1. **A system governed by differential equations being controlled with piecewise constant inputs**

Working through this question will help you understand better differential equations with inputs and the sampling of a continuous-time system of differential equations into a discrete-time view. This is important for control, since it is often easier to think about doing what we want in discrete-time.

(a) Consider the scalar system

\[
\frac{dx(t)}{dt} = \lambda x(t) + u(t). \tag{1}
\]

Suppose that our \( u(t) \) of interest is *constructed* to be piecewise constant over durations of width \( \Delta \), which we assume to be 1 for this problem. In other words:

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

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

**Answer:** Our differential equation takes the form,

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

where \( u(i) \) is a constant value of some input function \( u(t) \) at time \( t = i \). First we solve the differential equation by guessing

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

This gives,

\[
\frac{dx(t)}{dt} = \lambda \alpha e^{\lambda (t-i)}
\]

We know that this should equal to the right hand side of (3), so we get,

\[
\lambda \alpha e^{\lambda (t-i)} = \lambda x(t) + u(i) = \lambda (\alpha e^{\lambda (t-i)} + \beta) + u(i)
\]

\[\implies \lambda \alpha e^{\lambda (t-i)} = \lambda \alpha e^{\lambda (t-i)} + \lambda \beta + u(i)\]

Now using \( u(i) = u(i) \), we get,

\[
\beta = \frac{-u(i)}{\lambda}
\]

Further, we get,

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

And using, \( \beta = \frac{-u(i)}{\lambda} \) we get,

\[
x(i) = \alpha + \frac{-u(i)}{\lambda}
\]
\[ \alpha = x(i) + \frac{u(i)}{\lambda} \]

So, we get that,
\[ x(t) = (x(i) + \frac{u(i)}{\lambda})e^{\lambda(t-i)} - \frac{u(i)}{\lambda} \]
\[ \implies x(i) = x(i)e^{\lambda(t-i)} + (\frac{e^{\lambda(t-i)} - 1}{\lambda})u(i) \]

Thus,
\[ x(i+1) = x((i+1)) = x(i)e^{\lambda} + (\frac{e^{\lambda} - 1}{\lambda})u(i) \]

(b) Suppose that \( x(0) = x_0 \). **Unroll the implicit recursion you derived in the previous part to write** \( x(i+1) \) **as a sum that involves** \( x_0 \) **and the** \( u(j) \) **for** \( j = 0, \ldots, i \).

For this part, feel free to just consider the discrete-time system in a simpler form
\[ x(i+1) = ax(i) + bu(i) \tag{4} \]

and you don’t need to worry about what \( a \) and \( b \) actually are in terms of \( \lambda \) and \( \Delta \).

Your derivation here is actually an example of a simple proof by induction.
**Answer:** Let’s look at the pattern starting with \( x(1) \), given that \( x(i+1) = ax(i) + bu(i) \),
\[ x(1) = ax(0) + bu(0) \]
\[ x(2) = ax(1) + bu(1) \]
\[ \implies x(2) = a(ax(0) + bu(0)) + bu(1) = a^2(x(0)) + b(u(0))a + bu(1) \]
\[ x(3) = ax(2) + bu(2) = a(a^2(x(0)) + b(u(0))a + bu(1)) + bu(2) \]
\[ \implies x(3) = a^3x(0) + b(u(0))a^2 + u(1)a + u(2) \]

So, given this pattern, if we guess,
\[ x(i) = a^i x(0) + b\sum_{j=0}^{i-1} u(j)a^{i-1-j} \tag{5} \]

Then, let’s see what we get for \( x(i+1) \),
\[ x(i+1) = ax(i) + bu(i) = a(d^i x(0) + b\sum_{j=0}^{i-1} u(j)a^{i-1-j})) + bu(i) \]
\[ \implies x(i+1) = d^{i+1}x(0) + b(\sum_{j=0}^{i-1} u(j)a^{i-1-j} + u(i)) = d^{i+1}x(0) + b(\sum_{j=0}^{i} u(j)a^{i-j}) \]

This satisfies \( 5 \) for \( i+1 \) and hence our guess was correct!

This turns out to be a proof by induction, with base case \( x(1) = ax(0) + bu(0) \). Going from \( i \) to \( (i+1) \) is the inductive step. This is how we transform a recursively found pattern into a rigorous proof!
(c) Suppose we have a system of differential equations with an input that we express in vector form:

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

(6)

where \( \vec{x}_c(t) \) is \( n \)-dimensional.

Suppose further that the matrix \( A \) has distinct eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_n \) with corresponding eigenvectors \( \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \). Collect the eigenvectors together into a matrix \( V = [\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n] \).

If we apply a piecewise constant control input \( u(t) \) as in (2), and sample the system \( \vec{x}(i) = \vec{x}_c(i) \), **what are the corresponding** \( A_d \) and \( \vec{b}_d \) in:

\[
\vec{x}(i+1) = A_d \vec{x}(i) + \vec{b}_d u(i).
\]

(7)

**Answer:** First, we change coordinates so that \( \vec{x}_c(t) = V \vec{x}(t) \) and \( \vec{x}(t) = V^{-1} \vec{x}_c(t) \).

We have,

\[
(\vec{x}(i+1))[j] = (e^{\lambda_j}) (\vec{x}(i))[j] + \left( \frac{e^{\lambda_j} - 1}{\lambda_j} \right) (V^{-1} \vec{b})[j] (u(i))
\]

\[
\vec{x}(i+1) = \begin{bmatrix} e^{\lambda_1} & 0 & \cdots \\ \vdots & \ddots & 0 \\ 0 & \cdots & e^{\lambda_n} \end{bmatrix} \vec{x}(i) + \begin{bmatrix} \frac{e^{\lambda_1} - 1}{\lambda_1} & 0 & \cdots \\ \vdots & \ddots & 0 \\ 0 & \cdots & \frac{e^{\lambda_n} - 1}{\lambda_n} \end{bmatrix} V^{-1} \vec{b} u(i)
\]

Now we define the following notations,

\[
E_\Lambda = \begin{bmatrix} e^{\lambda_1} & 0 & \cdots \\ \vdots & \ddots & 0 \\ 0 & \cdots & e^{\lambda_n} \end{bmatrix}
\]

\[
\Lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & \cdots \\ \vdots & \ddots & 0 \\ 0 & \cdots & \frac{1}{\lambda_n} \end{bmatrix}
\]

So,

\[
x(i+1) = V \vec{x}(i+1) = (VE_\Lambda V^{-1}) x(i) + (VA^{-1} (E_\Lambda - I) V^{-1} \vec{b}) u(i)
\]

Hence,

\[
A_d = (VE_\Lambda V^{-1})
\]

and

\[
\vec{b}_d = (VA^{-1} (E_\Lambda - I) V^{-1} \vec{b})
\]
(d) Suppose that $\vec{x}(0) = \vec{x}_0$. Unroll the implicit recursion you derived in the previous part to write $\vec{x}(i+1)$ as a sum that involves $\vec{x}_0$ and the $u(j)$ for $j = 0, \ldots, i$.

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

$$\vec{x}(i+1) = A\vec{x}(i) + \vec{b}u(i)$$

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

**Answer:** Let’s look at the pattern starting with $\vec{x}(1)$, given that $\vec{x}(i+1) = A\vec{x}(i) + \vec{b}u(i)$,

$$\vec{x}(1) = A\vec{x}(0) + \vec{b}u(0)$$

$$\vec{x}(2) = A\vec{x}(1) + \vec{b}u(1)$$

$$\implies \vec{x}(2) = A(A\vec{x}(0) + \vec{b}u(0)) + \vec{b}u(1) = A^2(\vec{x}(0)) + (u(0))A\vec{b} + \vec{b}u(1)$$

$$\vec{x}(3) = A\vec{x}(2) + \vec{b}u(2) = A(A^2(\vec{x}(0)) + (u(0))A\vec{b} + \vec{b}u(1)) + \vec{b}u(2)$$

$$\implies \vec{x}(3) = A^3\vec{x}(0) + (u(0))A^2 + (u(1))A + (u(2))\vec{b}$$

So, given this pattern, if we guess,

$$\vec{x}(i) = A^i\vec{x}(0) + \left(\sum_{j=0}^{i-1} u(j)A^{i-1-j}\vec{b}\right)$$

(9)

Then, let’s see what we get for $\vec{x}(i+1)$,

$$\vec{x}(i+1) = A\vec{x}(i) + \vec{b}u(i) = A(A^i\vec{x}(0) + \left(\sum_{j=0}^{i-1} u(j)A^{i-1-j}\vec{b}\right)) + \vec{b}u(i)$$

$$\implies \vec{x}(i+1) = A^{i+1}\vec{x}(0) + \left((\sum_{j=0}^{i-1} u(j)A^{i-1-j}) + u(i)\vec{b}\right) = A^{i+1}\vec{x}(0) + \left((\sum_{j=0}^{i} u(j)A^{i-j})\vec{b}\right)$$

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

This turns out to be a proof by induction, with base case $\vec{x}(1) = A\vec{x}(0) + \vec{b}u(0)$. Going from $i$ to $(i+1)$ is the inductive step. This is how we transform a recursively found pattern into a rigorous proof!

2. Controlling states by designing sequences of inputs

This is something that you saw in 16A in the Segway problem. In that problem, you were given a semi-realistic model for a segway. Here, we are just going to consider a the following peculiar matrices chosen for intuitive ease of understanding what is going on:

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{bmatrix} \quad \vec{b} = \begin{bmatrix}
0 \\
0 \\
0 \\
1 \\
\end{bmatrix}$$

Let’s assume we have a discrete-time system that follows the following “difference equation.”

$$\vec{x}(t + 1) = A\vec{x}(t) + \vec{b}u(t).$$
(a) We are given the initial condition $\vec{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. Let’s say we want to achieve $\vec{x}(m) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ for some specific $m \geq 0$. We don’t need to stay there, we just want to be in this state at that time. What is the smallest $m$ such that this is possible? What is our choice of sequence of inputs $u(i)$?

**Answer:** To ease notation, let $x(n) = \begin{bmatrix} x(n)[1] \\ x(n)[2] \\ x(n)[3] \\ x(n)[4] \end{bmatrix}$.

Note that $\vec{x}(1) = A\vec{x}(0) + \vec{b}u(0) = \begin{bmatrix} x(0)[2] \\ x(0)[3] \\ x(0)[4] \\ u(0) \end{bmatrix}$.

and so we see that if $n \geq 4$,

$$\vec{x}(n) = \begin{bmatrix} u(n-4) \\ u(n-3) \\ u(n-2) \\ u(n-1) \end{bmatrix}.$$ 

Hence, the smallest $m$ is equal to 4, with $u(i) = (1, 2, 3, 4, \ldots)$ where the remaining terms are not relevant.

(b) What if we started from $\vec{x}(0) = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$? What is the smallest $m$ and what is our choice of $u(i)$?

**Answer:** We see that $\vec{x}(1) = A\vec{x}(0) + \vec{b}u(0) = \begin{bmatrix} x(0)[2] \\ x(0)[3] \\ x(0)[4] \\ u(0)[4] \end{bmatrix}$.

so we only need $m = 1$ and input $u(i) = (4, \ldots)$.

(c) What if we started from $\vec{x}(0) = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix}$? What is the smallest $m$ and what is our choice of $u(i)$?

**Answer:** We would still need $m \geq 4$ to achieve this. Input $u(i)$ should be equal to $(1, 2, 3, 4, \ldots)$.

**Contributors:**

- Anant Sahai.
- Nikhil Shinde.
• Sanjit Batra.
• Aditya Arun.
• Kuan-Yun Lee.