EE 105: First-order systems in Simulink

Jay Farrell, College of Engineering, University of California, Riverside

Abstract

The objectives of this laboratory are to introduce Simulink while exercising systems concepts such as transfer functions, time constants, pole locations, DC gain, and frequency response.

All of these concepts may be on exams. Each is discussed in the course lecture notes (check the index).

1 Background

This lab will focus on the analysis of linear ordinary differential equations (ODE's). Design is the choice of system parameters to achieve specifications. Design analysis will use *transfer functions*. Testing through simulation will use state space.

We will first develop ideas using first-order systems. Those ideas will then be generalized to higher order systems.

1.1 Terminology

The *transfer function* $H(s)$ from $u(t)$ to $y(t)$ is defined as the ratio of $Y(s)$ to $U(s)$:

$$H(s) = \frac{Y(s)}{U(s)},$$

when the initial state (i.e., all initial conditions) is zero.

For a linear system, without pure delay, the transfer function will be the ratio of a numerator polynomial in s to a denominator polynomial in s. Let $N(s)$ denote the numerator polynomial of $H(s)$. The roots of the equation $N(s) = 0$ are the *zeros* of the system. Let $D(s)$ denote the denominator polynomial of $H(s)$. The roots of the equation $D(s) = 0$ are the *poles* of the system. The location of the poles determines the stability properties of the system and greatly affect the transient behavior. If the real part of all poles are in the left half plane (LHP), then the system is *stable*. If the real part of any pole is in the RHP, then the system is *unstable*.

Steady state applies for values of t large enough that all transients are effectively finished. Steady state applies only for stable systems. Usually this is steady state is considered to be achieved for times t being greater than four times the dominant time constant.

The *DC Gain* of a stable system is the ratio of the magnitude of the steady state output to the magnitude of the applied step input. By the Final Value Theorem to $Y(s)$, the DC Gain can be computed as

$$DC\ Gain = \lim_{s \to 0} H(s). \tag{1}$$

For a sinusoidal input $u(t) = A\cos(\omega t)$ to a system with stable transfer function $H(s)$ as $t \to \infty$

$$y(t) \quad \to \quad M(w)A\cos(\omega t + \Phi(\omega))$$

where $H(s)\big|_{s=j\,\omega} = M(w)e^{j\,\Phi(\omega)}$ represents a complex number written in polar form. Graphs of $20\,log(M(w))$ and $\Phi(\omega)$ plotted versus ω on a $\log_{10}$ axis are called *Bode plots*. Matlab provides routines for plotting the Bode plots (type 'help bode').

The lecture notes contain more detailed discussion of the terms: transfer function, time constant, poles, gain, DC gain, phase shift, dominant pole, frequency response. To find the discussion, look in the index in the back of the lecture notes.

1.2 First-Order Systems

First-order, linear state-space models have the form

$$\frac{d}{dt}x(t) \quad = \quad -a\ x(t) + b\ u(t) \tag{2}$$
$$y(t) \quad = \quad c\ x(t) \tag{3}$$

where $u(t)$ and $y(t)$ are the input and output signals and $x(t)$ is the state of the system. The signals $u(t)$, $x(t)$, and $y(t)$ are each scalars. The parameters a, b, and c are real constants with numeric values determined by the designer of the system.

1.2.1 Analytic Solutions

The analytic method to determine the response $y(t)$ of the system to the input $u(t)$ is found using the Laplace transform of eqn. (2):

$$\begin{aligned} s\,X(s) - x(0) &= -a\,X(s) + b\,U(s) \\ (s-a)X(s) &= x(0) + b\,U(s) \\ X(s) &= \frac{x(0)}{s+a} + \frac{b}{s+a}U(s) \end{aligned}$$

and similarly

$$\begin{aligned} Y(s) &= c\,X(s) \\ Y(s) &= \frac{c\,x(0)}{s+a} + \frac{c\,b}{s+a}U(s). \end{aligned} \tag{4}$$

The inverse Laplace transform of eqn. (4) provides the general output response in the time-domain:

$$y(t) = c\,e^{-a\,t}x(0) + \int_0^t c\,e^{-a(t-\alpha)}b\,u(\alpha)\,d\alpha, \text{ for } t \geq 0, \tag{5}$$

where α is a dummy variable used only for integration. The two terms in the right hand side of eqn. (5) each have names related to their physical meaning.

Initial condition (or zero input) response: If the input $u(t) = 0$ for all $t \geq 0$, then $y(t) = ce^{-at}x(0)$. This is the part of the response that is only due to the initial coondition $x(0)$.

Forced (or zero state) response: If the initial value of the state is zero (i.e., $x(0) = 0$), then $y(t) = \int_0^t ce^{-a(t-\tau)}bu(\tau)d\tau$. This is the part of the response that is only due to the input.

The general solution is the sum of the zero input and the zero state responses.

1.2.2 First-order Transfer Function

From eqn. (4) we see that the transfer function for the system described by eqns. (2–3) is

$$H(s) = \frac{c\,b}{s+a}. \tag{6}$$

The polynomial in the denominator of the transfer function is called the *characteristic polynomial.* The roots (or zeros) of the characteristic polynomial are the *poles* of the system. The pole locations are critical to the determination of the stability of the system and affect the transient response.

For the first-order system of this lab, the pole is at $s = -a$. The *time constant* of a first-order system is $\tau = \frac{1}{|a|}$.

1.2.3 Impulse Response

The *impulse response* is the inverse Laplace transform of the transfer function. For the system described by eqns. (2–3) the impulse response is

$$h(t) = c\,e^{-at}\,b \tag{7}$$

which plays an important role in the convolution integral of the zero state response.

1.2.4 Stability

If $a > 0$, then the system is *stable*. Its zero input response will converge to zero as $t \to \infty$.

If $a < 0$, then the system is *unstable*. The magnitude of the system response will diverge toward infinity as $t \to \infty$. In the remainder of this document, we will only be interested in stable systems.

1.2.5 Forced Responses

Certain types of inputs are common enough that they are often used to specify the performance of systems. This section analyzes the response of the first-order state-space system to these common input signals.

Zero Input ($u(t) = 0$) The response of a first-order system, when the input is zero, is completely determined by the initial condition:

$$y(t) = c\,e^{-a\,t}\,x(0) = c\,e^{\frac{-t}{\tau}}\,x(0)$$

where the time constant $\tau = \frac{1}{a}$ is discussed in Section 1.2.2.

For a stable system, the time constant characterizes the rate of decay of the initial condition. At $t = \tau$, the output will have deceased to approximately 37% of its initial value, since

$$y(\tau) = c\,e^{-1}\,x(0) \approx 0.37\,y(0).$$

When a person has data corresponding to a first-order system, this formula provides one method for estimating the time constant (i.e., estimating the parameter a). The person simply finds the value of time at which $y(t) = 0.37\,y(0)$.

A second method to estimate the value of the time constant is to use the slope of $y(t)$ at $t = 0$. Note that by differentiation of $y(t) = c\,e^{-a\,t}\,x(0)$ it is straightforward to show that $\frac{dy}{dt}(t) = -a\,c\,e^{-a\,t}\,x(0) = -a\,y(t)$. The line tangent to the graph of $y(t)$ at $t = 0$ is

$$v(t) = y(0) - a\,y(0)\,t.$$

This line intersects the t-axis at $t = \frac{1}{a} = \tau$. To use this method, the person draws a line tangent to $y(t)$ at $t = 0$ and measures the time at which the line intersects the t-axis.

Step Input ($u(t) = A\ 1(t)$): The unit step function is represented as

$$1(t) = \begin{cases} 0 & \text{for} \quad t < 0 \\ 1 & \text{for} \quad t \geq 0. \end{cases}$$

The Laplace transform of this step function is $U(s) = \frac{A}{s}$. Therefore, using partial fractions on eqn. (4)

$$\begin{aligned} Y(s) &= \frac{c\,x(0)}{s+a} + \frac{c\,b\,A}{(s+a)\,s} \qquad (8) \\ &= \left(c\,x(0) - \frac{c\,b\,A}{a}\right)\frac{1}{s+a} + \frac{c\,b\,A}{a\,s}. \end{aligned}$$

Using the inverse Laplace transform, the time response of the system is

$$y(t) = \left(c\,x(0) - \frac{c\,b}{a}\,A\right)e^{-a\,t} + \frac{c\,b}{a}\,A, \quad \text{for } t \geq 0. \tag{9}$$

For first-order stable linear systems, because since $e^{-4} < 2\%$, steady state is often considered to apply for any $t \geq 4\,\tau$.

By eqn. (9) as $t \to \infty$, $y(t) \to \frac{c\,b}{a}\,A = y_\infty$. Therefore, the DC gain is $\frac{y_\infty}{A} = \frac{c\,b}{a}$.

The DC gain can also be computed directly from the transfer function without using partial fractions. To derive the formula, we apply the Final Value Theorem to $Y(s)$:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sH(s)U(s) = H(0)A. \tag{10}$$

Therefore, the DC gain is (again) $\frac{\lim_{t\to\infty} y(t)}{A} = H(0)$. This method for computing the DC gain easily extends to higher order systems. For the first-order example of this lab, the DC gain is $\frac{cb}{a}$.

Sinusoidal Input: **($u(t) = A\cos(\omega t)$ for $t \geq 0$)** When the input is the sinusoid $u(t) = A\cos(\omega t)$ for $t \geq 0$, where ω is the radian frequency, the output response is computed using Laplace methods as

$$\begin{aligned} Y(s) &= \frac{c\,x(0)}{s+a} + \frac{c\,b}{(s+a)}\frac{A\,s}{(s^2+\omega^2)} \\ &= \frac{c\,x(0)}{s+a} + \frac{c\,b\,a\,A}{(\omega^2+a^2)}\frac{1}{(s+a)} + \frac{c\,b\,A}{(\omega^2+a^2)}\frac{-as+\omega^2}{(s^2+\omega^2)}. \end{aligned}$$

The time domain equivalent is

$$\begin{aligned} y(t) &= \left(c\,x(0) + \frac{c\,a\,b\,A}{w^2+a^2}\right)e^{-a\,t} + \\ &\quad \frac{c\,b\,A}{\sqrt{\omega^2+a^2}}\cos\left(\omega t + \operatorname{atan2}\left(-\omega, a\right)\right). \end{aligned}$$

For a stable system ($a < 0$), the exponential term in the first line decays toward zero as time increases. In steady state, the response is

$$y(t) \approx \frac{c\,b}{\sqrt{\omega^2+a^2}}A\cos\left(\omega t + \operatorname{atan2}\left(-\omega, a\right)\right). \tag{11}$$

Note that the steady state output is also a sinusoid with amplitude amplified by the gain $\left(\frac{cb}{\sqrt{\omega^2+a^2}}\right)$ and phase shifted by $\operatorname{atan2}\left(-\omega, a\right)$ relative to the applied input. In the following, we show that this gain and phase shift can be directly calculate from the transfer function. This result also generalizes to higher order systems. This method called *frequency response* is a very useful engineering tool.

The transfer function $H(s)$ is a complex function of the complex variable s. The *frequency response* corresponds to $H(s)$ evaluated for $s = j\omega$ where $j = \sqrt{-1}$ and $\omega \in [0, \infty)$. For our first-order system, the frequency response is

$$\begin{aligned} H(j\omega) &= \frac{cb}{j\omega+a} = \frac{cb}{(j\omega+a)}\frac{(-j\omega+a)}{(-j\omega+a)} \\ &= \frac{cb(-j\omega+a)}{a^2+\omega^2} \\ &= \frac{cb}{\sqrt{a^2+\omega^2}}\exp\left(j\tan^{-1}\left(\frac{-\omega}{a}\right)\right) \\ &= M(w)e^{j\Phi(\omega)} \end{aligned}$$

where

$$M(w) = \frac{cb}{\sqrt{a^2+\omega^2}} \quad \text{and} \quad \Phi(\omega) = \tan^{-1}\left(\frac{-\omega}{a}\right).$$

Comparing these expressions with the steady state time response in eqn. (11) we see that as $t \to \infty$

$$y(t) \approx M(w)\,A\cos\left(\omega\,t + \Phi(\omega)\right).$$

The forced response has the same sinusoidal form as the input, with magnitude changed by the factor $M(\omega)$ and phase shifted by $\Phi(\omega)$.

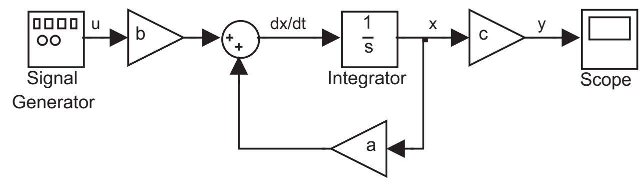

Figure 1: Simulink implementation of eqns. (2–3).

1.2.6 First-order Simulink Block Diagram

One (of many possible) Simulink implementation of eqns. (2–3) is shown in Figure 1. Each triangle with a symbol inside implements a multiplication of its input by the value of the symbol to compute its output. The rectanglular block that displays $\frac{1}{s}$ is an integrator that computes its output as the time integral of its input plus an initial value that the user can specify by double-clicking. The other blocks are self-explanatory.

2 Prelab

1. Read the above text, which provides information to do the following.
2. Figure 2 plots the zero-input (i.e., $u(t) = 0$) time-response of various different systems. For each:
   (a) State whether of not the response could be from a first-order linear system.
   (b) If not, state why not.
   (c) If so, state the sign and estimate the value of the time constant and the parameter a.
3. For the circuit shown in Figure 3:
   (a) Find the transfer function from u to y.
   (b) Using the capacitor voltage as the state variable, find the state space representation in the form of eqns. (2–3).

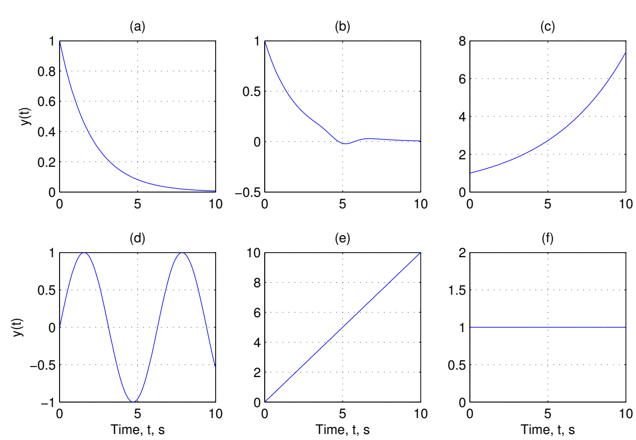

Figure 2: Figures for the Prelab Part 1.

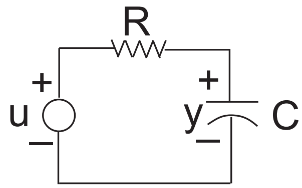

Figure 3: RC Circuit for Prelab Part 2 and Lab Part 2.

(c) Using the values $R = 20 \times 10^4$ ohms and $C = 0.5 \times 10^{-6}$ Farads, analytically find an expression for the magnitude and phase of the transfer function at $\omega = 0.00, 0.01, 0.10, 1.00, 10.00$, and 100.00 rad/sec.

(d) Use the Matlab 'Bode' function to plot the magnitude and phase versus ω. Make sure that these plots match your answers from Part 3c above.

3 Lab

This lab has two parts. In Part 1, you will be asked to compute the time constant applicable to a data set corresponding to a first-order system. In Part 2, you will simulate a first-order system to verify the DC gain and frequency response formulas.

Note that Parts 1 and 2 deal with two different systems. The time constant in Part 1 is not known and must be computed from the data. In Part 2, you know the value of the time constant from the Prelab.

Show your Prelab to TA to record that it is done.

3.1 Part 1 – Time Constant Estimation

1. Download from ILearn the data set listed for this lab. Use Matlab to plot y versus t. Make sure that the figure is properly labeled as indicated in Lab 1.
2. Section 1.2.5 describes two methods for estimation of time constants from data. Use each method to estimate the time constant for the data that you just plotted.[1]

Use the 'grid on' plot option. Note that you can zoom in on portions of the graph using the magnifying glass in the plot window menu. Ensure that your report clearly describes the method, computations, and data values that you use.

Show your results for Part 1 to TA.

[1] For the second method (tangent line) you can either draw the tangent manually or use the first two points to determine the slope of the tangent line and draw it with Matlab.

3.2 Part 2 – Simulation with Simulink

A reference guide for Simulink is attached at the end of this document.

1. Simulation setup and zero input response.
   (a) Implement a Simulink "all integrator" block diagram (similar to that shown in Figure 1) for the RC circuit of Figure 3. Enter the values of a, b, and c by double clicking the gain icons.
   (b) Make sure to have a 'scope' connected to both the signals u and y so that you can analyze their response. A good approach is to use the 'mux' block so that you can plot both u and y using a single scope.
   (c) In the Simulink window, under the 'Modeling/Model settings' , edit the 'Solver Parameters' to change the 'Max. step size' to about $0.01\,\tau$ and set the duration of the simulation to be about 10τ, where tau is the time constant. The first change ensures that you have at least 100 time steps per time constant. The second change ensures that you will see both the transient and steady state response.
   (d) Double click on the integrator icon and set the initial condition of the integrator to 1.0. Ensure that your report states the units of this initial condition.
   (e) Simulate the system with the $u = 0$. Verify that the convergence rate of the zero input response matches that expected for the time constant. If it does not, then you need to check your derivations and Simulink implementation.

Show your results for Part 2, Step 1 to TA.

2. Forced Response: Step input.
   (a) Set the initial condition of the integrator to 0.0.
   (b) Open 'Simulink/Simulation/Library Browser'.
   (c) From the 'Simulink/Sources' library folder, select the 'Pulse Generator' as the input u. Once you have it connected, double click on it to open its user interface. Set the amplitude to 1 and the period to 20 time constants (longer than the simulation). It should change from 0 to 1 exactly one time. Close that icon's user interface. Leave the simulation duration at 10 τ.
   (d) Simulate the system.
   (e) How long does the response take to approach steady state? How is this compared with the time constant?
   (f) What is the steady state value of y? Does this match the value predicted by the DC Gain analysis?
   (g) Remove the pulse generator from the block diagram.

Show your results for Part 2, Step 2 to TA.

3. Forced Response: Sinusoidal input.
   (a) From the 'Simulink:Sources' library folder, select and connect the 'Signal Generator' as the input u. Once you have it connected, double click on it to open its user interface.
   (b) Set the amplitude to 1 and the wave form to sine.
   (c) Set the simulation duration to 100s.
   (d) Simulate the system for each of the radian frequencies state in Prelab Part 3c.
   (e) For each frequency, compare the magnitude and phase between y and u. They should match the frequency response predictions.

_____Show your results for Part 2, Step 3 to TA. _____

EE 105: Simulink Guide

Jay Farrell, College of Engineering, University of California, Riverside

1 Matlab Simulink Guide

This section explains the basic methods for constructing Simulink signal flow models. Within Matlab, sinulink is opened using an icon in the middle of the top toolbar; alternatively, type ‘simulink’ in the command window.

1.1 First-order Example

The top (red) portion of Fig. 1 illustrates the idea that the block labeled ‘$\frac{1}{s}$’ is an integrator. Its output is the integral of its input. The input is labeled $\frac{d}{dt}x_1(t)$; therefore, the output is $x_1(t)$.

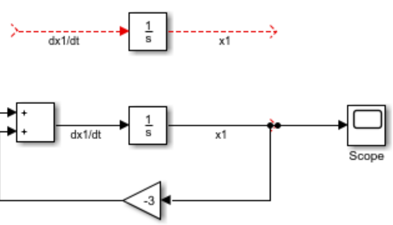

Figure 1: Basic Simulink signal flow diagram.

The lower portion of Fig. 2 illustrates the implementation of the signal flow model for the ODE:

$$\dot{x}_1(t) = -3x_1(t) + u(t).$$

This is a working simulink model. The input can be changed by double-clicking the signal generator. The initial conditions can be changed by double-clicking the integrator.

1.2 Third-order Example

Fig. 2 implements the third-order state-space model

$$\dot{x} = \begin{bmatrix} -5.0 & 0.0 & 0.2 \\ 4.0 & 0.0 & -2.0 \\ 0.0 & 5.0 & -0.2 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}. \tag{1}$$

Look at the diagram and check that each row of the matrix equation is correctly implemented.

From Fig. 2, it should become clear that as the order of the system grows, implementing each state separately will result in a messy tangle of signal flows.

Fig. 3 shows the same system implemented using the simulink State-Space block. The names of the matrices for the model parameters A, B, C, and D are communicated to this block by double-clicking on it. See Fig. 4.

The values of these parameter matrices must be defined in the Matlab workspace before the program can be run. The

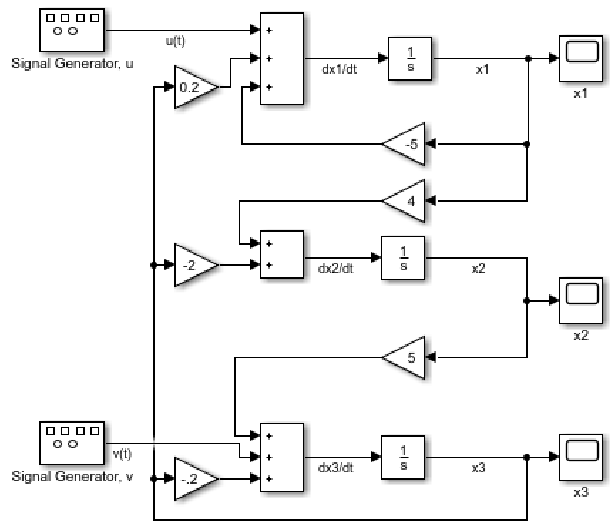

Figure 2: Third-order Simulink signal flow diagram.

matlab code to define these matrices for eqn. (1) with the output defined as the full state vector (i.e., $y(t) = x(t)$) is:

```
A = [-5.0  0.0   0.2
4.0   0.0  -2.0
0.0   5.0  -0.2];
B = [1  0
0  0
0  1];
C = eye(3);
D = zeros(3,2).
```

These can be programmed into an m-file that is run once in the command window before running the Simulink program.

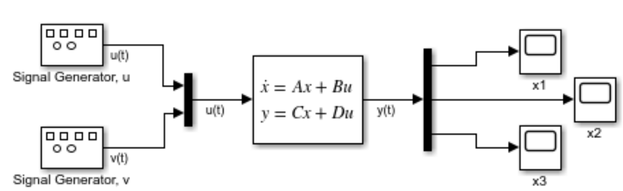

Figure 3: Simulink signal flow diagram of Fig. 2 implemented using the state-space block.

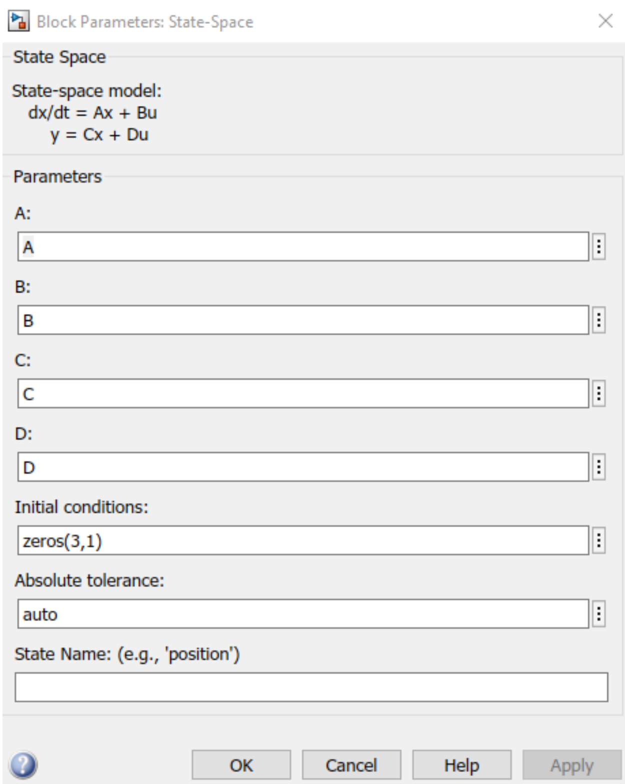

Figure 4: User-interface for the state-space block.

Fig. 5 shows the time response of each state when $u(t)$ is a unit amplitude square wave with frequency 0.5 Hz and $v(t)$ is is a square wave with amplitude 10 and frequency 0.2 Hz.

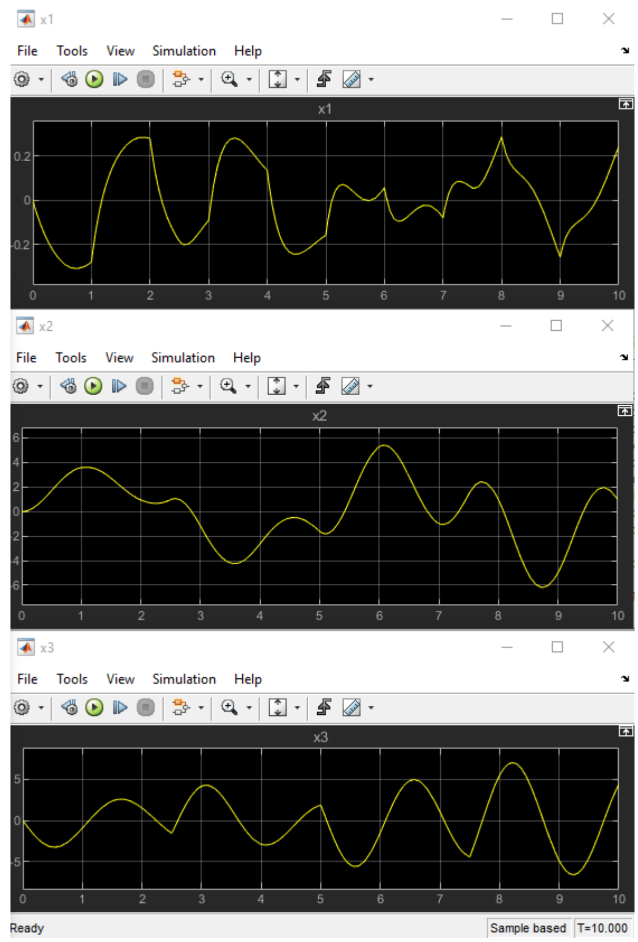

Figure 5: State trajectory corresponding to Fig. 2