EECS 16B Designing Information Systems and Devices II UC Berkeley Spring 2024

Note 9: Introduction to Controls

Starting from this note, the material starts to become a little more mathematically sophisticated. Here are some tips to read more mathematically dense notes (this really applies to papers as well).

- Skim through the note first, get an outline of the main ideas and results covered, then go through it again with an eye to detail for proofs.
- When reading carefully, work out some of the intermediate steps of proofs and calculations yourself using pencil and paper.
- Take notes on the notes! It really helps your working memory to have a shortlisted copy of all the important stuff within easy access.
- Last but not least, *don't get discouraged if it takes you a while to understand the notes!* For a mathematical text, reading a handful of pages an hour carefully is a very robust pace.

We will use the notation $\mathbb{R}_+$ to mean the set of non-negative real numbers, and the notation $\mathbb{N} = \{0, 1, 2, \ldots\}$ to mean the set of natural numbers.

1 Controls Overview

Recall that one of the big-picture motivations for the course is to understand what it takes to build systems that interact with the real world. Interacting with the real world to achieve our objectives is the subject of the field of *control*. Control is one of the foundational disciplines for robotics and artificial intelligence, and is also very critical for many other areas in EECS such as high performance computing and power systems.

Key Idea 1
The main idea of model-based control is that we leverage an explicit model for the physical system we're concerned with in order to plan and execute our actions.

Definition 2 (State, Control Input)

- The *state* $\vec{x}$ represents the collection of variables we care about.
- The *control input* $\vec{u}$ is the collection of variables that a controller is able to change in order to push the state where we want.

Definition 3 (Control Model)
A control model is a rule that

- takes in: (1) an initial condition for the state, and (2) all control inputs over time;
- and produces: a *state trajectory*, i.e., all states achieved over time.

Model 4 (Discrete-Time LTI Difference Equation Model)
The model is of the form

$$\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i] \quad (1)$$
$$\vec{x}[0] = \vec{x}_0 \quad (2)$$

for $\vec{x} \colon \mathbb{N} \to \mathbb{R}^n$ the state vector as a function of timestep, $\vec{u} \colon \mathbb{N} \to \mathbb{R}^m$ the control inputs as a function of timestep, and $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ matrices.

Model 5 (Continuous-Time LTI Differential Equation Model)
The model is of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \quad (3)$$
$$\vec{x}(0) = \vec{x}_0 \quad (4)$$

for $\vec{x} \colon \mathbb{R}_+ \to \mathbb{R}^n$ the state vector as a function of time, $\vec{u} \colon \mathbb{R}_+ \to \mathbb{R}^m$ the control inputs as a function of time, and $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ matrices.

NOTE: We use square brackets in discrete-time (instead of parentheses) to differentiate between discrete-time functions of timestep (i.e., sequences), and continuous-time functions of time.

Solving the differential equation or difference equation will tell us the state trajectory as a function of the initial condition and inputs. In some cases, we will be able to find an explicit closed-form solution. In other cases, we may need to determine the state trajectory using numerical methods.

1.1 An Aside on the Matrix Exponential

Definition 6 (Matrix Exponential)
Let $X \in \mathbb{R}^{n\times n}$ be a *square* matrix. Then the *matrix exponential* of X is defined as

$$\mathrm{e}^X = I_n + X + \frac{X^2}{2} + \frac{X^3}{6} + \cdots = \sum_{i=0}^{\infty} \frac{X^i}{i!}. \quad (5)$$

To calculate e^{At}, we can compute eq. (5) with $X = A \cdot t$.

It is insightful to compare eq. (5) with the Maclaurin series (i.e. Taylor series centered at 0) of the usual scalar exponential function:

$$\mathrm{e}^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots = \sum_{i=0}^{\infty} \frac{x^i}{i!}. \quad (6)$$

Lemma 7 (Properties of Matrix Exponential)

- If $X = \operatorname{diag}(x_{11}, x_{22}, \ldots, x_{nn})$, then

$$e^X = \begin{bmatrix} e^{x_{11}} & & & \\ & e^{x_{22}} & & \\ & & \ddots & \\ & & & e^{x_{nn}} \end{bmatrix}. \tag{7}$$

- If $X = VYV^{-1}$ is a change of basis of X, then

$$e^X = Ve^YV^{-1}. \tag{8}$$

- The function $\vec{x}(t) := e^{At}\vec{x}_0$ where $A = \operatorname{diag}(a_{11}, a_{22}, \ldots, a_{\mathrm{nn}})$ is the solution to the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) \tag{9}$$

$$\vec{x}(0) = \vec{x}_0. \tag{10}$$

Concept Check: Prove lemma 7.

2 Solving the Discrete-Time LTI Difference Equation Model

We will start with the discrete-time LTI difference equation model. Given an initial condition $\vec{x}_0$ and a sequence of inputs $\vec{u}$, we would like to compute $\vec{x}[i]$ for every i.

Theorem 8 (State Trajectory in Discrete-Time LTI Difference Equation Model)
In the discrete-time LTI difference equation model, we have

$$\vec{x}[i] = A^i\vec{x}_0 + \sum_{k=0}^{i-1} A^{i-k-1}B\vec{u}[k]. \tag{11}$$

Proof. We start with the fact that eq. (1) looks like a recursion, because to compute $\vec{x}[i+1]$ we require only the most recent state $\vec{x}[i]$, as well as the most recent input $\vec{u}[i]$. Motivated by this, we can try to unroll the recursion, and get

$$\begin{aligned}
\vec{x}[i] &= A\vec{x}[i-1] + B\vec{u}[i-1] && (12)\\
&= A(A\vec{x}[i-2] + B\vec{u}[i-2]) + B\vec{u}[i-1] && (13)\\
&= A^2\vec{x}[i-2] + AB\vec{u}[i-2] + B\vec{u}[i-1] && (14)\\
&= A^2(A\vec{x}[i-3] + B\vec{u}[i-3]) + AB\vec{u}[i-2] + B\vec{u}[i-1] && (15)\\
&= A^3\vec{x}[i-3] + A^2B\vec{u}[i-3] + AB\vec{u}[i-2] + B\vec{u}[i-1] && (16)\\
&= \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots && (17)\\
&= A^{i-1}(A\vec{x}[0] + B\vec{u}[0]) + \sum_{k=1}^{i-1} A^{i-k-1}B\vec{u}[k] && (18)
\end{aligned}$$

$$= A^i \vec{x}[0] + \sum_{k=0}^{i-1} A^{i-k-1} B\vec{u}[k] \tag{19}$$

$$= A^i \vec{x}_0 + \sum_{k=0}^{i-1} A^{i-k-1} B\vec{u}[k]. \tag{20}$$

This gives us the state trajectory of the discrete-time LTI difference equation model. Note that in the case where starting value of $k = 0 > i - 1$, the summation becomes an "empty sum" as no value of k satisfies the condition (i.e. the summation evaluates to zero in that case). You can also think of it as a "for" loop where the start value is larger than the end value.

Concept Check: Check that the trajectory given by eq. (11) is correct, in the sense that it obeys the difference eq. (1) and initial condition eq. (2). □

3 Solving the Continuous-Time LTI Differential Equation Model

We now study the Continuous-Time LTI Differential Equation Model. Given an initial condition $\vec{x}_0$ and all inputs $\vec{u}$, we would like to compute $\vec{x}(t)$ for every t.

As you might imagine, it is not possible to use the recursion approach for this model. Instead, we will rely on our knowledge of differential equations that we have already covered, and use it to extend the solution to matrices.

3.1 Scalar Model

In the scalar case, the Continuous-Time LTI Differential Equation Model becomes the following:

Model 9 (Scalar Continuous-Time LTI Differential Equation Model)
The model is of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t) + bu(t) \tag{21}$$

$$x(0) = x_0 \tag{22}$$

for $x\colon \mathbb{R}_+ \to \mathbb{R}$ the state scalar as a function of time, $u\colon \mathbb{R}_+ \to \mathbb{R}$ the control input as a function of time, and $a \in \mathbb{R}, b \in \mathbb{R}$ scalar coefficients.

Theorem 10 (State Trajectory in Scalar Continuous-Time LTI Differential Equation Model)
In the scalar continuous-time LTI differential equation model, we have

$$x(t) = \mathrm{e}^{at} x_0 + \int_0^t \mathrm{e}^{a(t-\tau)} \cdot bu(\tau)\, \mathrm{d}\tau\,. \tag{23}$$

Proof. The first thing to note is that eqs. (21) and (22) are reminiscent of a differential equation we already know how to solve! Recall that we showed that the (unique) solution to the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \lambda x(t) + u(t) \tag{24}$$

$$x(0) = x_0 \tag{25}$$

is given by

$$x(t) = e^{\lambda t}x_0 + \int_0^t e^{\lambda(t-\tau)}u(\tau)\,d\tau\,. \tag{26}$$

But eqs. (21) and (24) look really similar. In fact, if we replace "λ" with "a" and replace "$u(t)$" by "$bu(t)$", they are exactly the same. Thus, the solution to eqs. (21) and (22) is exactly the same as eq. (26) after we make those replacements. So we have

$$x(t) = e^{at}x_0 + \int_0^t e^{a(t-\tau)} \cdot bu(\tau)\,d\tau\,. \tag{27}$$

This gives us the trajectory of the scalar continuous-time LTI differential equation model. □

So, given an input function u and an initial condition x_0, we can solve for the state trajectory x using the formula eq. (23).

3.2 Diagonal State Matrix Case

The next step after solving one scalar differential equation is to solve n independent differential equations. This is accomplished whenever the state matrix is diagonal. Indeed, suppose Λ is the state matrix, say with entries $\lambda_{11}, \lambda_{22}, \ldots, \lambda_{nn}$ going from top-left to bottom-right (we write this as $\Lambda = \operatorname{diag}(\lambda_{11}, \lambda_{22}, \ldots, \lambda_{nn})$). This gives us the model:

Model 11 (Diagonal Continuous-Time LTI Differential Equation Model)
The model is of the form

$$\frac{d}{dt}\vec{x}(t) = \Lambda\vec{x}(t) + B\vec{u}(t) \tag{28}$$

$$\vec{x}(0) = \vec{x}_0 \tag{29}$$

for $\vec{x}\colon \mathbb{R}_+ \to \mathbb{R}^n$ the state vector as a function of time, $\vec{u}\colon \mathbb{R}_+ \to \mathbb{R}^m$ the control input as a function of time, $B \in \mathbb{R}^{n\times m}$ a matrix, and $\Lambda := \operatorname{diag}(\lambda_{11}, \lambda_{22}, \ldots, \lambda_{nn}) \in \mathbb{R}^{n\times n}$ a diagonal matrix.

Theorem 12 (State Trajectory in Diagonal Continuous-Time LTI Differential Equation Model)
In the diagonal continuous-time LTI differential equation model, we have

$$\vec{x}(t) = e^{\Lambda t}\vec{x}_0 + \int_0^t e^{\Lambda(t-\tau)}B\vec{u}(\tau)\,d\tau\,, \tag{30}$$

where we use the notation

$$e^{\Lambda t} := \begin{bmatrix} e^{\lambda_{11}t} & & & \\ & e^{\lambda_{22}t} & & \\ & & \ddots & \\ & & & e^{\lambda_{nn}t} \end{bmatrix} \tag{31}$$

for $\Lambda = \operatorname{diag}(\lambda_{11}, \lambda_{22}, \ldots, \lambda_{nn})$ a *square diagonal* matrix.

NOTE: $e^{\Lambda t}$ is a square matrix of the same size as Λ.

Proof. We can rewrite eq. (28) as

$$\frac{d}{dt}\vec{x}(t) = \Lambda\vec{x}(t) + B\vec{u}(t) \tag{32}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_{11} & & & \\ & \lambda_{22} & & \\ & & \ddots & \\ & & & \lambda_{nn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} (B\vec{u}(t))_1 \\ (B\vec{u}(t))_2 \\ \vdots \\ (B\vec{u}(t))_n \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t}x_1(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}x_2(t) \\ \vdots \\ \frac{\mathrm{d}}{\mathrm{d}t}x_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_{11}x_1(t) + (B\vec{u}(t))_1 \\ \lambda_{22}x_2(t) + (B\vec{u}(t))_2 \\ \vdots \\ \lambda_{nn}x_n(t) + (B\vec{u}(t))_n \end{bmatrix}. \tag{34}$$

Writing each row of the vector separately as its own equation, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}x_1(t) = \lambda_{11}x_1(t) + (B\vec{u}(t))_1 \tag{35}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}x_2(t) = \lambda_{22}x_2(t) + (B\vec{u}(t))_2 \tag{36}$$
$$\vdots \qquad\qquad \vdots \tag{37}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}x_n(t) = \lambda_{nn}x_n(t) + (B\vec{u}(t))_n \tag{38}$$

We know how to solve each of these equations; along with the initial conditions, which are seen from eq. (29) to be

$$x_1(0) = (\vec{x}_0)_1 \tag{39}$$
$$x_2(0) = (\vec{x}_0)_2 \tag{40}$$
$$\vdots \qquad \vdots \tag{41}$$
$$x_n(0) = (\vec{x}_0)_n \tag{42}$$

Then each equation is of the form that we solved in section 3.1. Hence we can solve each equation to get

$$x_1(t) = \mathrm{e}^{\lambda_{11}t}(\vec{x}_0)_1 + \int_0^t \mathrm{e}^{\lambda_{11}(t-\tau)}(B\vec{u}(\tau))_1\,\mathrm{d}\tau \tag{43}$$
$$x_2(t) = \mathrm{e}^{\lambda_{22}t}(\vec{x}_0)_2 + \int_0^t \mathrm{e}^{\lambda_{22}(t-\tau)}(B\vec{u}(\tau))_2\,\mathrm{d}\tau \tag{44}$$
$$\vdots \qquad\qquad \vdots \tag{45}$$
$$x_n(t) = \mathrm{e}^{\lambda_{nn}t}(\vec{x}_0)_n + \int_0^t \mathrm{e}^{\lambda_{nn}(t-\tau)}(B\vec{u}(\tau))_n\,\mathrm{d}\tau\,. \tag{46}$$

Stacking each scalar equation to get a vector solution, we get

$$\vec{x}(t) = \begin{bmatrix} \mathrm{e}^{\lambda_{11}t}(\vec{x}_0)_1 + \int_0^t \mathrm{e}^{\lambda_{11}(t-\tau)}(B\vec{u}(\tau))_1\,\mathrm{d}\tau \\ \mathrm{e}^{\lambda_{22}t}(\vec{x}_0)_2 + \int_0^t \mathrm{e}^{\lambda_{22}(t-\tau)}(B\vec{u}(\tau))_2\,\mathrm{d}\tau \\ \vdots \\ \mathrm{e}^{\lambda_{nn}t}(\vec{x}_0)_n + \int_0^t \mathrm{e}^{\lambda_{nn}(t-\tau)}(B\vec{u}(\tau))_n\,\mathrm{d}\tau \end{bmatrix} \tag{47}$$

$$= \begin{bmatrix} e^{\lambda_{11}t}(\vec{x}_0)_1 \\ e^{\lambda_{22}t}(\vec{x}_0)_2 \\ \vdots \\ e^{\lambda_{nn}t}(\vec{x}_0)_n \end{bmatrix} + \int_0^t \begin{bmatrix} e^{\lambda_{11}(t-\tau)}(B\vec{u}(\tau))_1 \\ e^{\lambda_{22}(t-\tau)}(B\vec{u}(\tau))_2 \\ \vdots \\ e^{\lambda_{nn}(t-\tau)}(B\vec{u}(\tau))_n \end{bmatrix} d\tau\,. \tag{48}$$

Now to simplify to the form we want, we use the definition of matrix multiplication to get

$$\begin{bmatrix} e^{\lambda_{11}t}(\vec{x}_0)_1 \\ e^{\lambda_{22}t}(\vec{x}_0)_2 \\ \vdots \\ e^{\lambda_{nn}t}(\vec{x}_0)_n \end{bmatrix} = \begin{bmatrix} e^{\lambda_{11}t} & & & \\ & e^{\lambda_{22}t} & & \\ & & \ddots & \\ & & & e^{\lambda_{nn}t} \end{bmatrix} \begin{bmatrix} (\vec{x}_0)_1 \\ (\vec{x}_0)_2 \\ \vdots \\ (\vec{x}_0)_n \end{bmatrix} = e^{\Lambda t}\vec{x}_0 \tag{49}$$

and

$$\begin{bmatrix} e^{\lambda_{11}(t-\tau)}(B\vec{u}(\tau))_1 \\ e^{\lambda_{22}(t-\tau)}(B\vec{u}(\tau))_2 \\ \vdots \\ e^{\lambda_{nn}(t-\tau)}(B\vec{u}(\tau))_n \end{bmatrix} = \begin{bmatrix} e^{\lambda_{11}(t-\tau)} & & & \\ & e^{\lambda_{22}(t-\tau)} & & \\ & & \ddots & \\ & & & e^{\lambda_{nn}(t-\tau)} \end{bmatrix} \begin{bmatrix} (B\vec{u}(\tau))_1 \\ (B\vec{u}(\tau))_2 \\ \vdots \\ (B\vec{u}(\tau))_n \end{bmatrix} = e^{\Lambda(t-\tau)}B\vec{u}(\tau). \tag{50}$$

Using this to simplify eq. (48), we get

$$e^{\Lambda t}\vec{x}_0 + \int_0^t e^{\Lambda(t-\tau)}B\vec{u}(\tau)\,d\tau \tag{51}$$

as desired. □

So, given an input function $\vec{u}$ and an initial condition $\vec{x}_0$, we can solve for the state trajectory $\vec{x}$ using the formula eq. (30).

3.3 A Diagonalizable Model

Now that we have solved the case of n independent differential equations, let us now see what happens if the differential equations are only independent after changing to a desirable basis. That is – let us suppose that the A matrix can be written as $A = V\Lambda V^{-1}$ for some given diagonal matrix Λ and invertible matrix V. Note that this is equivalent to A being diagonalizable, i.e., having n linearly independent eigenvectors; in this case Λ is actually the matrix of eigenvalues and V is the matrix of eigenvectors of A.

This yields the model:

Model 13 (Diagonalizable Continuous-Time LTI Differential Equation Model)
The model is of the form

$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \tag{52}$$

$$\vec{x}(0) = \vec{x}_0 \tag{53}$$

for $\vec{x}\colon \mathbb{R}_+ \to \mathbb{R}^n$ the state vector as a function of time, $\vec{u}\colon \mathbb{R}_+ \to \mathbb{R}^m$ the control input as a function of time, $B \in \mathbb{R}^{n\times m}$ a matrix, and $A \in \mathbb{R}^{n\times n}$ a diagonalizable matrix with diagonalization $A = V\Lambda V^{-1}$.

Theorem 14 (State Trajectory in Diagonalizable Continuous-Time LTI Differential Equation Model)
The general solution for the continuous-time LTI differential equation model in the vector case is reminiscent of the form of the solution in the scalar case eq. (23):

$$\vec{x}(t) = \mathrm{e}^{At}\vec{x}_0 + \int_0^t \mathrm{e}^{A(t-\tau)}B\vec{u}(\tau)\,\mathrm{d}\tau \tag{54}$$

In particular, when the state matrix A is diagonalizable, the solution can be diagonalized to be written as:

$$\vec{x}(t) = V\mathrm{e}^{\Lambda t}V^{-1}\vec{x}_0 + \int_0^t V\mathrm{e}^{\Lambda(t-\tau)}V^{-1}B\vec{u}(\tau)\,\mathrm{d}\tau\,, \tag{55}$$

Proof. We can rewrite eq. (52) as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) &= A\vec{x}(t) + B\vec{u}(t) && (56)\\ &= V\Lambda V^{-1}\vec{x}(t) + B\vec{u}(t). && (57)\end{aligned}$$

Now we can use the change-of-basis that was introduced when we first solved vector differential equations. Namely, let $\vec{z}(t) := V^{-1}\vec{x}(t)$. Then

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) &= V\Lambda V^{-1}\vec{x}(t) + B\vec{u}(t) && (58)\\ V^{-1}\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) &= V^{-1}V\Lambda V^{-1}\vec{x}(t) + V^{-1}B\vec{u}(t) && (59)\\ \frac{\mathrm{d}}{\mathrm{d}t}\left(V^{-1}\vec{x}(t)\right) &= \Lambda V^{-1}\vec{x}(t) + V^{-1}B\vec{u}(t) && (60)\\ \frac{\mathrm{d}}{\mathrm{d}t}\vec{z}(t) &= \Lambda\vec{z}(t) + V^{-1}B\vec{u}(t) && (61)\end{aligned}$$

Thus, after changing the basis for the initial condition eq. (53) by setting $\vec{z}_0 = V^{-1}\vec{x}_0$, we have the vector system

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\vec{z}(t) &= \Lambda\vec{z}(t) + V^{-1}B\vec{u}(t) && (62)\\ \vec{z}(0) &= \vec{z}_0. && (63)\end{aligned}$$

This is a system of differential equations where the A matrix – in this case Λ – is diagonal, so this is exactly the form of equation we solved in section 3.2! Thus, we can solve eqs. (62) and (63) to get a function $\vec{z}(t)$:

$$\vec{z}(t) = \mathrm{e}^{\Lambda t}\vec{z}_0 + \int_0^t \mathrm{e}^{\Lambda(t-\tau)}V^{-1}B\vec{u}(\tau)\,\mathrm{d}\tau \tag{64}$$

To finish the proof, it remains to convert back to the usual basis of x coordinates (as opposed to the z coordinates which the solution is in).

$$\begin{aligned}\vec{x}(t) &= V\vec{z}(t) && (65)\\ &= V\mathrm{e}^{\Lambda t}\vec{z}_0 + \int_0^t V\mathrm{e}^{\Lambda(t-\tau)}V^{-1}B\vec{u}(\tau)\,\mathrm{d}\tau && (66)\\ &= V\mathrm{e}^{\Lambda t}V^{-1}\vec{x}_0 + \int_0^t V\mathrm{e}^{\Lambda(t-\tau)}V^{-1}B\vec{u}(\tau)\,\mathrm{d}\tau && (67)\end{aligned}$$

as desired. □

And our diagonalized solution for the vector case written in terms of A:

$$\vec{x}(t) = e^{At}\vec{x}_0 + \int_0^t e^{A(t-\tau)} B\vec{u}(\tau)\, d\tau \tag{68}$$

is simply analogous to the solution in the scalar case eq. (23).

A summary of the above proof is captured by the familiar change of coordinates diagram:

$$\begin{tikzcd}
\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \arrow[r, squiggly, "\text{Too difficult}"'] \arrow[d, "\substack{\text{Change of}\\ \text{variables to get}\\ \text{a diagonal system}}"'] & \vec{x}(t) = \dots \\
\frac{d}{dt}\vec{z}(t) = \Lambda\vec{z}(t) + V^{-1}B\vec{u}(t) \arrow[r, "\substack{\text{Solve a}\\ \text{diagonal system}}"'] & \vec{z}(t) = \dots \arrow[u, "\substack{\text{Undo change}\\ \text{of variables}}"']
\end{tikzcd}$$

3.4 Beyond Diagonalizable A

Not all matrices are diagonalizable, so it is necessary to handle the case where we cannot perform the decomposition we used to get eqs. (62) and (63). Unfortunately, there is no neat closed form, such as the one we found in theorem 14.

The primary way to handle this case is to use *upper triangularization* instead of diagonalization. Upper triangularization provides a change of basis to an upper triangular matrix (instead of to a diagonal matrix), i.e., we write $A = UTU^{-1}$ where T is upper triangular. Once in the z basis, the differential equations we get will not be strictly independent; however, each equation will be solvable without needing any information from the equations above it. So we solve the equations bottom-to-top and back-substitute solutions to lower equations into upper equations. Once we have solved all the equations in the z basis, we convert back to the x basis.

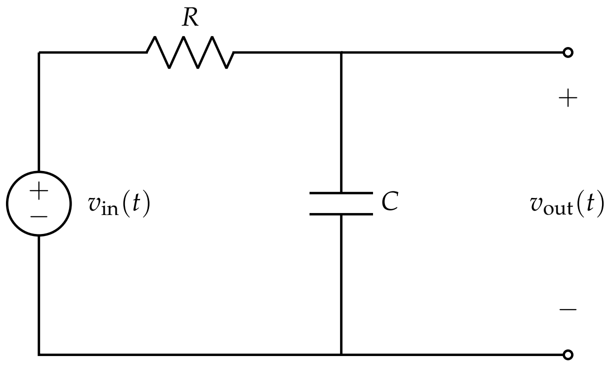
The process is captured in the modification to the change-of-coordinates diagram:

$$\begin{tikzcd}
\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \arrow[r, squiggly, "\text{Too difficult}"'] \arrow[d, "\substack{\text{Change of}\\ \text{variables to get an}\\ \text{upper triangular system}}"'] & \vec{x}(t) = \dots \\
\frac{d}{dt}\vec{z}(t) = T\vec{z}(t) + U^{-1}B\vec{u}(t) \arrow[r, "\substack{\text{Solve an upper}\\ \text{triangular system}}"'] & \vec{z}(t) = \dots \arrow[u, "\substack{\text{Undo change}\\ \text{of variables}}"']
\end{tikzcd}$$

4 Examples

4.1 Continuous-Time LTI Model Example

Consider the following familiar RC circuit:



Let's say we wanted to model $v_{\text{out}}(t)$. We can set up the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} v_{\text{out}}(t) = \frac{v_{\text{in}}(t) - v_{\text{out}}(t)}{RC} \tag{69}$$

$$v_{\text{out}}(0) = V_0 \tag{70}$$

Let's say we have the ability to generate any $v_{\text{in}}(t)$ we would like via a controller device. This is an explicit instance of a scalar continuous-time LTI differential equation model, which we covered in section 3.1, with the variables $A = -\frac{1}{RC}$, $B = \frac{1}{RC}$ being scalars. We can read off the solution as

$$v_{\text{out}}(t) = \mathrm{e}^{-\frac{1}{RC}t} V_0 + \frac{1}{RC} \int_0^t \mathrm{e}^{-\frac{1}{RC}(t-\tau)} v_{\text{in}}(\tau)\, \mathrm{d}\tau\,. \tag{71}$$

This is interesting to study as a control model if, for instance, we want to set the trajectory of $v_{\text{out}}(t)$ to, for example, be a decaying exponential function with a desired decay rate.

4.2 Discrete-Time LTI Model Example

Let's introduce a concrete example of modeling a real-world system using the Discrete-Time LTI Difference Equation Model. In lab, we are seeing how this model can be applied to our autonomous car, but the ideas used can easily generalize to other systems: let's consider a helicopter.

If we wish to model the helicopter's current state as the variable $\vec{s}$, then we need to capture at minimum both the helicopter's position and heading (angle relative to true north). This can be given as:

$$\vec{s} = \begin{bmatrix} x \\ y \\ z \\ \theta \end{bmatrix} \tag{72}$$

x, y and z determine the helicopter's 3D position, and θ gives the helicopter's heading. And our Discrete-Time LTI Difference Equation Model would be given as:

$$\vec{s}[i+1] = A\vec{s}[i] + B\vec{u}[i] \tag{73}$$

With solution:

$$\vec{s}[i] = A^i \vec{s}_0 + \sum_{k=0}^{i-1} A^{i-k-1} B\vec{u}[k]. \tag{74}$$

Besides, the state variable, here is what the other elements of the Discrete-Time LTI Difference Equation Model would represent.

- **State Transition Matrix A:** Captures the physics of the helicopter's movement. It dictates how the state changes in one time step due to the helicopter's inherent dynamics, assuming no external inputs are applied.
- **Input Matrix B:** Describes how the helicopter's state can be altered through control inputs.
- **Control Input $\vec{u}$:** These are the adjustments made by the control system to the helicopter's state. This is the part of the model we as engineers are fully able to determine ourselves. In a physical sense this can either be an automated or manual (human) controller that influences the helicopter's state via its transmission. In future notes, we will see how to select inputs to always keep the system stable and select particular inputs to arrive at a desired state.
- **Initial State $\vec{s}_0$:** The starting point for the helicopter's operation. In a practical case, we may arbitrarily set this to the zero vector or base the values off of some predetermined coordinate system and direction.

This model allows for the prediction and control of the helicopter's behavior over time. In future notes, we will learn methods based on the Discrete-Time LTI Difference Equation Model to enable us to complete tasks such as stabilization against disturbances like wind or autonomous navigation.

5 (OPTIONAL) A Look Ahead

There are many problems we can tackle once we have fixed a model, even after we know the state trajectory given the inputs and initial conditions. Examples of such problems are:

- Suppose we have a continuous-time model for the system, but we want to implement a digital controller. Since our controller doesn't have infinite memory and infinite computing power, it can only read to/modify the system at discrete timesteps, effectively forcing us to use a discrete-time model. So, given a continuous-time model, we convert it to an approximate discrete-time model and use that to implement the controller. The process of coming up with a discrete-time model that approximates a continuous-time model is called *discretization*.
- Suppose we have a discrete-time model (respectively, a continuous-time model) for the system, but we don't know the matrices A and B. We also have data from the system; specifically, we have $(\vec{x}[i], \vec{u}[i], \vec{x}[i+1])$ for lots of values of i (respectively $(\vec{x}(t), \vec{u}(t), \frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t))$ for lots of values of t). Then we can learn the A and B matrices from data and thereby get a discrete-time model that accurately reflects the system. The process of learning the model parameters A and B from data is called *system identification*.
- Suppose we have a model for the system, and want to know whether the state vector will grow unboundedly over time, and in particular which inputs will cause the state vector to grow unboundedly over time. We generally do not want this to happen in real systems. The process of determining, given a model, under what conditions the state trajectory grows unboundedly is called *stability analysis*

- Suppose we have a model for the system, and want to know whether the state can ever reach some desired state $\vec{x}_d$. If the state is, for example, the position of the robot car, then this is tantamount to asking whether the car can reach a desired position. This property is desirable in general. The process of determining, given a model, if there is a set of inputs that will push the state to a given desired state is called *controllability analysis*.
- Suppose we have a model for the system, and want to find inputs that push the state to a given desired state $\vec{x}_d$ while consuming minimum energy. The process of determining the correct inputs to push the state towards a given desired state while consuming minimum energy is called *minimum energy control*.

6 (OPTIONAL) More on LTI

Back when we started the note, we defined some models with the "linear time-invariant" (LTI) property. This property is really two properties which can apply to a model: *linearity* and *time-invariance*. To describe the definitions of these properties, we will introduce a new notation which makes our description easier.

More particularly, recall that we said a model is a rule which takes in an initial state $\vec{x}_0$ and a control input $\vec{u}$ (a function of time), and returns a state function $\vec{x}$ (again a function of time). We will say in this case that $(\vec{x}_0, \vec{u}) \to \vec{x}$ is an input-output pair of the model.

Now we are ready to give our definitions.

Definition 15 (Linearity)
A model is *linear* if for every $c_1, c_2 \in \mathbb{R}$ and any input-output pairs

$$(\vec{x}_{0;1}, \vec{u}_1) \to \vec{x}_1 \qquad \text{and} \qquad (\vec{x}_{0;2}, \vec{u}_2) \to \vec{x}_2 \tag{75}$$

then the model also has the input-output pair

$$(c_1\vec{x}_{0;1} + c_2\vec{x}_{0;2}, c_1\vec{u}_1 + c_2\vec{u}_2) \to c_1\vec{x}_1 + c_2\vec{x}_2. \tag{76}$$

This is the same definition we use for a linear transformation, or a matrix; it is just abstracted to deal with the inputs and outputs being *functions of time*.

Definition 16 (Time-Invariance)
A model is *time-invariant* if for every non-negative time $\tau \geq 0$, and every input-output pair $(\vec{x}_0, \vec{u}) \to \vec{x}$, if we define the notation

$$\vec{\tilde{x}}_0 := \vec{x}(\tau) \qquad \vec{\tilde{x}}(t) := \vec{x}(t+\tau) \qquad \vec{\tilde{u}}(t) := \vec{u}(t+\tau) \tag{77}$$

then the model has the input-output pair $(\vec{\tilde{x}}_0, \vec{\tilde{u}}) \to \vec{\tilde{x}}$. Essentially this means that the model doesn't change its behavior as time goes on.

Note that here we used the continuous-time notation, but this also applies for the discrete-time model. In discrete-time, $\tau \in \mathbb{N}$; in continuous-time, $\tau \in \mathbb{R}_+$.

Theorem 17
The Discrete-Time LTI Difference Equation Model and the Continuous-Time LTI Differential Equation Model are LTI.

Concept Check: Prove theorem 17 by using the solutions to the difference/differential equations (eq. (11) and **??**) to write the state function explicitly in terms of the initial condition and control input, and showing that it is linear.

Intuitively, the Discrete-Time LTI Difference Equation Model and the Continuous-Time LTI Differential Equation Model are linear because the right-hand side of the equations are linear in the current $\vec{x}$ value and the current $\vec{u}$ value, and they are time-invariant because the coefficients A and B don't depend on time.

For example, an example of a discrete-time linear difference equation model which is *not* time invariant (and in particular called time-varying) is:

$$\vec{x}[i+1] = i^2 \cdot \vec{x}[i] + i \cdot \vec{u}[i] \tag{78}$$

$$\vec{x}[0] = \vec{x}_0. \tag{79}$$

And an example of a continuous-time time-invariant differential equation model which is *not* linear is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = \|\vec{x}(t)\|^2\vec{x}(t) + B\vec{u}(t) \tag{80}$$

$$\vec{x}(0) = \vec{x}_0. \tag{81}$$

This is nonlinear because the first term

$$\|\vec{x}(t)\|^2\vec{x}(t) = \left(\sum_{i=1}^{n} x_i(t)^2\right)\vec{x}(t) \tag{82}$$

is at least quadratic in the $x_i(t)$ (and in particular each coordinate is cubic).

7 (OPTIONAL) More on Model-Based Control

We have discussed linear time-invariant models in discrete-time and continuous-time in the main part of the note, and in section 6 we have discussed ways to produce nonlinear and time-varying models. By mixing and matching these categories, we can generate several types of models we won't be using in this course, but are nonetheless practical and useful to study.

As a summary, here is a diagram of models that are studied in model-based control:

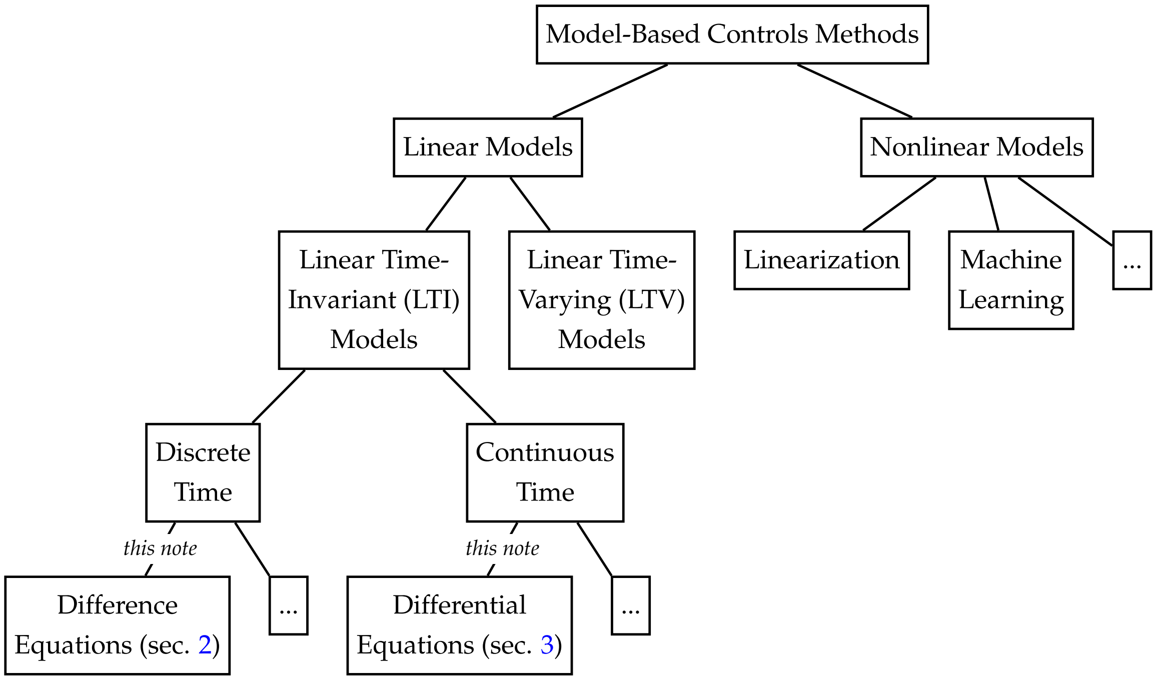

Model-Based Controls Methods
Linear Models
Nonlinear Models
Linear Time-Invariant (LTI) Models
Linear Time-Varying (LTV) Models
Linearization
Machine Learning
...
Discrete Time
Continuous Time
this note
Difference Equations (sec. 2)
...
this note
Differential Equations (sec. 3)
...

Figure 1: Classification of model-based control techniques. Taken from EE 221A Fall 2020 Lecture Note 1.

Contributors:

- Druv Pai.
- Anish Muthali.
- Matteo Guarrera.
- Chancharik Mitra.