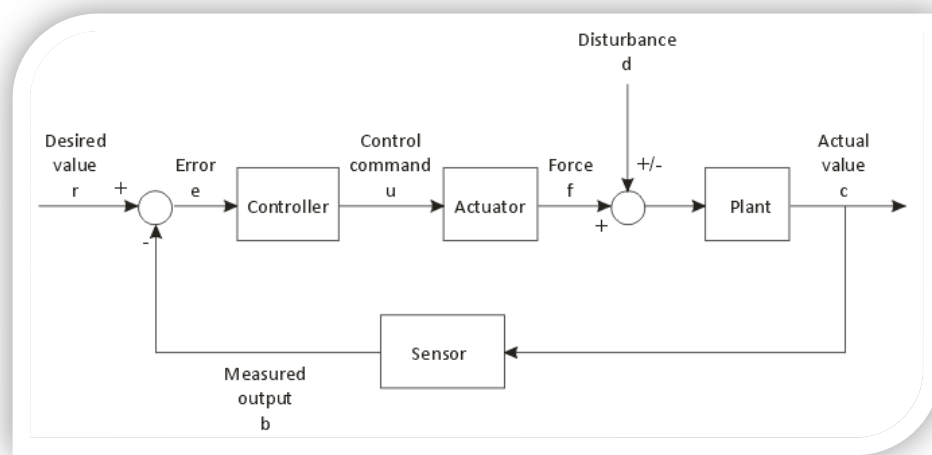


Experiment Three

Mathematical Modelling Using SimScape (Electrical Systems)

Control Systems Laboratory

Dr. Zaer Abo Hammour



1. Model and simulate MultiDomain Physical Systems

Simscape provides an environment for modeling and simulating physical systems spanning mechanical, electrical, hydraulic, and other physical domains. It provides fundamental building blocks from these domains that you can assemble into models of physical components, such as electric motors, inverting op-amps, hydraulic valves, and ratchet mechanisms. Because Simscape components use physical connections, your models match the structure of the system you are developing.

Simscape models can be used to develop control systems and test system-level performance. You can extend the libraries using the MATLAB[®] based Simscape language, which enables text-based authoring of physical modeling components, domains, and libraries. You can parameterize your models using MATLAB variables and expressions, and design control systems for your physical system in Simulink[®]. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape supports C-code generation.

2. Key Features

- Single environment for modeling and simulating mechanical, electrical, hydraulic, thermal, and other multidomain physical systems
- Libraries of physical modeling blocks and mathematical elements for developing custom components
- MATLAB based Simscape language, enabling text-based authoring of physical modeling components, domains, and libraries
- Physical units for parameters and variables, with all unit conversions handled automatically
- Ability to simulate models that include blocks from related physical modeling products without purchasing those products
- Support for C-code generation

3. Simscape Block Libraries

Simscape block library contains two libraries that belong to the Simscape[™] product:

- Foundation library — Contains basic hydraulic, pneumatic, mechanical, electrical, magnetic, thermal, thermal liquid, and physical signal blocks, organized into sublibraries according to technical discipline and function performed
- Utilities library — Contains essential environment blocks for creating Physical Networks models

In addition, if you have installed any of the add-on products of the Physical Modeling family, you will see the corresponding libraries under the main Simscape library.

Simscape Foundation libraries contain a comprehensive set of basic elements and building blocks, such as:

- Mechanical building blocks for representing one-dimensional translational and rotational motion
- Electrical building blocks for representing electrical components and circuits
- Magnetic building blocks that represent electromagnetic components

Experiment 3: Mathematical Modelling Using SimScape

- Hydraulic building blocks that model fundamental hydraulic effects and can be combined to create more complex hydraulic components
- Pneumatic building blocks that model fundamental pneumatic effects based on the ideal gas law
- Thermal building blocks that model fundamental thermal effects
- Thermal liquid building blocks that model fundamental thermodynamic effects in liquids
- Physical Signals block library that lets you perform math operations on physical signals, and graphically enter equations inside the physical network

Using the elements contained in these Foundation libraries, you can create more complex components that span different physical domains. You can then group this assembly of blocks into a subsystem and parameterize it to reuse and share these components.

In addition to Foundation libraries, there is also a Simscape Utilities library, which contains utility blocks, such as:

- Solver Configuration block, which contains parameters relevant to numerical algorithms for Simscape simulations. Each Simscape diagram (or each topologically distinct physical network in a diagram) must contain a Solver Configuration block.
- Simulink-PS Converter block and PS-Simulink Converter block, to connect Simscape and Simulink® blocks. Use the Simulink-PS Converter block to connect Simulink outputs to Physical Signal inports. Use the PS-Simulink Converter block to connect Physical Signal outputs to Simulink inports.

For examples of using these blocks in a Simscape model, see the tutorial *Creating and Simulating a Simple Model*.

You can combine all these blocks in your Simscape diagrams to model physical systems. You can also use the basic Simulink blocks in your diagrams, such as sources or scopes. See *Connecting Simscape Diagrams to Simulink Sources and Scopes* for more information on how to do this.

Simscape block libraries contain a comprehensive selection of blocks that represent engineering components such as valves, resistors, springs, and so on. These prebuilt blocks, however, may not be sufficient to address your particular engineering needs. When you need to extend the existing block libraries, use the Simscape language to define customized components, or even new physical domains, as textual files. Then convert your textual components into libraries of additional Simscape blocks that you can use in your model diagrams.

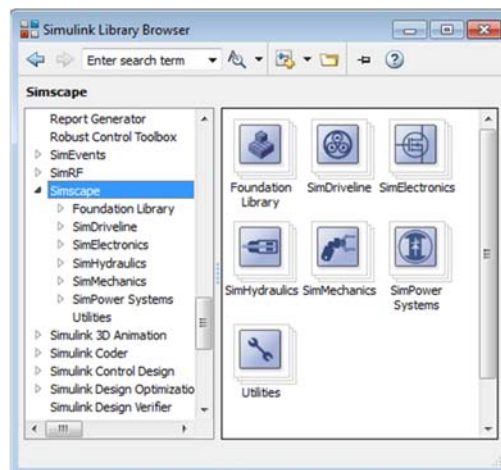


Figure 3.1: SimScape Library

Experiment 3: Mathematical Modelling Using SimScape

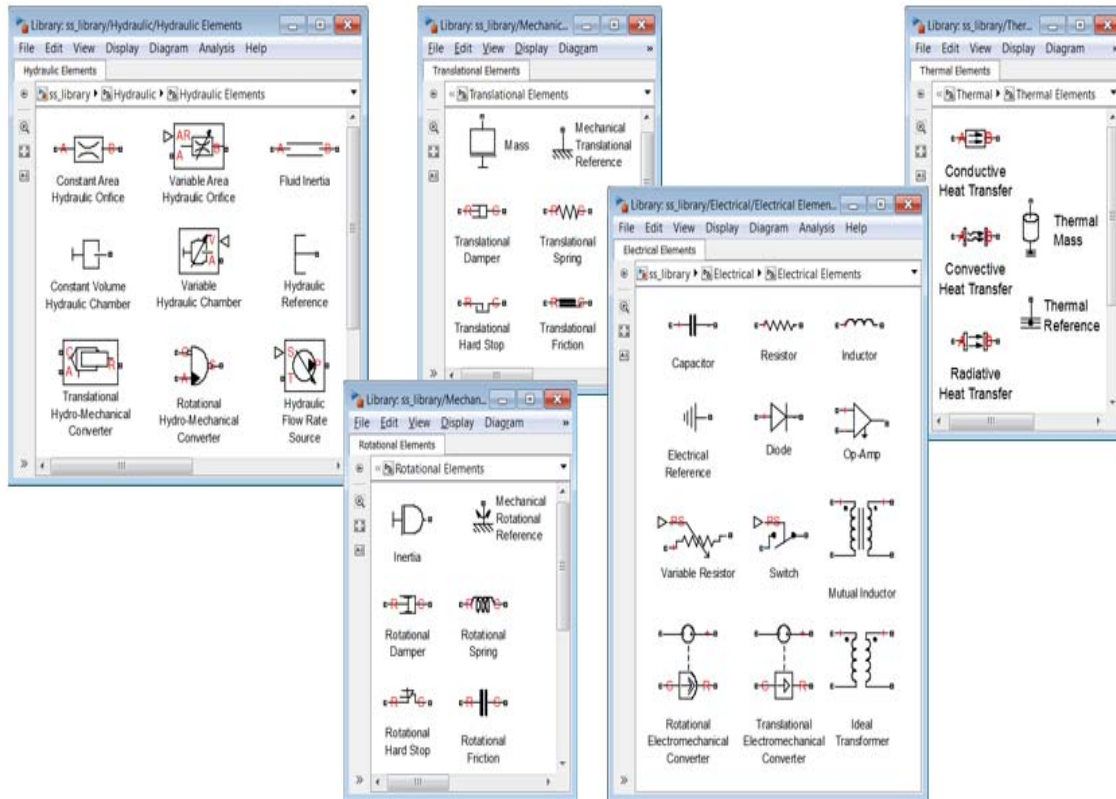


Figure 3.2: SimScape Library

4. How Simscape Simulation Works

Simscape software gives you multiple ways to simulate and analyze physical systems in the Simulink environment. Running a physical model simulation is similar to simulating any Simulink model. It entails setting various simulation options, starting the simulation, and viewing the simulation results. This topic describes various aspects of simulation specific to Simscape models. For specifics of simulating and analyzing with individual Simscape add-on products, refer to the documentation for those individual add-on products.

The flow chart in figure 3.3 presents the Simscape simulation sequence.

The flow chart consists of the following major phases:

1. Model Validation
2. Network Construction
3. Equation Construction
4. Initial Conditions Computation
5. Transient Initialization
6. Transient Solve

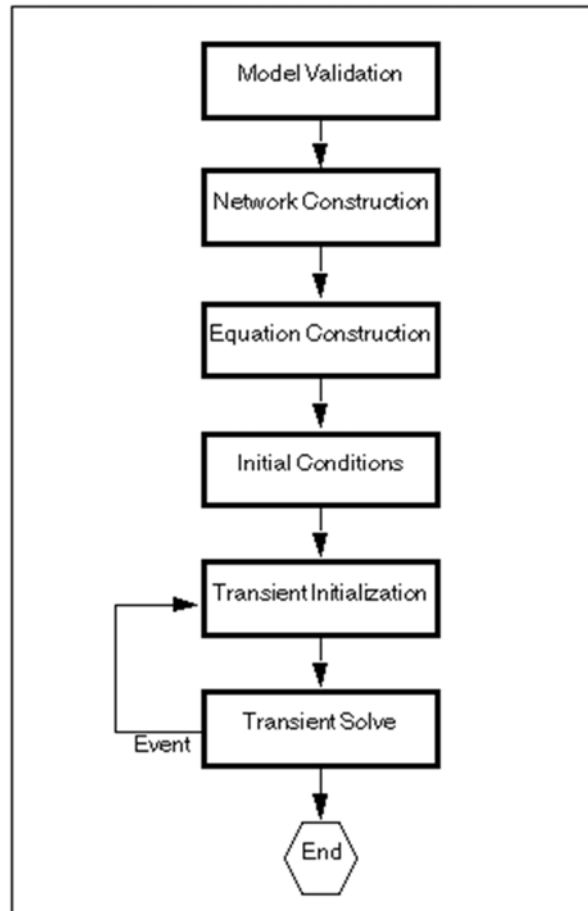


Figure 3.3: flow chart presents the Simscape

4.1 Model Validation

The Simscape solver first validates the model configuration and checks your data entries from the block dialog boxes.

All Simscape blocks in a diagram must be connected into one or more physical networks. Unconnected Conserving ports are not allowed.

- Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block.
- If your model contains hydraulic elements, each topologically distinct hydraulic circuit in a diagram must connect to a Custom Hydraulic Fluid block (or Hydraulic Fluid block, available with SimHydraulics® block libraries). These blocks define the fluid properties that act as global parameters for all the blocks that connect to the hydraulic circuit. If no hydraulic fluid block is attached to a loop, the hydraulic blocks in this loop use the default fluid. However, more than one hydraulic fluid block in a loop generates an error.
- Similarly, if your model contains pneumatic elements, default gas properties for a pneumatic network are for dry air and ambient conditions of 101325 Pa and 20 degrees Celsius. If you attach a Gas Properties block to a pneumatic circuit, you can change gas properties and ambient

conditions for all the blocks connected to the circuit. However, more than one Gas Properties block in a pneumatic circuit generates an error.

- Signal units specified in a Simulink-PS Converter block must match the input type expected by the Simscape block connected to it. For example, when you provide the input signal for an Ideal Angular Velocity Source block, specify angular velocity units, such as rad/s or rpm, in the Simulink-PS Converter block, or leave it unitless. Similarly, units specified in a PS-Simulink Converter block must match the type of physical signal provided by the Simscape block output.

4.2 Network Construction

After validating the model, the Simscape solver constructs the physical network based on the following principles:

- Two directly connected Conserving ports have the same values for all their Across variables (such as voltage or angular velocity).
- Any Through variable (such as current or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

4.3 Equation Construction

Based on the network configuration, the parameter values in the block dialog boxes, and the global parameters defined by the fluid properties, if applicable, the Simscape solver constructs the system of equations for the model.

These equations contain system variables of the following types:

- **Dynamic** : Time derivatives of these variables appear in equations. Dynamic, or differential, variables add dynamics to the system and require the solver to use numerical integration to compute their values. Dynamic variables can produce either independent or dependent states for simulation.
- **Algebraic** : Time derivatives of these variables do not appear in equations. These variables appear in algebraic equations but add no dynamics, and this typically occurs in physical systems due to conservation laws, such as conservation of mass and energy. The states of algebraic variables are always dependent on dynamic variables, other algebraic variables, or inputs.

The solver then performs the analysis and eliminates variables that are not needed to solve the system of equations. After variable elimination, the remaining variables (algebraic, dynamic dependent, and dynamic independent) get mapped to Simulink state vector of the model.

4.4 Initial Conditions Computation

The Simscape solver computes the initial conditions only once, at the beginning of simulation

($t = 0$). In the Solver Configuration block dialog box, the default is that the Start simulation from steady state check box is not selected. If it is selected in your model, see Finding an Initial Steady State.

The solver computes the initial conditions by finding initial values for all the system variables that exactly satisfy all the model equations. You can affect the initial conditions computation by block-level variable initialization, that is, by specifying the priority and target initial values on the Variables tab of the block dialog boxes.

The values you specify during block-level variable initialization are not the actual values of the respective variables, but rather their target values at the beginning of simulation ($t = 0$). Depending on the results of the solve, some of these targets may or may not be satisfied. The solver tries to satisfy the high-priority targets first, then the low-priority ones:

- At first, the solver tries to find a solution where all the high-priority variable targets are met exactly, and the low-priority targets are approximated as closely as possible. If the solution is found during this stage, it satisfies all the high-priority targets. Some of the low-priority targets might also be met exactly, the others are approximated.
- If the solver cannot find a solution that exactly satisfies all the high-priority targets, it issues a warning and enters the second stage, where High priority is relaxed to Low. That is, the solver tries to find a solution by approximating both the high-priority and the low-priority targets as closely as possible.

After you initialize the block variables and prior to simulating the model, you can open the Variable Viewer to see which of the variable targets have been satisfied. For more information on block-level variable initialization, see Variable Initialization.

Finding an Initial Steady State

When you select the Start simulation from steady state check box, the solver attempts to find the steady state that would result if the inputs to the system were held constant for a long enough time, starting from the initial state obtained from the initial conditions computation just described. If the steady-state solve succeeds, the state found is some steady state (within tolerance), but not necessarily the state expected from the given initial conditions. Steady state means that the system variables are no longer changing with time. Simulation then starts from this steady state.

A model can have more than one steady state. In this case, the solver selects the steady-state solution that is consistent with the variable targets specified during block-level variable initialization.

4.5 Transient Initialization

After computing the initial conditions, or after a subsequent event (such as a discontinuity resulting, for example, from a valve opening, or from a hard stop), the Simscape solver performs transient initialization. Transient initialization fixes all dynamic variables and solves for algebraic variables and derivatives of dynamic variables. The goal of transient initialization is to provide a consistent set of initial conditions for the next phase, transient solve.

4.6 Transient Solve

Finally, the Simscape solver performs transient solve of the system of equations. In transient solve, continuous differential equations are integrated in time to compute all the variables as a function of time.

The solver continues to perform the simulation according to the results of the transient solve until the solver encounters an event, such as a zero crossing or discontinuity. The event may be within the physical network or elsewhere in the Simulink model. If the solver encounters an event, the solver returns to the phase of transient initialization, and then back to transient solve. This cycle continues until the end of simulation.

5. Electrical System Using SimScape

In this example, you are going to model an electrical system and observe its behavior under various conditions. This tutorial illustrates the essential steps to building a physical model and makes you familiar with using the basic SimScape blocks.

The figure 3.4 shows a simple model of an electrical circuit. It consists of a Resistance, Capacitor and Inductor, which is supplied by a Voltage source.

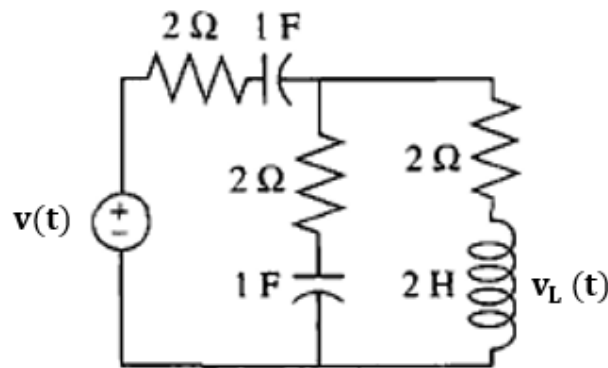


Figure 3.4: Electrical Circuit

• Modeling of System

To create an equivalent Simscape diagram, follow these steps:

1. Open the Simulink® Library Browser, as described in Simscape Block Libraries.
2. Create a new model. To do this, from the top menu bar of the Library Browser, select File > New > Model. The software creates an empty model in memory and displays it in a new model editor window.
3. Open the Simscape > Foundation Library > Electrical Element Elements library as shown in figure 3.5.
4. Drag the Mass, Translational Spring, Translational Damper, and two Mechanical Translational Reference blocks into the model window.

Experiment 3: Mathematical Modelling Using SimScape

5. Orient the blocks as shown in the figure 3.6. To rotate a block, select it and press Ctrl+R.
6. Connect the Resistor, Capacitor, and Inductor blocks to one of the Electrical Reference Ground blocks as shown in figure 3.7.

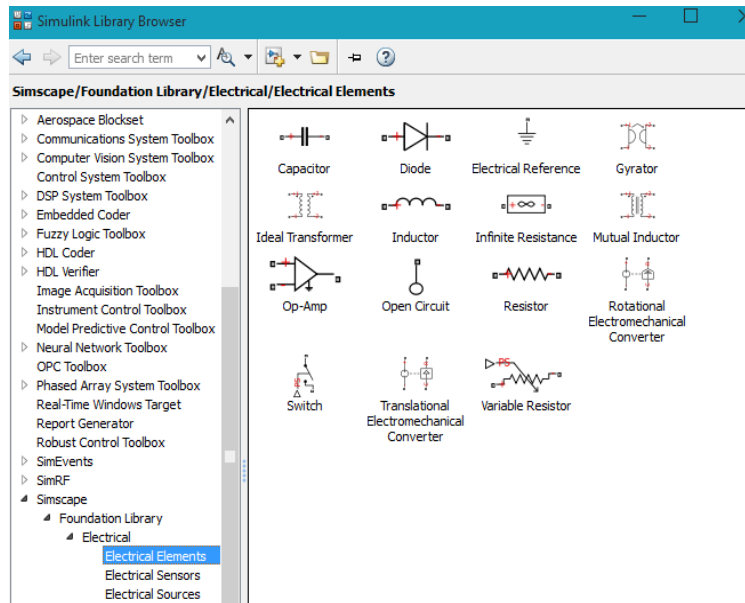


Figure 3.5: Electrical Element, step 3

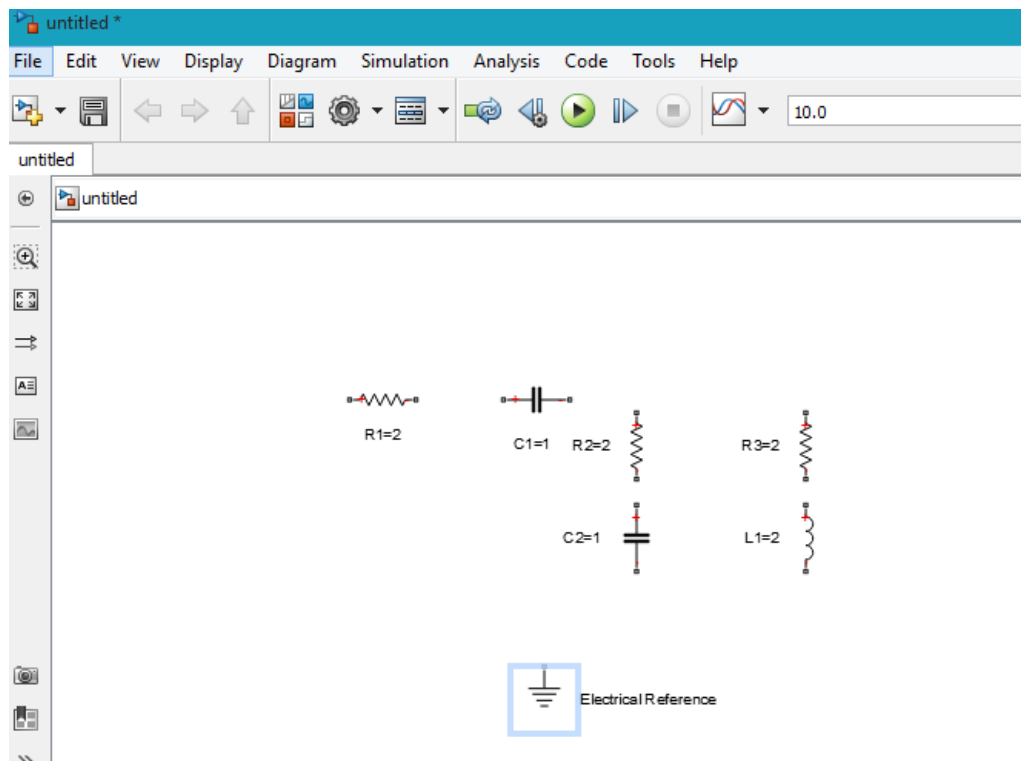


Figure 3.6: Selected Element, step 5

Experiment 3: Mathematical Modelling Using SimScape

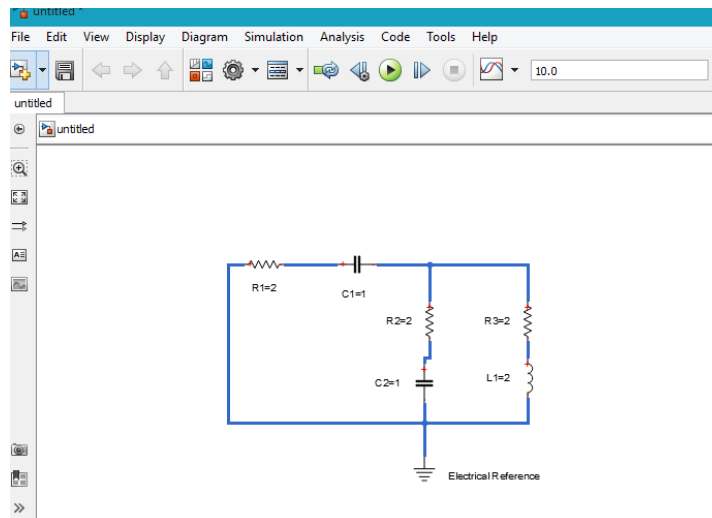


Figure 3.7: Connect the element, step6

7. To add the representation of the force acting on the mass, open the Simscape > Foundation Library > Electrical > Electrical Sources library figure 3.8, and add the Controlled Voltage Source block to your diagram as figure 3.9.

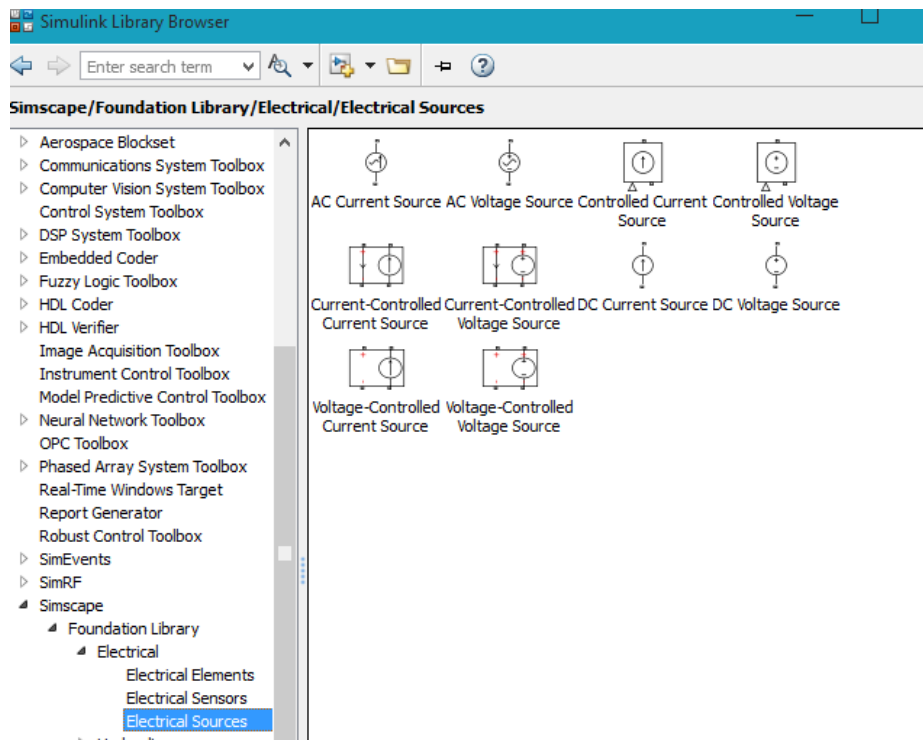


Figure 3.8: Electrical Source, step7

Experiment 3: Mathematical Modelling Using SimScape

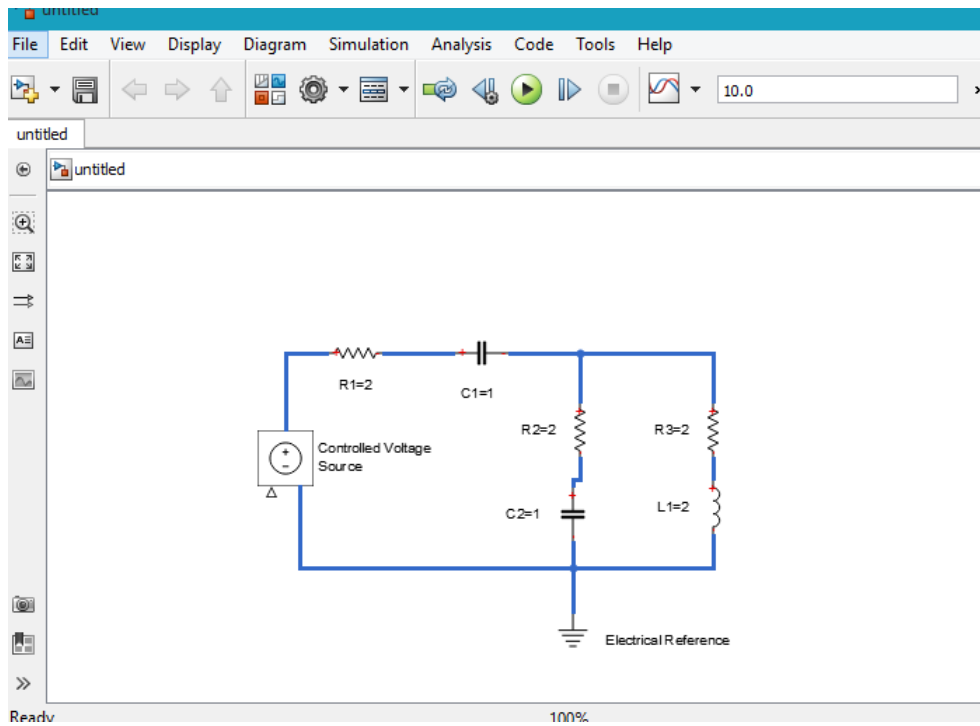


Figure 3.9: The block of circuit after add the voltage source, step7

8. Add the sensor to measure speed and position of the mass. Place the Voltage and current Sensors block from the Electrical Sensors library figure 3.10 into your diagram and connect it as shown in figure 3.11.

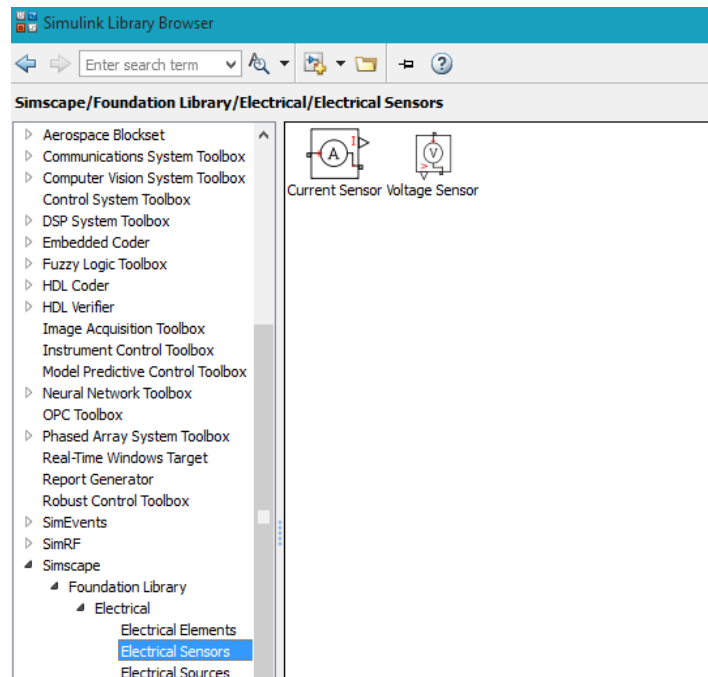


Figure 3.10: Electrical Sensor

Experiment 3: Mathematical Modelling Using SimScape

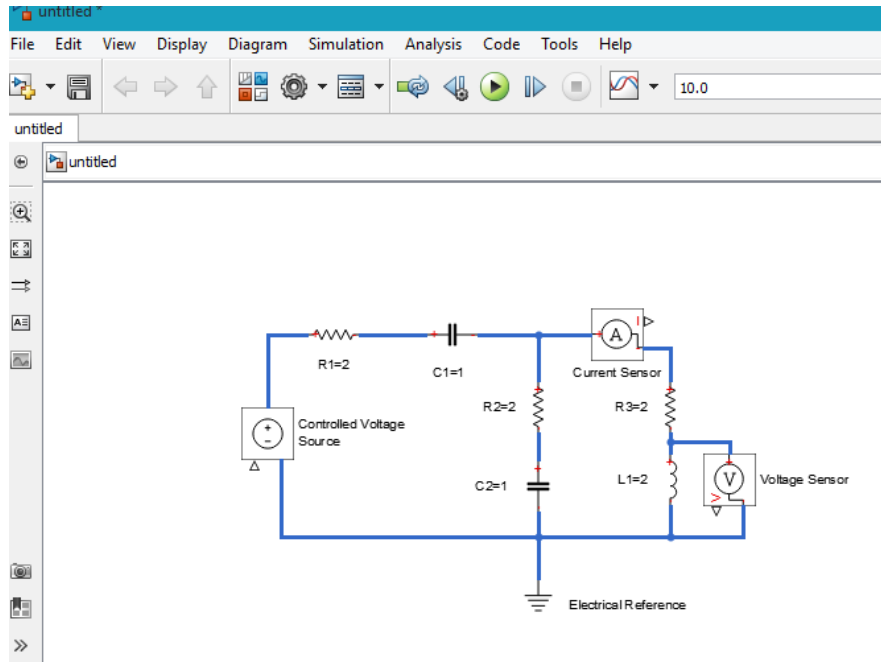


Figure 3.11: Electrical circuit using SimScape

9. Now you need to add the sources and scopes. They are found in the regular Simulink libraries. Open the Simulink > Sources library and copy the Signal Builder block into the model. Then open the Simulink > Sinks library and copy two Scope blocks..
10. Every time you connect a Simulink source or scope to a Simscape diagram, you have to use an appropriate converter block, to convert Simulink signals into physical signals and vice versa. Open the Simscape > Utilities library figure 3.12 and copy a Simulink-PS Converter block and two PS-Simulink Converter blocks into the model. Connect the blocks as shown in figure 3.13.
11. Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block, found in the Simscape > Utilities library figure 3.12. Copy this block into your model and connect it to the circuit by creating a branching point and connecting it to the only port of the Solver Configuration block. Your diagram now should look like figure 3.13.

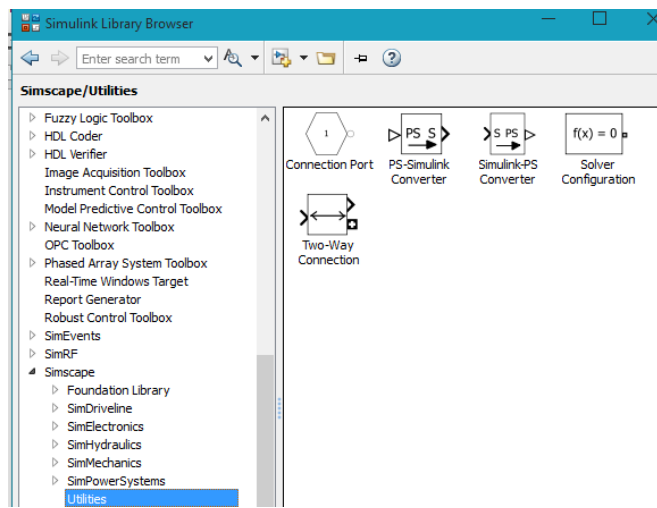


Figure 3.12: Simscape Utilities Library

Experiment 3: Mathematical Modelling Using SimScape

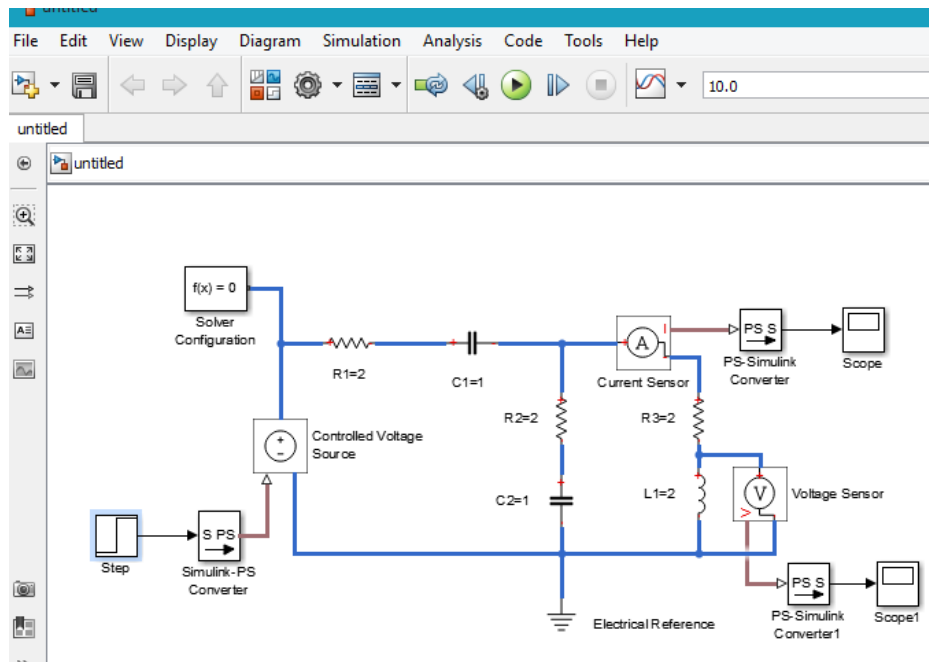


Figure 3.13: Full Simscape Representation for Electrical Circuit

The Simulation Result

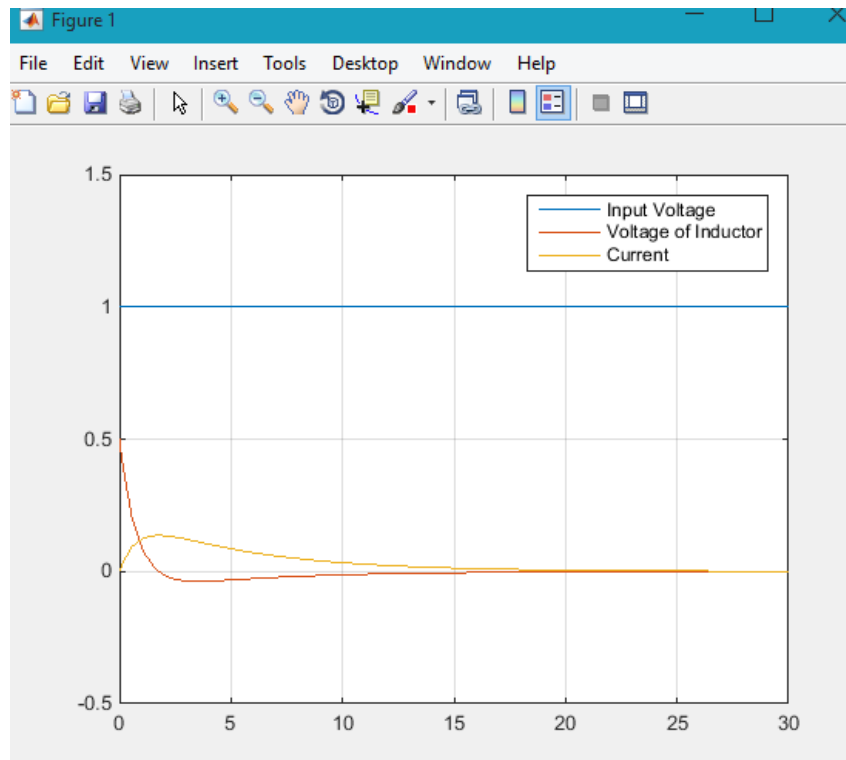


Figure 3.14: The Simulation Result

References

- 1) <http://www.mathworks.com/help/index.html>
- 2) Farid Golnaraghi, Benjamin C.Kuo; Automatic Control Systems; Ninth Edition
- 3) Norman S.Nise; Control Systems Engineering; Sixth Edition
- 4) Richard C.Dorf, Robert H.Bishop; Modern Control Systems; Twelfth Edition.