Overview

Analytical solutions of the state variable or input-output models is difficult or impossible for higher-order or nonlinear systems. Thus, numerical (or computer-based) solutions of these differential equations have become increasingly popular. This chapter provides a brief outline describing the use of some special-purpose MATLAB commands for simulating the response of linear, time invariant systems (these are systems which are governed by linear differential equations with constant coefficients).

Some MATLAB functions presented in this chapter are available on with MATLAB’s Control Systems Toolbox. The Control Systems Toolbox comes with the Student Edition of MATLAB. For more information about MATLAB products, see the MathWorks web site at http://www.mathworks.com.

Before beginning this chapter, you should be able to:

- Write state variable models for electric circuits (Chapter 2.6.1)
- Set up arrays in MATLAB
- Use MATLAB to create two-dimensional plots

After completing this chapter, you should be able to:

- Use MATLAB to simulate the impulse response of an electrical circuit
- Use MATLAB to simulate the step response of an electrical circuit
- Use MATLAB to plot the state trajectory of an electrical circuit

This chapter requires:

- N/A
MATLAB’s Control Systems Toolbox contains a number of special-purpose commands for simulating the response of linear time invariant systems. Among these commands are commands specific to determining the step response and natural response of systems; we will restrict our attention to these commands in this chapter. Later courses in your engineering curriculum will most likely present more general-purpose MATLAB commands.

**List of Commands and Basic Syntax**

We will use only MATLAB’s `step` and `initial` commands in this chapter. The `step` command calculates a unit step response for the system, while the `initial` command calculates the natural response of a system to some set of initial conditions. Basic synax for these commands is provided below.

**step** - The command \([y,x,t]=\text{step}(A,b,c,d)\) returns the step response of the state variable model described by the matrices \(A, b, c\) and \(d\). The vector \(y\) contains the system output, the matrix \(x\) contains the states, and the vector \(t\) contains the time samples. The command \([y,x]=\text{step}(A,b,c,d,t)\) returns the step response as above, but calculated over the specified time vector \(t\). \(\text{step}(A,b,c,d)\) with no left-hand arguments results in a plot of the step response output.

**initial** - Response of linear system to an initial condition. \([y,x,t]=\text{initial}(A,b,c,d,x0)\) returns the response of the system described by the state space model \(\{A,b,c,d\}\) to an initial condition contained in the vector \(x0\).

**Example: Use of Step Command to Determine System Response**

Determine and plot the response of the system shown below if \(u(t) = \begin{cases} 0V, t < 0 \\ 2V, t \geq 0 \end{cases}\) and the circuit is initially relaxed (i.e. all voltages and currents in the system are initially zero). Also plot the state trajectory for this input.

In order to simulate the system’s response using MATLAB, we need to generate a state variable model of the system. We define states as the current through the inductor and the voltage across the capacitor, as shown in the above figure. Applying KVL around the circuit loop, we obtain:

\[ u = x_1 + 0.5 \dot{x}_1 + x_2 \]
Applying KCL at the node between the inductor and capacitor results in:

\[ x_1 = 0.5 \dot{x}_2 \]

Rearranging the above equations and placing them in matrix form results in:

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} =
\begin{bmatrix}
-2 & -2 \\
2 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} +
\begin{bmatrix}
2 \\
0
\end{bmatrix} u(t)
\]

Since the output \( y = x_2 \), the output equation is:

\[ y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0 \cdot u \]

To simulate the response of the system we first need to input the state variable model. The commands to do this are:

\[
\begin{align*}
& \gg A = \begin{bmatrix}-2 & -2; 2 & 0\end{bmatrix}; \\
& \gg b = [2; 0]; \\
& \gg c = [0 1]; \\
& \gg d = 0;
\end{align*}
\]

The >> symbols denote the command prompt at MATLAB’s command window; they are included here to emphasize MATLAB commands.

We can simulate the response of this system to the desired input by using MATLAB’s step command. The step command assumes that the system is initially relaxed and the input to the system is

\[ u(t) = \begin{cases} 
0, & t < 0 \\
1, & t \geq 0
\end{cases} \]

The input to our system is exactly twice this input, so we can simply scale our output by a factor of two. (This works because the system is linear – don’t try this with a nonlinear system!) The appropriate commands are (note that we have to scale both the output and the states, since we will be plotting the output response and the state trajectory):

\[
\begin{align*}
& \gg [y,x,t] = \text{step}(A,b,c,d); \\
& \gg y = 2 \cdot y; \\
& \gg x = 2 \cdot x;
\end{align*}
\]

The final step is to plot the responses. Plotting the output response can be accomplished with the following commands:

\[
\begin{align*}
& \gg \text{figure} \\
& \gg \text{plot}(t,y) \\
& \gg \text{title}('System output vs. time') \\
& \gg \text{xlabel}('Time, sec') \\
& \gg \text{ylabel}('Response, Volts')
\end{align*}
\]

Which results in the figure below:
We can plot the state trajectory by plotting the second state vector, \( x_1(t) \), vs. the first state vector, \( x_2(t) \). MATLAB returns the first state vector as the first column of the \( x \) matrix, the second state vector as the second column of the \( x \) matrix, and so on. Thus, we can plot the state trajectory with the following commands:

```matlab
>> plot(x(:,1),x(:,2))
>> grid
>> xlabel('X_1(t)')
>> ylabel('X_2(t)')
>> title('State trajectory for Example 1')
```
2.6.2: Numerical Simulation of System Responses using MATLAB

State trajectory for Example 1

Increasing time