Semidefinite Programming and its Application to Process Control

Venkataramanan (Ragu) Balakrishnan
School of ECE, Purdue University

IIT Chennai, January 2003

General motivation

- Dramatic and continuing growth in computer power, advent of powerful algorithms for (numerical) convex optimization
- Can solve very rapidly many convex optimization problems for which no traditional "analytic" or "closed-form" solutions are known (or likely to exist)
- Changes fundamental notion "solution" to a problem: Reduction of a problem to a convex optimization problem is a "solution"

Objectives

- A brief introduction to Semidefinite Programming (SDP)
- Examples of SDP-based solutions for systems and control problems
- Robust Model Predictive Control (MPC)

Introduction to Semidefinite Programming (SDP)

Semidefinite Progamming

Convex optimization of the form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F_0 + x_1 F_1 + \cdots + x_p F_p > 0 \end{array}$$

$F_0, F_1, \ldots, F_p$ are given symmetric matrices, c is a vector, x is the vector of optimization variables

Compare with Linear Programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, i = 1, \ldots, N \end{array}$$

- Same linear objective
- Linear matrix inequality constraint instead of linear scalar inequalities

Linear matrix inequalities

$$F(x) = F_0 + x_1 F_1 + \cdots + x_p F_p > 0$$

called an "LMI"

- $x \in \mathbf{R}^p$ is the variable
- F_is are given symmetric matrices
- $F > 0$ means F is positive-definite, that is $u^T F u > 0$ for all nonzero vectors u

LMIs are nonlinear, but *convex* constraints:
If $F(x) > 0$ and $F(y) > 0$, then

$$F\left(\lambda x + (1-\lambda)y\right) = \lambda F(x) + (1-\lambda)F(y) > 0$$

for all $\lambda \in [0, 1]$

Matrices as variables

Example: Lyapunov inequality

$$A^T P + PA < 0$$

A is given, $P = P^T$ is the variable

This is an LMI in the entries of P:
With $P_1, \ldots, P_m$ a basis for symmetric matrices take $F_0 = 0$ and $F_i = -A^T P_i - P_i A$

Better to leave LMIs in a condensed form

- saves notation
- leads to more efficient computation

LMI examples

Many standard constraints can be written as LMIs

- *Linear constraints:*
  Componentwise inequality $Ax + b > 0$ is equivalent to the LMI
  $$\mathbf{diag}(Ax + b) > 0$$

- *Quadratic constraints:*
  Inequality $(Ax + b)^T(Ax + b) + c^T x + d < 0$ is equivalent to the LMI
  $$\begin{bmatrix} I & Ax + b \\ (Ax + b)^T & -(c^T x + d) \end{bmatrix} > 0$$

- *Matrix norm constraints:*
  If $A(x) \stackrel{\Delta}{=} A_0 + x_1 A_1 + \cdots + x_m A_m$, then $\|A(x)\|_2 < t$ is equivalent to the LMI

$$\begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} > 0$$

- Multiple LMIs $F^{(1)}(x) > 0, \ldots, F^{(N)}(x) > 0$ same as single LMI

$$\mathbf{diag}(F^{(1)}(x), \ldots, F^{(N)}(x)) > 0$$

More on LMIs

More constraints: With Q, R and S depending affinely on x

- Constraint $R(x) > 0, \quad Q(x) - S(x)R(x)^{-1}S(x)^T > 0$ equivalent to LMI

$$\left[\begin{array}{cc} Q(x) & S(x) \\ S(x)^T & R(x) \end{array}\right] > 0$$

- Constraint $\mathbf{Tr}\, S(x)^T Q(x)^{-1} S(x) < 1, \quad Q(x) > 0$, equivalent to LMI

$$\mathbf{Tr}\, X < 1, \quad \left[\begin{array}{cc} X & S(x)^T \\ S(x) & Q(x) \end{array}\right] > 0$$

(X is slack variable)

Solving SDPs

In theory:

- fundamental complexity is low
- there are polynomial-time algorithms

In practice:

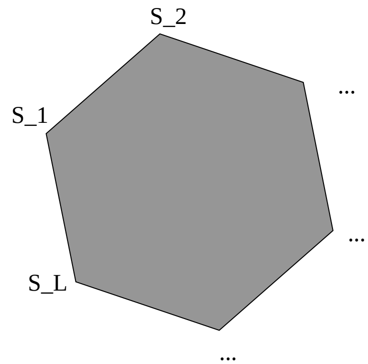
- classical optimization methods *do not* work
- new interior point methods solve LMI problems *extremely efficiently*
- matlab "LMI Control Toolbox" available

SDP applications in systems and control

A stability problem

Consider time-varying system

$$\frac{dx}{dt} = A(t)x(t), \qquad A(t) \in \mathbf{Co}\{A_1, \ldots, A_L\}$$



Is system stable, i.e., are all trajectories bounded?

- Widely-used system model for robust control
- $L = 1$ yields standard LTI system
- Can model uncertain systems, nonlinear systems, switching systems, etc

Analysis via quadratic Lyapunov functions

If there exists $V(\zeta) = \zeta^T P \zeta$, $P > 0$, s.t. $dV(x(t))/dt < 0$ along every trajectory, then system stable (a *sufficient* stability condition)

$dV(x(t))/dt < 0$ iff

$$A(t)^T P + PA(t) < 0, \; A(t) \in \mathbf{Co}\{A_1, \ldots, A_L\},$$

or

$$A_i^T P + PA_i < 0, \;\; i = 1, \ldots, L$$

Thus for *quadratic stability*, need P s.t.

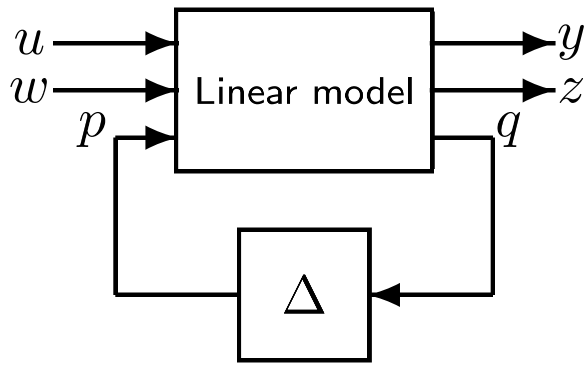
$$P > 0, \quad A_i^T P + PA_i < 0, \;\; i = 1, \ldots, L$$

Quadratic Lyapunov function search

- Condition $P > 0,\ A_i^T P + P A_i \leq 0,\ i = 1, \ldots, L$, an LMI in P; readily solved
- Looks simple, but is more powerful than many well known methods (multivariable circle criteria, . . .)
- There is a gap between stability of system and existence of quadratic Lyapunov function that proves it

Extensions

- Search for more general Lyapunov functions possible (parameter-dependent, Lur'e Postnikov, Lyapunov-Krasovskii, for example)
- Techniques extend to other robust control models:



  - Δ can be *structured*
  - Many assumptions on Δ can be handled (passivity, phase information, etc)

Robust performance analysis

Question: What is worst-case value, over uncertainties, of a "performance measure" $\int_0^\infty \phi(x(t), w(t), z(t))\, dt$?

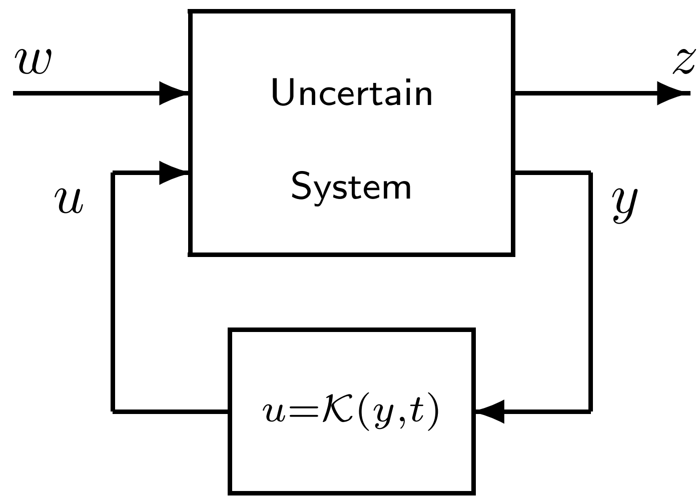
Approach: Suppose $V(x(t)) = x(t)^T P x(t)$, $P > 0$, and

$$\text{for all trajectories } \frac{d}{dt} V(x(t)) \leq -\phi(x(t), w(t), z(t)) \qquad (*)$$

Then, $x(0)^T P x(0) \geq \int_0^\infty \phi(x(t), w(t), z(t))\, dt$, or $x(0)^T P x(0)$ an upper bound on robust performance

- Condition $(*)$ is an LMI for a large class of functions ϕ
- Approach can be extended to handle more sophisticated Lyapunov functions, and to many robust control models

Robust control design



Problem: Find the optimal control strategy, $u = \mathcal{K}(y,t)$ s.t. closed-loop system enjoys good robustness properties

Some feedback possibilities:

- $y = x$, i.e., "state feedback"
- Only a linear combination of state components available, i.e., "output feedback"
- The uncertainties are measurable in real-time, and can be used for feedback, i.e., "gain-scheduled"

Constant state feedback

- Feedback strategy is $u(t) = Kx(t)$
- Yields SDPs in several important cases
- For example:

$$\dot{x}(t) = A(t)x(t)+Bu(t), \;\; u(t) = Kx(t), \;\; A(t) \in \mathbf{Co}\,\{A_1, \dots, A_L\}$$

Closed-loop system stable if

$$P = P^T > 0, \quad (A_i + BK)^T P + P(A_i + BK) < 0, \;\; i = 1, \dots, L$$

With $Q \stackrel{\Delta}{=} P^{-1}$, $Y \stackrel{\Delta}{=} KP^{-1}$,

$$Q = Q^T > 0, \quad QA_i^T + Y^T B^T + A_i Q + BY < 0, \; i = 1, \dots, L$$

(An LMI in variables Q and Y)

More general feedback design

- Feedback strategy $y = \mathcal{K}(y(t), t)$
- In general "much harder" than state feedback, get LMIs only in a handful of cases
- Constant or dynamic output feedback design do not yield LMIs in general
- With feedback "scheduled" by the measured uncertainties, i.e., "gain-scheduled feedback", get LMIs and SDPs

Robust Model Predictive Control

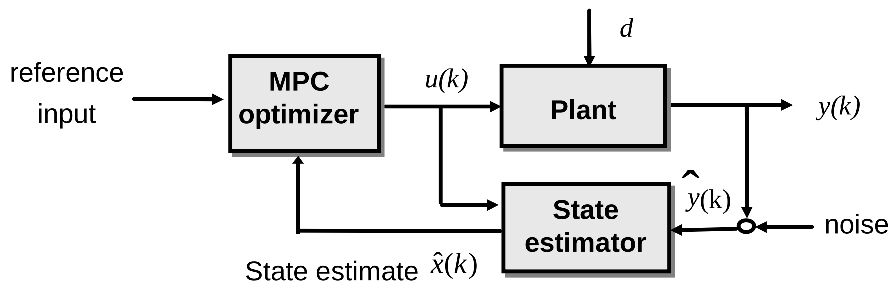
Model predictive control

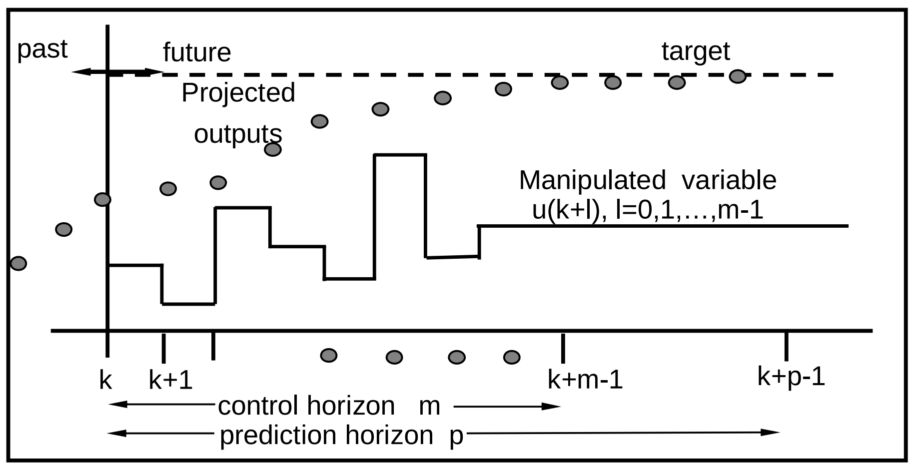
- An open-loop control design procedure
- At each k, m control moves $u(k+j|k)$, $j = 0, 1, \ldots, m-1$ computed using a prediction horizon p:

$$\min_{u(k+j|k),\ j=0,1,\ldots,m-1} J_p(k), \tag{1}$$

  subject to constraints on the control input, the predicted state, and the predicted output

- $J_p(k)$ is some sensible objective
- Only first control move implemented
- Process repeated


d
reference
input
MPC
optimizer
u(k)
Plant
y(k)
State
estimator
$\hat{y}$(k)
noise
State estimate $\hat{x}(k)$


past
future
target
Projected
outputs
Manipulated variable
u(k+l), l=0,1,…,m-1
k
k+1
k+m-1
k+p-1
control horizon m
prediction horizon p

A typical MPC implementation

- Take a nominal plant model
- Formulate control-move calculation as an optimization problem (LP/QP)

$$\begin{aligned} \text{minimize:} \quad & J_p(k) \\ \text{subject to:} \quad & u_{i,\min} \leq u_i(k+j|k) \leq u_{i,\max} \\ & x_{i,\min} \leq x_i(k+j|k) \leq x_{i,\max}, \end{aligned}$$

where

$$J_p(k) = \sum_{j=0}^{p-1} \Big(x(k+j|k)^T Q x(k+j|k) + u(k+j|k)^T R u(k+j|k) \Big)$$

- $R > 0$ determines "input penalty" and $Q \geq 0$ determines "state penalty", relative sizes determines trade-off

Stability of nominal MPC

- When nominal plant model is exact, then have well-known result (Rawlings, Muske (1993)):

  Take prediction horizon $p = \infty$. Then:

  $$\textit{feasibility of on-line MPC} \implies \textit{nominal MPC stability}$$

- However, when model errors are present:

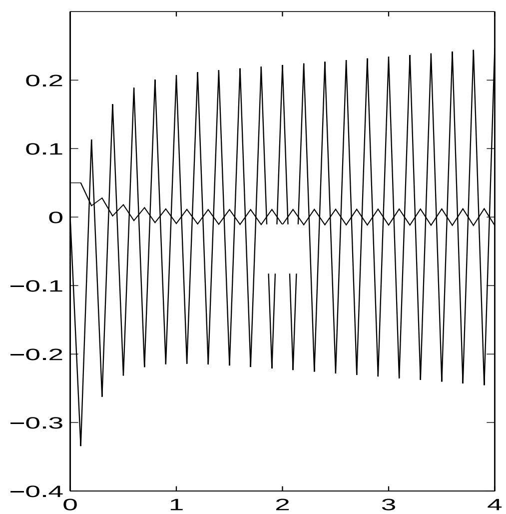
  $$\textit{feasibility of on-line MPC} \not\Longrightarrow \textit{robust MPC stability}$$

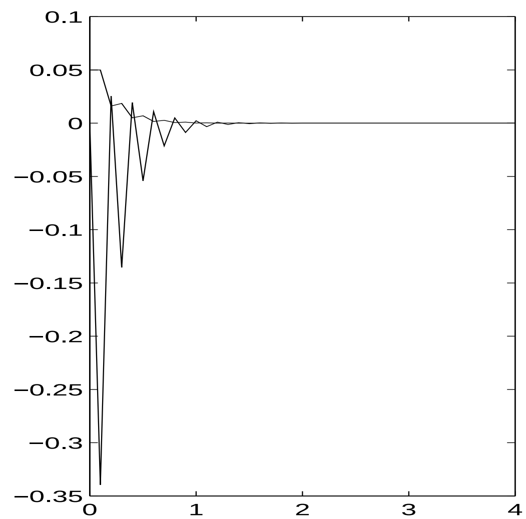
Counterexample

$$P(s) = \frac{k}{s(s+\alpha)}, \quad -2 \le u \le 2$$

Model uncertainty: $\quad 0.1 \le \alpha \le 10$

With nominal $\alpha = 1$ and actual $\alpha = 9$:



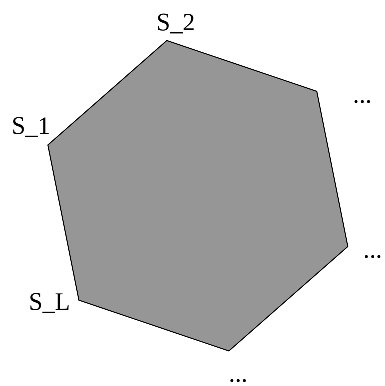


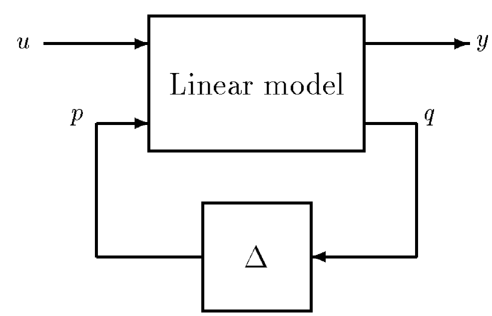
A robust MPC problem formulation

Discrete-time plant

$$\begin{aligned}
x(k+1) &= A(k)x(k) + B(k)u(k), \\
y(k) &= Cx(k), \\
[A(k) \quad B(k)] &\in \Omega
\end{aligned} \tag{2}$$

Uncertainty set Ω can be either a polytope, or be defined by a feedback uncertainty structure


S_2
S_1
...
...
S_L
...


u
y
Linear model
p
q
Δ

Robust MPC problem formulation

At time k, synthesize $u(k+j|k) = Kx(k+j|k)$ (i.e., find K) to solve

$$\text{minimize} \max_{[A(k+j) \quad B(k+j)] \in \Omega} J_\infty(k),$$

subject to:

$$|u_i(k+j|k)| \leq u_{i,\max}, \quad |y_i(k+j|k)| \leq y_{i,\max}$$

where

$$J_\infty(k) = \sum_{j=0}^{\infty} \left(x(k+j|k)^T Q x(k+j|k) + u(k+j|k)^T R u(k+j|k) \right)$$

Robust MPC problem solution idea

Suppose $P > 0$ satisfies

$$x(k+j+1|k)^T P x(k+j+1|k) - x(k+j|k)^T P x(k+j|k) \leq \\ -x(k+j|k)^T \left(Q + K^T R K\right) x(k+j|k) \quad (*)$$

for all $j \geq 0$

Then, summing both sides from 0 to ∞

$$x(k|k)^T P x(k|k) \geq \\ \sum_{j=0}^{\infty} \left(x(k+j|k)^T Q x(k+j|k) + u(k+j|k)^T R u(k+j|k)\right),$$

so $x(k|k)^T P x(k|k)$ is an upper bound on objective

SDP solution for polytopic Ω

With change of variables $Y = P^{-1}$, $W = KP^{-1}$, condition $(*)$ equivalent to LMI

$$\begin{bmatrix} -Y & YA_i^T - W^T B_i^T & YQ^{1/2} & W^T R^{1/2} \\ A_i Y - B_i W & -Y & 0 & 0 \\ Q^{1/2} Y & 0 & -I & 0 \\ R^{1/2} W & 0 & 0 & -I \end{bmatrix} \leq 0,$$

for $i = 1, \ldots, L$

Best upper bound gotten by soving at every k, the SDP

minimize:	$x(k\|k)^T Y^{-1} x(k\|k)$
subject to:	LMI

Actuator constraints

Define ellipsoid

$$\mathcal{E} = \left\{ \zeta \ : \ \zeta^T Y^{-1} \zeta \leq 1 \right\}$$

Then if $x(k|k) \in \mathcal{E}$, then $x(k+j|k) \in \mathcal{E}$ for all $j \geq 0$, i.e., $\mathcal{E}$ is an *invariant* ellipsoid

Using invariance of $\mathcal{E}$, and noting $u(k+j|k) = Kx(k+j|k)$, constraint

$$|u_i(k+j|k)| \leq u_{i,\max}$$

guaranteed by

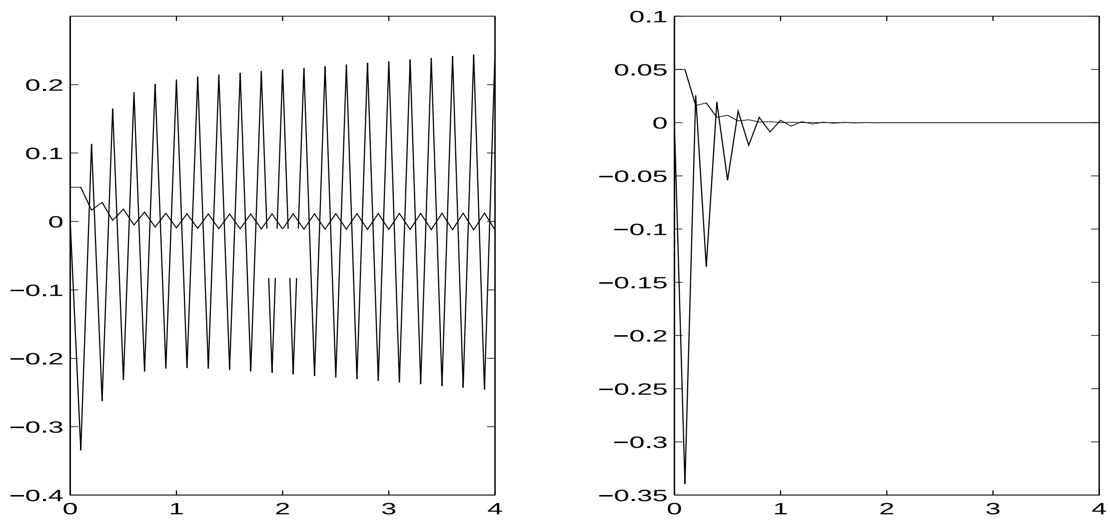
$$\begin{bmatrix} Z & W \\ W^T & Y \end{bmatrix} \geq 0, \quad Z_{ii} \leq u_{i,\max}^2$$

Example revisited

$$P(s) = \frac{k}{s(s+\alpha)}, \quad -2 \le u \le 2$$

Model uncertainty $0.1 \le \alpha \le 10$

With nominal $\alpha = 1$ and actual $\alpha = 9$:



Concluding remarks

- LMIs offer numerical solution of many robust control problems
- Interest began in the 1990s when efficient LMI algorithms became available, and continues to grow
- Rich literature available, many specialized results for special robust control problems

Challenges

- Many practical problems yield large-scale LMI problems, current solvers often inadequate. Special-purpose algorithms and solvers desirable
- Study of applications of LMIs to robust control fairly mature, other applications remain to be explored

Some references

- SIAM monograph *Linear Matrix Inequalities in System and Control Theory*, by Boyd et al.
- Kluwer *Handbook on Semidefinite Programming*, Wolkowicz et al. (editors)
- SIAM collection *Recent Advances on LMI Methods in Control*, El Ghaoui et al. (editors)
- Most recent control conferences ACC and CDC
- Kothare, et al., "An LMI approach to robust constrained MPC," *Automatica*, Oct 1996