Sahand University of Technology
Control Engineering Department

LMI in Control Systems

Hadi Azmi

Fall 2022

A Brief History of LMIs in Control Theory

The history of LMIs in the analysis of dynamical systems goes back more than 100 years. The story begins in about 1890, when Lyapunov published his seminal work introducing what we now call Lyapunov theory. He showed that the differential equation

$$\frac{d}{dt}x(t) = Ax(t) \tag{1.1}$$

is stable (i.e., all trajectories converge to zero) if and only if there exists a positive-definite matrix P such that

$$A^T P + PA < 0. \tag{1.2}$$

The requirement $P > 0$, $A^T P + PA < 0$ is what we now call a Lyapunov inequality on P, which is a special form of an LMI. Lyapunov also showed that this first LMI could be explicitly solved. Indeed, we can pick any $Q = Q^T > 0$ and then solve the linear equation $A^T P + PA = -Q$ for the matrix P, which is guaranteed to be positive-definite if the system (1.1) is stable. In summary, the first LMI used to analyze stability of a dynamical system was the Lyapunov inequality (1.2), which can be solved analytically (by solving a set of linear equations).

The next major milestone occurs in the 1940's. Lur'e, Postnikov, and others in the Soviet Union applied Lyapunov's methods to some specific practical problems in control engineering, especially, the problem of stability of a control system with a nonlinearity in the actuator. In summary, Lur'e and others were the first to apply Lyapunov's methods to practical control engineering problems.

The next major breakthrough came in the early 1960's, when Yakubovich, Popov, Kalman, and other researchers succeeded in reducing the solution of the LMIs that arose in the problem of Lur'e to simple graphical criteria, using what we now call the positive-real (PR) lemma.

The PR lemma and extensions were intensively studied in the latter half of the 1960s, and were found to be related to the ideas of passivity, the small-gain criteria introduced by Zames and Sandberg, and quadratic optimal control.

By 1970, it was known that the LMI appearing in the PR lemma could be solved not only by graphical means, but also by solving a certain algebraic Riccati equation (ARE). In a 1971 on quadratic optimal control, J. C. Willems is led to the LMI

$$\begin{bmatrix} A^T P + PA + Q & PB + C^T \\ B^T P + C & R \end{bmatrix} \geq 0, \tag{1.3}$$

and points out that it can be solved by studying the symmetric solutions of the ARE

$$A^T P + PA - (PB + C^T)R^{-1}(B^T P + C) + Q = 0,$$

So by 1971, researchers knew several methods for solving special types of LMIs: direct (for small systems), graphical methods, and by solving Lyapunov or Riccati equations. From our point of view, these methods are all \closed-form" or \analytic" solutions that can be used to solve special forms of LMIs.

In a 1976 paper, Horisberger and Belanger [HB76] had remarked that the existence of a quadratic Lyapunov function that simultaneously proves stability of a collection of linear systems is a convex problem involving LMIs.

In 1984, N. Karmarkar introduced a new linear programming algorithm that solves linear programs in polynomial-time, like the ellipsoid method, but in contrast to the ellipsoid method, is also very efficient in practice. Karmarkar's work spurred an enormous amount of work in the area of interior-point methods for linear programming.

A summary of key events in the history of LMIs in control theory is then:

- ✓ **1890:** First LMI appears; analytic solution of the Lyapunov LMI via Lyapunov equation.
- ✓ **1940:** Application of Lyapunov's methods to real control engineering problems. Small LMIs solved by hand".
- ✓ **Early 1960:** PR lemma gives graphical techniques for solving another family of LMIs.
- ✓ **Late 1960:** Observation that the same family of LMIs can be solved by solving an ARE.
- ✓ **Early 1980:** Recognition that many LMIs can be solved by computer via convex programming.
- ✓ **Late 1980:** Development of interior-point algorithms for LMIs.

It is fair to say that Yakubovich is the father of the field, and Lyapunov the grandfather of the field. The new development is the ability to directly solve (general) LMIs.

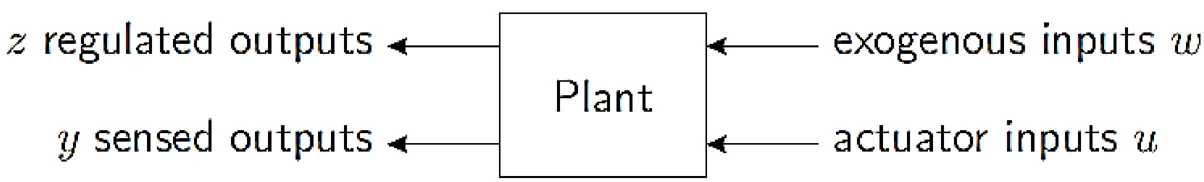
Well... What is a System?

Definition 1.

A **System** is anything with Inputs and Outputs

There should ALWAYS be Inputs and Outputs!

- **If No Inputs**: You can't change anything.
- **IF No Outputs**: Then it doesn't matter anyway.



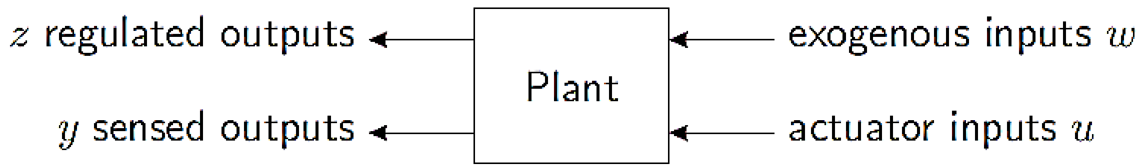
In Controls, we separate internal signals from external signals.

Output Signals:

- **z:** Output to be controlled/minimized
- **y:** Output used by the controller

Input Signals:

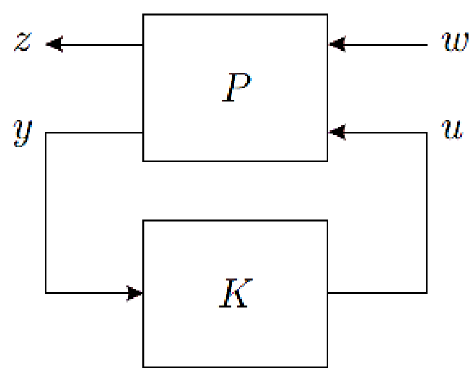
- **w:** Disturbance, Tracking Signal, etc.
- **u:** Output from controller
  - Input to actuator



A state-space system has the form (9-matrix representation)

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t) \\
z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\
y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t)
\end{aligned}
$$

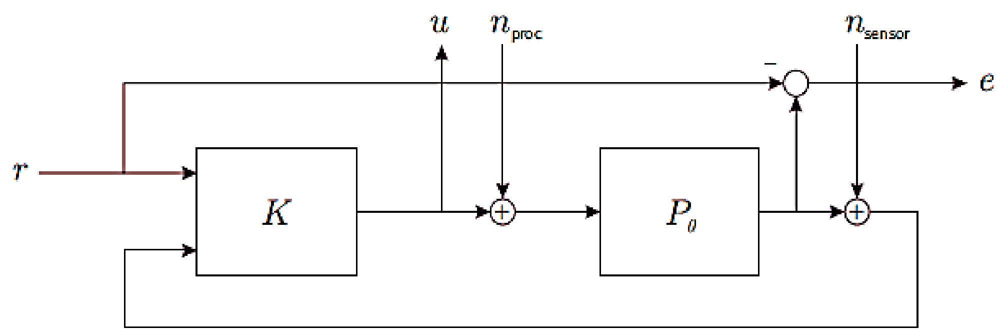
$x(t) \in \mathbb{R}^n$ is the *internal state.*
$x \in L_2^n$ is the *internal signal.*



The controller, K, determines how to use the **signal** y to get the **signal** u.

- Can be *dynamic*: $u(t) = F\hat{x}(t)$, $\dot{\hat{x}}(t) = A\hat{x}(t) + L(y(t) - C\hat{x}(t))$
- Can be *static*: $u(t) = Fy(t)$.

Our job is to find the BEST K.

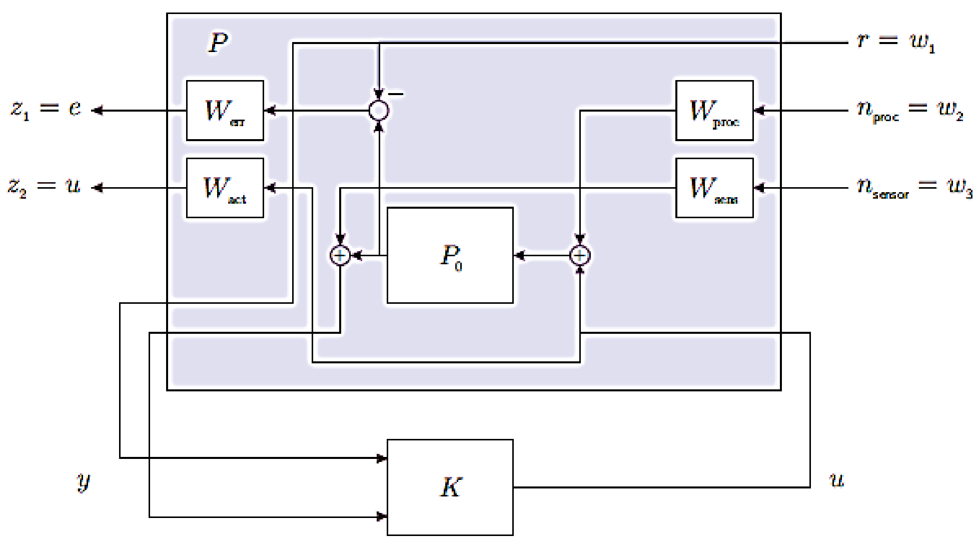


$r =$ reference input
$e =$ tracking error
$n_{proc} =$ process noise
$n_{sensor} =$ sensor noise

$w_2 = n_{proc}$
$w_3 = n_{sensor}$
$z_1 = e$
$z_2 = u$

$w_1 = r$
$u = u$
$y_1 = r$
$y_2 = y_p$

Tracking Control



$$P = \begin{bmatrix} I & -P_0 & 0 & -P_0 \\ 0 & 0 & 0 & I \\ I & 0 & 0 & 0 \\ 0 & P_0 & I & P_0 \end{bmatrix}$$

$$z_1 = r - P_0(n_{proc} + u)$$
$$z_2 = u$$
$$y_1 = r$$
$$y_2 = w_3 + P_0(n_{proc} + u)$$

What is Optimization?

An Optimization Problem has 3 parts.

$$\min_{x \in \mathbb{F}} \quad f(x): \qquad \text{subject to}$$
$$g_i(x) \geq 0 \qquad i = 1, \cdots K_1$$
$$h_i(x) = 0 \qquad i = 1, \cdots K_2$$

Variables: $x \in \mathbb{F}$

- The things you must choose.
- $\mathbb{F}$ represents the set of possible choices for the variables.
- Can be vectors, matrices, functions, systems, locations, colors...
  - However, computers prefer vectors or matrices.

Objective: $f(x)$

- A function which assigns a *scalar* value to any choice of variables.
  - e.g. $[x_1, x_2] \mapsto x_1 - x_2$; red $\mapsto 4$; et c.

Constraints: $g(x) \geq 0$; $h(x) = 0$

- Defines what is a minimally acceptable choice of variables.
- Equality forces two things to be the same
- Inequalities force one thing to be "better" than another
  - x is OK if $g(x) \geq 0$ and $h(x) = 0$.
- Constraints mean variables are not independent.

How Hard is it to Solve Optimization Problems

For Humans:

- Almost always IMPOSSIBLE (or at least tedious)

For Computers:

- Easy if the Problem is CONVEX. (Polynomial Time)
- Otherwise IMPOSSIBLE. (NP-Hard)

We will talk about this a bit more later!

Now What is an LMI?

An Example: The Lyapunov Inequality

The system

$$\dot{x} = Ax$$

is stable (eigenvalues have negative real part) if and only if there exists a $P > 0$ such that

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

YALMIP Code for Stability Analysis:

```
> A = [-1 2 0; -3 -4 1; 0 0 -2];
> P = sdpvar(3,3);
> F = [P >= eye(3)];
> F = [F, A'*P+P*A <= 0];
> optimize(F);
```

If Feasible, YALMIP Code to Retrieve the Solution:

```
> Pfeasible = value(P);
```

```
%% Lecture01_EX_01_mfile

clc;
clear;
close all;

%% Parameters

A = [-1 2 0; -3 -4 1; 0 0 -2];

%% LMI Definition

P = sdpvar(3,3);
F=A'*P+P*A;
const=[F<0]+[P>0];
optimize(const,[],sdpsettings('solver','sedumi'));

%% LMI Results Values

P_feasible = value(P)
```

```
Pfeasible =

    0.5604    0.0452   -0.0070
    0.0452    0.2489    0.0348
   -0.0070    0.0348    0.4116
```

```
%% Lecture01_EX_02_mfile

clc;
clear;
close all;

%% Parameters

A = [-1 2 0; -3 -4 1; 3 0 -2];
B= [1;-2;0];

%% LMI  Definition

P = sdpvar(3,1);
F=A*P-B;
const=[F<0];
optimize(const,[],sdpsettings('solver','sedumi'));

%% LMI Results Values

P_feasible = value(P)
```

P_feasible =

3.0000

0.5000

6.0000

```
%% Parameters

A = [-1 2 5; -3 -4 1; 3 7 -2];
B= [1;-2;4];
```

P_feasible =

5.7500

-2.2500

1.2500

ans

1x1 struct with 6 fields

Field	Value
yalmipversion	'20200930'
matlabversion	'9.7.0.1190202 (R2019b)'
yalmiptime	0.1035
solvertime	0.0395
info	'Successfully solved (SeDuMi-1.3)'
problem	0

تعریف LMI

شکل کلی LMI بصورت زیر است:

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0$$

$$x = [x_1 \quad \cdots \quad x_m]$$

مثال:

$$\begin{bmatrix} x_1 + 3x_2 & x_2 + x_1 & 5 + 4x_1 \\ x_2 + x_1 & 3 & 4 - x_1 \\ 5 + 4x_1 & 4 - x_1 & x_1 + x_2 \end{bmatrix} > 0$$

حل LMI

- LMI Parser YALMIP
- Solver
  - ✓ SeDuMi
  - ✓ SDPT3
  - ✓ MOSEK
  - ✓ PENBMI
  - ✓ ....

دستورات اصلی

- sdpvar
- optimize
- sdpsettings
- value
- uncertain
- check

sdpvar

- دستوری برای تعریف سمبلیک متغیرها
- فرمت دستور بصورت زیر است:

```
P = sdpvar(n,n,'symmetric')
P = sdpvar(n,n,'full')
sdpvar x y
P = sdpvar(n,n)      (SYMMETRIC)
P = sdpvar(n)        (SYMMETRIC)
P = sdpvar(n,n,'full','complex')
```

sdpsettings

این دستور برای تنظیم پارامترهای YALMIP و Solver استفاده می شود:

```
options = sdpsettings('field',value,'field',value,...)
optimize(Constraints, Objective, options)
```

مثال:

```
ops = sdpsettings('solver','mosek','verbose',0)
```

optimize

دستور کلی حل مسائل بهینه سازی

```
diagnostics = optimize(Constraints,Objective,options)
```

مثال: حل مساله LP

```
x = sdpvar(length(c),1)
F = [A*x<=b]
h = c'*x
optimize(F,h)
```

value

دستوری برای بدست آوردن مقادیر عددی متغیرهای تصمیم گیری بعد از بهینه سازی است:

Y=value(x)

مثال:

```
x = sdpvar(2,1)
F = [-1 <= x <= 1]
obj = x'*x + sum(x)
optimize(F,obj)
optobj = value(obj)
optx = value(x)
```

Read the following paper carefully and do all of examples:

2004 IEEE International Symposium on
Computer Aided Control Systems Design
Taipei, Taiwan, September 2-4, 2004

YALMIP : A toolbox for modeling and optimization in MATLAB

Johan Löfberg
Automatic Control Laboratory, ETHZ
CH-8092 Zürich, Switzerland.
loefberg@control.ee.ethz.ch