



# Fondamenti di Automatica

**Analisi di Funzioni di Trasferimento:  
Bode, Luogo delle Radici, ecc.  
(Matlab/Simulink + Control Systems)**

Prof. Marcello Bonfè

Dipartimento di Ingegneria - Università di Ferrara

Tel. +39 0532 974839

E-mail: [marcello.bonfe@unife.it](mailto:marcello.bonfe@unife.it)

# Matlab: diagrammi di Bode e luogo delle radici

➔ Funzioni specifiche per la creazione di grafici di analisi per funzioni di trasferimento del Control Systems Toolbox, cioè create con:

>> **tf(num, den)** num/den coefficienti di numeratore e denominatore della FdT

Oppure:

>> **s=tf('s')**

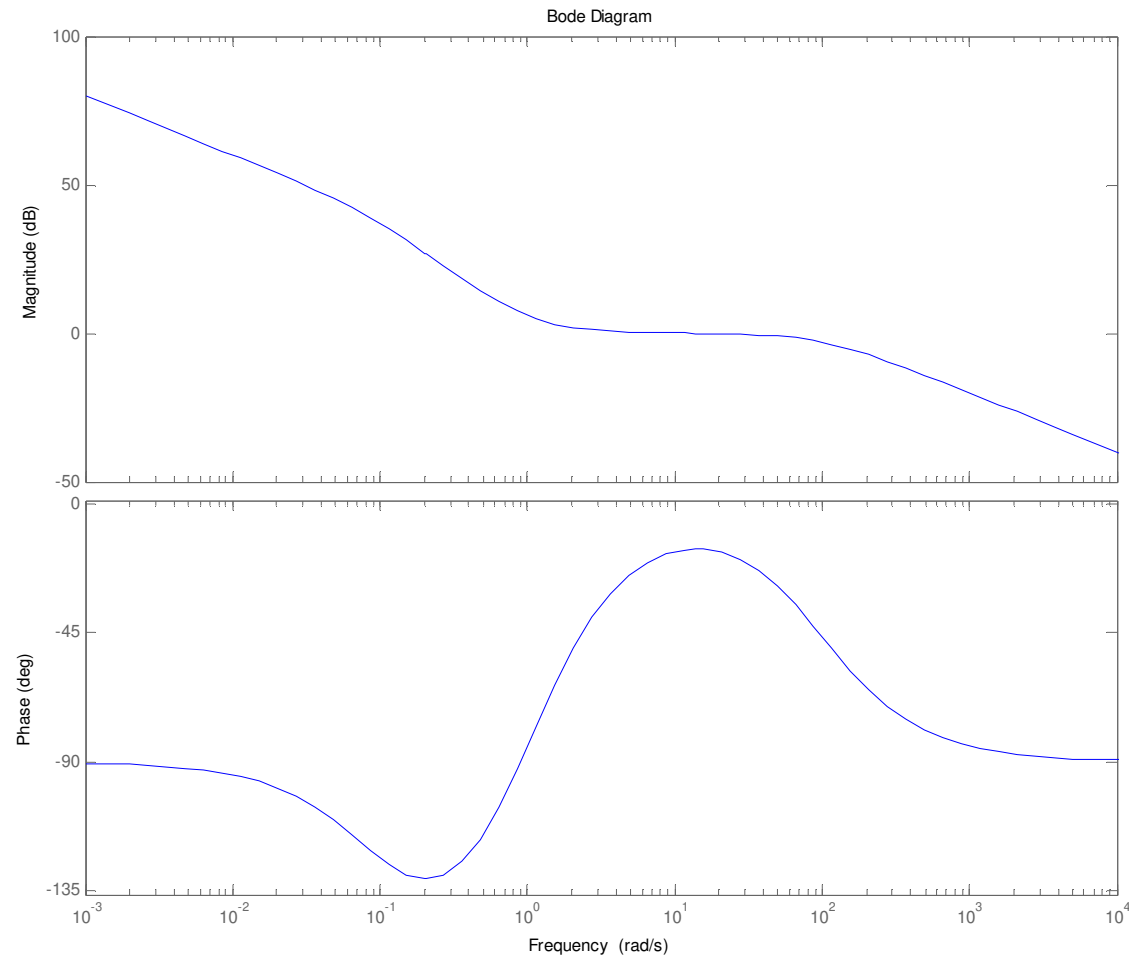
>> **G=10\*(1+s)^2/s/(1+s/0.1)/(1+s/100)**

**NOTA:** l'ultimo comando evidenzia l'utilizzo della fattorizzazione alle costanti di tempo: oltre al polo in 0, gli altri due poli hanno  $\tau_1=1/0.1=10$  e  $\tau_2=1/100=0.01$

# Matlab: diagrammi di Bode

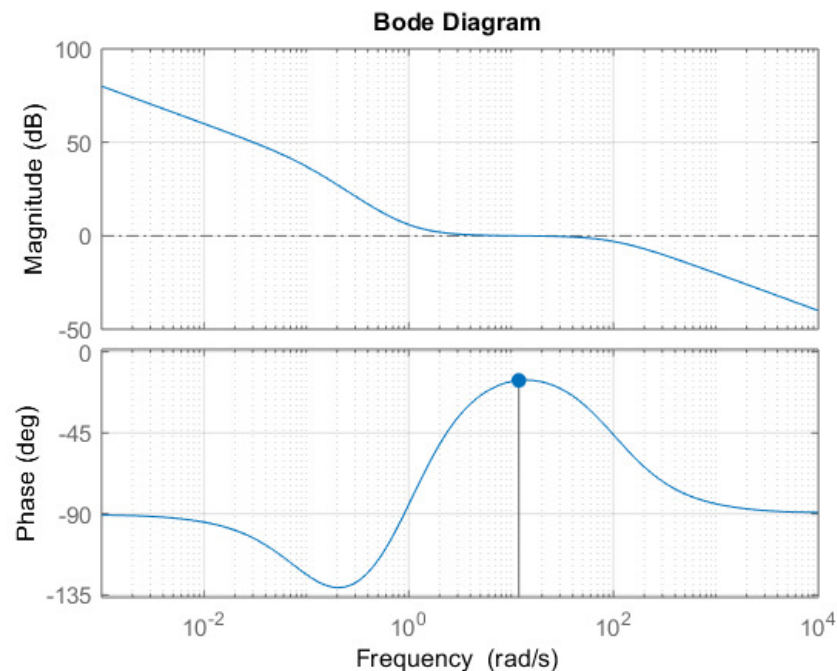
- ➔ Una volta definita la FdT, è immediato visualizzare il corrispondente **diagramma di Bode** (ESATTO!):

```
>> bode (G)
```



# Matlab: diagrammi di Bode

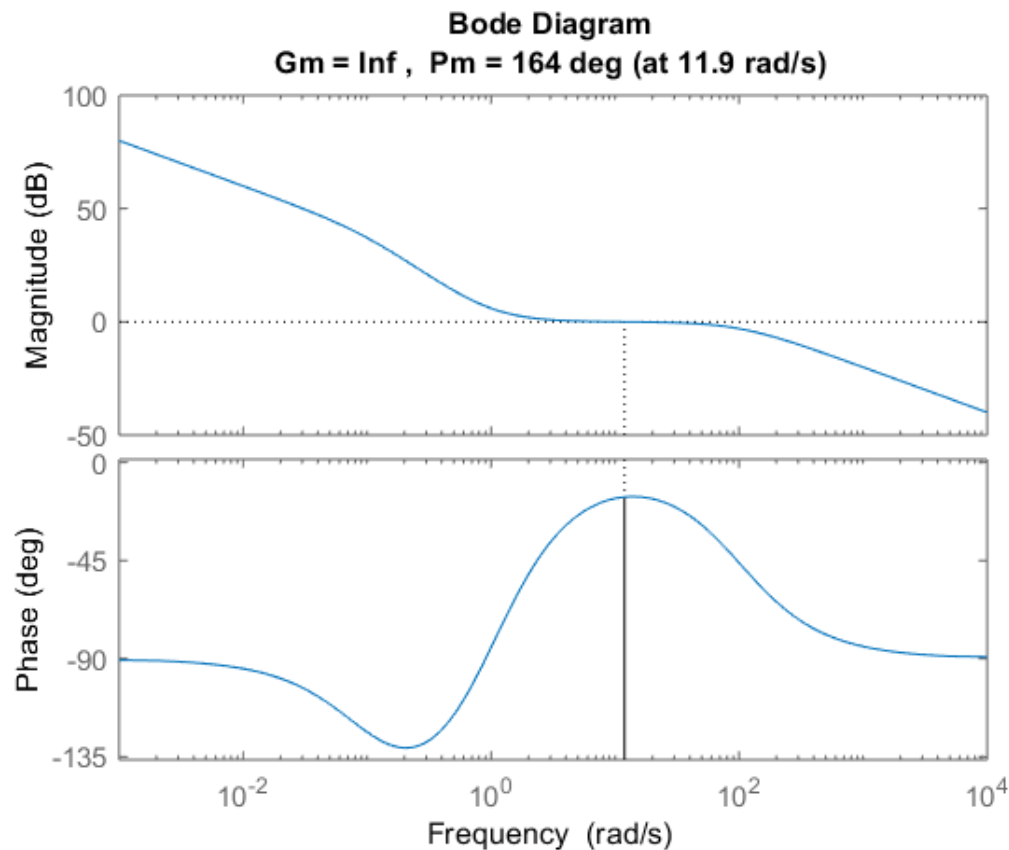
- ➔ Opzioni sul grafico del **diagramma di Bode**:
  - `>> grid on` evidenzia la griglia (in scala logaritmica)
  - Con tasto destro del mouse sul grafico: menu *Characteristics* permette di evidenziare il picco di risonanza (eventuale) e i margini di ampiezza/fase:



# Matlab: diagrammi di Bode

➔ I margini di ampiezza e fase si possono visualizzare direttamente con il comando:

```
>> margin(G)
```



# Matlab: diagrammi di Bode

➔ **I margini di ampiezza e fase** si possono anche memorizzare nel workspace, con il comando (senza plot):

```
>> [Gm, Pm, Wpi, Wc] = margin(G)
```

**Gm** =

**Inf** ➔ Margine di ampiezza in valore assoluto, NON dB

**Pm** =

**164.0857** ➔ Margine di fase

**Wpi** =

**NaN** ➔ Pulsazione  $\omega_{\pi}$  associata al margine di amp.

**Wc** =

**11.8875** ➔ Pulsazione  $\omega_c$  associata al margine di fase

# Matlab: diagrammi di Bode

➔ Anche i **valori numerici della risposta armonica** si possono memorizzare nel workspace (senza plot), con:

```
>> [mag, phase] = bode (G, omega)
```

**omega** è un vettore (opzionale) delle pulsazioni richieste

**mag** è il vettore delle ampiezze assolute, NON dB\*

**phase** è il vettore delle fasi (in gradi)

**\*NOTA:** la scala del grafico delle ampiezze di **bode ()** è invece in dB!! Per allineare i risultati, sono disponibili le funzioni **db2mag ()** e **mag2db ()** ..

# Matlab: diagrammi di Bode approssimati

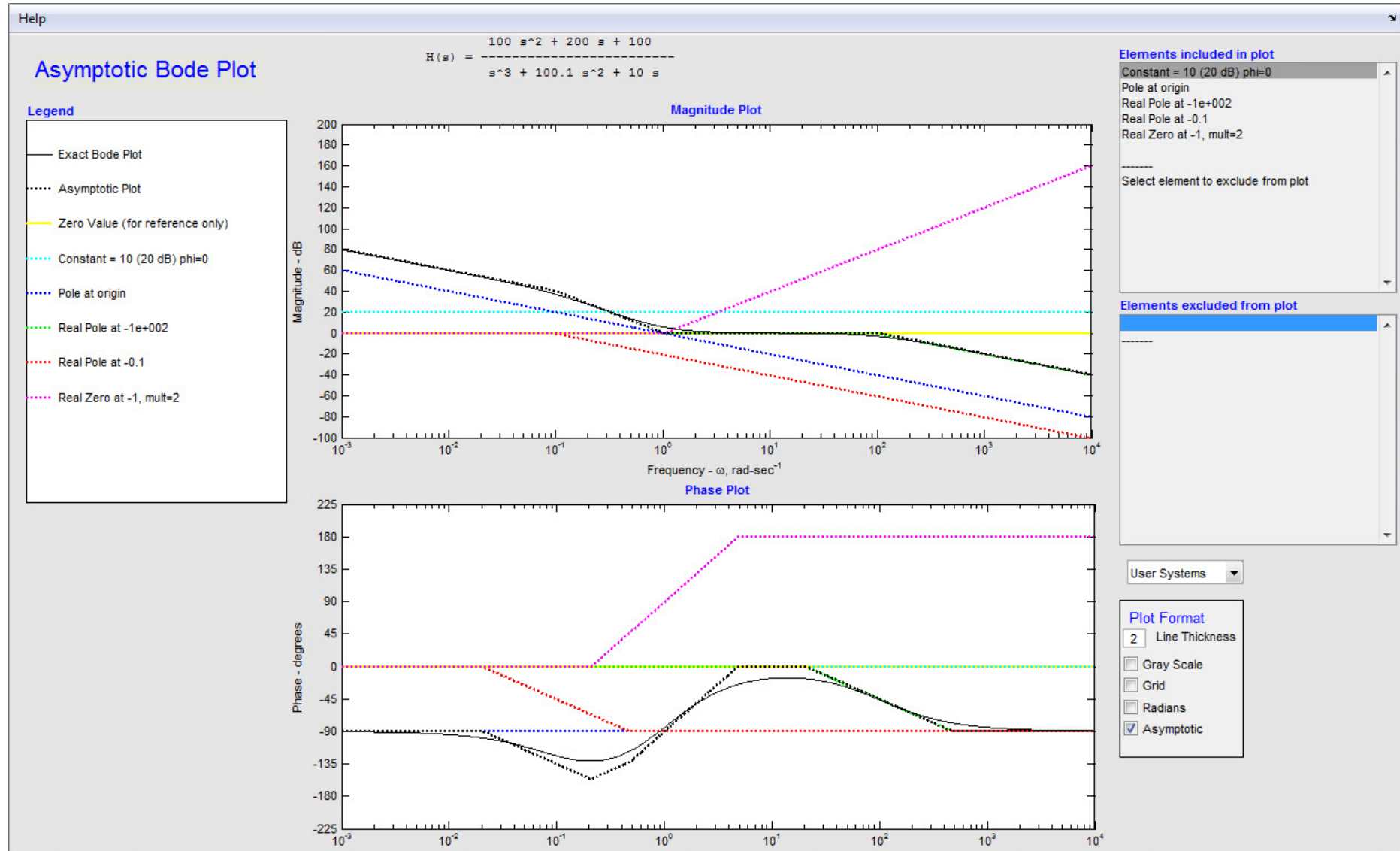
- ➔ **NOTA:** il Control Systems Toolbox non ha opzioni per visualizzare i diagrammi di Bode asintotici (i.e. approssimati) introdotti comunemente nei corsi di base di Automatica (peraltro proprio per un rapido svolgimento manuale, qualitativo)
- ➔ Molti siti didattici propongono funzioni Matlab per confrontare il diagramma esatto e quello approssimato, tra le quali la più interessante è stata reperita da:  
<http://ipsa.swarthmore.edu/Bode/BodePlotGui.html>  
e modificata per la pubblicazione sul sito del corso:

```
>> BodePlotGui (G)
```

**NOTA:** la modifica è solo nei punti di rottura per il diagramma delle fasi..



# Matlab: diagrammi di Bode



# Matlab: luogo delle radici

➔ Il **luogo delle radici** è mostrato dalla funzione **rlocus()** :

```
>> G=1/s/(s^2+4*s+5)
```

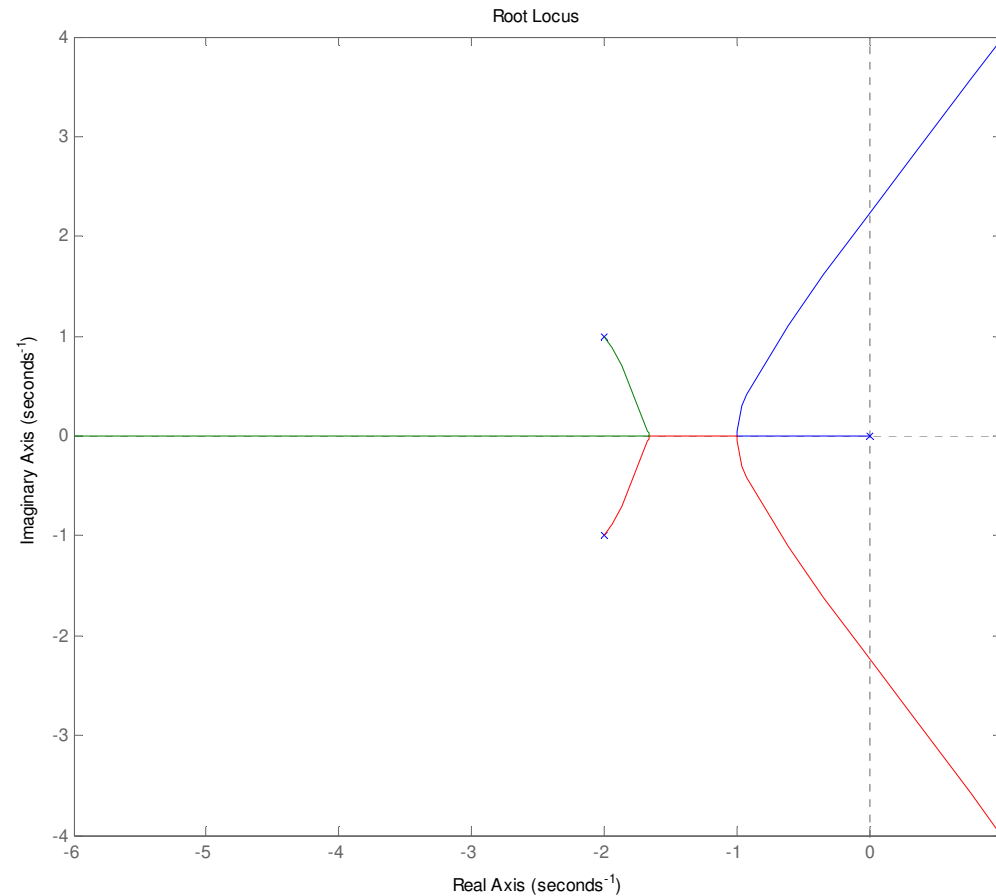
Transfer function:

1


-----

$s^3 + 4s^2 + 5s$

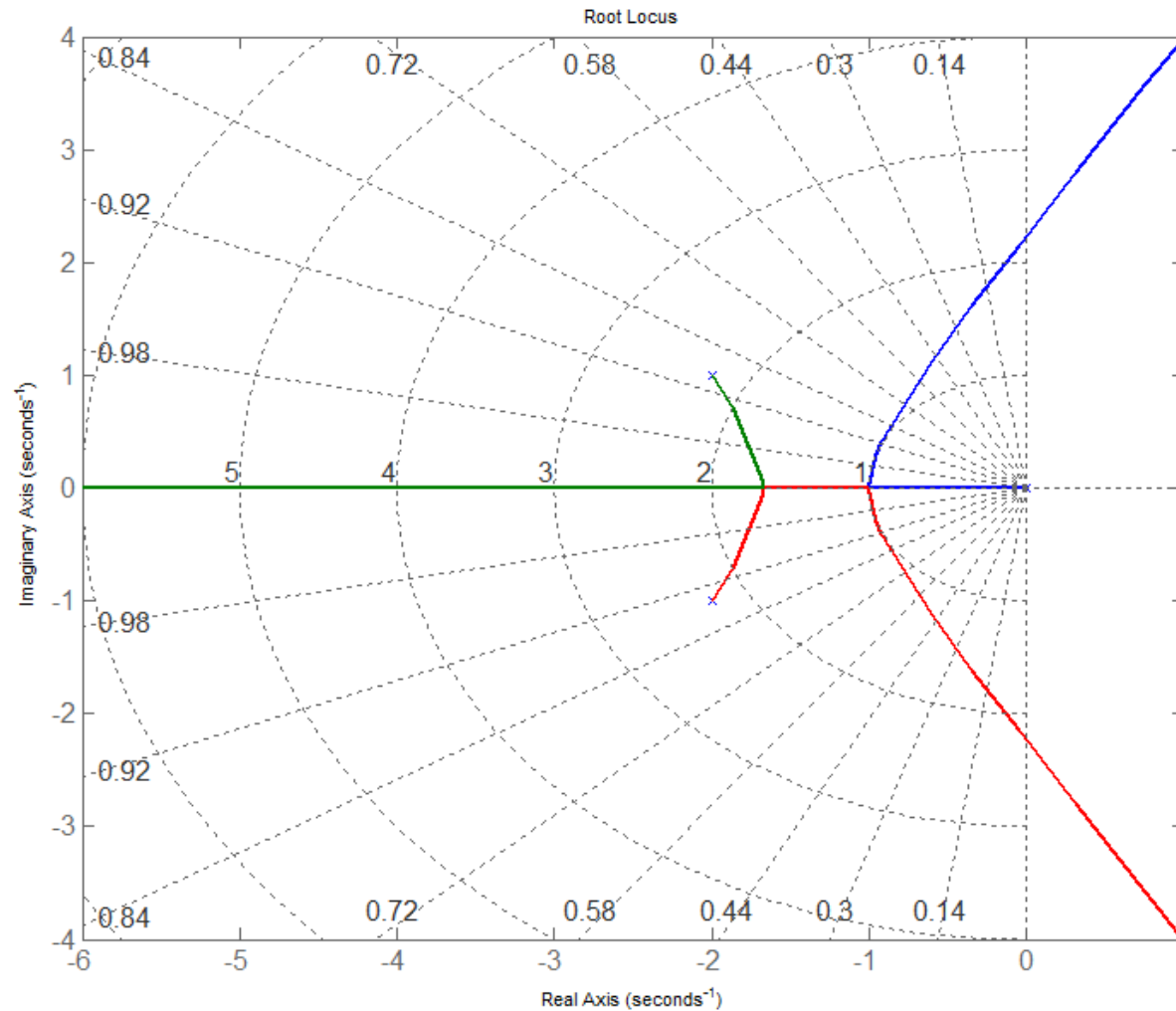
```
>> rlocus(G)
```



# Matlab: luogo delle radici

- 
- ➔ **NOTA:** abilitando la *grid* (tramite menu via tasto destro del mouse o da linea di comando, sempre con **grid on**) vengono visualizzati:
- le circonferenze sulle quali giacciono i poli complessi coniugati con la stessa pulsazione naturale  $\omega_n$
  - i raggi sui quali giacciono poli complessi coniugati con lo stesso coefficiente di smorzamento  $\delta$
  - ovviamente, solo per il semipiano sinistro, corrispondente alla regione in cui devono giacere i poli per funzioni di trasferimento asintoticamente stabili..

# Matlab: luogo delle radici

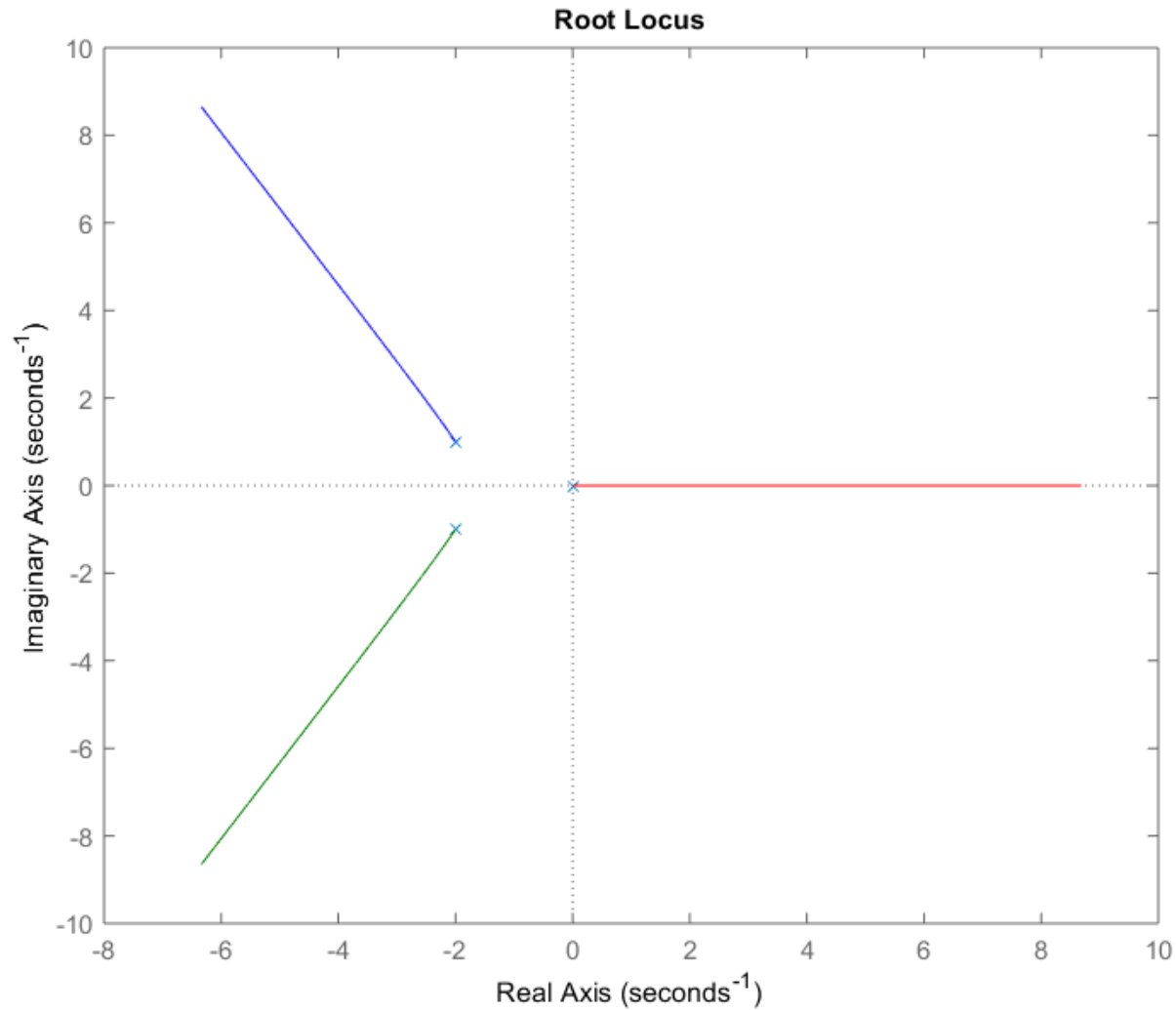


# Matlab: luogo delle radici

- ➔ **NOTA:** la funzione `rlocus(G)` considera solo valori di  $K > 0$  (i.e. luogo diretto) limitando automaticamente il valore massimo di  $K$  per ottimizzare l'aspetto grafico
- ➔ Per ottenere il luogo inverso, è necessario definire manualmente un intervallo di valori di  $K < 0$ :

```
>> Kvals=0:-0.01:-1000;  
>> rlocus(G,Kval)
```

# Matlab: luogo delle radici



# Matlab: diagrammi di Bode e luogo delle radici

- ➔ **NOTA:** cliccando con il tasto sinistro del mouse su una linea di qualunque grafico, ampiezza e fase di **bode ()** e uno dei ramo di **rlocus ()**, vengono visualizzate molte informazioni caratteristiche del grafico:
- **ampiezza** (in dB) o **fase** (in gradi) e pulsazione corrispondente, sui diagrammi di Bode
  - **Valore di K** corrispondente, **valore del polo** (o coppia di poli se complessi), **damping** ( $\delta$ ), **overshoot %** (in risposta al gradino) e **pulsazione naturale** ( $\omega_n$ ) sul luogo delle radici

# Matlab: altri comandi utili per l'analisi di FdT

## ➔ Calcolo di poli e zeri:

```
>> G=tf([4 5 6],[1 3 2 3])
```

```
>> pole(G)
```

```
ans =
```

```
-2.6717 + 0.0000i
```

```
-0.1642 + 1.0469i
```

```
-0.1642 - 1.0469i
```

```
>> zero(G)
```

```
ans =
```

```
-0.6250 + 1.0533i
```

```
-0.6250 - 1.0533i
```



# Matlab: altri comandi utili per l'analisi di FdT

➔ Calcolo di  $\tau$ ,  $\delta$ ,  $\omega_n$  per ogni polo:

```
>> damp (G)
```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-1.64e-01+1.05e+00i	1.55e-01	1.06e+00	6.09e+00
-1.64e-01-1.05e+00i	1.55e-01	1.06e+00	6.09e+00
-2.67e+00	1.00e+00	2.67e+00	3.74e-01

**NOTA:** **Damping** è sempre 1 per poli reali, **Frequency** è  $\omega_n$  per poli complessi e  $1/\tau$  per quelli reali, **Time Constant** è  $\delta\omega_n$  per poli complessi e  $\tau$  per quelli reali

# Matlab: altri comandi utili per l'analisi di FdT

➔ **Calcolo del guadagno statico** (i.e.  $G(j\omega)$  per  $\omega \rightarrow \infty$ )

```
>> dcgain(G)
```

**NOTA:** `dcgain()` può essere di aiuto per la **verifica dell'errore a regime**, insieme alle funzioni `step()` e/o `stepinfo()`. Il risultato di queste ultime, però, può essere inutilizzabile allo scopo, se la funzione di trasferimento è instabile (al contrario di `dcgain()`..)

# Matlab: altri comandi utili per l'analisi di FdT

➔ **Criterio di Routh** per la verifica dell'intervallo di valori di un coefficiente tale per cui il sistema chiuso in retroazione sia asintoticamente stabile (gli estremi dell'intervallo corrispondono alle condizioni di stabilità semplice o marginale): tale operazione non è ufficialmente supportata da Matlab. Si possono però trovare alcune funzioni sviluppate da utenti private sul sito di interscambio *Matlab Central File Exchange*, come ad esempio:

<https://it.mathworks.com/matlabcentral/fileexchange/58-routh-m>



# ANALISI DI FdT CON DIAGRAMMI DI BODE E LUOGO DELLE RADICI IN MATLAB

**FINE**