Formule LMI con codice Matlab YALMIP

Enrico Giordano

1 Descrizione del sistema

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -B_m/M_m & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -B_s/M_s \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 1/M_m & 0 \\ 0 & 0 \\ 0 & 1/M_s \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ 1/M_m & 0 \\ 0 & 0 \\ 0 & 1/M_s \end{bmatrix} \begin{bmatrix} f_m \\ f_s \end{bmatrix} \tag{1}$$

$$e = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} x \tag{2}$$

1.1 Valori scelti

$$M_s = 0.61$$

$$B_s = 11$$

$$K_v = 40$$

$$K_p = 40$$

$$M_m = 0.64$$

$$B_m = 12$$

1.2 Codice

```
Bs = 11;
Kv = 40;
Kp = 40;
Mm = 0.64;
Bm = 12;

A = [0 1 0 0; 0 (-Bm/Mm) 0 0; 0 0 0 1; 0 0 0 -(Bs/Ms)];
B = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
B0 = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
C0 = [1 0 -1 0; 0 1 0 -1];
C = [0 1 0 0; 0 0 0 1];
```

2 Passività 2x2

$$\begin{bmatrix} QA^T + M^T B^T + AQ + BM & B - QC^T \\ B^T - CQ & 0 \end{bmatrix} \prec 0 \tag{3}$$

$$Q \succ 0 \tag{4}$$

$$K = MP = MQ^{-1}$$

2.1 Codice

```
BlockUpLeft = Q*A'+M'*B0'+A*Q+B0*M;
BlockUpRight = B0-Q*C';
BlockDownLeft = B0'-C*Q;
BlockDownRight = zeros(size(BlockUpRight, 2), size(
    BlockDownLeft, 1));

LMI = [BlockUpLeft BlockUpRight; BlockDownLeft
    BlockDownRight];

F = [Q >= 0,LMI <= 0];

diagnostics = solvesdp(F);
disp(diagnostics.problem);
if diagnostics.problem == 0
 disp('Feasible')
 Q_s = value(Q);
 M_s = value(M);

 K = M_s * Q_s'

elseif diagnostics.problem == 1
 disp('Infeasible')
else
 disp('Something else happened')
end
```

3 Passività 3x3

$$\begin{bmatrix} I & 0 \\ A & B \\ C & D \end{bmatrix}^T \begin{bmatrix} 0 & X & 0 \\ X & 0 & 0 \\ 0 & 0 & P \end{bmatrix} \begin{bmatrix} I & 0 \\ A & B \\ C & D \end{bmatrix}^T \succ 0 \tag{5}$$

$$Q \succ 0 \tag{6}$$

3.1 Codice

```
n = size(A,1)
p = size(B,2)
P = sdpvar(size(A,1),size(A,1), 'symmetric');

F = [P >= 0, [[eye(n,n) zeros(n,p);A B; C D]'*[zeros(n,n)
     P zeros(n,p); P zeros(n,n) zeros(n,p); zeros(p,n)
    zeros(p,n) eye(p,p)]*[eye(n,n) zeros(n,p);A B; C D]]
    >= 0];

diagnostics = solvesdp(F)
disp(diagnostics.problem)
if diagnostics.problem == 0
 disp('Feasible')
 solution = value(P)
elseif diagnostics.problem == 1
 disp('Infeasible')
else
 disp('Something else happened')
end
```

4 Sintesi

$$\begin{bmatrix} (AY+BM)^T+(AY+BM) & B_0 & (C_0Y+EM)^T \\ B_0^T & \gamma^2 I & D_0^T \\ (C_0Y+EM) & D_0 & -I \end{bmatrix} \prec 0 \qquad (7)$$

$$Y \succ 0 \qquad (8)$$

$$D_c = K$$

$$E = 0$$

$$D_0 = 0$$

4.1 Codice

```

n = size(A,1)
p = size(B,1)
q = size(B,2)

Q = sdpvar(n,n, 'symmetric');
M = sdpvar(q,p);
K = sdpvar(q,p);
```

```
E = zeros(q,q);
D0 = E;
Acl = A;

gamma = 0.001;% valore piu' piccolo possibile (es:
    0.000041)

b11 = (A*Q + B*M)'+(A*Q + B*M);
b12 = B0;
b13 = (C0*Q + E*M)';
b21 = B0';
b22 = -gamma^2 * eye(2);
b23 = D0';
b31 = (C0*Q + E*M);
b32 = D0;
b33 = -eye(2);

LMI = [b11 b12 b13; b21 b22 b23; b31 b32 b33];

F = [Q >= 0, LMI <=0];
```

5 Codice completo

```

Ms = 0.61;
Bs = 11;
Kv = 40;
Kp = 40;
Mm = 0.64;
Bm = 12;

s = tf('s');

A = [0 1 0 0; 0 (-Bm/Mm) 0 0; 0 0 0 1; 0 0 0 -(Bs/Ms)];
B = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
B0 = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
C0 = [1 0 -1 0; 0 1 0 -1];
C = [0 1 0 0; 0 0 0 1];

n = size(A,1)
p = size(B,1)
q = size(B,2)

Q = sdpvar(n,n, 'symmetric');
M = sdpvar(q,p);
K = sdpvar(q,p);
```

```
E = zeros(q,q);
D0 = E;
Acl = A;

gamma = 0.001;% valore piu' piccolo possibile (es:
    0.000041)

b11 = (A*Q + B*M)'+(A*Q + B*M);
b12 = B0;
b13 = (C0*Q + E*M)';
b21 = B0';
b22 = -gamma^2 * eye(2);
b23 = D0';
b31 = (C0*Q + E*M);
b32 = D0;
b33 = -eye(2);

LMI1 = [b11 b12 b13; b21 b22 b23; b31 b32 b33];


BlockUpLeft = Q*Acl'+M'*B0'+Acl*Q+B0*M;
BlockUpRight = B0-Q*C';
BlockDownLeft = B0'-C*Q;
BlockDownRight = zeros(size(BlockUpRight, 2), size(
    BlockDownLeft, 1));

LMI2 = [BlockUpLeft BlockUpRight; BlockDownLeft
    BlockDownRight];

F = [Q >= 0, LMI1 <=0, LMI2 <= 0];

diagnostics = solvesdp(F);
disp(diagnostics.problem);
if diagnostics.problem == 0
 disp('Feasible')
 Q_s = value(Q);
 M_s = value(M);

 K = M_s * Q_s'

elseif diagnostics.problem == 1
 disp('Infeasible')
else
 disp('Something else happened')
end
```