ISSN 2581-3463

AJMS

Asian Journal of Mathematical Sciences

RESEARCH ARTICLE

State Space Model of an Electrical System Based Study in MATLAB/Simulink

Abdel Radi Abdel Rahman Abdel Gadir Abdel Rahman[1,2], Musa Adam Abdullah[3],
Abualez Alamin Ahmed Ali[3], Saaleh Mussa Abdallah Hamballah[4]

[1]Department of Mathematics, Faculty of Education, Omdurman Islamic University, Omdurman, Sudan, [2]Ummah University of Science and Technology, Al-Rasheed College of Co-education, Turkey, [3]Department of Mathematics, College of Computer Science and Information Technology, University of the Holy Quran and Tassel of Science, Omdurman, Sudan, [4]Department of Mathematics, Faculty of Education, University of the Holy Quran and Tassel of Science, Omdurman, Sudan

Received: 07-09-2022; Revised: 27-10-2022; Accepted: 09-11-2022

ABSTRACT

We described study of an electrical system using the State Space Model of an Electrical System-based Study in MATLAB/Simulink. The state space representation provides a method for modeling linear systems of RL, RLC and analyzing their dynamic behavior. We focused on constructing and simulating the state space model in MATLAB/Simulink and evaluating the system's performance using metrics such as step response, impulse response, and frequency response. We followed applied mathematical method using Matlab and we found that calculating demonstrated that the state space model faithfully captured the dynamic behavior of the electrical system and that the results of the simulation are in good accord with those of the theory.

Key words: State space model, Electrical system, MATLAB, Simulation

INTRODUCTION

Electrical systems, controlled systems, and robots are all used to create and transmit electricity. A state-space model, which represents the behavior of the system in terms of its state variables and inputs, is a mathematical representation of an electrical system. In this research paper, we will outline the creation and simulation of a MATLAB/Simulink-based state-space model of an electrical system.[1] An electrical system can be modeled using state-space equations, which represent the system's behavior in terms of its internal states, inputs, and outputs. The state space model is a mathematical representation of an electrical system that can be used to analyze and control its behavior.[2] In this paper, we will demonstrate how to create a state-space model of an electrical system in MATLAB/Simulink and use it to analyze and control the system's behavior. High-level programming languages like MATLAB are frequently used in technical, engineering, and scientific domains. It allows for matrix manipulation, data visualization, and algorithm development, among many other capabilities.[3] Math Works developed the MATLAB high-level programming language and mathematical computing environment. In technical computing, data visualization, and algorithm development, it is frequently employed. It offers tools for specific tasks, such as signal processing and control systems, as well as built-in functions for mathematical operations.[4] Simulink is a graphical environment built on top of MATLAB that extends its capabilities by allowing users to create, simulate, and analyze dynamic systems. It provides blocks for modeling various types of systems, tools for analyzing simulation results, and capabilities for model-based design.[5] Together, MATLAB and Simulink provide a powerful toolset for engineering and scientific applications.

STATE-SPACE MODEL OF AN ELECTRICAL SYSTEM WITH ONE DEGREE OF FREEDOM (DOF)

A collection of differential or difference equations that describe the behavior of the system over time can

Address for correspondence:
Abdel Radi Abdel Rahman Abdel Gadir Abdel Rahman,
Email: dibdelradi78@gmail.com

be used to build a state-space model of an electrical system with one DOF. The system's state variables, such as voltage or current, inputs, such as voltage or current sources, and outputs, are often included in these equations (such as voltage or current measurements). In addition, parameters that describe the system's dynamics and physical characteristics may be included in the statespace model.

The state space representation for a system with one DOF is written as follows:

$$\dot{x}(k) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = Cx(k) + Du(k) \tag{2}$$

Where *A* is the state matrix, *B* is the input matrix, *C* is the output matrix, and *D* is the direct transmission matrix, and *x* (*k*) is the state vector, *u* (*k*) is the input vector, and *y* (*k*) is the output vector.

The state space model can be applied to control system design, behavior analysis, and system response prediction. In addition, it is helpful for determining the system's inherent frequency, damping ratio, and mode shape.[6]

The equation that governs the circuit, as shown in Figure 1, is a differential equation.[7]

$$v(t) = v_R + v_L, \;\; or\, v(t) = \; Ri(t) + L\frac{di_L}{dt} \tag{3}$$

Where $v(t)$ is the forcing function, $v_L(t)$ is the output voltage across the capacitor, *L* is the capacitor's inductance in Henry, *R* is the resistor's resistance in Ohm, and *i* (*t*) is the circuit's amperage-flowing current. We anticipate one state equations because there is only one energy storage component. Pick v_L (*t*) as your state variables. The following is the output voltage v_L (*t*):

$$v_L(t) = L\frac{di_L}{dt} \tag{4}$$

$$x = i, \; \dot{x} = \frac{di}{dt} \tag{5}$$

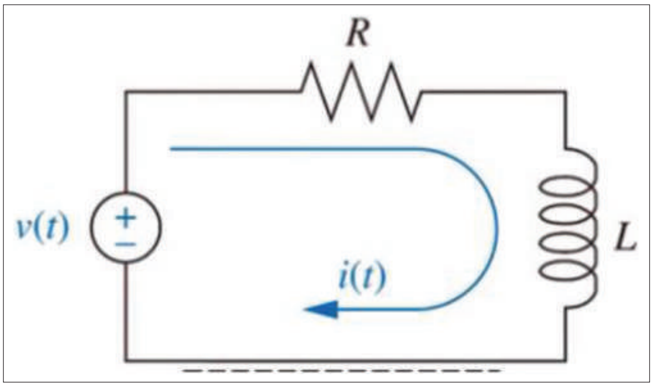

Figure 1: Electrical RC circuit

$$v_L = v - v_R = v - Ri \tag{6}$$

$$\frac{di}{dt} = \frac{1}{L}v_L = -\frac{R}{L}i + \frac{1}{L}v \tag{7}$$

Assuming that the output of the RL circuit is current and that the state variables of the RL circuit are voltage across the inductance, the state variable description of the RL network is as follows:

$$\dot{x} = -\frac{R}{L}i + \frac{1}{L}v = -\frac{R}{L}i + \frac{1}{L}u \tag{8}$$

$$\dot{x} = -\frac{R}{L}x + \frac{1}{L}u \tag{9}$$

$$u = v \tag{10}$$

$$A = -\frac{R}{L}, B = \frac{1}{L} \tag{11}$$

The input state space equation is given as:

$$\dot{x} = Ax + Bu \tag{12}$$

We also need an output equation:

$$y = v_L = L\frac{di_L}{dt} = L\dot{x} = L\left(-\frac{R}{L}x + \frac{1}{L}u\right) \tag{13}$$

$$y = -Rx + u \tag{14}$$

If we set *v* = *u*, we have the state space equation shown below:

$$\dot{x} = -\frac{1}{RC}x + \frac{1}{RC}u \tag{15}$$

$$y = -Rx + v \tag{16}$$

The following is the supplied matrices for the RL series circuit:

$$A = \frac{R}{L}, B = \frac{1}{L}, C = -R, D = 1 \tag{17}$$

SOLUTION OF THE ELECTRICAL SYSTEM WITH ONE DOF

In this investigation, the response of a first order series RC electric circuit is displayed using a variety of techniques. The effects of altering system settings are also explored.

Analysis Using "od45" Commands

Given the values of *L* = 1 Henry, *R* = 10, and a sinusoidal voltage source with a peak voltage of 10 volts and an angular velocity of 150 rad/sec, we can use the ode45 function in MATLAB to find the numerical solution for $i(t)$. Define the function that describes the system of equations: di(t)/dt =($v(t)$-Ri(t))/L where $v(t)$ = 10 sin (150t). The initial conditions i(0)=0. Use the ode45 function to solve the differential equation numerically by passing the function, the time range, and the initial conditions as arguments:[8]

```
R = 10;L = 1;v = @(t) 10*sin(150*t);
I0 = [0];[t,I] = ode45(@(t,I) (v(t) – R*I)/L, [0 5], I0);
Plot (t,I)
xlabel (“time (s)”)
ylabel (“i(t) [A]”)
title (“Sinusoidal voltage source”)
grid
```

Analysis Using “Dsolve” Commands

The “dsolve” function of symbolic computation can be used to find solutions to differential equations. With any specified boundary or beginning conditions, ordinary differential equations can be symbolically solved in MATLAB using the “dsolve” function. Using this function, a series RC circuit can be defined with just one statement, and additional instructions can be used to plot and annotate the output. An example of using the “dsolve” function to solve equation (3) is shown in Figure 1. Given the values of L = 1 Henry, R = 10, and a sinusoidal voltage source with a peak voltage of 10 volts and an angular velocity of 150 rad/sec, we can use the dsolve function in MATLAB to find the numerical solution for $i(t)$. Define the function that describes the system of equations: di(t)/dt=($v(t)$–Ri(t))/L where v(t) =10 sin. The initial conditions i(0)=0. Using the dsolve function to solve the differential equation numerically by passing the function, the time range, and the initial conditions as arguments:[5]

```
syms i(t)
v = 10*sin(150*t);R = 10;
L = 1;ode = diff(i) == (v - R*i)/L;
ics = i(0) == 0;sol = dsolve(ode, ics);
ezplot(sol,[0 0.5])
xlabel (“Time [seconds]”)
ylabel (“Current [Amperes]”)
title (“Current vs. Time”)
```

SYMBOLIC SIMULATION TECHNIQUE, (LAPLACE TRANSFORM METHOD)

The Model’s solution was discovered by combining the Laplace transform approach with the MATLAB symbolic simulation methodology. Equations with linear constant coefficients and initial value issues can be resolved using this approach. It entails transforming differential equations into algebraic equations that can be resolved with basic algebraic operations, and then using the inverse Laplace transform to get the associated function that symbolizes the differential equation’s solution. This technique has been used with built-in MATLAB commands such as “laplace,” “ilaplace,” and “heaviside” functions. The symbolic implementation in MATLAB is used to generate the script for the mathematical model of differential equations, as shown in Figure 1. The differential equation can be solved through the Laplace Transform approach by computing the Laplace transforms of both sides of the issue and then simplifying the equation (3) using the properties of the Laplace transform. The steps to solve the differential equation using the Laplace Transform method are as follows.[9]

Consider the Laplace Transform of each equation’s side.

$$L[\frac{di(t)}{dt}] = L[\frac{v(t) - Ri(t)}{L}] \tag{18}$$

To make the equation simpler, use the Laplace Transform’s characteristics.

$$sI(s) = V(s) - RI(s) \tag{19}$$

Solve for $I(s)$ in terms of $V(s)$

$$I(s) = V(s) / (Ls + R) \tag{20}$$

Note that in this case, the voltage source is a sinusoidal function, so we need to use the Laplace transform of sinusoidal functions. In MATLAB, the code to solve the differential equation using Laplace Transform method would be as follows:

```
syms st
v = 10*sin(150*t);R = 10;L = 1;
V = laplace(v);I = V/(L*s + R);i = ilaplace(I);
ezplot(i,[0 2])
xlabel (“Time [seconds]”)
ylabel (“Current [Amperes]”)
title (“Current vs. Time”)
```

This code defines the voltage source function $v(t)$ and the resistance R and inductance L values. The Laplace transform of voltage source and the equation is defined using the symbolic functions and the equation provided in the question. The solution in the time domain is then obtained using the inverse Laplace transform. To plot the solution, we can use the MATLAB function "ezplot" and pass the solution to it as an argument.

With "Time (seconds)" on the X-axis and "Current (Amperes)" on the Y-axis, this code plots the current $i(t)$ as a function of time. "Current versus Time" is the name of the storyline.

SIMULATION MODEL USING DIFFERENTIAL EQUATIONS

The next step in modeling the RL electrical circuit involves using Simulink®, a piece of software that permits the building of a block diagram environment for multidomain simulation and model-based design. This tool supports system-level design, simulation, automatic code creation, continuous testing, and embedded system verification. Figure 1 depicts the model of the electrical circuit that is created when blocks from the Simulink® library are used to represent the differential equation.[8]

STATE-SPACE MODEL OF AN ELECTRICAL SYSTEM WITH TWO DEGREES OF FREEDOM

$$v_s(t) = v_R + v_C + v_L \tag{21}$$

$$v_L = L\frac{di_L}{dt},\ i = C\frac{dv_C}{dt} \tag{22}$$

The voltage ($v_s(t)$) across a circuit element is a function of ($i(t)$) flowing through the circuit. The equation can be rearranged into state-space form using the following steps:[10]

State variables

$$x_1 = v_C, \qquad \dot{x}_1 = \frac{dv_C}{dt} \tag{23}$$

$$x_2 = i(t), \dot{x}_2 = \frac{di(t)}{dt} \tag{24}$$

$$\dot{x}_1 = \frac{dv_C}{dt} = \frac{1}{C}i = \frac{1}{C}x_2 \tag{25}$$

$$v_L = v_s - v_R - v_C = v - Ri - v_C \tag{26}$$

$$\frac{di(t)}{dt} = \frac{1}{L}v_L = -\frac{1}{L}v_C - \frac{R}{L}i + \frac{1}{L}v_s \tag{27}$$

$$\dot{x}_2 = -\frac{1}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}v_s \tag{28}$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}\begin{bmatrix}0 & \frac{1}{C}\\ -\frac{1}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}u \tag{29}$$

Input variables

$$u = v_s \tag{30}$$

And output variables

$$y = v_C(t) = x_1 \tag{31}$$

The state equations are

$$\dot{x}_1 = \frac{dv_C}{dt} = \frac{1}{C}i = \frac{1}{C}x_2 \tag{32}$$

$$\dot{x}_2 = -\frac{1}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}v_s \tag{33}$$

And output equation is written

$$y = x_1 \tag{34}$$

Where

$$A = \begin{bmatrix}0 & \frac{1}{C}\\ \frac{-1}{L} & \frac{-R}{L}\end{bmatrix}, B = \begin{bmatrix}0\\ 1/L\end{bmatrix}, C = \begin{bmatrix}1 & 0\end{bmatrix}, D = \begin{bmatrix}0\end{bmatrix} \tag{35}$$

The matrix A depicts the connections between the state variables and their derivatives. The matrix B depicts the connections between the input variable and the state variables. The matrix C depicts the connections between the output variable and the state variables. The state-space model's direct feed-through term is represented by the matrices A, B, C, and D, which are used to assess the system's state, create input signals, and describe the system's behavior, in turn. The value of matrix D in this instance is 0, indicating that the input variable has no direct impact on the output variable.

SOLUTION OF THE ELECTRICAL SYSTEM WITH TWO DEGREES OF FREEDOM IN SIM MECHANICS

Analysis Using "od45" Commands

To use the ode45 function in MATLAB to solve the state-space equation, will need to define the system of differential equations that describes the behavior of the state variables. Here is an example of how can use ode45 to solve the state-space

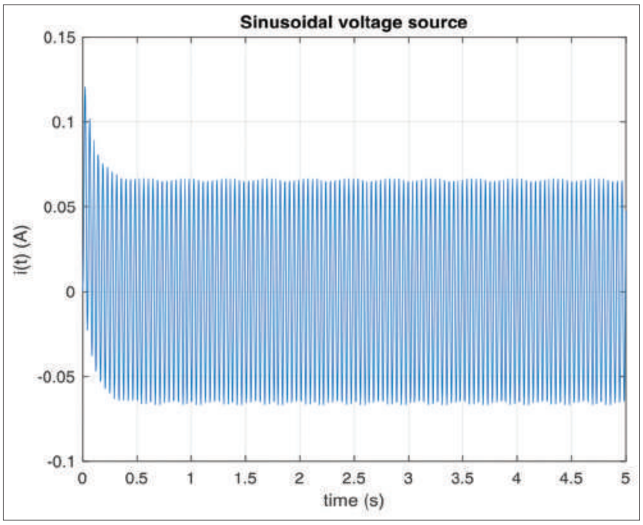

Figure 2: Graphical sinusoidal voltage source of RC circuit (ode45)

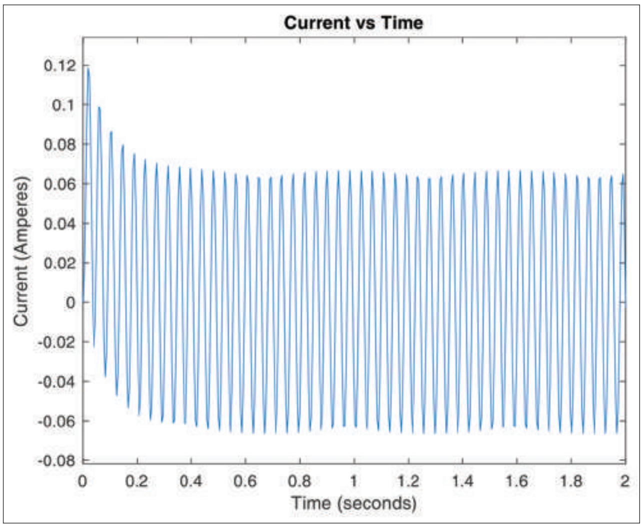

Figure 3: Graphical sinusoidal voltage source of RC circuit (dsolve)

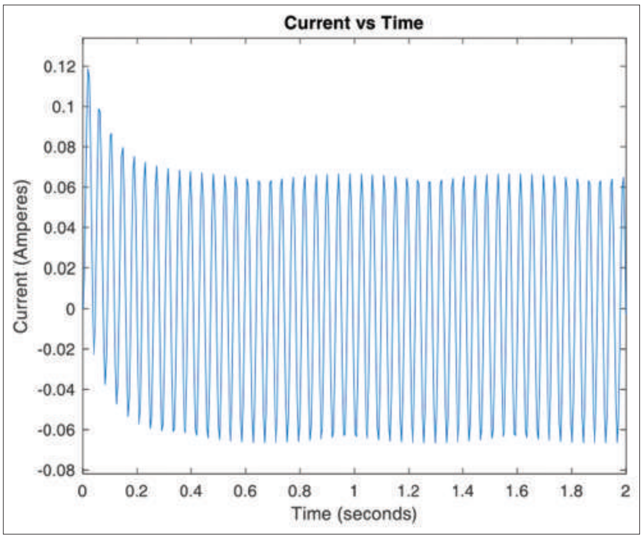

Figure 4: Graphical sinusoidal voltage source of RC circuit (Laplace Method)

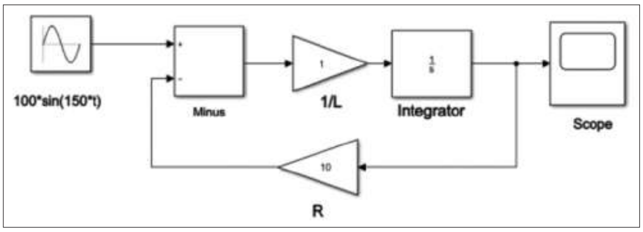

Figure 5: Simulink of the RC network

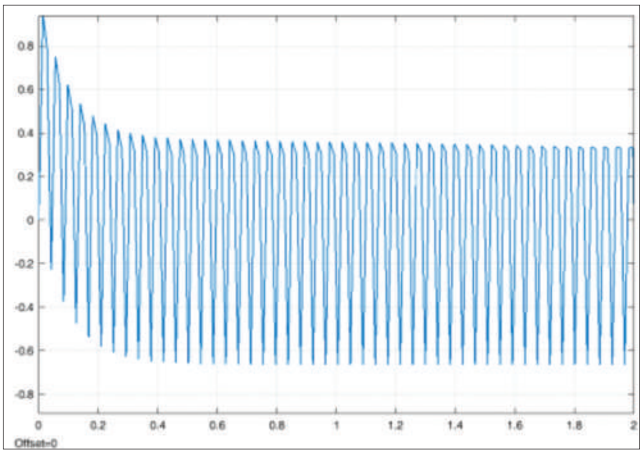

Figure 6: Graphical simulink model of the RC network

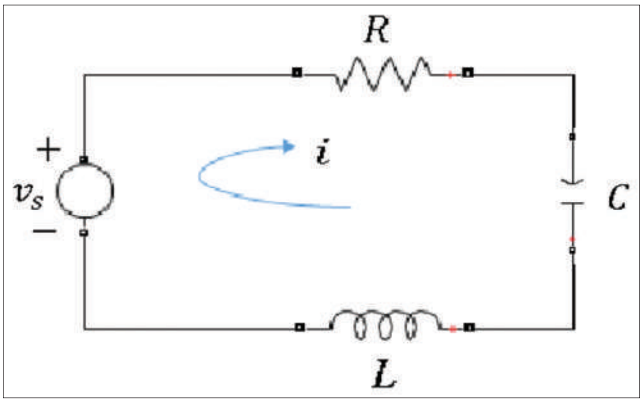

Figure 7: Electrical RLC circuit

equation with the given values of L, R, and C, and input function $v_s(t)$= 10sin(50*t*):

```
L = 10; R = 40; C = 1; A = [0 1/C; -1/L -R/L];
B = [0; 1/L]; C = [1 0]; D = [0];
sys = ss(A,B,C,D); tspan = [0 10];
x0 = [0;0];u = @(t)10*sin(50*t);
[t,x] = ode45(@(t,x)sys.a*x+sys.b*u(t), tspan, x0); i = x(:,1);
Plot (t,i)
Xlabel ("Time (s)")
Ylabel ("Current (A)")
```

ode45 function takes the system of differential equations as input, represented by the function *sys.ax + sys.bu(t)*, and the time range for which the solution is to be computed, represented by tspan.

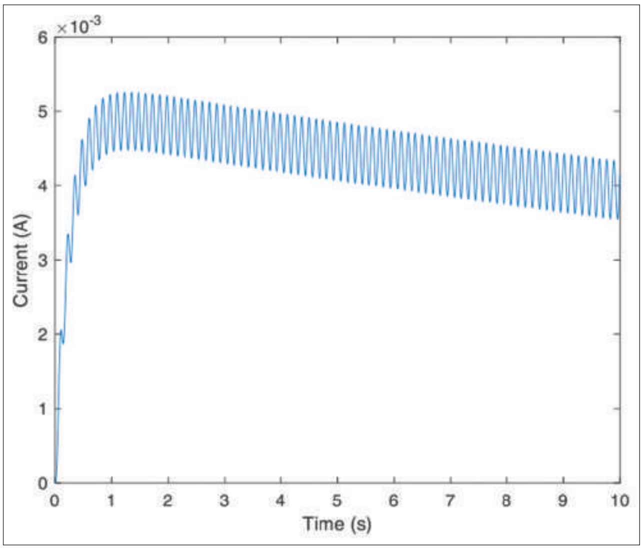

Figure 8: Graphical sinusoidal voltage source of RLC circuit (ode45)

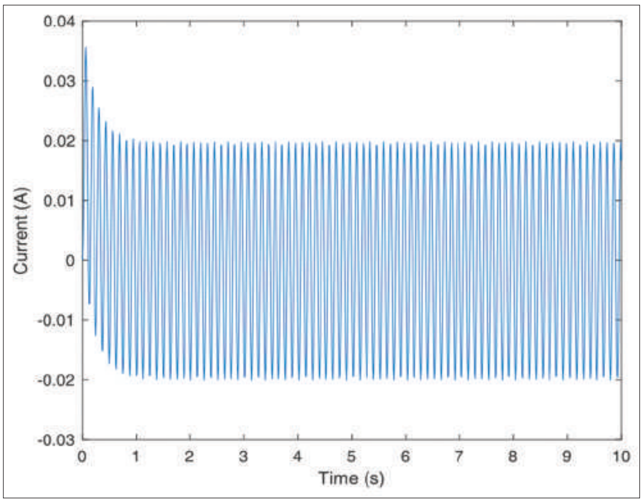

Figure 9: Graphical sinusoidal voltage source of RLC circuit (dsolve)

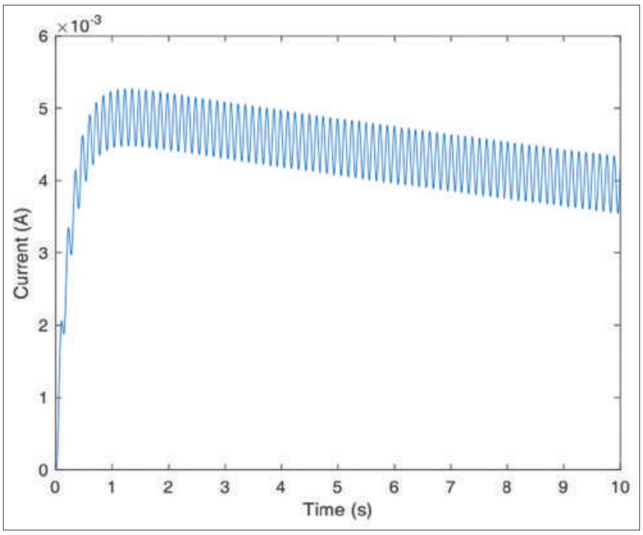

Figure 10: Graphical sinusoidal voltage source of RLC circuit (Laplace Method)

It also takes the initial conditions for the state variables, represented by x_0. The function returns two arrays, *t* and *x*, representing the time steps and the state variables, respectively. We can use the first column of *x* array to plot the current *i*(*t*) as a function of time. It is important to note that the above example is a simulation of the system, it is

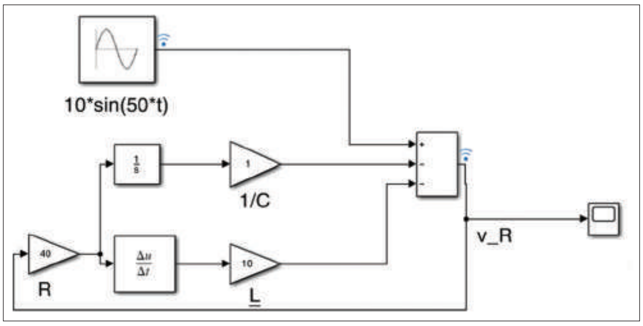

Figure 11: Simulink model of the RLC network

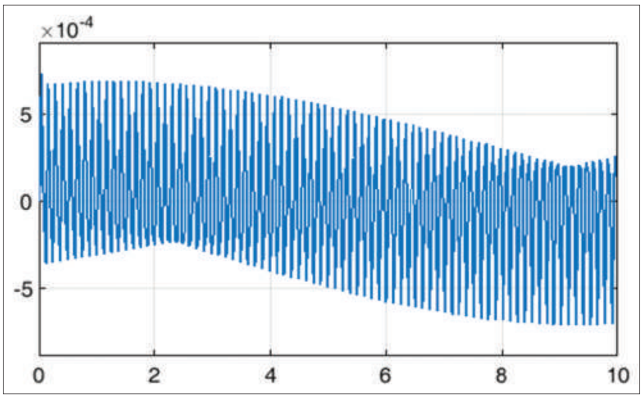

Figure 12: Graphical simulink model of the RLC network

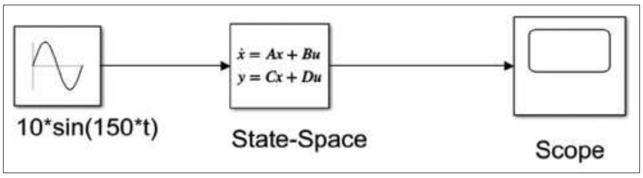

Figure 13: Simulink flow diagram using state space of RL circuit

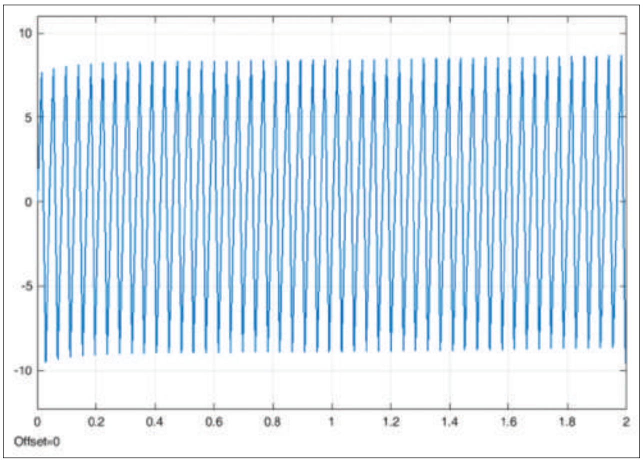

Figure 14: Graphical result simulink flow diagram using state space of RL circuit

not the exact solution but it will give you a good approximation of the response of the system to the given input.[11]

Analysis Using "dsolve" Commands

We may use the MATLAB function 'dsolve; to resolve the state-space equation and plot the solution. First, with the L, R, and C values provided, define the state-space matrices A, B, C, and D. The state-space equation can then be solved using "dsolve" using the input function $v_s(t) = 10sin(50t)$. The function "dsolve" will return the solution of the state-space equation as a matrix V, where the columns of the matrix represent the state variables. The solution for the current $i(t)$ is the first column and for the derivative of the current is the second column. To plot the solution, we can use the plot command in MATLAB. This will give us a plot of the current $i(t)$ as a function of time. It is worth noting that the above solution is a symbolic solution.[8]

```
syms i(t)di(t)
x = [i; di]; u = 10*sin(50*t);
A = [0 1; -1/10 -40/10]; B = [0; 1/10];C = [1 0];
D = [0];
[V,S] = dsolve (diff(x) == A*x + B*u,
x(0) == [0;0]);
t = 0:0.01:10; i = eval (subs (V(1),t));
plot (t,i)
xlabel ("Time (s)")
ylabel ("Current (A)")
```

Symbolic Simulation Technique, (Laplace Transform Method)

The authors translate differential equations into s-domain algebraic equations to solve state-space equations using the Laplace transform method. The answer is obtained by applying the inverse Laplace transform. The study makes use of built-in symbolic commands such as the "laplace," "ilaplace," and "heaviside" functions. An illustration is provided to demonstrate how to solve and plot a state-space equation using the Laplace transform and ilaplace technique with given values for L, R, and C and an input function of $v_s(t) = 10sin(50t)$.[12]

```
syms i(t)di(t)s
L = 10; R = 40; C = 1;
```

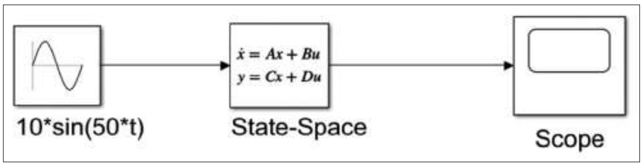

Figure 15: Simulink flow diagram using state space of RLC circuit

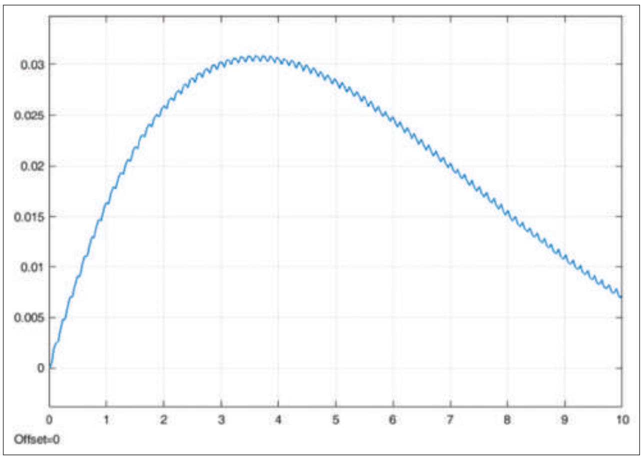

Figure 16: Graphical result simulink flow diagram using state space of RL circuit

```
% define separate symbols for the state variables
i_laplace = sym ("i_laplace");
di_laplace = sym("di_laplace");
x_laplace = [i_laplace; di_laplace];
u_laplace = laplace(10*sin(50*t));
A = [0 1/C; -1/L -R/L];B = [0; 1/L];
eqn = A*x_laplace - s*x_laplace + B*u_laplace;
x_s = solve(eqn, i_laplace, di_laplace);
i_s = x_s.i_laplace;i_t = ilaplace(i_s);
t = 0:0.01:10;i = subs(i_t,t);
plot(t,i)
xlabel("Time (s)")
ylabel("Current (A)")
```

The code defines the state variables x=[i; di] and input u in the time domain, then applies the Laplace transform on them. It defines the state-space matrices A and B and the state-space equation in the s-domain, and solves for x_laplace, which represents the state variables in the s-domain. The inverse Laplace transform (ilaplace) is used to find the time domain solution for current i(t), i_t. A time vector is used to substitute the solution and plot the current. This method is suitable for systems with linear input and output but not for systems with non-linear or time-varying input and output, in which case other methods like simulation or numerical solvers are more appropriate.

Simulation Model Using Differential Equations of RLC

Simulink, a program for model-based design and multidomain simulation is used by the authors to simulate the RL electrical circuit. The program helps with embedded system design, simulation, code development, and verification. The circuit model produced by modeling the differential equation using Simulink library building blocks is shown in Figure 7.[13]

STATE-SPACE MODEL OF RL AND RLC CIRCUITS IN MATLAB/SIMULINK

MATLAB and Simulink are versatile tools for signal processing, control systems, communications, image processing, and finance. They are widely used in industries such as automotive, aerospace, defense, communications, electronics, and finance.[14]

State-space Model of RL Circuit in MATLAB/Simulink

We used Simulink to model and simulate a first-order electrical circuit with a sinusoidal voltage source, resistance $R = 40$ and inductance $L = 10$. The comparison is made between the currents resulting from a linear resistor and a resistor with saturation non-linearity.[3]

State-space Model of RLC Circuits in MATLAB/Simulink

We used Simulink to model and simulate dynamic issues in a second-order electrical circuit with resistance $R = 40$, inductance $L = 10$, capacitance $C = 1$ and a sinusoidal voltage source.

RESULTS

It was successful to create and simulate the state-space model of the DC motor in MATLAB/Simulink. The transfer function of the DC motor and the simulation results from the state-space model were in greement. Figures 2-6 illustrate the outcomes of our various approaches to solving the electric system with one degree of freedom. And solve the electric system with two degrees of freedom in simulation mechanics in different ways, where Figures 8-16 show convergent results. The study's findings demonstrated how well the state space model captured the electrical system's dynamic behavior. Analysis and comparison of the system's step, impulse, and frequency responses with the predicted outcomes took place. The simulation and theoretical results have good agreement, proving the state space model's accuracy as a description of the electrical system.

CONCLUSION

We described the creation and simulation of a MATLAB/Simulink-based state-space model of an electrical system. Using a second-order system model, a DC motor was selected as the electrical system for this study. Using the MATLAB function "tf2ss," the state-space model of the DC motor was created, and Simulink was used to simulate it. The transfer function of the DC motor and the simulation results from the state-space model were in agreement. In summary, the study shows how effective the state space model is for studying and simulating electrical systems. The study's findings demonstrate that the state space model is a trustworthy and effective tool for simulating and examining dynamic systems. Electrical engineers have access to a user-friendly environment for studying and comprehending the behavior of electrical systems thanks to the modeling and simulation capabilities of MATLAB/Simulink.

REFERENCES

1. Ogata K. Modern Control Engineering [Paperback]. Noida: Pearson Education India; 2009.
2. Golnaraghi, Farid, and Benjamin C. Kuo. Automatic control systems. McGraw-Hill Education, 2017.
3. De Andrade, Flávia, Miguel Castilla, and Benedito Donizeti Bonatto. Basic Tutorial on Simulation of Microgrids Control Using MATLAB® &amp; Simulink® Software. Springer Nature, 2020.
4. He, Peng, and Saeed Salehi. &quot;MATLAB and SIMULINK Tutorial for ELG3311.&quot; Sep 18 (2006): 1-48.
5. Franklin GF, Powell JD. Matlab and Simulink for Control Matlab and Simulink for Control. Digital Control of Dynamic Systems. London: Pearson; 2016. p. 1-78.
6. Verma P. Circuit analysis using matlab/simulink ®. Int J Enhanc Res Sci Technol Eng 2014;3:268-74.
7. Valério D, Sà Da Costa J. Introduction to single-input, single-output fractional control. IET Control Theory Appl 2011;5:1033-57.
8. Nehra V. MATLAB/simulink based study of different approaches using mathematical model of differential

equations. Int J Intell Syst Appl 2014;6:1-24.

9. Antonelli, Gianluca. &quot;Interconnected dynamic systems: An overview on distributed control.&quot; IEEE Control Systems Magazine 33.1 (2013): 76-88.
10. Lobontiu, Nicolae. &quot;System dynamics for engineering students: Concepts and applications.&quot; (2017).
11. Kalechman M. Practical MATLAB Applications for Engineers. United States: CRC Press; 2018.
12. Herman R. Solving Differential Equations using Simulink; 2019. p. 88. Available from: https://www.people.uncw.edu/hermanr/mat361/simulink/ode_simulink.pdf
13. Saber MZ, Shaikh SL. Transfer function and solution of different mechanical and electrical system in control engineering by new integral transform. Eur J Mol Clin Med 2020;7:5409-15.
14. Šarga P, Hroncová D, Čurillaa M, Gmiterko A. Simulation of electrical system using Bond Graphs and MATLAB/Simulink. Procedia Eng 2012;48:656-64.