ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

LMI solvers---Feasibility problem solver

LMI Solvers

- Three types of LMI solvers
- To test whether or not there exists a solution $\boldsymbol{x}$ to $\boldsymbol{F}(\boldsymbol{x}) \succ 0$ is called a **feasibility problem**
- Minimization of a linear objective under LMI constraints
- Generalized eigenvalue minimization problem

Solving the feasibility problem

- Can solve LMIs of the form

$$\boldsymbol{N}^\top \boldsymbol{\mathcal{L}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{N} \prec \boldsymbol{M}^\top \boldsymbol{\mathcal{R}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)\boldsymbol{M}$$

- $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$—matrix variables
- $\boldsymbol{N}$—left outer factor, $\boldsymbol{M}$—right outer factor
- $\boldsymbol{\mathcal{L}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$—left inner factor, $\boldsymbol{\mathcal{R}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$—right inner factor

Left-hand side vs. the right-hand side

- The term "left-hand side" refers to what is on the "smaller" side of the inequality $0 \prec \boldsymbol{X}$
- In $\boldsymbol{X} \succ 0$, the matrix $\boldsymbol{X}$ is on the right-hand side—it is on the "larger" side of the inequality

General structure for finding a feasible soln

```
setlmis([])
lmivar
lmiterm
.
.
.
lmiterm
getlmis
feasp
dec2mat
```

X=lmivar(type,structure)

- The input `type` specifies the structure of the variable `X`
- Three structures of matrix variables
- `type=1`—symmetric block diagonal matrix variable
- `type=2`—full rectangular matrix variable
- `type=3`—other cases

Second input of X=lmivar(type,structure)

- Additional info on the structure of the matrix variable $\boldsymbol{X}$
- Example

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{D}_2 & \cdots & \boldsymbol{O} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{D}_r \end{bmatrix}$$

- Each $\boldsymbol{D}_i$ is a square symmetric matrix—`type=1`
- r blocks—`structure` is $r \times 2$

The input structure

- The first component of each row of the input structure—corresponding block size
- The second element of each row—the block type
- `X=lmivar(1,[3 1])`
  full symmetric 3×3 matrix variable
- `X=lmivar(2,[2 3])`
  rectangular 2×3 matrix variable

Scalar block

- $$\boldsymbol{S} = \begin{bmatrix} s_1 & 0 & | & 0 & 0 \\ 0 & s_1 & | & 0 & 0 \\ -- & -- & | & -- & -- \\ 0 & 0 & | & s_2 & s_3 \\ 0 & 0 & | & s_3 & s_4 \end{bmatrix},$$
- `S=lmivar(1,[2 0;2 1])` describes a scalar block matrix, $\boldsymbol{D}_1 = s_1 \boldsymbol{I}_2$
- The second block is a 2×2 symmetric full block

lmiterm(termid,A,B,flag)

- `termid`—row with four elements specify the terms of each LMI of the LMI system
- `termid(1)=n` to specify the left-hand side of the n-th LMI
- `termid(1)=-n` to specify the right-hand side of the n-th LMI
- `termid(2,3)=[i j]` specifies the term of the (i, j) block of the LMI specified by the first component

More on lmiterm(termid,A,B,flag)

- `termid(4)=0` for the constant term
- `termid(4)=X` for the variable term in the form $\boldsymbol{AXB}$
- `termid(4)=-X` for the variable term in the form $\boldsymbol{AX}^\top\boldsymbol{B}$

Second and third inputs in lmiterm(termid,A,B,flag)

- `A` and `B` give the value of the constant outer factors in the variable terms, $\boldsymbol{AXB}$ or in $\boldsymbol{AX}^\top\boldsymbol{B}$
- the flag input to `lmiterm` serves as a compact way to specify the expression

$$\boldsymbol{AXB} + (\boldsymbol{AXB})^\top$$

Using flag in lmiterm(termid,A,B,flag)

- `flag='s'` use for symmetrized expression
- $\boldsymbol{PA} + \boldsymbol{A}^\top \boldsymbol{P} \prec 0$

  ```
  lmiterm([1 1 1 P],1,A)
  lmiterm([1 1 1 -P],A',1)
  ```
- Note that $\boldsymbol{PA} + \boldsymbol{A}^\top \boldsymbol{P} = \boldsymbol{PA} + (\boldsymbol{PA})^\top$
- `lmiterm([1 1 1 P],1,A,'s')`

[tmin,xfeas]=feasp(lmis)

- Feasibility problem

$$\begin{aligned} \text{find} \quad & \boldsymbol{x} \\ \text{such that} \quad & \boldsymbol{L}(\boldsymbol{x}) \prec \boldsymbol{R}(\boldsymbol{x}) \end{aligned}$$

- `feasp` solves the auxiliary convex

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & \boldsymbol{L}(\boldsymbol{x}) \prec \boldsymbol{R}(\boldsymbol{x}) + t\boldsymbol{I}. \end{aligned}$$

P=dec2mat(lmis,xfeas,P)

- The system of LMIs is feasible if the minimal $t < 0$
- `P=dec2mat(lmis,xfeas,P)` converts the output of the LMI solver into matrix variables