Summary of TTK18

Morten Fyhn Amundsen

November 27, 2016

Contents

1 Bi-level programming in constrained control

A bi-level program is an optimization problem that depends on the solution of another optimization problem, called "upper level" (UL) and "lower level" (LL) problems, respectively.

$$
\begin{aligned}
&\min_{x} f_{\text{UL}}(x, z) \\
&G_{\text{UI}}(x, z) \le 0 \\
&G_{\text{UE}}(x, z) = 0 \\
&z = \arg\min_{z} f_{\text{LL}}(x, z) \\
&G_{\text{LI}}(x, z) \le 0 \\
&G_{\text{LE}}(x, z) = 0
\end{aligned}
\tag{1}
$$

1.1 Restricting problem classes

The formulation (1) is very general. Must restrict the problem in order to solve efficiently, by assuming:

- LL problem is convex and regular (TODO: what is regular?),

- linear constraints, and
- linear or quadratic objective functions.

Then we can replace the LL problem with its KKT conditions. The Lagrangian function is

$$\mathcal{L}(x, z, \lambda, \mu) = f_{\mathrm{LL}}(x, z) + \lambda^T G_{\mathrm{LI}}(x, z) + \mu^T G_{\mathrm{LE}}(x, z) \tag{2}$$

and the KKT conditions are

$$
\begin{aligned}
\nabla_z \mathcal{L}(x, z, \lambda, \mu) &= 0 \\
G_{\mathrm{LI}}(x, z) &\leq 0 \\
G_{\mathrm{LE}}(x, z) &= 0 \\
\lambda &\geq 0 \\
\lambda \otimes G_{\mathrm{LI}}(x, z) &= 0
\end{aligned}
\tag{3}
$$

where $\otimes$ is the entrywise product. Line by line, the KKT conditions can be interpreted as

- the Lagrangian does not change,
- the inequality constraints hold,
- the equality constraints hold,
- the Lagrange multipliers are nonnegative,
- each inequality constraint is either active, or its Lagrange multiplier is zero.

The overall problem is then

$$
\begin{aligned}
\min_{x, z, \lambda, \mu} \quad & f_{\mathrm{UL}}(x, z) \\
& G_{\mathrm{UI}}(x, z) \leq 0 \\
& G_{\mathrm{UE}}(x, z) = 0 \\
& \nabla_z \mathcal{L}(x, z, \lambda, \mu) = 0 \\
& G_{\mathrm{LI}}(x, z) \leq 0 \\
& G_{\mathrm{LE}}(x, z) = 0 \\
& \lambda \geq 0 \\
& \lambda \otimes G_{\mathrm{LI}}(x, z) = 0
\end{aligned}
\tag{4}
$$

which is solvable for smaller problems, using e.g. YALMIP.

1.2 Big-M notation

Big-M formulation replaces nonlinear complementarity constraints with linear constraints using binary variables $s \in \{0, 1\}$ to indicate activeness for inequality

constraints:

$$G_{\text{LI}}(x, z) \leq 0$$
$$G_{\text{LI}}(x, z) \geq -M^u(1 - s)$$
$$\lambda \geq 0$$
$$\lambda \leq M^\lambda s$$

$$(5)$$

This notation fulfills the complementarity constraints, and given large enough M^u and M^λ, the solution is unchanged. When $s = 1$ we get $G_{\text{LI}}(x, z) = 0$ (inequality constraint active), and when $s = 0$ we get $\lambda = 0$ (inequality constraint inactive).

The overall problem formulation becomes

$$\min_{x, z, \lambda, \mu, s} f_{\text{UL}}(x, z)$$
$$G_{\text{UI}}(x, z) \leq 0$$
$$G_{\text{UE}}(x, z) = 0$$
$$\nabla_z \mathcal{L}(x, z, \lambda, \mu) = 0$$
$$G_{\text{LI}}(x, z) \leq 0$$
$$G_{\text{LE}}(x, z) = 0$$
$$\lambda \geq 0$$
$$G_{\text{LI}}(x, z) \geq -M^u(1 - s)$$
$$\lambda \leq M^\lambda s$$
$$s \in \{0, 1\}$$

$$(6)$$

Linear f_{UL} gives a MILP[1], quadratic f_{UL} gives a MIQP[2]. Nonconvex, *np*-hard, but efficient software exists.

1.3 Solving MILP/MIQP

Some steps can be taken to make MILP/MIQPs easier to solve.

1.3.1 Branch-and-bound

A branch-and-bound solver partitions the search space into regions, and finds an upper bound UB and lower bound LB for the solution in each region. If

$$LB_i > UB_j$$

$$(7)$$

then we know the solution is not in i, and we discard that region.

You can also remove known symmetries from symmetric problems by adding extra constraints to make only one of the symmetric solutions feasible.

---

[1]Mixed integer linear program.
[2]Mixed integer quadratic program.

1.3.2 Restrict combinations of binary variables

The binary variables indicate which constraints are active, and some combinations of active constraints are known to be impossible. Linear constraints on the binary variables can remove these combinations from the search space.

1.3.3 Use small M^u/M^λ

For a correct analytical solution, M^u and M^λ must be sufficiently large. However, setting them too large gives numerical issues and an inaccurate solution. Choosing them is simpler by setting them diagonal and positive.

A good M^u can sometimes be found by solving a series of LPs, if the LL constraints are bounded.

M^λ may need trial-and-error: If a value of λ is constrained by M^λ, retry after increasing the corresponding element of M^λ.

1.4 Approximate explicit MPC

Explicit MPC means to solve the optimization problems offline, so that the solution can be calculated easily and explicitly online. Bilevel optimization can be used for this.

- Find a set of points V whose convex hull describes the feasible region of the MPC.
- Solve MPC problem for each such point in the state space.
- Partition the convex hull of V into simplices with triangulation.
- Use linear interpolation in each simplex S_i to define a piecewise affine state feedback

$$u = K^i x + k^i, \quad x \in S_i \tag{8}$$

  which will be feasible and continuous.
- Further subdivide triangulation if $|u - u_{\text{optimal}}|$ too large or stability cannot be proven.
  - Can check difference from optimal input by solving a bilevel program for each simplex.
  - If the error is too large, create a new point where the error is greatest.
  - The bilevel program will be nonlinear ($f_{\text{UL}} = \max_x \|u_0 - u\|_\infty$), but can be linearized using big M.
- Can prove stability by showing that the value function is an LF for the closed loop system.

2 Linear matrix inequalities

Many problems in systems and control can be reduced to optimization problems involving LMIs, for which there exist efficient numerical solvers.

An LMI has the form

$$F(p) = F_0 + \sum_{i=1}^{m} p_i F_i > 0 \tag{9}$$

where $p \in \mathbb{R}^m$ is the variable, and $F_i = F_i^T \in \mathbb{R}^{n \times n}$, $i = 0, \ldots, m$. Multiple LMIs can be rewritten as a single LMI by stacking each matrix on the diagonal:

$$\text{diag}\left(F^{(1)}(x), \ldots, F^{(p)}(x)\right) > 0 \tag{10}$$

2.1 Some LMI problems

Feasibility problem   Is $\dot{x} = Ax$ stable? It is if there is a $P > 0$ for Lyapunov function $V(x) = x^T Px$ such that $A^T P + PA < 0$. This can be written

$$\begin{bmatrix} A^T P + PA & 0 \\ 0 & -P \end{bmatrix} < 0 \tag{11}$$

Convex function minimization   For the system

$$\begin{aligned} \dot{x} &= Ax + Bw \\ z &= Cx + Dw \end{aligned} \tag{12}$$

the H_∞-norm of the transfer function from w to z can be found by

$$\min \gamma$$
$$\begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \tag{13}$$

2.2 LMI tricks

2.2.1 Preliminaries

If W is full rank, then pre-multiplication with W^T and post-multiplication with W does not change sign-definiteness:

$$Q > 0 \Leftrightarrow W^T QW > 0 \quad \text{(for full rank } W) \tag{14}$$

5

2.2.2 Change of variables

Sometimes changing the variables can linearize the problem.

Example   State feedback: Find $u = Kx$ to stabilize the system. Must find $P > 0$ and K such that

$$(A + BK)^T P + P(A + BK) < 0. \tag{15}$$

If this is fullfilled, we have shown Lyapunov stability, but this is not linear in P and K. Like in (14), we can pre- and postmultiply by $Q = P^{-1}$ to get

$$QA^T + AQ + QK^T B^T + BKQ < 0 \tag{16}$$

and define $L = KQ$ to get (remember $Q = P^{-1}$ so $Q = Q^T$)

$$QA^T + AQ + L^T B^T + BL < 0. \tag{17}$$

This is an LMI in $Q > 0$ and L!

2.2.3 Congruence transform

With

$$Q = \begin{bmatrix} A^T P + PA & PBK + C^T V \\ * & -2V \end{bmatrix} < 0 \tag{18}$$

in variables $P > 0$, $V > 0$, K. Choosing the full-rank W

$$W = \begin{bmatrix} P^{-1} & 0 \\ 0 & V^{-1} \end{bmatrix} \tag{19}$$

gives

$$W^T Q W = \begin{bmatrix} XA^T + AX & BL + XC^T \\ * & -2U \end{bmatrix} \tag{20}$$

which is an LMI in $X = P^{-1}$, $U = V^{-1}$, $L = KV^{-1}$.

2.2.4 Schur complement

These are equivalent:

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{12}^T & \Phi_{22} \end{bmatrix} < 0 \Leftrightarrow \begin{cases} \Phi_{22} < 0 \\ \Phi_{11} - \Phi_{12}\Phi_{22}^{-1}\Phi_{12}^T < 0 \end{cases} \tag{21}$$

Example   Given $Q \geq 0$, $R > 0$, find $P > 0$ such that the Ricatti inequality

$$A^T P + PA + PBR^{-1}B^T P + Q < 0. \tag{22}$$

We can rearrange to reveal a similarity

$$\underbrace{A^T P + PA + Q}_{\Phi_{11}} - \underbrace{PB}_{\Phi_{12}} \underbrace{(-R^{-1})}_{\Phi_{22}^{-1}} \underbrace{B^T P}_{\Phi_{12}^T} < 0. \tag{23}$$

From the Schur complement and that $-R < 0$, this is equivalent to

$$\begin{bmatrix} A^T P + PA + Q & PB \\ * & -R \end{bmatrix} < 0. \tag{24}$$

2.2.5   S-procedure

Used when we need a criterion fulfilled locally, such as $F_0(x) \leq 0$ only when $F_i(x) > 0$, that is, the S-procedure gives a criterion for when one inequality is implied by another.

We want $F_0(x) \leq 0$ when $F_i(x) > 0$. This is true given

$$F_{\mathrm{aug}}(x) = F_0(x) + \sum_{i=1}^{q} \tau_i F_i(x) \leq 0, \quad \tau_i \geq 0 \tag{25}$$

Example   Find $P > 0$ such that

$$\begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} A^T P & PB \\ * & 0 \end{bmatrix}}_{F_0} \begin{bmatrix} x \\ z \end{bmatrix} < 0 \tag{26}$$

when

$$z^T z \leq x^T C^T C x \Leftrightarrow \begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} C^T C & 0 \\ * & -I \end{bmatrix}}_{F_1} \begin{bmatrix} x \\ z \end{bmatrix} \geq 0. \tag{27}$$

This can be transformed into an LMI by the S-procedure:

$$\begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} A^T P + PA + \tau C^T C & PB \\ * & -\tau I \end{bmatrix}}_{F_{\mathrm{aug}}} \begin{bmatrix} x \\ z \end{bmatrix} < 0. \tag{28}$$

This is an LMI in $P > 0$ and $\tau \geq 0$.

2.2.6 Projection lemma

Sometimes we get equations like

$$\Psi(X) + G(X)\Lambda^T H(X) + H(X)\Lambda^T G^T(X) < 0 \tag{29}$$

in matrix variables X and Λ. Ψ, G, and H are affine functions of X. Equation (29) is equivalent to

$$\begin{aligned}
W_{G(X)}^T \Psi(X) W_{G(X)} &< 0 \\
W_{H(X)}^T \Psi(X) W_{H(X)} &< 0
\end{aligned} \tag{30}$$

where $W_{G(X)}$ and $W_{H(X)}$ are orthogonal complements to G and H:

$$W_{G(X)} G(X) = 0, \quad W_{H(X)} H(X) = 0 \tag{31}$$

$$\operatorname{rank} W_{G(X)} + \operatorname{rank} G = n, \quad \operatorname{rank} W_{H(X)} + \operatorname{rank} H = n. \tag{32}$$

This is useful as may linearize the formulation, and it may reduce the number of variables in the computation.

Example   State feedback:

$$(A + BK)^T P + P(A + BK) < 0, \quad P > 0 \tag{33}$$

can be rewritten by a change of variables (Section 2.2.2) as

$$QA^T + AQ + L^T B^T + BL < 0, \quad Q > 0. \tag{34}$$

Using the projection lemma this can be written without L:

$$\left. \begin{aligned}
W_B^T (AQ + QA^T) W_B &< 0 \\
W_I^T (AQ + QA^T) W_I &< 0
\end{aligned} \right\} Q > 0 \tag{35}$$

In this case the second inequality is unnecessary, because I is full rank. First inequality can be interpreted as "the system is stable if it is stable in the null space of the inputs."

2.2.7 Finsler's lemma

Finsler's lemma is another way of writing (29) more compactly:

$$\begin{aligned}
\Psi(X) - \sigma G(X) G(X)^T &< 0 \\
\Psi(X) - \sigma H(X) H(X)^T &< 0
\end{aligned} \tag{36}$$

for some real σ.

Example State feedback:

$$(A + BK)^T P + P(A + BK) < 0, \quad P > 0 \tag{37}$$

can be rewritten

$$QA^T + AQ + L^T B^T + BL < 0, \quad Q > 0. \tag{38}$$

Using Finsler's lemma this can be written without L

$$\left.\begin{array}{c} AQ + QA^T - \sigma BB^T < 0 \\ AQ + QA^T - \sigma I < 0 \end{array}\right\} Q > 0. \tag{39}$$

As long as Q and σ fulfilling first inequality are found, it is always possible to find σ to fulfill the second.

2.3 Applications

2.3.1 Discrete-time state feedback stabilization

Discrete systems use Lyapunov differences. For the LF

$$V(x_k) = x_k^T P x_k, \quad P > 0 \tag{40}$$

the system is stable if

$$V(x_k) - V(x_{k+1}) = x_k^T P x_k - x_{k+1}^T P x_{k+1} > 0, \tag{41}$$

which is equivalent to

$$P - (A + BK)^T P(A + BK) > 0. \tag{42}$$

The Schur complement is

$$\begin{bmatrix} P & (A + BK)^T P \\ P(A + BK) & P \end{bmatrix} > 0, \tag{43}$$

and congruence transform with $W = \begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix}$ gives

$$\begin{bmatrix} P^{-1} & P^{-1}(A + BK)^T \\ (A + BK)P^{-1} & P^{-1} \end{bmatrix} > 0. \tag{44}$$

A change of variables $Q = P^{-1}, L = KP^{-1}$ gives

$$\begin{bmatrix} Q & QA^T + L^T B^T \\ AQ + BL & Q \end{bmatrix} > 0 \tag{45}$$

which is an LMI in $Q > 0$ and L.

2.3.2 Robust feedback stabilization

Plant not exactly known, but known to be in a set

$$\begin{bmatrix} A \\ B \end{bmatrix} \in \text{Co} \begin{bmatrix} A_1 \dots A_q \\ B_1 \dots B_q \end{bmatrix} \tag{46}$$

where Co denotes convex combination. Equivalent:

$$A = \sum_{i=1}^{q} a_i A_i, \quad B = \sum_{i=1}^{q} a_i B_i \tag{47}$$

with

$$\sum_{i=1}^{q} a_i = 1, a_i \geq 0. \tag{48}$$

A controller that solves the LMIs for all q extreme plants with the same Q and L is stable for all A and B.

2.3.3 Parameter dependent LFs

With a model

$$A = A_0 + \sum_{i=1}^{q} \delta_i A_i \tag{49}$$

with bounds and rate limits

$$\delta_i \in [\underline{\delta}_i, \overline{\delta}_i], \quad \dot{\delta}_i \in [\underline{\lambda}_i, \overline{\lambda}_i] \tag{50}$$

The set Δ is the set of all δ within its bounds, and Λ is the set of all $\dot{\delta}$ within its rate limits. Δ_0 and Λ_0 are their respective vertex sets.

Some LFs have the form

$$V(\delta, x) = x^T P(\delta)x, \quad P(\delta) = P_0 + \sum_{i=1}^{q} \delta_i P_i. \tag{51}$$

which gives

$$\frac{dP(\delta)}{dt} = \dot{\delta}_1 P_1 + \cdots + \dot{\delta}_q P_q = P(\dot{\delta}) - P_0 \tag{52}$$

leading to

$$\dot{V}(\delta, x) = x^T \left(A^T(\delta)P(\delta) + P(\delta)A(\delta) + P(\dot{\delta}) - P_0 \right) x. \tag{53}$$

Which is stable if there exists matrices $P_0 \dots P_q$ such that

$$\begin{aligned} A^T(\delta)P(\delta) + P(\delta)A(\delta) + P(\dot{\delta}) &< P_0, \quad \forall \delta \in \Delta_0, \forall \dot{\delta} \in \Lambda_0 \\ P(\delta) &> I, \quad \forall \delta \in \Delta_0 \\ A_i^T + P_i A_i &\leq 0, \quad i = 1, \dots, q. \end{aligned} \tag{54}$$

This means that we only have to find valid LFs for the "extreme points", i.e. when δ or $\dot{\delta}$ is at saturation.

3   Linear matrix inequalities for piecewise affine systems

3.1   Piecewise affine systems

A piecewise affine system is a system where the state space X is divided into non-overlapping partitions X_i with distinct models in each partition:
- Partitions containing the origin are linear:

$$x = A_i x$$
$$x_{k+1} = A_i x_k \tag{55}$$

- Partitions not containing the origin are affine:

$$x = A_i x + a_i$$
$$x_{k+1} = A_i x_k + a_i \tag{56}$$

The partitions are indexed, with an index set

$$I = I_0 \cup I_1 \tag{57}$$

where I_0 are the pariticions containing the origin, and I_1 are the partitions not containing the origin.

3.2   Lyapunov stability

Sometimes you can find a Lyapunov function for the whole PWA system:
- If $a_i = 0 \forall i$ and $\exists P = P^T > 0$ such that $A_i^T P + P A_i < 0 \forall i \in I$, then the origin is exponentially stable.
- if $a_i = 0 \forall i$ and $\exists R_i > 0 \forall i \in I$ such that $\sum_{i \in I}(A_i^T R_i + R_i A_i) > 0$, then a common Lyapunov function can be found.

When a common function cannot be found, we must look for one that depends on the partition.

3.2.1   Notation

$$\bar{A}_i = \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix} \tag{58}$$

11

Each partition is a polyhedron, so we can write

$$\bar{E}_i = \begin{bmatrix} E_i & e_i \end{bmatrix}, \quad \bar{F}_i = \begin{bmatrix} F_i & f_i \end{bmatrix} \tag{59}$$

where $e_i = 0$ and $f_i = 0$ for all $i \in I_0$ such that

$$\bar{E}_i \begin{bmatrix} x \\ 1 \end{bmatrix} \geq 0 \forall x \in X_i \forall i \in I \tag{60}$$

$$\bar{F}_i \begin{bmatrix} x \\ 1 \end{bmatrix} = \bar{F}_j \begin{bmatrix} x \\ 1 \end{bmatrix} \forall \; x \in X_i \cap X_j, \forall i, j \in I \tag{61}$$

That is, $E_i x + e_i \geq 0$ for all values of x, and $F_i x + f_i = F_j x + f_j$ for all x along borders between partitions.

3.2.2 Lyapunov function

$$V(x) = \begin{cases} x^T P_i x, & x \in X_i, i \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{P}_i \begin{bmatrix} x \\ 1 \end{bmatrix}, & x \in X_i, i \in I_1 \end{cases} \tag{62}$$

where

$$P_i = F_i^T T F_i, \quad \bar{P}_i = \bar{F}_i^T T \bar{F}_i \tag{63}$$

and T is some symmetric matrix.

3.2.3 Relaxing the LF

While $V(x) > 0$ is required for all x, we only need $P_i > 0$ for $x \in X_i$. The S-procedure can be used to alter the LF to reflect this:

$$V(x) = \begin{cases} x^T E_i^T U_i E_i x, & x \in X_i, i \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{E}_i^T U_i \bar{E}_i^T \begin{bmatrix} x \\ 1 \end{bmatrix}, & x \in X_i, i \in I_1 \end{cases} \tag{64}$$

where U_i has only non-negative elements. The condition $\dot{V}(x) < 0$ only needs to hold within each partition, as well.

3.2.4 Overall LF design

Find symmetric matrices T, U_i, W_i (U_i and W_i with no negative elements), such that

$$\left.\begin{array}{l} A_i^T P_i + P_i A_i + E_i^T U_i E_i < 0 \\ P_i - E_i^T W_i E_i > 0 \end{array}\right\} i \in I_0$$

$$\left.\begin{array}{l} \bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i + \bar{E}_i^T U_i \bar{E}_i < 0 \\ \bar{P}_i - \bar{E}_i^T W_i \bar{E}_i > 0 \end{array}\right\} i \in I_1 \tag{65}$$

with (63) still holding:

$$P_i = F_i^T T F_i, \quad \bar{P}_i = \bar{F}_i^T T \bar{F}_i \tag{66}$$

$\bar{E}_i$ and $\bar{F}_i$ can be determined e.g. with explicit formulas found in papers by Johansson and Ratzer.

3.3 Discrete-time Lyapunov stability

With discrete systems
- we don't need a continuous LF,
- we use LF difference rather than LF differential,
- we must keep track of possible next-timestep partitions.

If there exists matrices $R_i > 0$ such that

$$\sum_{i \in I} (A_i^T R_i A_i - R_I) > 0 \tag{67}$$

then no common LF can be found for the PWL system.

3.3.1 Stability criteria

$f(X_l)$ is a quadratic function positive on X_l, such as

$$f(X_i) = \begin{cases} x^T E_i^T U_i E_i x, & x \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{E}_i^T U_i \bar{E}_i \begin{bmatrix} x \\ 1 \end{bmatrix}, & x \in I_1 \end{cases} \tag{68}$$

Rewrite the system

$$\bar{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}, \quad \bar{x}_{k+1} = \bar{A}_i \bar{x}_k \tag{69}$$

where

$$\bar{A}_i = \begin{cases} \begin{bmatrix} A_i & a_i \\ 0 & 1 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_1, x_{k+1} \in X_j, j \in I_1}_{x \text{ moving between non-origin partitions}} \\ \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_1, x_{k+1} \in X_j, j \in I_0}_{x \text{ moving from non-origin to origin partition}} \\ \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_0, x_{k+1} \in X_j, j \in I_0}_{x \text{ moving between origin-partitions}}. \end{cases} \tag{70}$$

then the system is stable if

$$P_i - f(X_i) > 0, \quad \forall i$$
$$P_i - \bar{A}_i^T P_j \bar{A}_i - f(X_{ij}) > 0, \quad \forall(i,j) \text{ with nonempty } X_{ij} \tag{71}$$

where $X_{ij} \subset X_i$ is the region where the state in the next timestep will move to X_j.

3.3.2 Relaxations

R is a bounded polyhedron and $V(R)$ is its vertice set. Then this function is quadratic and positive over R:

$$f(R) = \begin{bmatrix} x \\ 1 \end{bmatrix}^T H \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{72}$$

where

$$H = \bar{H} + C$$
$$\bar{H} < 0$$
$$C = \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix} \tag{73}$$
$$c > 0$$
$$\begin{bmatrix} v_i \\ 1 \end{bmatrix}^T (\bar{H} + C) \begin{bmatrix} v_i \\ 1 \end{bmatrix} > 0, \quad \forall v_i \in V(R)$$

4 Sum of squares programming

SoS decomposition can prove non-negativity of polynomials, and can be used for control analysis and design for polynomial nonlinear systems.

4.1 Polynomials and monomials

A (multivariate) polynomial is

$$f(x) = \sum_k c_k \underbrace{x_1^{a_{k1}} \ldots x_n^{a_{kn}}}_{\text{a monomial}}, \quad a_{ki} \in \mathbb{Z}, \tag{74}$$

and a monomial is one term in a polynomial, without the coefficient:

$$m_k(x) = x_1^{a_{k1}} \ldots x_n^{a_{kn}}. \tag{75}$$

4.2 Sum of squares decomposition

A *polynomial* $f(x)$ is a SoS if it can be written as a sum of squares (lol):

$$f(x) = \sum_{i=1}^N h_i^2(x) = \sum_{i=1}^N \left(q_i^T v(x) \right)^2 = v^T(x) Q v(x) \tag{76}$$

where $v(x)$ is a vector of monomials, and $Q \geq 0$.

A symmetric *polynomial matrix $M(x)$* is an SoS matrix if it can be written

$$M(x) = H^T(x) H(x) \tag{77}$$

for some polynomial matrix $H(x)$.

Note: There exists nonnegative polynomials that have to SoS decomposition. SoS decompositions are only guaranteed for nonnegative polynomials

- in two variables,
- in quadratic forms, or
- in three variables and of fourth order.

Note 2: A polynomial may have more than one SoS decomposition, such as these:

$$\underbrace{2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4}_{f(x)} = \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}}_{v(x)^T}^T \underbrace{\begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix}}_{Q_1} \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}}_{v(x)} \tag{78}$$

and

$$\underbrace{2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4}_{f(x)} = \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}}_{v(x)^T}^T \underbrace{\begin{bmatrix} 2 & -\lambda & 1 \\ -\lambda & 5 & 0 \\ 1 & 0 & 2\lambda - 1 \end{bmatrix}}_{Q_2} \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}}_{v(x)} \tag{79}$$

We have $\det(Q_1) = -15$, and therefore $Q_1 \not\geq 0$. Therefore, Q_1 does not imply that $f(x)$ is a SoS polynomial. However, we can find a λ in Q_2 that gives $Q_2 \geq 0$, such as $\lambda = 1 \Rightarrow \det(Q_2) = 4 \Rightarrow Q_2 \geq 0$. Thus $f(x)$ is a SoS polynomial after all.

It turns out that in general the set of parameters that make a symmetric matrix $Q \geq 0$ is convex when it depends linearly on its parameters, and thus SoS constraints are convex.

4.2.1 Numerical issues

Sometimes Q is very ill-conditioned, with tiny eigenvalues. Can be improved by
- removing unused elements of $v(x)$
- making Q block-diagonal,
- altering Q based on an initial solution,
- using software that pre- and post-processes automatically.

4.2.2 Proof of positivity

Primal SoS optimization always yields a P.D. Q, but the solution may be so inaccurate that the polynomial is still not SoS. With

$$\lambda_{\min}(Q) = \min \operatorname{eig}(Q) \tag{80}$$

$$v(x) \in \mathbb{R}^M \tag{81}$$

$$r = f(x) - v^T(x)Qv(x) \tag{82}$$

we can guarantee positivity of $f(x)$ given

$$\begin{aligned} \lambda_{\min}(Q) &\geq 0 \\ \lambda_{\min}(Q) &\geq M \cdot \operatorname{abs}(r) \end{aligned} \tag{83}$$

Sometimes, a dual solver will find a valid solution that the primal does not find.

4.3 LMIs and SoS

SoS problems are a special case of LMIs, but only some LMI "tricks" can be applied to SoSs:

4.3.1 Linearizing change of variables

If a problem depends linearly on $P > 0$ and PK (no standalone Ks), then we can use $L = PK$ to make the problem linear in L and P.

4.3.2   Congruence transform

For matrix SoSs with $M(x) \geq 0$ and a full rank $W(x)$:

$$M(x) \geq 0 \Leftrightarrow W^T(x)M(x)W(x) \geq 0 \tag{84}$$

Note: Cannot be used with scalarized Schur complement.

4.3.3   Scalarization

For an LMI, positive definiteness has the relation

$$L > 0 \Leftrightarrow x^T L x > 0 \quad \forall x \tag{85}$$

while for an SoS it is instead

$$M(x) > 0 \Leftrightarrow z^T M(x) z > 0 \quad \forall x, \forall z. \tag{86}$$

As it turns out, it's better to formulate scalar-valued SoS problems.

4.3.4   Scalarized Schur complement

With
$$M(x) = \begin{bmatrix} E(x) & F^T(x) \\ F(x) & P(x) \end{bmatrix}, \quad P(x) \text{ symmetric and invertible} \tag{87}$$

then
$$\begin{bmatrix} x \\ z \end{bmatrix}^T M(x) \begin{bmatrix} x \\ z \end{bmatrix} > 0, \quad \forall \{x, z\} \neq \{0, 0\} \tag{88}$$

is equivalent to

$$
\begin{aligned}
x^T \left(E - F^T P^{-1} F \right) x > 0, & \quad \forall x \neq 0 \\
z^T P z > 0, & \quad \forall z \neq 0, \forall x
\end{aligned}
\tag{89}
$$

because of

$$M(x) = \begin{bmatrix} I & F^T P^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E - F^T P^{-1} F & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} I & 0 \\ P^{-1} F & I \end{bmatrix}. \tag{90}$$

If we have
$$\begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} I & 0 \\ P^{-1}(x)F(x) & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \tag{91}$$

then w can take any value regardless of x, by choosing z, and z can take any value by choosing w.

4.3.5   S-procedure

We want to prove $f(x) > 0$ when $g(x) < 0$. Can be written

$$f(x) + s(x)g(x) > 0 \tag{92}$$

for a arbitrary SoS polynomial $s(x)$. With this formulation, optimizing over parameters in $s(x)$ and $g(x)$ gives a bilinear problem.

4.4   Bilinear discrete systems

The bilinear[3] system

$$x_{k+1} = Ax_k + \sum_{i=1}^{m}(B_i x_k + b_i)u_{i,k} = Ax_k + (B_x + B)u_k \tag{93}$$

is stable under state feedback if

$$x_k^T P x_k - x_{k+1}^T P x_{k+1} > 0 \tag{94}$$

$$x_k^T P x_k - \left(Ax_k + (B_x + B)u_k(x_k)\right)^T P\left(Ax_k + (B_x + B)u_k(x_k)\right) > 0 \tag{95}$$

If the system is open-loop unstable, $u_k(x_k)$ must be a ratio of same-order polynomials to achieve global quadratic stability, such as:

$$u_k(x_k) = \frac{C(x_k)x_k}{c_0(x_k) + 1} \tag{96}$$

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

4.4.1   Region of convergence

With a quadratic LF $V(x_k) = x_k^T P x_k$, a polynomial matrix $C(x_k)$, and SoS polynomials $c_0(x_k)$, $s_1(x_k, z)$, then the closed loop system is stable for all $x_k | x_k^T P x_k < \gamma$ given

$$\begin{bmatrix} x_k \\ z \end{bmatrix}^T M(x) \begin{bmatrix} x_k \\ z \end{bmatrix} - s_1(x_k, z)(\gamma - x_k^T P x_k) > 0 \tag{97}$$

where (omitting arguments for brevity)

$$M(x_k) = \begin{bmatrix} (c_0 + 1)P & \left((c_0 + 1)A + (B_x + B)C\right)^T P \\ P\left((c_0 + 1)A + (B_x + B)C\right) & (c_0 + 1)P \end{bmatrix} \tag{98}$$

which comes from $(c_0 + 1)$ being strictly positive, and using the scalarized Schur complement and the S-procedure.

---

[3]Bilinear because the next state is a function with a term that is a product of the state and the input. In a linear system, the B_x term would not be present.

4.4.2 Input saturation

The input constraints

$$-u_{i,\max} \le u_i \le u_{i,\max} \tag{99}$$

are satisfied for all $x_k | x_k^T P x_k < \gamma$ given

$$\begin{bmatrix} \left(c_0(x_k) + 1\right)u_{i,\max}^2 - q_i(x_k)(\gamma - x_k^T P x_k) & c_i(x_k) \\ c_i(x_k) & c_0(x_k) + 1 \end{bmatrix} > 0 \tag{100}$$

where $c_i(x_k)$ are rows of $C(x_k)$, and $c_0(x_k)$ and $q_i(x_k)$ are SoS polynomials.

4.4.3 Rate of convergence

We can change $M(x)$ from (98) to

$$M(x_k) = \begin{bmatrix} (1 - \alpha)(c_0 + 1)P & \left((c_0 + 1)A + (B_x + B)C\right)^T P \\ P\left((c_0 + 1)A + (B_x + B)C\right) & (c_0 + 1)P \end{bmatrix} \tag{101}$$

to guarantee an exponential convergence defined by α. There will be a tradeoff between maximizing the region of convergence and rate of convergence. Using this $M(x)$ allows us to determine the tradeoff.

4.4.4 Optimization formulation

1. If α, γ, and P are known, then C, c_0, q_i, s_1 enter linearly into inequalities above, and they can be solved as is.
2. If γ and P are known, we can find C, c_0, q_i, s_1 by formulating a feasibility problem (optimization without objective).
3. If C, c_0, q_i, and s_1 are known, we can maximize γ with P as a free variable. Must normalize P, i.e. by a constraint trace$(P) = k$, for some constant k.
4. Can iterate between the two points directly above to create a controller with gradually larger region of convergence.