Manuale sintetico per l'uso del Control System Toolbox di Matlab

Alessandro Melis Pierluigi Muntoni

2 Dicembre 2002

Introduzione

Questo documento ha lo scopo di presentare, in una versione opportunamente semplificata, i comandi più utilizzati del Control System Toolbox di MATLAB. Per un maggiore approfondimento si rimanda all'uso dei manuali specialistici indicati nella bibliografia e all'help in linea del programma stesso.

Il documento è suddiviso in tre sezioni:

1) Definizione di modelli

TF	*Transfer function* (funzione di trasferimento). Creazione di funzioni di trasferimento o conversione in funzioni di trasferimento.
SS	*State space* (rappresentazione in termini di variabili di stato). Crea modelli in termini di variabili di stato o trasforma in modelli in termini di variabili di stato.
ZPK	*Zero, pole, K.* Creazione di modelli zero, poli, guadagno o conversione in forma zero-poli-guadagno.
SS2TF	*State space to transfer function.* Conversione da modello in variabili di stato a funzione di trasferimento.
TF2SS	*Transfer function to state space*. Conversione da una funzione di trasferimento a un modello in variabili di stato.
RESIDUE	*Residue* (residui) Espansione in frazioni parziali.

2) Risposta di sistemi dinamici

IMPULSE	Risposta all'impulso *(impulse)* di sistemi lineari e stazionari.
STEP	S*tep response* (risposta indiciale).
LSIM	T*ime response* (risposta ad un segnale di ingresso qualunque).

3) Altri comandi utili

ROOTS	Trova le radici *(roots)* di un polinomio.
CONV	Convoluzione e moltiplicazione di polinomi.
DECONV	Deconvoluzione e divisione polinomiale.
REAL	Parte reale *(real)* di un numero complesso.
IMAG	Parte immaginaria *(imaginary)* di un numero complesso.
ZEROS	Array di zeri *(zeros)*.
ONES	Array di uno *(ones)*.
EIG	Autovalori *(eigenvalues)* ed autovettori *(eigenvectors)*.
INV	Matrice inversa.
LAPLACE	Trasformata di Laplace
ILAPLACE	Trasformata inversa di Laplace.

Da notare come sia frequente nell’help in linea del programma MATLAB l’uso dell’acronimo inglese *LTI (linear time-invariant)* ad indicare sistemi lineari e stazionari.

1. Definizione di modelli

TF

E’ possibile creare funzioni di trasferimento con il comando

SYS = TF(NUM,DEN)   modello a tempo continuo NUM(s)/DEN(s)

Per modelli SISO, NUM e DEN sono vettori riga in cui i coefficienti del numeratore e del denominatore sono ordinati secondo potenze decrescenti di s.

SYS = TF(SYS) trasforma un modello lineare e stazionario arbitrario SYS in una funzione di trasferimento. L’uscita SYS è un oggetto del tipo TF.

****

SS

E’ possibile creare un modello in variabili di stato con il comando:

SYS = SS(A,B,C,D) Modello a tempo continuo dove (A,B,C,D) sono le matrici della rappresentazione

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

SYS1 = SS(SYS) trasforma un modello arbitrario SYS1 in un modello in termini di variabili di stato. L’uscita SYS1 è un oggetto del tipo SS.

****
SS2TF

[NUM,DEN] = SS2TF(A,B,C,D) calcola la funzione di trasferimento

$$H(s) = \frac{NUM(s)}{DEN(s)} = C(sI - A)^{-1} B + D$$

del sistema SISO:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Lo stesso comando vale anche per sistemi MIMO.

****
TF2SS

[A,B,C,D] = TF2SS(NUM,DEN) calcola la rappresentazione in variabili di stato

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

del sistema:

$$H(s) = \frac{NUM(s)}{DEN(s)}$$

Il vettore DEN deve contenere i coefficienti del denominatore ordinati secondo potenze decrescenti di s. La matrice NUM deve contenere i coefficienti del numeratore con tante righe quante sono le uscite y. Le matrici A,B,C,D sono indicate nella forma canonica.

****
ZPK

SYS = ZPK(Z,P,K) crea un modello SYS a tempo continuo zero-poli-guadagno (ZPK)
con zeri Z, poli P, e guadagno K. L’uscita SYS è un oggetto ZPK.
Per modelli SISO, Z e P sono vettori di poli e zeri (impostare Z=[] se non ci sono zeri) e K è il guadagno scalare.

SYS = ZPK(SYS) converte un modello SYS lineare e stazionario arbitrario in una rappresentazione ZPK. Il risultato è un oggetto ZPK. Per convertire in un modello SS e TF o viceversa, vedasi anche le funzioni ZP2TF, TF2ZP, ZP2SS, SS2ZP.

****

RESIDUE

[R,P,K] = RESIDUE(B,A) trova i residui, i poli e il termine K = b_n/ a_n (che vale zero se n>m) di un’espansione in frazioni parziali del rapporto di due polinomi B(s)/A(s).
Se non vi sono radici di molteplicità maggiore di uno,

$$\frac{B(s)}{A(s)} = \frac{R(1)}{s-P(1)} + \frac{R(2)}{s-P(2)} + ... + \frac{R(n)}{s-P(n)} + K(s)$$

I vettori B ed A indicano i coefficienti del numeratore e del denominatore del polinomio secondo l’ordine decrescente di s. I residui vengono indicati nel vettore colonna R, la posizione dei poli nel vettore colonna R, ed il termine K = b_n/a_n nel vettore riga K. Il numero di poli è n = length(A)-1 = length(R) = length(P). Il vettore del coefficiente K = b_n/a_n è nullo se length(B) < length(A), altrimenti length(K) = length(B)-length(A)+1.

Se P(j) = ... = P(j+m-1) è un polo di molteplicità m, allora l’espansione include termine della forma

$$\frac{R(j)}{s-P(j)} + \frac{R(j+1)}{(s-P(j))^2} + ... + \frac{R(j+m-1)}{(s-P(j))^2}$$

[B,A] = RESIDUE(R,P,K), con 3 argomenti in ingresso e 2 in uscita,
converte l’espansione in frazioni parziali nuovamente in un polinomio con coefficienti in A e B.

2. Risposta di sistemi dinamici

IMPULSE

IMPULSE(SYS) traccia la risposta all’impulso di un modello LTI SYS (creata con TF, ZPK, o SS). L’intervallo di tempo e il numero di punti sono selezionati in automatico.

IMPULSE(SYS,TFINAL) simula la risposta all’impulso da t = 0 all’istante finale t = TFINAL.

IMPULSE(SYS,T) utilizza il vettore dei tempi T costruito dall’utente per la simulazione. Per sistemi continui, T dovrebbe essere nella forma Ti:dt:Tf dove dt diventerà il tempo di campionamento di una approssimazione discreta al sistema continuo. L’ingresso ad impulso si assume venga sempre applicato nell’istante t = 0 (indipendentemente da Ti).

IMPULSE(SYS1,SYS2,...,T) traccia la risposta all’impulso di più sistemi LTI SYS1, SYS2 in un singolo grafico. Il vettore dei tempi T è opzionale. Si può inoltre specificare un colore, lo stile della linea, una indicazione per ciascun sistema, come in impulse (sys1,'r',sys2,'y--',sys3,'gx').

Quando richiamata con argomenti a primo membro,
[Y,T] = IMPULSE(SYS, ...)

restituisce la risposta d’uscita Y ed il vettore dei tempi T usato per la simulazione. Non viene visualizzato alcun grafico sullo schermo.

****

STEP

STEP(SYS) traccia la risposta al gradino di un modello LTI SYS (creata con TF, ZPK, o SS). L’intervallo di tempo e il numero di punti sono selezionati in automatico.

STEP (SYS,TFINAL) simula la risposta al gradino dall’istante t = 0 all’istante finale t = TFINAL .

STEP (SYS,T) utilizza il vettore dei tempi T costruito dall’utente per la simulazione. Per sistemi continui, T dovrebbe essere nella forma Ti:dt:Tf dove dt diventerà il tempo di campionamento di una approssimazione discreta al sistema continuo. L’ingresso a gradino si assume venga sempre applicato a t = 0 (indipendentemente da Ti).

STEP (SYS1,SYS2,...,T) plotta la risposta al gradino di più sistemi LTI SYS1, SYS2 in un singolo grafico. Il vettore dei tempi T è opzionale. Si può inoltre specificare un colore, lo stile della linea, e un segnaposto per ciascun sistema, come in impulse(sys1,'r',sys2,'y--',sys3,'gx').

Quando richiamata con argomenti a primo membro,
[Y,T] = STEP (SYS, ...)
restituisce la risposta d’uscita Y ed il vettore dei tempi T usato per la simulazione. Non viene visualizzato alcun grafico sullo schermo.

****

LSIM

LSIM(SYS,U,T) traccia la risposta nel tempo di un modello LTI SYS al segnale d’ingresso descritto da U e T. Il vettore dei tempi è costituito di campioni spaziati regolarmente nel tempo ed U è una matrice con tante colonne quanti sono gli ingressi e la cui i-esima riga specifica il valore dell’ingresso al tempo T(i). Per esempio,

t = 0:0.01:5;   u = sin(t);   lsim(sys,u,t)

simula la risposta di SYS u(t) = sin(t) durante 5 secondi.

Nel continuo, il periodo di campionamento T(2)-T(1) dovrebbe essere scelto abbastanza piccolo da evidenziare i dettagli del segnale di ingresso.
LSIM(SYS,U,T,X0) specifica uno stato X0 iniziale diverso da zero (solo per sistemi a variabili di stato).

LSIM(SYS1,SYS2,...,U,T,X0) simula la risposta di un sistema multiplo di sistemi LTI SYS1,SYS2,... in un solo grafico. La condizione iniziale X0 è opzionale. Si può inoltre specificare un colore, uno stile della linea, e un segnaposto per ciascun sistema, come in lsim(sys1,'r',sys2,'y--',sys3,'gx').

Quando richiamata con argomenti a primo membro,

[Y,T] = LSIM(SYS,U,...)

restituisce la storia della risposta d’uscita Y ed il vettore dei tempi T usato per la simulazione. Non viene visualizzato alcun grafico sullo schermo. La matrice Y ha LENGTH(T) righe e tante colonne quante sono le uscite in SYS.

3. Altri comandi utili

ROOTS

ROOTS(C) calcola le radici di un polinomio i cui coefficienti sono elementi del vettore C. Se C ha N+1 componenti, il polinomio è C(1)*X^N + ... + C(N)*X + C(N+1).

****
CONV Convoluzione e moltiplicazione di polinomi.

C = CONV(A, B) esegue la convoluzione dei vettori A e B. Il vettore risultante ha una dimensione pari a LENGTH(A)+LENGTH(B)-1. Se A e B sono vettori le cui componenti rappresentano i coefficienti di un polinomio, farne la convoluzione è equivalente a moltiplicare i due polinomi.

****
DECONV

[Q,R] = DECONV(B,A) esegue la deconvoluzione tra il vettore A ed il vettore B. Il risultato viene restituito nel vettore Q e il resto nel vettore R così che
B = conv(A,Q) + R.

Se A e B sono vettori di coefficienti polinomiali, la deconvoluzione è equivalente alla divisione polinomiale. Il risultato della divisione fra B ed A è il quoziente Q e il resto R.

****
REAL

REAL(X) è la parte reale di X.

****
IMAG

IMAG(X) è la parte immaginaria di X.

****
ZEROS

ZEROS(N) è una matrice NxN di zeri.
ZEROS(M,N) or ZEROS([M,N]) è una matrice MxN di zeri.
ZEROS(SIZE(A)) è una matrice di zeri della stessa dimensione di A.

****
ONES

ONES (N) è una matrice NxN di uno.
ONES (M,N) or ONES ([M,N]) è una matrice MxN di uno.
ONES (SIZE(A)) è una matrice di uno della stessa dimensione di A.

****

EIG

E = EIG(X) è un vettore che contiene gli autovalori di una matrice quadrata X.

[V,D] = EIG(X) genera una matrice diagonale D di autovalori ed una matrice completa V le cui colonne sono i corrispondenti autovettori cosicché X*V = V*D.

****

INV

INV(X) è la matrice inversa della matrice quadrata X.

****

LAPLACE

L = LAPLACE(F) è la trasformata di Laplace della sym scalare F con la variabile indipendente di default t. Il risultato di default è una funzione di s. Se F = F(s), LAPLACE restituisce una funzione di t: L = L(t). Per definizione, L(s) = int(F(t)*exp(-s*t),0,inf), dove la variabile di integrazione è t.

Esempi

```
syms a s t w x
     laplace(t^5)            restituisce   120/s^6
     laplace(exp(a*t))       restituisce   1/(s -a)
     laplace(diff(sym('F(t)')))   restituisce   laplace(F(t),t,s)*s-F(0)
```

****

ILAPLACE

F = ILAPLACE(L) è la trasformata inversa di Laplace della funzione simbolica scalare L (per default, s variabile indipendente e come risultato una funzione di t).

Esempi.

```
syms s t w x y
ilaplace(1/(s-1))          restituisce   exp(t)
ilaplace(1/(s^2+1))        restituisce   sin(s)
```

Bibliografia

[1] *Manuale Matlab a cura di Giuseppe Ciaburro (info@ciaburro.it)*

[2] A.Grace, A.J.Laub, J.N.Little, C.M.Thompson. *Control System Toolbox For Use with Matlab.* The Mathworks Inc.

[3] William J. Palm III. *Matlab 6 per l’ingegneria e le scienze.* Mc Graw Hill