ROBUST DESIGN WITH
LINEAR MATRIX INEQUALITIES
AND POLYNOMIAL MATRICES

Didier HENRION

www.laas.fr/∼henrion

henrion@laas.fr

Škoda Octavia produced in Mladá Boleslav

October-November 2001

From analysis to design

In the previous course we have seen that strict positive realness of rational matrices and positivity of polynomial matrices can be combined to assess robust stability of polynomial matrices and state-space systems in a unified way

Just recall the main result
(in a slightly modified form):

Polynomial matrix $C(s)$ is stable iff there exists a stable polynomial matrix $D(s)$ and a matrix $P = P^\star$ satisfying the LMI

$$D^\star C + C^\star D - S(P) > 0$$

Observe the decoupling between system matrix $C(s)$ and LMI unknown matrix P

Linearity hence convexity is ensured as soon as a stable polynomial matrix $D(s)$ is given

Robust design

Assume now that system matrix $C(s, \lambda)$ is obtained from a polynomial matrix Diophantine equation

$$C(s, \lambda) = A(s, \lambda)X(s) + B(s, \lambda)Y(s)$$

where system matrices A and B are subject to multi-linear polytopic uncertainty λ

In order to ensure robust SPRness of the rational matrix $D^{-1}(s)C(s, \lambda)$ polynomial matrix $D(s)$ must be close to the nominal closed-loop matrix, such as

$$D(s) = C(s, \lambda_0)$$

where λ_0 is the nominal parameter vector

A sensible simple choice of $D(s)$ is therefore the nominal closed-loop denominator polynomial matrix, obtained by any standard design method (pole assignment, LQ, H_∞)

Reactor

Consider the stirred tank reactor model described by non-linear state-space equations

$$\xi_1 = (\xi_2 + 0.5)\exp(E\xi_1/(\xi_1 + 2)) - (2 + u)(\xi_1 + 0.25)$$
$$\xi_2 = 0.5 - \xi_2 - (\xi_2 + 0.5)\exp(E\xi_1/(\xi_1 + 2))$$

where E is a parameter related to the activation energy

During the life of the reactor, some representative values of E are 20, 25 and 30

Using only ξ_1 for feedback we obtain a polytopic system with 3 linearized vertex transfer functions

$$b_1(s)/a_1(s) = (0.5 - 0.25s)/(11 - 5s + s^2)$$
$$b_2(s)/a_2(s) = (-0.5 - 0.25s)/(-2.25 - 2.25s + s^2)$$
$$b_3(s)/a_3(s) = (-0.5 - 0.25s)/(-3.5 - 3.5s + s^2).$$

Our objective is to stabilize the whole polytopic system with a single static output feedback controller $y(s)/x(s)$

Reactor

For the choice $d(s) = (s + 1)^2$ as a parameter denom-inator polynomial we solve the LMI in 0.30 seconds of CPU time to obtain the robustly stabilizing controller

$$y(s)/x(s) = k = -21.4409$$

Using Hermite matrices and the eigenvalue criterion (see course on simultaneous stabilization) we obtain indeed that there exists a static output feedback controller simultaneously stabilizing the three plants iff
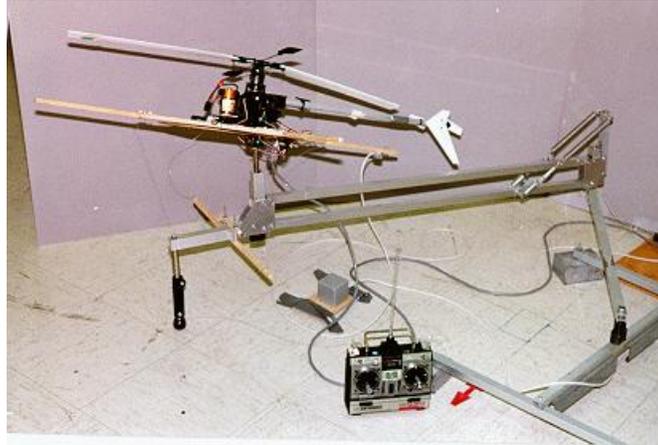
$$-22 < k < -20$$

Note however that the problem solved here is more dif-ficult since we must stabilize the whole polytope, not only its vertices
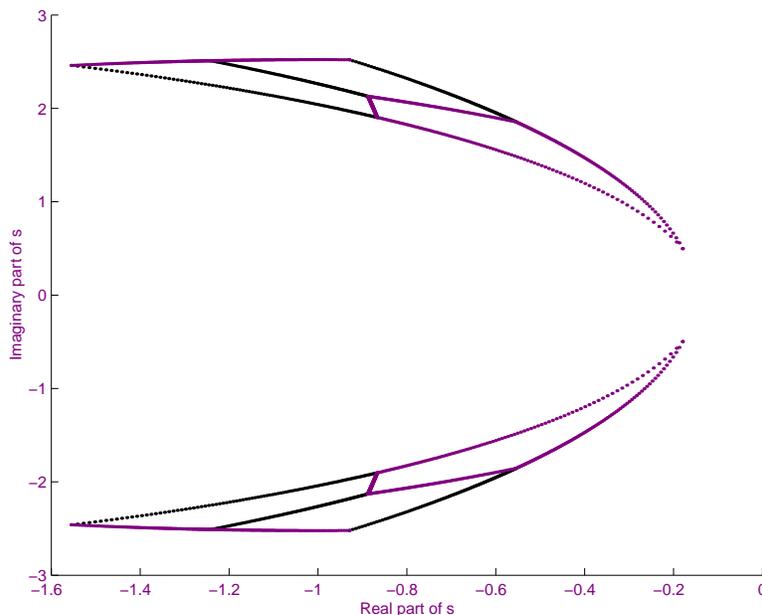
Edges of root locus

Helicopter

Consider a helicopter toy used in a university lab

whose simplified linearized model is given by the interval transfer function

$$\frac{[20,\ 60]}{s(0.1s+1)(s+[0,\ 1])}$$

We are seeking a second-order controller $y(s)/x(s)$ robustly stabilizing the interval plant

Using some design method we obtain a nominally stabilizing controller (for the central parameters 40 and 0.5)

$$\frac{y^0}{x^0} = \frac{2 + 2.2s + 2.2s^2}{10s + s^2}$$

yielding the nominal characteristic polynomial
$d = a(q^0)x^0 + b(q^0)y^0 = 80 + 88s + 93s^2 + 11s^3 + 2.05s^4 + 0.10s^5$
After 5 seconds of CPU time the LMI solver returns the robustly stabilizing controller

$$\frac{y}{x} = \frac{0.9876 + 0.3045s + 1.5747s^2}{1.6136 + 6.2396s + s^2}$$

Helicopter

In our experiments we attempt to minimize the Euclidean norm of the vector of controller coefficients (this is still an LMI problem, see previous course)
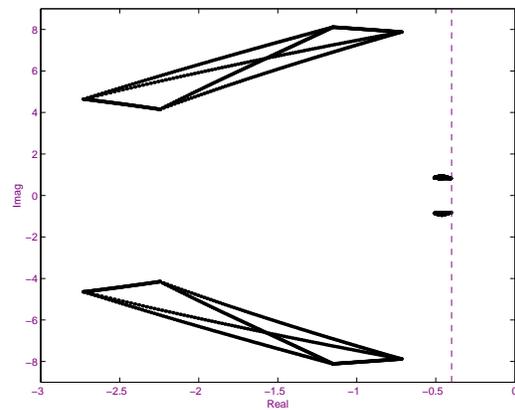
We obtain a controller of norm 6.79, but from the root locus edges (see bottom left) stability is ensured only marginally, so we shift the stability region to the left

$$\mathcal{S} = \{s : \operatorname{Re} s < -0.4\} \quad S = \begin{bmatrix} 0.8 & 1 \\ 1 & 0 \end{bmatrix}$$

Solving the new LMI problem with the stability margin, we obtain a controller of norm 11.9 greater than the previous one, since more effort is needed to shift the poles farther from the imaginary axis (see bottom right)
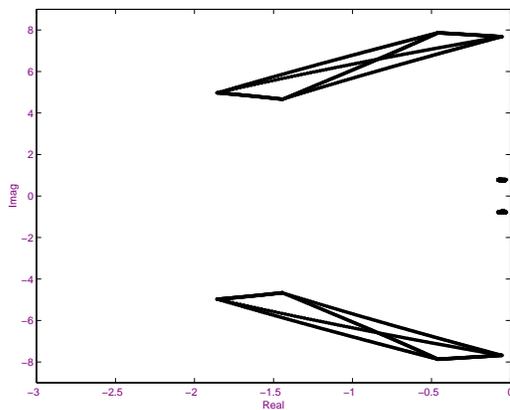
Without stability margin   With stability margin

Edges of robust root locii

Oblique wing aircraft

We consider the model of an experimental oblique wing aircraft

The linearized transfer function

$$\frac{[90,\ 166] + [54,\ 74]s}{[-0.1,\ 0.1] + [30.1,\ 33.9]s + [50.4,\ 80.8]s^2 + [2.8,\ 4.6]s^3 + s^4}$$

features 6 uncertain parameters, and must be stabilized with a PI controller

$$\frac{y(s)}{x(s)} = K_p + \frac{K_i}{s}$$

One can check easily that the choice $K_p = 1$ and $K_i = 1$ stabilizes the vertex plant

$$\frac{90 + 54s}{-0.1 + 30s + 50s^2 + 2.8s^3 + s^4}$$

Oblique wing aircraft

With the choice $d(s) = sa_1(s) + (1 + s)b_1(s)$ as the central polynomial, the LMI problem with additional structural (LMI) constraints on the controller coefficients was found infeasible

It does not necessarily mean that the system is not robustly PI stabilizable, it may happen that the sufficient LMI robust stability condition is too conservative

Indeed, when considering another vertex plant

$$\frac{b_2(s)}{a_2(s)} = \frac{90 + 54s}{-0.1 + 30s + 50s^2 + 4.6s^3 + s^4}$$

the choice $d(s) = sa_2(s) + (1 + s)b_2(s)$ now proves successful, the LMI is solved after about 5 secs and we obtain the robustly stabilizing PI controller

$$\frac{y(s)}{x(s)} = 0.8634 + \frac{0.6454}{s}$$
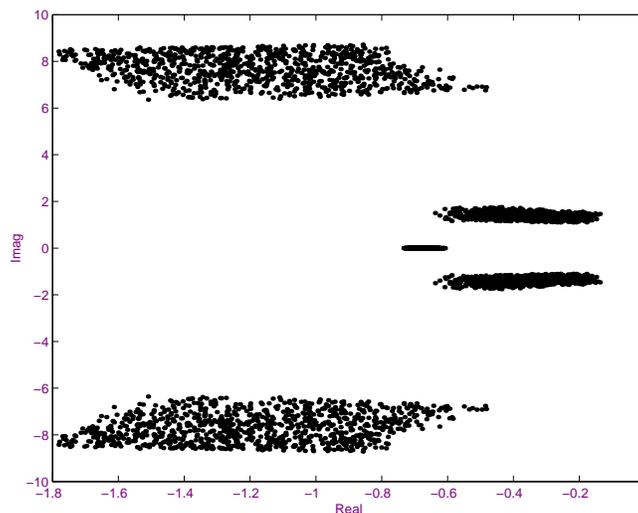
Satellite

We consider a satellite control problem where the satellite model is two masses with the same inertia connected by a spring with torque constant q_1 and viscous damping constant q_2

The transfer function between the control torque and the satellite angle is given by

$$\frac{b(s,q)}{a(s,q)} = \frac{q_1 + q_2 s + s^2}{s^2(2q_1 + 2q_2 s + s^2)}$$

where uncertain parameters are assumed to vary within the bounds

$$q_1 \in [0.09, 4], \ q_2 \in [0.04\sqrt{q_1}10, 0.2\sqrt{q_2}10]$$

With the choice $d(s) = (s+1)^6$ as a central polynomial, the LMI design method cannot stabilize the whole polytope

Satellite

However we can stabilize each vertex $a_i(s)$, $b_i(s)$ individually with controller polynomials $x_i(s)$, $y_i(s)$

The roots of the corresponding closed-loop characteristic polynomials $c_i(s) = a_i(s)x_i(s) + b_i(s)y_i(s)$ are given below

i	roots of $c_i(s)$
1	$-0.4520 \pm j1.8129,\ -0.0365 \pm j0.8954,\ -0.0019 \pm j0.2533$
2	$-0.6161 \pm j1.3829,\ -0.2190 \pm j0.6745,\ -0.0237 \pm j0.2136$
3	$-1.0484,\ -0.0917 \pm j2.1598,\ -0.0425 \pm j0.5019,\ -0.0004$
4	$-0.2220 \pm j2.0695,\ -0.1695 \pm j0.5354,\ -0.0709 \pm j0.0045$

The third vertex has poles nearest to the imaginary axis, and with the choice of $d(s) = c_3(s)$ as a central polynomial, the LMI is found feasible after slightly more than 1 second of CPU time and returns the robustly stabilizing controller

$$\frac{y(s)}{x(s)} = \frac{0.0181 + 0.0170s - 1.4048s^2}{3.6082 + 1.0237s + s^2}$$

Robot

We consider the problem of designing a robust controller for the approximate ARMAX model of a PUMA robotic disk grinding process

From the results of identification and because of the nonlinearity of the robot, the coefficients of the numerator of the plant transfer function change for different positions of the robot arm. We consider variations of up to 20% around the nominal value of the parameters

The fourth-order discrete-time model is given by

$$\frac{b(z^{-1}, q)}{a(z^{-1}, q)} = \frac{\left(\begin{array}{c} (0.0257 + q_1) + (-0.0764 + q_2)z^{-1} \\ +(-0.1619 + q_3)z^{-2} + (-0.1688 + q_4)z^{-3} \end{array} \right)}{1 - 1.914z^{-1} + 1.779z^{-2} - 1.0265z^{-3} + 0.2508z^{-4}}$$

where

$$|q_1| \leq 0.00514, |q_2| \leq 0.01528, |q_3| \leq 0.03238, |q_4| \leq 0.03376$$

Robot

The characteristic polynomial of the closed-loop system is given by

$$d(z,q) = z^{12}[(1-z^{-1})a(z^{-1},q)x(z^{-1})+z^{-5}b(z^{-1},q)y(z^{-1})]$$

where the term $1 - z^{-1}$ is introduced in the controller denominator to maintain the steady state error to zero when parameters are changed

With the input central polynomial $d(z) = z^{19}$ the LMI is solved after about 14 seconds of CPU time and returns the seventh-order robust controller

$$\frac{y(z^{-1})}{x(z^{-1})} = \frac{\left(\begin{array}{c} -0.2863 + 0.2928z^{-1} + 0.0221z^{-2} \\ -0.1558z^{-3} + 0.0809z^{-4} + 0.1420z^{-5} \\ -0.1254z^{-6} + 0.0281z^{-7} \end{array}\right)}{\left(\begin{array}{c} 1 + 1.1590z^{-1} + 0.9428z^{-2} \\ +0.4996z^{-3} + 0.3044z^{-4} + 0.4881z^{-5} \\ +0.4003z^{-6} + 0.3660z^{-7} \end{array}\right)}$$
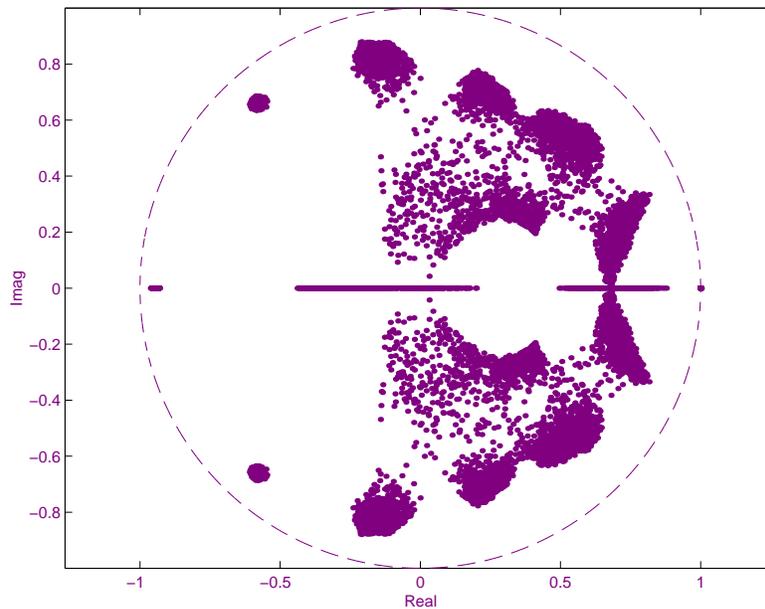
Stability margin optimization

Finally we show how the stability margin can be maximized by proper choice of the central polynomial

We consider Doyle's problem of robustly stabilizing the plant

$$\frac{b(s,q)}{a(s,q)} = \frac{q(s-1)}{(s+1)(s-2)}$$

for all real gains q in the interval $[1, k_1]$

It is known that a robustly stabilizing controller (of arbitrarily high order) exists if and only if $k_1 < 4$

From the Hurwitz stability criterion there is no static controller stabilizing the plant, so we try the central polynomial $d(s) = (s+1)^p$ with $p \geq 3$ in the LMI in order to seek a controller of order $p - 2 \geq 1$

When $p = 3$ we represent in the figure below the minimized Euclidean norm of the first order controller obtained by as a function of k_1
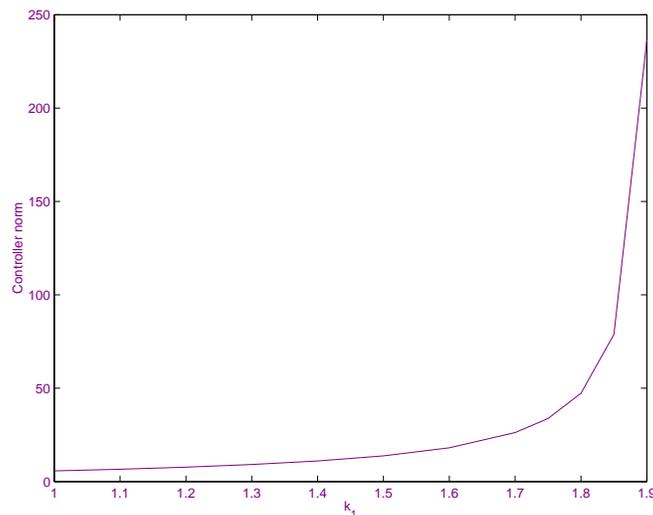
Stability margin optimization

In the figure below we reported as a function of degree p the maximum value of k_1 for which a robust controller is found with the LMI

For values of p greater than 15 we are reaching the limits of the SeDuMi solver, and the function fails for numerical reasons
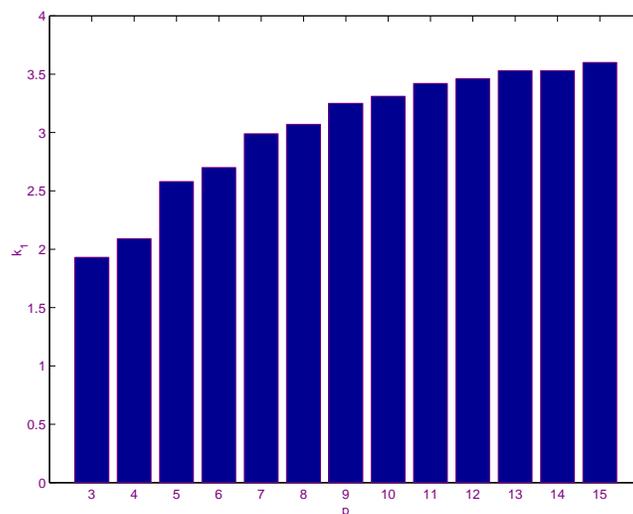
In the sequel we describe a heuristic to improve the stability margin with low-order controllers

Let $k_1 = 2$. We know from the previous results that $ds) = (s+1)^3$ is not a suitable choice of central polynomial for this value of k_1. It may mean that the poles of central polynomial $d(s)$ are not correctly chosen

So we try to move one pole nearest to the imaginary axis, just as in $d(s) = (s+1)^2(s+0.1)$ but the LMI is not feasible…

Stability margin optimization

We try the opposite direction, i.e. $d(s) = (s+1)^2(s+10)$ and the LMI now successfully returns a robustly stabilizing first-order controller

$$\frac{y(s)}{x(s)} = \frac{254.9 + 348.1s}{-327.9 + s}$$

With this choice of central polynomial $d(s)$ the LMI finds a first-order controller up to $k_1 = 2.38$, so we have improved the stability margin without increasing the controller order

Proceeding similarly with higher order controllers, we have been able to stabilize robustly the plant for $k_1 = 3.5$ with the third-order controller

$$\frac{y(s)}{x(s)} = \frac{240.8 + 755.5s + 871.7s^2 + 420.1s^3}{-423.5 - 739.8s - 409.5s^2 + s^3}$$

with the choice $d(s) = (s + 0.5)^3(s + 10)(s + 100)$ as a central polynomial

Doyle with Nevanlinna-Pick interpolation obtained a eighth-order controller

Generally speaking, we may think about designing a heuristic to select poles of $d(s)$ and improve the stability margin without increasing the controller order

Numerical aspects and open problems

The LMI problems arising from SPRness have a special Toeplitz structure that can be exploited to design faster algorithms

Numerical stability must be studied in further details because it is known that problems formulated in the standard polynomial basis

$$\begin{bmatrix} 1 & s & \cdots & s^n \end{bmatrix}$$

generally feature poor conditioning. Alternative polynomial bases such as Chebyshev or Bernstein polynomials may prove useful

The choice of a nominal polynomial $d(s)$ is not always the good one (we found counterexamples) so it would be good to have a systematic way of determining which choice is the best one

Relations with the infinite-dimensional Rantzer-Megretski parametrization of robustly stabilizing controllers for rank-one uncertainty must be clarified

Robustness is not the only objective, other specifications must be considered, taking into account inherent performance limitations, see Karl Åström's web page
www.control.lth.se/∼kja