ESERCITAZIONI
MATLAB – Nozioni

Symbolic Math Toolbox

https://mathworks.com/help/symbolic

```
syms x
y = x^2;
```

Assign 2 to x. The value of y is still x^2 instead of 4.

```
x = 2;
y
```

y = x^2

Evaluate y with the new value of x by using subs.

```
subs(y)
```

```
ans = 4
```

ESERCITAZIONI
MATLAB – Nozioni

Symbolic Math Toolbox

https://mathworks.com/help/symbolic

```
syms x
f = sin(x)^2;
diff(f)
```

```
ans =
2*cos(x)*sin(x)
```

```
syms x
f = sin(x)^2;
```

To find the indefinite integral, enter

```
int(f)
```

```
ans =
x/2 - sin(2*x)/4
```

Solve Algebraic Equations with One Symbolic Variable

Use the double equal sign (==) to define an equation. Then you can `solve` the equation by calling the solve function. For example, solve this equation:

```
syms x
solve(x^3 - 6*x^2 == 6 - 11*x)
```

```
ans =
 1
 2
 3
```

If you do not specify the right side of the equation, `solve` assumes that it is zero:

```
syms x
solve(x^3 - 6*x^2 + 11*x - 6)
```

```
ans =
 1
 2
 3
```

ESERCITAZIONI
MATLAB – Nozioni

Symbolic Math Toolbox

https://mathworks.com/help/symbolic

Simplify Symbolic Expressions

Symbolic Math Toolbox provides a set of simplification functions allowing you to manipulate the output of a symbolic expression.

```
phi = (1 + sqrt(sym(5)))/2;
f = phi^2 - phi - 1
```

returns

```
f =
(5^(1/2)/2 + 1/2)^2 - 5^(1/2)/2 - 3/2
```

You can simplify this answer by entering

```
simplify(f)
```

and get a very short answer:

```
ans =
0
```

ESERCITAZIONI
MATLAB – Nozioni

Symbolic Math Toolbox

https://mathworks.com/help/symbolic

Simplify Symbolic Expressions

Symbolic simplification is not always so straightforward.

For example, to show the order of a polynomial or symbolically differentiate or integrate a polynomial, use the standard polynomial form with all the parentheses multiplied out and all the similar terms summed up.

To rewrite a polynomial in the standard form, use the `expand` function:

```
syms x
f = (x ^2- 1)*(x^4 + x^3 + x^2 + x + 1)*(x^4 - x^3 + x^2 - x + 1);
expand(f)
```

```
ans =
x^10 - 1
```

The `factor` simplification function shows the polynomial roots.

If a polynomial cannot be factored over the rational numbers, the output of the `factor` function is the standard polynomial form. For example, to factor the third-order polynomial, enter:

```
syms x
g = x^3 + 6*x^2 + 11*x + 6;
factor(g)
```

```
ans =
[ x + 3, x + 2, x + 1]
```

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

https://mathworks.com/help/control/

Rappresentazioni di modelli lineari

- Rappresentazione in spazio di stato

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

- Rappresentazione mediante funzione di trasferimento

$$H(s) = \frac{s+2}{s^2+s+10}$$

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Funzioni di trasferimento

Rappresentazioni della funzione di trasferimento

Una funzione di trasferimento SISO a tempo continuo è espressa come il rapporto:

$$G(s) = \frac{N(s)}{D(s)},$$

di polinomi $N(s)$ e $D(s)$, chiamati rispettivamente polinomi del numeratore e del denominatore.

È possibile rappresentare i sistemi lineari come funzioni di trasferimento in forma polinomiale o fattorizzata.
Ad esempio, la funzione di trasferimento in forma polinomiale:

$$G(s) = \frac{s^2 - 3s - 4}{s^2 + 5s + 6}$$

Comando
`tf`

può essere riscritta in forma fattorizzata come:

$$G(s) = \frac{(s+1)(s-4)}{(s+2)(s+3)}.$$

Comando
`zpk`

Le funzioni di trasferimento MIMO sono array di funzioni di trasferimento SISO. Ad esempio:

$$G(s) = \begin{bmatrix} \frac{s-3}{s+4} \\ \frac{s+1}{s+2} \end{bmatrix}$$

è una funzione di trasferimento a un input e due output.

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Funzioni di trasferimento

Rappresentazioni della funzione di trasferimento

Creazione di funzioni di trasferimento utilizzando i coefficienti del numeratore e del denominatore

Questo esempio mostra come creare funzioni di trasferimento a tempo continuo a singolo input-singolo output (SISO) dai coefficienti del numeratore e del denominatore utilizzando `tf`.

Creare la funzione di trasferimento $G(s) = \frac{s}{s^2 + 3s + 2}$:

```
num = [1 0];
den = [1 3 2];
G = tf(num,den);
```

`num` e `den` sono i coefficienti polinomiali del numeratore e del denominatore in potenze decrescenti di s. Ad esempio, `den = [1 3 2]` rappresenta il polinomio del denominatore $s^2 + 3s + 2$.

`G` è un oggetto del modello `tf`, che è un contenitore di dati per la rappresentazione delle funzioni di trasferimento in forma polinomiale.

Suggerimento

In alternativa, è possibile specificare la funzione di trasferimento $G(s)$ come espressione in s:

1. Creare un modello della funzione di trasferimento per la variabile s.

```
s = tf('s');
```

2. Specificare $G(s)$ come rapporto di polinomi in s.

```
G = s/(s^2 + 3*s + 2);
```

ESERCITAZIONI
MATLAB – Nozioni

MathWorks®

Control System Toolbox

Rappresentazioni di modelli lineari

Funzioni di trasferimento

Rappresentazioni della funzione di trasferimento

Creazione del modello della funzione di trasferimento utilizzando zeri, poli e guadagno

Questo esempio mostra come creare funzioni di trasferimento a singolo input-singolo output (SISO) in forma fattorizzata utilizzando zpk.

Creare la funzione di trasferimento fattorizzata $G(s) = 5\frac{s}{(s+1+i)(s+1-i)(s+2)}$:

```
Z = [0];
P = [-1-1i -1+1i -2];
K = 5;
G = zpk(Z,P,K);
```

$$G(s) = \frac{\rho \prod_i (s+z_i) \prod_i (s^2 + 2\zeta_i \alpha_{ni} s + \alpha_{ni}^2)}{s^g \prod_i (s+p_i) \prod_i (s^2 + 2\xi_i \omega_{ni} s + \omega_{ni}^2)}$$

$$G(s) = \frac{\mu \prod_i (1+\tau_i s) \prod_i (1 + 2\zeta_i s/\alpha_{ni} + s^2/\alpha_{ni}^2)}{s^g \prod_i (1+T_i s) \prod_i (1 + 2\xi_i s/\omega_{ni} + s^2/\omega_{ni}^2)}$$

Z e P sono gli zeri e i poli (rispettivamente le radici del numeratore e del denominatore). K è il guadagno della forma fattorizzata.

Ad esempio, $G(s)$ presenta un polo reale in corrispondenza di $s = -2$ e una coppia di poli complessi in corrispondenza di $s = -1 \pm i$. Il vettore `P = [-1-1i -1+1i -2]` specifica le posizioni di questi poli.

G è un oggetto del modello zpk, che è un contenitore di dati per rappresentare le funzioni di trasferimento in forma fattorizzata.

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Funzioni di trasferimento

Rappresentazioni della funzione di trasferimento

Funzioni di trasferimento MIMO

Le funzioni di trasferimento MIMO sono array bidimensionali di funzioni di trasferimento SISO elementari. Ci sono due modi di specificare i modelli delle funzioni di trasferimento MIMO:

- Concatenazione dei modelli delle funzioni di trasferimento SISO
- Utilizzando `tf` con argomenti di array di celle

Concatenazione di modelli SISO

Considerare la seguente funzione di trasferimento

$$H(s) = \begin{bmatrix} \frac{s-1}{s+1} \\ \frac{s+2}{s^2+4s+5} \end{bmatrix}.$$

È possibile specificare $H(s)$ mediante la concatenazione dei suoi elementi SISO. Per esempio,

```
h11 = tf([1 -1],[1 1]);
h21 = tf([1 2],[1 4 5]);
```

oppure, in modo equivalente,

```
s = tf('s')
h11 = (s-1)/(s+1);
h21 = (s+2)/(s^2+4*s+5);
```

può essere concatenato in modo da formare $H(s)$.

```
H = [h11; h21]
```

Questa sintassi imita la concatenazione delle matrici standard e tende ad essere più semplice e più leggibile per i sistemi MIMO che presentano molti input e/o output.

Suggerimento

Utilizzare zpk anziché `tf` per creare funzioni di trasferimento MIMO in forma fattorizzata.

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Funzioni di trasferimento

Rappresentazioni della funzione di trasferimento

Funzioni di trasferimento MIMO

Utilizzo della funzione `tf` con gli array di celle

In alternativa, per definire le funzioni di trasferimento MIMO con la funzione `tf`, sono necessari due array di celle (ad es. `N` e `D`) per rappresentare rispettivamente gli insiemi dei polinomi al numeratore e al denominatore.

Ad esempio, per la matrice di trasferimento razionale $H(s)$, i due array di celle `N` e `D` devono contenere le rappresentazioni riga-vettore degli elementi polinomiali di

$$N(s) = \left[\frac{s-1}{s+2}\right], \quad D(s) = \left[\frac{s+1}{s^2+4s+5}\right].$$

È possibile specificare questa matrice di trasferimento MIMO $H(s)$ digitando

```
N = {[1 -1];[1 2]};   % Cell array for N(s)
D = {[1 1];[1 4 5]}; % Cell array for D(s)
H = tf(N,D)
```

```
Transfer function from input to output...
      s - 1
 #1:  -----
      s + 1

          s + 2
 #2:  -------------
      s^2 + 4 s + 5
```

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

SS

State-space model

Examples

SISO State-Space Model

Create the SISO state-space model defined by the following state-space matrices:

$$A = \begin{bmatrix} -1.5 & -2 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \quad C = [0 \quad 1] \quad D = 0$$

Specify the A, B, C and D matrices, and create the state-space model.

```
A = [-1.5,-2;1,0];
B = [0.5;0];
C = [0,1];
D = 0;
sys = ss(A,B,C,D)
```

ESERCITAZIONI
MATLAB – Nozioni

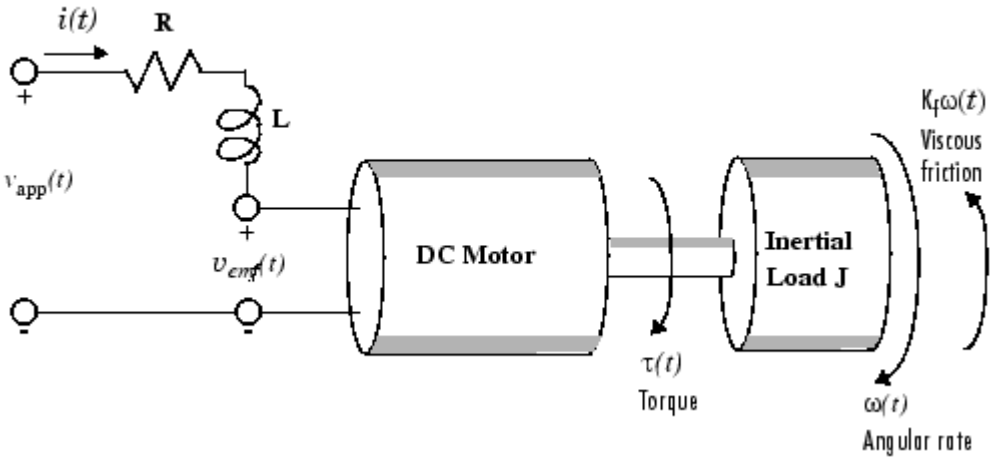
Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale



In questo modello, le dinamiche del motore stesso sono idealizzate: ad esempio, si assume che il campo magnetico sia costante. La resistenza del circuito è denotata da R e l'auto-induttanza dell'armatura da L. In questo esempio, i rapporti tra potenziale elettrico e forza meccanica sono la legge di Faraday dell'induzione e la legge di Ampère per la forza di un conduttore che si muove attraverso un campo magnetico.

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale

Equazioni stato-spazio per il motore CC

Date le due equazioni differenziali derivate dall'ultima sezione, ora è possibile sviluppare una rappresentazione stato-spazio del motore CC come sistema dinamico. La corrente i e la velocità angolare ω sono i due stati del sistema. La tensione applicata, v_{app}, è l'input del sistema, mentre la velocità angolare ω è l'output.

$$\frac{d}{dt}\begin{bmatrix} i \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{J} & -\frac{K_f}{J} \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \cdot v_{app}(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + [0] \cdot v_{app}(t)$$

- K_b è la costante di forza controelettromotrice indotta (essa dipende anche da certe proprietà fisiche del sistema)
- K_m è la costante di armatura (essa è correlata alle proprietà fisiche del motore)
- K_f è la costante relativa all'attrito viscoso
- R è la resistenza del circuito
- L è l'induttanza del circuito
- J è il momento di inerzia

ESERCITAZIONI
MATLAB – Nozioni

MathWorks®

Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale

Costruzione di modelli SISO

Una volta sviluppato un set di equazioni differenziali che descrivono il proprio impianto, è possibile costruire dei modelli SISO tramite semplici comandi.

Costruzione di un modello in spazio di stato del motore CC

Inserire i seguenti valori nominali per i vari parametri di un motore CC.

```
R= 2.0 % Ohms
L= 0.5 % Henrys
Km = .015 % torque constant
Kb = .015 % emf constant
Kf = 0.2 % Nms
J= 0.02 % kg.m^2
```

Dati questi valori, è possibile costruire la rappresentazione stato-spazio numerica utilizzando la funzione `ss`.

```
A = [-R/L -Kb/L; Km/J -Kf/J]
B = [1/L; 0];
C = [0 1];
D = [0];
sys_dc = ss(A,B,C,D)
```

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale

Costruzione di modelli SISO

Una volta sviluppato un set di equazioni differenziali che descrivono il proprio impianto, è possibile costruire dei modelli SISO tramite semplici comandi.

Costruzione di un modello in spazio di stato del motore CC

Questi comandi restituiscono il seguente risultato:

```
a =
                x1          x2
       x1       -4       -0.03
       x2     0.75         -10

b =
                u1
       x1        2
       x2        0

c =
                x1          x2
       y1        0           1

d =
                u1
       y1        0
```

```
>> pole(sys_dc)

ans =

   -4.0038
   -9.9962

>> zero(sys_dc)

ans =

  0×1 empty double column vector
```

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale

Costruzione di modelli SISO

Una volta sviluppato un set di equazioni differenziali che descrivono il proprio impianto, è possibile costruire dei modelli SISO tramite semplici comandi.

Conversione tra diverse rappresentazioni del modello

Conversione tra diverse rappresentazioni del modello

Ora che si dispone di una rappresentazione stato-spazio del motore CC, è possibile effettuare la conversione in altre rappresentazioni del modello, compresi i modelli TF (funzione di trasferimento) e ZPK (zero/poli/guadagno).

Rappresentazione come funzione di trasferimento.

```
sys_tf = tf(sys_dc)

Transfer function:
       1.5
------------------
s^2 + 14 s + 40.02
```

Rappresentazione Zero/Poli/Guadagno.

```
sys_zpk = zpk(sys_dc)

Zero/pole/gain:
         1.5
-------------------
(s+4.004) (s+9.996)
```

ESERCITAZIONI
MATLAB – Nozioni

Control System Toolbox

Rappresentazioni di modelli lineari

Esempio SISO: Il motore CC

Un semplice modello di un motore CC con un carico inerziale mostra la velocità angolare del carico, $\omega(t)$, come output e la tensione applicata, $v_{app}(t)$, come input. Questa figura mostra un semplice modello del motore CC.

Un semplice modello di motore CC con carico inerziale

Costruzione di modelli SISO

Una volta sviluppato un set di equazioni differenziali che descrivono il proprio impianto, è possibile costruire dei modelli SISO tramite semplici comandi.

Creazione di modelli (funzione di trasferimento, zeri/poli/guadagno)

Ad esempio, si può creare la funzione di trasferimento specificando con questo codice il numeratore e il denominatore.

```
sys_tf = tf(1.5,[1 14 40.02])

Transfer function:
       1.5
------------------
s^2 + 14 s + 40.02
```

In alternativa, se si desidera creare la funzione di trasferimento del motore CC direttamente, utilizzare questi comandi.

```
s = tf('s');
sys_tf = 1.5/(s^2+14*s+40.02)
```

Questi comandi portano a questa funzione di trasferimento.

```
Transfer function:
        1.5
--------------------
s^2 + 14 s + 40.02
```

Per costruire il modello in formato zero/poli/guadagno, utilizzare questo comando.

```
sys_zpk = zpk([],[-9.996 -4.004], 1.5)
```

Questo comando restituisce la seguente rappresentazione zero/poli/guadagno.

```
Zero/pole/gain:
        1.5
-------------------
(s+9.996) (s+4.004)
```

Riferimenti Bibliografici

[1] https://it.mathworks.com