

10 Non-Classic State Feedback Control Design

10.1 Decentralized LQR Control

Problem: Given the LTI system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t).\end{aligned}$$

where

$$x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{pmatrix}, \quad x_i \in \mathbb{R}^{n_i}, \quad \sum_{i=1}^N n_i = n,$$

compute a decentralized state feedback controller

$$u_i(t) = K_i x_i(t), \quad u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_N(t) \end{pmatrix}$$

that stabilizes the closed loop system and minimizes

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)]$$

Assumptions:

a) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

Solution: We do not know of a complete solution to this problem. Here we will be happy with a sufficient design (conservative). The key is to note that

$$u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_N(t) \end{pmatrix} = \begin{pmatrix} K_1 x_1(t) \\ \vdots \\ K_N x_N(t) \end{pmatrix} = K_D x(t)$$

where

$$K_D = \begin{bmatrix} K_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & K_N \end{bmatrix}$$

is block-diagonal!

Using the congruence+change-of-variables the LQR problem can be recast as the SDP

$$\begin{aligned} \min_{X \in \mathbb{S}^n, L \in \mathbb{R}^{m \times n}, Z \in \mathbb{S}^r} \quad & \text{trace}(ZW) \\ \text{s.t.} \quad & \begin{bmatrix} AX + XA^T + B_u L + L^T B_u^T & XC_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0, \\ & \begin{bmatrix} Z & B_w^T \\ B_w & X \end{bmatrix} \succeq 0, \quad X \succ 0. \end{aligned}$$

where $K = LX^{-1}$. Now note that under the additional constraint $X = X_D$ and $L = L_D$, i.e. X and L are block-diagonal then

$$X = X_D \text{ and } L = L_D \quad \implies \quad L_D X_D^{-1} = K_D.$$

Note that the original problem does not require $X = X_D$!

Also note that

$$\text{trace}(ZW) \geq J = \lim_{t \rightarrow \infty} E [z(t)^T z(t)]$$

Recipe II (existence condition) does not provide any solution to this problem. (why?)

10.1.1 Summary: Decentralized LQR Control

The decentralized state feedback controller

$$u(t) = K_D x(t), \quad K_D = L_D X_D^{-1}$$

where $X \in \mathbb{S}^n$, $L \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{S}^r$ solve the SDP

$$\begin{aligned} \min_{X \in \mathbb{S}^n, L \in \mathbb{R}^{m \times n}, Z \in \mathbb{S}^r} \quad & \text{trace}(ZW) \\ \text{s.t.} \quad & \begin{bmatrix} AX + XA^T + B_u L + L^T B_u^T & X C_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0, \\ & \begin{bmatrix} Z & B_w^T \\ B_w & X \end{bmatrix} \succeq 0, \quad X \succ 0, \\ & X = X_D, \quad L = L_D, \end{aligned}$$

is a suboptimal solution to the problem of stabilizing the closed loop system and minimizing the cost function

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)] \leq \text{trace}(ZW)$$

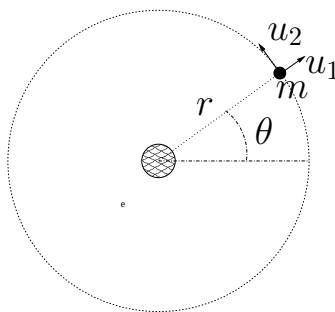
for the LTI system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t) \end{aligned}$$

under the assumptions

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) (A, B_u) stabilizable;
- c) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

10.2 Example: satellite in circular orbit



Satellite of mass m with thrust in the radial direction u_1 and in the tangential direction u_2 . Continuing...

$$\begin{aligned} m(\ddot{r} - r\dot{\theta}^2) &= u_1 - \frac{km}{r^2} + w_1, \\ m(2\dot{r}\dot{\theta} + r\ddot{\theta}) &= u_2 + w_2, \end{aligned}$$

where w_1 and w_2 are independent white noise disturbances with variances δ_1 and δ_2 .

As before, putting in state space and linearizing around

$$\bar{x}(t) = \begin{pmatrix} \bar{r} \\ \bar{\omega}t \\ 0 \\ \bar{\omega} \end{pmatrix}, \quad \bar{u} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} = 0, \quad \bar{w} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \end{pmatrix} = 0, \quad \bar{\omega} = \sqrt{k/\bar{r}^3}$$

one obtains the linearized system

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\bar{\omega}^2 & 0 & 0 & 2\bar{r}\bar{\omega} \\ 0 & 0 & -2\bar{\omega}/\bar{r} & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

Problem: Given

$$m = 100 \text{ kg}, \quad \bar{r} = R + 300 \text{ km}, \quad \bar{k} = GM$$

where $G \approx 6.673 \times 10^{-11} \text{ N m}^2/\text{kg}^2$ is the universal gravitational constant, and $M \approx 5.98 \times 10^{24} \text{ kg}$ and $R \approx 6.37 \times 10^3 \text{ km}$ are the mass and radius of the earth. If the variances δ_1 and δ_2 are $0.1N$ find solutions to the LQR control problem where

$$Q = I, \quad R = \rho I,$$

and u_1 only feedbacks r and $\dot{r}$, u_2 only feedbacks θ and $\dot{\theta}$.

In order for this problem to be cast as a 'decentralized' control problem we need to reorder the state. This can be done through a similarity transformation

$$\tilde{x} = \begin{pmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{pmatrix} = Tx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{pmatrix}$$

and partition the state

$$\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}, \quad \tilde{x}_1 = \begin{pmatrix} r \\ \dot{r} \end{pmatrix}, \quad \tilde{x}_2 = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$$

so that the decentralized control

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad u_1 = K_1 \tilde{x}_1, \quad u_2 = K_2 \tilde{x}_2$$

have the desired structure.

```
% MAE 280 B - Linear Control Design
% Mauricio de Oliveira
%
% LMI LQR Control
%
m = 100;           % 100 kg
r = 300E3;         % 300 km
R = 6.37E6;        % 6.37 10^3 km
G = 6.673E-11;     % 6.673 N m^2/kg^2
M = 5.98E24;       % 5.98 10^24 kg
k = G * M;         % gravitational force constant
w = sqrt(k/((R+r)^3)); % angular velocity (rad/s)
v = w * (R + r);   % "ground" velocity (m/s)

% linearized system matrices
A = [0 0 1 0; 0 0 0 1; 3*w^2 0 0 2*(r+R)*w; 0 0 -2*w/(r+R) 0];
Bu = [0 0; 0 0; 1/m 0; 0 1/(m*r)];
Bw = [0 0; 0 0; 1/m 0; 0 1/(m*r)];

% noise variances
W = 0.1 * eye(2)
W =
    1.0000e-01    0
         0    1.0000e-01

% scale
T = diag([1 r 1 r])
T =
         1         0         0         0
```

```

0      300000      0      0
0      0      1      0
0      0      0      300000

% similarity transformation
At = T * A / T
At =
0      0      1.0000e+00      0
0      0      0      1.0000e+00
4.0343e-06      0      0      5.1565e-02
0      0      -1.0432e-04      0

But = T * Bu
But =
0      0
0      0
1.0000e-02      0
0      1.0000e-02

Bwt = T * Bw
Bwt =
0      0
0      0
1.0000e-02      0
0      1.0000e-02

% another similarity to change state
T = [1 0 0 0; 0 0 1 0; 0 1 0 0; 0 0 0 1]
T =
1      0      0      0
0      0      1      0
0      1      0      0
0      0      0      1

Att = T * At / T
Att =
0      1.0000e+00      0      0
4.0343e-06      0      0      5.1565e-02
0      0      0      1.0000e+00
0      -1.0432e-04      0      0

Butt = T * But
Butt =
0      0
1.0000e-02      0
0      0
0      1.0000e-02

Bwtt = T * Bwt
Bwtt =
0      0
1.0000e-02      0
0      0
0      1.0000e-02

% optimal LQR state feedback control (using u1 and u2)
n = size(Att,1);
m = size(Butt,2);

```

```

% LQR solution
Cz = [eye(n); zeros(m,n)];
Dzu = [zeros(n,m); eye(m)];

[K,X,S] = lqr(Att, Butt, Cz' * Cz, Dzu' * Dzu);
Klqr = - K
Klqr =
    -9.8444e-01    -1.3843e+01     1.7795e-01    -2.4919e+00
    -1.7795e-01    -2.4919e+00    -9.8404e-01    -1.4741e+01

% LMI solution with decentralization
% declare variables
X = sdpvar(n,n,'symmetric');
Z = sdpvar(m,m,'symmetric');
L = sdpvar(m,n);

% declare LMIs
LMI1 = [Att*X+X*Att'+Butt*L+L'*Butt', X*Cz'+L'*Dzu';...
        Cz*X+Dzu*L, -eye(m+n)];
LMI2 = [Z, Bwtt'; Bwtt, X];

% Structural constraints
Xmask = [zeros(2,2) ones(2,2); ones(2,2) zeros(2,2)]
Xmask =
     0     0     1     1
     0     0     1     1
     1     1     0     0
     1     1     0     0
Lmask = [zeros(1,2) ones(1,2); ones(1,2) zeros(1,2)]
Lmask =
     0     0     1     1
     1     1     0     0

pause

LMI = set(LMI1 < 0) + set(LMI2 > 0) + set(Xmask.*X == 0) + set(Lmask.*L == 0);
options = sdpsettings('solver','sedumi');
solution = solvesdp(LMI,trace(W*X),options)
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
Put 24 free variables in a quadratic cone
eqs m = 21, order n = 19, dim = 162, blocks = 4
nnz(A) = 77 + 0, nnz(ADA) = 393, nnz(L) = 207
it :      b*y      gap   delta  rate   t/tP*   t/tD*   feas cg cg   prec
 0 :              5.84E+00 0.000
 1 :  -1.21E-01  1.26E+00 0.000 0.2164 0.9000 0.9000   2.32  1  1  1.5E+00
 2 :  -3.23E-02  2.56E-01 0.000 0.2028 0.9000 0.9000   1.54  1  1  7.1E-01
 3 :  -6.52E-03  5.86E-02 0.000 0.2284 0.9000 0.9000   1.23  1  1  5.4E-01
 4 :  -3.90E-03  1.63E-02 0.000 0.2775 0.9000 0.9000   1.04  1  1  7.3E-01
 5 :  -5.48E-03  4.82E-03 0.000 0.2964 0.9000 0.9000   0.56  1  1  2.8E-01
 6 :  -7.91E-03  1.61E-03 0.000 0.3348 0.9000 0.9000   0.14  1  1  1.1E-01
 7 :  -1.13E-02  5.44E-04 0.000 0.3370 0.9000 0.9000   0.14  1  1  7.1E-03
 8 :  -1.47E-02  2.05E-04 0.000 0.3770 0.9000 0.9000   0.10  1  1  6.6E-04
 9 :  -1.92E-02  6.99E-05 0.000 0.3412 0.9000 0.9000   0.17  1  1  3.6E-04

```

```

10 : -2.28E-02 2.80E-05 0.000 0.4005 0.9000 0.9000 0.13 1 1 2.4E-04
11 : -2.76E-02 9.32E-06 0.000 0.3328 0.9000 0.9000 0.34 1 1 1.1E-04
12 : -3.09E-02 3.44E-06 0.000 0.3686 0.9000 0.9000 0.39 1 1 5.9E-05
13 : -3.35E-02 9.53E-07 0.000 0.2775 0.9000 0.9000 0.64 1 1 1.9E-05
14 : -3.46E-02 2.27E-07 0.000 0.2379 0.9000 0.9000 0.79 1 1 5.2E-06
15 : -3.50E-02 1.23E-08 0.000 0.0542 0.9900 0.9900 0.94 1 1 2.9E-07
16 : -3.50E-02 2.75E-09 0.000 0.2234 0.9000 0.9000 0.97 1 1 6.6E-08
17 : -3.50E-02 2.11E-11 0.056 0.0077 0.9990 0.9990 1.00 1 1 5.1E-10

iter seconds digits      c*x          b*y
17      0.3    7.0 -3.5011343878e-02 -3.5011340045e-02
|Ax-b| = 3.0e-10, [Ay-c]_+ = 1.4E-10, |x|= 2.8e+02, |y|= 2.9e-01

Detailed timing (sec)
      Pre      IPM      Post
1.840E-01    2.947E-01    3.121E-02
Max-norms: ||b||=1.000000e-01, ||c|| = 1,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 391.968.
solution =
      yalmiptime: 6.4425e-01
      solvertime: 5.7987e-01
      info: 'No problems detected (SeDuMi-1.3)'
      problem: 0
      dimacs: [NaN NaN NaN NaN NaN NaN]

pause
??? Operation terminated by user during ==> sat8 at 76
double(X)
ans =
      1.3231e-01 -8.7973e-03 -5.8520e-13 -1.0460e-11
      -8.7973e-03 1.1741e-03 -4.3354e-12 -1.0041e-10
      -5.8520e-13 -4.3354e-12 7.9953e-02 -4.4339e-03
      -1.0460e-11 -1.0041e-10 -4.4339e-03 8.0025e-04
quit

```

10.3 Decentralized LQR Control — Dual form

10.3.1 LQR Control — Dual form

When we converted the LQR problem to LMIs we worked with the cost function

$$J = \text{trace} (P_o B_w W B_w^T)$$

where P_o is the observability Gramian

$$(A + B_u K)^T P_o + P_o (A + B_u K) + (C_z + D_{zu} K)^T (C_z + D_{zu} K) = 0.$$

An alternative form is provided by using the dual cost function

$$J = \text{trace} [(C_z + D_{zu} K) P_c (C_z + D_{zu} K)^T]$$

where P_c is the controllability Gramian

$$(A + B_u K) P_c + P_c (A + B_u K)^T + B_w W B_w^T = 0.$$

As before, use the Comparison Lemma to conclude that for any K such that $(A + B_u K)$ is Hurwitz then

$$X \succeq P_c \quad \implies \quad \text{trace} [(C_z + D_{zu} K) X (C_z + D_{zu} K)^T] \geq J$$

where X satisfies the inequality

$$(A + B_u K) X + X (A + B_u K)^T + B_w W B_w^T \preceq 0.$$

Recipe I (congruence + change-of-variables):
 Apply change-of-variables

$$L := KX$$

to obtain the LMI

$$AX + XA^T + B_u L + L^T B_u^T + B_w W B_w^T \preceq 0$$

As for the cost function, introduce an extra variable Z such that

$$Z \succeq (C_z + D_{zu}K)X(C_z + D_{zu}K)^T$$

and use Schur Complement

$$Z \succeq (C_z + D_{zu}K)X(C_z + D_{zu}K)^T \Leftrightarrow \begin{bmatrix} Z & (C_z + D_{zu}K)X \\ X(C_z + D_{zu}K)^T & X \end{bmatrix} \succeq 0.$$

Applying the change-of-variables $L = KX^{-1}$ this becomes

$$\begin{bmatrix} Z & C_z X + D_{zu} L \\ X C_z^T + L^T D_{zu}^T & X \end{bmatrix} \succeq 0.$$

10.3.2 Summary: LQR revisited (dual form)

The state feedback controller

$$u(t) = Kx(t), \quad K = LX^{-1}$$

where $X \in \mathbb{S}^n$, $L \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{S}^r$ solve the SDP

$$\begin{aligned} \min_{X \in \mathbb{S}^n, L \in \mathbb{R}^{m \times n}, Z \in \mathbb{S}^r} \quad & \text{trace}(Z) \\ \text{s.t.} \quad & AX + XA^T + B_u L + L^T B_u^T + B_w W B_w^T \preceq 0, \\ & \begin{bmatrix} Z & C_z X + D_{zu} L \\ X C_z^T + L^T D_{zu}^T & X \end{bmatrix} \succeq 0, \quad X \succ 0. \end{aligned}$$

stabilizes the closed loop system and minimizes the cost function

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)]$$

for the LTI system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t) \end{aligned}$$

under the assumptions

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) (A, B_u) stabilizable;
- c) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

10.3.3 Summary: Decentralized LQR Control (dual form)

The decentralized state feedback controller

$$u(t) = K_D x(t), \quad K_D = L_D X_D^{-1}$$

where $X \in \mathbb{S}^n$, $L \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{S}^r$ solve the SDP

$$\begin{aligned} \min_{X \in \mathbb{S}^n, L \in \mathbb{R}^{m \times n}, Z \in \mathbb{S}^r} \quad & \text{trace}(Z) \\ \text{s.t.} \quad & AX + XA^T + B_u L + L^T B_u^T + B_w W B_w^T \preceq 0, \\ & \begin{bmatrix} Z & C_z X + D_{zu} L \\ X C_z^T + L^T D_{zu}^T & X \end{bmatrix} \succeq 0, \quad X \succ 0 \\ & X = X_D, \quad L = L_D, \end{aligned}$$

is a suboptimal solution to the problem of stabilizing the closed loop system and minimizing the cost function

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)] \leq \text{trace}(Z)$$

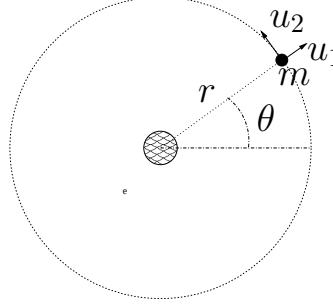
for the LTI system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t) \end{aligned}$$

under the assumptions

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) (A, B_u) stabilizable;
- c) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

10.4 Example: satellite in circular orbit



Satellite of mass m with thrust in the radial direction u_1 and in the tangential direction u_2 . Continuing...

$$\begin{aligned} m(\ddot{r} - r\dot{\theta}^2) &= u_1 - \frac{km}{r^2} + w_1, \\ m(2\dot{r}\dot{\theta} + r\ddot{\theta}) &= u_2 + w_2, \end{aligned}$$

where w_1 and w_2 are independent white noise disturbances with variances δ_1 and δ_2 .

As before, putting in state space and linearizing around

$$\bar{x}(t) = \begin{pmatrix} \bar{r} \\ \bar{\omega}t \\ 0 \\ \bar{\omega} \end{pmatrix}, \quad \bar{u} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} = 0, \quad \bar{w} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \end{pmatrix} = 0, \quad \bar{\omega} = \sqrt{k/\bar{r}^3}$$

one obtains the linearized system

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\bar{\omega}^2 & 0 & 0 & 2\bar{r}\bar{\omega} \\ 0 & 0 & -2\bar{\omega}/\bar{r} & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

Problem: Given

$$m = 100 \text{ kg}, \quad \bar{r} = R + 300 \text{ km}, \quad \bar{k} = GM$$

where $G \approx 6.673 \times 10^{-11} \text{ N m}^2/\text{kg}^2$ is the universal gravitational constant, and $M \approx 5.98 \times 10^{24} \text{ kg}$ and $R \approx 6.37 \times 10^3 \text{ km}$ are the mass and radius of the earth. If the variances δ_1 and δ_2 are $0.1N$ find solutions to the LQR control problem where

$$Q = I, \quad R = \rho I,$$

and u_1 only feedbacks r and $\dot{r}$, u_2 only feedbacks θ and $\dot{\theta}$.

As before apply the similarity transformation

$$\tilde{x} = \begin{pmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{pmatrix} = T x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{pmatrix}$$

and partition the state

$$\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}, \quad \tilde{x}_1 = \begin{pmatrix} r \\ \dot{r} \end{pmatrix}, \quad \tilde{x}_2 = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$$

so that the decentralized control

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad u_1 = K_1 \tilde{x}_1, \quad u_2 = K_2 \tilde{x}_2$$

have the desired structure, but now use the dual form.

```
% MAE 280 B - Linear Control Design
% Mauricio de Oliveira
%
% LMI LQR Control
%
m = 100;                % 100 kg
r = 300E3;              % 300 km
R = 6.37E6;             % 6.37 10^3 km
G = 6.673E-11;          % 6.673 N m^2/kg^2
M = 5.98E24;           % 5.98 10^24 kg
k = G * M;             % gravitational force constant
w = sqrt(k/((R+r)^3)); % angular velocity (rad/s)
v = w * (R + r);       % "ground" velocity (m/s)

% linearized system matrices
A = [0 0 1 0; 0 0 0 1; 3*w^2 0 0 2*(r+R)*w; 0 0 -2*w/(r+R) 0];
Bu = [0 0; 0 0; 1/m 0; 0 1/(m*r)];
Bw = [0 0; 0 0; 1/m 0; 0 1/(m*r)];

% noise variances
W = 0.1 * eye(2)
W =
    1.0000e-01    0
         0    1.0000e-01

% scale
T = diag([1 r 1 r])
T =
         1         0         0         0
         0    300000         0         0
         0         0         1         0
```

```

0          0          0          300000

% similarity transformation
At = T * A / T
At =
    0          0  1.0000e+00          0
    0          0          0  1.0000e+00
  4.0343e-06          0          0  5.1565e-02
    0          0 -1.0432e-04          0

But = T * Bu
But =
    0          0
    0          0
  1.0000e-02          0
    0  1.0000e-02

Bwt = T * Bw
Bwt =
    0          0
    0          0
  1.0000e-02          0
    0  1.0000e-02

% another similarity to change state
T = [1 0 0 0; 0 0 1 0; 0 1 0 0; 0 0 0 1]
T =
    1    0    0    0
    0    0    1    0
    0    1    0    0
    0    0    0    1

Att = T * At / T
Att =
    0  1.0000e+00          0          0
  4.0343e-06          0          0  5.1565e-02
    0          0          0  1.0000e+00
    0 -1.0432e-04          0          0

Butt = T * But
Butt =
    0          0
  1.0000e-02          0
    0          0
    0  1.0000e-02

Bwtt = T * Bwt
Bwtt =
    0          0
  1.0000e-02          0
    0          0
    0  1.0000e-02

% optimal LQR state feedback control (using u1 and u2)
n = size(Att,1);
m = size(Butt,2);

% LQR solution
Cz = [eye(n); zeros(m,n)];

```

```

Dzu = [zeros(n,m); eye(m)];

[K,X,S] = lqr(Att, Butt, Cz' * Cz, Dzu' * Dzu);
Klqr = - K
Klqr =
    -9.8444e-01    -1.3843e+01     1.7795e-01    -2.4919e+00
    -1.7795e-01    -2.4919e+00    -9.8404e-01    -1.4741e+01

% LMI solution with decentralization
% declare variables
X = sdpvar(n,n,'symmetric');
Z = sdpvar(m+n,m+n,'symmetric');
L = sdpvar(m,n);

% declare LMIs
LMI1 = Att*X+X*Att'+Butt*L+L'*Butt'+Bwtt*W*Bwtt';
LMI2 = [Z, Cz*X+Dzu*L; X*Cz'+L'*Dzu', X];

% Structural constraints
Xmask = [zeros(2,2) ones(2,2); ones(2,2) zeros(2,2)]
Xmask =
     0     0     1     1
     0     0     1     1
     1     1     0     0
     1     1     0     0
Lmask = [zeros(1,2) ones(1,2); ones(1,2) zeros(1,2)]
Lmask =
     0     0     1     1
     1     1     0     0

LMI = set(LMI1 < 0) + set(LMI2 > 0) + set(Xmask.*X == 0) + set(Lmask.*L == 0);
options = sdpsettings('solver','sedumi');
solution = solvesdp(LMI,trace(Z),options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
Put 24 free variables in a quadratic cone
eqs m = 39, order n = 17, dim = 142, blocks = 4
nnz(A) = 95 + 0, nnz(ADA) = 1521, nnz(L) = 780
it :      b*y      gap   delta  rate  t/tP*  t/tD*   feas cg cg  prec
 0 :              2.29E+00 0.000
 1 :    6.89E-02 4.60E-01 0.000 0.2008 0.9000 0.9000   2.20 1 1 8.0E-01
 2 :    2.73E-04 1.74E-02 0.000 0.0377 0.9900 0.9900   1.36 1 1 2.5E-02
 3 :   -1.31E-05 7.63E-05 0.000 0.0044 0.9990 0.9990   1.02 1 1 1.1E-04
 4 :   -1.69E-04 3.14E-05 0.000 0.4116 0.9000 0.9000   0.14 1 1 1.3E-04
 5 :   -6.40E-04 1.02E-05 0.000 0.3230 0.9000 0.9000  -0.50 1 1 1.2E-04
 6 :   -2.74E-03 2.34E-06 0.000 0.2310 0.9000 0.9000  -0.75 1 1 9.3E-05
 7 :   -1.02E-02 6.18E-07 0.000 0.2634 0.9000 0.9000  -0.90 1 1 8.7E-05
 8 :   -2.00E-02 2.64E-07 0.000 0.4279 0.9000 0.9000  -0.79 1 1 7.1E-05
 9 :   -2.41E-02 9.08E-08 0.000 0.3436 0.9000 0.9000   0.10 1 1 2.7E-05
10 :   -3.08E-02 2.13E-08 0.000 0.2340 0.9000 0.9000   0.39 1 1 8.4E-06
11 :   -3.33E-02 1.15E-09 0.000 0.0542 0.9900 0.9900   0.83 1 1 4.9E-07
12 :   -3.34E-02 1.28E-10 0.000 0.1112 0.9450 0.9450   0.99 1 1 5.5E-08
13 :   -3.34E-02 2.31E-11 0.003 0.1808 0.9000 0.9000   1.00 1 2 9.9E-09
14 :   -3.34E-02 4.66E-12 0.000 0.2013 0.9000 0.9000   1.00 2 2 2.0E-09

```

```

15 : -3.34E-02 2.81E-13 0.342 0.0603 0.9900 0.9900 1.00 2 2 1.2E-10

iter seconds digits      c*x          b*y
15      0.1      Inf -3.3440240240e-02 -3.3440235163e-02
|Ax-b| = 1.3e-10, [Ay-c]_+ = 1.6E-11, |x|= 3.5e+03, |y|= 2.1e-02

Detailed timing (sec)
      Pre      IPM      Post
0.000E+00 1.400E-01 0.000E+00
Max-norms: ||b||=1, ||c|| = 1.000000e-05,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 282.571.
solution =
      yalmiptime: 8.7827e-02
      solvertime: 1.4310e-01
      info: 'No problems detected (SeDuMi-1.1)'
      problem: 0
      dimacs: [6.4794e-11 0 0 1.6040e-11 -4.7582e-09 -4.7574e-09]

% Construct K and check closed loop stability and performance
double(L)
ans =
-4.1530e-03 -5.8787e-04 -5.4486e-14 -1.1347e-12
-8.1442e-14 -1.1010e-12 -2.9304e-03 -5.6459e-04
double(X)
ans =
4.1553e-03 -3.1031e-10 3.2566e-15 -3.7495e-13
-3.1031e-10 4.1513e-05 -1.9902e-13 -9.7942e-12
3.2566e-15 -1.9902e-13 2.9331e-03 -1.3147e-10
-3.7495e-13 -9.7942e-12 -1.3147e-10 2.9304e-05
Ld = double(L) - double(L) .* Lmask
Ld =
-4.1530e-03 -5.8787e-04 0 0
0 0 -2.9304e-03 -5.6459e-04
Xd = double(X) - double(X) .* Xmask
Xd =
4.1553e-03 -3.1031e-10 0 0
-3.1031e-10 4.1513e-05 0 0
0 0 2.9331e-03 -1.3147e-10
0 0 -1.3147e-10 2.9304e-05
Klmi = Ld / Xd
Klmi =
-9.9944e-01 -1.4161e+01 0 0
0 0 -9.9908e-01 -1.9267e+01
Acl = Att + Buttt*Klmi;
eig(Acl)
ans =
-7.0859e-02 + 7.0494e-02i
-7.0859e-02 - 7.0494e-02i
-9.6280e-02 + 2.6849e-02i
-9.6280e-02 - 2.6849e-02i
Y = lyap(Acl, Bwtt * W * Bwtt')
Y =
3.6796e-03 7.4539e-20 3.9952e-04 -4.3182e-10
7.4539e-20 3.6761e-05 4.3182e-10 3.9915e-06

```

```

    3.9952e-04    4.3182e-10    2.5973e-03    1.0164e-19
   -4.3182e-10    3.9915e-06    1.0164e-19    2.5949e-05
sqrt(trace(Y))
ans =
    7.9622e-02
sqrt(diag(Y))
ans =
    6.0660e-02
    6.0631e-03
    5.0964e-02
    5.0941e-03

% Try stripping lqr gain
Klqr
Klqr =
   -9.8444e-01   -1.3843e+01    1.7795e-01   -2.4919e+00
   -1.7795e-01   -2.4919e+00   -9.8404e-01   -1.4741e+01
Klqrd = double(Klqr) - double(Klqr) .* Imask
Klqrd =
   -9.8444e-01   -1.3843e+01         0         0
         0         0   -9.8404e-01   -1.4741e+01
Acl = Att + Butt*Klqrd;
eig(Acl)
ans =
   -6.9537e-02 + 7.0746e-02i
   -6.9537e-02 - 7.0746e-02i
   -7.3382e-02 + 6.6749e-02i
   -7.3382e-02 - 6.6749e-02i
Y = lyap(Acl, Bwtt * W * Bwtt')
Y =
    3.9016e-03   -1.0842e-19    6.2032e-04    1.7378e-12
   -1.0842e-19    3.8393e-05   -1.7378e-12    6.1042e-06
    6.2032e-04   -1.7378e-12    3.4465e-03    1.2197e-19
    1.7378e-12    6.1042e-06    1.2197e-19    3.3915e-05
sqrt(trace(Y))
ans =
    8.6142e-02
sqrt(diag(Y))
ans =
    6.2463e-02
    6.1962e-03
    5.8707e-02
    5.8236e-03

% LQR solution
Cz = [eye(n); zeros(m,n)];
Dzu = 1e4*[zeros(n,m); eye(m)];

[K,X,S] = lqr(Att, Butt, Cz' * Cz, Dzu' * Dzu);
Klqr = - K
Klqr =
   -1.9615e-05   -6.5712e-03    3.8376e-05   -5.2724e-02
   -1.5950e-04   -5.2724e-02    9.2343e-05   -7.2284e-01

```

```

% LMI solution with decentralization
% declare variables
X = sdpvar(n,n,'symmetric');
Z = sdpvar(m+n,m+n,'symmetric');
L = sdpvar(m,n);

% declare LMIs
LMI1 = Att*X+X*Att'+Butt*L+L'*Butt'+Bwtt*W*Bwtt';
LMI2 = [Z, Cz*X+Dzu*L; X*Cz'+L'*Dzu', X];

% Structural constraints
Xmask = [zeros(2,2) ones(2,2); ones(2,2) zeros(2,2)]
Xmask =
    0     0     1     1
    0     0     1     1
    1     1     0     0
    1     1     0     0
Lmask = [zeros(1,2) ones(1,2); ones(1,2) zeros(1,2)]
Lmask =
    0     0     1     1
    1     1     0     0

LMI = set(LMI1 < 0) + set(LMI2 > 0) + set(Xmask.*X == 0) + set(Lmask.*L == 0);
options = sdpsettings('solver','sedumi');
solution = solvesdp(LMI,trace(Z),options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
Put 24 free variables in a quadratic cone
eqs m = 39, order n = 17, dim = 142, blocks = 4
nnz(A) = 95 + 0, nnz(ADA) = 1521, nnz(L) = 780
it :      b*y      gap   delta  rate  t/tP*  t/tD*   feas cg cg   prec
 0 :              2.29E+00 0.000
 1 :   6.89E-02 4.60E-01 0.000 0.2008 0.9000 0.9000   2.20 1 1 8.0E-01
 2 :   2.73E-04 1.74E-02 0.000 0.0377 0.9900 0.9900   1.36 1 1 2.5E-02
 3 :  -1.31E-05 7.63E-05 0.000 0.0044 0.9990 0.9990   1.02 1 1 1.1E-04
 4 :  -1.69E-04 3.14E-05 0.000 0.4117 0.9000 0.9000   0.14 1 1 1.3E-04
 5 :  -6.45E-04 1.01E-05 0.000 0.3228 0.9000 0.9000  -0.50 1 1 1.2E-04
 6 :  -3.19E-03 2.05E-06 0.000 0.2018 0.9000 0.9000  -0.77 1 1 9.4E-05
 7 :  -1.57E-02 4.54E-07 0.000 0.2219 0.9000 0.9000  -0.97 1 1 9.6E-05
 8 :  -6.61E-02 1.17E-07 0.000 0.2575 0.9000 0.9000  -1.05 1 1 1.0E-04
 9 :  -2.73E-01 2.93E-08 0.000 0.2504 0.9000 0.9000  -1.11 1 1 1.0E-04
10 :  -1.19E+00 7.30E-09 0.000 0.2492 0.9000 0.9000  -1.14 1 1 1.1E-04
11 :  -5.83E+00 1.53E-09 0.000 0.2091 0.9000 0.9000  -1.12 1 1 1.1E-04
12 :  -2.89E+01 3.11E-10 0.000 0.2041 0.9000 0.9000  -1.07 1 1 1.1E-04
13 :  -1.41E+02 6.27E-11 0.000 0.2012 0.9000 0.9000  -1.03 1 1 1.1E-04
14 :  -6.20E+02 1.34E-11 0.000 0.2142 0.9000 0.9000  -0.99 1 1 1.0E-04
15 :  -2.20E+03 3.62E-12 0.000 0.2693 0.9000 0.9000  -0.94 4 4 9.8E-05
16 :  -2.46E+04 3.04E-13 0.000 0.0840 0.9900 0.9900  -0.96 6 6 9.1E-05
17 :  -4.89E+04 1.31E-13 0.000 0.4319 0.9000 0.9000  -0.77 11 10 7.5E-05
18 :  -1.39E+05 5.07E-14 0.000 0.3859 0.9000 0.9000  -0.74 11 11 8.1E-05
19 :  -2.27E+05 2.11E-14 0.000 0.4170 0.9000 0.9000  -0.53 9 9 5.3E-05
20 :  -3.42E+05 5.37E-15 0.000 0.2540 0.9000 0.9000   0.03 20 20 1.9E-05
21 :  -3.76E+05 3.91E-16 0.000 0.0729 0.9900 0.9900   0.82 15 14 1.5E-06
22 :  -3.78E+05 1.32E-16 0.000 0.3377 0.9000 0.9000   0.98 21 21 5.1E-07

```

```

23 : -3.79E+05 3.30E-17 0.000 0.2496 0.9000 0.9000 0.99 21 18 1.3E-07
24 : -3.79E+05 1.46E-18 0.000 0.0442 0.9900 0.9900 1.00 21 21 5.7E-09
25 : -3.79E+05 3.15E-19 0.000 0.2165 0.9000 0.9000 1.00 28 28 1.2E-09
26 : -3.79E+05 1.79E-20 0.303 0.0567 0.9900 0.9900 1.00 27 27 7.0E-11

iter seconds digits c*x b*y
26 0.5 Inf -3.7933106979e+05 -3.7933104329e+05
|Ax-b| = 4.1e-06, [Ay-c]_+ = 9.5E-12, |x|= 3.8e+10, |y|= 2.7e+05

Detailed timing (sec)
Pre IPM Post
0.000E+00 5.300E-01 1.000E-02
Max-norms: ||b||=1, ||c|| = 1.000000e-05,
Cholesky |add|=8, |skip| = 0, ||L.L|| = 500000.
solution =
    yalmiptime: 9.1947e-02
    solvertime: 5.8071e-01
    info: 'No problems detected (SeDuMi-1.1)'
    problem: 0
    dimacs: [2.2056e-06 0 0 9.5198e-12 -3.4933e-08 -3.4928e-08]

% Construct K and check closed loop stability and performance
double(L)
ans =
    -1.0051e+00    -1.9109e-03    3.9831e-16    -8.6545e-13
    -1.9023e-15    -8.7299e-13    -1.9149e-02    -6.5467e-04
double(X)
ans =
    1.2270e+03    -5.0097e-05    -2.3356e-17    -1.7194e-15
    -5.0097e-05    5.1018e-03    -1.4304e-16    -8.4199e-13
    -2.3356e-17    -1.4304e-16    1.8884e+02    -6.5158e-07
    -1.7194e-15    -8.4199e-13    -6.5158e-07    1.9150e-04
Ld = double(L) - double(L) .* Lmask
Ld =
    -1.0051e+00    -1.9109e-03    0    0
    0    0    -1.9149e-02    -6.5467e-04
Xd = double(X) - double(X) .* Xmask
Xd =
    1.2270e+03    -5.0097e-05    0    0
    -5.0097e-05    5.1018e-03    0    0
    0    0    1.8884e+02    -6.5158e-07
    0    0    -6.5158e-07    1.9150e-04
Klmi = Ld / Xd
Klmi =
    -8.1924e-04    -3.7456e-01    0    0
    0    0    -1.0142e-04    -3.4185e+00
Acl = Att + Buttt*Klmi;
eig(Acl)
ans =
    -3.3978e-02
    -1.9615e-03 + 5.7235e-04i
    -1.9615e-03 - 5.7235e-04i
    -2.9726e-05
Y = lyap(Acl, Bwtt * W * Bwtt')
```

```

Y =
    9.9003e+02   -8.1879e-16    4.8595e+01    4.0278e-03
   -8.1879e-16    3.9089e-03   -4.0278e-03    1.8697e-04
    4.8595e+01   -4.0278e-03    1.4407e+02   -4.3368e-19
    4.0278e-03    1.8697e-04   -4.3368e-19    1.4569e-04
sqrt(trace(Y))
ans =
    3.3677e+01
sqrt(diag(Y))
ans =
    3.1465e+01
    6.2522e-02
    1.2003e+01
    1.2070e-02

% Try stripping lqr gain
Klqr
Klqr =
   -1.9615e-05   -6.5712e-03    3.8376e-05   -5.2724e-02
   -1.5950e-04   -5.2724e-02    9.2343e-05   -7.2284e-01
Klqrd = double(Klqr) - double(Klqr) .* Lmask
Klqrd =
   -1.9615e-05   -6.5712e-03         0         0
         0         0    9.2343e-05   -7.2284e-01
Acl = Att + Butt*Klqrd;
eig(Acl)
ans =
   -6.4410e-03
   -2.6162e-03
    1.6344e-03
    1.2869e-04

diary off

```

10.5 Robust Control

The following discussion is one of many possible approaches to this difficult problem.

10.5.1 Robust Stability

Consider the CTLTI uncertain system

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0, \quad x(t) \in \mathbb{R}^n, \quad A \in \mathcal{A}, \quad (18)$$

where $\mathcal{A}$ is an arbitrary *closed convex set*.

Definition: The CTLTI uncertain system (18) is said to be *robustly stable* if (18) is asymptotically stable for all $A \in \mathcal{A}$.

Robust Stability: Let Ω be the set of stable matrices. If $\mathcal{A} \subseteq \Omega$ then the CTLTI uncertain system (18) is robustly stable.

Proposition: The set of all stable matrices Ω is *not* a convex set.

Proof: The set $\Omega := \{X \in \mathbb{R}^{n \times n} : \max_i \operatorname{Re}\{\lambda_i(X)\} < 0\}$ is a non convex cone.

10.5.2 Polyhedral Uncertainty

Let the set of uncertain systems be characterized by

$$\mathcal{A} := \operatorname{co}(A_1, \dots, A_N) = \left\{ A(\xi) : A(\xi) = \sum_{i=1}^N \xi_i A_i, \quad \forall \xi \in \Xi \right\},$$
$$\Xi := \left\{ \xi : \xi = (\xi_1, \dots, \xi_N) \geq 0, \quad \sum_{i=1}^N \xi_i = 1 \right\}.$$

Proposition: Any closed convex set $\mathcal{A}$ can be arbitrarily approximated by a convex polyhedron.

Drawback: Often a very large number of vertices is required.

10.5.3 Quadratic Stability

The CTLTI uncertain system (18) where $\mathcal{A} = \text{co}(A_1, \dots, A_N)$ is *Quadratically Stable* if there exists $P \in \mathbb{S}^n$ such that the LMI

$$P \succ 0, \quad A_i^T P + P A_i \prec 0, \quad \text{for all } i = 1, \dots, N,$$

is feasible.

Proposition: If the CTLTI uncertain system (18) where $\mathcal{A} = \text{co}(A_1, \dots, A_N)$ is quadratically stable then it is also robustly stable.

Proof: For any $\tilde{A} \in \mathcal{A}$ there exists a $\tilde{\xi} \in \Xi$ such that $\tilde{A} = \sum_{i=1}^N \tilde{\xi}_i A_i$. Since $\tilde{\xi} \in \Xi \Rightarrow \tilde{\xi}_i \geq 0$, multiply each inequality in the QS condition by $\tilde{\xi}_i$ to obtain

$$P \succ 0, \quad \left(\tilde{\xi}_i A_i^T \right) P + P \left(\tilde{\xi}_i A_i \right) \prec 0, \quad \text{for all } i = 1, \dots, N.$$

The sum of all the above inequalities provides

$$P \succ 0, \quad \sum_{i=1}^N \left(\tilde{\xi}_i A_i^T \right) P + P \sum_{i=1}^N \left(\tilde{\xi}_i A_i \right) = \tilde{A}^T P + P \tilde{A} \prec 0$$

which implies that $\tilde{A}$ is stable. The proof is completed by noticing that the above holds for any $A \in \mathcal{A}$.

Remarks:

- a) Robust stability requires stability to be checked for all $\xi \in \Xi$. This is equivalent to solve an infinite number of SDP feasibility problems (Lyapunov inequalities) on $P(\xi) \succ 0$. The resulting $V(x, \xi) = x^T P(\xi) x$ is called a *parameter-dependent Lyapunov function*.
- b) *Quadratic stability* requires stability to be checked at the N vertices of $\mathcal{A}$. This is equivalent to solve N SDP feasibility problems (Lyapunov inequalities) on $P \succ 0$. The resulting $V(x, \xi) = V(x) = x^T P x$ does not depend on ξ .
- c) Quadratic stability $\Rightarrow$ robust stability.
- d) Quadratic stability is broader. Consider the time-varying linear system

$$\dot{x}(t) = A(\xi(t))x(t), \quad x(0) = x_0, \quad \xi(t) \in \Xi \text{ for all } t \geq 0.$$

If $\mathcal{A} = \text{co}(A_1, \dots, A_N)$ is quadratically stable then the origin is an asymptotically stable equilibrium. This is because $V(x) = x^T P x$ is a Lyapunov function and

$$\begin{aligned} \dot{V}(x(t)) &= 2x(t)^T P \dot{x}(t) \\ &= x(t)^T [A(\xi(t))^T P + P A(\xi(t))] x(t) < 0 \end{aligned}$$

for all $x(t) \neq 0$. Note that $\dot{V}(x(t))$ does not depend on the $\dot{\xi}(t)$ so that ξ can vary *arbitrarily fast*!

10.5.4 Example

Find $\underline{\theta} < \theta < \bar{\theta}$ such that the CTLTI system

$$H(s) = \frac{N_H(s)}{D_H(s)} = \frac{2s^3 - s^2 + 1}{s^4 + s^3 + 25s^2 + 3(4 + \theta)s + 3(4 + \theta)}$$

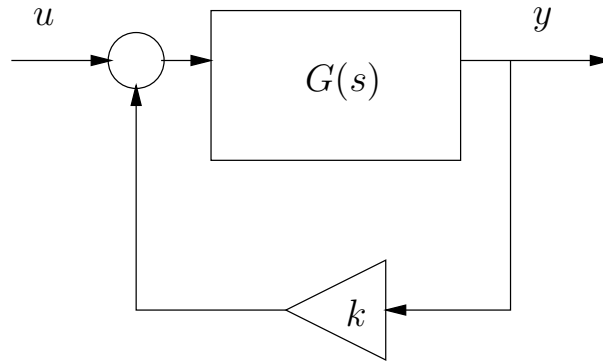
is robustly stable. In state-space

$$\dot{x} = A(\theta)x, \quad A(\theta) \in \text{co} \left(A(\underline{\theta}), A(\bar{\theta}) \right),$$

$$A(\theta) := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -3(4 + \theta) & -3(4 + \theta) & -25 & -1 \end{bmatrix}$$

Results:

Stability	$\underline{\theta}$	$\bar{\theta}$
Robust	-4	4
Quadratic	-1.8	2.2



A feedback control connection:

$$\frac{Y(s)}{U(s)} = \frac{G(s)}{1 + kG(s)} = \frac{N_G(s)}{D_G(s) + kN_G(s)}$$

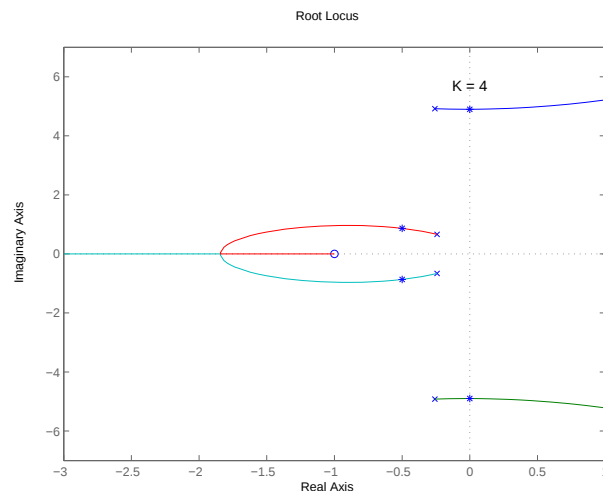
Hence, an appropriate $G(s)$ for stability analysis can be computed from $D_G(s) + kN_G(s) = D_H(s)$, that is

$$\underbrace{(s^4 + s^3 + 25s^2 + 12s + 12)}_{D_G(s)} + k \underbrace{3(s + 1)}_{N_G(s)} = \underbrace{s^4 + s^3 + 25s^2 + 3(4 + k)s + 3(4 + k)}_{D_H(s)}.$$

Set

$$G(s) = \frac{N_G(s)}{D_G(s)} = \frac{3(s + 1)}{s^4 + s^3 + 25s^2 + 12s + 12}$$

and, from root locus plot, it can be shown that the CTLTI system is (robustly) stable for $-4 < \theta < 4$. Alternatively, one could also use the Routh-Hurwitz criterion directly on $D_H(s)$.



10.5.5 Robust LQR Control

Problem: Given the uncertain LTI system

$$\begin{aligned}\dot{x}(t) &= A(\xi)x(t) + B_u(\xi)u(t) + B_w w(t), & x(0) &= 0, & x &\in \mathbb{R}^n, \\ z(t) &= C_z x(t) + D_{zu}u(t).\end{aligned}$$

where $\begin{bmatrix} A(\xi) & B_u(\xi) \end{bmatrix} \in \mathcal{M}$,

$$\begin{aligned}\mathcal{M} &:= \text{co}(M_1, \dots, M_N) = \left\{ M(\xi) : M(\xi) = \sum_{i=1}^N \xi_i M_i, \quad \forall \xi \in \Xi \right\}, \\ \Xi &:= \left\{ \xi : \xi = (\xi_1, \dots, \xi_N) \geq 0, \quad \sum_{i=1}^N \xi_i = 1 \right\}.\end{aligned}$$

compute a state feedback controller

$$u(t) = K x(t),$$

that stabilizes the closed loop system and minimizes

$$J := \lim_{t \rightarrow \infty} E \left[z(t)^T z(t) \right]$$

Assumptions:

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

Solution: We do not know of a complete solution to this problem. Here we will be happy with a sufficient design (conservative) using quadratic stability.

Using the congruence+change-of-variables the LQR problem can be recast as the SDP

$$\begin{aligned} \min_{X,L,Z} \quad & \text{trace}(ZW) \\ \text{s.t.} \quad & \begin{bmatrix} AX + XA^T + B_u L + L^T B_u^T & XC_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0, \\ & \begin{bmatrix} Z & B_w^T \\ B_w & X \end{bmatrix} \succeq 0, \quad X \succ 0. \end{aligned}$$

where $K = LX^{-1}$. A natural extension using quadratic stability is that

$$\begin{aligned} \min_{X,L,Z} \quad & \text{trace}(ZW) \\ \text{s.t.} \quad & \begin{bmatrix} A_i X + XA_i^T + B_{u_i} L + L^T B_{u_i}^T & XC_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0, \quad i = 1, \dots, N, \\ & \begin{bmatrix} Z & B_w^T \\ B_w & X \end{bmatrix} \succeq 0, \quad X \succ 0. \end{aligned}$$

Multiplying each inequality by $\xi_i \geq 0$ and summing we obtain

$$\begin{bmatrix} A(\xi)X + XA(\xi)^T + B_u(\xi)L + L^T B_u(\xi)^T & XC_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0$$

for all $\xi \in \Xi$. Using the Comparison Lemma $X \succeq P(\xi)$ where $P(\xi)$ is the parameter dependent observability Gramian computed with the controller $K = LX^{-1}$ so that

$$\text{trace}(ZW) \geq J = \lim_{t \rightarrow \infty} E[z(t)^T z(t)]$$

The dual form follows the same pattern.

10.5.6 Summary: Robust LQR Control

The state feedback controller

$$u(t) = Kx(t), \quad K = LX^{-1}$$

where $X \in \mathbb{S}^n$, $L \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{S}^r$ solve the SDP

$$\begin{aligned} \min_{X, L, Z} \quad & \text{trace}(ZW) \\ \text{s.t.} \quad & \begin{bmatrix} A_i X + X A_i^T + B_{u_i} L + L^T B_{u_i}^T & X C_z^T + L^T D_{zu}^T \\ C_z X + D_{zu} L & -I \end{bmatrix} \preceq 0, \quad i = 1, \dots, N, \\ & \begin{bmatrix} Z & B_w^T \\ B_w & X \end{bmatrix} \succeq 0, \quad X \succ 0, \end{aligned}$$

is a suboptimal solution to the problem of stabilizing the closed loop system and minimizing the cost function

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)] \leq \text{trace}(ZW)$$

for the uncertain LTI system

$$\begin{aligned} \dot{x}(t) &= A(\xi)x(t) + B_u(\xi)u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t) \end{aligned}$$

where $[A(\xi) \ B_u(\xi)] \in \mathcal{M}$,

$$\begin{aligned} \mathcal{M} &:= \text{co}(M_1, \dots, M_N) = \left\{ M(\xi) : M(\xi) = \sum_{i=1}^N \xi_i M_i, \quad \forall \xi \in \Xi \right\}, \\ \Xi &:= \left\{ \xi : \xi = (\xi_1, \dots, \xi_N) \geq 0, \quad \sum_{i=1}^N \xi_i = 1 \right\}, \end{aligned}$$

under the assumptions

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.

10.5.7 Summary: Robust LQR Control (dual form)

The state feedback controller

$$u(t) = Kx(t), \quad K = LX^{-1}$$

where $X \in \mathbb{S}^n$, $L \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{S}^r$ solve the SDP

$$\begin{aligned} \min_{X, L, Z} \quad & \text{trace}(Z) \\ \text{s.t.} \quad & A_i X + X A_i^T + B_{u_i} L + L^T B_{u_i}^T + B_w W B_w^T \preceq 0, \quad i = 1, \dots, N, \\ & \begin{bmatrix} Z & C_z X + D_{zu} L \\ X C_z^T + L^T D_{zu}^T & X \end{bmatrix} \succeq 0, \quad X \succ 0, \end{aligned}$$

is a suboptimal solution to the problem of stabilizing the closed loop system and minimizing the cost function

$$J := \lim_{t \rightarrow \infty} E [z(t)^T z(t)] \leq \text{trace}(Z)$$

for the uncertain LTI system

$$\begin{aligned} \dot{x}(t) &= A(\xi)x(t) + B_u(\xi)u(t) + B_w w(t), \quad x(0) = 0 \\ z(t) &= C_z x(t) + D_{zu} u(t) \end{aligned}$$

where $\begin{bmatrix} A(\xi) & B_u(\xi) \end{bmatrix} \in \mathcal{M}$,

$$\begin{aligned} \mathcal{M} &:= \text{co}(M_1, \dots, M_N) = \left\{ M(\xi) : M(\xi) = \sum_{i=1}^N \xi_i M_i, \quad \forall \xi \in \Xi \right\}, \\ \Xi &:= \left\{ \xi : \xi = (\xi_1, \dots, \xi_N) \geq 0, \quad \sum_{i=1}^N \xi_i = 1 \right\}, \end{aligned}$$

under the assumptions

- a) $D_{zu}^T D_{zu} \succ 0$;
- b) $w(t)$ is a Gaussian white noise vector with zero mean and covariance $W \succ 0$.