Introduction, linear system representations

Control Theory, Lecture 1

by Sergei Savin

Spring 2022

Content

- 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?

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**

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!

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

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

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

$$\dot{x} = -3x^3 - 7 \tag{2}$$

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

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

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

Example

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

Example

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

Linear ODEs of the first order have normal form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{6}$$

Example

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

Example

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

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

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

Example

$$12\, \dddot{x} - 3\ddot{x} + 5.5\dot{x} + 2x = 10.5 \tag{10}$$

Example

$$5\ddot{x} - 2\dot{x} + 10x = 2 \tag{11}$$

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

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

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{13}$$

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. $\dddot{x} + a_2\ddot{x} + a_1\dot{x} + a_0 x = b$.

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

$$\begin{cases} \dot{z}_1 = \dot{x} = z_2 \\ \dot{z}_2 = \ddot{x} = z_3 \\ \dot{z}_3 = -a_2\ddot{x} - a_1\dot{x} - a_0 x + b = -a_2 z_3 - a_1 z_2 - a_0 z_1 + b \end{cases} \tag{14}$$

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

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

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

(advanced, not going to be on the test)

Consider a system in state-space form:

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

We want to find such equation

$$\ddot{y} + b_2 \dot{y} + b_1 y = 0 \tag{17}$$

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.

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 $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$:

$$\dot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{x} \tag{18}$$

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

Analogous for $\ddot{y}$:

$$\ddot{y} = \mathbf{w}^\top \mathbf{A} \mathbf{A} \mathbf{x} \tag{20}$$

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

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

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

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

Remember that:

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

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

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

$$\mathbf{b} = -\mathbf{w}^\top \mathbf{A}\mathbf{A}\mathbf{T}^+ \tag{25}$$

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

Check out the code implementation.

Read more

- 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.

Stability

Control Theory, Lecture 2

by Sergei Savin

Spring 2022

Content

- Critical point (node)
- Stability
- Asymptotic stability
- Stability vs Asymptotic stability
- LTI and autonomous LTI
- Stability of autonomous LTI
  - Example: real eigenvalues
  - Example: complex eigenvalues
  - General case
  - Illustration
- Read more

Critical point (node)

Consider the following ODE:

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

Let $\mathbf{x}_0$ be such a state that:

$$\mathbf{f}(\mathbf{x}_0, t) = 0 \tag{2}$$

Then such state $\mathbf{x}_0$ is called a *node* or a *critical point*.

Node $\mathbf{x}_0$ is called *stable* iff for any constant δ there exists constant ε such that:

$$||\mathbf{x}(0) - \mathbf{x}_0|| < \delta \;\longrightarrow\; ||\mathbf{x}(t) - \mathbf{x}_0|| < \varepsilon \tag{3}$$

Think of it as "for any initial point that lies at most δ away from $\mathbf{x}_0$, the rest of the trajectory $\mathbf{x}(t)$ will be at most ε away from $\mathbf{x}_0$".

Or, more picturesque, think of it as "the solutions with different initial conditions do not diverge from the node"

Node $\mathbf{x}_0$ is called *asymptotically stable* iff for any constant δ it is true that:

$$||\mathbf{x}(0) - \mathbf{x}_0|| < \delta \;\longrightarrow\; \lim_{t \to \infty} \mathbf{x}(t) = \mathbf{x}_0 \tag{4}$$

Think of it as "for any initial point that lies at most δ away from $\mathbf{x}_0$, the trajectory $\mathbf{x}(t)$ will asymptotically approach the point $\mathbf{x}_0$".

Or, more picturesque, think of it as "the solutions with different initial conditions converge to the node"

Stability vs Asymptotic stability

Example

Consider dynamical system $\dot{x} = 0$, and solution $x = 7$. This solution is stable, but not asymptotically stable (other solutions do not diverge from $x = 7$, but do not converge to it either).

Example

Consider dynamical system $\dot{x} = -x$, and solution $x = 0$. This solution is stable and asymptotically stable (other solutions converge to $x = 0$).

Example

Consider dynamical system $\dot{x} = x$, and solution $x = 0$. This solution is unstable (other solutions diverge from $x = 0$).

Consider the following linear ODE:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5}$$

This is called a *linear time-invariant system*, or *LTI*.

Consider the following linear ODE:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{6}$$

This is also an LTI, but it is also called an *autonomous system*, since its evolution depends only on the state of the system.

Consider autonomous LTI:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{7}$$

where $\mathbf{A}$ can be decomposed via eigen-decomposition as $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$, where $\mathbf{D}$ is a diagonal matrix.

$$\dot{\mathbf{x}} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\mathbf{x} \tag{8}$$

Multiply it by $\mathbf{V}^{-1} \longrightarrow \mathbf{V}^{-1}\dot{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{V}\mathbf{D}\mathbf{V}^{-1}\mathbf{x}$.
Define $\mathbf{z} = \mathbf{V}^{-1}\mathbf{x} \longrightarrow \dot{\mathbf{z}} = \mathbf{D}\mathbf{z}$.

Since elements of $\mathbf{D}$ are real, we can clearly see, that iff they are *all negative* will the system be asymptotically stable. If they are non-positive, the system is stable. And those elements are eigenvalues of $\mathbf{A}$.

Let us consider the following system:

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{9}$$

The eigenvalues of the system are $\alpha \pm i\beta$. We denote $\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{x}$.

We start by claiming that the system will be stable iff the $\dot{\mathbf{x}}^\top \mathbf{x} < 0$. Indeed, vector $\dot{\mathbf{x}}$ can always be decomposed into two components, $\dot{\mathbf{x}}_{||}$ parallel to $\mathbf{x}$, and $\dot{\mathbf{x}}_{\perp}$ perpendicular to $\mathbf{x}$. By definition $\dot{\mathbf{x}}_{\perp}^\top \mathbf{x} = 0$, and is responsible for the change in orientation of $\mathbf{x}$. The value of $\dot{\mathbf{x}}_{||}$ is responsible for the change in the length of $\mathbf{x}$; the length would shrink iff $\dot{\mathbf{x}}_{||}$ is of opposite direction to $\mathbf{x}$, giving negative value of the dot product $\dot{\mathbf{x}}^\top \mathbf{x}$.

Let us compute $\dot{\mathbf{x}}^\top \mathbf{x}$:

$$\dot{\mathbf{x}}^\top \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{10}$$

$$\dot{\mathbf{x}}^\top \mathbf{x} = \alpha(\mathbf{x}_1^2 + \mathbf{x}_2^2) \tag{11}$$

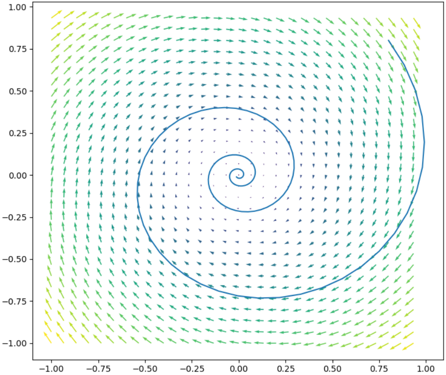
From this it is clear that the product $\dot{\mathbf{x}}^\top \mathbf{x} < 0$ is negative iff $\alpha < 0$.

Definition

As long as the *real parts of the eigenvalues* of the system are *strictly negative*, the system is *asymptotically stable.* If the real parts of the eigenvalues of the system are zero, the system is *marginally stable.*

Complex eigenvalues, 2-dimensional case (3)

Vector field of $\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$ is shown below:



Given $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ can be decomposed via eigen-decomposition as $\mathbf{A} = \mathbf{U}\mathbf{C}\mathbf{U}^{-1}$, where $\mathbf{C}$ is a complex-valued diagonal matrix and $\mathbf{U}$ is a complex-valued inevitable matrix.

We multiply both sides by $\mathbf{U}^{-1}$, then define $\mathbf{z} = \mathbf{U}^{-1}\mathbf{x}$ to arrive at:

$$\dot{\mathbf{z}} = \mathbf{C}\mathbf{z} \tag{12}$$

which falls into a set of independent equations, with complex coefficients c_j:

$$\dot{z}_j = c_j z_j \tag{13}$$

General case (2)

Expanding $c_j = \alpha + i\beta$, and $z_j = u + iv$ (we dismiss subscripts for clarity), we find that $\dot{z}_j = c_j z_j$ can be expanded as:

$$\dot{u} + i\dot{v} = \dot{z}_j = c_j z_j = (\alpha + i\beta)(u + iv) \tag{14}$$

$$\dot{u} + i\dot{v} = \alpha u + i\beta u + i\alpha v - \beta v \tag{15}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{16}$$

As we can see, $\dot{z}_j = c_j z_j$ is asymptotically stable iff $\mathrm{Re}(c_j) < 0$, and marginally stable if $\alpha = \mathrm{Re}(c_j) = 0$. Same is true for $\dot{\mathbf{z}} = \mathbf{C}\mathbf{z}$ and hence, for $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, as $\mathbf{U}$ is invertible.

Consider an autonomous LTI:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{17}$$

Definition

Eq. (17) is stable iff real parts of eigenvalues of $\mathbf{A}$ are non-positive.

Definition

Eq. (17) is asymptotically stable iff real parts of eigenvalues of $\mathbf{A}$ are negative.

Here is an illustration of *phase portraits* of two-dimensional LTIs with different types of stability:

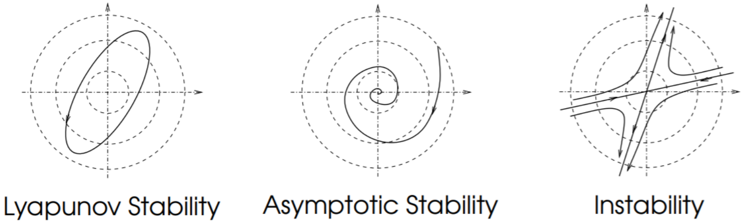
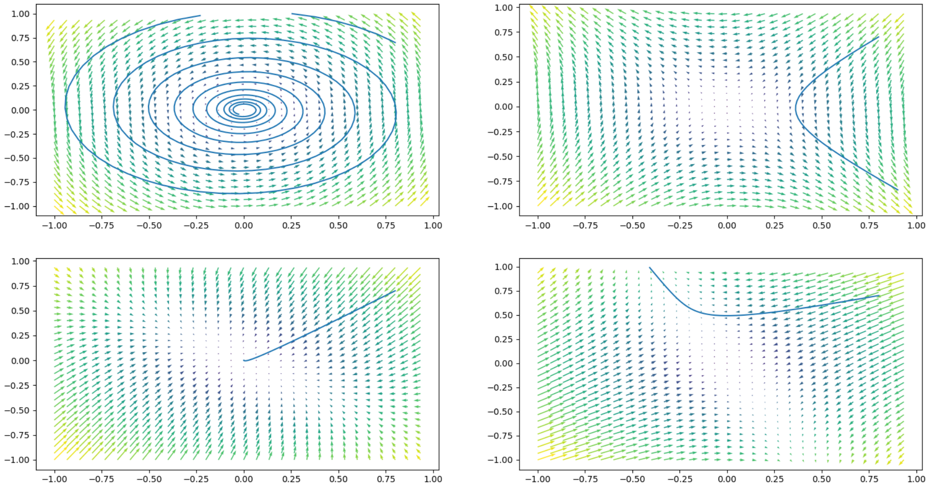


Figure 1: phase portraits for different types of stability

Credit: staff.uz.zgora.pl/wpaszke/materialy/spc/Lec13.pdf


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

Read more

- Control Systems Design, by Julio H. Braslavsky
  staff.uz.zgora.pl/wpaszke/materialy/spc/Lec13.pdf

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.

Laplace Transform and Transfer Functions

Control Theory, Lecture 3

by Sergei Savin

Spring 2022

Content

- ODE solutions
- Laplace Transform
- Laplace Transform of a derivative
- Derivative operator
- Transfer Function
  - Example
  - Interesting things done easy
- State-Space to Transfer Function conversion
- Read more

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -10 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} u \tag{1}$$

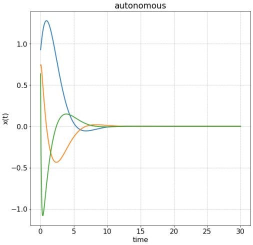


Figure 1: Autonomous ODE $(u = 0)$

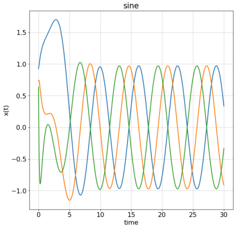


Figure 2: reaction to sine wave $(u = sin(t))$

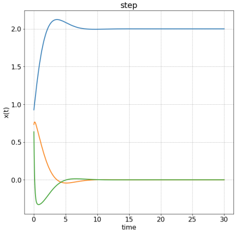


Figure 3: Reaction to step function $(u = 1)$

By definition, Laplace transform of a function $f(t)$ is given as:

$$F(s) = \int_0^\infty f(t)e^{-st}dt \tag{2}$$

where $F(s)$ is called an *image* of the function.

The study of Laplace transform is a separate mathematical field with applications in solving ODEs, which we won't cover. However, we will consider transform of one case of interest - transform of a derivative.

Consider a derivative $\frac{dx}{dt}$ and its transform:

$$\mathcal{L}\left(\frac{dx}{dt}\right) = \int_0^{\infty} \frac{dx}{dt} e^{-st} dt \tag{3}$$

we will make use of the integration by parts formula:

Integration by parts

$$\int v \frac{du}{dt} dt = vu - \int \frac{dv}{dt} u dt \tag{4}$$

In our case, $\frac{du}{dt} = \frac{dx}{dt}$, $u = x$, $v = e^{-st}$, $\frac{dv}{dt} = -se^{-st}$:

$$\mathcal{L}\left(\frac{dx}{dt}\right) = \left[xe^{-st}\right]_0^{\infty} - \int_0^{\infty} -se^{-st} x dt \tag{5}$$

$$\mathcal{L}\left(\frac{dx}{dt}\right) = -x(0) + s\mathcal{L}(x) \tag{6}$$

Thus, assuming that $x(0) = 0$, we can obtain a *derivative operator*:

$$\mathcal{L}\left(\frac{dx}{dt}\right) = s\mathcal{L}\left(x\right) \tag{7}$$

Please notice that (7) is only true when $x(0) = 0$; it generally does not look very elegant either. Introducing a big-time abuse of notation, we can denote $x(s) = \mathcal{L}\left(x\right)$ and then drop the brackets, leaving us with:

$$\frac{dx}{dt} \longrightarrow sx \tag{8}$$

This form of a derivative operator has a very strange notation in terms of the Laplace transform theory, but is very simple to use in practice.

Consider the following ODE, where u is an input (function of time that influences the solution of the ODE):

$$\ddot{x} + a\dot{x} + bx = u \tag{9}$$

We can rewrite it using the derivative operator:

$$s^2 x + asx + bx = u \tag{10}$$

and then collect x on the left-hand-side:

$$x = \frac{1}{s^2 + as + b} u \tag{11}$$

At this point the mathematical meaning of this expression as an ODE is very vague, but it has a different direct use; this form is called a *transfer function*.

Example

Given ODE: $2\dddot{x} + 5\dot{x} - 40x = 10u$

The transfer function for it looks: $x = \frac{10}{2s^3 + 5s - 40} u$

Example

Given ODE: $2\dot{x} - 4x = u$

The transfer function for it looks: $x = \frac{1}{2s - 4} u$

Example

Given ODE: $3\dddot{x} + 4x = u$

The transfer function for it looks: $x = \frac{1}{2s^3 + 4} u$

Consider the following (strange) ODE:

$$2\ddot{x} + 3\dot{x} + 2x = 10\dot{u} - u \tag{12}$$

Using the differential equation:

$$2s^2x + 3sx + 2x = 10su - u \tag{13}$$

...which is the same as:

$$(2s^2 + 3s + 2)x = (10s - 1)u \tag{14}$$

The transfer function for it looks:

$$x = \frac{10s - 1}{2s^2 + 3s + 2}u \tag{15}$$

Transfer functions are being used to study the relation between the input and the output of the dynamical system.

Consider standard form state-space dynamical system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases} \tag{16}$$

We can rewrite it using the derivative operator:

$$\begin{cases} s\mathbf{I}\mathbf{x} - \mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases} \tag{17}$$

and then collect $\mathbf{x}$ on the left-hand-side: $\mathbf{x} = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}$ and finally, express $\mathbf{y}$ out:

$$\mathbf{y} = \left(\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right)\mathbf{u} \tag{18}$$

Let the dynamic system be described as a transfer function:

$$y = G(s)x \tag{19}$$

We can try to modify the input based on how the output looks-like. Since we always do it in a linear way, we can write it as:

$$y = G(s)(x - H(s)y) \tag{20}$$

where $H(s)y$ is called *feedback*.

How would the transfer function from x to y look like?

From $y = G(s)(x - H(s)y)$ we go:

$$y = G(s)x - G(s)H(s)y \tag{21}$$

$$y + G(s)H(s)y = G(s)x \tag{22}$$

$$y = \frac{G(s)}{1 + G(s)H(s)}x \tag{23}$$

Thus, we found *closed-loop* transfer function:

$$W(s) = \frac{G(s)}{1 + G(s)H(s)} \tag{24}$$

Read more

- Chapter 6 Transfer Functions
- Control Systems Lectures - Transfer Functions, by Brian Douglas
- The Laplace Transform - A Graphical Approach, by Brian Douglas

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.

Frequency response, Bode

Control Theory, Lecture 4

by Sergei Savin

Spring 2022

Content

- Laplace and Fourier transforms
- Laplace and steady state solution
- Bode plot
- Bode plot - example
- Stability margins
- Code example

- *Fourier series* can be seen as representing a periodic function as a sum of harmonics (sines and cosines). These sines and cosines can be thought of as forming a basis in a linear space. The coefficients of the series can be thought of as a discrete spectrum of the function.
- *Fourier transform* gives a continuous spectrum of the function. The "basis" is still made of harmonic functions.
- *Laplace transform* also gives a continuous spectrum of the function, but in a different basis: the basis is given by complex exponentials. I like to think of this basis as solutions of second order ODEs.

Let's compare. Fourier transform:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi j t \omega} dt, \quad \omega \in \mathbb{R} \tag{1}$$

Laplace transform:

$$F(s) = \int_{0}^{\infty} f(t) e^{-st} dt, \quad s \in \mathbb{C} \tag{2}$$

We can see that Fourier looks like Laplace with purely imaginary number in the exponent.

Laplace and steady state solution

From analysing solutions of linear ODEs we know that, given harmonic input (sine, cosine, their combination) "after the transient process is over, the solution approaches a harmonic with the same frequency", but possibly different amplitude and phase.

Intuitively we can think of the imaginary part of s as having to do with this frequency response.

The kernel function of the Laplace transform is e^{-st} with $s = \sigma + j\omega$ being a complex variable. If $\sigma = 0$, the kernel becomes $e^{-j\omega t} = \cos(\omega t) - j\sin(\omega t)$. You can see the similarity with Fourier transform kernel.

The first key idea of a Bode plot is substitution of purely complex variable $j\omega$ in place of Laplace variable s, which can have non-zero real part.

Given a transfer function $W(s)$, $s = \sigma + j\omega$ we can analyse its behaviour when $\sigma = 0$. We can plot its amplitude $a(\omega) = |W(j\omega)|$ and its phase $\varphi(\omega) = \text{atan2}(\text{im}(W(j\omega)),\ \text{real}(W(j\omega)))$.

Bode plot is actually two plots, 1) $20 \cdot \log(a(\omega))$ and 2) $\frac{180}{\pi}\varphi(\omega)$. The 20 and log has to do with the vertical axis being in decibels.

Consider $W(s) = \frac{1}{1+s}$. Then $W(j\omega) = \frac{1}{1+j\omega}$. We can transform it as:

$$W(j\omega) = \frac{1 - j\omega}{(1 + j\omega)(1 - j\omega)} = \frac{1 - j\omega}{1 + \omega^2} \tag{3}$$

We that we know that $\text{real}(W(j\omega)) = \frac{1}{1+\omega^2}$ and $\text{im}(W(j\omega)) = -\frac{\omega}{1+\omega^2}$.

Bode plot is then given as:

$$a(\omega) = \sqrt{\frac{1 - \omega^2}{(1 + \omega^2)^2}} \tag{4}$$

$$\varphi(\omega) = \text{atan2}\left(-\frac{\omega}{1 + \omega^2},\ \frac{1}{1 + \omega^2}\right) \tag{5}$$

Before we discuss the use of Bode plot, let us remember that closed-loop transfer function has form (when simple feedback is used):

$$W(s) = \frac{G(s)}{1 + G(s)} \tag{6}$$

Substituting $s \longrightarrow j\omega$ we get:

$$W(\omega) = \frac{G(j\omega)}{1 + G(j\omega)} \tag{7}$$

From this we can see that $W(\omega)$ becomes ill-defined if $G(j\omega) = -1$. Meaning, we want to avoid two things happening simultaneously: the amplitude of $G(j\omega)$ being equal to 1, and its phase (argument) being equal to $180°$ (remember, phase of $0°$ is pure positive real number, phase of $90°$ is pure positive imaginary number, $180°$ is pure negative real number, etc.).

Stability margins - graphical example

Let's check an illustration:

Code example

Check the colab notebook based on the example above for an illustration of how the Bode plot can be made by hand or via scipy signal library.

- Control System Lectures - Bode Plots, Introduction

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.

Stabilizing Control

Control Theory, Lecture 5

by Sergei Savin

Spring 2022

Content

- Changing stability
- Stabilizing control
- Linear control: Closed-loop system,
- Affine control
- Error dynamics
- Affine trajectory tracking
- Point-to-point control
- Pure state feedback
- Read more

Here are two LTIs:

$$\dot{x} = 2x \tag{1}$$

$$\dot{x} = 2x + u \tag{2}$$

First one is autonomous and unstable. Second one is not autonomous, and we won't know whether or not it is stable, until we know what u is.

If we pick $u = 0$, the result is an unstable equation. But we can also pick u such that the resulting dynamics is stable, such as $u = -3x$:

$$\dot{x} = 2x + u = 2x - 3x = -x \tag{3}$$

So, we can use *control input* u to change stability of the system!

Definition

The problem of finding control law $\mathbf{u}$ that make a certain solution $\mathbf{x}^*$ of dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ stable is called *stabilizing control problem*

This is true for both linear and non-linear systems. But for linear systems we can get a lot more details about this problem, if we restrict our choice of control law.

Consider an LTI system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4}$$

and let us chose *control as a linear function of the state* x:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{5}$$

Thus, we know how the system is going to look when the control is applied:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} \tag{6}$$

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \tag{7}$$

Note that (7) is an autonomous system. We call this a *closed loop* system.

Observing the system $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}$ we obtained, we can notice that we already have the tools to analyse its stability:

Stability condition for LTI closed-loop system

The real parts of the eigenvalues of the matrix $(\mathbf{A} - \mathbf{BK})$ should be negative for asymptotic stability, or non-positive for stability in the sense of Lyapunov.

Hurwitz matrix

If square matrix $\mathbf{M}$ has eigenvalues with strictly negative real parts, it is called Hurwitz. We will denote it as $\mathbf{M} \in \mathcal{H}$.

So, all you need to do is to find such $\mathbf{K}$ that $(\mathbf{A} - \mathbf{BK})$ is Hurwitz, and you made a an asymptotically stable closed-loop system!

We don't have to limit ourselves to just this $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and $\mathbf{u} = -\mathbf{K}\mathbf{x}$ pair.

In fact, this pair mostly works for the simple case when the solution we want to stabilize is trivial $\mathbf{x}^*(t) = 0$.

Let us consider a slightly more complicated system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{c} \tag{8}$$

This is called *affine system*, because of the constant term $\mathbf{c}$. What is the control that stabilizes this system? Let us propose an *affine control law*:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{u}^* \tag{9}$$

where $\mathbf{u}^*$ is a constant term.

Thus, from $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{c}$ and $\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{u}^*$ we get the following closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} + \mathbf{B}\mathbf{u}^* + \mathbf{c} \tag{10}$$

And as long as we can choose such $\mathbf{u}^*$ that $\mathbf{B}\mathbf{u}^* = -\mathbf{c}$, we will get back to the previously seen form $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$.

Existence of the stabilizing control

Same as it is possible that there exists no such $\mathbf{K}$ that $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$ is stable, there might exist no such $\mathbf{u}^*$ that $\mathbf{B}\mathbf{u}^* = -\mathbf{c}$

Let us now consider an arbitrary solution $\mathbf{x}^* = \mathbf{x}^*(t)$ for the linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{11}$$

and try to find a stabilizing control for it.

Our first step is to notice that, if $\mathbf{x}^* = \mathbf{x}^*(t)$ is a solution, that means that it satisfies the ODE (11):

$$\dot{\mathbf{x}}^* = \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^* \tag{12}$$

where $\mathbf{u}^* = \mathbf{u}^*(t)$ is some control law, for which the solution $\mathbf{x}^* = \mathbf{x}^*(t)$ is obtained.

If we are not given $\mathbf{u}^* = \mathbf{u}^*(t)$, we can compute it as:

$$\mathbf{u}^* = \mathbf{B}^+(\dot{\mathbf{x}}^* - \mathbf{A}\mathbf{x}^*) \tag{13}$$

where $\mathbf{B}^+$ is a pseudo-inverse, and the solution to this least-squared problem will have to have no residual (since $\mathbf{x}^* = \mathbf{x}^*(t)$ is a solution).

$$||\dot{\mathbf{x}}^* - \mathbf{A}\mathbf{x}^* - \mathbf{B}\mathbf{B}^+(\dot{\mathbf{x}}^* - \mathbf{A}\mathbf{x}^*)|| = 0 \tag{14}$$

Now, let us introduce the concept of *control error* $\mathbf{e}$:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}^* \tag{15}$$

Control error and stability

If control error goes to zero asymptotically, every solution goes to $\mathbf{x}^*$.

Remember that we have two simultaneous equations: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and $\dot{\mathbf{x}}^* = \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^*$. We can now subtract one from the other to get:

$$\dot{\mathbf{x}} - \dot{\mathbf{x}}^* = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u} - \mathbf{B}\mathbf{u}^* \tag{16}$$

in other words:

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{v} \tag{17}$$

where $\mathbf{v} = \mathbf{u} - \mathbf{u}^*$

We arrived at a new dynamical system $\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{v}$, which is an LTI, and we are interested in stabilizing the solution $\mathbf{e}^* = 0$. We know how to do it with a linear control law:

$$\mathbf{v} = -\mathbf{K}\mathbf{e} \tag{18}$$

Now remember that $\mathbf{v} = \mathbf{u} - \mathbf{u}^*$ and $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$, this will become:

$$\mathbf{u} = -\mathbf{K}(\mathbf{x} - \mathbf{x}^*) + \mathbf{u}^* \tag{19}$$

This control law $\mathbf{u} = -\mathbf{K}(\mathbf{x} - \mathbf{x}^*) + \mathbf{u}^*$ can be thought of as consisting of two parts:

- Feedback control $\mathbf{u}_{FB} = -\mathbf{K}(\mathbf{x} - \mathbf{x}^*)$, which depends on the control error (which requires a feedback about the current state of your system)
- Feed-forward control $\mathbf{u}_{FF} = \mathbf{u}^*$, which depends only on the trajectory and the equations of dynamics of your system, but not on your current state

What we just did - stabilization of the arbitrary trajectory $\mathbf{x}^* = \mathbf{x}^*(t)$ - is also called *trajectory tracking control*, or *trajectory stabilization*. The solution we stabilized is called *trajectory*.

Just for completeness, let's consider the system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{c} \tag{20}$$

and stabilize trajectory $\mathbf{x}^* = \mathbf{x}^*(t)$.

We start by observing that, as before, our solution gives us equality:

$$\dot{\mathbf{x}}^* = \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^* + \mathbf{c} \tag{21}$$

and after introducing control error and subtracting (21) from the original dynamics (20), we get:

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{v} \tag{22}$$

where $\mathbf{v} = \mathbf{u} - \mathbf{u}^*$, which we already saw before. The only difference is that now $\mathbf{u}^*$ is found as:

$$\mathbf{u}^* = \mathbf{B}^+(\dot{\mathbf{x}}^* - \mathbf{A}\mathbf{x}^* - \mathbf{c}) \tag{23}$$

What if we want to move our system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ from initial condition to some desired state $\mathbf{x}^*$. This can be called *point-to-point control*.

This is the same as trajectory control with $\mathbf{x}^* = \text{const}$ and $\dot{\mathbf{x}}^* = 0$.

Read more

- Richard M. Murray Control and Dynamical Systems California Institute of Technology Optimization-Based Control
- Dynamic Simulation in Python

Thank you!

Lecture slides are available via Moodle.

You can help improve these slides at:
github.com/SergeiSa/Control-Theory-Slides-Spring-2021

Check Moodle for additional links, videos, textbook suggestions.