

Introduction, linear system representations

Control Theory, Lecture 1

by Sergei Savin

Spring 2022

- Motivation
- Ordinary differential equations
 - ▶ 1st order
 - ▶ n-th order
- Linear differential equations
 - ▶ 1st order
 - ▶ n-th order
- Changing n-th order ODE to a State-Space form
- State-Space to ODE
- Read more

WHAT IS CONTROL?

The first obvious question is, what is control theory? The easiest strategy to answer this question is to bring examples of systems that you can *learn how to control*:



Figure 1: Drone



Figure 2: Robot arms

But beware, this is not the whole answer!

WHY CONTROL?

The second most natural question to ask is - why do we need to study Control Theory? *Why do Computer scientists need Control Theory?*

The easy answer is:

it is very useful in case you will work in robotics, industrial automation, self-driving vehicles, drones, aerospace, etc.

But!

this answer does not tell the main part of the story - what about people who are NOT going to work in the listed areas?

CONTROL AS AN APPLIED PROBLEM

We propose to view Control Theory as not only yet-another-subject. Instead we can try to see Control Theory course as **an application of your combined skills as a CS student**

CONTROL AS AN APPLIED PROBLEM

Skills you will learn and practice

In this course we provide you with learning and practical tasks that require:

- Linear Algebra, Differential Equations, Computational methods
- Dynamical systems, Stability (concept build on top of Theory of Ordinary Differential Equations).
- Simulation of dynamical systems (closely related to computational methods in Differential Equations), as a programming problem.
- Development of experiments in Google Colab, using Python, mathematical libraries, solving concrete, real world-related math-oriented problems.
- Representation (parametrization) of equations as a tool in both mathematical analysis and simulation, software development and problem solving.
- ...and many other things.

...SO, WHY CONTROL?

Control Theory, as given here, is focused on:

- ① Giving you a challenge to simultaneously learn a new concepts, new general and subject-specific math, and new programming tools.
- ② Providing you with clear outcomes in terms of *understanding* and ability to *solve well-defined and meaningful real-world problems*.
- ③ Being very useful for those who will proceed to work in robotics, automation, self-driving vehicles, drones, etc.

See it as a test case for your abilities as a CS specialist.

ENOUGH FOR THE MOTIVATION

Now that we know (kinda) why we do it:

Let's start with the content of the course!

ORDINARY DIFFERENTIAL EQUATIONS

1st order

Let us remember the normal form of first-order *ordinary differential equations (ODEs)*:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad (1)$$

where $\mathbf{x} = \mathbf{x}(t)$ is the solution of the equation and t is a free variable.

Definition

We can call this equation (same as any other ODE) a *dynamical system*, and \mathbf{x} is called the *state* of the dynamical system.

Example

$$\dot{x} = -3x^3 - 7 \quad (2)$$

ORDINARY DIFFERENTIAL EQUATIONS

n-th order

The normal form of an *n-th order* ordinary differential equation is:

$$x^{(n)} = f(x^{(n-1)}, x^{(n-2)}, \dots, \ddot{x}, \dot{x}, x, t) \quad (3)$$

where $x = x(t)$ is the solution of the equation. Same as before, it is a *dynamical system*, but this time the set $\{x, \dot{x} \dots, x^{(n-1)}\}$ is called the *state* of the dynamical system.

Example

$$\ddot{x} = \cos(2\dot{x}) - 10x + 7 \quad (4)$$

Example

$$\begin{cases} \ddot{x}_1 = \dot{x}_1 + x_1 + x_2^2 - 4 \\ \ddot{x}_2 = 10x_1^3 + \ddot{x}_2 \end{cases} \quad (5)$$

LINEAR DIFFERENTIAL EQUATIONS

1st order

Linear ODEs of the first order have normal form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \quad (6)$$

Example

$$\begin{cases} \dot{x}_1 = -20x_1 + 7x_2 + 17 \\ \dot{x}_2 = 10.5x_1 - 3x_2 - 5 \end{cases} \quad (7)$$

Example

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -8 & 5 & 2 \\ 0.5 & -10 & -2 \\ 1 & -1 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 4 \\ 10 \\ -5 \end{bmatrix} \quad (8)$$

LINEAR DIFFERENTIAL EQUATIONS

n-th order

A single linear ODE of the n-th order are often written in the form:

$$a_n x^{(n)} + a_{(n-1)} x^{(n-1)} + \dots + a_2 \ddot{x} + a_1 \dot{x} + a_0 x = b \quad (9)$$

Example

$$12 \ddot{x} - 3\ddot{x} + 5.5\dot{x} + 2x = 10.5 \quad (10)$$

Example

$$5\ddot{x} - 2\dot{x} + 10x = 2 \quad (11)$$

LINEAR DIFFERENTIAL EQUATIONS

...are what we will study

In this course we will focus entirely on linear dynamical systems. In particular, we will take a good use of the following two forms:

$$a_n x^{(n)} + a_{(n-1)} x^{(n-1)} + \dots + a_2 \ddot{x} + a_1 \dot{x} + a_0 x = b \quad (12)$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \quad (13)$$

the last one is called *state-space representation*.

Good news:

Both of those can be used to express any linear system, hence we can change one into the other.

CHANGING N-TH ORDER ODE TO A STATE-SPACE FORM

Consider eq. $\ddot{x} + a_2\ddot{x} + a_1\dot{x} + a_0x = b$.

Make a substitution: $z_1 = x$, $z_2 = \dot{x}$, $z_3 = \ddot{x}$. Therefore:

$$\begin{cases} \dot{z}_1 = \dot{x} = z_2 \\ \dot{z}_2 = \ddot{x} = z_3 \\ \dot{z}_3 = -a_2\ddot{x} - a_1\dot{x} - a_0x + b = -a_2z_3 - a_1z_2 - a_0z_1 + b \end{cases} \quad (14)$$

Which can be directly put in the state-space form:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \quad (15)$$

An example of how linear algebra serves
to solve a seemingly difficult problem

(advanced, not going to be on the test)

STATE SPACE TO ODE

part 1

Consider a system in state-space form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \iff \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \quad (16)$$

We want to find such equation

$$\ddot{y} + b_2\dot{y} + b_1y = 0 \quad (17)$$

that there exists a linear transformation of the initial conditions of (16) to the initial conditions of (17), such that the resulting solutions of the initial value problem for both (16) and (17) can be transformed into one-another via another linear transformation.

STATE SPACE TO ODE

part 2

We start by recognizing that differentiation is a linear operation, so $\dot{y}(t)$ is a linear transformation of (17) of the solution $y(t)$.

Next, we know that $y = \mathbf{w}^\top \mathbf{x}$ for some $\mathbf{w} = [w_1 \ w_2]$:

$$\dot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{x} \quad (18)$$

$$\dot{y} = [(a_{11}w_1 + a_{21}w_2) \ (a_{12}w_1 + a_{22}w_2)] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (19)$$

Analogous for \ddot{y} :

$$\ddot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{A} \mathbf{x} \quad (20)$$

STATE SPACE TO ODE

part 3

Combining our results we find the linear transformation between the variables x_1, x_2 and y, \dot{y} :

$$\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} w_1 & w_2 \\ (a_{11}w_1 + a_{21}w_2) & (a_{12}w_1 + a_{22}w_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (21)$$

We can choose any w_1, w_2 , as long as the resulting transformation matrix \mathbf{T} is not degenerate:

$$\mathbf{T} = \begin{bmatrix} w_1 & w_2 \\ (a_{11}w_1 + a_{21}w_2) & (a_{12}w_1 + a_{22}w_2) \end{bmatrix} \quad (22)$$

STATE SPACE TO ODE

part 4

Remember that:

$$\ddot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{A} \mathbf{x} \iff \ddot{y} = -b_1 y - b_2 \dot{y} = -\mathbf{b}^\top \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \quad (23)$$

Using the map we found previously, we obtain \ddot{y} as a linear function of y , \dot{y} , with parameters w_1 , w_2 :

$$\ddot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{A} \mathbf{T}^+ \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \quad (24)$$

$$\mathbf{b} = -\mathbf{w}^\top \mathbf{A} \mathbf{A} \mathbf{T}^+ \quad (25)$$

From this it is clear how the same can be generalized to higher dimensions.

STATE SPACE TO ODE

part 5

[Check out the code implementation.](#)



- State Space Representations of Linear Physical Systems
lpsa.swarthmore.edu/Representations/SysRepSS.html
- Transformation: Differential Equation to State Space
lpsa.swarthmore.edu/.../DE2SS.html

THANK YOU!

Lecture slides are available via Moodle.

You can help improve these slides at:

github.com/SergeiSa/Control-Theory-Slides-Spring-2022

Check Moodle for additional links, videos, textbook suggestions.

