

# Semidefinite Programming

## Computational Intelligence, Lecture 10

by Sergei Savin

Fall 2020

- Semidefinite Programming (SDP)
  - General form
  - Multiple LMI
  - SDP decision variable
- Example 1: Continuous Lyapunov equation as an SDP/LMI
  - Mathematical formulation
  - Code
- Example 2: Discrete Lyapunov equation as an SDP/LMI
  - Mathematical formulation
  - Code
- Example 3: LMI Controller design for continuous LTI
  - Mathematical formulation
  - Code
- Homework

# Semidefinite Programming (SDP)

## General form

General form of a semidefinite program is:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{c}^\top \mathbf{x}, \\ \text{subject to} & \begin{cases} \mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0, \\ \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \end{array} \quad (1)$$

where  $\mathbf{F}_i \succeq 0$  and  $\mathbf{G} \succeq 0$  (meaning they are positive semidefinite).

Constraint  $\mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0$  is called *linear matrix inequality* or *LMI*.

# Semidefinite Programming (SDP)

## Multiple LMI

SDP can have several LMIs. Assume you have:

$$\begin{cases} \mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0 \\ \mathbf{D} + \sum \mathbf{H}_i x_i \preceq 0 \end{cases} \quad (2)$$

This is equivalent to:

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} + \sum \begin{bmatrix} \mathbf{F}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_i \end{bmatrix} x_i \preceq 0 \quad (3)$$

# Semidefinite Programming (SDP)

SDP decision variable

Sometimes it is easier to directly think of semidefinite matrices as of decision variables. This leads to programs with such formulation:

$$\begin{array}{ll} \underset{\mathbf{X}}{\text{minimize}} & f(\mathbf{X}), \\ \text{subject to} & \begin{cases} \mathbf{X} \preceq 0, \\ \mathbf{g}(\mathbf{X}) = \mathbf{0}. \end{cases} \end{array} \quad (4)$$

where cost and constraints should adhere to SDP limitations.

# Example 1: Continuous Lyapunov equation as an SDP/LMI

## Mathematical formulation

In control theory, Lyapunov equation is a condition of whether or not a continuous LTI system  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$  is stabilizable:

$$\begin{cases} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = 0 \\ \mathbf{P} \succeq 0 \\ \mathbf{Q} \succeq 0 \end{cases} \quad (5)$$

where decision variable is  $\mathbf{P}$ . This can be represented as an SDP:

$$\begin{aligned} & \underset{\mathbf{P}}{\text{minimize}} && 0, \\ & \text{subject to} && \begin{cases} \mathbf{P} \succeq 0, \\ \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = 0. \end{cases} \end{aligned} \quad (6)$$

# Example 1: Continuous Lyapunov equation as an SDP/LMI

Code

```
0 n = 7; A = randn(n, n) - 3*rand*eye(n);  
  Q = eye(n);  
2  
  cvx_begin sdp  
4      variable P(n, n) symmetric  
      minimize 0  
6      subject to  
          P >= 0;  
8          A'*P + P*A + Q <= 0;  
  cvx_end  
10  
  if strcmp(cvx_status, 'Solved')  
12      [eig(A), eig(A*P + P*A' + Q), eig(P)]  
  else  
14      eig(A)  
  end
```

## Example 2: Discrete Lyapunov equation as an SDP/LMI

### Mathematical formulation

In control theory, Discrete Lyapunov equation is a condition of whether or not a discrete LTI system  $\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i$  is stabilizable:

$$\begin{cases} \mathbf{A}^\top \mathbf{P} \mathbf{A} - \mathbf{P} + \mathbf{Q} = 0 \\ \mathbf{P} \succeq 0 \\ \mathbf{Q} \succeq 0 \end{cases} \quad (7)$$

where decision variable is  $\mathbf{P}$ . This can be represented as an SDP:

$$\begin{aligned} & \underset{\mathbf{P}}{\text{minimize}} && 0, \\ & \text{subject to} && \begin{cases} \mathbf{P} \succeq 0, \\ \mathbf{A}^\top \mathbf{P} \mathbf{A} - \mathbf{P} + \mathbf{Q} = 0. \end{cases} \end{aligned} \quad (8)$$



## Example 2: Discrete Lyapunov equation as an SDP/LMI

Code

```
0 n = 7; A = 0.35*randn(n, n);  
  Q = eye(n);  
2  
  cvx_begin sdp  
4      variable P(n, n) symmetric  
      minimize 0  
6      subject to  
          P >= 0;  
8          A'*P*A - P + Q <= 0;  
  cvx_end  
10  
  if strcmp(cvx_status, 'Solved')  
12      [abs(eig(A)), eig(A'*P*A - P), eig(P)]  
  else  
14      abs(eig(A))  
  end
```

## Example 3: LMI Controller design for continuous LTI

### Mathematical formulation

For an LTI system of the form  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$  there is an LMI condition to determine if it can be stabilized:

$$\begin{cases} \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{B}\mathbf{L} + \mathbf{L}\mathbf{B}^\top + \mathbf{Q} = 0 \\ \mathbf{P} \succeq 0 \\ \mathbf{Q} \succeq 0 \end{cases} \quad (9)$$

where decision variables are  $\mathbf{P}$  and  $\mathbf{L}$ .

This gives as a direct way to calculate linear feedback controller  $\mathbf{u} = \mathbf{K}\mathbf{x}$  (note the sign!) gains:

$$\mathbf{K} = \mathbf{L}\mathbf{P}^{-1} \quad (10)$$

# Example 3: LMI Controller design for continuous LTI

## Code

```
0 n = 5; m = 2;
  A = randn(n, n);
2 B = randn(n, m);
  Q = eye(n)*0.1;
4 cvx_begin sdp
    variable P(n, n) symmetric
6    variable Z(m, n)

    minimize 0
    subject to
10    P >= 0;
        A*P + P*A' + B*Z + Z'*B' + Q <= 0;
12 cvx_end
  P = full(P);
14 Z = full(Z);
  K_LMI = Z*pinv(P);
16
  disp('K_LMI eig:')
18 eig(A + B*K_LMI)
```

# Homework

Implement both examples from page 2 of the [LMI CVX documents](#).

Lecture slides are available via Moodle.

You can help improve these slides at:

[github.com/SergeiSa/Computational-Intelligence-Slides-Fall-2020](https://github.com/SergeiSa/Computational-Intelligence-Slides-Fall-2020)

Check Moodle for additional links, videos, textbook suggestions.