Linear Infinite State Systems

Last time we
- Looked at state machines with feedback
- Considered state machines with constant or absent external input and determined whether they are well-formed
- Considered state machines with external input and feedback, and examined accuracy of output tracking

Today we will
- Look at systems with an infinite number of states (Reals)
- Examine property of linearity
- Formulate system definition and calculate state response

Infinite State Machines

Chapter 5 addresses systems with an infinite number of states:

\[ States = \text{Reals}^N \]

We refer to N as the **dimension** of the system.

It would be very hard to describe this state machine with a diagram, so we stick to the 5-tuple definition. The systems we will consider have

\[ Inputs = \text{Reals}^M \]

\[ Outputs = \text{Reals}^K \]

We will look at the general case of multiple-input, multiple-output (MI MO) systems as well as single-input, single-output (SI SO).
Input, Output, and State Sequences

It is important to remember that, while for each $n \in \text{Naturals}_0$

\[
x(n) \in \text{Reals}^M \\
s(n) \in \text{Reals}^N \\
y(n) \in \text{Reals}^K
\]

when written without the index, $x$, $s$, and $y$ are sequences:

\[
x \in [\text{Naturals}_0 \rightarrow \text{Reals}^M] \\
s \in [\text{Naturals}_0 \rightarrow \text{Reals}^N] \\
y \in [\text{Naturals}_0 \rightarrow \text{Reals}^K]
\]

Defining the Infinite-State Machine

We use the same 5-tuple to define the system:

\[(\text{States}, \text{Inputs}, \text{Outputs}, \text{update}, \text{initialState})\]

Recall that $\text{update}$ takes the current state and input as arguments, and provides the next state and output, so

\[
\text{update} : \text{Reals}^N \times \text{Reals}^M \rightarrow \text{Reals}^N \times \text{Reals}^K
\]

Breaking up $\text{update}$ into a state update equation $\text{nextState}$ and separate output equation $\text{output}$ can make things easier:

\[
s(n+1) = \text{nextState}(s(n), x(n)) \\
y(n) = \text{output}(s(n), x(n))
\]
Time

- We have been referring to the index $n$ as the step number.
- The systems in Chapter 5 can be viewed as **discrete-time** versions of ordinary differential equations.
- The step number is often identified with some time $n\delta$, where $\delta$ is some fixed time between steps. So in this chapter, we call $n$ the **time index**.
- The *update* function may depend on the time index $n$. These systems are called **time-varying**.
- In this chapter, we will consider systems that do not depend on $n$, called **time-invariant** systems.
- We require that the system input $x(n)$ and output $y(n)$ have real, physical values at each time step $n$. Thus stuttering is not allowed, and the *absent* symbol is not used.

Linearity

- A function $f: \text{Reals}^N \rightarrow \text{Reals}^M$ is **linear** if for all $a \in \text{Reals}$, $u \in \text{Reals}^N$, and $v \in \text{Reals}^N$,

$$f(au) = a f(u) \quad \text{homogeneity}$$

and

$$f(u + v) = f(u) + f(v) \quad \text{additivity}$$

- These two properties together are equivalent to the **superposition** property:

$$\forall a, b \in \text{Reals}, \text{ and } u, v \in \text{Reals}^N, \ f(au + bv) = a f(u) = b f(v)$$

- Every linear function can be represented by a matrix $A$ such that

$$f(u) = A u$$
Linear Systems

For the systems considered in Chapter 5, with

\[ States = \text{Reals}^N \quad Inputs = \text{Reals}^M \quad Outputs = \text{Reals}^K \]

we say that a system is linear if

- \( initialState \) is an \( N \)-tuple of zeros, and
- both \( nextState \) and \( output \) are linear functions.

If the system is also time-invariant (\( update \) does not depend on the time index \( n \)), then we say that the system is \textit{linear time-invariant} (LTI).

Representing Linear Systems

- Recall that \( nextState \) takes \( s(n) \in \text{Reals}^N \) and \( x(n) \in \text{Reals}^M \) as arguments, and it provides \( s(n+1) \in \text{Reals}^N \).

- So if we view the ordered pair \( (s(n), x(n)) \) as an \( N+M \)-tuple, with the first \( N \) elements consisting of \( s(n) \) and the last \( M \) elements being \( x(n) \), then any linear \( nextState \) function can be expressed as

  \[ nextState(s(n),x(n)) = P (s(n), x(n)) \]

  where \( P \) is an \( N \times N+M \) matrix.

- Notice that the first \( N \) columns of \( P \) get multiplied by \( s(n) \), and the last \( M \) columns of \( P \) get multiplied by \( x(n) \).

- So if we let \( A \) denote the first \( N \) columns of \( P \), and \( B \) denote the last \( M \) columns of \( P \), we can write

  \[ nextState(s(n),x(n)) = A s(n) + B x(n) \]
[A, B, C, D] Representation

Similarly, we can represent the output function as

\[ \text{output}(s(n), x(n)) = C \ s(n) + D \ x(n) \]

where \( C \) is a \( K \times N \) matrix and \( D \) is a \( K \times M \) matrix.

We thus obtain the state-space model of the system:

\[
\begin{align*}
  s(n+1) &= A \ s(n) + B \ x(n) \\
  y(n) &= C \ s(n) + D \ x(n)
\end{align*}
\]

This is also known as the [A, B, C, D] representation.

Obtaining the State Response

Suppose we have an input sequence \( x \) and an initial state \( s(0) \).

Let's try to calculate the sequence of states, \( s \).

\[
\begin{align*}
  s(1) &= A \ s(0) + B \ x(0) \\
  s(2) &= A \ s(1) + B \ x(1) \\
  &= A[A \ s(0) + B \ x(0)] + B \ x(1) \\
  &= A^2 \ s(0) + A \ B \ x(0) + B \ x(1) \\
  s(3) &= A \ s(2) + B \ x(2) \\
  &= A[A^2 \ s(0) + A \ B \ x(0) + B \ x(1)] + B \ x(2) \\
  &= A^3 \ s(0) + A^2 \ B \ x(0) + A \ B \ x(1) + B \ x(2)
\end{align*}
\]

In general,

\[
  s(n) = A^n \ s(0) + \sum_{m=0}^{n-1} A^{n-1-m} B \ x(m)
\]

Once you have \( s(n) \), find \( y(n) \) by plugging into

\[ y(n) = C \ s(n) + D \ x(n) \]
Zero-State and Zero-Input Response

Notice that our expression

\[ s(n) = A^n s(0) + \sum_{m=0}^{n-1} A^{n-1-m} B x(m) \]

is the sum of two terms: \( A^n s(0) \) and \( \sum_{m=0}^{n-1} A^{n-1-m} B x(m) \).

The term \( A^n s(0) \) is known as the zero-input state response since it is the sequence of states that occurs if each \( x(n) = 0 \).

The term \( \sum_{m=0}^{n-1} A^{n-1-m} B x(m) \) is known as the zero-state state response since it is the sequence of states that occurs if \( s(0) = 0 \).

Example

Exercise 5.10: Consider the system with

\[ A = \sigma \begin{bmatrix} \cos (\pi/6) & \sin (\pi/6) \\ -\sin (\pi/6) & \cos (\pi/6) \end{bmatrix} \quad s(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

Sketch the zero-input state response for \( n = 0, 1, ..., 12 \) for

a) \( \sigma = 0 \)

b) \( \sigma = 0.9 \)

c) \( \sigma = 1.1 \)