

Conversione tra Rappresentazioni di Sistemi

Un sistema dinamico viene descritto e rappresentato generalmente in vari modi:

1. da un sistema di equazioni di stato con le relative matrici (A,B,C,D) alla f.d.T.;
2. da una funzione di trasferimento con dei polinomi al numeratore e al denominatore allo spazio di stato.

Vediamo come sia possibile passare velocemente da una rappresentazione ad un'altra.

1) Dallo spazio di stato alla f.d.T.

Supponiamo di avere un sistema di equazioni dello spazio di stato e di volerle convertire nella corrispondente f.d.T. Per fare ciò possiamo usare il comando **ss2tf** che applicato ad un sistema lineare fornisce in uscita una matrice di trasferimento o una terna formata separatamente dal polinomio resto, dalla matrice polinomiale al numeratore e dal polinomio caratteristico [Ds,NUM,chi], in modo che $h=NUM/chi + Ds$.

Per prima cosa dobbiamo avere le matrici delle variabili di stato e gli ingressi se più di uno. Poi, grazie al comando **syslin** (sistema lineare), si costruisce la tlist che rappresenta il sistema lineare. A questo punto, con il comando **ss2tf** si ottiene la funzione di trasferimento.

Esempio:

Il nostro sistema di equazioni di stato è il seguente:

$$\begin{aligned}\dot{x} &= v; \\ \dot{v} &= -0.05 v + 0.001 u; \\ y &= v.\end{aligned}$$

Corrispondentemente si ha in Scilab:

```
-->// Iniziamo ad inserire le matrici A, B, C, [D (se diversa da zero)]
```

```
-->A=[0 1;
```

```
-->  0 -0.05];
```

```
-->B=[0; 0.001];
```

```
-->C=[0 1];
```

```
-->D=0; //Essendo D=0, si può anche non inserire.
```

```
-->// Si deve mettere all'inizio 'c' per sistema continuo, 'd' per sistema discreto, [] se il dominio del
```

```
-->//tempo è indefinito o uno scalare che indica il periodo di campionamento in secondi per
```

```
-->// un sistema campionato.
```

-->// Il vettore di ingresso x0 indica lo stato iniziale e per default vale 0.

-->SL=syslin('c',A,B,C) //Ingresso nullo e D nullo.

SL =

SL(1) (state-space system:)

!lss A B C D X0 dt !

SL(2) = A matrix =

! 0. 1. !

! 0. -0.05 !

SL(3) = B matrix =

! 0. !

! 0.001 !

SL(4) = C matrix =

! 0. 1. !

SL(5) = D matrix =

0.

SL(6) = X0 (initial state) =

! 0. !

! 0. !

SL(7) = Time domain =

c

-->// Adesso semplicemente usiamo il comando ss2tf e si ottiene la f.d.T. relativa:

-->h=ss2tf(SL)

h =

0.001

0.05 + s

--> Oppure si ha:

-->[Ds,NUM,chi]=ss2tf(SL)

chi =

0.05 + s

NUM =

0.001

Ds =
0.

Nel caso in cui si hanno dei valori molti piccoli ma non nulli per evitare problemi si utilizza la funzione `clean(A [,epsa [,epsr]])` che mette a 0 tutti i coefficienti con valore assoluto minore di `epsa` e valore relativo minore di `epsr`. Per default sono `epsa=1.d-10` e `epsr=1.d-10`. Per una matrice costante non polinomiale, `clean(A,epsa)` azzerava tutti gli ingressi di $A < \text{di } epsa$.

In questo modo si evita il grosso problema degli zeri all'infinito.

Consideriamo, infatti, come esempio, il seguente sistema di equazioni di stato

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Phi \\ \dot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} \cdot u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \Phi \\ \dot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u$$

Questa volta immettiamo i dati delle matrici A, B, C e D nel file `esempio1.sci`, che deve essere caricato in Scilab, con `exec('C:\Scilab\esempio1.sci',-1)`.

Come nel caso precedente ci creiamo il relativo sistema lineare e troviamo la f.d.T:

```
-->SL1=syslin('c',A,B,C,D);
```

```
-->h1=ss2tf(SL1)
```

h1 =

```
! - 44.545991 - 6.128D-14s + 1.8182s2 - 8.882D-16s3 !
! ----- !
! 6.4785703s - 31.1818s2 + 0.1818s3 + s4 !
! !
! - 7.4373471 + 4.5455s - 6.217D-15s2 !
! ----- !
! 6.4785703 - 31.1818s + 0.1818s2 + s3 !
```

I valori in grassetto sono estremamente piccoli, che possono essere annullati opportunamente con la funzione `clean`.

```
-->h2=clean(ss2tf(SL1))
```

h2 =

```
! - 44.545991 + 1.8182s2 !
```

```

! ----- !
! 6.4785703s - 31.1818s2 + 0.1818s3 + s4 !
!                                     !
!   - 7.4373471 + 4.5455s             !
! ----- !
! 6.4785703 - 31.1818s + 0.1818s2 + s3 !

```

E' possibile fare lo stesso conoscendo le matrici polinomiali N e D oppure conoscendo H: la rappresentazione di una matrice razionale o dello spazio di stato lineare.

2) Dalla f.d.T allo spazio di stato

L'inverso del comando **ss2tf** è **tf2ss**. Esso converte la f.d.T. in un sistema lineare SS.

```
-->SS=tf2ss(h2)
```

```
SS =
```

```
SS(1) (state-space system:)
```

```
!lss A B C D X0 dt !
```

```
SS(2) = A matrix =
```

```
! - 0.4451403 - 0.0034827 5.2653661 8.882D-16 !
! 0.0853253 0. 0.0476334 - 0.0460045 !
! 2.1203829 0.0123107 - 1.6561486 2.0294306 !
! 0.5702658 0.0038676 8.2392148 1.919489 !
```

```
SS(3) = B matrix =
```

```
! 0. !
! 0. !
! 6.6101877 !
! 0.9227186
```

```
SS(4) = C matrix =
```

```
! - 5.065D-16 6.6743339 8.674D-19 - 6.939D-18 !
! 0.1305967 0.0002780 0. 6.939D-18 !
```

```
SS(5) = D matrix =
```

```
! 0. !
! 0. !
```

```
SS(6) = X0 (initial state) =
```

```
! 0. !
```

! 0.!

! 0.!

! 0.!

SS(7) = Time domain =

c

Esempio di come avere le matrici.

-->A1=SS(2)

A1 =

! - 0.4451403 - 0.0034827 5.2653661 8.882D-16 !

! 0.0853253 0. 0.0476334 - 0.0460045 !

! 2.1203829 0.0123107 - 1.6561486 2.0294306 !

! 0.5702658 0.0038676 8.2392148 1.919489 !

Come si può notare abbiamo ottenuto un insieme di matrici diverso da quello di partenza, ma il comportamento dell'ingresso/uscita è lo stesso del precedente. Esistono, infatti, infiniti modi per rappresentare una funzione di trasferimento data nella forma dello spazio di stato. Gli stati non avranno il significato originario.