ECE504: Lecture 3

D. Richard Brown III

Worcester Polytechnic Institute

16-Sep-2008

Lecture 3 Major Topics

We are finishing up Part I of ECE504: **Mathematical description of systems**

model $\rightarrow$ mathematical description

Today:

1. Discrete-time systems
2. Time varying systems
3. Linearization of smooth nonlinear systems
4. Examples

This concludes Chen Chapter 2. You may also want to begin reading ahead in Chen Chapter 3.

Discrete-Time Systems

Our focus has been a continuous-time systems so far. What about discrete-time systems?

1. Input-output difference equation
2. Transfer function
3. Impulse response
4. State-space description

The same capabilities and limitations apply in the DT case as in the CT case. The tools are slightly different however...

Discrete-Time Input-Output Difference Equation

Single-input, single-output, causal:

$$y[k] = f(y[k-1], y[k-2], \ldots, u[k], u[k-1], \ldots)$$

Example:

$$y[k] = y[k-1] + u[k]$$

What is this?

Example:

$$y[k] = \frac{u[k] + u[k-1] + u[k-2] + u[k-3]}{4}$$

What is this?

For p-input q-output causal systems, we can write:

$$y_i[k] = f(y_i[k-1], y_i[k-2], \ldots, u_1[k], u_1[k-1], \ldots, u_p[k], u_p[k-1], \ldots)$$

for $i = 1, \ldots, q$.

Transfer Function: The One-Sided z-Transform

- Suppose $\boldsymbol{f}[k] : \mathbb{N} \mapsto \mathbb{R}^{q \times p}$ is a matrix valued function of $k = 0, 1, \dots$

$$\boldsymbol{f}[k] = \begin{bmatrix} f_{11}[k] & \dots & f_{1p}[k] \\ \vdots & & \vdots \\ f_{q1}[k] & \dots & f_{qp}[k] \end{bmatrix}$$

- Define the one-sided z-transform of $\boldsymbol{f}[k]$

$$\boxed{\hat{\boldsymbol{f}}(z) = \sum_{k=0}^{\infty} \boldsymbol{f}[k] z^{-k}}$$

where $z \in \mathbb{C}$ and the sum of a matrix is done element by element.

- Notation:

$$\begin{aligned} \hat{\boldsymbol{f}}(z) &= \mathcal{Z}\left[\boldsymbol{f}[k]\right] \\ \boldsymbol{f}[k] &= \mathcal{Z}^{-1}\left[\hat{\boldsymbol{f}}(z)\right] \\ \boldsymbol{f}[k] &\leftrightarrow \hat{\boldsymbol{f}}(z) \end{aligned}$$

Convergence of the One-Sided z-Transform

$$\hat{\boldsymbol{f}}(z) = \sum_{k=0}^{\infty} \boldsymbol{f}[k] z^{-k}$$

As was the case with the Laplace transform integral, we need to be a little bit careful about the convergence of this summation. Let $\Lambda_{i\ell}$ be the set of all $\sigma \in \mathbb{R}_+$ such that

$$\sum_{k=0}^{\infty} |f_{i\ell}[k]| \sigma^{-k} < \infty.$$

If this sum converges for $\sigma = 2$, will it also converge for $\sigma = 3$?

If this sum converges for $\sigma = 2$, will it also converge for $\sigma = 1$?

Convergence of the One-Sided z-Transform (continued)

$\Lambda_{i\ell} = \left\{\sigma \in \mathbb{R}_+ \,:\, \sum_{k=0}^{\infty} |f_{i\ell}[k]|\sigma^{-k} < \infty\right\}$. If

$$\Lambda = \bigcap_{i=1}^{q} \bigcap_{\ell=1}^{p} \Lambda_{i\ell} = \varnothing$$

then the one-sided z-transform doesn't exist for $\boldsymbol{f}[k]$. Otherwise, let $\lambda = \min \Lambda$. The set of complex numbers such that $\{z \in \mathbb{C} \,:\, |z| > \lambda\}$ is called the "region of absolute convergence" of the z-transform of the matrix valued function $\boldsymbol{f}[k]$.

Example: Unit step function.

$$f[k] = \begin{cases} 1 & k = 0, 1, \ldots \\ 0 & \text{otherwise} \end{cases}$$

What is the region of absolute convergence? ________. $\hat{f}(z) =$________.

Another z-Transform Region of Convergence Example

$$\boldsymbol{f}[k] = \begin{cases} \begin{bmatrix} a^k & b^k \\ c^k & d^k \end{bmatrix} & k = 0, 1, \dots \\ 0 & \text{otherwise} \end{cases}$$

What is the region of absolute convergence?

What is the one-sided z-transform?

The Inverse z-Transform

$$\boldsymbol{f}[k] = \frac{1}{2\pi j} \oint \hat{\boldsymbol{f}}(z) z^{k-1}\, dz$$

where $j := \sqrt{-1}$ and the integral is along a counterclockwise closed circular contour in the complex plane, centered at the origin and with radius $r > \lambda$ (in the region of absolute convergence).

- It can be shown that, when the z-transform is a rational function of z, the inverse z-transform can be computed without evaluating this integral (partial fraction expansion).
- In general, this integral is usually not easy to compute.
- Use tables whenever possible.

Discrete-Time Transfer Function

Definition

Given a causal, linear, time-invariant DT system with p input terminals, q output terminals, and relaxed initial conditions at time $k = 0$

$$\left.\begin{array}{l}\boldsymbol{x}[0] = 0 \\ \boldsymbol{u}[k],\ k = 0, 1, 2, \dots\end{array}\right\} \rightarrow \boldsymbol{y}[k],\ k = 0, 1, 2, \dots$$

then the transfer function matrix is defined as

$$\hat{\boldsymbol{g}}(z) \quad := \quad \begin{bmatrix} \hat{g}_{11}(z) & \dots & \hat{g}_{1p}(z) \\ \vdots & & \vdots \\ \hat{g}_{q1}(z) & \dots & \hat{g}_{qp}(z) \end{bmatrix} \text{ where } \hat{g}_{i\ell}(z) := \frac{\hat{y}_i(z)}{\hat{u}_\ell(z)}$$

for $i = 1, \dots, q$ and $\ell = 1, \dots, p$.

The overall system is then $\hat{\boldsymbol{y}}(z) = \hat{\boldsymbol{g}}(z)\hat{\boldsymbol{u}}(z)$.

Discrete-Time Transfer Function and Impulse Response

- The transfer function $\hat{\boldsymbol{g}}(z)$ describes the **zero-state response** of a linear, time-invariant, discrete-time system.
- Let's set the input

$$u[k] = \delta[k] = \begin{cases} 1 & k = 0 \\ 0 & \text{otherwise} \end{cases}$$

What is the z-transform of $\delta[k]$ (and what is the ROAC)?

- *The transfer function (matrix) is the z-transform of the discrete-time impulse response (matrix).*

Rational Transfer Functions

Theorem (stated as a fact, Chen pp.32-33)

If a discrete-time, linear, time-invariant system is lumped, each transfer function in the transfer function matrix is a rational function of z.

This result means that

$$\hat{g}_{i\ell}(z) = \frac{N(z)}{D(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0} \qquad (1)$$

where $N(z)$ and $D(z)$ are polynomials of z.

Note that discrete-time transfer functions are also often written as

$$\hat{g}_{i\ell}(z) = \frac{N'(z)}{D'(z)} = \frac{p_0 + p_1 z^{-1} + \cdots + p_m z^{-m}}{1 + q_1 z^{-1} + \cdots + q_n z^{-n}} \qquad (2)$$

where $N'(z)$ and $D'(z)$ are polynomials in z^{-1}.

Rational Transfer Functions

Are (1) and (2) equivalent?

When $m = n$, it should be clear that (1) and (2) are equivalent since

$$\begin{aligned} N'(z) &= z^{-n} N(z) \\ D'(z) &= z^{-n} D(z) \\ \{p_0, p_1, \dots, p_n\} &= \{b_n, b_{n-1}, \dots, b_0\} \\ \{q_1, q_2, \dots, q_n\} &= \{a_{n-1}, b_{n-2}, \dots, a_0\} \end{aligned}$$

When $m < n$, we can still write (1) as

$$\hat{g}_{i\ell}(z) = \frac{N(z)}{D(z)} = \frac{b_n z^n + b_{n-1} z^{n-1} + \dots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}$$

where the first $n - m$ of the b coefficients are equal to zero. Then we can use the same trick as when $m = n$ to show that (1) and (2) are equivalent.

What about the case $m > n$?

The Three Most Important z-Transforms: #1

$$\boxed{a^k \leftrightarrow \frac{z}{z-a}}$$

where a can be real or complex-valued.

- Easy to show using our trick for computing the sum of a power series.
- Region of absolute convergence: $\{z \in \mathbb{C} \,:\, |z| > |a|\}$.
- Application: For a discrete-time, linear, time-invariant, lumped system, we have

  $$\hat{g}(z) = \frac{N(z)}{D(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

  Compute partial fraction expansion ...

  $$\hat{g}(z) = \frac{c_1 z}{z - d_1} + \cdots + \frac{c_n z}{z - d_n}$$

  Then what can we say about impulse response $g[k]$?
- Note: Repeated roots make this a little bit more complicated.

The Three Most Important z-Transforms: #2

Advance in time:

$$\boxed{g[k+\ell] \leftrightarrow z^{\ell}\hat{g}(z) - z^{\ell}g[0] - z^{\ell-1}g[1] - \cdots - zg[\ell-1]}$$

Delay in time:

$$\boxed{g[k-\ell] \leftrightarrow z^{-\ell}\hat{g}(z) + z^{-\ell+1}g[-1] + \cdots + z^{-1}g[-\ell+1] + g[-\ell]}$$

- $\ell = 1$ case is especially useful:

$$\begin{aligned} g[k+1] &\leftrightarrow z\hat{g}(z) - zg[0] \\ g[k-1] &\leftrightarrow z^{-1}\hat{g}(z) + g[-1] \end{aligned}$$

- General relationship can be shown inductively using the definition.

The Three Most Important Laplace Transforms: #2

Application #1: Given the input-output differential equation description of a discrete-time, linear, time-invariant, lumped system, we can easily compute the transfer function.

A causal example:

$$y[k] + q_1 y[k-1] + \cdots + q_n y[k-n] =$$
$$p_0 u[k] + p_1 u[k-1] + \cdots + p_m u[k-m]$$

Note that the same idea applies to non-causal systems.

A Note About Causality in Discrete-Time Systems

Theorem

A lumped discrete-time system with rational transfer function

$$\hat{g}(z) = \frac{N(z)}{D(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

with $b_m \neq 0$ *is causal if and only if* $m \leq n$.

Note that the degrees here (m and n) are based on the transfer function representation with positive powers of z.

A Note About Causality in Discrete-Time Systems (cont)

$$\hat{g}(z) = \frac{N(z)}{D(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

To see why the theorem must be true, just convert $\hat{g}(z)$ to a difference equation representation:

$$y[k+n] + a_{n-1} y[k+n-1] + \cdots + a_1 y[k+1] + a_0 y[k] =$$
$$b_m u[k+m] + b_{m-1} u[k+m-1] + \cdots + b_1 u[k+1] + b_0 u[k]$$

Let $\kappa = k + n$ and rearrange to get

$$y[\kappa] = -a_{n-1} y[\kappa - 1] - \cdots - a_1 y[\kappa - n + 1] - a_0 y[\kappa - n] +$$
$$b_m u[\kappa - n + m] + b_{m-1} u[\kappa - n + m - 1] + \cdots + b_1 u[\kappa - n + 1] + b_0 u[\kappa - n]$$

When is this difference equation causal?

The Three Most Important z-Transforms: #2

Application #2: Given the state space description of a discrete-time, linear, time-invariant, lumped system, we can easily compute the transfer function.

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] \\ \boldsymbol{y}[k] &= \boldsymbol{C}\boldsymbol{x}[k] + \boldsymbol{D}\boldsymbol{u}[k] \end{aligned}$$

The Three Most Important z-Transforms: #3

Notation (convolution assuming a causal, linear, time-invariant, relaxed system):

$$f[k] * g[k] := \sum_{i=0}^{k} f[k-i]g[i] = \sum_{i=0}^{k} f[i]g[k-i].$$

It isn't too hard to show that

$$\boxed{f[k] * g[k] \leftrightarrow \hat{f}(z)\hat{g}(z)}$$

An Easy Way to Go From an LTI I/O to SS Description

We would like to be able to go from a linear time-invariant I/O difference equation

$$y[k+n] + a_{n-1}y[k+n-1] + \cdots + a_0 y[k] = b_m u[k+m] + b_{m-1}u[k+m-1] + \cdots + b_0 u[k]$$

to a state-space description. Recall that the state-space description is only applicable to causal and lumped systems, hence we can assume here that $m \le n < \infty$.

Not that, since $m \le n$, we can also write (without any loss of generality) our difference equation as

$$y[k+n] + a_{n-1}y[k+n-1] + \cdots + a_0 y[k] = b_n u[k+n] + b_{n-1}u[k+n-1] + \cdots + b_0 u[k]$$

where the first $n-m$ of the b coefficients are equal to zero.

From an LTI SS Description to an I/O Difference Equation

Our strategy here is the same as the continuous time case:

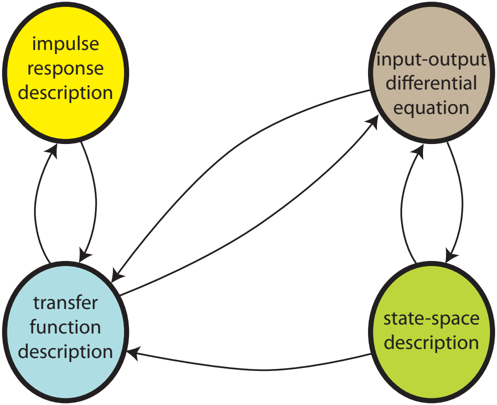
state-space → transfer function → I/O difference equation

$$\hat{\boldsymbol{g}}(z) = \boldsymbol{C}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D} = \frac{\tilde{N}(z)}{D(z)} + \boldsymbol{D} = \frac{N(z)}{D(z)}$$

The only thing that has changed here from the continuous-time case is that the s is now a z. Hence, we know that

- $\deg(D(z)) =$ ____________
- $\deg(\tilde{N}(z)) \leq$ ____________
- $\deg(N(z)) \leq$ ____________
- You should be able to easily convert this transfer function to a causal I/O difference equation (see slides 17-18).

What We Know: Moving Between System Descriptions



You should feel comfortable making all of these conversions when dealing with linear, time-invariant, causal, lumped continuous-time or discrete-time systems.

What About Linear Time-Varying Systems?

- I/O differential/difference equation representation ok?
- Impulse response representation ok?
- Transfer function representation ok?
- State-space representation ok?

The continuous-time state-space representation becomes

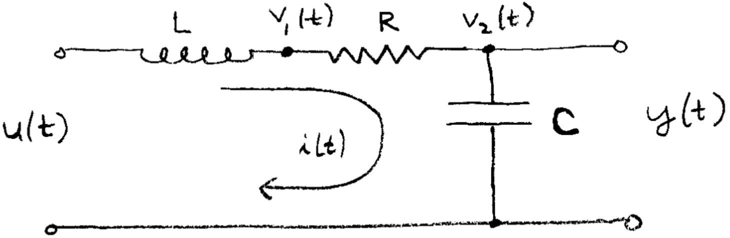
$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t)\\ \boldsymbol{y}(t) &= \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t)\end{aligned}$$

and the discrete-time state-space representation becomes

$$\begin{aligned}\boldsymbol{x}[k+1] &= \boldsymbol{A}[k]\boldsymbol{x}[k] + \boldsymbol{B}[k]\boldsymbol{u}[k]\\ \boldsymbol{y}[k] &= \boldsymbol{C}[k]\boldsymbol{x}[k] + \boldsymbol{D}[k]\boldsymbol{u}[k]\end{aligned}$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ can change over time.

Example: Linear Time-Varying System



Suppose the resistor is time varying now, i.e. $R = R(t)$.

Same procedure as lecture 1 to derive the I/O differential equation description (just use standard KVL and KCL):

$$\frac{d^2y(t)}{dt^2} + \frac{R(t)}{L}\frac{dy(t)}{dt} + \frac{1}{LC}y(t) = \frac{1}{LC}u(t)$$

Example: Linear Time-Varying System

I/O differential equation description:

$$\frac{d^2y(t)}{dt^2} + \frac{R(t)}{L}\frac{dy(t)}{dt} + \frac{1}{LC}y(t) = \frac{1}{LC}u(t)$$

No transfer function representation. How can we write the continuous-time state-space representation?

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t) \end{aligned}$$

What About Nonlinear Systems?

Suppose we have a nonlinear continuous-time state-space description:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\\ \boldsymbol{y}(t) &= \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\end{aligned}$$

where

$$\begin{aligned}\dot{x}_1(t) &= f_1(t, x_1(t), \dots, x_n(t), u_1(t), \dots, u_p(t))\\ \vdots \quad &\vdots \quad \vdots\\ \dot{x}_n(t) &= f_n(t, x_1(t), \dots, x_n(t), u_1(t), \dots, u_p(t))\end{aligned}$$

and

$$\begin{aligned}y_1(t) &= g_1(t, x_1(t), \dots, x_n(t), u_1(t), \dots, u_p(t))\\ \vdots \quad &\vdots \quad \vdots\\ y_q(t) &= g_q(t, x_1(t), \dots, x_n(t), u_1(t), \dots, u_p(t))\end{aligned}$$

Linearization (Step 1a)

As mentioned in Lecture 1, "smoothly nonlinear" systems can often be linearized around a particular operating point, resulting in a standard linear state-space description that can be analyzed with linear algebra.

How can we do this?

Step 1: Find a solution to the state dynamics equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$ that holds for all $t \in \mathbb{R}$.

- Often, this is something like $\boldsymbol{x}(t) = \mathbf{0}$ and $\boldsymbol{u}(t) = \mathbf{0}$ (the system is relaxed with no input).
- There is likely to be more than one possible solution. You should pick the one that represents the nominal operating conditions around which you wish to analyze the behavior of the system.
- Call this nominal solution $\tilde{\boldsymbol{x}}(t)$ and $\tilde{\boldsymbol{u}}(t)$.

Linearization (Step 1b)

So we now have the nominal solution

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}}(t) &= \boldsymbol{f}(t, \tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{u}}(t))\\ \tilde{\boldsymbol{y}}(t) &= \boldsymbol{g}(t, \tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{u}}(t))\end{aligned}$$

Our "smooth" assumption on $\boldsymbol{f}$ implies that small changes in the input and initial state lead to small changes in the solution, i.e.

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}}(t) + \dot{\boldsymbol{x}}_\epsilon(t) &\approx \boldsymbol{f}\left(t, \tilde{\boldsymbol{x}}(t) + \boldsymbol{x}_\epsilon(t), \tilde{\boldsymbol{u}}(t) + \boldsymbol{u}_\epsilon(t)\right)\\ &\approx \boldsymbol{f}\left(t, \tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{u}}(t)\right) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \boldsymbol{x}_\epsilon(t) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \boldsymbol{u}_\epsilon(t)\end{aligned}$$

which implies that

$$\dot{\boldsymbol{x}}_\epsilon(t) \approx \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \boldsymbol{x}_\epsilon(t) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \boldsymbol{u}_\epsilon(t).$$

Linearization (Step 1c)

Similarly, our "smooth" assumption on $\boldsymbol{g}$ implies that

$$\begin{aligned}\tilde{\boldsymbol{y}}(t)+\boldsymbol{y}_\epsilon(t) &\approx \boldsymbol{g}\left(t,\tilde{\boldsymbol{x}}(t)+\boldsymbol{x}_\epsilon(t),\tilde{\boldsymbol{u}}(t)+\boldsymbol{u}_\epsilon(t)\right)\\ &\approx \boldsymbol{g}\left(t,\tilde{\boldsymbol{x}}(t),\tilde{\boldsymbol{u}}(t)\right)+\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\,\boldsymbol{u}=\tilde{\boldsymbol{u}}}\boldsymbol{x}_\epsilon(t)+\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\,\boldsymbol{u}=\tilde{\boldsymbol{u}}}\boldsymbol{u}_\epsilon(t)\end{aligned}$$

which further implies that

$$\boldsymbol{y}_\epsilon(t) \approx \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\,\boldsymbol{u}=\tilde{\boldsymbol{u}}}\boldsymbol{x}_\epsilon(t)+\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\,\boldsymbol{u}=\tilde{\boldsymbol{u}}}\boldsymbol{u}_\epsilon(t).$$

Linearization (Step 2)

Step 2: Compute the four derivatives (called "Jacobians")

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \qquad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_p} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_p} \end{bmatrix} \qquad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_p} \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_p} \end{bmatrix}$$

and evaluate each at the nominal solution $\boldsymbol{x}(t) = \tilde{\boldsymbol{x}}(t)$, $\boldsymbol{u}(t) = \tilde{\boldsymbol{u}}(t)$.

Linearization (Steps 3-4)

Step 3: Set

$$\begin{aligned}
\boldsymbol{A}(t) &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \\
\boldsymbol{B}(t) &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \\
\boldsymbol{C}(t) &= \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}} \\
\boldsymbol{D}(t) &= \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\tilde{\boldsymbol{x}},\, \boldsymbol{u}=\tilde{\boldsymbol{u}}}
\end{aligned}$$

Step 4: Put it all together:

$$\begin{aligned}
\dot{\boldsymbol{x}}_\epsilon(t) &= \boldsymbol{A}(t)\boldsymbol{x}_\epsilon(t) + \boldsymbol{B}(t)\boldsymbol{u}_\epsilon(t) \\
\boldsymbol{y}_\epsilon(t) &= \boldsymbol{C}(t)\boldsymbol{x}_\epsilon(t) + \boldsymbol{D}(t)\boldsymbol{u}_\epsilon(t)
\end{aligned}$$

This is a linear (possibly time-varying), causal, lumped system. Solutions are only accurate in the "neighborhood" of $\tilde{\boldsymbol{x}}(t)$, $\tilde{\boldsymbol{u}}(t)$, and $\tilde{\boldsymbol{y}}(t)$.

Linearization Example

A Remark on Linearization

The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ *must be smooth* in the neighborhood of the solution $\tilde{\boldsymbol{x}}(t)$, $\tilde{\boldsymbol{u}}(t)$, and $\tilde{\boldsymbol{y}}(t)$, otherwise linearization is not going to be useful (the Taylor series approximations in the derivation won't hold).

Here is an example of a state-dynamic function that definitely isn't smooth at $x(t) = 0$:

$$\dot{x}(t) = \mathrm{sign}(x(t)) + u(t)$$

where

$$\mathrm{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

It should be clear that $\tilde{x}(t) = 0$ and $\tilde{u}(t) = 0$ is a solution. But the necessary derivatives don't exist here. Can't linearize.

Conclusions

- We're done discussing the different mathematical descriptions of systems. You should now understand:
  - Continuous-time and discrete-time I/O differential/difference equations.
  - Continuous-time and discrete-time transfer functions.
  - Continuous-time and discrete-time impulse responses.
  - Continuous-time and discrete-time state-space representations.
  - Capabilities and limitations of different mathematical descriptions.
  - How to move between different mathematical descriptions when a system is linear, time-invariant, causal, and lumped.
  - Time-varying state-space system description.
  - Linearization of nonlinear state-space system descriptions.
- Along the way, you had to learn a little bit of linear algebra:
  - Identity matrix
  - Matrix inverse
  - Determinant
  - Adjoint
- Next week: Begin analysis of linear state-space system descriptions.