ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

LMI solvers---mincx and gevp

Minimizer of a Linear Objective Subject to LMI Constraints

- Invoked using the function `mincx`
- $$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}(\boldsymbol{x}) \prec \boldsymbol{B}(\boldsymbol{x}). \end{aligned}$$
- $\boldsymbol{A}(\boldsymbol{x}) \prec \boldsymbol{B}(\boldsymbol{x})$ is a shorthand notation for general structured LMI systems

Example

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \end{array}$$

where

$$\begin{aligned} \boldsymbol{c}^\top &= \begin{bmatrix} 4 & 5 \end{bmatrix}, \\ \boldsymbol{A} &= \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 8 \\ 18 \\ 14 \end{bmatrix}. \end{aligned}$$

- We first solve the feasibility problem, that is, we find an $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, using the `feasp` solver
- Then, we solve the above minimization problem using the `mincx` solver

Example— feasibility problem code

```
% Enter problem data
A =[1 1;1 3;2 1];b=[8 18 14]';c=[-4 -5]';
setlmis([]);
X=lmivar(2,[2 1]);
lmiterm([1 1 1 X],A(1,:),1);
lmiterm([1 1 1 0],-b(1));
lmiterm([1 2 2 X],A(2,:),1);
lmiterm([1 2 2 0],-b(2));
lmiterm([1 3 3 X],A(3,:),1);
lmiterm([1 3 3 0],-b(3));
lmis=getlmis;
disp('---feasp result---')
[tmin,xfeas]=feasp(lmis);
x_feasp=dec2mat(lmis,xfeas,X)
```

Example—minimization problem code

```
% Enter problem data
A =[1 1;1 3;2 1];b=[8 18 14]';c=[-4 -5]';
setlmis([]);
X=lmivar(2,[2 1]);
lmiterm([1 1 1 X],A(1,:),1);
lmiterm([1 1 1 0],-b(1));
lmiterm([1 2 2 X],A(2,:),1);
lmiterm([1 2 2 0],-b(2));
lmiterm([1 3 3 X],A(3,:),1);
lmiterm([1 3 3 0],-b(3));
lmis=getlmis;
disp('---mincx result---')
[obj,x_min]=mincx(lmis,c,[0.0001 1000 0 0 1])
```

Example—output

- The `feasp` produces

$$\boldsymbol{x}_{feasp} = \begin{bmatrix} -64.3996 \\ -25.1712 \end{bmatrix}.$$

- The `mincx` produces

$$\boldsymbol{x}_{mincx} = \begin{bmatrix} 3.0000 \\ 5.0000 \end{bmatrix}.$$

Function defcx

- The function `defcx` is used to construct the vector $\boldsymbol{c}$ used by the LMI solver `mincx`
- Suppose that we wish to solve the optimization problem

$$\begin{array}{ll} \text{minimize} & \operatorname{trace}(\boldsymbol{P}) \\ \text{subject to} & \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} \prec 0, \end{array}$$

  where $\boldsymbol{P} = \boldsymbol{P}^\top \succ 0$.
- Can use the function `mincx` to solve the above problem
- However, to use the function `mincx`, we need a vector $\boldsymbol{c}$ such that

$$\boldsymbol{c}^\top \boldsymbol{x} = \operatorname{trace}(\boldsymbol{P}).$$

Example: using function defcx

- After specifying the LMIs and obtaining their internal representation using, for example, the command `lmisys=getlmis`, we can obtain the desired $\boldsymbol{c}$ with MATLAB's code,

```
q=decnbr(lmisys);
c=zeros(q,1);
for j=1:q
    Pj=defcx(lmisys,j,P);
    c(j)=trace(Pj);
end
```

- Having obtained the vector $\boldsymbol{c}$, we can use the function `mincx` to solve the optimization problem

Generalized Eigenvalue Minimization Problem

- The problem:

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \\ & \begin{array}{rcl} \boldsymbol{C}(\boldsymbol{x}) & \prec & \boldsymbol{D}(\boldsymbol{x}) \\ 0 & \prec & \boldsymbol{B}(\boldsymbol{x}) \\ \boldsymbol{A}(\boldsymbol{x}) & \prec & \lambda \boldsymbol{B}(\boldsymbol{x}). \end{array} \end{array}$$

- Need to distinguish between standard LMI constraints of the form $\boldsymbol{C}(\boldsymbol{x}) \prec \boldsymbol{D}(\boldsymbol{x})$ and the linear-fractional LMIs of the form $\boldsymbol{A}(\boldsymbol{x}) \prec \lambda \boldsymbol{B}(\boldsymbol{x})$ that are concerned with the generalized eigenvalue λ, that is, the LMIs involving λ
- The number of linear-fractional constraints is specified with `nflc`
- The generalized eigenvalue minimization problem under LMI constraints is solved by calling the solver `gevp`

Basic structure of the gevp solver

- The number of linear-fractional constraints is specified with `nflc`
- The generalized eigenvalue minimization problem under LMI constraints is solved by calling the solver `gevp`
- The basic structure of the `gevp` solver:

  `[lopt,xopt]=gevp{lmisys,nflc}`
- It returns `lopt`, which is the global minimum of the generalized eigenvalue, and `xopt`, which is the optimal decision vector variable
- The argument `lmisys` is the system of LMIs, $\boldsymbol{C}(\boldsymbol{x}) \prec \boldsymbol{D}(\boldsymbol{x})$, $0 \prec \boldsymbol{B}(\boldsymbol{x})$, and $\boldsymbol{A}(\boldsymbol{x}) \prec \lambda \boldsymbol{B}(\boldsymbol{x})$ for $\lambda = 1$.
- The corresponding optimal values of the matrix variables are obtained using `dec2mat`

Application of the gevp solver

- Consider a system model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

  where the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is asymptotically stable
- We wish to estimate the decay rate of the system's trajectory $\boldsymbol{x}(t)$, that is, we wish to find constants $\eta > 0$ and $M > 0$ such that

$$\|\boldsymbol{x}(t)\| \leq e^{-\eta t} M(\boldsymbol{x}_0)$$

- Because $\boldsymbol{A}$ is asymptotically stable, by Lyapunov's theorem, for any $\boldsymbol{Q} = \boldsymbol{Q}^\top \succ 0$ there exists $\boldsymbol{P} = \boldsymbol{P}^\top \succ 0$ such that

$$\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q},$$

  that is,

$$\boldsymbol{x}^\top \left(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}\right) \boldsymbol{x} = -\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$$

Application of the gevp solver—contd.

- Let
$$V = \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x}$$
- Then
$$\dot{V} = \boldsymbol{x}^\top \left(\boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A} \right) \boldsymbol{x},$$
which is the Lyapunov derivative of V evaluated on trajectories of $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$
- Let
$$\alpha = \min \left(-\frac{\dot{V}}{V} \right).$$

Decay rate

- Let
$$\alpha = \min\left(-\frac{\dot{V}}{V}\right).$$
- We have
$$\dot{V} \leq -\alpha V$$
- Therefore,
$$V(t) \leq e^{-\alpha t} V(0).$$
- We refer to α as the decay rate of V
- We have
$$\boldsymbol{x}(t)^\top \boldsymbol{P} \boldsymbol{x}(t) \leq e^{-\alpha t} \boldsymbol{x}_0^\top \boldsymbol{P} \boldsymbol{x}_0$$
- Hence
$$\lambda_{\min}(\boldsymbol{P}) \|\boldsymbol{x}\|_2^2 \leq \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x}$$

Decay rate—contd.

- Because $\boldsymbol{P} = \boldsymbol{P}^\top \succ 0$, we have that $\lambda_{\min}(\boldsymbol{P}) > 0$
- Combining and dividing both sides by $\lambda_{\min}(\boldsymbol{P}) > 0$ gives

$$\|\boldsymbol{x}(t)\|^2 \leq e^{-\alpha t} \frac{\boldsymbol{x}_0^\top \boldsymbol{P} \boldsymbol{x}_0}{\lambda_{\min}(\boldsymbol{P})}$$

- Hence,

$$\|\boldsymbol{x}(t)\| \leq e^{-\frac{\alpha}{2} t} \sqrt{\frac{\boldsymbol{x}_0^\top \boldsymbol{P} \boldsymbol{x}_0}{\lambda_{\min}(\boldsymbol{P})}}\,.$$

- Represent $\boldsymbol{P} = \boldsymbol{P}^\top$ as

$$\boldsymbol{P} = \boldsymbol{P}^{1/2} \boldsymbol{P}^{1/2}.$$

Computing decay rate

- Hence,

$$\begin{aligned}\boldsymbol{x}_0^\top \boldsymbol{P}\boldsymbol{x}_0 &= \boldsymbol{x}_0^\top \boldsymbol{P}^{1/2}\boldsymbol{P}^{1/2}\boldsymbol{x}_0 \\ &= \|\boldsymbol{P}^{1/2}\boldsymbol{x}_0\|^2\end{aligned}$$

- Hence

$$\|\boldsymbol{x}(t)\| \leq e^{-\frac{\alpha}{2}t}\frac{\|\boldsymbol{P}^{1/2}\boldsymbol{x}_0\|}{\sqrt{\lambda_{\min}(\boldsymbol{P})}}$$

and we obtain

$$\eta = \alpha/2 \quad \text{and} \quad M(\boldsymbol{x}_0) = \frac{\|\boldsymbol{P}^{1/2}\boldsymbol{x}_0\|}{\sqrt{\lambda_{\min}(\boldsymbol{P})}}\,.$$

- Finding α is equivalent to minimizing α subject to

$$\begin{aligned}\boldsymbol{P} &\succ 0 \\ \boldsymbol{A}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} &\prec -\alpha\boldsymbol{P}.\end{aligned}$$

Example: Computing decay rate

```
A =[-1.1853    0.9134    0.2785
     0.9058   -1.3676    0.5469
     0.1270    0.0975   -3.0000];
setlmis([]);
P = lmivar(1,[3 1])
lmiterm([-1 1 1 P],1,1)                % P
lmiterm([1 1 1 0],.01)         % P > 0.01*I
lmiterm([2 1 1 P],1,A,'s')
 % linear fractional constraint: left-hand side
lmiterm([-2 1 1 P],1,1)
 % linear fractional constraint: right-hand side
lmis = getlmis;
[gamma,P_opt]=gevp(lmis,1);
P=dec2mat(lmis,P_opt,P)
alpha=-gamma
```

Example: decay rate

- Matrix A

```
A  =[-1.1853    0.9134     0.2785
      0.9058   -1.3676     0.5469
      0.1270    0.0975    -3.0000];
```

- Solution

$$\alpha = 0.6561 \quad \text{and} \quad \boldsymbol{P} = \begin{bmatrix} 0.6996 & -0.7466 & -0.0296 \\ -0.7466 & 0.8537 & -0.2488 \\ -0.0296 & -0.2488 & 3.2307 \end{bmatrix}$$