Sum of Squares programming

TK18 2020 - Fifth colloquium

Morten Hovd

1

Aims

- Introducing Sum of Squares (SoS)
  - Introducing SoS decomposition as a means of proving non-negativity of polynomials
  - Illustrating SoS programming as a tool for control analysis and design for polynomial non-linear systems.
- Illustrations based on recent work on discrete-time bilinear systems

2

Polynomials and monomials

- A multivariate polynomial $f(x)$ has the form

$$f(x) = \sum_k c_k x_1^{a_{k1}} x_2^{a_{k2}} \cdots x_n^{a_{kn}}$$

where a_{ki} are non-negative integers

- A monomial is a term in a polynomial (without the coefficient c_k):

$$m_k(x) = x_1^{a_{k1}} x_2^{a_{k2}} \ldots x_n^{a_{kn}}$$

Sum of Squares decomposition

- A polynomial $f(x)$ is a sum of squares if it can be expressed as a sum of squared terms

$$f(x) = \sum_{i=1}^{N} h_i^2(x) = \sum_{i=1}^{N} (q_i^T v(x))^2 = v^T(x) Q v(x)$$

for some vector of monomials $v(x)$ and PSD matrix Q

- Similarly, a symmetric polynomial matrix $M(x)$ is an SoS matrix if it can be decomposed as

$$M(x) = H^T(x) H(x)$$

for some polynomial matrix $H(x)$

- The primary use of an SoS decomposition is to find a guarantee for the non-negativity (or positivity) of a polynomial expression

Positivity and SoS

- The existence of a SoS decomposition is only a sufficient condition for non-negativity of a polynomial. For instance, the following polynomial is non-negative, but no SoS decomposition exists:

$$M(x,y,z) = x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$$

- The existence of an SoS decomposition is equivalent to the non-negativity of a polynomial only in the following cases:
  - Polynomials in two variables
  - Quadratic forms (polynomials of maximum order 2)
  - Fourth order polynomials in three variables
- Trivial conditions for the existence of an SoS decomposition:
  - Any constant term must be non-negative
  - The highest order of the polynomial must be even

SoS polynomials

- From definition on previous slide, we have an SoS polynomial if it can be expressed as

$$f(x) = \sum_{i=1}^{N} h_i^2(x) = \sum_{i=1}^{N} (q_i^T v(x))^2 = v^T(x) Q v(x)$$

for some vector of monomials $v(x)$ and some $Q \geq 0$

- However, since the elements of $v(x)$ are not independent, the matrix Q is not unique.

Example

- Consider the following example (Parillo, thesis, p. 41):

$$f(x_1,x_2)=2x_1^4+2x_1^3x_2-x_1^2x_2^2+5x_2^4=\begin{bmatrix}x_1^2\\x_2^2\\x_1x_2\end{bmatrix}^T\begin{bmatrix}2&0&1\\0&5&0\\1&0&-1\end{bmatrix}\begin{bmatrix}x_1^2\\x_2^2\\x_1x_2\end{bmatrix}$$

- The matrix Q is trivially not positive (semi-)definite. However, we can use the fact that the elements of $v(x)$ are not independent to introduce a new variable in the matrix Q:

$$f(x_1,x_2)=2x_1^4+2x_1^3x_2-x_1^2x_2^2+5x_2^4=\begin{bmatrix}x_1^2\\x_2^2\\x_1x_2\end{bmatrix}^T\begin{bmatrix}2&-\lambda&1\\-\lambda&5&0\\1&0&-1+2\lambda\end{bmatrix}\begin{bmatrix}x_1^2\\x_2^2\\x_1x_2\end{bmatrix}$$

- Easy to choose λ such that Q is positive definite
  - E.g., $\lambda = 1$
  - The set of parameters λ which makes $Q \geq 0$ is convex
    - Holds in general for symmetric matrices that depend linearly on one or more parameters

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

7

SoS programming

- SoS constraints are convex
  - Assuming linear dependence on decision variables in control design or analysis formulations
  - Linear in free variables resulting from algebraic dependence of elements of monomial vector $v(x)$
- Can be combined with a convex optimization criterion to define a convex semi-definite programming (SDP) problem.

- Available software
  - YALMIP
  - SOSTOOLS http://www.cds.caltech.edu/sostools/
  - Both are free for non-commercial use, and are toolboxes for Matlab
    - Personal experience only with Yalmip
  - In addition, an appropriate SDP solver is required

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

8

Numerical issues

- The optimal matrix Q will often be severely ill-conditioned, with some very small eigenvalues
- Numerical conditioning may be improved by preprocessing:
  - Removing from the vector $v(x)$ elements (monomials) that will not be required
  - Imposing a block-diagonal structure on Q, depending on symmetries in the polynomial
- Analysis of an initial solution may give hints about additional structure in the optimal Q that can further improve conditioning
  - Re-run after implementing such additional structure
- Good software (e.g. Yalmip) will perform the pre-processing automatically, and allow for automatic post-processing
- For the user, there is therefore little need to know the details of these manipulations
  - Interested readers may consult Löfberg (IEEE TAC, 2009)

Proof of positivity of a polynomial

- SoS optimization will always yield a positive definite Q (if a 'primal' optimization algorithm is chosen). Does this mean that the original polynomial is an SOS???
- Important to consider the accuracy of the solution
- The *residual* is defined as the largest coefficient in the polynomial $f(x) - v^T(x)Qv(x)$
  - Due to floating point arithmetic in solvers, it is rare to find a zero residual
- Define
  - $\lambda_{min}(Q) = \min eig(Q)$
  - M: number of monomials in $v(x)$
- The polynomial $f(x)$ is guaranteed to be positive provided
  - $\lambda_{min}(Q) \geq 0$
  - $\lambda_{min}(Q) \geq M \times abs(residual)$
  - See Löfberg (2009)

Proof of positivity

- Q is often ill-conditioned, can be important to check
- A 'primal' solver will always give $Q \geq 0$
  - But still important to check the eigenvalues compared to the size of the residual
- A 'dual'/infeasible path solver may give an indefinite Q

- Occasionally, chosing a dual solver may (for numerical reasons) give a valid solution that is not found using a primal formulation.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

11

LMIs and SoS programming

- SoS problems are solved as a 'special case' of LMI problems: does that mean that the same tricks can be used for SoS's as for LMI's?
  - Answer: only some of them!

- Linearizing change of variables:
  - Let a problem depend (linearly) on $P > 0$ and PK, but without K appearing independently. May then introduce $L = PK$, to formulate a problem linear in L and P.
  - Note that
    - K can always be calculated from L and P.
    - A constraint such as $L = PK$ is bilinear in the variables P and K, and is hence NOT convex.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

12

Congruence transform

- For a matrix valued SoS problem described by $M(x) \geq 0$, and a full rank $W(x)$ (not necessarily symmetric or positive definite)

$$M(x) \geq 0 \Leftrightarrow W^T(x)M(x)W(x) \geq 0$$

- However, this congruence transform can NOT be used together with the scalarized Schur complement (to be shown below)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

13

Scalarization

- For LMIs, $L > 0 \Leftrightarrow x^T L x > 0 \forall x$
- For SoS, $M(x) > 0$ and $x^T M(x) x > 0$ are very different
  - $M(x) > 0$ implies $z^T M(x) z > 0 \forall x, \forall z$ (with no relationship between x and z).
  - It is therefore an advantage to formulate scalar-valued SoS problems (i.e., to *scalarize* the problems) , see below for illustrations.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

14

A scalarized Schur complement

Consider:

$M(x) = \begin{bmatrix} E(x) & F^T(x) \\ F(x) & P(x) \end{bmatrix}$ with $P(x)$ symmetric and invertible.

Then:

$$\begin{bmatrix} x \\ z \end{bmatrix}^T M(x) \begin{bmatrix} x \\ z \end{bmatrix} > 0, \forall \{x, z\} \neq \{0,0\}$$

is equivalent to

$$x^T \big(E(x) - F^T(x) P^{-1}(x) F(x) \big) x > 0 \forall x \neq 0$$

and $z^T P(x) z > 0 \forall z \neq 0, \forall x$

(Note that $z^T P(x) z > 0 \forall z \neq 0, \forall x \Leftrightarrow z^T P^{-1}(x) z > 0 \forall z \neq 0, \forall x$)

This follows from

$$M(x) = \begin{bmatrix} I_E & F^T(x)P^{-1}(x) \\ 0 & I_P \end{bmatrix} \begin{bmatrix} E(x) - F^T(x)P^{-1}(x)F(x) & 0 \\ 0 & P(x) \end{bmatrix} \begin{bmatrix} I_E & 0 \\ P^{-1}(x)F(x) & I_P \end{bmatrix}$$

15

A scalarized Schur product

- Consider

$$\begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} I_E & 0 \\ P^{-1}(x)\mathrm{F}(\mathrm{x}) & I_P \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

- Clearly,w can take any value whatever the value of x, and for any w there is a corresponding value of z whatever the value of x.

- Note that using the congruence transform – pre- and postmultiplying the matrix $\mathrm{M(x)}$ with non-singular matrices – will ruin the equality between the values of x in $\mathrm{M(x)}$ and the values in the vectors pre-and postmultiplying $\mathrm{M(x)}$.
  - Hence the congruence transform should not be used with the scalarized Scur complement.

16

The S-procedure

- This applies in much the same way as for LMIs:

We want to prove that $f(x) > 0$ wherever $g(x) < 0$. This may be formulated as

$$f(x) + s(x)g(x) > 0$$

for some SoS polynomial $s(x)$.

Note that

- $s(x)$ is not restricted to be a positive constant, it can be any SoS polynomial
- Optimizing over parameters in both $s(x)$ and $g(x)$ (at the same time) will lead to a bilinear (non-convex) problem.

17

SoS programming for bilinear discrete-time systems

18

Bilinear discrete-time systems

- Systems of the form

$$x_{k+1} = Ax_k + \sum_{i=1}^{m} \left(B_i x_k + b_i\right) u_{i,k} = Ax_k + \left(B_x + B\right) u_k$$

- Assuming (possibly non-linear) state feedback, stability is ensured by fulfilling the Lyapunov difference inequality

$$x_k^T P x_k - x_{k+1}^T P x_{k+1} > 0$$
$$x_k^T P x_k - \left(Ax_k + (B_x + B)u_k(x_k)\right)^T P\left(Ax_k + (B_x + B)u_k(x_k)\right) > 0$$

- If system is unstable with $u_k = 0$ (for a state affected by bilinearity):
  - No $u_k(x_k)$ polynomial in the state can give global quadratic stability
  - $u_k(x_k)$ must be *a ratio* of two polynomials of the same order to achieve global quadratic stability
  - Note that continuous-time bilinear systems may allow polynomial $u(x)$ and still achieve global quadratic stability

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

19

Proposed controller

- The proposed controller takes the form

$$u_k(\mathrm{x}_k) = \frac{C(x_k)x_k}{c_0(x_k)+1}$$

Where $C(x_k)$ is a polynomial matrix and $c_0(x_k)$ is an SoS polynomial

  - Adding 1 to the numerator prevents the numerator from becoming very small, which could lead to an ‘explosion’ in the input

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

20

Region of convergence

- Given a quadratic Lyapunov function $V = x_k^T P x_k$, a polynomial matrix $C(x_k)$ and SoS polynomials $c_0(x_k)$ and $s_1(x_k, z)$, the closed loop systems is stable $\forall x_k \mid x_k^T P x_k < \gamma$ provided

$$\begin{bmatrix} x_k \\ z \end{bmatrix}^T M(x) \begin{bmatrix} x_k \\ z \end{bmatrix} - s_1(x_k, z)(\gamma - x_k^T P x_k) > 0$$

where

$$M(x_k) = \begin{bmatrix} (c_0(x_k)+1)P & \left((c_0(x_k)+1)\mathrm{A} + (\mathrm{B}_x + B)C(x_k)\right)^T P \\ P\left((c_0(x_k)+1)\mathrm{A} + (\mathrm{B}_x + B)C(x_k)\right) & (c_0(x_k)+1)P \end{bmatrix}$$

- Follows from $(c_0(x_k) + 1)$ being strictly positive, the scalarized Schur complement and the S-procedure

Input saturation

- Consider input constraints of the form $-u_{i,max} \leq u_i \leq u_{i,max}$
- Let $c_i(x_k)$ denote an element of the polynomial vector $(C(x_k)x_k)$, $c_o(x_k)$ be an SoS polynomial (as before), and $q_i(x_k)$ be an SoS polynomial. Then the input constraint is satisfied provided $\forall x_k \mid x_k^T P x_k < \gamma$

$$\begin{bmatrix} (c_0(x_k)+1)\,\mathrm{u}_{i,\max}^2 - q_i(x_k)(\gamma - x_k^T P x_k) & c_i(x_k) \\ c_i(x_k) & (c_0(x_k)+1) \end{bmatrix} > 0$$

- Follows from Schur complement and S-procedure, and $(c_0(x_k) + 1)$ being strictly positive

Rate of convergence

- An exponential rate of convergence specified by the scalar α is easily included by changing the matrix $M(x_k)$ above:

$$M(x_k)=\begin{bmatrix}(1-\alpha)(c_0(x_k)+1)P & \left((c_0(x_k)+1)\,\mathrm{A}+(\mathrm{B}_x+B)C(x_k)\right)^T P \\ P\left((c_0(x_k)+1)\,\mathrm{A}+(\mathrm{B}_x+B)C(x_k)\right) & (c_0(x_k)+1)P\end{bmatrix}$$

- Maximizing the region of convergence can easily lead to slow control, specifying a certain rate of convergence can avoid this (at the expense of having a smaller region with this specified rate of convergence)

Optimization formulation

- If α, γ, and P are specified, the coefficients of $C(x_k), c_0(x_k), q_i(x_k)$, and $s_1(x_k, z)$ all enter linearly in the inequalities above (and vice versa).
- For given γ, and P we can find $C(x_k), c_0(x_k), q_i(x_k)$, and $s_1(x_k, z)$
  - Formulated as a feasibility problem, i.e., an optimization with an empty objective in Yalmip
- For given $C(x_k), c_0(x_k), q_i(x_k)$, and $s_1(x_k, z)$, we can maximize γ with P as a free variable.
  - To avoid γ and P growing without bound (without describing a larger region), we will need to add a normalizing constraint such as $trace(P) = constant$.
- Iterating between the two optimization problems, we gan design controllers stabilizing gradually increasing regions.
- Will typically initialize with LQ solution of linearized system

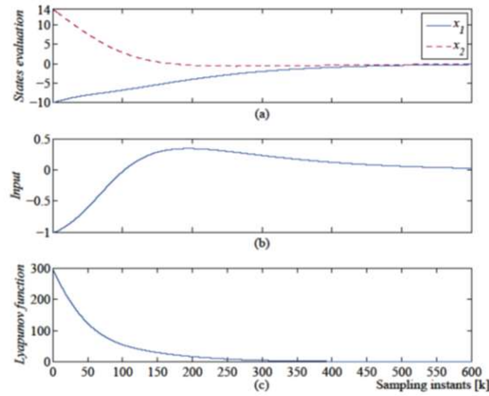
Example 1

Model

$$A = \begin{bmatrix} 1 & 0.01 \\ 0.01 & 1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.001 & 0 \\ 0 & -0.004 \end{bmatrix},$$

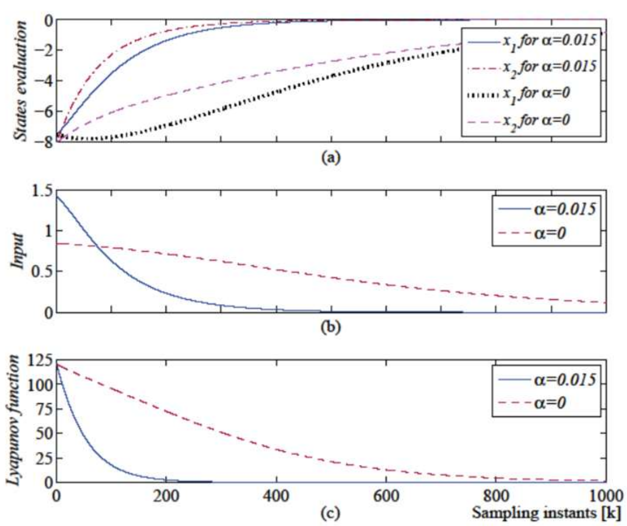
$$b_1 = \begin{bmatrix} 0.09 \\ 0.09 \end{bmatrix}$$



Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

25

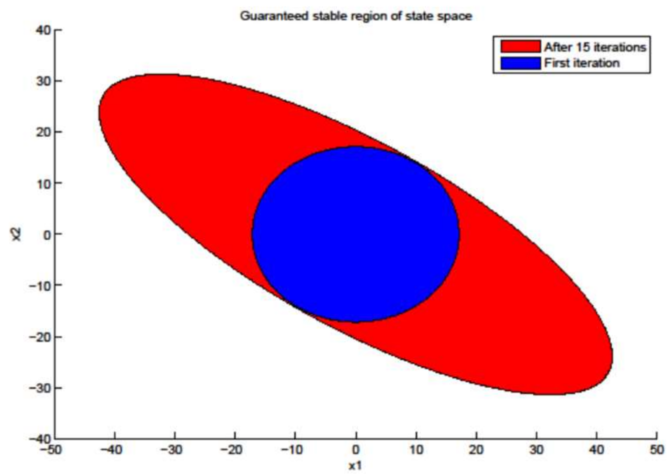
Example 1



Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

26

Example 1



Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

27

Example 2

- Model:

$$A = \begin{bmatrix} 1.10 & -0.2 & -0.34 \\ -0.06 & 0.7 & -0.42 \\ 0.41 & 0.41 & 0.90 \end{bmatrix},$$

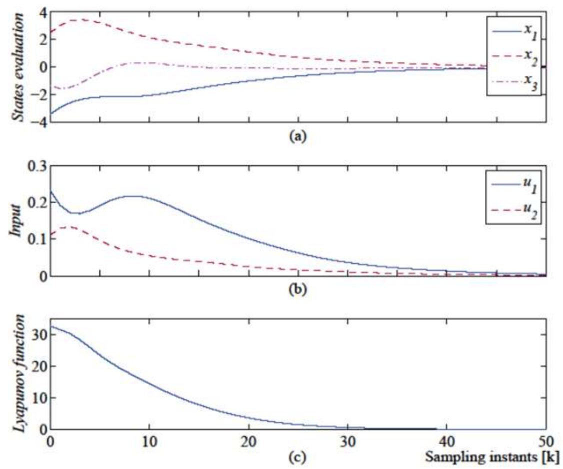
$$B_1 = \begin{bmatrix} -0.12 & -0.22 & 0.36 \\ -0.32 & 0.48 & 0.36 \\ -0.35 & 0.36 & -0.18 \end{bmatrix}, B_2 = \begin{bmatrix} -0.18 & 0.30 & 0.07 \\ -0.03 & -0.18 & -0.38 \\ 0.55 & -0.74 & -0.77 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 3.75 \\ 1.05 \\ -0.85 \end{bmatrix}, b_2 = \begin{bmatrix} 0 \\ -1.33 \\ -0.49 \end{bmatrix}$$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

28

Example 2



Achieved much larger stable region than previous investigators for the same system

ALTERNATIVE FORMULATIONS FOR CONTROLLER DESIGN

x_{k+1} in an augmented 'state space'

- Previously: use model equations to express x_{k+1}using x_k and u_k.
- May instead express the problem using both x_k and x_{k+1}.
- Given a polynomial LF $V(x)$ and model $x_{k+1} = f(x_k, u_k)$ where $f(x_k, u_k)$ is a polynomial vector:

$$V(x_k) - V(x_{k+1}) + m^T(x_k, x_{k+1})\big(x_{k+1} - f(x_k, u_k)\big) \\ - s(x_k, x_{k+1})\big(\gamma - V(x_k)\big) > 0$$

- Here: $s(x_k, x_{k+1})$ is SoS, $m^T(x_k, x_{k+1})$ free (not necessarily SoS)
- If expression holds globally (in $[x_k^T \quad x_{k+1}^T]^T$), it also has to hold along the trajectories of the system, where $\big(x_{k+1} - f(x_k, u_k)\big)$=0.

31

x_{k+1} in an augmented 'state space'

- Allows for simultaneously optimizing the parameters of LF and controller.
  - May be an advantage in some cases
  - The vector $m(x_k, x_{k+1})$ may introduce many new parameters if there are many states
  - Higher order LFs more easily expressed than with previous formulation
- Straight forward if $u_k$enters linearly in $f(x_k, u_k)$.
  - For controller parametrization as used previously, or purely polynomial controllers
- Need additional SoS constraints to ensure
  - LF>0
  - Stable region increasing from one iteration to the next
  - Details omitted for brevity
- Both formulations can relatively easily be adapted to continuous-time systems
  - Using the gradient of the LF instead of the LF difference

32

u_k non-linear in $f(x_k, u_k)$

- Augmented state space including also u_k
- Expand $f(x_k, u_k)$ to include controller, e.g., for $u_k = c(x_k)$:

$$V(x_k) - V(x_{k+1}) + m^T(x_k, x_{k+1}, u_k)\left(\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} - \begin{bmatrix} f(x_k, u_k) \\ c(x_k) \end{bmatrix}\right)$$
$$- s(x_k, x_{k+1}, u_k)\big(\gamma - V(x_k)\big) > 0$$

APPROXIMATING OPTIMIZATION-BASED CONTROLLERS

Optimization problem P_u

- Original problem
- $\min_{u_k, x_{k+1}} \frac{1}{2} x_{k+1}^T Q x_{k+1} + u_k^T R u_k$
- $x_{k+1} = A x_k + B u_k$
- $H_u u_k \leq 1$
- $V(x_{k+1}) \leq \gamma V(x_k)$
- 'One-step' MPC
  - Can be generalized to more common MPC formulations
- Imposing decrease of Lyapunov function directly ($\gamma < 1$), a.k.a. a *contraction constraint*
- Instead of using the value function as an LF

- Equivalent problem
- $\min_{u_k} \frac{1}{2} u_k^T H u_k + x_k^T F u_k$
- $H_u u_k \leq 1$
- $V(x_{k+1}) \leq \gamma V(x_k)$

Convex problem if $V(x_{k+1})$ is convex

Unconstrained solution can be found.
LMI techniques can be used to find a corresponding ellipsoid inside which this is valid.
Subsequent developments are applied in the constrained region, inside the set where
$V(x_k) < 1$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

35

KKT conditions of problem P_u

- $H u_k + F^T x_k + H_u^T \lambda_u + V_{k+1}^1(u_k) \lambda_q = 0$
- $H_u u_k \leq 1$
- $V(x_{k+1}) - \gamma V(x_k) \leq 0$
- $\lambda_u \geq 0$
- $\lambda_q \geq 0$
- $\lambda_u^T (H_u u_k - 1) + \lambda_q \big(V(x_{k+1}) - \gamma V(x_k)\big) = 0$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

36

Optimization problem P_c

- $\min\limits_{c, u_k, \lambda_u, \lambda_q} c$
- $Hu_k + F^T x_k + H_u^T \lambda_u + V_{k+1}^1(u_k)\lambda_q = 0$
- $H_u u_k \le 1$
- $V(x_{k+1}) - \gamma V(x_k) \le 0$
- $\lambda_u \ge 0$
- $\lambda_q \ge 0$
- $\lambda_u^T(H_u u_k - 1) + \lambda_q\big(V(x_{k+1}) - \gamma V(x_k)\big) \le c$
- A feasible solution to P_c is also a feasible solution to P_u
- If, at optimum, $c = 0$, the optimal solution to P_c is also an optimal solution to P_u

Approximate solution to P_u

- Let $J(u_k^*(c))$ be the value of the objective function of P_u evaluated using the input u_k^* from the solution to problem P_c.
- Let $J(u_k^*(0))$ be the (optimal) value of the objective function of P_c.
- Then $J\big(u_k^*(c)\big) - J\big(u_k^*(0)\big) < c$
  - The degree of suboptimality of P_c (relative to P_u) is bounded by the degree of violation of the complementarity constraints
  - Also holds for other convex quadratic objective functons with convex constraints

Optimization formulation

- Basic idea: use polynomial expressions for $u_k, \lambda_e, \lambda_q, V$
- However, when optimizing also over V, must include additional constraint $V_{xx} > 0$
  - Where V_{xx} is the matrix of second derivatives, i.e., $[V_{xx}]_{i,j} = \frac{\partial^2 V}{\partial i \partial j}$
  - Otherwise the initial formulation is not convex, and we cannot replace the optimization problem by the KKT conditions
- However, using the model equation to eliminate x_{k+1} will lead to a non-linear dependence of $V(k+1)$ on the controller parameters
  - And hence a severely non-convex problem
- Instead, the problem is lifted to a higher-dimensional space described by (x_k, x_{k+1})
  - Starting from the original problem formulation on slide 35

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

39

Optimization formulation

- $\min_{c, u_k, \lambda_u, \lambda_q, \lambda_e, \mu_i, V} c$
- $H_u u_k \leq 1,\ \ \lambda_u \geq 0,\ \ \lambda_q \geq 0$
- $R u_k - B^T \lambda_e + H_u^T \lambda_u + \mu_1^T (x_{k+1} - A x_k - B u_k) = 0$
- $Q x_{k+1} + \lambda_e + \lambda_q \nabla V(x_{k+1}) + \mu_2^T (x_{k+1} - A x_k - B u_k) = 0$
- $V(x_{k+1}) - \gamma V(x_k) + \mu_3^T (x_{k+1} - A x_k - B u_k) \leq 0$
- $\lambda_u^T (H_u u_k - 1) + \lambda_q \big(V(x_{k+1}) - \gamma V(x_k)\big) + \mu_4^T (x_{k+1} - A x_k - B u_k) \leq c$
- $V_{xx} > 0$
- V SOS, no sign constraint for μ's
- λ_u, λ_q only need to be positive in the constrained region, inside the set $V(x_k) < 1$. The same holds for all other inequality constraints. Implemented through S-procedure.
- c scalar, u_k function of x_k (only), V function of x_k or x_{k+1} as appropriate (depending on which time instant it refers to)
- Other variables functions of x_k, x_{k+1}
- Optimizing over polynomial coefficients! These enter linearly or bi-linearly above.
- Where $(x_{k+1} - A x_k - B u_k) = 0$ and $c = 0$ the original KKT conditions are fulfilled (complementarity constraint relaxed)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

40

Optimization formulation

- Additional features to
  - Account for state constraints by ensuring the level set of the LF is inside the state constraint
  - Systematically 'pushing out' the level set to make the guaranteed stable region as large as possible
  - Otherwise, a similar iterative solution procedure as described before

- Note the additional constraint specifying that the Hessian (second derivative matrix) of the LF is an SOS matrix
  - To guarantee the convexity of the contraction constraint.
  - The second derivative matrix of a polynomial is found by a standard function in Yalmip

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

41

Non-polynomial systems

- SoS design assumes polynomial or rational polynomial dynamics and LFs
- Some non-polynomial systems can be handled with the appropriate reformulation (see Papachhristodoulou and Pranja)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

42

NTNU
Trondheim

Furuta pendulum

- Pendulum on a rotary base
  - Rotate base to stabilize pendulum in the upright vertical position

$$\left(m_pL_r^2+\frac{1}{4}m_pL_p^2-\frac{1}{4}m_pL_p^2\cos(\alpha)^2+J_r\right)\ddot{\theta}-\left(\frac{1}{2}m_pL_pL_r\cos(\alpha)\right)\ddot{\alpha} \qquad (1)$$
$$+\left(\frac{1}{2}m_pL_p^2\sin\alpha\cos\alpha\right)\dot{\theta}\dot{\alpha}+\left(\frac{1}{2}m_pL_pL_r\sin(\alpha)\right)\dot{\alpha}^2 \;=\; \tau-B_r\dot{\theta}$$
$$-\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta}+\left(J_p+\frac{1}{4}m_pL_p^2\right)\ddot{\alpha}-\frac{1}{4}m_pL_p^2\cos(\alpha)\sin(\alpha)\dot{\theta}^2 \qquad (2)$$
$$-\frac{1}{2}m_pL_pg\sin(\alpha) \;=\; -B_p\dot{\alpha}$$

where

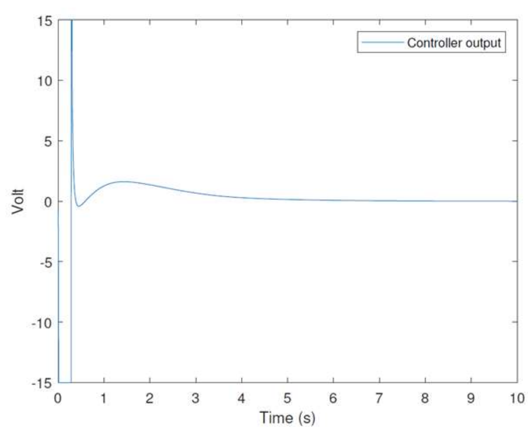
$$\tau=\frac{\eta_gK_g\eta_mk_t(V_m-K_gk_m\dot{\theta})}{R_m}$$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

43

NTNU
Trondheim

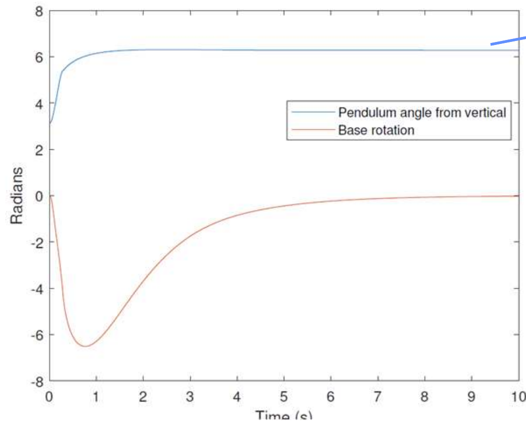
Furuta - input



Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

44

Furuta - states



Upright vertical position

Fourth order LF, third order controller.

One controller throughout, not separate controllers for ‘swingup’ and ‘stabilization’

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

45

Some literature

- P. Parillo, PhD thesis (mainly Chapter 4)
- J. Löfberg, Pre- and post-processing Sum-of-Squares programs in practice, IEEE TAC, vol. 54, no. 5, pp. 1007-1011, 2009
- A. Papachristodoulou and S. Pranja, Analysis of Non-polynomial Systems using the Sum of Squares Decomposition. In Henrion and Garulli (Eds), Positive Polynomials in Control, Springer LNCIS 312
- M. Vatani, Advanced Control Methods for Power Converters, Focusing on Modular Multilevel Converters. PhD Thesis, Dept. of Engineering Cybernetics, NTNU, 2016
- Y. Oishi. Direct design of a polynomial Model Predictive Controller, IFAC Symposium ROCOND, 2012
- S. Munir, M. Hovd, S. Olaru. Low complexity constrained control using higher degree Lyapunov functions. Automatica, vol 98, pp. 215-222, 2018.
- YALMIP wiki pages
- Books by G. Chesi, J. B. Lasserre and others, +++ (a lot)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

46