

# Formule LMI

Enrico Giordano

September 26, 2016

## 1 Descrizione del sistema

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

$$e = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} x \quad (2)$$

Codice:

```

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

```

## 2 Passività

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

$$Q \succ 0$$

Equazioni:

$$\begin{aligned} Q &= P^{-1} \\ M &= KP^{-1} \\ K &= MP = MQ^{-1} \end{aligned}$$

Codice:

```

1 BlockUpLeft = Q*A'+M'*B0'+A*Q+B0*M;
2 BlockUpRight = B0-Q*C';
3 BlockDownLeft = B0'-C*Q;
4 BlockDownRight = zeros(size(BlockUpRight, 2), size(
    BlockDownLeft, 1));
5
6 LMI = [BlockUpLeft BlockUpRight; BlockDownLeft
    BlockDownRight];
7
8 F = [Q >= 0, LMI <= 0];
9
10 diagnostics = solvesdp(F);
11 disp(diagnostics.problem);
12 if diagnostics.problem == 0
13     disp('Feasible')
14     Q_s = value(Q);
15     M_s = value(M);
16
17     K = M_s * Q_s';
18
19 elseif diagnostics.problem == 1
20     disp('Infeasible')
21 else
22     disp('Something else happened')
23 end

```

### 3 Sintesi

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

$Y \succ 0$

Equazioni:

$$\begin{aligned}
 D_c &= K \\
 E &= 0
 \end{aligned}$$

$$D_0 = 0$$

Codice:

```

1
2 n = size(A,1)
3 p = size(B,1)
4 q = size(B,2)
5
6 Q = sdpvar(n,n, 'symmetric');
7 M = sdpvar(q,p);
8 K = sdpvar(q,p);
9 E = zeros(q,q);
10 D0 = E;
11 %Acl = A + B*K;
12 Acl = A;
13
14 gamma = 0.001;%0.000041;
15
16 b11 = (A*Q + B*M)' + (A*Q + B*M);
17 b12 = B0;
18 b13 = (C0*Q + E*M)';
19 b21 = B0';
20 b22 = -gamma^2 * eye(2);
21 b23 = D0';
22 b31 = (C0*Q + E*M);
23 b32 = D0;
24 b33 = -eye(2);
25
26 LMI = [b11 b12 b13; b21 b22 b23; b31 b32 b33];
27
28 F = [Q >= 0, LMI1 <=0];

```

## 4 Codice completo

```

1
2 Ms = 0.61;
3 Bs = 11;
4 Kv = 40;
5 Kp = 40;
6 Mn = 0.64;
7 Bm = 12;
8

```

```

9  s = tf( 's' );
10
11  A = [0 1 0 0; 0 (-Bm/Mm) 0 0; 0 0 0 1; 0 0 0 -(Bs/Ms)];
12  B = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
13  B0 = [0 0; 1/Mm 0; 0 0; 0 1/Ms];
14  C0 = [1 0 -1 0; 0 1 0 -1];
15  C = [0 1 0 0; 0 0 0 1];
16
17  n = size(A,1)
18  p = size(B,1)
19  q = size(B,2)
20
21  Q = sdpvar(n,n, 'symmetric');
22  M = sdpvar(q,p);
23  K = sdpvar(q,p);
24  E = zeros(q,q);
25  D0 = E;
26  %Acl = A + B*K;
27  Acl = A;
28
29  gamma = 0.001;%0.000041;
30
31  b11 = (A*Q + B*M)' + (A*Q + B*M);
32  b12 = B0;
33  b13 = (C0*Q + E*M)';
34  b21 = B0';
35  b22 = -gamma^2 * eye(2);
36  b23 = D0';
37  b31 = (C0*Q + E*M);
38  b32 = D0;
39  b33 = -eye(2);
40
41  LMI1 = [b11 b12 b13; b21 b22 b23; b31 b32 b33];
42
43
44  BlockUpLeft = Q*Acl' + M'*B0' + Acl*Q + B0*M;
45  BlockUpRight = B0 - Q*C';
46  BlockDownLeft = B0' - C*Q;
47  BlockDownRight = zeros(size(BlockUpRight, 2), size(
    BlockDownLeft, 1));
48
49  LMI2 = [BlockUpLeft BlockUpRight; BlockDownLeft
    BlockDownRight];
50
51  F = [Q >= 0, LMI1 <= 0, LMI2 <= 0];
52

```

```

53 diagnostics = solvesdp(F);
54 disp(diagnostics.problem);
55 if diagnostics.problem == 0
56     disp('Feasible')
57     Q_s = value(Q);
58     M_s = value(M);
59
60     K = M_s * Q_s'
61
62 elseif diagnostics.problem == 1
63     disp('Infeasible')
64 else
65     disp('Something else happened')
66 end

```