

Overview

- ▶ course mechanics
- ▶ outline & topics
- ▶ what is a linear dynamical system?
- ▶ why study linear systems?
- ▶ some examples

Lecture notes and course materials originally by Stephen Boyd. Revisions by Sanjay Lall.

Course mechanics

- ▶ class web page: ee263.stanford.edu
- ▶ announcements and forum: piazza.com/stanford/summer2019/ee263
- ▶ grades: canvas.stanford.edu
- ▶ homework submissions: <https://www.gradescope.com/courses/51614>
access code: 9GVJ5D
- ▶ email: [\[gorish, amomenis\]@stanford.edu](mailto:[gorish, amomenis]@stanford.edu)
- ▶ lecture videos online: mvideox.stanford.edu

Course requirements

- ▶ Weekly homework, due Wednesday at 11:59 PM
- ▶ 8-hour takehome midterm exam (06/20-06/21)
- ▶ 12-hour takehome final exam (07/16-07/17)
- ▶ Homework 40%, midterm 25%, final 35% (tentative)

check the website for updates

Prerequisites

- ▶ exposure to linear algebra (*e.g.*, Math 104)
- ▶ exposure to Laplace transform, differential equations

not needed, but might increase appreciation:

- ▶ control systems
- ▶ circuits & systems
- ▶ dynamics

Major topics & outline

- ▶ linear algebra & applications
- ▶ autonomous linear dynamical systems
- ▶ linear dynamical systems with inputs & outputs
- ▶ basic quadratic control & estimation

Linear dynamical system

continuous-time linear dynamical system (CT LDS) has the form

$$\frac{dx}{dt} = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t) + D(t)u(t)$$

where:

- ▶ $t \in \mathbb{R}$ denotes *time*
- ▶ $x(t) \in \mathbb{R}^n$ is the *state* (vector)
- ▶ $u(t) \in \mathbb{R}^m$ is the *input* or *control*
- ▶ $y(t) \in \mathbb{R}^p$ is the *output*
- ▶ $A(t) \in \mathbb{R}^{n \times n}$ is the *dynamics matrix*
- ▶ $B(t) \in \mathbb{R}^{n \times m}$ is the *input matrix*
- ▶ $C(t) \in \mathbb{R}^{p \times n}$ is the *output* or *sensor matrix*
- ▶ $D(t) \in \mathbb{R}^{p \times m}$ is the *feedthrough matrix*

Linear dynamical system

for lighter appearance, equations are often written

$$\dot{x} = Ax + Bu, \quad y = Cx + Du$$

- ▶ CT LDS is a first order vector *differential equation*
- ▶ also called *state equations*, or '*m*-input, *n*-state, *p*-output' LDS

Some LDS terminology

- ▶ most linear systems encountered are *time-invariant*: A, B, C, D are constant, *i.e.*, don't depend on t
- ▶ when there is no input u (hence, no B or D) system is called *autonomous*
- ▶ very often there is no feedthrough, *i.e.*, $D = 0$
- ▶ when $u(t)$ and $y(t)$ are scalar, system is called *single-input, single-output* (SISO); when input & output signal dimensions are more than one, MIMO

Discrete-time linear dynamical system

discrete-time linear dynamical system (DT LDS) has the form

$$x(t+1) = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t) + D(t)u(t)$$

where

- ▶ $t \in \mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$
- ▶ (vector) signals x , u , y are *sequences*

DT LDS is a first-order vector *recursion*

Why study linear systems?

applications arise in **many** areas, *e.g.*

- ▶ automatic control systems
- ▶ signal processing
- ▶ communications
- ▶ economics, finance
- ▶ circuit analysis, simulation, design
- ▶ mechanical and civil engineering
- ▶ aeronautics
- ▶ navigation, guidance

Usefulness of LDS

- ▶ depends on availability of **computing power**, which is large & increasing exponentially
- ▶ used for
 - ▶ analysis & design
 - ▶ implementation, embedded in real-time systems

Origins and history

- ▶ parts of LDS theory can be traced to 19th century
- ▶ builds on classical circuits & systems (1920s on) (transfer functions ...) but with more emphasis on linear algebra
- ▶ first engineering application: aerospace, 1960s
- ▶ transitioned from specialized topic to ubiquitous in 1980s (just like digital signal processing, information theory, ...)

Nonlinear dynamical systems

many dynamical systems are **nonlinear** (a fascinating topic) so why study **linear** systems?

- ▶ most techniques for nonlinear systems are based on linear methods
- ▶ methods for linear systems often work unreasonably well, in practice, for nonlinear systems
- ▶ if you don't understand linear dynamical systems you certainly can't understand nonlinear dynamical systems