Fondamenti di Automatica

Cenni su Matlab
(e toolbox Control Systems + Symbolic)

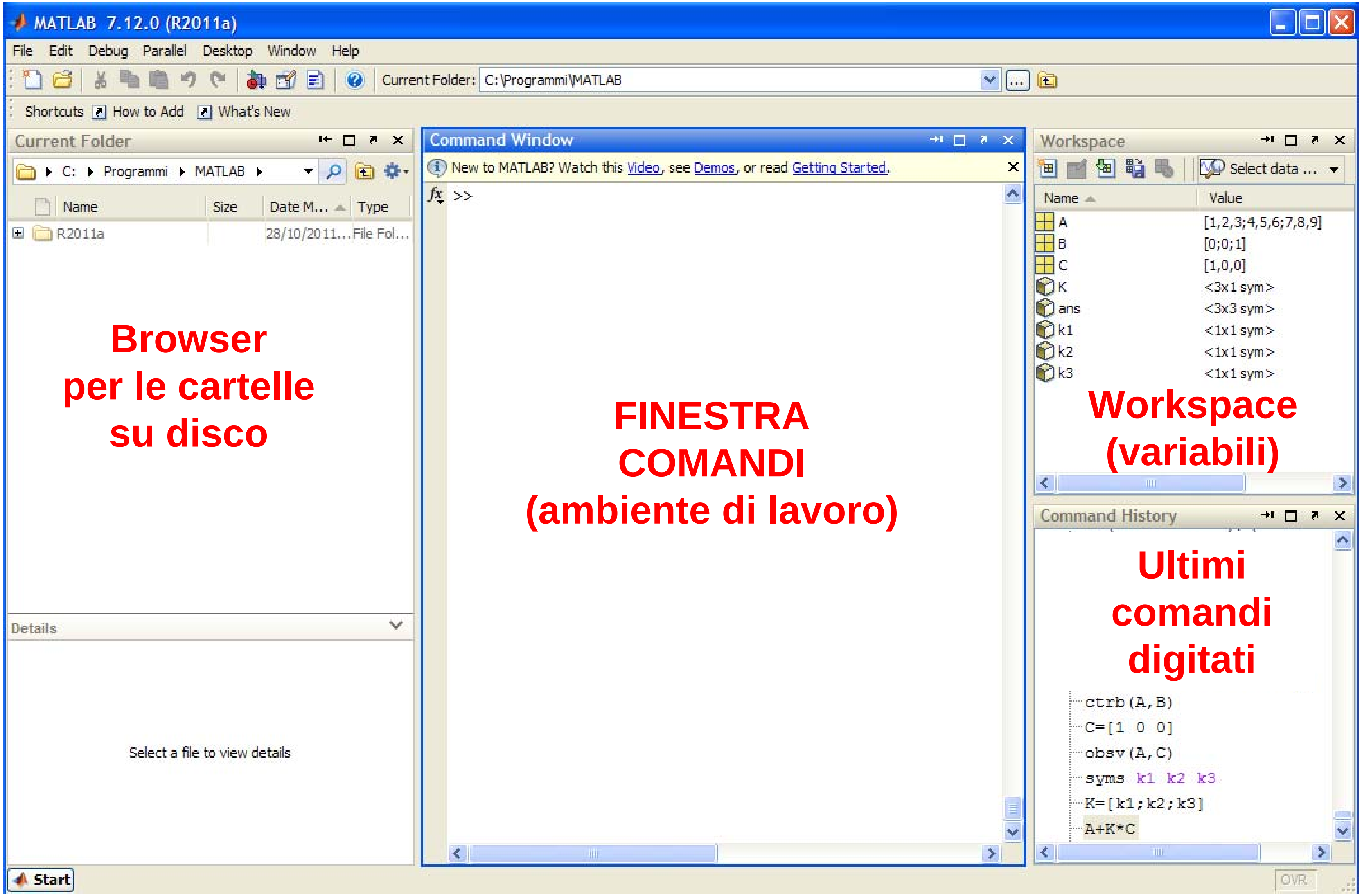
Dott. Ingg. Marcello Bonfè e Silvio Simani

Dipartimento di Ingegneria - Università di Ferrara

Tel. +39 0532 974839 / 974844

E-mail: marcello.bonfe@unife.it / silvio.simani@unife.it

Matlab: interfaccia principale



Matlab: definizione di variabili, vettori e matrici

Definire variabile scalare

```
>> x = 3
```

Definire vettore riga (1×3)

```
>> x = [1 2 3]
```

Idem, ma senza echo dell’output

```
>> x = [1 2 3];
```

Definire vettore colonna (3x1)

```
>> x = [1; 2; 3]
```

(oppure `>> x = [1 2 3]’`)

Definire matrice 3x4

```
>> A = [1 2 3 4;5 6 7 8;9 10 11 12]
```

Accedere / modificare elemento di riga 2 e colonna 1

```
>> A(2,1) = 0
```

Matlab: operazioni su matrici

- Le "solite" operazioni matematiche: **`+,-,*,/,^`**
- **Es. >> `A^3`** (potenza di matrice, solo se quadrata!)
- Precedute dal punto, sono eseguite elemento per elemento anziché in senso matriciale/vettoriale
- Operazioni specifiche per matrici / vettori:
  - Trasposta: **`A'`**
  - Determinante: **`det(A)`**
  - Inversa: **`inv(A)`**
  - Autovalori: **`eig(A)`**
  - Rango: **`rank(A)`**
  - Polinomio caratteristico: **`poly(A)`**
  - Esponenziale di matrice: **`expm(A)`**
  - Radici di un polinomio: **`roots(x)`** (x vettore dei coeff.)

Matlab: esponenziale di matrice (calcolo simbolico)

- Si consideri l'esercizio 1 della prova 6 (dalle scansioni di testi del Prof. Sergio Beghelli):

1) Per il sistema $\dot{x}(t) = \begin{bmatrix} -4 & 0 \\ 1 & -4 \end{bmatrix} x(t)$ si calcoli $x(4)$ sapendo che $x(3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Risposta : $x(4) =$

- La soluzione richiede il calcolo dell'esponenziale di matrice e^{At}, il cui procedimento analitico è riportato in FdA-1.3-Analisi_2013.pdf (slide 69 e succ.)

Matlab: esponenziale di matrice (calcolo simbolico)

- In Matlab, è necessario definire la matrice A e il simbolo **t**, quest'ultima operazione possibile grazie al Symbolic Toolbox

```
>> A=[-4 0; 1 -4]
>> syms t
>> expm(A*t)
ans =
[ 1/exp(4*t),          0]
[ t/exp(4*t), 1/exp(4*t)]
```

NOTA: il risultato è simbolico, i termini esponenziali sono a denominatore, il che equivale ad esponente negativo

Matlab: esponenziale di matrice (calcolo simbolico)

- Per ottenere il risultato dell'esercizio, è necessario sostituire gli opportuni valori per t e moltiplicare la matrice esponenziale per il valore dello stato fornito dal testo

```
>> x3=[1; 0]
>> x4=expm(A*(4-3))*x3
x4 =
    0.0183
    0.0183
```

NOTA: il risultato numerico equivale a e^{-4} (in Matlab **`exp(-4)`**) per entrambe le variabili di stato..

Matlab: test di controllabilità / osservabilità

- Grazie al Control Systems Toolbox, il test è eseguibile semplicemente lanciando i comandi:

```
>> P=ctrb(A,B)
```

per il test di controllabilità, e:

```
>> Q=obsv(A,C)
```

per il test di osservabilità

Matlab: test di controllabilità / osservabilità

- Si consideri l'esercizio 9 della prova 6 (dalle scansioni di testi del Prof. Sergio Beghelli)

9) Per il sistema

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & -3 & 1 & 0 & 2 \\ 2 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \end{bmatrix} x(t)$$

- Il testo richiede di determinare la forma minima, cioè la parte osservabile del sistema dato

Matlab: test di controllabilità / osservabilità

- In Matlab:

```
>> A=[0 0 0 0 -1; 0 -3 1 0 2; 2 0 1 0 0; -1 0 0 1 1; 0 0 0 0 2]
>> B=[1; 0; 1; -1; 1]
>> C=[0 0 -1 0 0]
>> Q=obsv(A,C)
Q =
     0     0    -1     0     0
    -2     0    -1     0     0
    -2     0    -1     0     2
    -2     0    -1     0     6
    -2     0    -1     0    14
>> Qt=Q'
```

NOTA: con quest'ultima operazione, la matrice di osservabilità viene posta in modo da rendere l'analisi omogenea rispetto a quella di controllabilità, in modo cioè che si debba osservare quali COLONNE risultino indipendenti.

Matlab: test di controllabilità / osservabilità

- Dall'analisi delle colonne della matrice Qt, si può vedere che solo le prime tre risultano indipendenti, pertanto la forma minima del sistema è di ordine 3 ➔ **`rank(Qt)`**
- Con il procedimento di ortonormalizzazione di Gram-Schmidt (usare la funzione **`gramsch(A)`** fornita come materiale integrativo), tali prime tre colonne diventano:

```
>> T1=gramsch(Qt(:,1:3))
T1 =
     0    -1     0
     0     0     0
    -1     0     0
     0     0     0
     0     0     1
```

Matlab: test di controllabilità / osservabilità

- Trascurando il segno negativo (o meglio, invertendolo..), si può quindi costruire una matrice di trasformazione dello stato con le "classiche" colonne della base ortonormale:

```
>> T1(:,1)=-T1(:,1)
>> T1(:,2)=-T1(:,2)
>> T2=[0 1 0 0 0;0 0 0 1 0]'
>> T=[T1 T2]
T =
     0     1     0     0     0
     0     0     0     1     0
     1     0     0     0     0
     0     0     0     0     1
     0     0     1     0     0
```

NOTA: evidenziati gli elementi 1 nelle posizioni corrispondenti alle variabili di stato OSSERVABILI

Matlab: test di controllabilità / osservabilità

- Trasformando le matrici A,B,C, si ottiene la forma minima, contenuta nelle prime tre righe / tre colonne (**ES**. solo per A)

```
>> Ao=T'*A*T
>> Ao1=Ao(1:3,1:3)
Ao1 =
     1     2     0
     0     0    -1
     0     0     2
```

i cui autovalori (non richiesti esplicitamente dal testo) sono:

```
>> eig(Ao1)
ans = 1
      0
      2
```

Matlab: test di controllabilità / osservabilità

- **NOTA:** come suggerito in FdA-SolutionsGuide.pdf, il risultato ottenuto corrisponde di fatto all'eliminazione dalle matrici A,B,C delle righe/colonne corrispondenti alle variabili di stato NON OSSERVABILI, in questo caso la seconda e la quarta:

```
>>  Ao1=A([1 3 5],[1 3 5])
Ao1 =
      0      0     -1
      2      1      0
      0      0      2
```

Vettore delle righe/colonne selezionate..

$$A=\begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & -3 & 1 & 0 & 2 \\ 2 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Matlab: progetto analitico di controllo

- Si consideri l’esercizio 10 della prova 6 (dalle scansioni di testi del Prof. Sergio Beghelli):

10) Per il sistema

$$\dot{x}(t) = \begin{bmatrix} -6 & 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & -2 & -3 & 0 \\ 2 & 1 & 0 & -3 & 0 \\ 1 & 0 & 1 & 2 & -6 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t)$$

si progetti una retroazione uscita-ingresso $u(t) = K y(t) + v(t)$ per assegnare al massimo numero di autovalori il valore -3.

Risposta $K =$

Matlab: progetto analitico di controllo

- In Matlab:

```
>> A=[-6 1 0 0 0;0 -3 0 0 0;0 0 -2 -3 0; 2 1 0 -3 0;1 0 1 2 -6]
>> B=[0; 0; 1; 0; 0]
>> C=[0 0 1 0 0;0 0 0 1 0]
```

- Il progetto della retroazione uscita-ingresso richiede di determinare gli autovalori della matrice A+BKC, con K in questo caso di dimensione 1x2
- Con il calcolo simbolico:

```
>> syms k1 k2
>> K=[k1 k2]
>> Acl=A+B*K*C      Acl =
        [ -6,  1,      0,      0,  0]
        [  0, -3,      0,      0,  0]
        [  0,  0, k1 - 2, k2 - 3,  0]
        [  2,  1,      0,     -3,  0]
        [  1,  0,      1,      2, -6]
```

Matlab: progetto analitico di controllo

- Dall'analisi degli autovalori della matrice Acl, si può osservare che solo uno di questi è assegnabile ed è in funzione di k1:

```
>> eig(Acl)
ans =
     -6
     -6
     -3
     -3
     k1 – 2
```

Pertanto, il risultato finale richiesto è K = [-1   arbitrario]

NOTA: in molti casi, l'espressione simbolica ottenuta con il comando eig() può essere troppo complessa per risolvere agevolmente l'assegnazione degli autovalori. In generale, per calcoli manuali si faccia riferimento ai suggerimenti contenuti in FdA-SolutionsGuide.pdf …

Matlab: trasformate e antitrasformate di Laplace

- Il Symbolic Toolbox contiene le funzioni per il calcolo simbolico (appunto) delle trasformate di Fourier, Laplace, Z e relative inverse
- Ad esempio, si consideri l’esercizio 2 della prova 6 (dalle scansioni di testi del Prof. Beghelli):

2) Per il sistema con funzione di trasferimento $G(s)=\dfrac{5s+12}{(s+2)(s+3)}$ e con ingresso $u(s)=\dfrac{1}{s}$ si calcoli $y(t)$.

Risposta $y(t) =$

- La soluzione manuale richiede l’applicazione del metodo di scomposizione in fratti semplici (v. slide 4-12 in FdA-2.2-RispostaSistemiElementari_2013.pdf)

Matlab: trasformate e antitrasformate di Laplace

- In Matlab:

```
>> syms s
>> G=(5*s+12)/(s+2)/(s+3)
G =
(5*s + 12)/((s + 2)*(s + 3))
>> U = 1/s
>> Y = G*U
>> y = ilaplace(Y)
y =
2 - 1/exp(3*t) - 1/exp(2*t)
```

Matlab: trasformate e antitrasformate di Laplace

- Ovviamente, il Symbolic Toolbox potrebbe anche essere usato per calcolare esplicitamente il passaggio da una rappresentazione nello spazio degli stati (matrici A,B,C,D) alla corrispondente funzione (o matrice) di trasferimento
- Tuttavia, essendo tale operazione tipicamente necessaria nel progetto di sistemi di controllo, il Control Systems Toolbox contiene funzioni specifiche, basate su strutture dati ancora più specifiche

Matlab: trasformate e antitrasformate di Laplace

- Si consideri l'esercizio 3 della prova 9 (dalle scansioni di testi del Prof. Beghelli):

3) Per il sistema $\dot{x}(t) = \begin{bmatrix} -3 & 0 \\ 1 & -6 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$

$$y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t)$$

si calcoli la funzione di trasferimento $G(s)$

Risposta $G(s) =$

- In Matlab:

```
>> A=[-3 0; 1 -6]
>> B=[1; 1]
>> C=[1 1]
```

Matlab: trasformate e antitrasformate di Laplace

- Soluzione 1 (Symbolic Toolbox):

```
>> syms s
>> sA=inv(s*eye(2) – A)
```
← eye(2)= identità 2x2..

```
>> G=C*sA*B
G =
1/(s + 3) + 1/(s + 6) + 1/((s + 3)*(s + 6))
>> G=collect(G)
G =
(2*s + 10)/(s^2 + 9*s + 18)
```

Matlab: trasformate e antitrasformate di Laplace

- Soluzione 2 (Control Systems Toolbox):

```
>> sys=ss(A,B,C,0)
```
← **D=0, necessario quarto parametro..**

```
>> G=tf(sys)
Transfer function:
   2 s + 10
--------------
s^2 + 9 s + 18
```

Oppure anche, calcolando i coefficienti di numeratore e denominatore della FdT:

```
>> [N,D]=ss2tf(A,B,C,0)
N = 0     2    10
D = 1     9    18
>> G=tf(N,D)
```

Matlab: diagrammi di Bode e luogo delle radici

- Oltre al passaggio alla funzione **`tf(num,den)`** dei due vettori contenenti i coefficienti della FdT, esiste un'alternativa comoda per definire la FdT con la struttura del Control Systems Toolbox:

`>> s=tf('s')` ← **"definisce" la variabile di Laplace**

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

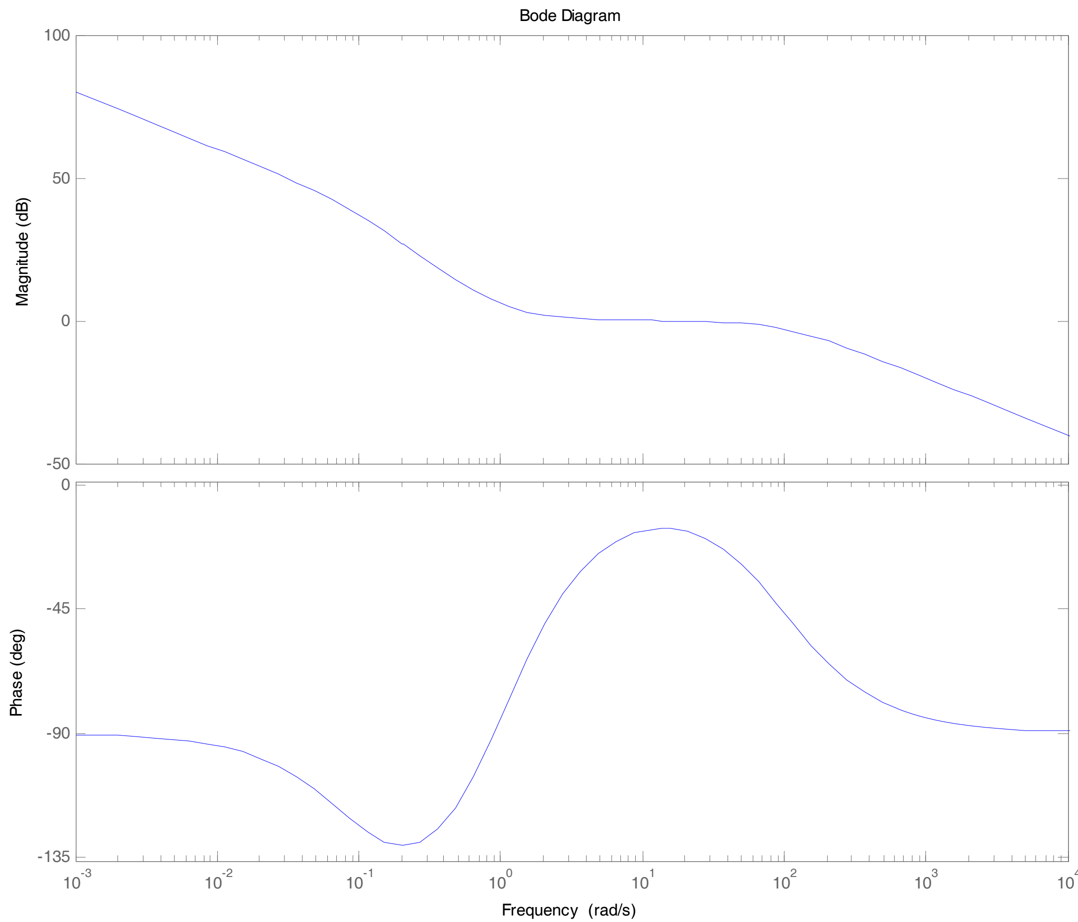
NOTA1: l'esempio è tratto dalla soluzione dell'esercizio 5 della prova 8 (dalle scansioni dei testi del Prof. Beghelli), v. Anche FdA-SolutionsGuide.pdf sui diagrammi di Bode..

NOTA2: in questo caso **`s`** NON è una variabile simbolica, ma una vera e propria FdT rappresentata con la struttura dati corrispondente del Control Systems Toolbox..

Matlab: diagrammi di Bode e luogo delle radici

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

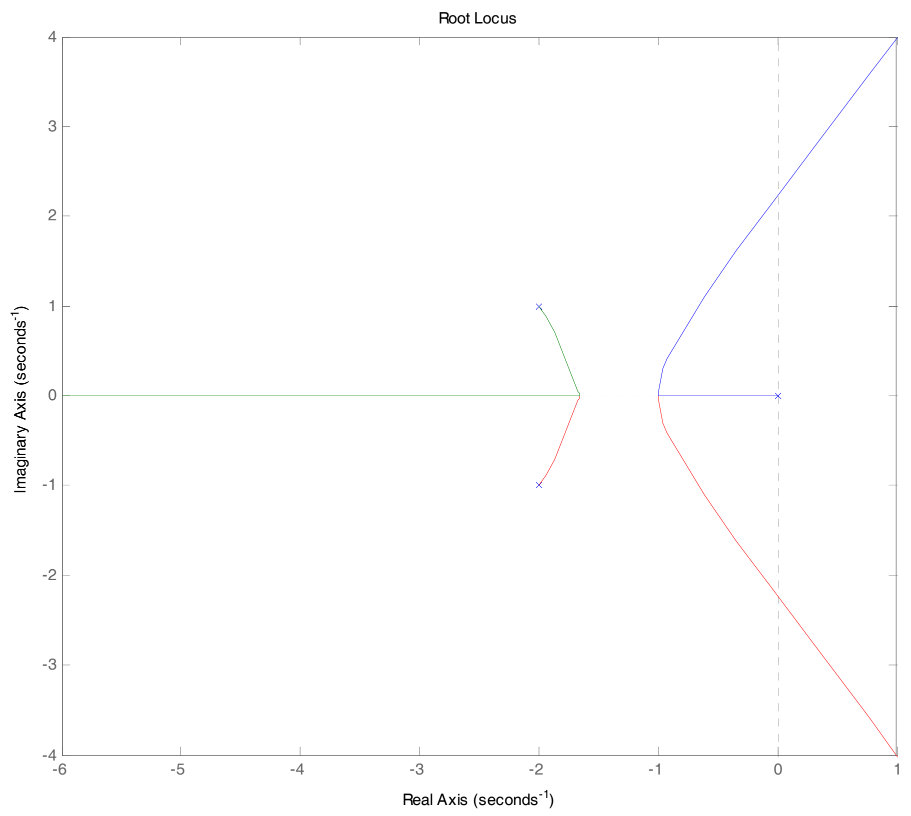
```
>> bode(G)
```



Matlab: diagrammi di Bode e luogo delle radici

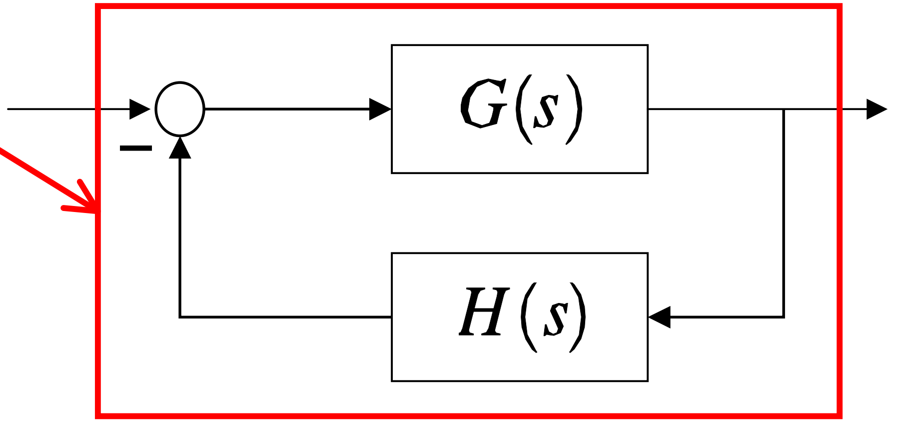
- Analogamente, è possibile visualizzare immediatamente il luogo delle radici

```
>> G=1/s/(s^2+4*s+5)
Transfer function:
        1
-----------------
s^3 + 4 s^2 + 5 s
>> rlocus(G)
```



Matlab: altre funzioni del Control Systems Toolbox

```
>> Gcl = feedback(G,H)
```




```
>> Gp = parallel(G1,G2)
```
← parallelo di FdT

```
>> Gs = series(G1,G2)
```
← serie di FdT

```
>> step(G)
```
← grafica la risposta al gradino

```
>> impulse(G)
```
← grafica la risp. impulsiva

Matlab: criterio di Routh, errori a regime, ecc..

PURTROPPO, non è possibile mescolare elementi del Symbolic Toolbox con quelli del Control Systems Toolbox (es. FdT con coefficienti simbolici..)

PERTANTO, esercizi come quelli proposti all'esame su:

- riduzione di diagrammi a blocchi
- determinazione di intervalli di stabilità per sistemi in retroazione (criterio di Routh)
- calcolo di coefficienti t.c. si abbia
  - un certo errore a regime
  - oppure, una certa pulsazione naturale ω_n, coefficiente di smorzamento δ, tempo di assestamento T_a, ecc.

risultano in genere più articolati (o impossibili) da risolvere con l'ausilio di Matlab, piuttosto che manualmente, pertanto non verranno trattati in questa introduzione..

CENNI SU MATLAB

FINE