

# Esercitazione sulla raggiungibilità e controllabilità

## Descrizione del comando SS2SS

Similarity transform.

- $[At, Bt, Ct, Dt] = \text{SS2SS}(A, B, C, D, T)$  performs the similarity transform  $z = Tx$ . The resulting state space system is:  $\dot{z} = [TAT^{-1}] z + [TB] u$ ,  $y = [CT^{-1}] z + D u$

## Descrizione del comando CTRB

Form controllability matrix.

- $\text{CTRB}(A, B)$  returns the controllability matrix  $Kr = [B \ AB \ A^2B \ \dots]$

## Descrizione del comando RANK

Number of linearly independent rows or columns.

- $K = \text{RANK}(X)$  is the number of singular values of  $X$  that are larger than a default value, equal to  $\text{MAX}(\text{SIZE}(X)) * \text{NORM}(X) * \text{EPS}$ .
- $K = \text{RANK}(X, \text{tol})$  is the number of singular values of  $X$  that are larger than  $\text{tol}$ .

## Descrizione del comando CTRBF

Controllability staircase form.

- $[Abar, Bbar, Cbar, T, k] = \text{CTRBF}(A, B, C)$  returns a decomposition into the controllable/uncontrollable subspaces, as described below.  $T$  is the similarity transformation and  $k$  is a row vector such that  $\text{sum}(k)$  is the dimension of the controllable portion of  $A$ .
- $[Abar, Bbar, Cbar, T, k] = \text{CTRBF}(A, B, C, \text{TOL})$  uses tolerance  $\text{TOL}$ .

If  $Kr = \text{CTRB}(A, B)$  has rank  $r \leq n$ , then there is a similarity transformation  $T$  such that

$$Abar = T * A * T' \quad , \quad Bbar = T * B \quad , \quad Cbar = C * T'$$

and the transformed system has the form

$$Abar = \begin{bmatrix} Anc & 0 \\ A21 & Ac \end{bmatrix}, \quad Bbar = \begin{bmatrix} 0 \\ Bc \end{bmatrix}, \quad Cbar = \begin{bmatrix} Cnc & Cc \end{bmatrix}$$

where  $(Ac, Bc)$  is controllable, and  $Cc(sI - Ac)^{-1}Bc = C(sI - A)^{-1}B$ .

## Esempio di utilizzo dei comandi CTRB e CTRBF

Dato il seguente sistema lineare, tempo invariante, a tempo continuo, descritto dalle equazioni  $\dot{x}(t) =$

$$Fx(t) + Gu(t), \quad y(t) = Hx(t), \quad \text{con } F = \begin{bmatrix} 0 & 1/2 & 0 \\ -1/4 & -1/4 & -1/4 \\ 0 & 1/2 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1/4 \\ 0 \end{bmatrix} \text{ ed } H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

studiarne la proprietà di raggiungibilità, determinando se necessario il sistema equivalente in forma canonica di Kalman di raggiungibilità.

$$F = [0, 1/2, 0; -1/4, -1/4, -1/4; 0, 1/2, 0]; \quad G = [0; 1/4; 0]; \quad H = [0, 0, 1];$$

$$Kr = \text{ctrb}(F, G);$$

$$r = \text{rank}(Kr);$$

$$[Fbar, Gbar, Hbar, T, k] = \text{ctrbf}(F, G, H);$$

# Esercitazione sulla retroazione degli stati mediante posizionamento dei poli

## Descrizione del comando PLACE

- $\mathbf{K} = \text{PLACE}(\mathbf{A}, \mathbf{B}, \mathbf{P})$  computes the state feedback matrix  $\mathbf{K}$  such that the eigenvalues of  $\mathbf{A} - \mathbf{B}^* \mathbf{K}$  are those specified in vector  $\mathbf{P}$ .

The complex eigenvalues in the vector  $\mathbf{P}$  must appear in consecutive complex conjugate pairs. No eigenvalue may be placed with multiplicity greater than the number of inputs.

The displayed "ndigits" is an estimate of how well the eigenvalues were placed. The value seems to give an estimate of how many decimal digits in the eigenvalues of  $\mathbf{A} - \mathbf{B}^* \mathbf{K}$  match the specified numbers given in the array  $\mathbf{P}$ . A warning message is printed if the nonzero closed loop poles are greater than 10% from the desired locations specified in  $\mathbf{P}$ .

## Descrizione del comando ACKER

Pole placement gain selection using Ackermann's formula.

- $\mathbf{K} = \text{ACKER}(\mathbf{A}, \mathbf{B}, \mathbf{P})$  calculates the feedback gain matrix  $\mathbf{K}$  such that the single input system  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$  with a feedback law of  $u = -\mathbf{K}\mathbf{x}$  has closed loop poles at the values specified in vector  $\mathbf{P}$ , i.e.,  $\mathbf{P} = \text{eig}(\mathbf{A} - \mathbf{B}^* \mathbf{K})$ .

Note: This algorithm uses Ackermann's formula. This method is NOT numerically reliable and starts to break down rapidly for problems of order greater than 10, or for weakly controllable systems. A warning message is printed if the nonzero closed loop poles are greater than 10% from the desired locations specified in  $\mathbf{P}$ .

## Esempio di utilizzo dei comandi PLACE ed ACKER

Dato il seguente sistema lineare, tempo invariante, a tempo continuo, descritto dalle equazioni  $\dot{\mathbf{x}}(t) =$

$$F\mathbf{x}(t) + G\mathbf{u}(t), \mathbf{y}(t) = H\mathbf{x}(t), \text{ con } F = \begin{bmatrix} 0 & 1/2 & 0 \\ -1/4 & -1/4 & -1/4 \\ 0 & 1/2 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 1/4 \\ 0 \end{bmatrix} \text{ ed } H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

verificare se è possibile fare in modo che gli autovalori del sistema retroazionato mediante posizionamento dei poli siano collocati in:  $\begin{bmatrix} -0.1 & -0.5 & -1 \end{bmatrix}$ ;  $\begin{bmatrix} 0 & -0.5 & -1 \end{bmatrix}$ ;  $\begin{bmatrix} 0 & -0.5 & -0.5 \end{bmatrix}$ .  
Nei casi in cui ciò sia possibile, calcolare il vettore  $\mathbf{K}$  che compare nella legge di controllo  $u(t) = -\mathbf{K}\mathbf{x}(t) + v(t)$ , verificando che i poli in catena chiusa risultino correttamente posizionati.

```
F=[0,1/2,0;-1/4,-1/4,-1/4;0,1/2,0]; G=[0;1/4;0]; H=[0,0,1];
P0=[-0.1,-0.5,-1]; P1=[0,-0.5,-1]; P2=[0,-0.5,-0.5];
tol=1e-6;
r=rank(ctrb(F,G),tol); n=length(F); % r<n => (F,G) non raggiungibile
[Fbar,Gbar,Hbar,T,k]=ctrbf(F,G,H,tol);
Fnr=Fbar(1:n-r,1:n-r);
disp(eig(Fnr)); % n-r autovalori di F-G*K mai posizionabili ad arbitrio
Fr=Fbar(n-r+1:n,n-r+1:n);
Gr=Gbar(n-r+1:n,:);
P1r=[-0.5,-1]; % r autovalori di F-G*K1 posizionabili come in P1
Kbar1=place(Fr,Gr,P1r);
K1=[zeros(1,n-r),Kbar1]*T;
disp(eig(F-G*K1));
P2r=[-0.5,-0.5]; % r autovalori di F-G*K2 posizionabili come in P2
Kbar2=acker(Fr,Gr,P2r);
K2=[zeros(1,n-r),Kbar2]*T;
disp(eig(F-G*K2));
```

# Esercitazione sull'osservabilità

## Descrizione del comando OBSV

Form observability matrix.

- **OBSV(A,C)** returns the observability matrix  $Ko = [C; CA; CA^2 \dots]$

## Descrizione del comando OBSVF

Observability staircase form.

- **[Abar,Bbar,Cbar,T,k] = OBSVF(A,B,C)** returns a decomposition into the observable/unobservable subspaces, as described below. T is the similarity transformation and k is a row vector such that  $\text{sum}(k)$  is the dimension of the observable portion of A.
- **[Abar,Bbar,Cbar,T,k] = OBSVF(A,B,C,TOL)** uses tolerance TOL.

If  $Ko = \text{OBSV}(A,C)$  has rank  $r \leq n$ , then there is a similarity transformation T such that

$$Abar = T * A * T' , Bbar = T * B , Cbar = C * T'$$

and the transformed system has the form

$$Abar = \begin{bmatrix} A_{no} & A_{12} \\ 0 & A_o \end{bmatrix}, Bbar = \begin{bmatrix} B_{no} \\ B_o \end{bmatrix}, Cbar = \begin{bmatrix} 0 & C_o \end{bmatrix}$$

where  $(A_o, C_o)$  is observable, and  $C_o(sI - A_o)^{-1}B_o = C(sI - A)^{-1}B$ .

## Descrizione del comando CANON

State-space to canonical form transformation.

- **[Ab,Bb,Cb,Db] = CANON(A,B,C,D,'type')** transforms the continuous state-space system  $(A,B,C,D)$  into the canonical form specified by 'type':
  - 'modal' transforms the state-space system into modal form where the system eigenvalues appear on the diagonal. The system must be diagonalizable.
  - 'companion' transforms the state-space system into companion canonical form where the characteristic polynomial appears in the right column.
- **[Ab,Bb,Cb,Db,T] = CANON(A,B,C,D,'type')** returns also the transformation matrix T, such that  $z = Tx$

The modal form is useful for determining the relative controllability of the system modes.

The companion form is ill-conditioned and should be avoided if possible.

## Esempio di utilizzo dei comandi OBSV e OBSVF

Dato il seguente sistema lineare, tempo invariante, a tempo continuo, descritto dalle equazioni  $\dot{x}(t) =$

$$Fx(t) + Gu(t), y(t) = Hx(t), \text{ con } F = \begin{bmatrix} 0 & 1/2 & 0 \\ -1/4 & -1/4 & -1/4 \\ 0 & 1/2 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 1/4 \\ 0 \end{bmatrix} \text{ ed } H = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},$$

studiarne la proprietà di osservabilità, determinando se necessario il sistema equivalente in forma canonica di Kalman di osservabilità.

$$F = [0, 1/2, 0; -1/4, -1/4, -1/4; 0, 1/2, 0]; \quad G = [0; 1/4; 0]; \quad H = [0, 1, 0];$$

$$Ko = \text{obsv}(F, H);$$

$$r = \text{rank}(Ko);$$

$$[Fbar, Gbar, Hbar, T, k] = \text{obsvf}(F, G, H);$$

# Esercitazione sulla simulazione di sistemi lineari

## Descrizione del comando LSIM

Simulation of continuous-time linear systems to arbitrary inputs.

- **LSIM(A,B,C,D,U,T)** plots the time response of the linear system:  $\dot{x} = Ax + Bu$ ,  $y = Cx + Du$ , to the input time history U. Matrix U must have as many columns as there are inputs, U. Each row of U corresponds to a new time point, and U must have LENGTH(T) rows. The time vector T must be regularly spaced.
- **LSIM(A,B,C,D,U,T,X0)** can be used if initial conditions exist.
- **LSIM(NUM,DEN,U,T)** plots the time response of the polynomial transfer function  $G(s) = \text{NUM}(s)/\text{DEN}(s)$  where NUM and DEN contain the polynomial coefficients in descending powers of s.

When invoked with left hand arguments,

$$\begin{aligned} [Y,X] &= \text{LSIM}(A,B,C,D,U,T) \\ [Y,X] &= \text{LSIM}(\text{NUM},\text{DEN},U,T) \end{aligned}$$

returns the output and state time history in the matrices Y and X. No plot is drawn on the screen. Y has as many columns as there are outputs, y, and with LENGTH(T) rows. X has as many columns as there are states.

## Descrizione del comando STEP

Step response of continuous-time linear systems.

- **STEP(A,B,C,D,IU)** plots the time response of the linear system:  $\dot{x} = Ax + Bu$ ,  $y = Cx + Du$  to a step applied to the input IU. The time vector is automatically determined.
- **STEP(A,B,C,D,IU,T)** allows the specification of a regularly spaced time vector T.
- **[Y,X] = STEP(A,B,C,D,IU,T)** or **[Y,X,T] = STEP(A,B,C,D,IU)** returns the output and state time response in the matrices Y and X respectively. No plot is drawn on the screen. The matrix Y has as many columns as there are outputs, and LENGTH(T) rows. The matrix X has as many columns as there are states. If the time vector is not specified, then the automatically determined time vector is returned in T.
- **[Y,X] = STEP(NUM,DEN,T)** or **[Y,X,T] = STEP(NUM,DEN)** calculates the step response from the transfer function description  $G(s) = \text{NUM}(s)/\text{DEN}(s)$  where NUM and DEN contain the polynomial coefficients in descending powers of s.

## Esempio #1

Dato il seguente sistema lineare, tempo invariante, a tempo continuo, descritto dalle equazioni  $\dot{x}(t) =$

$$Fx(t) + Gu(t), y(t) = Hx(t), \text{ con } F = \begin{bmatrix} 0 & 1/2 & 0 \\ -1/4 & -1/4 & -1/4 \\ 0 & 1/2 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 1/4 \\ 0 \end{bmatrix} \text{ ed } H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

progettare uno stimatore asintotico dello stato, calcolando il vettore L necessario per posizionarne gli autovalori in  $\begin{bmatrix} -0.2 & -0.3 & -0.4 \end{bmatrix}$ . Simulare l'evoluzione nel tempo del sistema e dello stimatore asintotico, partendo dallo stato iniziale  $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$  del sistema e assumendo nullo lo stato iniziale dello stimatore asintotico, considerando sia il caso di ingresso nullo sia il caso di ingresso a gradino di ampiezza 2.

```

F=[0,1/2,0;-1/4,-1/4,-1/4;0,1/2,0]; G=[0;1/4;0]; H=[0,0,1];
Pstimatore=[-.2,-.3,-.4];
L=place(F',H',Pstimatore)';
[q,n]=size(H);
Ftot=[F,zeros(n,n);L*H,F-L*H]; Gtot=[G;G];
Htot=[H,zeros(q,n);zeros(q,n),H]; Dtot=zeros(2*q,1);
X0=[1;2;3]; X0stimatore=zeros(n,1); X0tot=[X0;X0stimatore];
t=linspace(0,50,1000); u1=zeros(size(t)); u2=2*ones(size(t));
[Ytot1,Xtot1]=lsim(Ftot,Gtot,Htot,Dtot,u1,t,X0tot);
figure(1), plot(t,Xtot1(:,1),'r-',t,Xtot1(:,4),'y:'), title('X1 vero e stimato');
figure(2), plot(t,Xtot1(:,2),'g-',t,Xtot1(:,5),'y:'), title('X2 vero e stimato');
figure(3), plot(t,Xtot1(:,3),'b-',t,Xtot1(:,6),'y:'), title('X3 vero e stimato');
figure(4), plot(t,Ytot1(:,1),'c-',t,Ytot1(:,2),'y:'), title('Y vera e stimata');
[Ytot2,Xtot2]=lsim(Ftot,Gtot,Htot,Dtot,u2,t,X0tot);
figure(5), plot(t,Xtot2(:,1),'r-',t,Xtot2(:,4),'y:'), title('X1 vero e stimato');
figure(6), plot(t,Xtot2(:,2),'g-',t,Xtot2(:,5),'y:'), title('X2 vero e stimato');
figure(7), plot(t,Xtot2(:,3),'b-',t,Xtot2(:,6),'y:'), title('X3 vero e stimato');
figure(8), plot(t,Ytot2(:,1),'c-',t,Ytot2(:,2),'y:'), title('Y vera e stimata');

```

## Esempio #2

Dato il seguente sistema lineare, tempo invariante, a tempo continuo, descritto dalle equazioni  $\dot{x}(t) =$

$$Fx(t) + Gu(t), y(t) = Hx(t), \text{ con } F = \begin{bmatrix} 0 & 1/2 & 0 \\ -1/4 & -1/4 & -1/4 \\ 0 & 1/2 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 1/4 \\ 0 \end{bmatrix} \text{ ed } H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

progettare un regolatore, calcolando i vettori  $K$  ed  $L$  necessari per posizionare i poli del sistema retroazionato in  $\begin{bmatrix} 0 & -0.5 & -1 \end{bmatrix}$  e gli autovalori dello stimatore asintotico in  $\begin{bmatrix} -1 & -1 & -1 \end{bmatrix}$ . Simulare l'evoluzione libera del sistema retroazionato, partendo dallo stato iniziale  $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$  del sistema e assumendo nullo lo stato iniziale dello stimatore asintotico. Si confronti tale risposta con quella del sistema non retroazionato e con quella del sistema controllato mediante retroazione degli stati, utilizzando lo stesso vettore  $K$  precedentemente determinato.

```

F=[0,1/2,0;-1/4,-1/4,-1/4;0,1/2,0]; G=[0;1/4;0]; H=[0,0,1]; D=[0];
Pregolatore=[0,-0.5,-1]; tol=1e-6;
[q,n]=size(H); r=rank(ctrb(F,G),tol); % r<n => (F,G) non raggiungibile
[Fbar,Gbar,Hbar,T,k]=ctrbf(F,G,H,tol);
Fnr=Fbar(1:n-r,1:n-r);
disp(eig(Fnr)); % n-r autovalori di F-G*K mai posizionabili ad arbitrio
Fr=Fbar(n-r+1:n,n-r+1:n); Gr=Gbar(n-r+1:n,:);
Pr=[-0.5,-1]; % r autovalori di F-G*K posizionabili come in P
Kbar=place(Fr,Gr,Pr);
K=[zeros(1,n-r),Kbar]*T;
Pstimatore=[-1,-1,-1];
L=acker(F',H',Pstimatore)';
Ftot=[F,-G*K;L*H,F-G*K-L*H]; Gtot=[G;G];
Htot=[H,zeros(q,n);zeros(q,n),H]; Dtot=zeros(2*q,1);
X0=[1;2;3]; X0stimatore=zeros(n,1); X0tot=[X0;X0stimatore];
t=linspace(0,50,1000); u=zeros(size(t));
[Ynr,Xnr]=lsim(F,G,H,D,u,t,X0); % sistema non retroazionato
[Yrsv,Xrsv]=lsim(F-G*K,G,H,D,u,t,X0); % sistema con retroazione degli stati veri
[Yrsv,Xrsv]=lsim(Ftot,Gtot,Htot,Dtot,u,t,X0tot); % sistema con retroazione
% degli stati stimati
plot(t,Ynr,'r',t,Yrsv,'g',t,Yrsv(:,1),'b',t,Yrsv(:,2),'b:'), grid;

```