EECS 16B Designing Information Devices and Systems II
Spring 2021 UC Berkeley

# Homework 8

**This homework is due on Friday, March 12, 2021, at 11:00PM. Self-grades and HW Resubmission are due on Tuesday, March 16, 2021, at 11:00PM. Note that to save time for midterm prep, the solutions will be released early on Monday, but we still will require a HW submission on Gradescope.**

**1. Reading Lecture Notes**

Staying up to date with lectures is an important part of the learning process in this course. Here are links to the notes that you need to read for this week: Note 7B, Note 8

(a) For the overdetermined system $A\vec{x} = \vec{b}$, what condition is needed to use least squares to estimate $\hat{x}$?

**Solution:** $A$ must be full rank, and since $A$ is a tall matrix, this means it has linearly independent columns.

(b) What are the eigenvalue tests for stability for both discrete-time systems and continuous-time systems?

**Solution:** In discrete-time, the system is stable when all of $A$'s eigenvalues have magnitude less than 1. In continuous-time, the system is stable when all of $A$'s eigenvalues have real part less than 0.

(c) How do you use feedback control to change the eigenvalues of a closed-loop continuous-time system?

**Solution:** We let $u(t) = Kx(t)$ so that $\frac{d}{dt}\vec{x} = A\vec{x} + B\vec{u} = A\vec{x} + BK\vec{x} = (A+BK)\vec{x}$. Then, we calculate the determinant of $A + BK - \lambda I$ and change the values of the $K$ matrix to get our desired eigenvalues.

The same thing can be done with discrete-time systems as well. Now, $u[t] = Kx[t]$ so $x[t+1] = Ax[t] + Bu[t] = Ax[t] + BKx[t] = (A+BK)x[t]$. In both cases, the state transition matrix becomes $A+BK$ and its eigenvalues can be changed by setting $K$.

## 2. System Identification

You are given a discrete-time system as a black-box. You don't know the specifics of the system but you know that it takes one scalar input and has two states that you can observe. You assume that the system is linear and of the form

$$\vec{x}[t+1] = A\vec{x}[t] + Bu[t] + \vec{w}[t], \tag{1}$$

where $\vec{w}[t]$ is an external unseen disturbance that you hope is small, $u[t]$ is a scalar input, and

$$A = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix}, \quad B = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad x[t] = \begin{bmatrix} x_0[t] \\ x_1[t] \end{bmatrix}. \tag{2}$$

You want to identify the system parameters from measured data. You need to find the unknowns: $a_0$, $a_1$, $a_2$, $a_3$, $b_0$ and $b_1$. However, you can only interact with the system via a blackbox model, i.e. you can see the states $\vec{x}[t]$ and set the inputs $u[t]$ that allow the system to move to the next state.

(a) You observe that the system has state $\vec{x}[t] = [x_0[t], x_1[t]]^T$ at time $t$. You pass input $u[t]$ into the blackbox and observe the next state of the system: $\vec{x}[t+1] = [x_0[t+1], x_1[t+1]]^T$.
**Write scalar equations for the new states, $x_0[t+1]$ and $x_1[t+1]$.** Write these equations in terms of the $a_i$, $b_i$, the states $x_0[t]$, $x_1[t]$ and the input $u[t]$. Here, assume that $\vec{w}[t] = \vec{0}$ (i.e. the model is perfect).

**Solution:**

$$x_0[t+1] = a_0 x_0[t] + a_1 x_1[t] + b_0 u[t] \tag{3}$$
$$x_1[t+1] = a_2 x_0[t] + a_3 x_1[t] + b_1 u[t] \tag{4}$$

(b) Now we want to identify the system parameters. We observe the system at the start state $\vec{x}[0] = \begin{bmatrix} x_0[0] \\ x_1[0] \end{bmatrix}$.
We can then input $u[0]$ and observe the next state $\vec{x}[1] = \begin{bmatrix} x_0[1] \\ x_1[1] \end{bmatrix}$. We can continue this for an $m$ long sequence of inputs.
Let us define an $m$ long trajectory to be $[x_0[0], x_1[0], u[0], x_0[1], x_1[1], u[1], x_0[2], x_1[2], u[2], \ldots, x_0[m-1], x_1[m-1], u[m-1], x_0[m], x_1[m]]$. **Assuming there is no noise ($\vec{w}[t] = \vec{0}$), what is the minimum value of $m$ you need to identify the system parameters?**

**Solution:** There are 6 unknowns so you need 6 equations to properly identify the system. To form the 6 equations we need to give the blackbox $m = 3$ inputs. Namely $u[0], u[1], u[2]$ so we can see the state at times $0, 1, 2, 3$ to give us our six equations.
Notice that the initial condition on its own gives us no equations because the unknowns we are interested in do not impact the initial condition. They govern the evolution of the system, and hence the states at times $1, 2, 3$ each give us two equations.

(c) Now assume that there is a nonzero noise/disturbance $\vec{w}[t]$. **Would using more than equations than in part (b) help you in this case? If so, explain why.**

**Solution:** Since now there is noise, there is no guarantee that the 3 timesteps that we collected will actually give a consistent solution to the 6 parameters in $A$ and $B$. To deal with the noise, we want to use least squares to estimate our parameters, rather than just a direct inverse. From 16A, we can remember that the more rows in your data matrix, the more accurate your least squares estimate will be. Thus, we actually want to collect as many timesteps of data as possible so that our least squares estimate will be as accurate to the true model as possible.

(d) Say we feed in a total of 4 inputs $[u[0], u[1], u[2], u[3]]$ into our blackbox. This allows us to observe the following states $[x_0[0], x_0[1], x_0[2], x_0[3], x_0[4]]$ and $[x_1[0], x_1[1], x_1[2], x_1[3], x_1[4]]$, which we can use to identify the system.

To identify the system we need to set up an approximate (because of potential disturbances) matrix equation

$$D\vec{p} \approx \vec{y}$$

using the observed values above and the unknown parameters we want to find. We know our parameter vector should be $\vec{p} = \begin{bmatrix} a_0 & a_1 & b_0 & a_2 & a_3 & b_1 \end{bmatrix}^T$. **Find the corresponding $D$ and $\vec{y}$ to do system indentification. Write out both explicitly.**

**Solution:**

Due to the structure of the parameter vector, the equations for $x_0$ should be on the left part of $D$ and the equations for $x_1$ on the right part. Then,

$$D\vec{p} = \vec{y} \tag{5}$$

$$\begin{bmatrix} x_0[0] & x_1[0] & u[0] & 0 & 0 & 0 \\ x_0[1] & x_1[1] & u[1] & 0 & 0 & 0 \\ x_0[2] & x_1[2] & u[2] & 0 & 0 & 0 \\ x_0[3] & x_1[3] & u[3] & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0[0] & x_1[0] & u[0] \\ 0 & 0 & 0 & x_0[1] & x_1[1] & u[1] \\ 0 & 0 & 0 & x_0[2] & x_1[2] & u[2] \\ 0 & 0 & 0 & x_0[3] & x_1[3] & u[3] \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ a_2 \\ a_3 \\ b_1 \end{bmatrix} = \begin{bmatrix} x_0[1] \\ x_0[2] \\ x_0[3] \\ x_0[4] \\ x_1[1] \\ x_1[2] \\ x_1[3] \\ x_1[4] \end{bmatrix} \tag{6}$$

Note that you could have swapped any rows of $D$ and their corresponding rows in $y$ and the answer would still be correct.

**Sidenote:** We gave you this specific $\vec{p}$ in this problem. However, in general you could've also formulated the system ID problem similar to in Note 7B where instead of 1 long vector, we would use a parameter matrix so

$$P = \begin{bmatrix} A^T \\ B^T \end{bmatrix} = \begin{bmatrix} a_0 & a_2 \\ a_1 & a_3 \\ b_0 & b_1 \end{bmatrix} \tag{7}$$

Then the new, smaller least squares problem becomes

$$\begin{bmatrix} x_0[0] & x_1[0] & u[0] \\ x_0[1] & x_1[1] & u[1] \\ x_0[2] & x_1[2] & u[2] \\ x_0[3] & x_1[3] & u[3] \end{bmatrix} \begin{bmatrix} a_0 & a_2 \\ a_1 & a_3 \\ b_0 & b_1 \end{bmatrix} = \begin{bmatrix} x_0[1] & x_1[1] \\ x_0[2] & x_1[2] \\ x_0[3] & x_1[3] \\ x_0[4] & x_1[4] \end{bmatrix} \tag{8}$$

(e) Now that we have set up $D\vec{p} \approx \vec{y}$, **explain how you would use this approximate equation to estimate the unknown values $a_0, a_1, a_2, a_3, b_0, b_1$.** In particular, give an expression for your estimate $\widehat{\vec{p}}$ in terms of the $D$ and $\vec{y}$. Assume that the columns of $D$ are linearly independent.

*(HINT: Don't forget that D is not a square matrix. It is taller than it is wide.)*

**Solution:** Using the equation above we realize that we just need to solve for $\vec{p}$ to learn the system. Since the matrix $D$ is not invertible we can use the standard least squares formula from 16A

$$\widehat{\vec{p}} = (D^T D)^{-1} D^T \vec{y} \tag{9}$$

to find the unknown values.

This is ok because we had told you to feel free to assume that the columns of $D$ are linearly independent. If the columns of $D$ are linearly-independent, then $D$ has no nontrivial nullspace by definition. And we know from lecture/16A that $D^T D$ has the same nullspace as $D$ and so it is safe to invert $D^T D$. (You didn't have to make this argument, but it is good to be able to make such arguments.)

**3. Identifying systems from their responses to known inputs**

In many problems, we have an unknown system, and would like to characterize it. One of the ways of doing so is to observe the system response with different initial conditions (or inputs). This problem is also called system identification. It is a prototypical example of a problem that today is called machine learning — inferring an underlying pattern from data, and doing so well enough to be able to exploit that pattern in some practical setting. Go through the attached Jupyter notebook "DemoSystemID_16b.ipynb" and answer the following questions.

(a) In Example 2, we assume that instead of measuring the state $\vec{x}$, we are instead measuring a transformation of the state $\vec{y} = T\vec{x}$ where $T$ is a full rank matrix. Assume that we perform system ID on our observations $\vec{y}[t]$ to recover $A_y$, $B_y$ such that $\vec{y}[t+1] = A_y\vec{y}[t] + B_y u[t]$. **How do the identified $A_y$ and $B_y$ matrices relate to the original $A$ and $B$ matrices in the dynamics of $\vec{x}$?** Remember that our original state dynamics are $\vec{x}[t+1] = A\vec{x} + B\vec{u}$.

*HINT: The answer is given in the Jupyter notebook but remember to show your work.*

**Solution:** Using our given transformation that $\vec{y} = T\vec{x}$,

$$\vec{y}[t+1] = A_y\vec{y}[t] + B_y u[t] \tag{10}$$

$$T\vec{x}[t+1] = A_y T\vec{x}[t] + B_y u[t] \tag{11}$$

$$\vec{x}[t+1] = T^{-1}A_y T\vec{x}[t] + T^{-1}B_y u[t] \tag{12}$$

Thus, $A = T^{-1}A_y T$ and $B = T^{-1}B_y$. This can be rewritten as $A_y = TAT^{-1}$ and $B_y = TB$, which is exactly what is in the Jupyter notebook.

(b) **Please share your observations on Example 2.**

**Solution:** It's nice to see that a linear transformation of the state trace does not have a tremendous effect on our ability to perform system identification. Basically, this means that we have some leeway in choosing what data to observe in practice. In a circuit, for example, we would typically choose capacitor voltages and inductor currents as our state variables. However, it's difficult to make current measurements in real time in a non-invasive way, so we would prefer for our observations to consist of only voltages. As long as we can in principle recover the inductor currents as some linear combination of voltages (this is usually possible), then we can just measure those voltages and proceed as normal.

Furthermore, even if an unwanted state tranformation occurs, we know that our estimate of the system does not change drastically. As we saw, the estimated eigenvalues are still correct. Furthermore, controllability and observability are preserved in coordinate changes, so the system we identified will have almost all of the same control-theoretic properties as the true system. Believe me, that's a relief!

(c) **Prove that for any full rank transformation matrix $T$, the eigenvalues of $A_y$ and $A$ from part (a) are the same.**

**Solution:** Assume that the eigenvalue eigenvector pairs of $A$ are $(\lambda_1, \vec{v}_1), (\lambda_2, \vec{v}_2), \dots, (\lambda_n, \vec{v}_n)$. We claim that $T\vec{v}_i$ will be the eigenvectors of $A_y$. We can see this with

$$A\vec{v}_i = \lambda_i\vec{v}_i \tag{13}$$

$$A_y T\vec{v}_i = TAT^{-1}T\vec{v}_i = TA\vec{v}_i = T(\lambda_i\vec{v}_i) \tag{14}$$

$$= \lambda_i T\vec{v}_i \tag{15}$$

Thus, $A_y$ also has eigenvalues $\lambda_i$ with its corresponding eigenvector being $T\vec{v}_i$.

(d) **Please share your observations on Example 3.**

**Solution:** From playing around with the system and state trace parameters, you may have noticed a few trends:

- Increasing the noise magnitude $\sigma$ reduced the accuracy of the identified eigenvalues (that is, the identified eigenvalues were farther away from the true ones).
- Increasing the number of samples improved the accuracy of the identification.
- Increasing the number of states has a large impact on how accurate the identification is, for a fixed noise magnitude and number of data points.

The final point is important in practice. It suggests that, if we have some control over how many states we model a system with, and if all other things are equal, then we should choose to have fewer states rather than more, so that our system identification requires less data to be accurate.

This is a point that turns out to be extremely important in machine-learning more generally — we do not necessarily always want the most complicated model.

(e) **Please share your observations on Example 4.**

**Solution:**

This example took you beyond what you have learned in lecture. It involved figuring things out without being able to observe the state and instead just seeing scalar observations of the state.

You probably noticed that identifying a system with scalar observations requires a longer sample trace to be accurate than identifying a system with state observations does. The reason why is pretty clear: you only have one scalar point of data at every time step now, instead of a vector. Nonetheless, it's a really great thing that we can still do system identification with scalar observations at all! In many applications, a scalar output is all that's available.

Having to select a value for $n$ with no prior knowledge introduces more trial-and-error than we would like, but in the method used in this section it's a necessary evil. But is it necessary in general? I shouldn't overly anthropomorphize the data, but *it* must know how many states there are, since it was generated by the system. So it seems like some knowledge of the "true $n$" is hiding somewhere in the data, if only we know how to look...

We will return to this issue later in the course.

(f) **Please share your observations on Example 5.**

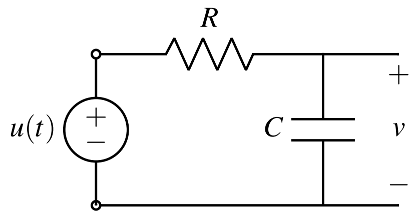
**Solution:** This is another example that takes you a bit beyond what you have seen in lecture, but in a natural way. What happens if your model size is wrong? Do you have to get the model size right?

It's interesting to see that the effect of many eigenvalues near the origin can be effectively approximated by just a couple of eigenvalues in a smaller system. Just like example 3, this suggests that there is such a thing as "too many states". Basically, 13 out of the 16 states of this system contribute almost nothing to the behavior of the system– we were able to throw them away and still capture the important behavior.

So what happens because we are ignoring these states and the true complexity of the underlying model? We will hopefully see in a later homework that what this does is contribute to the disturbance that our estimated model experiences. It not only has an disturbance because it didn't estimate the parameters of the model perfectly (say because of observation noises), but also because it chose a model that was simpler than reality. But as long as we can be robust to this disturbance, we are still fine.

## 4. BIBO Stability

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




This circuit can be modeled by the differential equation

$$\frac{d}{dt}v(t) = -2v(t) + 2u(t) \tag{16}$$

**Show that $v(t)$ remains bounded for all time if the input $u(t)$ is bounded, i.e. $|u(t)| < k, \forall t \geq 0$.** *HINT: You may want to write the expression for $x(t)$ in terms of $u(t)$ and $x(0)$ and then use the triangle inequality to prove that $x(t)$ is bounded. Also use the fact that the norm of an integral is $\leq$ the integral of a norm. In other words $|\int_a^b f(\tau)d\tau| \leq \int_a^b |f(\tau)|d\tau$. This inequality is just the regular triangle inequality you have seen before, generalized to integrals. The proof strategy used in lecture for the discrete time case will extend here as well with the modified triangle inequality usage.*

Thinking about this helps you understand what bounded-input-bounded-output stability means in a physical circuit.

**Solution:**

For the physical system, we can intuitively see that the voltage on the capacitor can never exceed the voltage from the voltage source.

We can also try to understand this as a differential equation and see why it must be bounded directly. Let us assume that $|u(t)| \leq k, \forall t$. We know that the solution to the scalar differential equation is given by

$$x(t) = e^{-2t}x(0) + \int_0^t e^{-2(t-\tau)}2u(\tau)d\tau$$

Then we can try to bound $x(t)$ for $t \geq 0$. We first use the triangle inequality ($|a+b| \leq |a| + |b|$) to get

$$|x(t)| = |e^{-2t}x(0) + \int_0^t e^{-2(t-\tau)}2u(\tau)d\tau| \tag{17}$$

$$|x(t)| \leq |e^{-2t}x(0)| + |\int_0^t e^{-2(t-\tau)}2u(\tau)d\tau| \tag{18}$$

We then use the property that the integral of absolute value will always be greater than the absolute value of the integral (equation 18 to 19), and that an exponential is always positive (equation 19 to 20):

$$|x(t)| \leq |e^{-2t}x(0)| + \int_0^t |e^{-2(t-\tau)}2u(\tau)|d\tau \tag{19}$$

$$= e^{-2t}|x(0)| + \int_0^t e^{-2(t-\tau)}2|u(\tau)|d\tau \tag{20}$$

Finally, plugging in our bound for $|u(t)|$ and solving,

$$\leq e^{-2t}|x(0)| + \int_0^t e^{-2(t-\tau)} 2k d\tau \tag{21}$$
$$= e^{-2t}|x(0)| + 2ke^{-2t} \int_0^t e^{2\tau} d\tau \tag{22}$$
$$= e^{-2t}|x(0)| + 2ke^{-2t} \frac{1}{2}(e^{2t} - 1) \tag{23}$$
$$= e^{-2t}|x(0)| + k(1 - e^{-2t}) \tag{24}$$
$$\leq |x(0)| + k \tag{25}$$

The negative exponential is what makes this system stay bounded.

(b) Consider a continuous-time scalar real differential equation with known solution

$$\frac{d}{dt}x(t) = ax(t) + bu(t) \qquad x(t) = e^{at}x(0) + \int_0^t e^{a(t-\tau)} bu(\tau) d\tau.$$

**Show that if the system has $\mathrm{Re}\{a\} > 0$, then there exists a bounded input ($|u(t)| \leq \varepsilon$) that can result in an unbounded output, even with a zero initial condition. In other words, show the system is BIBO unstable.**

**Solution:** To start, let's consider the case when $x(0) = 0$. Now, we are left with the integral term to show that a bounded input can result in an unbounded output. Let's consider the case when $u(t) = k\ \forall t$, giving us

$$x(t) = \int_0^t e^{a(t-\tau)} bk d\tau.$$

Rearranging the integral, we can focus on the exponential term:

$$x(t) = bk \int_0^t e^{a(t-\tau)} d\tau.$$

If $a = 0$, this is just $x(t) = bkt$ which is clearly growing without bound. For other $a \neq 0$, with a change of variables, we can evaluate this integral

$$\int_0^t e^{a(t-\tau)} d\tau = -\frac{1 - e^{at}}{a}.$$

So, when $t \to \infty$, this will be unbounded since $e^{at}$ will grow exponentially for $a > 0$.
Another way of thinking about this is that we know that with $a \geq 0$, as $t$ goes to $\infty$, there is no upper bound on the size of $x(t)$. Specifically,

$$x(t) = bk \int_0^t e^{a(t-\tau)} = -bk \frac{1 - e^{at}}{a}.$$

So, the state is unbounded as the magnitude grows endlessly with $t$.

(c) **Repeat the previous part for the specific case of complex $a = r + j2\pi$ where $r > 0$ and zero initial condition $x(0) = 0$.**

All the other truly complex unstable cases are the same way for the same essential reason.

**Solution:** Following on from the integral form above,

$$x(t) = \int_0^t e^{(r+j2\pi)(t-\tau)} bk d\tau.$$

Using the solution to the integral above, we are left with

$$\int_0^t e^{a(t-\tau)} d\tau = -\frac{1-e^{at}}{a} = -\frac{1-e^{(r+j2\pi)t}}{r+j2\pi}.$$

It is not immediately clear that this will diverge, but let's consider the magnitude of this number.

$$|x(t)| = |\frac{-1+e^{(r+j2\pi)t}}{r+j2\pi}| = \frac{|-1+e^{(r+j2\pi)t}|}{|r+j2\pi|}$$

The magnitude of the denominator is simply $\sqrt{r^2+4\pi^2}$ which is a constant value. To find the magnitude of the numerator, we split it into its real and imaginary components using Euler's Formula.

$$\begin{aligned}
|-1+e^{(r+j2\pi)t}| &= |-1+e^{rt}e^{2\pi jt}| \\
&= |-1+e^{rt}\cos(2\pi t)+je^{rt}\sin(2\pi t)| \\
&= \sqrt{(-1+e^{rt}\cos(2\pi t))^2+e^{2rt}\sin^2(2\pi t)} \\
&= \sqrt{1-2e^{rt}\cos(2\pi t)+e^{2rt}\cos^2(2\pi t)+e^{2rt}\sin^2(2\pi t)} \\
&= \sqrt{1-2e^{rt}\cos(2\pi t)+e^{2rt}}
\end{aligned}$$

Note that in the last equation we use that $\sin^2(\omega t)+\cos^2(\omega t) = 1$ for any $\omega$. If $t \to \infty$, the magnitude of the numerator will blow up from the exponential term since $r > 0$. Then, since the magnitude of the denominator is a constant, the entire magnitude of $x(t)$ will blow up to $\infty$. Thus, the system is unstable.

(d) **[Optional] Repeat the previous part for the specific case of purely imaginary $a = j2\pi$ with zero initial condition $x(0) = 0$.**

All the other purely imaginary unstable cases are the same way for the same essential reason.

**Solution:** This part is a little different than the last part, since it diverged due to a positive exponential term. Here if you try the same input equal to all 1, it won't work to show that the state grows without bound. Recall the solution of $x(t)$ with the initial condition at zero

$$x(t) = e^{at}x(0) + \int_0^t e^{a(t-\tau)} bu(\tau) d\tau.$$

Consider this case of bounded input $u(t) = e^{j2\pi t}$, which is bounded for all $t$. This proof will follow by counterexample (remember the question asks to show that *some* bounded input exists that will make the state grow without bound). Plugging this input and $a$ value in, we see

$$x(t) = \int_0^t e^{j2\pi(t-\tau)} be^{j2\pi\tau} d\tau = \int_0^t e^{j2\pi t} bd\tau.$$

Factoring out the terms that do not depend on $\tau$, we are left with

$$x(t) = be^{j2\pi t} \int_0^t d\tau.$$

Solving this integral, we get

$$x(t) = bte^{j2\pi t}.$$

Which clearly diverges as $t \to \infty$ since the magnitude $|x(t)| = |bt|$.

(e) Consider the discrete-time system

$$x[t+1] = -2x[t] + 2u[t] \tag{26}$$

with $x[0] = 0$.

**Is this system stable or unstable? If stable, prove it. If unstable, find a bounded input sequence $u[t]$ that causes the system to 'blow up'.**

**Solution:** The system is unstable. This can be seen by considering the input

$$u[t] = 1, 0, 0, 0, 0, \dots$$

| $t$ | 0 | 1 | 2 | 3 | $\cdots$ |
|---|---|---|---|---|---|
| $x[t]$ | 0 | 2 | -4 | 8 | $\cdots$ |
| $u[t]$ | 1 | 0 | 0 | 0 | $\cdots$ |
| $-2x[t]+2u[t]$ | 2 | -4 | 8 | -16 | $\cdots$ |

This results in the state at time $t \geq 1$,

$$x[t] = 2^t \cdot (-1)^{t+1}.$$

And so

$$|x[t]| = 2^t.$$

This is clearly exploding exponentially with $t$, not staying bounded.

(f) For the example in the previous part, **give an explicit sequence of inputs that are not zero but for which the state $x[t]$ will always stay bounded.** *(HINT: building off of the previous part see if you can find any input pattern that result in an oscillatory behavior.)*

**Solution:**

There is a case for which a non-zero bounded input results in a bounded output:

$$u[t] = 1, 2, 1, 2, 1, 2, \dots$$

| $t$ | 0 | 1 | 2 | 3 | $\cdots$ |
|---|---|---|---|---|---|
| $x[t]$ | 0 | 2 | 0 | 2 | $\cdots$ |
| $u[t]$ | 1 | 2 | 1 | 2 | $\cdots$ |
| $-2x[t]+2u[t]$ | 2 | 0 | 2 | 0 | $\cdots$ |

In this case, we get $x[t] = 0$ when $t$ is even, and $x[t] = 2$ when $t$ is odd. In fact, there are an infinite number of input sequences that would result in bounded outputs. But because we can find a single example of a bounded input sequence that leads to an unbounded output, the system is deemed unstable. We can't trust that we will only get nice inputs in engineering contexts.

(g) Consider the discrete-time real system with known solution:

$$x[t+1] = ax[t] + bu[t] \qquad x[t] = a^t x[0] + \sum_{\ell=0}^{t-1} a^{t-1-\ell} bu[\ell]$$

**Show that if the system is unstable (has $|a| > 1$), then a bounded input can result in an unbounded output.** Assume a zero initial condition here.

**Solution:** For simplicity, let's say that $u[t] = 1\forall t$. This gives us a new expression of

$$x[t+1] = ax[t] + bu[t] \qquad x[t] = a^t x[0] + \sum_{\ell=0}^{t-1} a^{t-1-\ell} b$$

At $t = 0$, we start with

$$x[0] = 0.$$

Then, for $t = 1$:

$$x[1] = \sum_{\ell=0}^{0} a^0 b = b.$$

But this quickly becomes problematic, consider

$$x[2] = \sum_{\ell=0}^{1} a^{t-1-\ell} b = ab + b.$$

Then,

$$x[t] = \sum_{\ell=0}^{t-1} a^{t-1-\ell} b = a^{t-1} b + \cdots + b$$

For $|a| > 1$, these terms quickly diverge. We can see this by looking at the geometric series sum for $|a| \neq 1$,

$$x[t] = \sum_{\ell=0}^{t-1} a^{t-1-\ell} b = b\frac{a^t - 1}{a-1}$$

This is diverging since $a^t$ has magnitude that growns without bound if $|a| > 1$.

(h) **[Optional] Repeat the previous part for the specific case of $a = -1$.**

**Solution:** This part follows very closely to the previous part. Let's start in considering the expanded sum form of the recurrence relation, with a non-specified input $u[t]$

$$x[t] = \sum_{\ell=0}^{t-1} a^{t-1-\ell} bu[t] = a^{t-1} bu[t-1] + a^{t-2} bu[t-2] + \cdots + abu[1] + bu[0]$$

Now, if we consider the bounded input of $u[t] = 1$ when $t$ is even, and $u[t] = -1$ when $t$ is odd. Namely, $u[t] = -1^t$. Observe that

$$x[t] = \sum_{\ell=0}^{t-1} (-1)^{t-1-\ell} b(-1)^l = \sum_{\ell=0}^{t-1} (-1)^{t-1} b.$$

Then, because the sum no longer depends on $\ell$, we are left with

$$x[t] = \sum_{\ell=0}^{t-1} (-1)^{t-1} b = t(-1)^{t-1} b.$$

This solution diverges over time with a bounded input.

(i) [Optional: this part was derived in lecture] Now consider the discrete-time stable case where $a$ is complex and has $|a| < 1$. **Show that as long as $|u[t]| < k$ for some $k$, that the solution $x[t]$ will be bounded for all time $t$.**

*(HINT: There are a few helpful facts about absolute values and inequalities that are helpful in such proofs. First: $|\sum_j a_j| \leq \sum_j |a_j|$. Second $|ab| = |a| \cdot |b|$. Third: $|e^{j\theta}| = 1$ no matter what real number $\theta$ is. And fourth, if $a_i > 0$ and $b_i > 0$, and $b_i \leq B$, then $\sum_i a_i b_i \leq \sum_i a_i B = B \sum a_i$. )*

**Solution:** We want to show that all bounded inputs will result in a bounded output.

First, we need to think about whether we want to consider the initial condition. Any reasonable definition of stability must let the bound on the output depend on the initial condition, otherwise there would always be a large enough initial condition that would violate the bound even if the system were obviously stable.

Let's say that $0 \leq |x[0]| < \infty$ (a finite initial condition with a value that may be nonzero) and that $u[t] \leq k$ for all $t$ (the input is bounded at all timesteps). If we can find a bound for $|x[t]|$ for all timesteps (i.e. $|x[t]| \leq \alpha$ where $\alpha$ is some positive value) then we have shown that this system is stable. The $\alpha$ is allowed to depend on both $k$ and $x[0]$.

Once again, it is good to first understand why this is true before setting out to prove it.

$$x[1] = ax[0] + bu[0] \tag{27}$$

$$x[2] = ax[1] + bu[1] \rightarrow x[2] = a^2 x[0] + abu[0] + bu[1] \tag{28}$$

$$x[3] = ax[2] + bu[2] \rightarrow x[3] = a^3 x[0] + a^2 bu[0] + abu[1] + bu[2] \tag{29}$$

Following this pattern we find:

$$x[t] = a^t x[0] + bu[t-1] + abu[t-2] + \cdots + a^{t-1} bu[0] = a^t x[0] + \sum_{i=0}^{t-1} bu[i] a^{t-i-1}$$

Finding the magnitude of $|x[t]|$ (we are trying to find an upper bound for this value), we get

$$|x[t]| = |a^t x[0] + bu[t-1] + abu[t-2] + \cdots + a^{t-1} bu[0]|.$$

At this point, it is useful to use the summation form so we can apply the hints more effectively

$$
\begin{aligned}
|x[t]| &= |a^t x[0] + \sum_{i=0}^{t-1} bu[i]a^{t-i-1}| && (30)\\
&\leq |a^t x[0]| + |\sum_{i=0}^{t-1} bu[i]a^{t-i-1}| && (31)\\
&\leq |a^t x[0]| + \sum_{i=0}^{t-1} |bu[i]a^{t-i-1}| && (32)\\
&= |a^t||x[0]| + \sum_{i=0}^{t-1} |b||u[i]||a^{t-i-1}| && (33)\\
&\leq |a|^t|x[0]| + \sum_{i=0}^{t-1} |b|k|a|^{t-i-1} && (34)\\
&= |a|^t|x[0]| + |b|k\sum_{i=0}^{t-1} |a|^{t-i-1} && (35)\\
&= |a|^t|x[0]| + |b|k\sum_{j=0}^{t-1} |a|^{j} && (36)\\
&< |a|^t|x[0]| + |b|k\sum_{j=0}^{\infty} |a|^{j} && (37)\\
&= |a|^t|x[0]| + \frac{|b|k}{1-|a|} && (38)
\end{aligned}
$$

At this point, we have used most of the hints that were given. We bounded an absolute value of a sum by the sum of absolute values in the first two inequalities (step 30 to 31 to 32), and then we used the fact that the absolute value of a product is the product of absolute values in the next two equalities (step 32 to 33). Then we used our bound on $u[t]$ to upper bound the overall summation (step 33 to 34). Then we pulled out a constant from a sum (step 34 to 35). We then convert the index of the summation from $i$ to $j$ using the change of variable that $j = t-1-i$ (step 35 to 36). Note that $j$ still will iterate from 0 to $t-1$. Then we used the fact that adding positive terms to a sum only makes it bigger to get the strict inequality (step 36 to 37), and finally we used the formula for an infinite geometric series to get the last equation (step 37 to 38). This was also the path taken in lecture.

But we still have the pesky $|a|^t$ term out in front. Since we are interested in $|a| < 1$, we know that $|a|^t < 1$ as well. Thus, we know that $|x[t]| < |x[0]| + \frac{|b|k}{1-|a|}$ and we are done.

As we have found a bound for $|x[t]|$ given any initial condition and bounded input, we have shown that this system must be BIBO stable.

These kind of manipulations of sums and inequalities are a part of basic mathematical maturity. You have seen some of this in your basic calculus courses, and we need to keep up the practice so that you get to the right level for later courses that touch probability (70 and then 126), optimization (127 and then 189), control (128 and then 221a), signal processing (120 and then 123), etc. The ideas of bounding are also critical for doing more advanced circuit analysis and design.

**5. Homework Process and Study Group**

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

**Contributors:**

- Ashwin Vangipuram.
- Nikhil Shinde.
- Alex Devonport.
- Sidney Buchbinder.
- Nathan Lambert.
- Anant Sahai.