EECS 16B Designing Information Devices and Systems II
Spring 2021 Discussion Worksheet Discussion 7A

The first question in this discussion clarifies some results cited in Note 7A and extends the results from Thursday's lecture on Discretization. The second question discusses system evolution over time, and sets the stage for discussing system stability.

1. A System Governed by Diffential Equations: Piecewise Constant Inputs

Working through this question will help you better understand differential equations with inputs, and the sampling of a continuous-time system of differential equations into a discrete-time view. These concepts are important for control, since it is often easier to think about doing what we want in discrete-time. This question should initially feel similar to dis03A, and in later subparts, we extend our analysis to the case of a vector differential equation.

(a) Consider the scalar system below:

$$\frac{d}{dt}x(t) = \lambda x(t) + u(t). \tag{1}$$

Further suppose that our input $u(t)$ of interest is piecewise constant over durations of width Δ. This is the same case we considered in dis03A. In other words:

$$u(t) = u(i\Delta) = u_d[i] \text{ if } t \in [i\Delta, (i+1)\Delta). \tag{2}$$

Similarly, for $x(t)$.

$$x_d[i] = x(i\Delta).$$

Given that we start at $x(i\Delta)$, where do we end up at $x((i+1)\Delta)$?

Answer: For $t \in [i\Delta, (i+1)\Delta)$, the differential equation takes the form

$$\frac{d}{dt}x(t) = \lambda x(t) + u(t) = \lambda x(t) + u_d[i] \tag{3}$$

We can guess that the form of the solution will be:

$$x(t) = \alpha e^{\lambda(t-i\Delta)} + \beta$$

Why is it in terms of $t - i\Delta$? Given the value $x_d[i] = x(i\Delta)$, we want to model the growth of x between $i\Delta$ and t, independently of the specific values of $i\Delta$ and t. We only care about their difference, given that we are in a specific interval.

To fit $x(t)$ to eq. (3), we equate the LHS of eq. (3) to the RHS. The LHS is:

$$\frac{d}{dt}x(t) = \frac{d}{dt}\left(\alpha e^{\lambda(t-i\Delta)} + \beta\right) = \lambda\alpha e^{\lambda(t-i\Delta)}$$

so equating the LHS with the RHS gives:

$$\begin{aligned}\lambda \alpha e^{\lambda(t-i\Delta)} &= \lambda x(t) + u_d[i] \\ &= \lambda\left(\alpha e^{\lambda(t-i\Delta)} + \beta\right) + u_d[i] \\ &= \lambda \alpha e^{\lambda(t-i\Delta)} + \lambda\beta + u_d[i] \\ \implies 0 &= \lambda\beta + u_d[i] \\ \implies \beta &= -\frac{u_d[i]}{\lambda}.\end{aligned}$$

Now we use the initial condition, $x(i\Delta) = x_d[i]$. Expanding $x(i\Delta)$ as per our guess,

$$x_d[i] = x(i\Delta) = \alpha e^{\lambda(i\Delta - i\Delta)} + \beta = \alpha + \beta$$

And using $\beta = -\frac{u_d[i]}{\lambda}$ we get

$$\begin{aligned}x_d[i] &= \alpha + \frac{-u_d[i]}{\lambda} \\ \implies \alpha &= x_d[i] + \frac{u_d[i]}{\lambda}.\end{aligned}$$

Now we have the values of both α and β, which is all we need to write $x(t)$ fully. So for $t \in [i\Delta, (i+1)\Delta)$ (which is the assumption we made for eq. (3) to hold),

$$\begin{aligned}x(t) = \alpha e^{\lambda(t-i\Delta)} + \beta &= \left(x_d[i] + \frac{u_d[i]}{\lambda}\right) e^{\lambda(t-i\Delta)} - \frac{u_d[i]}{\lambda} \\ &= e^{\lambda(t-i\Delta)} x_d[i] + \frac{e^{\lambda(t-i\Delta)} - 1}{\lambda} u_d[i]\end{aligned}$$

The reason we simplify in this manner is because we want to split the value of $x(t)$ into the effect of the initial condition $x_d[i]$, and the input $u_d[i]$. Now we can see how each independent part affects $x(t)$.

Since $x(t)$ is continuous across all t, $x_d[i+1] = x((i+1)\Delta)$. The continuity condition ensures that the function doesn't have bad behavior at only the points $i\Delta$ or $(i+1)\Delta$. Of course, these discontinuities don't happen in real systems, so our assumption makes sense. Thus

$$\begin{aligned}x_d[i+1] = x((i+1)\Delta) &= e^{\lambda((i+1)\Delta - i\Delta)} x_d[i] + \frac{e^{\lambda((i+1)\Delta - i\Delta)} - 1}{\lambda} u_d[i] \\ &= e^{\lambda\Delta} x_d[i] + \frac{e^{\lambda\Delta} - 1}{\lambda} u_d[i].\end{aligned}$$

This is the quantity we want.

(b) Now that we've found a one-step recurrence for $x_d[i+1]$ in terms of $x_d[i]$, we want to get an expression for $x_d[i]$ in terms of the original value $x(0) = x_d[0]$, and all the inputs u. This is so that we can eventually convert this function for $x_d[i]$ into a function for $x(t)$.

Unroll the implicit recursion you derived in the previous part to write $x_d[i+1]$ as a sum that involves $x_d[0]$ and the $u_d[j]$ for $j = 0, 1, \ldots, i$. The idea is to express the value of the discrete system at any arbitrary time solely in terms of where it started, and the accumulating set of inputs until that time.

For this part, to lighten the notation, feel free to just consider the discrete-time system in a simpler form

$$x_d[i+1] = a x_d[i] + b u_d[i] \tag{4}$$

and you don't need to worry about what a and b actually are in terms of λ and Δ.

Answer: Given that:

$$x_d[i+1] = ax_d[i] + bu_d[i]$$

we can build up a pattern. Starting from $i = 0$, we get

$$\begin{aligned}
x_d[1] &= ax_d[0] + bu_d[0] \\
x_d[2] &= ax_d[1] + bu_d[1] \\
&= a\left(ax_d[0] + bu_d[0]\right) + bu_d[1] \\
&= a^2 x_d[0] + b\left(au_d[0] + u_d[1]\right) \\
&= a^2 x_d[0] + abu_d[0] + bu_d[1] \\
x_d[3] &= ax_d[2] + bu_d[2] \\
&= a\left(a^2 x_d[0] + b\left(au_d[0] + u_d[1]\right)\right) + bu_d[2] \\
&= a^3 x_d[0] + b\left(u_d[2] + au_d[1] + a^2 u_d[0]\right) \\
&= a^3 x_d[0] + a^2 bu_d[0] + abu_d[1] + bu_d[2]
\end{aligned}$$

The idea is to collect terms with all the x_d's in one term and all the u_d's in the other term. Again, this separates out the effect of the initial condition $x_d[0]$ and all the inputs $u_d[j]$.
So, given this pattern, we guess

$$x_d[i] = a^i x_d[0] + b\sum_{j=0}^{i-1} a^{i-1-j} u_d[j]. \tag{5}$$

Let's check that this works! The way we do this is compute $x_d[i+1]$ through this formula, and also from eq. (4), and check that they're equal.

$$\begin{aligned}
x_d[i+1] = ax_d[i] + bu_d[i] &= a\left(a^i x_d[0] + b\sum_{j=0}^{i-1} a^{i-1-j} u_d[j]\right) + bu_d[i] \\
&= a^{i+1} x_d[0] + b\left(\sum_{j=0}^{i-1} a^{i-j} u_d[j]\right) + bu_d[i] \\
&= a^{i+1} x_d[0] + b\left(u_d[i] + \sum_{j=0}^{i-1} a^{i-j} u_d[j]\right) \\
&= a^{i+1} x_d[0] + b\sum_{j=0}^{i} a^{i-j} u_d[j]
\end{aligned}$$

This satisfies eq. (5), for $i+1$ and hence our guess was correct!

(c) Suppose we now have a continuous-time system of differential equations, that forms a vector differential equation. We express this with an input in vector form:

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

where $\vec{x}(t)$ is n-dimensional. We will assume that from here onwards, $\Delta = 1$, so that $i\Delta = i$.

Suppose further that the matrix A has distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. with corresponding eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$. We collect the eigenvectors together and form the matrix $V = [\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n]$. *(Hint: What's the significance of this information?)*

If we apply a piecewise constant control input $u_d[i]$ as in (2), and sample the system $\vec{x}(t)$ at time intervals $t = i$, what are the corresponding A_d and $\vec{b}_d$ in:

$$\vec{x}((i+1)\Delta) \equiv \vec{x}(i+1) = A_d\vec{x}(i) + \vec{b}_d u_d[i] \tag{6}$$

Answer: First, following the hint, we notice that with a full set of distinct eigenvalues and corresponding eigenvectors, we can change coordinates so that $\vec{x}(t) = V\vec{\breve{x}}(t)$ and $\vec{\breve{x}}(t) = V^{-1}\vec{x}(t)$. We use subscripts k to index into vectors, below.

We have:

$$(\breve{x}_k(i+1)) = e^{\lambda_k}\breve{x}_k(i) + \left(\frac{e^{\lambda_k}-1}{\lambda_k}\right)\left(V^{-1}b\right)_k u_d[i]$$

We are in a diagonal basis now! We can compose these individual scalar elements, and write the full vector as:

$$\vec{\breve{x}}(i+1) = \left(\begin{bmatrix} e^{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & e^{\lambda_n} \end{bmatrix}\right)\vec{\breve{x}}(i) + \left(\begin{bmatrix} \frac{e^{\lambda_1}-1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{e^{\lambda_n}-1}{\lambda_n} \end{bmatrix}\right)V^{-1}\vec{b}u_d[i] \tag{7}$$

Now, we name some of the terms above, for notational convenience going forward:

$$\Lambda_e = \begin{bmatrix} e^{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & e^{\lambda_n} \end{bmatrix} \qquad \Lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{1}{\lambda_n} \end{bmatrix}$$

So, with this new notation, we can write the second matrix in eq. (7) as:[1]

$$\begin{aligned}
\begin{bmatrix} \frac{e^{\lambda_1}-1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{e^{\lambda_n}-1}{\lambda_n} \end{bmatrix} &= \begin{bmatrix} \frac{e^{\lambda_1}}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{e^{\lambda_n}}{\lambda_n} \end{bmatrix} + \begin{bmatrix} \frac{-1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{-1}{\lambda_n} \end{bmatrix} \\
&= \begin{bmatrix} e^{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & e^{\lambda_n} \end{bmatrix}\begin{bmatrix} \frac{1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{1}{\lambda_n} \end{bmatrix} - \begin{bmatrix} \frac{1}{\lambda_1} & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \frac{1}{\lambda_n} \end{bmatrix} \\
&= \Lambda^{-1}\Lambda_e - \Lambda^{-1}I \\
&= \Lambda^{-1}(\Lambda_e - I)
\end{aligned}$$

[1] In a matrix product, if both matrices are diagonal, the product is commutative.

Now, using this form in the simplification, we find that:

$$\begin{aligned}\vec{x}(i+1) &= V\vec{\tilde{x}}(i+1)\\ &= V\left(\Lambda_e\vec{\tilde{x}}(i) + \left(\Lambda^{-1}\left(\Lambda_e - I\right)\right)V^{-1}\vec{b}u_d[i]\right)\\ &= \left(V\Lambda_e V^{-1}\right)\vec{x}(i) + \left(V\Lambda^{-1}\left(\Lambda_e - I\right)V^{-1}\vec{b}\right)u_d[i]\end{aligned}$$

Now, recall that our original goal was to write out A_d and $\vec{b}_d$, and we can do that now with our expression. We have:

$$A_d = V\Lambda_e V^{-1}$$

and

$$\vec{b}_d = V\Lambda^{-1}\left(\Lambda_e - I\right)V^{-1}\vec{b}$$

(d) We can write the relationship between the continuous and discrete systems as $\vec{x}(i) = \vec{x}_d[i]$, specifically since $\Delta = 1$. Suppose that $\vec{x}_d[0] = \vec{x}_0$. In the style of part b), unroll the implicit recursion you derived in the previous part to write $\vec{x}_d[i+1]$ as a sum that involves $\vec{x}_0$ and the $u_d[j]$ for $j = 0, 1, \ldots, i$.

For this part, feel free to just consider the discrete-time system in the simpler form

$$\vec{x}_d[i+1] = A_d\vec{x}_d[i] + \vec{b}_d u_d[i] \tag{8}$$

and you don't need to worry about what A_d and $\vec{b}_d$ actually are in terms of the original parameters.

Answer: There are two ways to approach this problem; the first is to extend the results of part (b) to the vector case by making appropriate substitutions, and the second is to re-derive the unrolled recursion and make a guess at the form of the solution in summation notation. Either method is valid, and for completeness, we present option 2 below.

Let's look at the pattern starting with $\vec{x}_d[1]$, given that $\vec{x}_d[i+1] = A_d\vec{x}_d[i] + \vec{b}_d u_d[i]$,

$$\begin{aligned}\vec{x}_d[1] &= A_d\vec{x}_d[0] + \vec{b}_d u_d[0]\\ \vec{x}_d[2] &= A_d\vec{x}_d[1] + \vec{b}_d u_d[1]\\ &= A_d(A\vec{x}_d[0] + \vec{b}_d u_d[0]) + \vec{b}_d u_d[1]\\ &= A_d^2\vec{x}_d[0] + A_d\vec{b}_d u_d[0] + \vec{b}_d u_d[1]\\ \vec{x}_d[3] &= A_d\vec{x}_d[2] + \vec{b}_d u_d[2]\\ &= A_d\left(A_d^2\vec{x}_d[0] + A_d\vec{b}_d u_d[0] + \vec{b}_d u_d[1]\right) + \vec{b}_d u_d[2]\\ &= A_d^3\vec{x}_d[0] + A_d^2\vec{b}_d u_d[0] + A_d\vec{b}_d u_d[1] + \vec{b}_d u_d[2]\end{aligned}$$

So, given this pattern, if we guess:

$$\vec{x}_d[i] = A_d^i\vec{x}_d[0] + \left(\sum_{j=0}^{i-1} u_d[j]A_d^{i-1-j}\right)\vec{b}_d \tag{9}$$

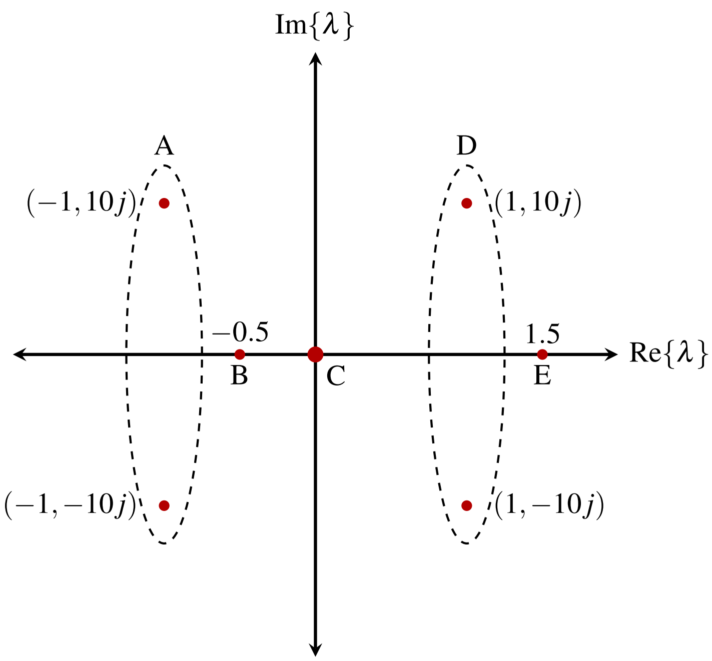
Then, let's see what we get for $\vec{x}_d[i+1]$, and make sure our guess is correct:

$$\begin{aligned}
\vec{x}_d[i+1] &= A_d\vec{x}_d[i] + \vec{b}_d u_d[i] \\
&= A_d\left(A_d^i\vec{x}_d[0] + \left(\sum_{j=0}^{i-1} u_d[j]A^{i-1-j}\right)\vec{b}_d\right) + \vec{b}_d u_d[i] \\
&= A_d^{i+1}\vec{x}_d[0] + \left(\left(\sum_{j=0}^{i-1} u_d[j]A^{i-j}\right) + u_d[i]\right)\vec{b}_d \\
&= A_d^{i+1}\vec{x}_d[0] + \left(\sum_{j=0}^{i} u_d[j]A^{i-j}\right)\vec{b}_d
\end{aligned}$$

This satisfies (9), for $i+1$ and hence our guess was correct!

2. Continuous-time System Responses

We have a differential equation $\frac{d\vec{x}(t)}{dt} = A\vec{x}(t)$, where A is diagonal and has eigenvalues λ. For systems A, D, which have more than 1 eigenvalue, this equation is a vector differential equation; in the other cases (B, C, E), it is a scalar differential equation. For each set of λ values plotted on the real-imaginary axis, sketch $\vec{x}_1(t)$ with an initial condition of $\vec{x}_1(0) = 1$. In the scalar case, $\vec{x}_1(t) \equiv x_1(t) \equiv x(t)$.



Answer: We recall that if the imaginary component of an eigenvalue is nonzero, the system will experience oscillations in its settling response. Any eigenvalues with nonzero imaginary components must appear in complex conjugate pairs, which explains why A and D are grouped by pairs of eigenvalues.

The solution to the system is:

$$\begin{aligned} x_1(t) &= x_1(0)e^{\lambda t} \\ &= x_1(0)e^{(\text{Re}\{\lambda\}+j\text{Im}\{\lambda\})t} \\ &= x_1(0)e^{\lambda_r t + j\lambda_j t} \\ &= x_1(0)e^{\lambda_r t} \cdot e^{j\lambda_j t} \\ &= x_1(0)e^{\lambda_r t} \cdot \left(\cos\left(\lambda_j t\right) + j\sin\left(\lambda_j t\right)\right) \end{aligned}$$

Taking just the real part:

$$x_1(t) = e^{\lambda_r t} \cdot \cos\left(\lambda_j t\right)$$

In the solution for $\vec{x}(t)$, the real component shows up in the exponent of an exponential. Depending on this real component, the value of $\vec{x}_1(t)$ will increase, decrease, or stay the same over time. The cos component always has magnitude at most 1; it appears in the form of oscillations whenever $\lambda_j \neq 0$.

We analyze each case sequentially to determine what the system's response might look like:

(A): These eigenvalues have negative real components, indicating that the system is stable, because the effect of the initial condition will decay over time. However, the imaginary component is nonzero, and these imaginary components are connected to sinusoids in ways that we have seen. This indicates that the response will experience oscillations around the value 0, as it decays from 1 to 0.

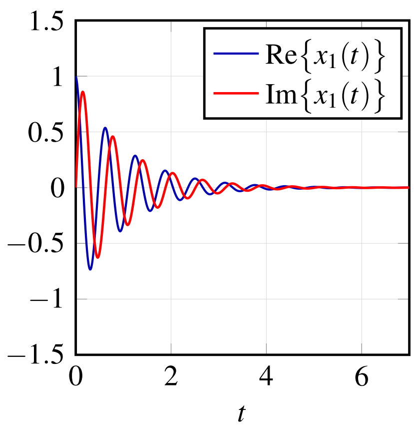
(B): We see a similar decay as in case (A) since the real component is negative, but here, since the imaginary component is zero, there will be no oscillations (there is a direct exponential decay from 1 to 0). The decay will also be a bit slower than in (A) since the real part has smaller magnitude.

(C): When the eigenvalue has a real component exactly on zero, then the (ideal) system here will neither grow nor decay over time; it will remain at the initial condition, which is 1. Since any disturbance to the system might impact the system's behavior, we can say this system is marginally stable.

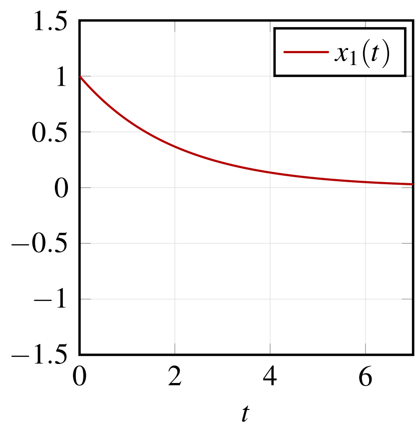
(D): The real component here is positive, which leads to exponential growth of the initial condition over time. Since the imaginary components of this complex conjugate pair of eigenvalues is nonzero, the system will oscillate as it grows.

(E): This case is similar to case (D) in that the real component is positive, so the system will grow over time. The real part is larger in magnitude, so the growth will proceed at a greater rate. Since there is no imaginary component here, the system will not oscillate, and instead grows directly from the initial condition towards ∞.

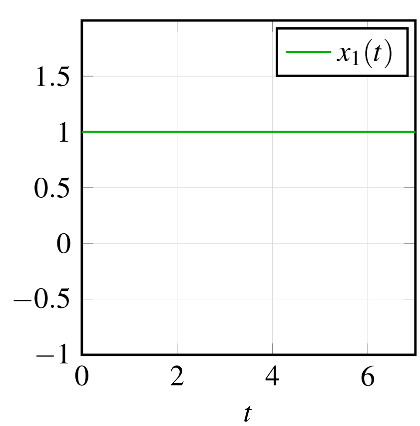
The plots below demonstrate the qualitative behavior described above (and include the imaginary component as well for systems A/D).



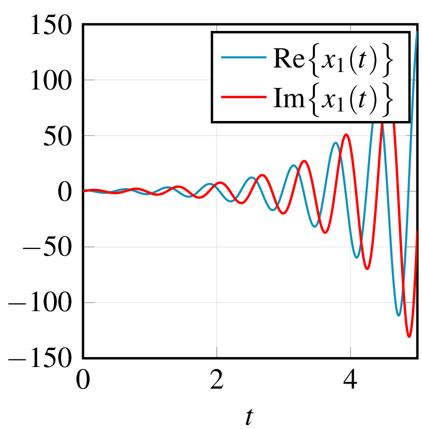
(a) System A ($\lambda = -1 \pm 10j$)



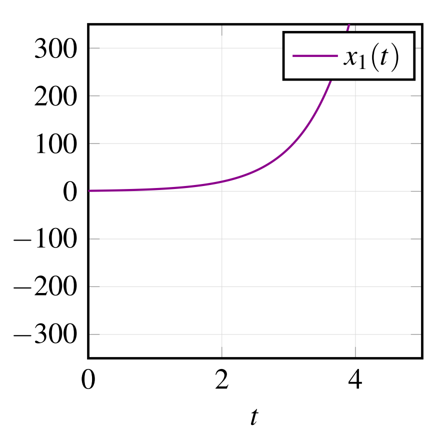
(b) System B ($\lambda = -0.5$)



(c) System C ($\lambda = 0$)



(d) System D ($\lambda = 1 \pm 10j$)



(e) System E ($\lambda = 1.5$)

Contributors:

- Neelesh Ramachandran.
- Druv Pai.
- Anant Sahai.
- Nikhil Shinde.
- Sanjit Batra.
- Aditya Arun.
- Kuan-Yun Lee.