W-3/3

EECS 16B Section 7B

Main Topic: System ID + Stability

Administrivia:

- HW 7 due Fri, 3/5
- Anonymous Feedback:
  bit.ly/maxwell-16B-feedback-sp21
- Midterm coming up (oof™)
  - Staff, CSM/HKN Review Sessions next week
  - Scope up to 3/4 Lec
  - Logistics out Friday or Monday

Agenda:

- Least Squares Review
  - Q1: System ID via Least Squares
- Stability
  - Q2: D.T. Stability Plots
  - Q3: Circuit Stability

Least Squares Review

"Overdetermined System"

- "Tall" matrix $\in \mathbb{R}^{m \times n}$
- Not invertible
- No exact solution ("unique") – best we can do is approx. + min. error:

$$\vec{y} = \underbrace{\begin{bmatrix} & & \\ & m > n & \\ & & \end{bmatrix}}_{D:\ m \times n} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}}_{\vec{p}:\ n}$$

We try approximating $\vec{p}$ with $\hat{\vec{p}}$ s.t. $D\hat{\vec{p}} \cong \vec{y}$

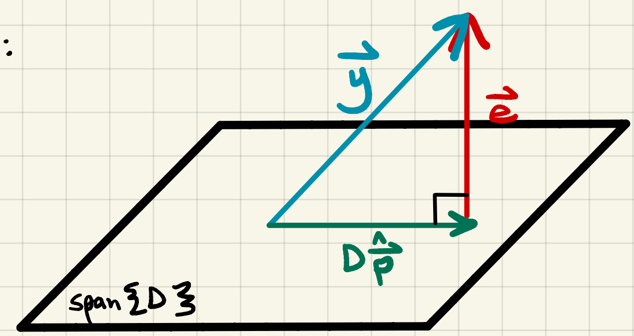
$$\underbrace{\vec{y}_{m\times 1}}_{\text{Measurements (Data)}} = \underbrace{D_{m\times n}}_{\text{known tall matrix}} \underbrace{\hat{\vec{p}}_{n\times 1}}_{\text{unknown parameters}} + \underbrace{\vec{e}_{m\times 1}}_{\text{uncertainty (error)}}$$

$\Rightarrow$ pick $\hat{\vec{p}}$ to minimize $\|\vec{y} - D\hat{\vec{p}}\|_2^2$, i.e. the error

$$\vec{e} = \vec{y} - D\hat{\vec{p}}$$

Consider this from a geometric perspective:

- We do a vector projection of $\vec{y}$ onto span$\{D\}$ $\longrightarrow D\hat{\vec{p}}$
  - y = actual output vector
  - D = input matrix (D = "Data")
  - $\hat{p}$ = potential parameters



$\vec{e} \perp$ span$\{D\}$

$$\vec{y} = D\hat{\vec{p}} + \vec{e}$$

$$D^T(\vec{y} - D\hat{\vec{p}} = \vec{e}) \longrightarrow D^T\vec{y} - D^TD\hat{\vec{p}} = \underbrace{D^T\vec{e}}_{=0}$$

$$(D^TD)^{-1}(D^T\vec{y} = D^TD\hat{\vec{p}})$$

$$\boxed{\hat{\vec{p}} = (D^TD)^{-1}D^T\vec{y}}$$

$\longrightarrow$ (D^TD) is invertible

D must have lin. indep. col.

System ID Motivation

General Form: $\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i] + \vec{w}[i]$

Goal: Learn parameters A and B (or scalars a, b)

- Use input/output data to learn relationship

1. System identification by means of least squares

Working through this question will help you understand better how we can use experimental data taken from a (presumably) linear system to learn a discrete-time linear model for it using the least-squares techniques you learned in 16A. You will later do this in lab for your robot car.

As you were told in 16A, least-squares and its variants are not just the basic workhorses of machine learning in practice, they are play a conceptually central place in our understanding of machine learning well beyond least-squares.

Throughout this question, you should consider measurements to have been taken from one long trace through time.

(a) Consider the scalar discrete-time system

$$x[i+1] = ax[i] + bu[i] + w[i] \tag{1}$$

Where the scalar state at time i is $x[i]$, the input applied at time i is $u[i]$ and $w[i]$ represents some external disturbance that also participated at time i.

Assume that you have measurements for the states $x[i]$ from $i = 0$ to m and also measurements for the controls $u[i]$ from $i = 0$ to $m-1$.

Set up a least-squares problem that you can solve to get an estimate of the unknown system parameters a and b.

$x[0] \ldots x[m-1],\ x[m]$

$u[0] \ldots u[m-1]$

$x[i+1] = ax[i] + bu[i] + w[i]$

$\vec{y} = D\vec{p}$ → param → a, b

output: $x[i+1]$

data: $x[i]$, $u[i]$

$$\underbrace{\begin{bmatrix} x[1] \\ \vdots \\ x[m] \end{bmatrix}}_{\vec{y}} = \underbrace{\begin{bmatrix} x[0] & u[0] \\ \vdots & \vdots \\ x[m-1] & u[m-1] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\hat{\vec{p}}} \quad \cancel{x}$$

(b) What if there were now two distinct scalar inputs to a scalar system

$$x[i+1] = ax[i] + b_1u_1[i] + b_2u_2[i] + w[i] \tag{2}$$

and that we have measurements as before, but now also for both of the control inputs.
Set up a least-squares problem that you can solve to get an estimate of the unknown system parameters a, b_1, b_2**.**

(c) What could go wrong in the previous case? For what kind of inputs would make least-squares fail to give you the parameters you want?

(d) Returning to the scalar case with only one input, what could go wrong? When would you be unable to use least-squares to get the parameters you want?

$u_2[i]$

b_2

$$\underbrace{\begin{bmatrix} x[1] \\ \vdots \\ x[m] \end{bmatrix}}_{\vec{y}} = \underbrace{\begin{bmatrix} x[0] & u_1[0] & u_2[0] \\ \vdots & \vdots & \vdots \\ x[m-1] & u_1[m-1] & u_2[m-1] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a \\ b_1 \\ b_2 \end{bmatrix}}_{\hat{\vec{p}}}$$

(c) Bad if: D has linearly dependent columns

D is not full rank

$\vec{u_1} = \alpha \vec{u_2}$

(d) Ditto. $\vec{x} = \alpha \vec{u}$

(e) Now consider the two dimensional state case with a single input.

$$\vec{x}[i+1] = \begin{bmatrix} x_1[i+1] \\ x_2[i+1] \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \vec{x}[i] + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u[i] + \vec{w}[i] \quad (3)$$

$\begin{bmatrix} x_1[i] \\ x_2[i] \end{bmatrix}$

How can we treat this like two parallel problems to set this up using least-squares to get estimates for the unknown parameters $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2$? What work/computation can we reuse across the two problems?

$$\vec{x}_1[i+1] = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix} \vec{x}[i] + b_1 u[i] + \vec{w}[i]$$

$$\vec{x}_2[i+1] = \begin{bmatrix} a_{21} & a_{22} \end{bmatrix} \vec{x}[i] + b_2 u[i] + \vec{w}[i]$$

$$\begin{bmatrix} x_1[1] \\ \vdots \\ x_1[m] \end{bmatrix} = \begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ \vdots & \vdots & \vdots \\ x_1[m-1] & x_2[m-1] & u[m-1] \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \end{bmatrix}$$

$$\begin{bmatrix} x_2[1] \\ \vdots \\ x_2[m] \end{bmatrix} = \begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ \vdots & \vdots & \vdots \\ x_1[m-1] & x_2[m-1] & u[m-1] \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \\ b_2 \end{bmatrix}$$

$$m \underbrace{\begin{bmatrix} x_1[1] & x_2[1] \\ \vdots & \vdots \\ x_1[m] & x_2[m] \end{bmatrix}}_{2} = m \underbrace{\begin{bmatrix} x_1[0] & x_2[0] & u[0] \\ \vdots & \vdots & \vdots \\ x_1[m-1] & x_2[m-1] & u[m-1] \end{bmatrix}}_{3} \; 3 \underbrace{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \end{bmatrix}}_{2}$$

1 Stability

output

A system is *stable* if $\vec{x}(t)$ is bounded for any initial condition $\vec{x}(0)$ and any bounded input $u(t)$. A system is *unstable* if there is an $\vec{x}(0)$ and some bounded input $u(t)$ for which $|\vec{x}(t)| \to \infty$ as $t \to \infty$.

Discrete time systems

A discrete time system is of the form:

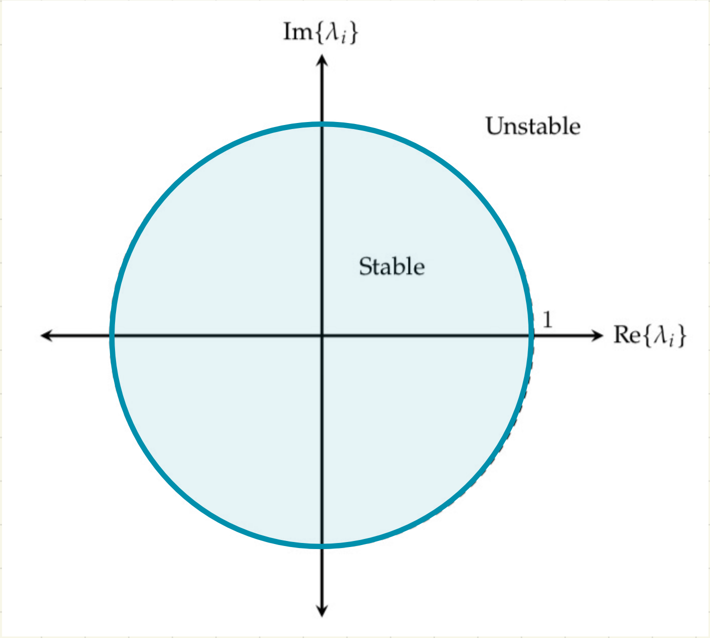
$$\vec{x}(t+1) = A\vec{x}(t) + B\vec{u}(t)$$

$$\vec{x}[t+1] = A\vec{x}[t] + B\vec{u}[t]$$

Let λ be any particular eigenvalue of A.

This system is stable if for all λ, $|\lambda| < 1$.

This system is unstable if there exists an eigenvalue λ, $|\lambda| \geq 1$.



$$x[t+1] = \lambda x[t] + bu[t]$$

$$x[k] = \lambda^k x[0] + \_\ \_\ .\ .\ .\ \_\ .$$

Continuous time systems

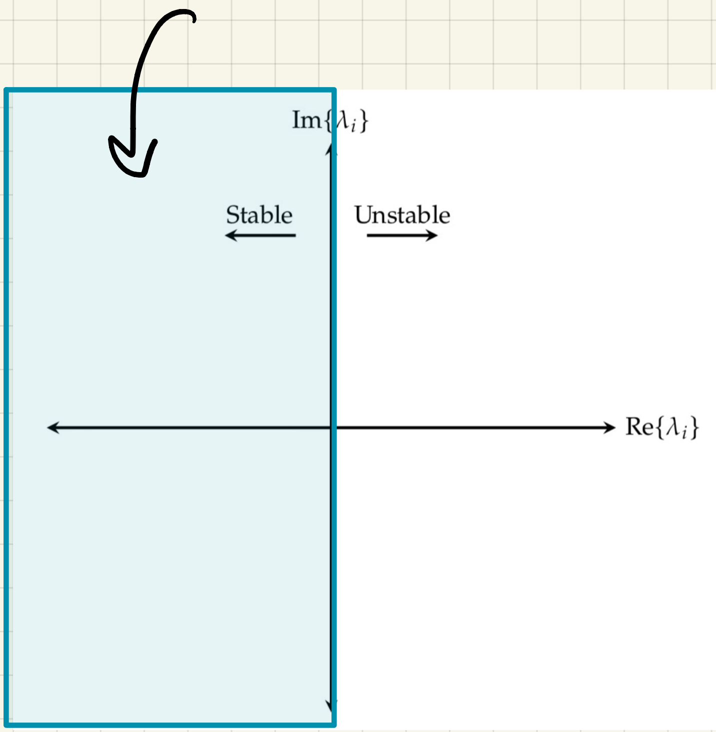
A continuous time system is of the form:

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

Let λ be any particular eigenvalue of A.

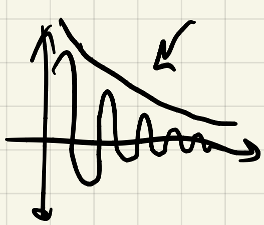
This system is stable if for all λ, $\mathrm{Re}\{\lambda\} < 0$.

This system is unstable if there exists an eigenvalue λ, $\mathrm{Re}\{\lambda\} \geq 0$.



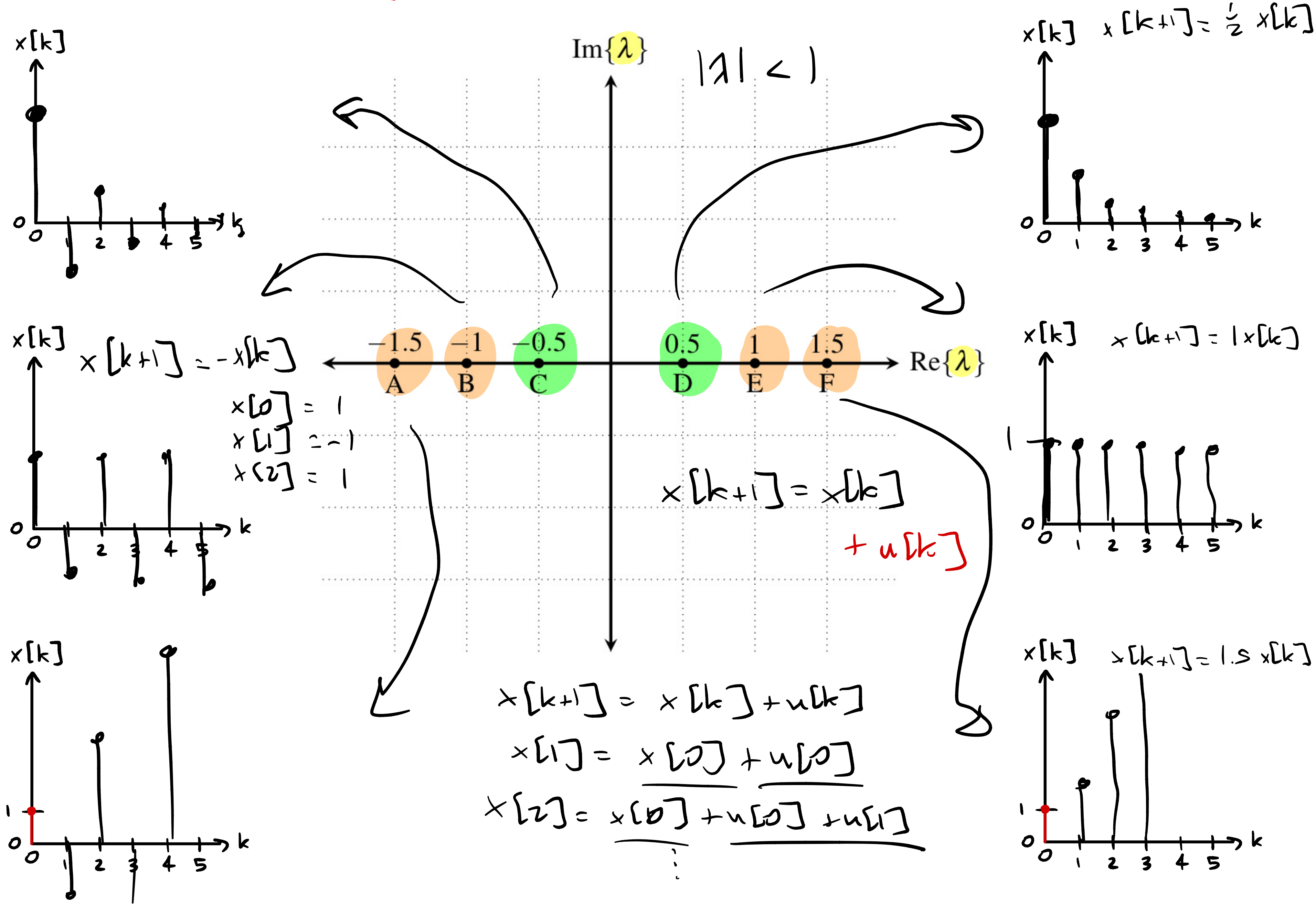
Solution is of the form $Ae^{\lambda t}$

$$\lambda = \underbrace{\mathrm{Re}\{\lambda\}}_{\text{"envelope"}} + j\,\underbrace{\mathrm{Im}\{\lambda\}}_{\text{"oscillation"}}$$



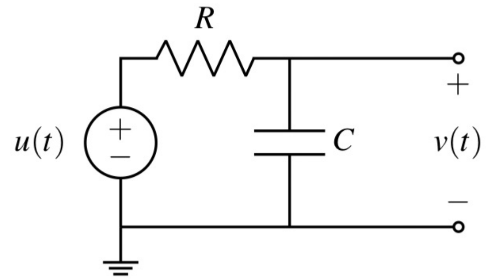
2. Discrete time system responses

We have a system $x[k+1] = \lambda x[k]$. For each λ value plotted on the real-imaginary axis, sketch $x[k]$ with an initial condition of $x[0] = 1$. Determine if each system is stable.



3. Stability Examples and Counterexamples

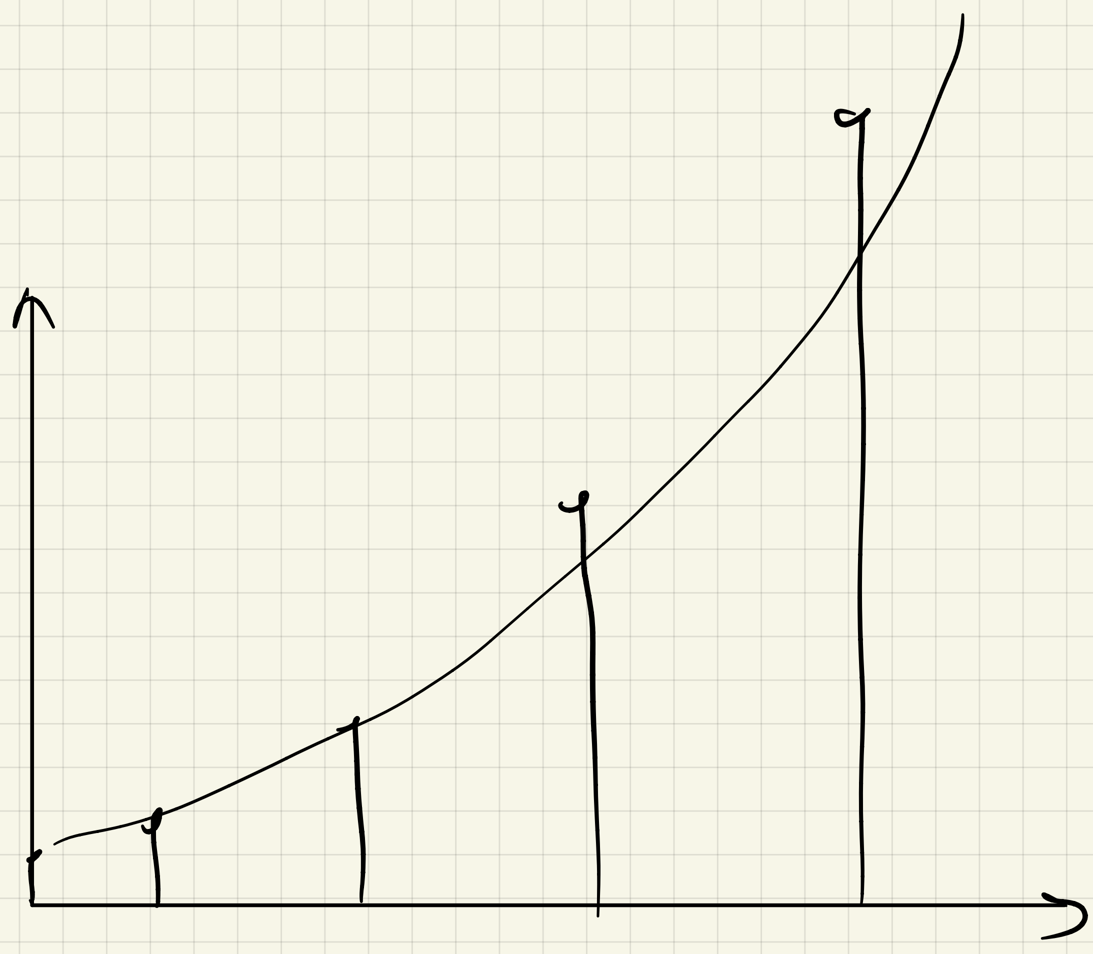
(a) Consider the circuit below with $R = 1\Omega, C = 0.5\text{F}$, and $u(t) = \cos(t)$. Furthermore assume that $v(0) = 0$ (that the capacitor is initially discharged).



This circuit can be modeled by the differential equation

$$\frac{d}{dt}v(t) = -2v(t) + 2u(t) \tag{4}$$

Show that the differential equation is always stable. Consider what this means in the physical circuit.



(b) Consider the discrete system

$$x[k+1] = 2x[k] + u[k] \tag{5}$$

with $x[0] = 0$.

Is the system stable or unstable? If unstable, find a bounded input sequence $u[k]$ that causes the system to "blow up". If unstable, is there still a (non-trivial) bounded input sequence that does not cause the system to "blow up"?

$$x[k+1] = \frac{1}{2} x[k] + u[k]$$

$$x[1] = \frac{1}{2} x[0] + u[0]$$

$$x[2] = \frac{1}{2} (x[1]) + u[1]$$

$$= \frac{1}{4} x[0] + \frac{1}{2} u[0] + u[1]$$

bit.ly/maxwell-16B-feedback-sp21