SOME CONTROL ENGINEERING APPLICATIONS OF STRUCTURED MATRICES

Didier HENRION LAAS-CNRS, FR & FEL-ČVUT, CZ

joint work with

Juan-Carlos ZÚÑIGA Univ. Guadalajara, MX
Jean-Bernard LASSERRE LAAS-CNRS, FR & IMT, FR

LIMOGES, Jan. 2006

OUTLINE

1. POLYNOMIAL MATRIX EQUATIONS

   1.1. pole placement control (BLOCK)

   1.2. null-space extraction TOEPLITZ

2. POLYNOMIAL OPTIMIZATION (PSEUDO) HANKEL

POLE PLACEMENT CONTROL

Linear dynamical systems $\equiv$ time invariant linear ODE

$$a_0 z(t) + a_1 \frac{dz}{dt}(t) + \dots + a_n \frac{d^n z}{dt^n}(t) = b_0 u(t) + \dots + b_n \frac{d^n u}{dt^n}(t)$$

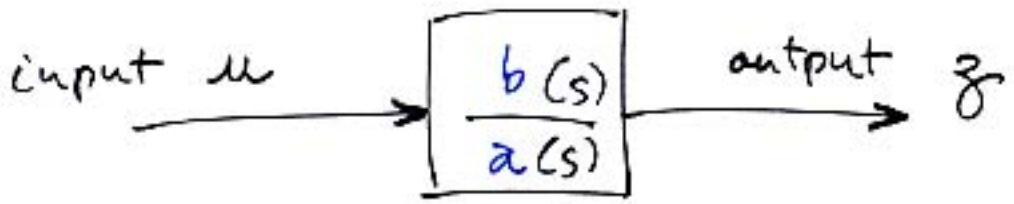
$\downarrow$ Laplace transform (IC = 0)

$$\frac{z(s)}{u(s)} = \frac{b(s)}{a(s)} = g(s)$$

system transfer function

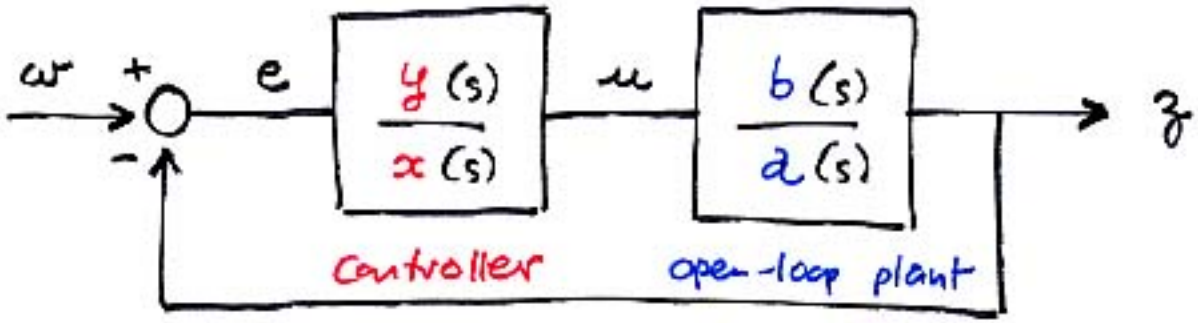
$$\begin{cases} b(s) = b_0 + \dots b_m s^m \\ a(s) = a_0 + \dots a_n s^n \end{cases}$$

polynomials in Laplace variable $s \in \mathbb{C}$



open-loop system

POLE PLACEMENT CONTROL



closed-loop system transfer function

$$z = \frac{b}{a} u = \frac{b}{a}\frac{y}{x} e = \frac{b}{a}\frac{y}{x}(w - z)$$

$$\left(1 + \frac{by}{ax}\right) z = \frac{by}{ax} w \qquad \frac{z}{u} = \frac{by}{ax + by}$$

system dynamics (partly) captured by poles = roots of denom.

$$\boxed{a(s)\, x(s) + b(s)\, y(s) = c(s)}$$

Polynomial Diophantine equation in x, y

DIOPHANTINE EQUATION AND SYLVESTER MATRIX

Identify increasing powers of s^k in $a(s)\,x(s) + b(s)\,y(s) = c(s)$

$a(s) = \sum_k a_k s^k$ $\quad x(s) = \sum_k x_k s^k$ $\quad \ldots$

$$\begin{bmatrix} a_0 & & & b_0 & & \\ a_1 & a_0 & & b_1 & b_0 & \\ a_2 & a_1 & a_0 & b_2 & b_1 & b_0 \\ \vdots & a_2 & a_1 \;\ddots & \vdots & b_2 & b_1 \;\ddots \\ & \vdots & a_2 & & \vdots & b_2 \\ & & \vdots & & & \vdots \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ \\ \\ \\ \end{bmatrix}$$

Sylvester matrix with block-Toeplitz structure

square when $\deg x = \deg y = \deg a - 1 = \deg b - 1$

non-singular iff determinant = resultant non-zero
iff a, b have no common root

NUMERICAL ANALYSIS FOR DIOPHANTINE EQUATION

Conditioning of Sylvester matrix ≡ distance to singularity
≡ distance of pair a, b to nearest pair with common root

Typically in control problems Sylvester matrix is poorly conditioned – use of scaling / pre-conditioning ?

Conditioning depends on polynomial basis

$$\begin{bmatrix} a(s) \\ b(s) \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & .. \\ b_0 & b_1 & b_2 & .. \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \\ : \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & .. \\ b_0 & b_1 & b_2 & .. \end{bmatrix} P \begin{bmatrix} p_0(s) \\ p_1(s) \\ p_2(s) \\ : \end{bmatrix}$$

used to represent a, b

and x, y

?

IMPROVING CONDITIONING - EXAMPLE

Replace s with ρs for some freq. scaling $\rho \in \mathbb{R}_+$

$$[1 \quad s \quad s^2 \quad \dots] \rightarrow [1 \quad \rho s \quad \rho^2 s^2 \quad \dots]$$

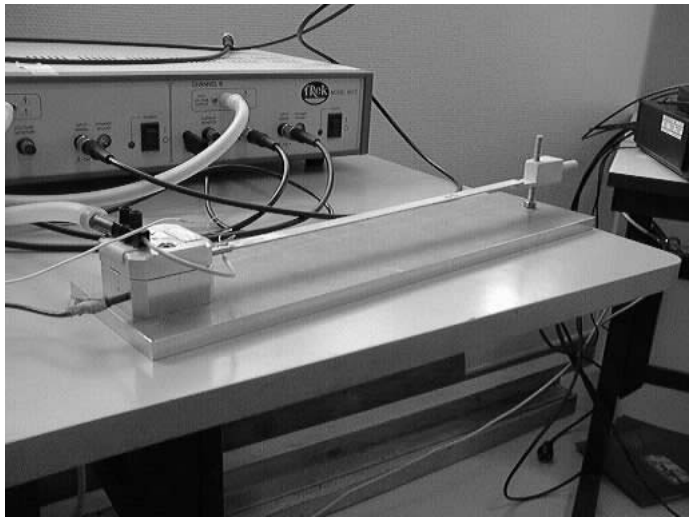
Similarity transform

$$\begin{bmatrix} 1 & & & \\ & \rho & & \\ & & \rho^2 & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} a_0 & & b_0 & \\ a_1 & a_0 & b_1 & b_0 \\ a_2 & a_1 & b_2 & b_1 \\ \vdots & a_2 & \vdots & b_2 \\ & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} 1 & & & \\ & \rho^{-1} & & \\ & & \rho^{-2} & \\ & & & \ddots \end{bmatrix} = S(\rho)$$

Find ρ solving $\min_{\rho \in \mathbb{R}_+} \text{cond}\, S(\rho)$

Is it a convex problem ?

FLEXIBLE BEAM



Experimental setup at ENS Cachan

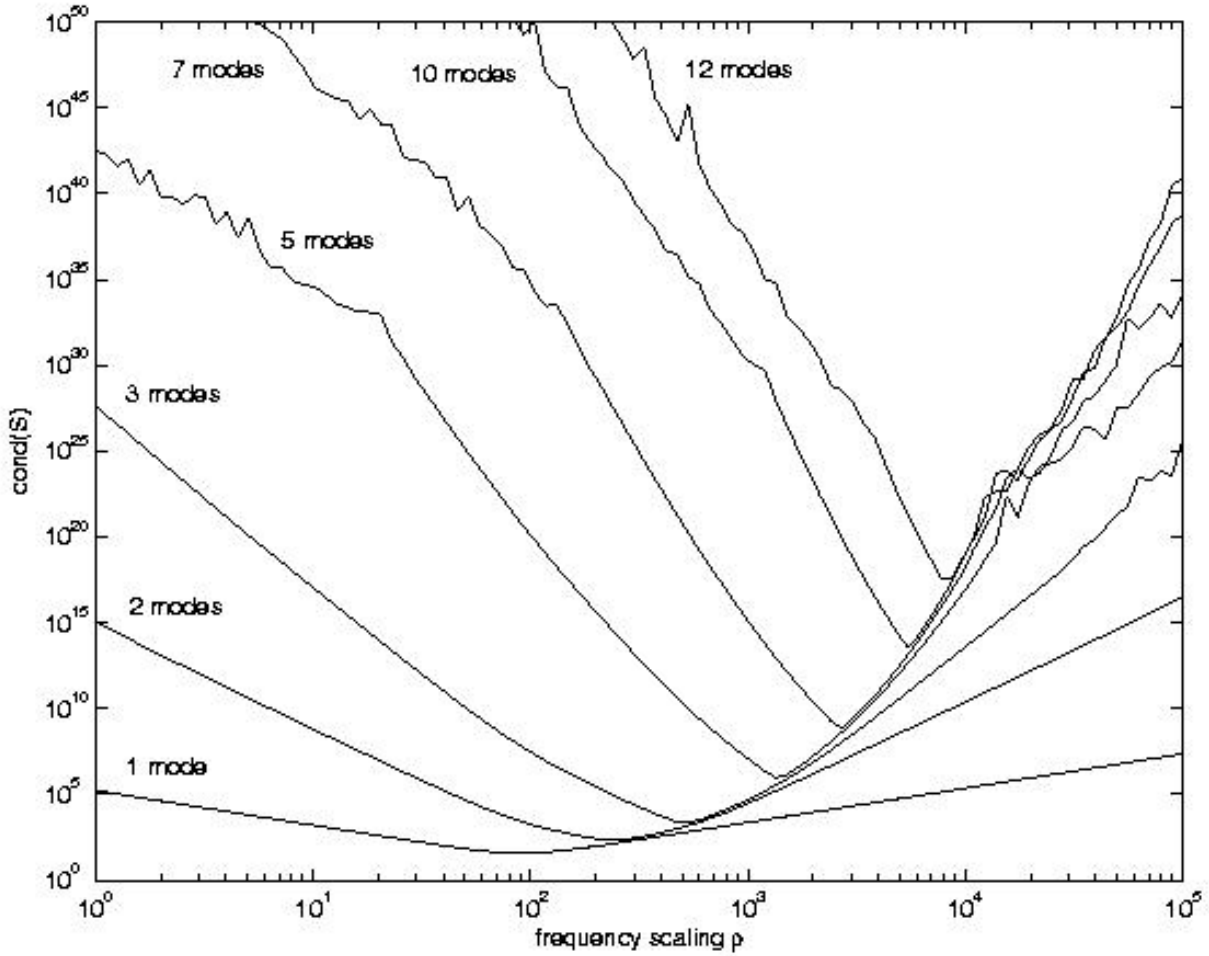
Joint work with Sami TLIBA
and Christophe PRIEUR

Linear model derived from PDE

$$\frac{b(s)}{a(s)} = k \frac{\prod_{i=2}^{N} (s^2 + 2\hat{\xi}_i \hat{\omega}_i + \hat{\omega}_i^2)}{\prod_{i=1}^{N} (s^2 + 2\xi_i \omega_i + \omega_i^2)}$$

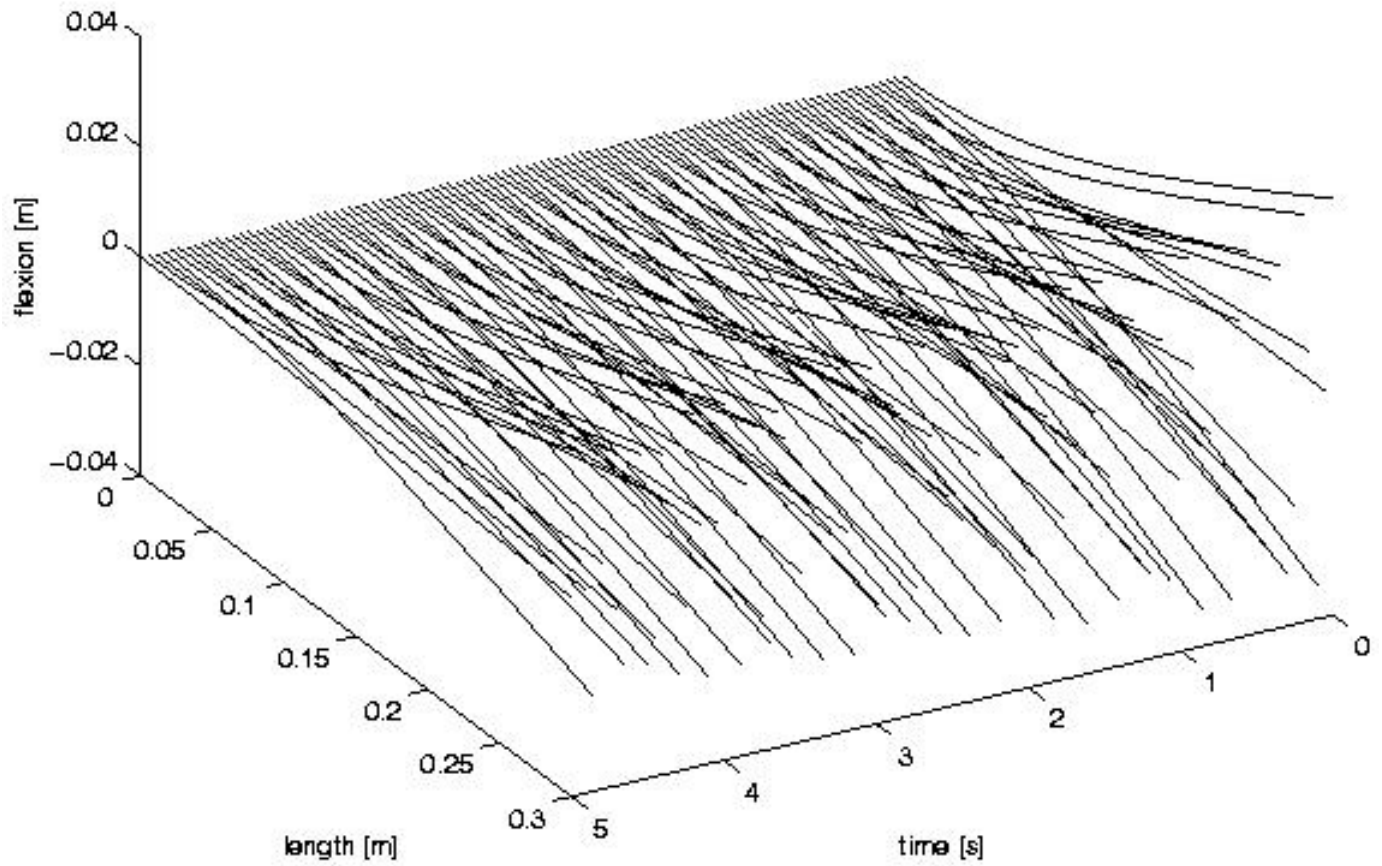
N flexible modes

FLEXIBLE BEAM



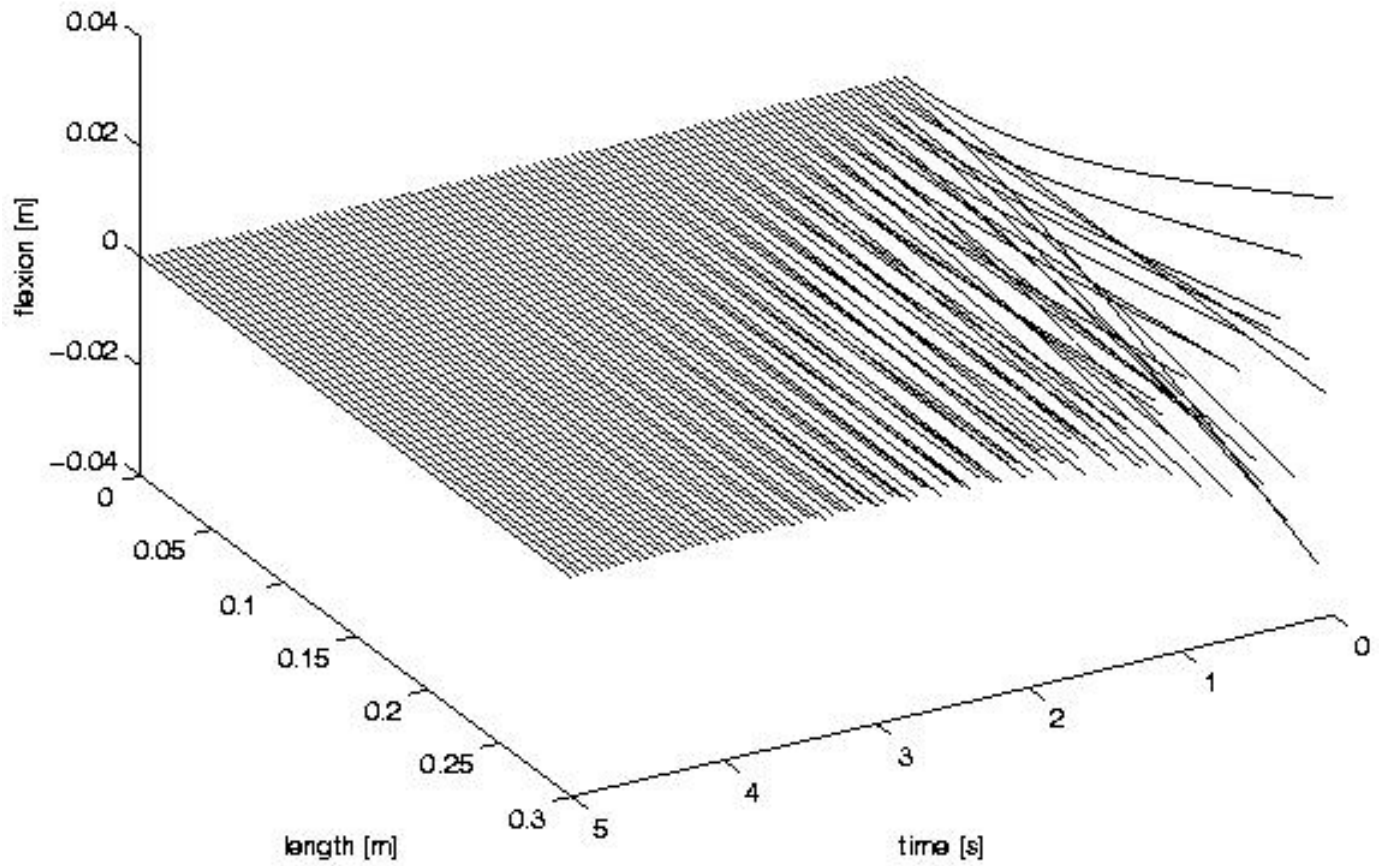
Scaling significantly improves conditioning of pole placement mapping

FLEXIBLE BEAM



Open-loop time profile: response to a first-order deformation mode drop

FLEXIBLE BEAM



Closed-loop time profile: response to a first-order deformation mode drop

OUTLINE

1. POLYNOMIAL MATRIX EQUATIONS

   1.1. pole placement control (BLOCK)

   1.2. null-space extraction TOEPLITZ

2. POLYNOMIAL OPTIMIZATION (PSEUDO) HANKEL

POLYNOMIAL NULL-SPACE EXTRACTION

Given a polynomial matrix $A(s) \in \mathbb{R}^{m \times n}[s]$

we want to compute a polynomial matrix $N(s) \in \mathbb{R}^{n \times p}[s]$ s.t.

$$\boxed{A(s)\,N(s) = 0}$$

Assumption : rank $A(s) = m$ (otherwise ill-posed)

Requirement : $N(s)$ spans basis for null-space, i.e.

rank $N(s) = p = n - m$ and $\sum_k$ col $\deg_k N(s)$ minimum

Applications : linear system modeling

parametrization of all stabilizing controllers

APPLICATIONS OF NULL-SPACE EXTRACTION

From state-space to polynomial matrix fraction description

$$C(sI - A)^{-1}B + D = Y(s)X^{-1}(s)$$

first compute $\bar{X}(s), \bar{Y}(s)$ s.t. $(sI - A)^{-1}B = \bar{Y}(s)\bar{X}^{-1}(s)$ $\quad \begin{bmatrix} sI - A & -B \end{bmatrix} \begin{bmatrix} \bar{Y}(s) \\ \bar{X}(s) \end{bmatrix} = 0$

then $C(sI - A)^{-1}B = C\bar{Y}(s)\bar{X}^{-1}(s)$ and $C(sI-A)^{-1}B + D = (C\underbrace{\bar{Y}(s)}_{Y(s)} + D\bar{X}(s))\underbrace{\bar{X}^{-1}(s)}_{X(s)}$

Parametrization of all stabilizing controllers via null-space of Diophantine equation

$$\begin{bmatrix} A(s) & B(s) \end{bmatrix} \begin{bmatrix} N_A(s) \\ N_B(s) \end{bmatrix} = 0$$

BLOCK TOEPLITZ NULL-SPACE EXTRACTION

Polynomial matrix equation $A(s)\,N(s) = (A_0 + A_1 s + \dots)(N_0 + N_1 s + \dots) = 0$

written as <u>block-Toeplitz</u> system of equations

$$\begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ A_2 & A_1 & A_0 & \cdots \\ \vdots & A_2 & A_1 & \\ & \vdots & A_2 & \\ & & \vdots & \end{bmatrix} \begin{bmatrix} N_0 \\ N_1 \\ N_2 \\ \vdots \end{bmatrix} = 0$$

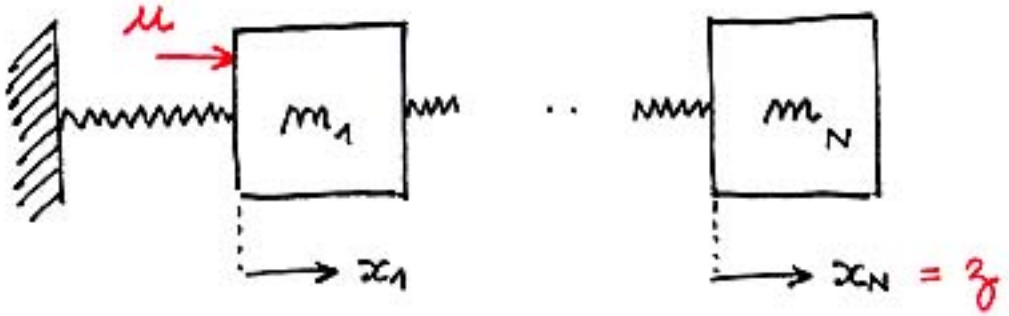
Generally degree of $N(s)$ is unknown

Bisection search initialized with lower/upper bounds

trade-off to be found between fast convergence

and structure exploitation ..

MASS-SPRING EXAMPLE



$$\begin{bmatrix} 1+s^2 & -1 & & & \\ -1 & 2+s^2 & -1 & & \\ & -1 & 2+s^2 & & \\ & & & \ddots & -1 \\ & & & -1 & 2+s^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u$$

$$\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} x = z$$

System transfer function from u to z

$$(Ms^2 + K)\, x = Gu, \quad z = Hx$$

$$\frac{z(s)}{u(s)} = H\,(Ms^2+K)^{-1}G = \frac{b(s)}{a(s)}$$

$$\begin{bmatrix} Ms^2+K & -G \end{bmatrix} \begin{bmatrix} \vdots \\ b(s) \\ a(s) \end{bmatrix} = 0$$

null-space extraction dim. = $(N) \times (N+1)$ degree = 2

OUTLINE

1. POLYNOMIAL MATRIX EQUATIONS

   1.1. pole placement control (BLOCK)

   1.2. null-space extraction TOEPLITZ

2. POLYNOMIAL OPTIMIZATION (PSEUDO) HANKEL

POLYNOMIAL OPTIMIZATION

$$\min_{x \in \mathbb{R}^n} \quad p_0(x)$$
$$\text{s.t.} \quad p_k(x) \geq 0 \quad k = 1, 2 \dots m$$

$p_0, p_1 \dots p_m$ polynomials

Difficult problem in general - NP-hard

includes 0-1 programming $x_i \in \{0,1\} \iff x_i(x_i - 1) = 0$

$$\iff x_i^2 - x_i \geq 0$$
$$x_i^2 - x_i \leq 0$$

Non-convex problem in general

many local and global minimizers

$$\max \quad \|x\|_2 = \sum_i x_i^2$$
$$\text{s.t.} \quad x_i^2 \leq 1$$

POLYNOMIAL OPTIMIZATION IN CONTROL

Robustness analysis - μ theory

$$\begin{aligned} \min \quad & p_0(x) \\ \text{s.t.} \quad & x_i \geq 0 \\ & \sum_i x_i = 1 \end{aligned}$$

Fixed-order controller design

$$\begin{aligned} \min \quad & p_0(x) \\ \text{s.t.} \quad & P(x) = P^T(x) \succeq 0 \end{aligned}$$

$m \times m$ semidef. positive

Descartes' rule of signs : $P(x) \succeq 0 \iff p_k(x) \geq 0 \quad k = 1..m$

$$\det(t\mathbb{I}_m - P(x)) = t^m + \sum_{k=1}^{m} (-1)^k p_k(x)\, t^{m-k}$$

POLYNOMIAL PROGRAMMING

Local search = NLP (Newton's method, Lagrangian ..)

Global search exhaustive branch and bound

or ...

use of convex relaxations

Most suitable are semidefinite programming relaxations

a generalization of linear programming

and convex quadratic programming

Theory of moments

Polynomial global optimization

$$\begin{array}{ll} \min_x & p_0(x) \\ \text{s.t.} & x \in \mathbb{K} \end{array}$$

on (non-convex) semialgebraic set

$$\mathbb{K} = \{x \in \mathbb{R}^n \,:\, p_i(x) \geq 0,\ i = 1, 2, \ldots\} \subset \mathbb{R}^n$$

Defining μ as a probability measure
the equivalent moment problem is

$$\begin{array}{ll} \min_\mu & \int p_0(x)\mu(dx) \\ \text{s.t.} & \mu(\mathbb{K}) = 1 \end{array}$$

meaning that μ is supported on $\mathbb{K}$
implying that $\mu(\mathbb{R}^n/\mathbb{K}) = 0$

Measure support constraints

Let $1, x_1, \ldots, x_n, x_1^2, \ldots, x_n^k, \ldots$ be a basis for the vector space of n-variate polynomials

$$f(x) = \sum_\alpha f_\alpha x^\alpha$$

Let $y = \{y_\alpha\}$ be an infinite sequence indexed in this basis, and consider the linear functional

$$f \mapsto L_y(f) = \sum_\alpha f_\alpha y_\alpha$$

Sequence y has a representing measure μ with support contained in $\mathbb{K}$, i.e.

$$\exists\, \mu \quad \text{s.t.} \quad y_\alpha = \int_{\mathbb{K}} x^\alpha d\mu \quad \forall \alpha$$

iff

$$L_y(f^2) \geq 0 \quad L_y(f^2 p_i) \geq 0 \quad i = 1, 2, \ldots$$

for all polynomials f

For a fixed degree of f, measure constraints are then relaxed...

LMI relaxations of moment problems

..to a convex linear matrix inequality (LMI) constraint on y:

$$L_y(f^2) \geq 0 \quad \forall f \text{ deg } f \leq k \iff M_k(y) \succeq 0$$

$M_k(y)$ is a moment matrix of order k

Ex. $n = 2$ variables, degree $k = 2$

$$M_2(y) = \left[\begin{array}{c|cc|ccc} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ \hline y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ \hline y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right]$$

If μ probability measure then $y_{00} = 1$

Indices are obtained by summing up row and column indices

Blocks feature monomials of fixed degrees

Localizing moment constraints

Similarly, the support constraint $\mu(\mathbb{R}^n/\mathbb{K}) = 0$ is relaxed to an LMI moment constraint

$$L_y(f^2 p_i) \geq 0 \quad \forall f \text{ deg } f \leq k \iff M_k(p_i, y) \succeq 0$$

$M_k(p_i, y)$ is a localization matrix

Ex. $n = 2$ variables, degree $k = 1$ and $p(x) = 1 - x_1^2 - x_2^2$

$$M_1(p, y) = \left[\begin{array}{c|cc} y_{00} - y_{20} - y_{02} & y_{10} - y_{30} - y_{12} & y_{01} - y_{21} - y_{03} \\ \hline y_{10} - y_{30} - y_{12} & y_{20} - y_{40} - y_{22} & y_{11} - y_{21} - y_{12} \\ y_{01} - y_{21} - y_{03} & y_{11} - y_{21} - y_{12} & y_{02} - y_{22} - y_{04} \end{array}\right]$$

Combining moment and localization constraints, we obtain an

LMI relaxation of given order k

of a polynomial moment problem

GLObal OPTImization over POLYnomials

GloptiPoly is as an open-source, general purpose and user-friendly Matlab software

Aimed at small to medium non-convex global optimization problems with multivariable real-valued polynomial objective functions and constraints

Software and documentation available at

www.laas.fr/∼henrion/software/gloptipoly

Version 2 described in ACM TOMS June 2003

Methodology

GloptiPoly builds and solves a hierarchy of successive convex linear matrix inequality (LMI) relaxations of increasing size, whose optima are guaranteed to converge asymptotically to the global optimum

Relaxations are built from LMI formulation of
- primal sum of squares (SOS) decomposition
- dual theory of moments

of positive multivariable polynomials

In practice convergence is ensured fast
typically at 2nd or 3rd LMI relaxation

LMI relaxation technique

Polynomial optimization problem

$$\begin{array}{ll} \min & p_0(x) \\ \text{s.t.} & p_k(x) \geq 0, \quad k = 1, \ldots, m \end{array}$$

When $p^\star$ is the global optimum, SOS representation of positive polynomial

$$p_0(x) - p^\star = q_0(x) + \sum_{k=1}^{m} p_k(x) q_k(x)$$

where unknowns $q_k(x)$ are SOS polynomials similar to Karush/Kuhn/Tucker multipliers

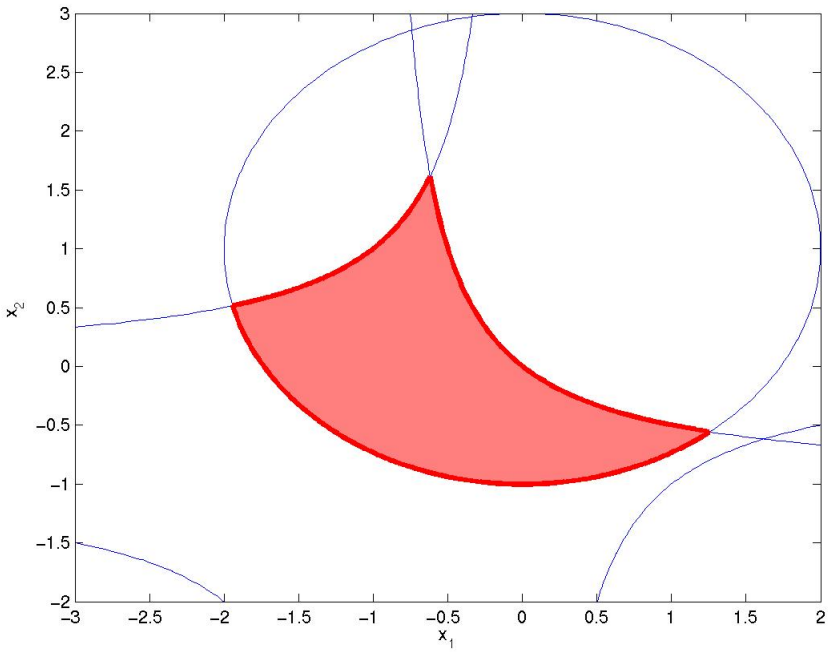
Using LMI representation of SOS polynomials successive LMI relaxations are obtained by increasing degrees of sought polynomials $q_k(x)$

Theoretical proof of asymptotic convergence...
..but no tight degree upper bounds (not yet)

LMI relaxations: illustration (1)

Non-convex quadratic problem

$$\begin{array}{ll} \max & x_2 \\ \text{s.t.} & 3 - 2x_2 - x_2^1 - x_2^2 \geq 0 \\ & -x_1 - x_2 - x_1 x_2 \geq 0 \\ & 1 + x_1 x_2 \geq 0 \end{array}$$



Non-convex feasible set delimited by circular and hyperbolic arcs

LMI relaxations: illustration (2)

LMI relaxation built by replacing each monomial $x_1^i x_2^j$ with a new decision variable x_{ij}

For example, quadratic expression

$$3 - 2x_2 - x_1^2 - x_2^2 \geq 0$$

replaced with linear expression

$$3 - 2x_{01} - x_{20} - x_{02} \geq 0$$

New decision variables x_{ij} satisfy non-convex relations such as $x_{10}x_{01} = x_{11}$ or $x_{20} = x_{10}^2$

Relax these non-convex relations by enforcing LMI constraint

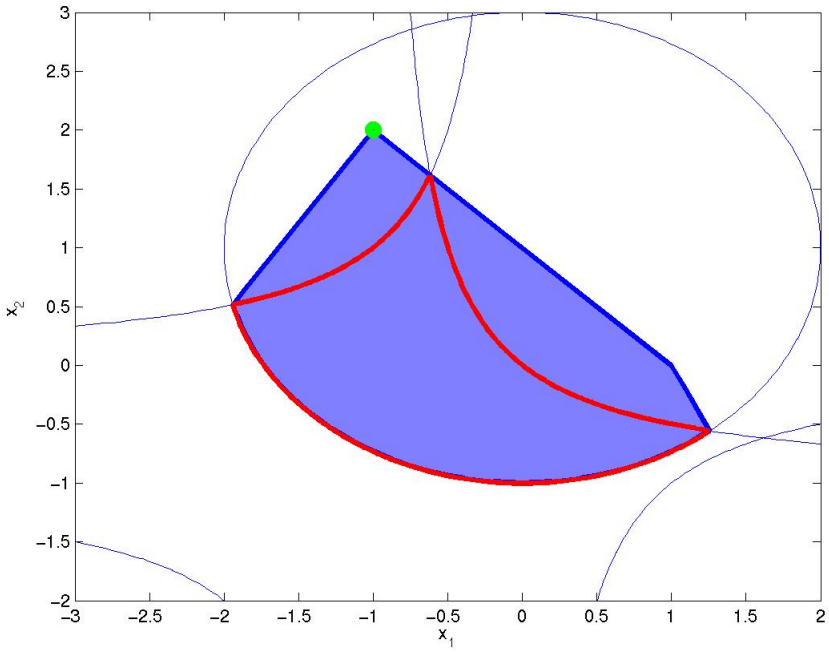
$$M_1^1 = \left[\begin{array}{c|cc} 1 & x_{10} & x_{01} \\ \hline x_{10} & x_{20} & x_{11} \\ x_{01} & x_{11} & x_{02} \end{array}\right] \geq 0$$

Moment or measure matrix of first order relaxing monomials of degree up to 2

LMI relaxations: illustration (3)

First LMI relaxation given by

$$
\begin{array}{ll}
\max & x_{01} \\
\text{s.t.} & \begin{bmatrix} 1 & x_{10} & x_{01} \\ x_{10} & x_{20} & x_{11} \\ x_{01} & x_{11} & x_{02} \end{bmatrix} \succeq 0 \\
& 3 - 2x_{01} - x_{20} - x_{02} \geq 0 \\
& -x_{10} - x_{01} - x_{11} \geq 0 \\
& 1 + x_{11} \geq 0
\end{array}
$$



Projection of the LMI feasible set onto the plane x_{10}, x_{01} of first-order moments

LMI optimum $= 2 =$ upper-bound on global optimum

LMI relaxations: illustration (4)

To build second LMI relaxation, the moment matrix must capture expressions of degrees up to 4

$$M_2^2(y) = \left[\begin{array}{c|cc|cc|c} 1 & x_{10} & x_{01} & x_{20} & x_{11} & x_{02} \\ \hline x_{10} & x_{20} & x_{11} & x_{30} & x_{21} & x_{12} \\ x_{01} & x_{11} & x_{02} & x_{21} & x_{12} & x_{03} \\ \hline x_{20} & x_{30} & x_{21} & x_{40} & x_{31} & x_{22} \\ x_{11} & x_{21} & x_{12} & x_{31} & x_{22} & x_{13} \\ x_{02} & x_{12} & x_{03} & x_{22} & x_{13} & x_{04} \end{array}\right]$$

Constraints are also lifted and relaxed with the help of localization matrices

Second LMI provides tighter relaxation

$$\begin{array}{ll} \max & x_{01} \\ \text{s.t.} & \left[\begin{array}{cccccc} 1 & x_{10} & x_{01} & x_{20} & x_{11} & x_{02} \\ x_{10} & x_{20} & x_{11} & x_{30} & x_{21} & x_{12} \\ x_{01} & x_{11} & x_{02} & x_{21} & x_{12} & x_{03} \\ x_{20} & x_{30} & x_{21} & x_{40} & x_{31} & x_{22} \\ x_{11} & x_{21} & x_{12} & x_{31} & x_{22} & x_{13} \\ x_{02} & x_{12} & x_{03} & x_{22} & x_{13} & x_{04} \end{array}\right] \succeq 0 \end{array}$$

$$\left[\begin{array}{ccc} 3-2x_{01}-x_{20}-x_{02} & 3x_{10}-2x_{11}-x_{30}-x_{12} & 3x_{01}-2x_{02}-x_{21}-x_{03} \\ 3x_{10}-2x_{11}-x_{30}-x_{12} & 3x_{20}-2x_{21}-x_{40}-x_{22} & 3x_{11}-2x_{12}-x_{31}-x_{13} \\ 3x_{01}-2x_{02}-x_{21}-x_{03} & 3x_{11}-2x_{12}-x_{31}-x_{13} & 3x_{02}-2x_{03}-x_{22}-x_{04} \end{array}\right] \succeq 0$$

$$\left[\begin{array}{ccc} -x_{10}-x_{01}-x_{11} & -x_{20}-x_{11}-x_{21} & -x_{11}-x_{02}-x_{12} \\ -x_{20}-x_{11}-x_{21} & -x_{30}-x_{21}-x_{31} & -x_{21}-x_{12}-x_{22} \\ -x_{11}-x_{02}-x_{12} & -x_{21}-x_{12}-x_{22} & -x_{12}-x_{03}-x_{13} \end{array}\right] \succeq 0$$

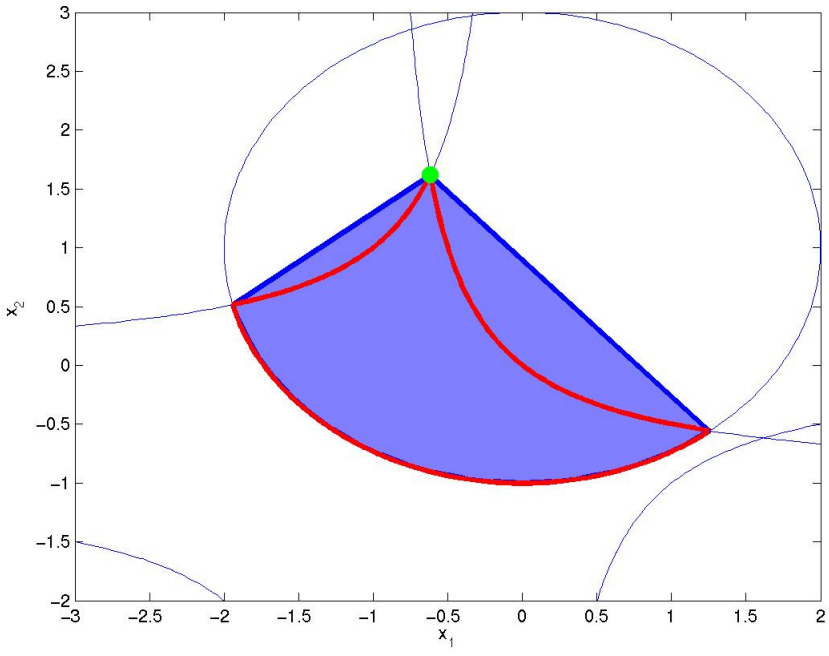
$$\left[\begin{array}{ccc} 1+x_{11} & x_{10}+x_{21} & x_{01}+x_{12} \\ x_{10}+x_{21} & x_{20}+x_{31} & x_{11}+x_{22} \\ x_{01}+x_{12} & x_{11}+x_{22} & x_{02}+x_{13} \end{array}\right] \succeq 0$$

LMI relaxations: illustration (5)

Optimal value of 2nd LMI relaxation = 1.6180 = global optimum within numerical accuracy

Numerical certificate = moment matrix has rank one

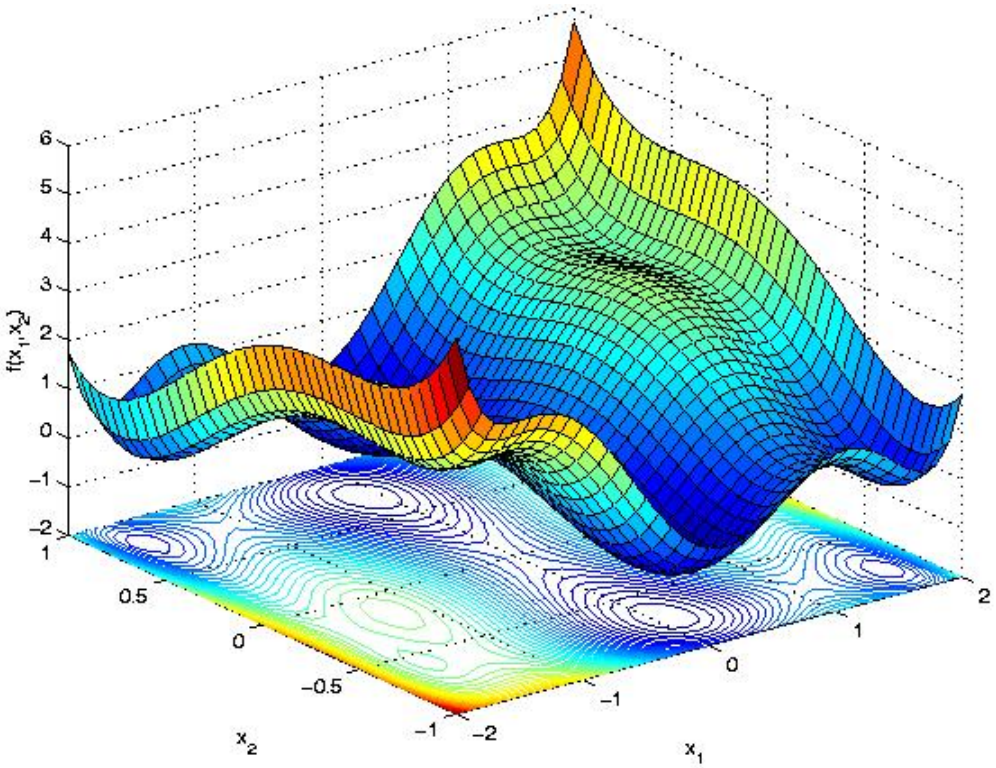
First order moments $(x^*_{10}, x^*_{01}) = (-0.6180, 1.6180)$ provides optimal solution of original problem



Benchmark examples
Continuous problems

Mostly from Floudas/Pardalos 1999 handbook

About 80 % of pbs solved with LMI relaxation of small order (typically 2 or 3) in less than 3 seconds on a PC Pentium IV at 1.6 MHz with 512 Mb RAM

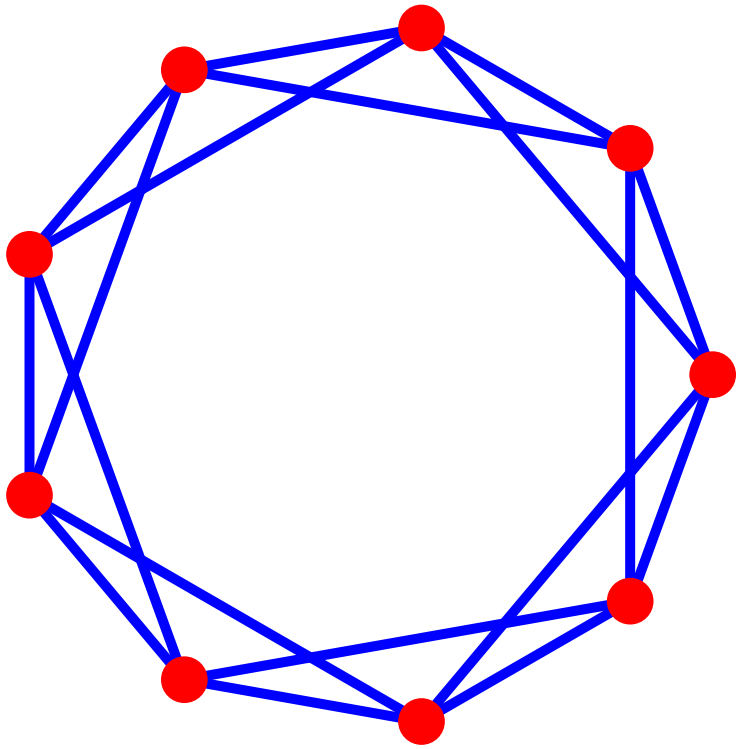


Six-hump camel back function

Benchmark exmaples
Discrete problems

From Floudas/Pardalos handbook and also Anjos' Ph.D (Univ Waterloo)

By perturbing criterion (destroys symmetry) global convergence ensured on 80 % of pbs in less than 4 seconds



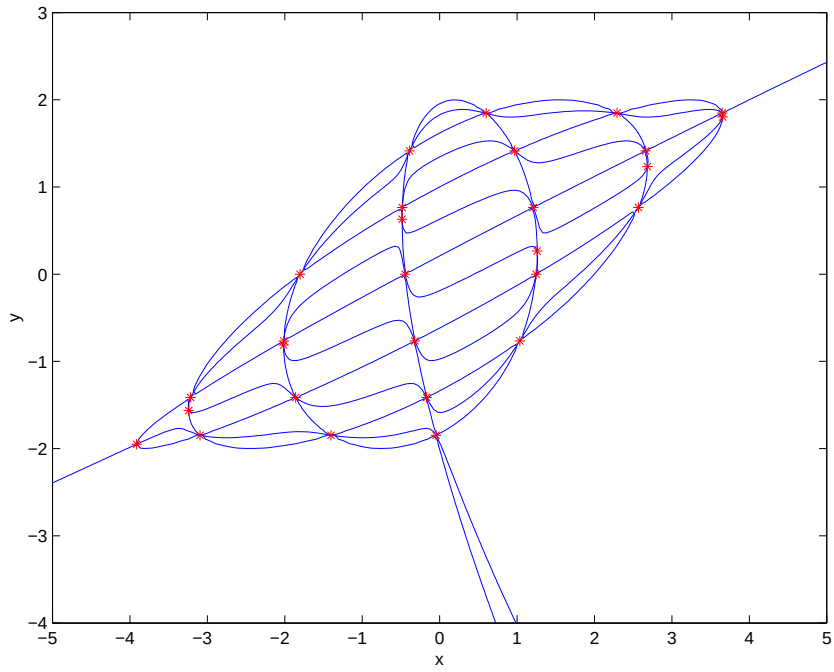
MAXCUT on antiweb AW_9^2 graph

Benchmark examples
Polynomial systems of equations

From Verschelde's database and Frisco INRIA project
Real coefficients & solutions only

Out of 59 systems:
- 61 % solved in t $<$ 10 secs
- 20 % solved in 10 $<$ t $<$ 100 secs
- 10 % solved in t $\geq$ 100 secs
- 9 % out of memory

No criterion optimized
No enumeration of all solutions



Intersections of seventh and eighth degree polynomial curves

Dedicated SDP methods

Structure of moment matrices ?

Univariate = Hankel moment matrix

Multivariate = sparse matrix coeffs

$$\left[\begin{array}{c|cc|ccc} 1 & x_{10} & x_{01} & x_{20} & x_{11} & x_{02} \\ \hline x_{10} & x_{20} & x_{11} & x_{30} & x_{21} & x_{12} \\ x_{01} & x_{11} & x_{02} & x_{21} & x_{12} & x_{03} \\ \hline x_{20} & x_{30} & x_{21} & x_{40} & x_{31} & x_{22} \\ x_{11} & x_{21} & x_{12} & x_{31} & x_{22} & x_{13} \\ x_{02} & x_{12} & x_{03} & x_{22} & x_{13} & x_{04} \end{array}\right]$$

Known sparsity pattern: LMI matrix coeffs in pre-factored form (Cholesky)

Efficient low-rank algebra to compute gradient & Hessian in interior-point codes ?

Moment substitutions: trade-off between sparsity and number of variables