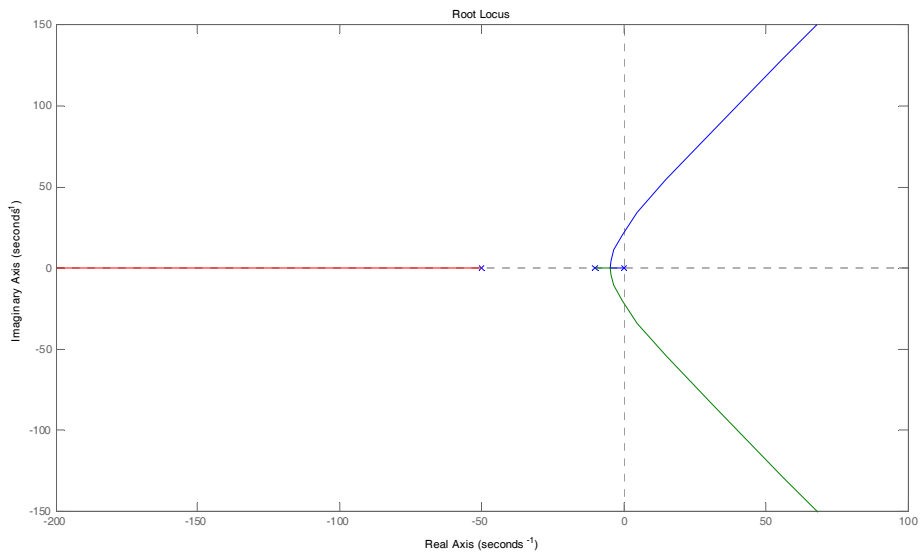
selected\_point =

```
0.0219 +22.2384i
```

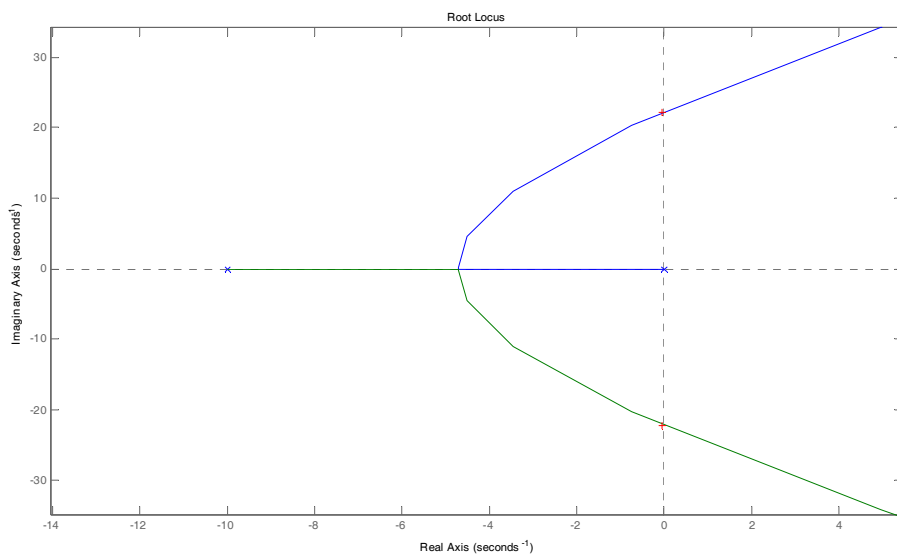
Ku =

2.9695e+004

>>

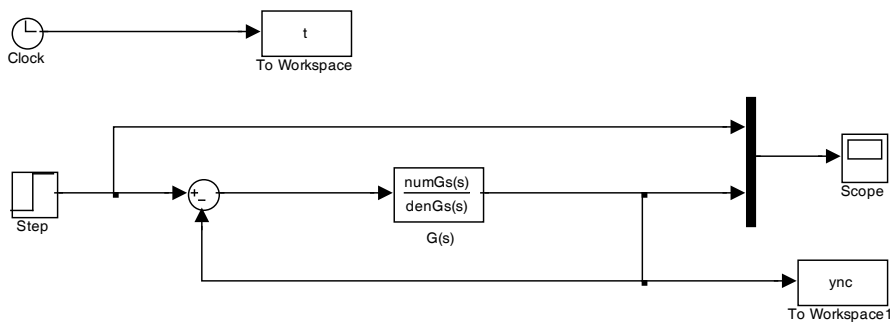


Che determina il seguente grafico, una volta determinati col mouse e il puntatore i punti di intersezione del luogo delle radici con l'asse delle ordinate:



Tale valore approssimativo di  $K_u$  corrisponde al guadagno di retroazione che porterebbe il sistema in retroazione ad avere una risposta oscillatoria persistente al gradino unitario.

In Simulink si realizza il sistema in retroazione unitaria sotto, e si inseriscono i blocchi “To Workspace” per esportare le variabili che verranno usate per il calcolo del tempo di assestamento e sovraelongazione.



Impostato un tempo di simulazione ad esempio di 3000 s., tali parametri in transitorio vengono determinati dalla funzione `lsiminfo` Matlab nel seguente modo:

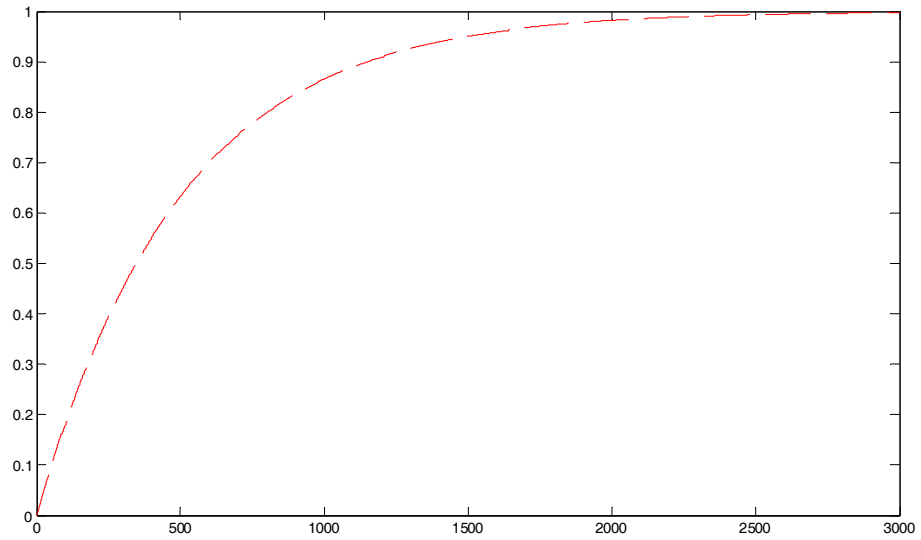
```
>> lsiminfo(ync,t)
```

```
ans =
```

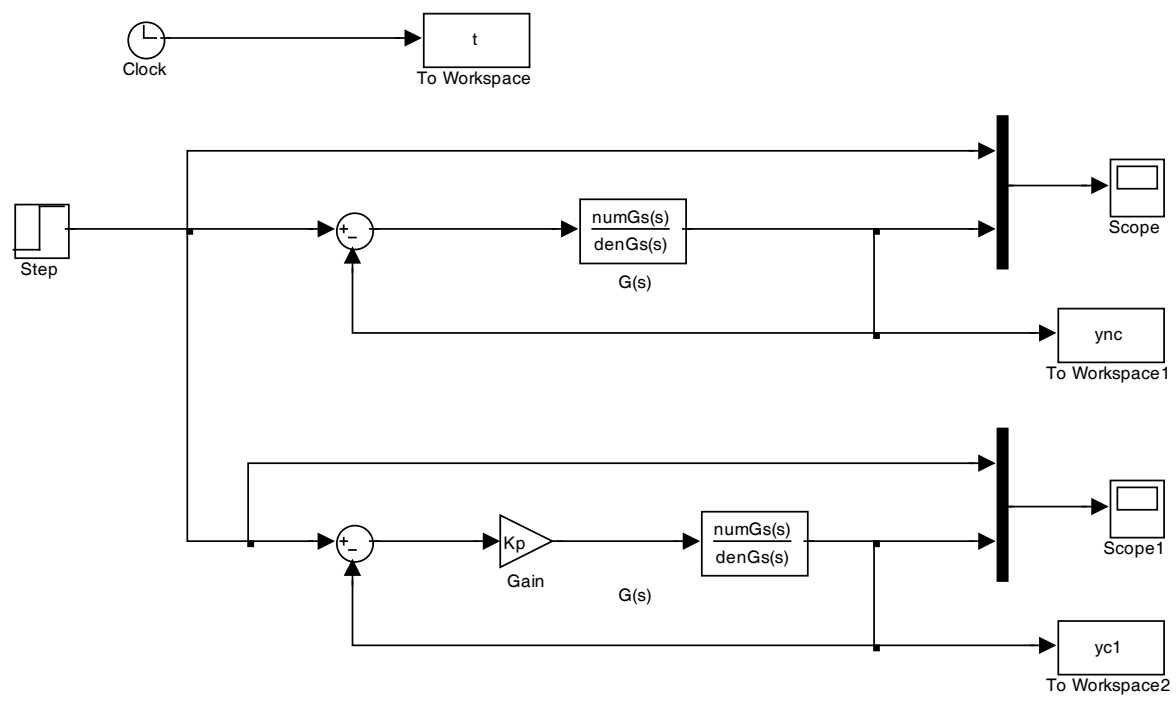
```
SettlingTime: 1.8984e+003  
Min: 0  
MinTime: 0  
Max: 0.9975  
MaxTime: 3000
```

```
>>
```

Si ha quindi un sistema molto lento, senza sovraelongazione, con tempo di assestamento di 1898 s. circa. La risposta del sistema in retroazione unitaria è riportata nel seguito.



È ragionevole quindi provare a rendere più veloce tale sistema in retroazione tentando un primo intervento costituito da un semplice regolatore proporzionale, il cui guadagno  $K_p$  deve essere opportunamente determinato, in modo da soddisfare le specifiche definite all'inizio, secondo lo schema Simulink definito come segue.



Il guadagno  $K_p$  deve essere determinato in base alle specifiche e usando il luogo delle radici in cui vengono evidenziati i punti a sovralongazione costante (cioè i luoghi a  $\delta$  costante), definiti come segue, ed usando opportunamente lo strumento di ingrandimento della finestra grafica del luogo delle radici.

```
>> rlocus(Gs)
>> sgrid
>> Kp=rlocfind(Gs)
Select a point in the graphics window
```

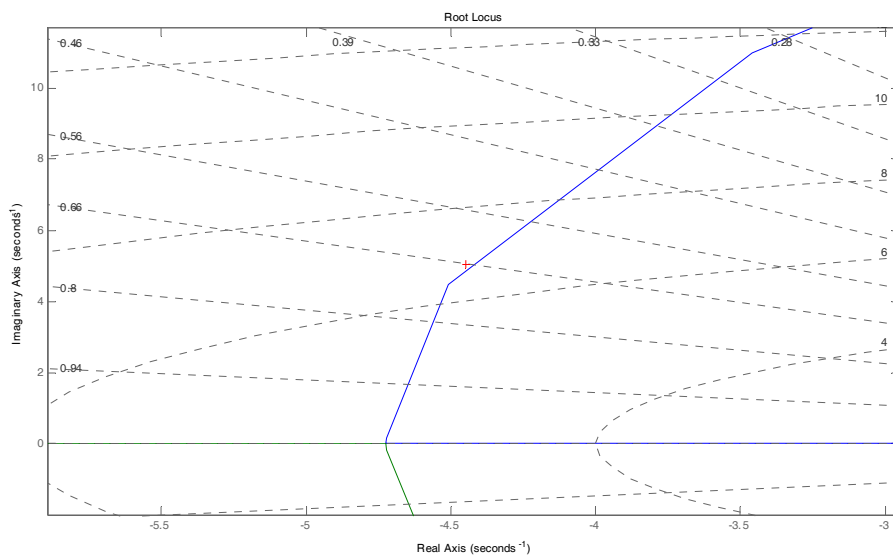
```
selected_point =

-4.4169 + 5.0497i
```

```
Kp =

2.3162e+003
```

```
>>
```



Si determina come primo tentativo un  $K_p \cong 2316$ , che simulato nel sistema in Simulink mi porta a questi risultati di tempo di assestamento e massima sovralongazione:

```
>> lsiminfo(yc1,t)
```

```
ans =
```

```
SettlingTime: 0.9143  
           Min: 0  
           MinTime: 0  
           Max: 1.0622  
           MaxTime: 0.6336
```

```
>>
```

Si osservi che la risposta è stata resa notevolmente più veloce rispetto al sistema chiuso in retroazione con guadagno unitario, e la durata della simulazione può essere ridotta a 3 s.

Si vede che il tempo di assestamento non è soddisfatto (deve essere almeno di 0.4s.) mentre la sovraelongazione è verificata ( $\leq 7.5\%$ ). Si può procedere quindi per tentativi aumentando il guadagno proporzionale  $K_p$  per tentativi, al fine di vedere di soddisfare entrambe le specifiche.

Dopo alcuni tentativi (2350, 2400, 2450, dal momento che, in general, un aumento del guadagno proporzionale, comporta ad una diminuzione del tempo di assestamento e un aumento della sovraelongazione), si trova ad esempio che:

```
>> Kp=2450
```

```
Kp =
```

```
2450
```

```
>> lsiminfo(yc1,t)
```

```
ans =
```

```
SettlingTime: 0.8885  
           Min: 0  
           MinTime: 0  
           Max: 1.0724  
           MaxTime: 0.6316
```

```
>>
```

ovvero  $T_a = 0.89$  s. e  $S\% = 7.2\%$  .

Provando poi col valore di  $K_p = 2500$  si ha:

```
>> Kp=2500
```

```
Kp =
```

```
2500
```

```
>> lsiminfo(yc1,t)
```

```
ans =
```

```
SettlingTime: 0.8782
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.0755
```

```
MaxTime: 0.6309
```

```
>>
```

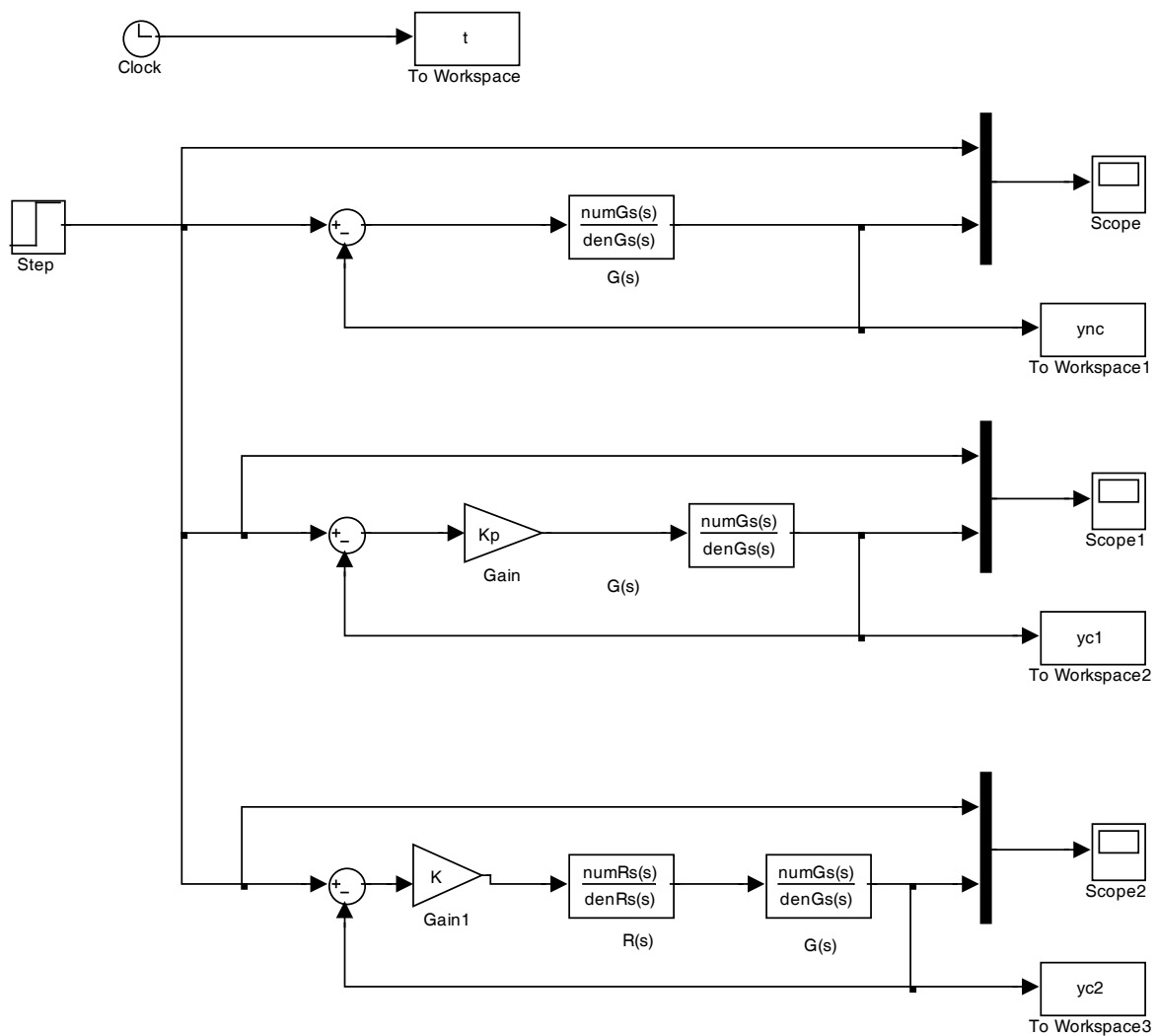
cioè vi ha la violazione anche del tempo di assestamento minimo. Questi risultati portano a constatare che con un semplice guadagno proporzionale risulta impossibile soddisfare le specifiche richieste dal progetto.

Si definisce quindi la rete correttiva come suggerito all'inizio:

$$R(s) = \frac{1 + s/15}{1 + s/100}$$

che ha guadagno in continua unitario ( $R(0)=1$ ) e viene considerato un guadagno  $K$  per la rete correttiva (esterno al blocco Simulink corrispondente ad  $R(s)$ ) come definito nello schema Simulink che segue.





Tale guadagno  $K$  sarà determinato nuovamente usando il luogo delle radici, ma applicato al nuovo sistema da compensare, con guadagno di anello  $R(s) \cdot G(s)$ . Si inserisce quindi la funzione di trasferimento del regolatore, che verrà usata anche in Simulink:

```
>> Rs=(1+s/15)/(1+s/100)
```

Transfer function:

$$100 s + 1500$$

-----

$$15 s + 1500$$

```
>> [numRs,denRs]=tfdata(Rs,'v')
```

```
numRs =
```

```
100 1500
```

```
denRs =
```

```
15 1500
```

```
>>
```

Per comodità viene definita la funzione guadagno di anello  $G_a(s) = R(s) \cdot G(s)$ :

```
>> Ga=Rs*Gs
```

```
Transfer function:
```

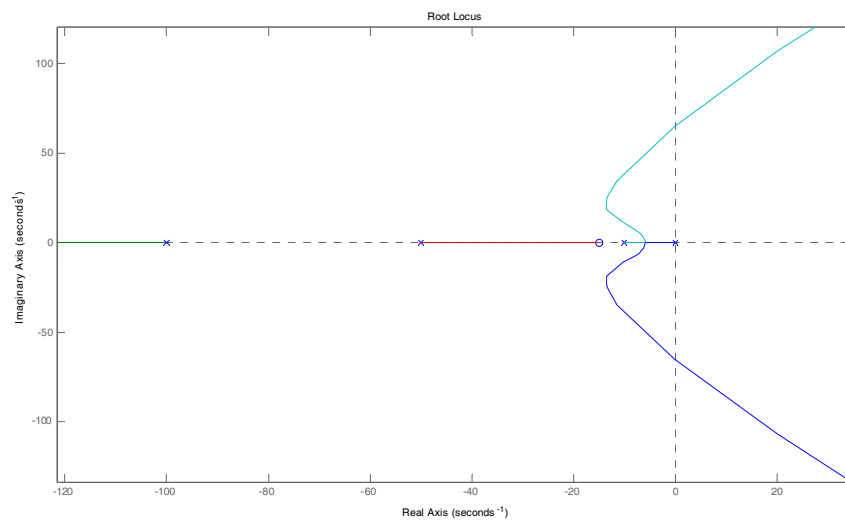
```
100 s + 1500
```

```
-----  
15 s^4 + 2400 s^3 + 97500 s^2 + 750000 s
```

```
>>
```

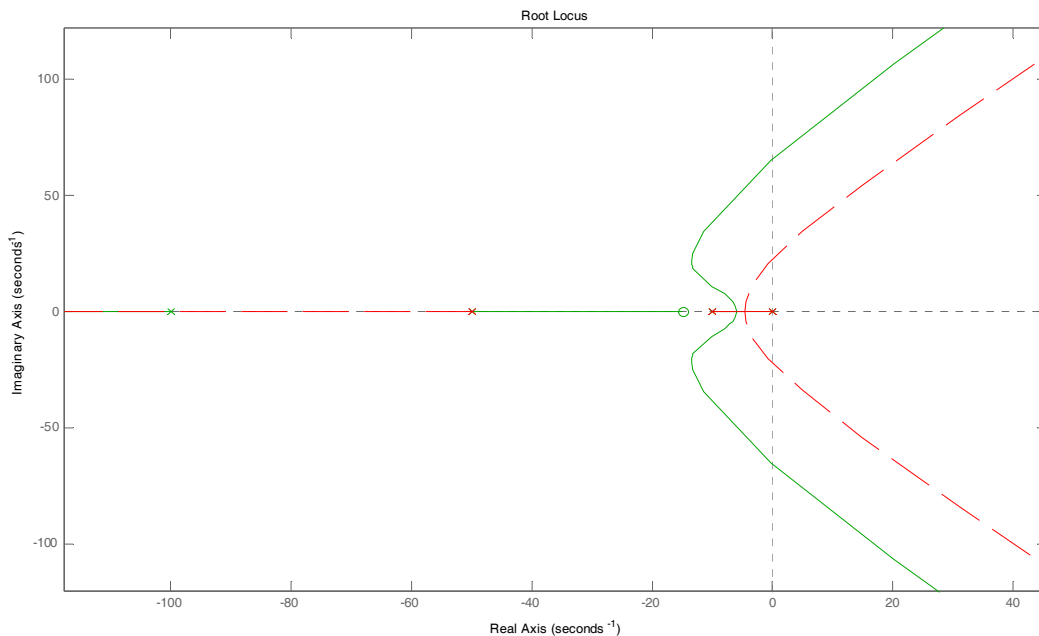
Il cui luogo delle radici risulta:

```
>> rlocus(Ga)
```



Si osservi l'effetto "deformante" introdotto dallo zero della rete correttiva. Lo zero della rete agisce come attrattore del luogo delle radici originale. In maniera intuitiva, ricordando i luoghi dei punti a tempo di assestamento costante, risulta anche ragionevole supporre che possa esistere un valore di  $K$  che consente di assegnare al nuovo sistema compensato caratteristiche in transitorio migliori rispetto a quelle del sistema privo di rete correttiva. Nel seguito si evidenzia il confronto dei luoghi delle radici del sistema senza (in rosso, linea tratteggiata) e con rete correttiva (in verde, linea continua).

```
>> rlocus(Ga, 'g-',Gs, 'r--')
```



Ripeto quindi il procedimento effettuato per l'impianto senza correzione, e lo applico al sistema corretto, ovvero alla funzione guadagno di anello  $G_a(s) = R(s) \cdot G(s)$ . Si agisce in maniera opportuna con ingrandimento del luogo delle radici attorno al luogo dei punti a  $\delta$  costante di interesse per il problema.

```
>> rlocus(Ga)
>> sgrid
>> K=rlocfind(Ga)
Select a point in the graphics window

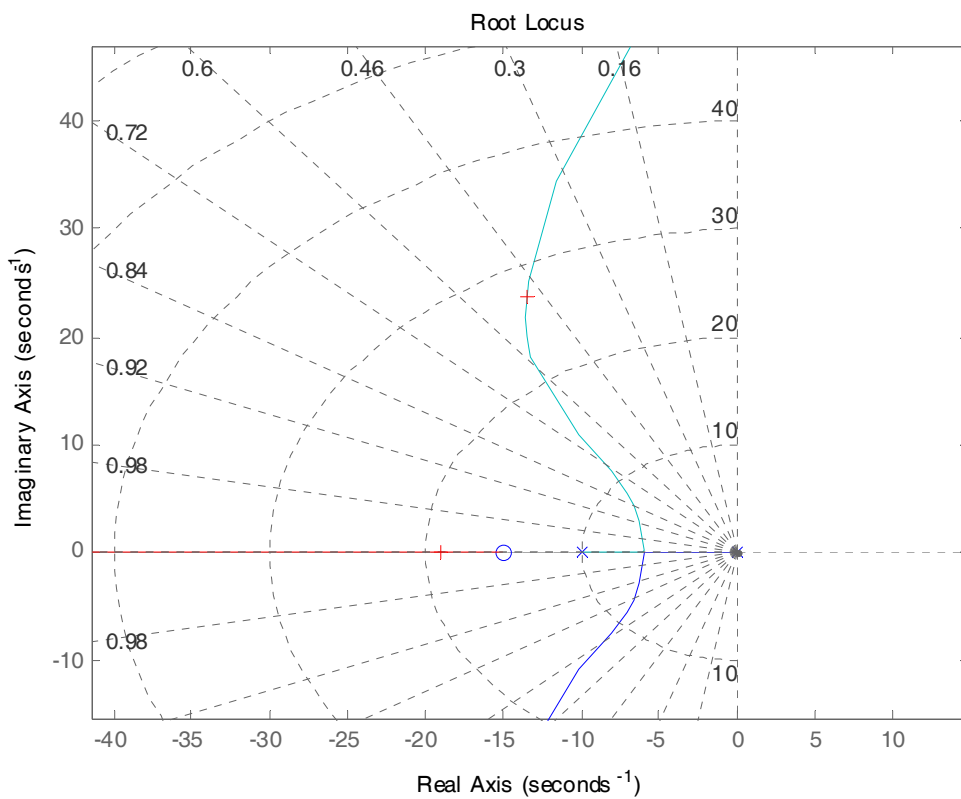
selected_point =
```

-13.3841 +23.8171i

K =

1.6204e+004

>>



E scelgo col puntatore del mouse, una volta chiamata la funzione , un punto in prossimità della retta corrispondente a  $\delta \cong 0.63$  come evidenziato in figura. Tale valore corrisponde al guadagno di primo tentativo  $K \cong 16200$ . Tale valore comporta un tempo di assestamento e sovralongazione pari a:

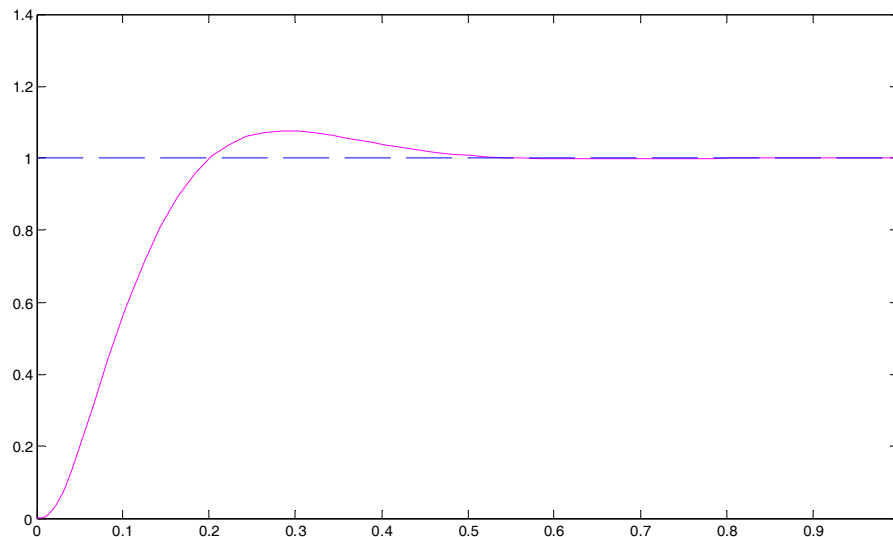
>> lsiminfo(yc2,t)

ans =

```
SettlingTime: 0.3094
      Min: 0
      MinTime: 0
      Max: 1.2510
      MaxTime: 0.1267
```

```
>>
```

Si osserva che questo valore di  $K$  consente di soddisfare il tempo di assestamento (minore di 0.4 s.), ma non la sovraelongazione (almeno 7.5%). Occorre quindi diminuire il guadagno determinato come primo tentativo. Dopo alcune prove per valori 16000, 15500, 14000, 10000, 9000, 7000, 6000, si arriva al valore accettabile finale di  $K = 6050$  che porta ai seguenti risultati, ed alla risposta al gradino riportata sotto.



```
K =
```

```
6050
```

```
>> lsiminfo(yc2,t)
```

```
ans =
```

```
SettlingTime: 0.4511
      Min: 0
```

MinTime: 0  
Max: 1.0755  
MaxTime: 0.2832

Che risultano accettabili ai fini della risoluzione del progetto.