Summary of TTK18

Morten Fyhn Amundsen

November 24, 2016

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.

$$\begin{aligned} \min_x \; & f_{\mathrm{UL}}(x, z) \\ & G_{\mathrm{UI}}(x, z) \leq 0 \\ & G_{\mathrm{UE}}(x, z) = 0 \\ z = \arg\min_z \; & f_{\mathrm{LL}}(x, z) \\ & G_{\mathrm{LI}}(x, z) \leq 0 \\ & G_{\mathrm{LE}}(x, z) = 0 \end{aligned} \tag{1}$$

1.1 Restricting problem classes

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

- Assume LL problem is convex and regular.
- Assume linear constraints.
- Assume 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

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

The overall problem is then

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

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:

$$\begin{aligned} G_{\mathrm{LI}}(x, z) &\leq 0 \\ G_{\mathrm{LI}}(x, z) &\geq -M^u(1 - s) \\ \lambda &\geq 0 \\ \lambda &\leq M^\lambda s \end{aligned} \tag{5}$$

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

The overall problem formulation becomes

$$\begin{aligned}
\min_{x,z,\lambda,\mu,s} \; & f_{\mathrm{UL}}(x,z) \\
G_{\mathrm{UI}}(x,z) &\leq 0 \\
G_{\mathrm{UE}}(x,z) &= 0 \\
\nabla_z \mathcal{L}(x,z,\lambda,\mu) &= 0 \\
G_{\mathrm{LI}}(x,z) &\leq 0 \\
G_{\mathrm{LE}}(x,z) &= 0 \\
\lambda &\geq 0 \\
G_{\mathrm{LI}}(x,z) &\geq -M^u(1-s) \\
\lambda &\leq M^\lambda s \\
s &\in \{0,1\}
\end{aligned} \tag{6}$$

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

$$LB_i > UB_j \tag{7}$$

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.

1.3.2 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] Mixed integer linear program.

[2] Mixed integer quadratic program.

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.

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

$$F(p) = F_0 + \sum_{i=1}^{m} p_i F_i > 0 \tag{8}$$

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:

$$\operatorname{diag}\left(F^{(1)}(x), \ldots, F^{(p)}(x)\right) > 0 \tag{9}$$

2.1 Some LMI problems

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

$$\begin{bmatrix} A^T P + PA & 0 \\ 0 & -P \end{bmatrix} < 0 \tag{10}$$

Convex function minimization For the system

$$\dot{x} = Ax + Bwz = Cx + Dw \tag{11}$$

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

$$\begin{gathered} \min \gamma \\ \begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \end{gathered} \tag{12}$$

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:

$$Q > 0 \Leftrightarrow W^T Q W > 0 \quad \text{(for full rank } W\text{)} \tag{13}$$

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

$$(A + BK)^T + P(A + BK) < 0. \tag{14}$$

If this is fullfilled, we have shown Lyapunov stability, but this is not linear in P and K. Like in (13), we can pre- and postmultiply by $Q = P^{-1}$ to get

$$QA^T + AQ + QK^T B^T + BKQ < 0 \tag{15}$$

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

$$QA^T + AQ + L^T B^T + BL < 0. \tag{16}$$

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

2.2.3 Congruence transform

With

$$Q = \begin{bmatrix} A^T P + PA & PBK + C^T V \\ * & -2V \end{bmatrix} < 0 \tag{17}$$

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

$$W = \begin{bmatrix} P^{-1} & 0 \\ 0 & V^{-1} \end{bmatrix} \tag{18}$$

gives

$$W^T Q W = \begin{bmatrix} XA^T + AX & BL + XC^T \\ * & -2U \end{bmatrix} \tag{19}$$

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

2.2.4 Schur complement

These are equivalent:

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{12}^T & \Phi_{22} \end{bmatrix} < 0 \Leftrightarrow \begin{cases} \Phi_{22} < 0 \\ \Phi_{11} - \Phi_{12}\Phi_{22}^{-1}\Phi_{12}^T < 0 \end{cases} \tag{20}$$

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

$$A^T P + PA + PBR^{-1}B^T P + Q < 0. \tag{21}$$

We can rearrange to reveal a similarity

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

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

$$\begin{bmatrix} A^T P + PA + Q & PB \\ * & -R \end{bmatrix} < 0. \tag{23}$$

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

$$F_{\text{aug}}(x) = F_0(x) + \sum_{i=1}^{q} \tau_i F_i(x) \leq 0, \quad \tau_i \geq 0 \tag{24}$$

Example Find $P > 0$ such that

$$\begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} A^T P & PB \\ * & 0 \end{bmatrix}}_{F_0} \begin{bmatrix} x \\ z \end{bmatrix} < 0 \tag{25}$$

when

$$z^T z \leq x^T C^T C x \Leftrightarrow \begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} C^T C & 0 \\ * & -I \end{bmatrix}}_{F_1} \begin{bmatrix} x \\ z \end{bmatrix} \geq 0. \tag{26}$$

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

$$\begin{bmatrix} x \\ z \end{bmatrix}^T \underbrace{\begin{bmatrix} A^T P + PA + \tau C^T C & PB \\ * & -\tau I \end{bmatrix}}_{F_{\text{aug}}} \begin{bmatrix} x \\ z \end{bmatrix} < 0. \tag{27}$$

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

2.2.6 More tricks

- Projection lemma
- Finsler's lemma

Both can be used to make problems smaller.

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:

$$\begin{aligned} x &= A_i x \\ x_{k+1} &= A_i x_k \end{aligned} \tag{28}$$

- Partitions not containing the origin are affine:

$$\begin{aligned} x &= A_i x + a_i \\ x_{k+1} &= A_i x_k + a_i \end{aligned} \tag{29}$$

The partitions are indexed, with an index set

$$I = I_0 \cup I_1 \tag{30}$$

where I_0 are the parititions 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

$$\bar{A}_i = \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix} \tag{31}$$

Each partition is a polyhedron, so we can write

$$\bar{E}_i = \begin{bmatrix} E_i & e_i \end{bmatrix}, \quad \bar{F}_i = \begin{bmatrix} F_i & f_i \end{bmatrix} \tag{32}$$

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

$$\bar{E}_i \begin{bmatrix} x \\ 1 \end{bmatrix} \geq 0 \forall x \in X_i \forall i \in I \tag{33}$$

$$\bar{F}_i \begin{bmatrix} x \\ 1 \end{bmatrix} = \bar{F}_j \begin{bmatrix} x \\ 1 \end{bmatrix} \forall\, x \in X_i \cap X_j, \forall i, j \in I \tag{34}$$

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

$$V(x) = \begin{cases} x^T P_i x & x \in X_i, i \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{P}_i \begin{bmatrix} x \\ 1 \end{bmatrix} & x \in X_i, i \in I_1 \end{cases} \tag{35}$$

where[3]

$$P_i = F_i^T T F_i, \quad \bar{P}_i = \bar{F}_i^T T \bar{F}_i \tag{36}$$

[3] TODO: wtf is the T 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[4]:

$$V(x) = \begin{cases} x^T E_i^T U_i E_i x & x \in X_i, i \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{E}_i^T U_i \bar{E}_i^T \begin{bmatrix} x \\ 1 \end{bmatrix} & x \in X_i, i \in I_1 \end{cases} \tag{37}$$

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

$$\begin{cases} A_i^T P_i + P_i A_i + E_i^T U_i E_i < 0 \\ P_i - E_i^T W_i E_i > 0 \end{cases} \quad i \in I_0$$
$$\begin{cases} \bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i + \bar{E}_i^T U_i \bar{E}_i < 0 \\ \bar{P}_i - \bar{E}_i^T W_i \bar{E}_i > 0 \end{cases} \quad i \in I_1 \tag{38}$$

with (36) still holding:

$$P_i = F_i^T T F_i, \quad \bar{P}_i = \bar{F}_i^T T \bar{F}_i \tag{39}$$

$\bar{E}_i$ 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

$$\sum_{i \in I} (A_i^T R_i A_i - R_I) > 0 \tag{40}$$

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

[4]TODO: Double check that this actually is the Lyapunov function.

3.3.1 Stability criteria

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

$$f(X_i) = \begin{cases} x^T E_i^T U_i E_i x, & x \in I_0 \\ \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right]^T \bar{E}_i^T U_i \bar{E}_i \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right], & x \in I_1 \end{cases} \tag{41}$$

Rewrite the system

$$\bar{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}, \quad \bar{x}_{k+1} = \bar{A}_i \bar{x}_k \tag{42}$$

where

$$\bar{A}_i = \begin{cases} \begin{bmatrix} A_i & a_i \\ 0 & 1 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_1, x_{k+1} \in X_j, j \in I_1}_{x \text{ moving between non-origin partitions}} \\ \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_1, x_{k+1} \in X_j, j \in I_0}_{x \text{ moving from non-origin to origin partition}} \\ \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, & \underbrace{x_k \in X_i, i \in I_0, x_{k+1} \in X_j, j \in I_0}_{x \text{ moving between origin-partitions}}. \end{cases} \tag{43}$$

then the system is stable if

$$\begin{aligned} P_i - f(X_i) > 0, \quad & \forall i \\ P_i - \bar{A}_i^T P_j \bar{A}_i - f(X_{ij}) > 0, \quad & \forall (i,j) \text{ with nonempty } X_{ij} \end{aligned} \tag{44}$$

where X_{ij} is the subset of X_i in which the next state will be in 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:

$$f(R) = \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right]^T H \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix}\right] \tag{45}$$

where

$$\begin{aligned} H &= \bar{H} + C \\ \bar{H} &< 0 \\ C &= \left[\begin{smallmatrix} 0 & 0 \\ 0 & c \end{smallmatrix}\right] \\ c &> 0 \\ \left[\begin{smallmatrix} v_i \\ 1 \end{smallmatrix}\right]^T (\bar{H} + C) \left[\begin{smallmatrix} v_i \\ 1 \end{smallmatrix}\right] &> 0, \quad \forall v_i \in V(R) \end{aligned} \tag{46}$$

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

$$f(x) = \sum_k c_k x_1^{a_{k1}} \dots x_n^{a_{kn}}, \quad a_{ki} \in \mathbb{Z}, \tag{47}$$

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

$$m_k(x) = x_1^{a_{k1}} \dots x_n^{a_{kn}}. \tag{48}$$

4.2 Sum of squares decomposition

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

$$f(x) = \sum_{i=1}^{N} h_i^2(x) = \sum_{i=1}^{N} \left(q_i^T v(x)\right)^2 = v^T(x) Q v(x) \tag{49}$$

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

$$M(x) = H^T(x) H(x) \tag{50}$$

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:

$$\underbrace{2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4}_{f(x)} = \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}^T}_{v(x)^T} \underbrace{\begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix}}_{Q_1} \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}}_{v(x)} \tag{51}$$

and

$$\underbrace{2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4}_{f(x)} = \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T}_{v(x)^T} \underbrace{\begin{bmatrix} 2 & -\lambda & 1 \\ -\lambda & 5 & 0 \\ 1 & 0 & 2\lambda - 1 \end{bmatrix}}_{Q_2} \underbrace{\begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}}_{v(x)} \tag{52}$$

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

It turns out that in general the set of parameters that make a symmetric matrix $Q \succeq 0$ 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

$$\lambda_{\min}(Q) = \min \operatorname{eig}(Q) \tag{53}$$

$$v(x) \in \mathbb{R}^M \tag{54}$$

$$r = f(x) - v^T(x)Qv(x) \tag{55}$$

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

$$\begin{aligned} \lambda_{\min}(Q) &\geq 0 \\ \lambda_{\min}(Q) &\geq M \cdot \operatorname{abs}(r) \end{aligned} \tag{56}$$

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 $W(x)$:

$$M(x) \geq 0 \Leftrightarrow W^T(x)M(x)W(x) \geq 0 \tag{57}$$

Note: Cannot be used with scalarized Schur complement.

4.3.3 Scalarization

For an LMI:

$$L > 0 \Leftrightarrow x^T Lx > 0 \quad \forall x \tag{58}$$

For an SoS:

$$M(x) > 0 \not\Leftrightarrow x^T M(x)x > 0 \tag{59}$$

Instead:

$$M(x) > 0 \Leftrightarrow z^T M(x)z > 0 \quad \forall x, \forall z \tag{60}$$

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

4.3.4 Scalarized Schur complement

With

$$M(x) = \begin{bmatrix} E(x) & F^T(x) \\ F(x) & P(x) \end{bmatrix}, \quad P(x) \text{ symmetric and invertible} \tag{61}$$

then

$$\left[\begin{smallmatrix} x \\ z \end{smallmatrix}\right]^T M(x) \left[\begin{smallmatrix} x \\ z \end{smallmatrix}\right] > 0, \quad \forall \{x, z\} \neq \{0, 0\} \tag{62}$$

is equivalent to

$$\begin{aligned} x^T \left(E - F^T P^{-1} F\right) x > 0, \quad & \forall x \neq 0 \\ z^T P z > 0, \quad & \forall z \neq 0, \forall x \end{aligned} \tag{63}$$

because of

$$M(x) = \begin{bmatrix} I & F^T P^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E - F^T P^{-1} F & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} I & 0 \\ P^{-1} F & I \end{bmatrix}. \tag{64}$$

If we have

$$\begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} I & 0 \\ P^{-1}(x)F(x) & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \tag{65}$$

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

$$f(x) + s(x)g(x) > 0 \tag{66}$$

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[5] system

$$x_{k+1} = Ax_k + \sum_{i=1}^{m} (B_i x_k + b_i) u_{i,k} = Ax_k + (B_x + B)u_k \tag{67}$$

is stable under state feedback if

$$x_k^T P x_k - x_{k+1}^T P x_{k+1} > 0 \tag{68}$$

$$x_k^T P x_k - \Big(Ax_k + (B_k + B)u_k(x_k)\Big)^T P \Big(Ax_k + (B_k + B)u_k(x_k)\Big) > 0 \tag{69}$$

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

$$u_k(x_k) = \frac{C(x_k)x_k}{c_0(x_k) + 1} \tag{70}$$

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

[5]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.

4.4.1 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

$$\begin{bmatrix} x_k \\ z \end{bmatrix}^T M(x) \begin{bmatrix} x_k \\ z \end{bmatrix} - s_1(x_k, z)(\gamma - x_k^T P x_k) > 0 \tag{71}$$

where (omitting arguments for brevity)

$$M(x_k) = \begin{bmatrix} (c_0 + 1)P & \Big((c_0 + 1)A + (B_x + B)C\Big)^T P \\ P\Big((c_0 + 1)A + (B_x + B)C\Big) & (c_0 + 1)P \end{bmatrix} \tag{72}$$

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

4.4.2 Input saturation

The input constraints

$$-u_{i,\max} \leq u_i \leq u_{i,\max} \tag{73}$$

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

$$\begin{bmatrix} \Big(c_0(x_k) + 1\Big)u_{i,\max}^2 - q_i(x_k)(\gamma - x_k^T P x_k) & c_i(x_k) \\ c_i(x_k) & c_0(x_k) + 1 \end{bmatrix} > 0 \tag{74}$$

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 (72) to

$$M(x_k) = \begin{bmatrix} (1 - \alpha)(c_0 + 1)P & \Big((c_0 + 1)A + (B_x + B)C\Big)^T P \\ P\Big((c_0 + 1)A + (B_x + B)C\Big) & (c_0 + 1)P \end{bmatrix} \tag{75}$$

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 $\text{trace}(P) = k$, for some constant k.

4. Can iterate between the two points directly above to create a controller with gradually larger region of convergence.