PIRAEUS UNIVERSITY OF APPLIED SCIENCES

TECHNOLOGICAL EDUCATION INSTITUTE OF PIRAEUS

DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING

THESIS

Physical Systems Simulation using SIMSCAPE

Panagiotis C. Mentzelopoulos

Rapporteur: Anastasia Veloni, Applications Lecturer

Athens
JANUARY 2018

THESIS

Physical Systems Simulation using SIMSCAPE

Panagiotis C. Mentzelopoulos
A.M 40740

Rapporteur:

Anastasia Veloni, Applications Lecturer

Examination Committee:

………………………………………………..

Examination Date ../../2017

THANKINGS

First and foremost, I would like to thank Mrs Anastasia Veloni for her continuous support and guidance throughout the completion of my thesis.

A big thank you to my family: my parents and my brother for supporting me academically and in life.

Lastly, I would like to thank the people in my life that knowingly or not helped me finish my thesis.

ABSTRACT

The thesis introduces the user on how to create physical systems using SIMSCAPE and run simulations for these systems. The thesis walks the reader through the process of getting familiar with the MATLAB and Simscape environment, understanding the Simscape toolbox options , listing the available components provided by Simscape as well as creating custom components through the Simscape Language. Also some of the built-in ready-to-use demos of completed models ready for simulation are shown as well as the creation of a new custom model from start to finish.

The whole thesis approach is very user friendly providing many images and step by step instructions.

SCIENTIFIC AREA: Computer Engineering
KEYWORDS: Simscape, Physical Systems, Simulation, MATLAB, Simulink

CONTENTS

SCHEMATICS CATALOGUE

ARRAY CATALOGUE

ABBREVIATIONS

OS: Operating System

CHAPTER 1

INTRODUCTION

Ever since the discovery of computers there has been massive advancement in technology, engineering, physics and countless other domains. Life has been made easier, most of the products we use today were developed by using computers. What that means is that in order for a new product to hit the market there is a whole process behind it. First there is the designing process in which an idea for a product is converted into a representation of a physical system. Next is the simulation of this system in order to test the design, see how it reacts against factors like wind, gravity, etc. Computer simulations are a very powerful and useful tool, they can be either small-scale running very fast in small devices for a limited period of time or they can be large-scale running for days using extreme computing power.

Computer software combines the design of a model and its simulation. [1] A software company called MathWorks specializes in mathematical computing and their major products are MATLAB and Simulink which support data analysis and simulation. Simulink is a graphical and simulation environment for Model-Based Design. They are both used in automotive, aerospace, software and countless other fields. Simscape is a package available through Simulink.

[2] Simscape gives users the ability to create models of physical systems by building physical component models based on physical connections that directly integrate with block diagrams and other modeling paradigms. By assembling fundamental components into a schematic users can model systems and run simulations. We can test these systems on system-level performance. A great feature of Simscape is the Simscape Language where users can create their own custom components. Users can alter the parameters of the models using MATLAB variables and expressions. Also there is support for cross-platform simulation environments by C-code generation.

CHAPTER 2

MATLAB

2.1 What is MATLAB?

[3] MATLAB(**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment. It is a high performance programming language written in C, C++, Java developed by MathWorks. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python. Its multi-functionality and endless applications have made MATLAB extremely popular across the industry and academia.

2.2 History of MATLAB

MATLAB was developed in the late 1970's by Cleve Moler, chairman of the computer science department at the UNM(University of New Mexico). The idea behind it's development was to give his students access to LINPACK and EISPACK without them having to learn Fortran. Many universities started to acknowledge the power of MATLAB and started using it, mainly the applied mathematics community. After Moler's visit to Stanford, an engineer named Jack Little, recognized the commercial potential of MATLAB. He then joined with Moler and Steve Bengert, they rewrote MATLAB in C and founded MathWorks in 1984.

2.3 MATLAB's evolution

At first, MATLAB was used by researchers and practitioners in control engineering. Over the years MATLAB has evolved with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. It is also used in image processing.

2.4 MATLAB Breakdown

[4] The MATLAB system consists of five main parts :

i) The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

ii)The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

iii)Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

iv)The MATLAB mathematical function library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

v)The MATLAB Application Program Interface (API)

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

2.5 MATLAB Interface

MATLAB’s user interface is easy-to-use. The core of each version is very similar to each other. It looks like this.

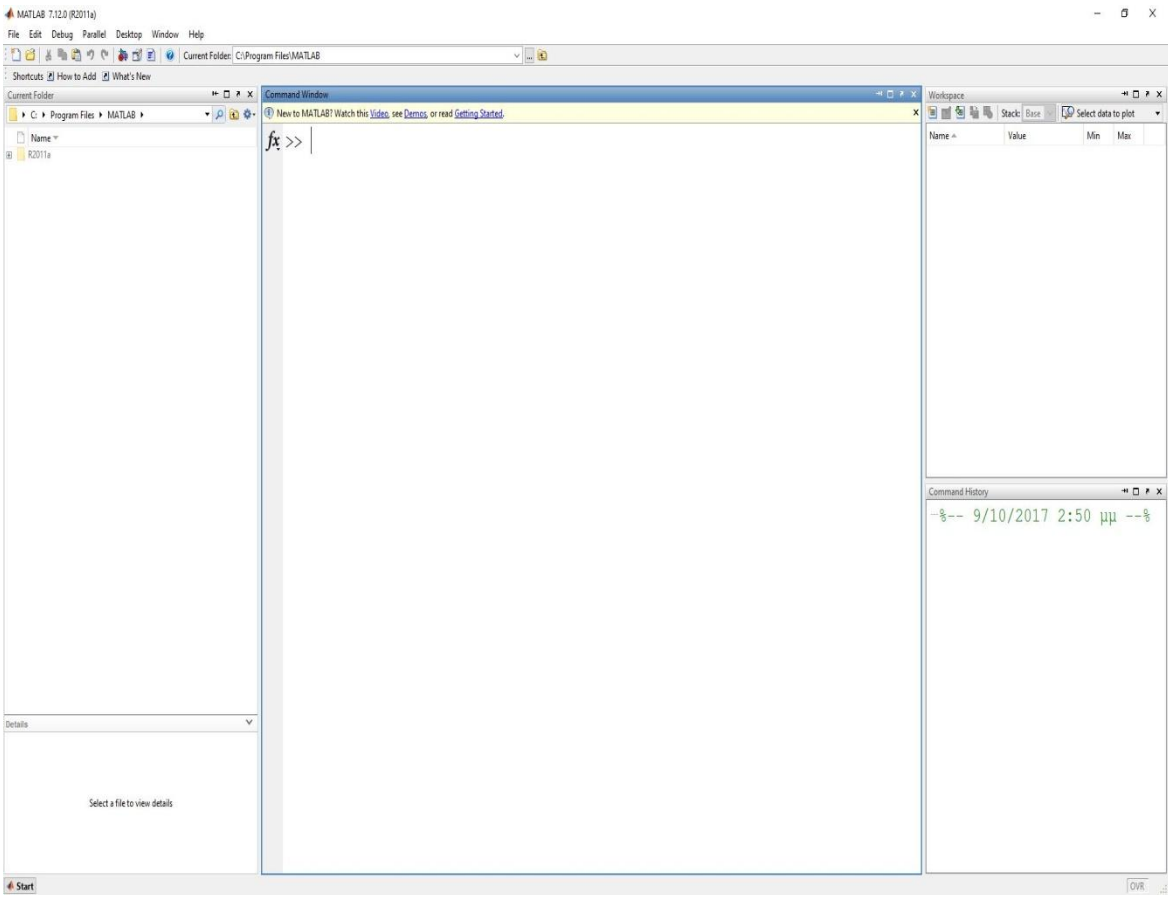

Image 2.1: MATLAB's User Interface

At the top left corner there is a menu bar consisting of **File**, **Edit**, **View**, **Debug**, **Parallel**, **Desktop**, **Window**, **Help**. Below that, there is the **MATLAB Toolbar** and below that the **Shortcuts Toolbar**. In the **Current folder** section users can navigate wherever in the file system of the OS in order to access files, usually **m-files**. The main window is the **Command Window** in which the user types the code. Whenever a command is written which stores values, it can be found and accessed in the **Workspace Window**. Last, there is a **Command History** window. Each and any window can be minimized, maximized, docked, undocked or closed.

CHAPTER 3

SIMSCAPE

3.1 What is SIMSCAPE?

[5] SIMSCAPE is a toolbox for physical modeling developed by MathWorks for SIMULINK. It has been available since version R2007A of the MATLAB suite. It includes a foundation library, which contains basic components for electrical, hydraulic, magnetic, mechanical, physical signals, Pneumatic and thermal systems. There are also more specialized toolboxes for physical modeling such as SimDriveline, SimElectronics, SimHydraulics and SimMechanics that now are considered as parts of the Simscape product family although some of them had been around before Simscape. In R2008b, a major upgrade of Simscape was made, introducing the Simscape language which allows the user to create their own physical models, and even new physical domains, with new conserving ports. The language is based on MATLAB syntax. Simscape distinguishes between the variables as: Through Variables that are measured with a gauge connected in series to an element. Across Variables that are measured with a gauge connected in parallel to an element.

3.2 Getting to know SIMSCAPE

In order to run SIMSCAPE, start MATLAB and type “**simscape**” in the MATLAB Command Window.

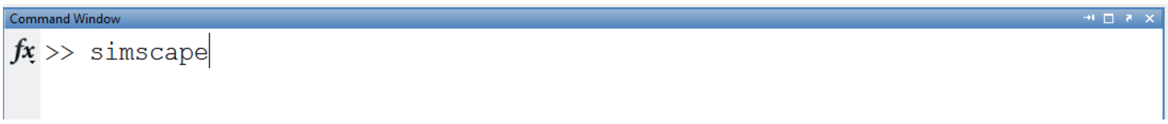

Image 3.1: MATLAB’s Command Window

When that is done, SIMSCAPE is going to run and will be ready for use.

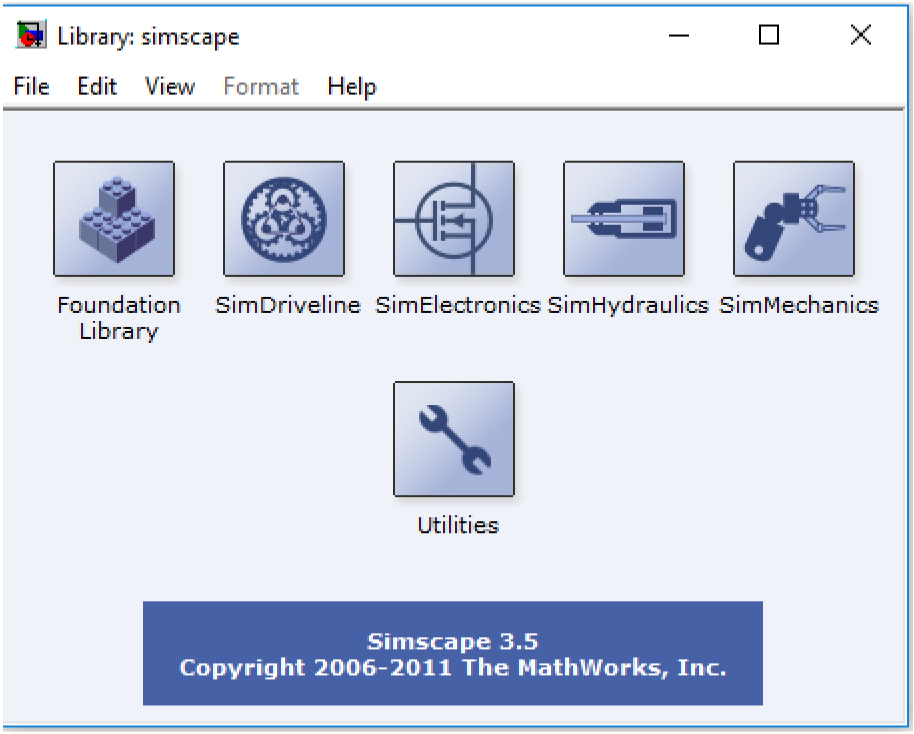

Image 3.2: SIMSCAPE's Interface

SIMSCAPE's Toolbar

The menu bar consists of **File, Edit, View, Format** and **Help**

Image 3.3: SIMSCAPE's Toolbar

Let's take a closer look at the File option.

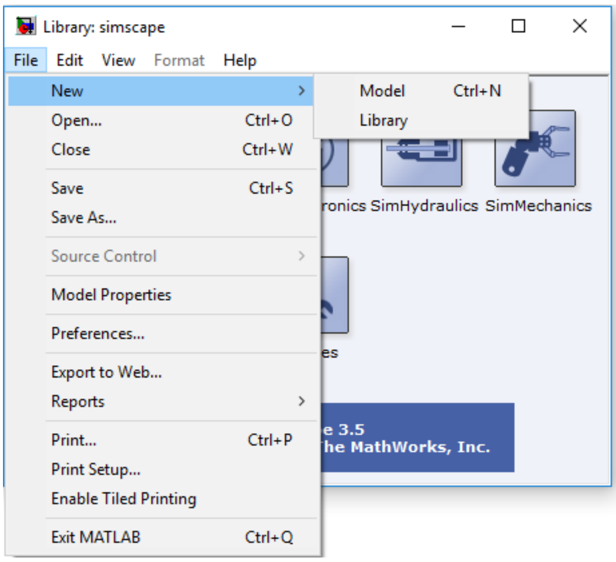

Image 3.4: SIMSCAPE's File toolbar

By selecting New -> you get two options. You can create a new model or a new library

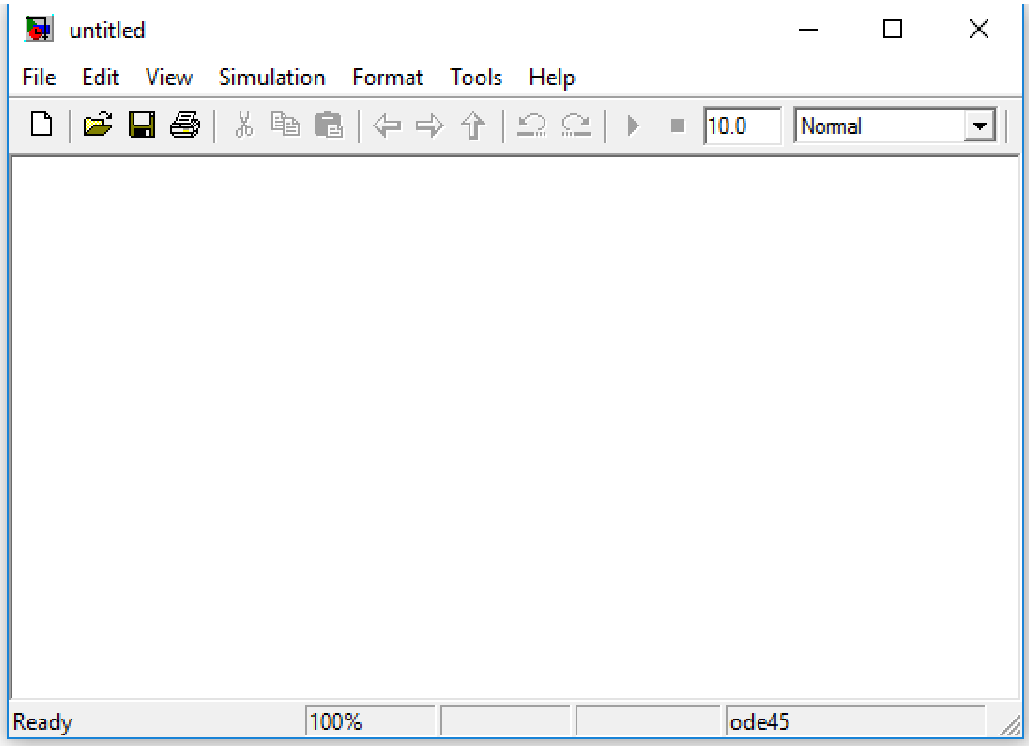

Image 3.5: SIMSCAPE's New Model Window

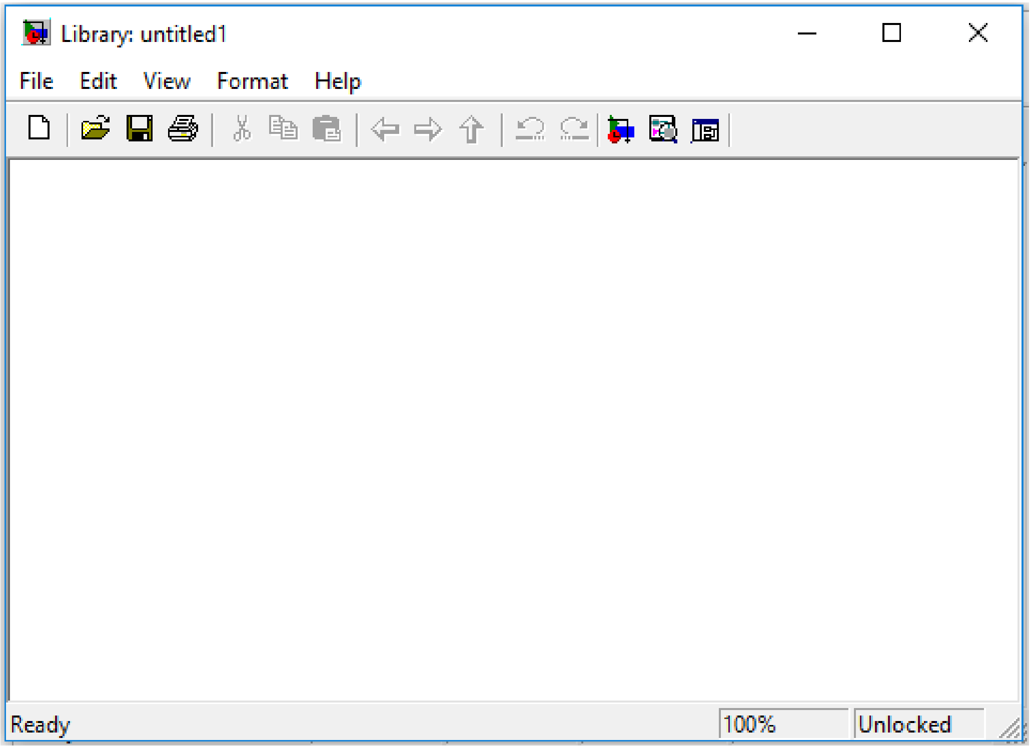

Image 3.6: SIMSCAPE's New Library Window

By selecting **Open**(Ctrl+O) you can load a previous model or library that is saved on your computer.

Close(Ctrl+W) terminates SIMSCAPE. In order to load SIMSCAPE again, type "**simscape**" at the MATLAB Command Window.

Save(Ctrl+S) and **Save as** give you the option to save your work as a .mdl file which is a Simulink Model extension.

Model Properties dialog box allows you to set various version control parameters and model callback functions.

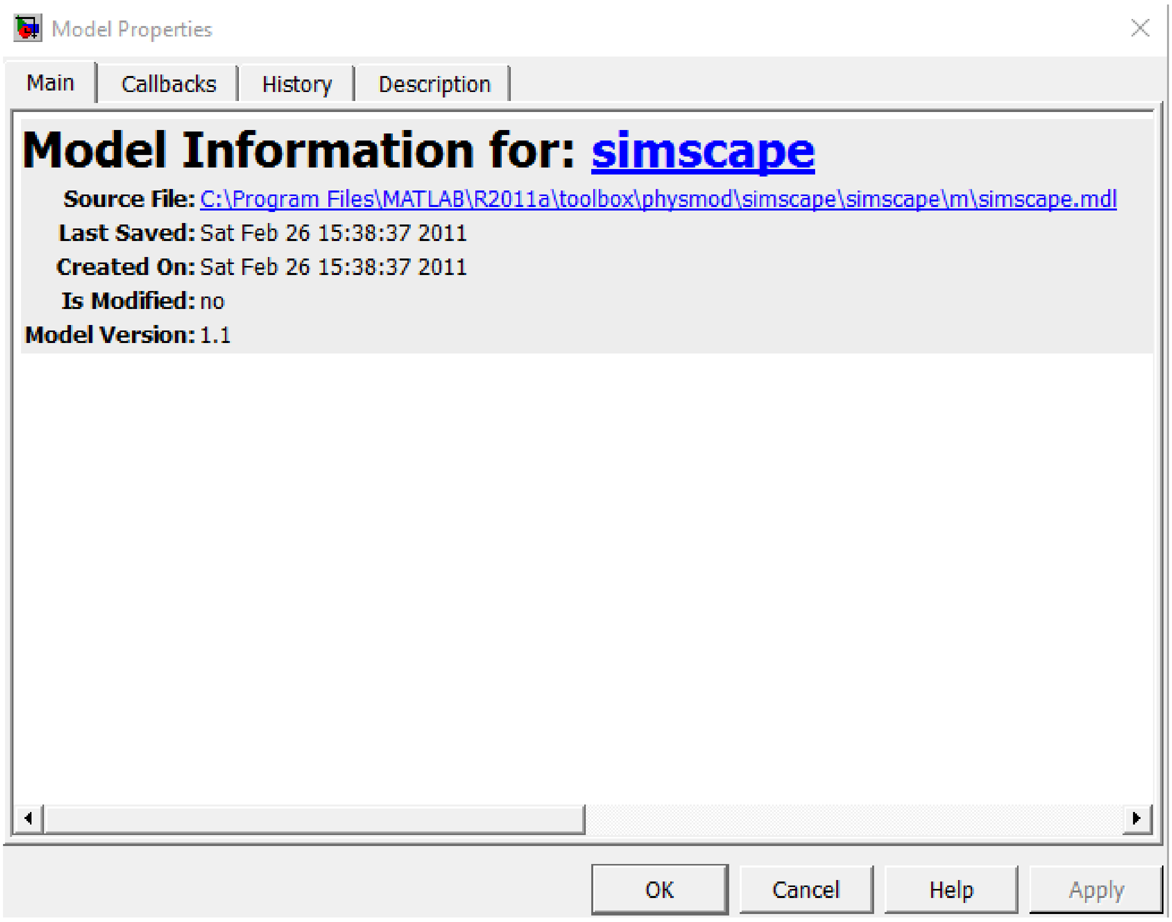

Image 3.7: SIMSCAPE's Model Properties

Preferences tab opens up Simulink's preferences where you can change the display, fonts etc.

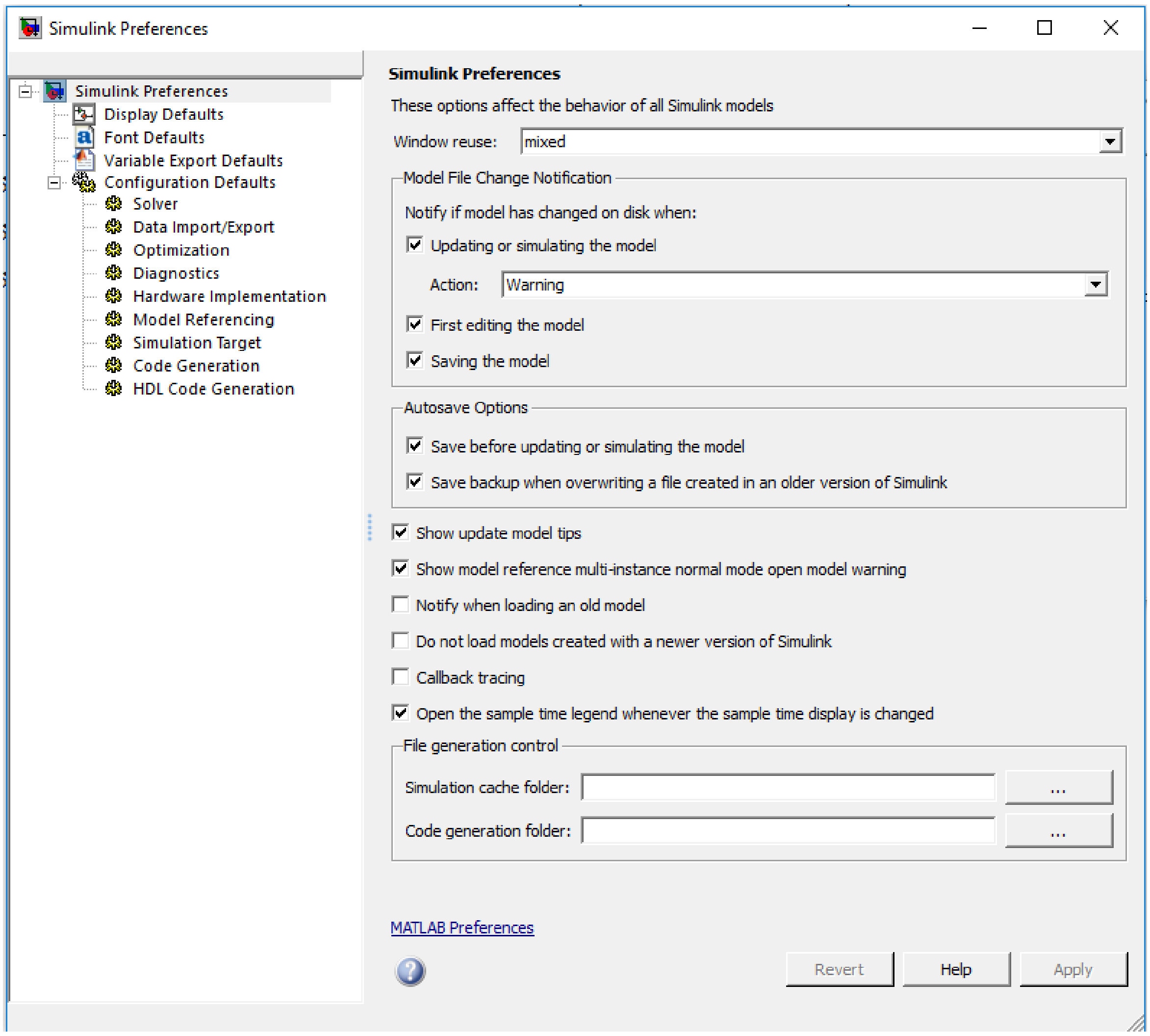

Image 3.8: Simulink's Preferences

Export to Web… You can export a model to a Web view using either a Simulink Report Generator dialog box or a Report Explorer panel.

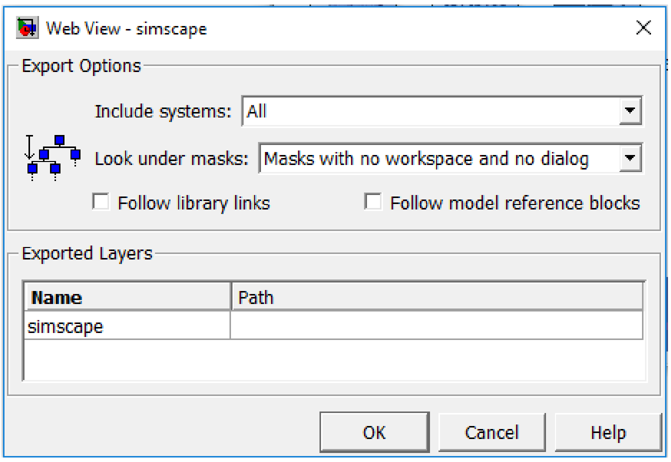

Image 3.9: SIMSCAPE's Export to Web

Reports -> System Design Description Here you can choose options for the content, format, and location of the generated System Design Description report. To customize the report template, click the Customize button to open the report in the Report Explorer.

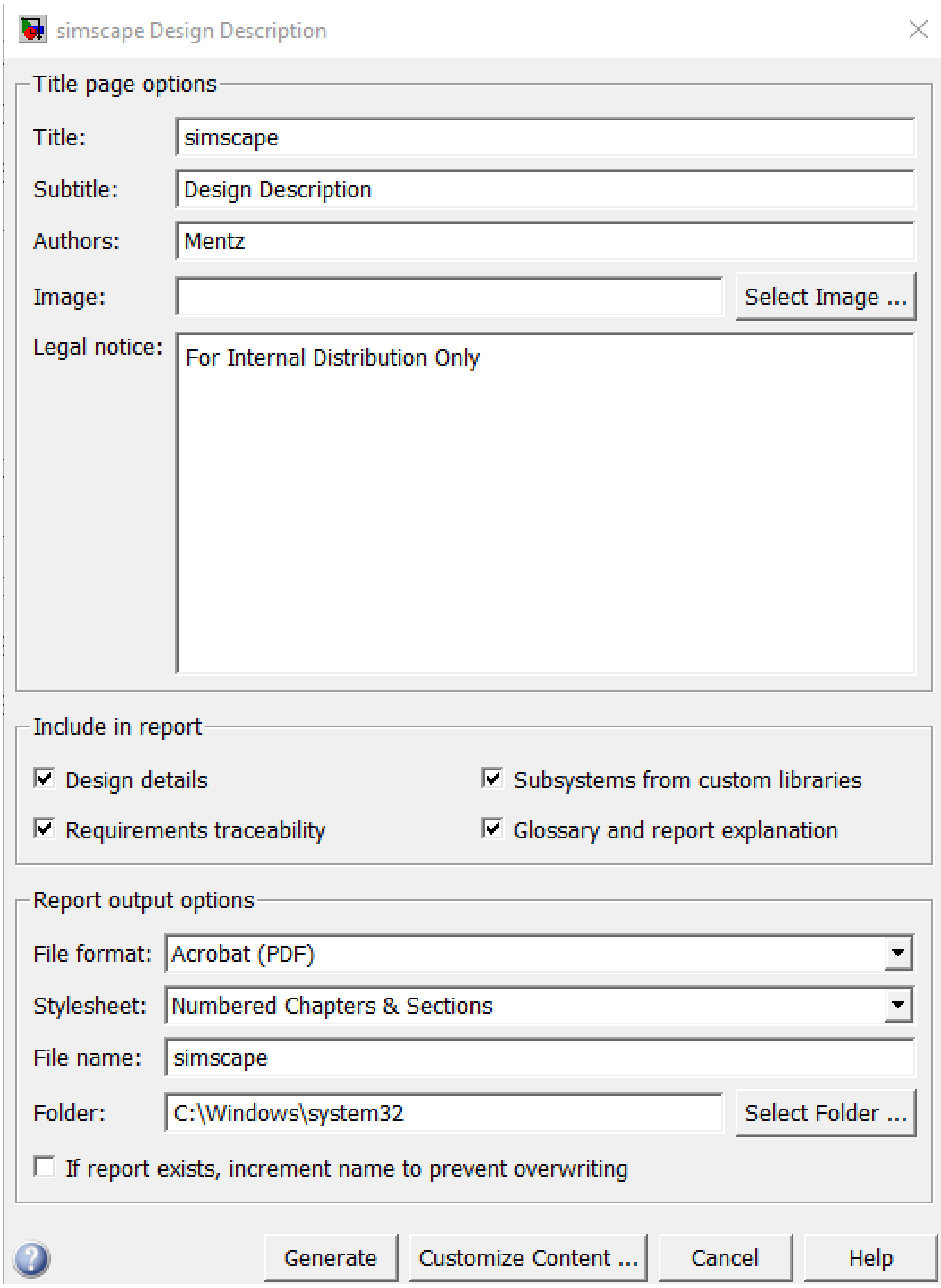

Image 3.10: SIMSCAPE's Design Description

Reports-> Design Requirements opens up a Requirements Report for simscape.

Print(Ctrl+P) allows you to print the Models with multiple options.

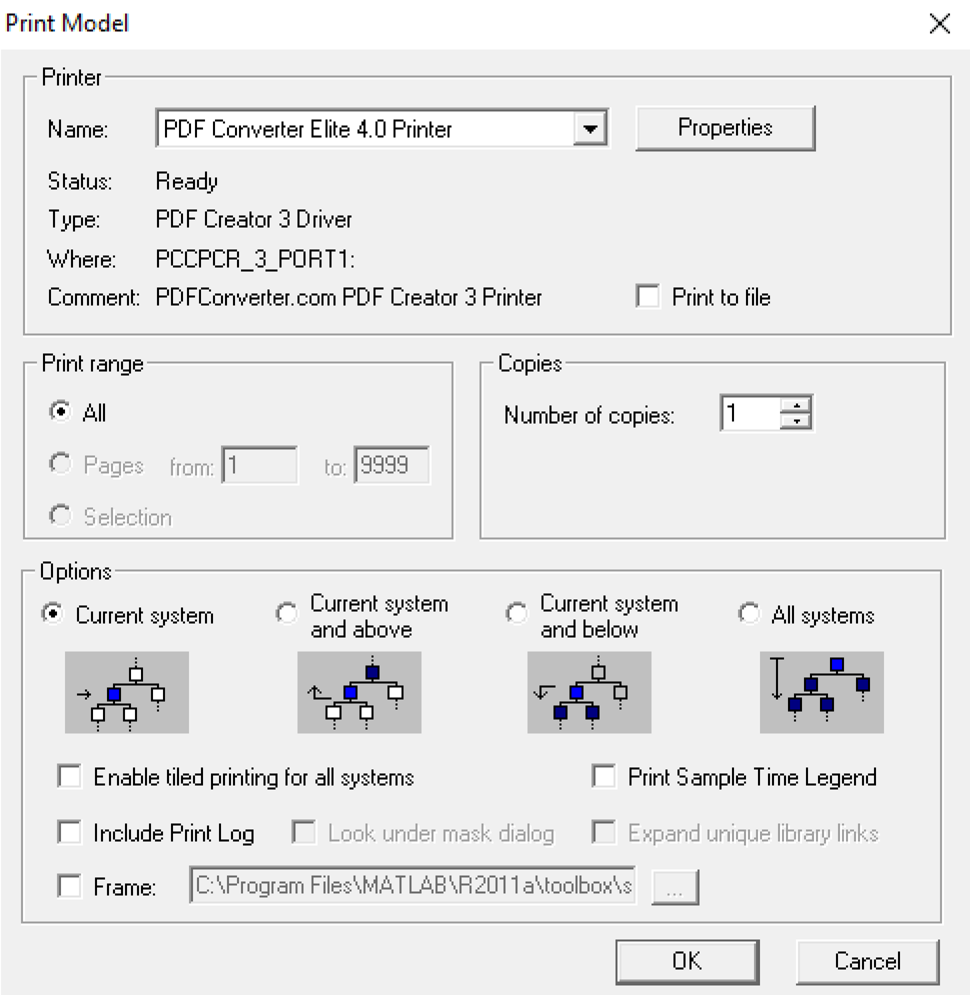

Image 3.11: SIMSCAPE's Print

Print Setup allows you customize the print options.

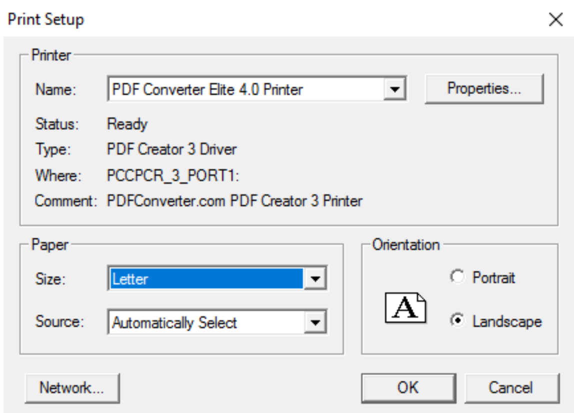

Image 3.12: SIMSCAPE's Print Setup

Enable Tiled Printing By default, each block diagram is scaled during the printing process so that it fits on a single page. In the case of a large diagram, this automatic scaling can make the printed image difficult to read.

Tiled printing enables you to print even the largest block diagrams without sacrificing clarity and detail. Tiled printing allows you to distribute a block diagram over multiple pages. For example, you can use tiling to divide a model as shown in the figure, with each white box and each gray box representing a separate printed page.

Exit MATLAB(Ctrl+Q) closes SIMSCAPE and Matlab.

Now let's take a look at the **Edit** section.

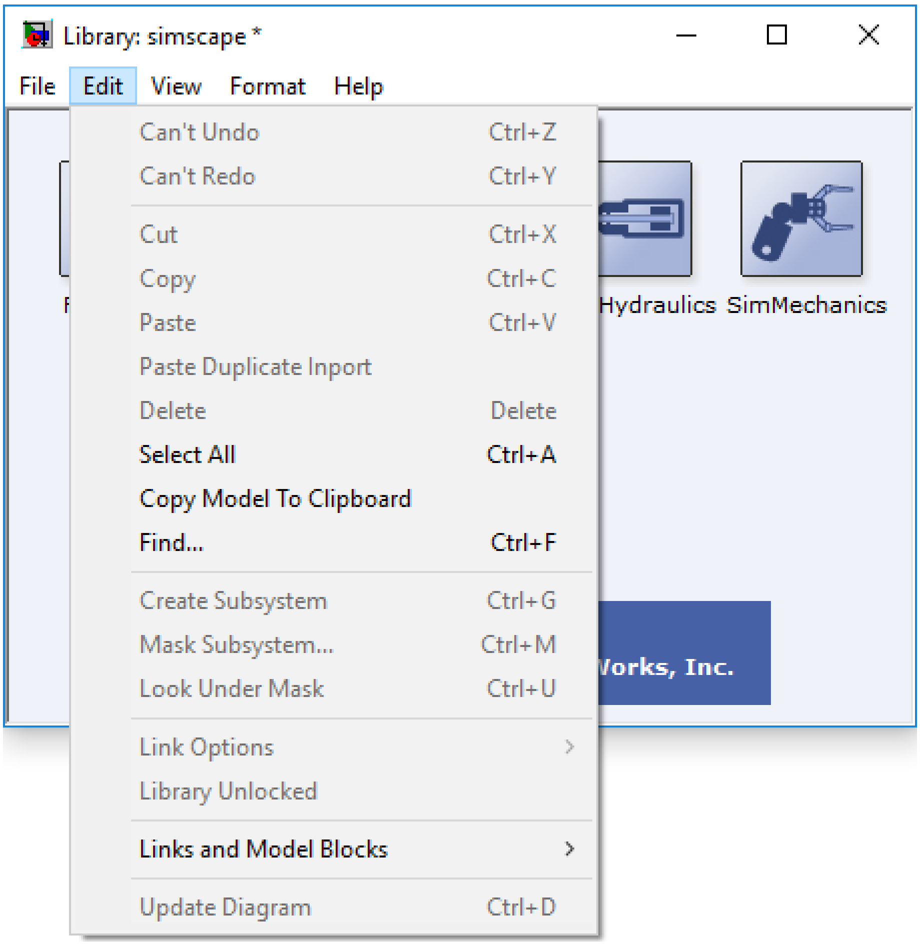

Image 3.13: SIMSCAPE's Edit toolbar

Undo(Ctrl+Z) Undoes the last command.

Redo(Ctrl+Y) Redoes the last command.

Cut(Ctrl+X) Cuts the selected components.

Copy(Ctrl+C) Copies the selected components.

Paste(Ctrl+V) Pastes the copied/cat components.

Paste Duplicate Inport -> Pastes duplicates of an Inport block.

Select All(Ctrl+A) Selects every component.

Copy Model to Clipboard -> Copies the entire model

Find…(Ctrl+F) Opens up a search window

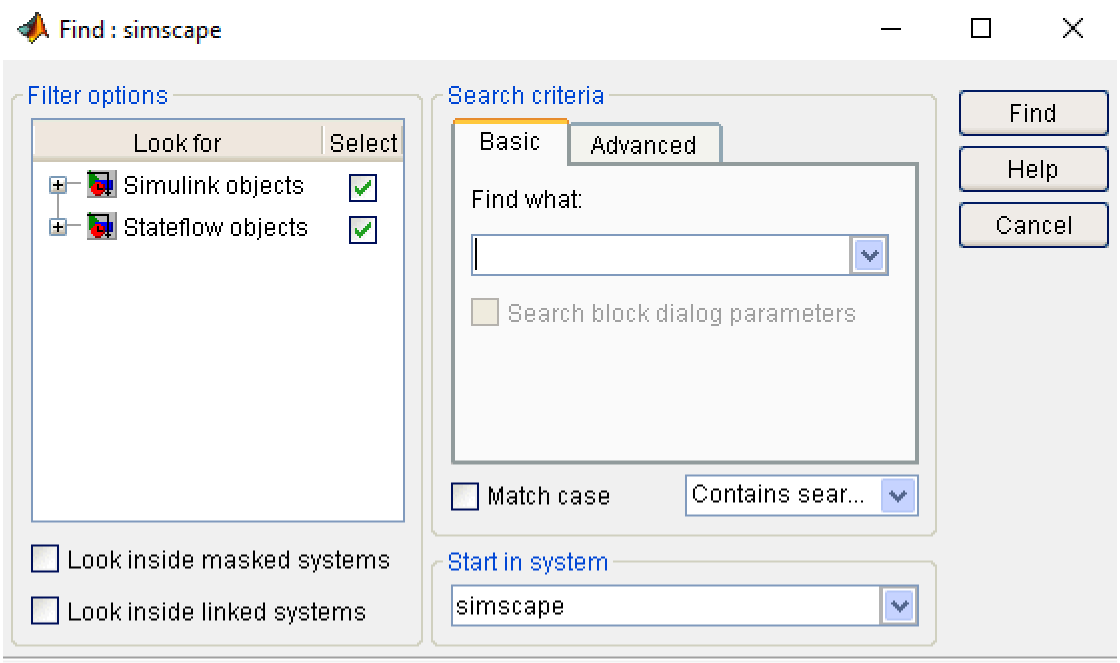

Image 3.14: SIMSCAPE's Find Window

Create Subsystem(Ctrl+G) Subsystems allow the creation of a hierarchical model of many layers.

Mask Subsystem(Ctrl+M) Masks a subsystem.

Look Under Mask(Ctrl+U) Shows what is beneath a Mask.

Unlock Library -> Unlocks the Library.

Links and Model Blocks -> Refreshes the Model in case of additions not showing in model.(Ctrl+K)

Update Diagram(Ctrl+D) Updates the currently selected Diagram.

View Section

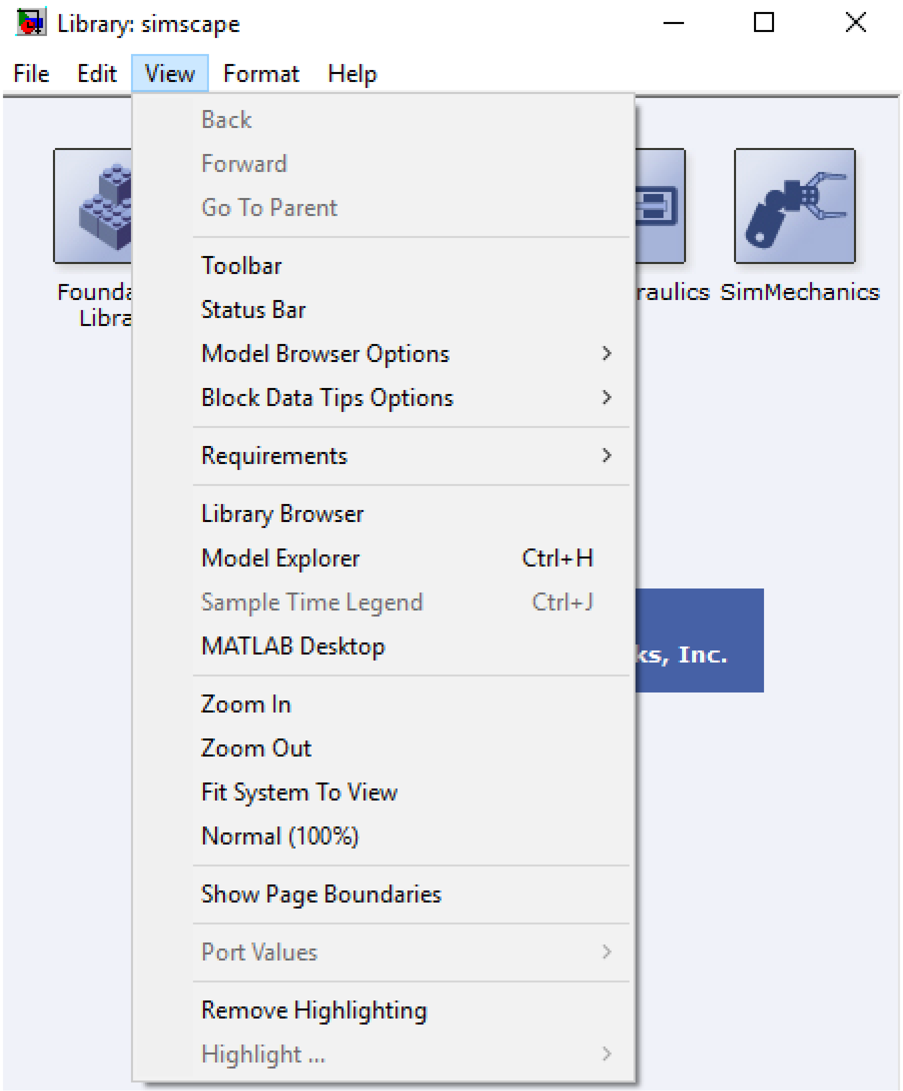

Image 3.15: SIMSCAPE's View Window

Back -> Goes to the previous selection
Forward -> Goes to the next selection
Go To Parent -> Goes to the source
Toolbar -> Shows/Hides the toolbar
Status Bar ->Shows/Hides the status bar
Model Browser Options Shows/Hides:
->Model Browser
->Show library links
->Show Masked subsystems
Block Data Tips Options Shows/Hides:
->Block Name
->Parameter Names And Values
->User Description String

Requirements

Add link to Word Selection -> Outputs to word file
Add link to active Excel cell-> Outputs to excel cell
Edit/Add Links… :

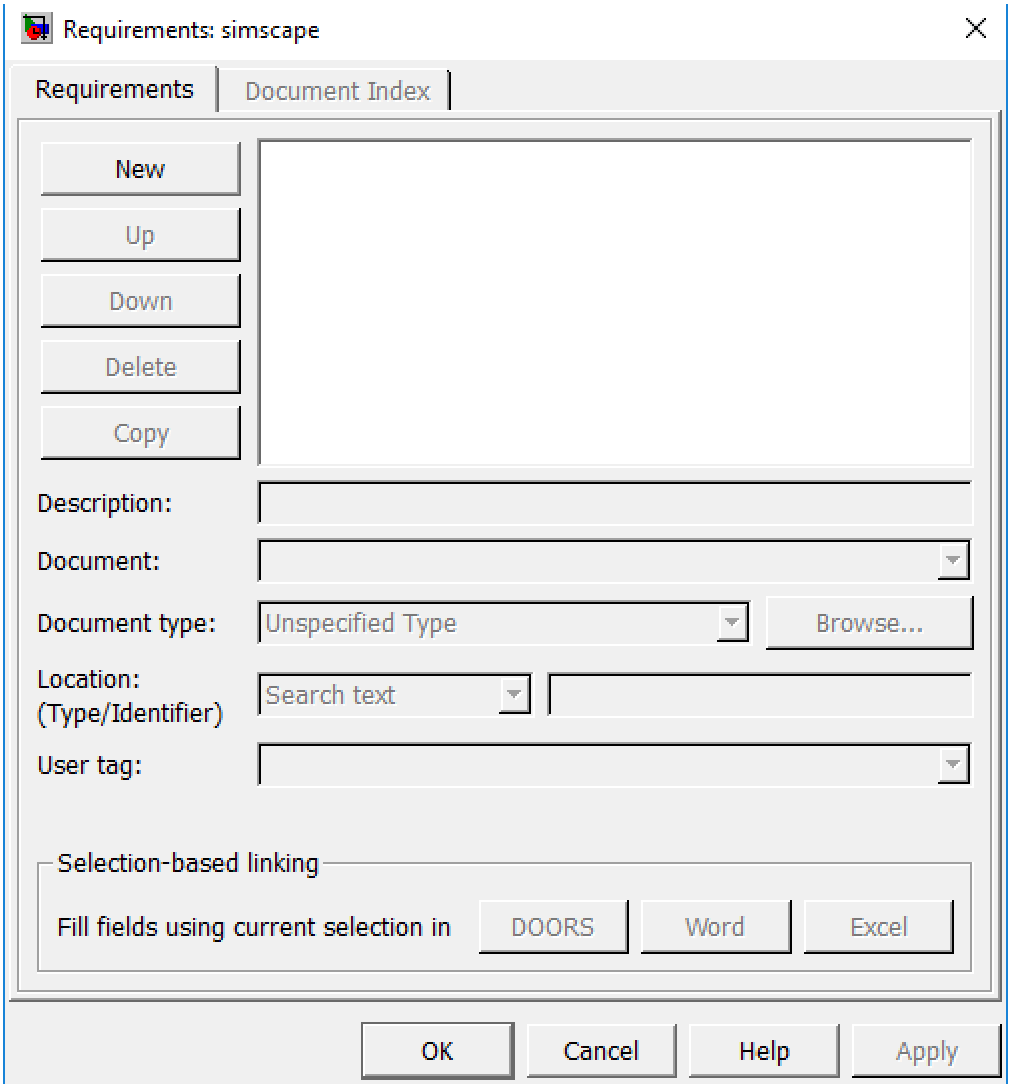

Image 3.16: SIMSCAPE's Requirements Window

In this window you can edit the Requirements for each model.

Settings…

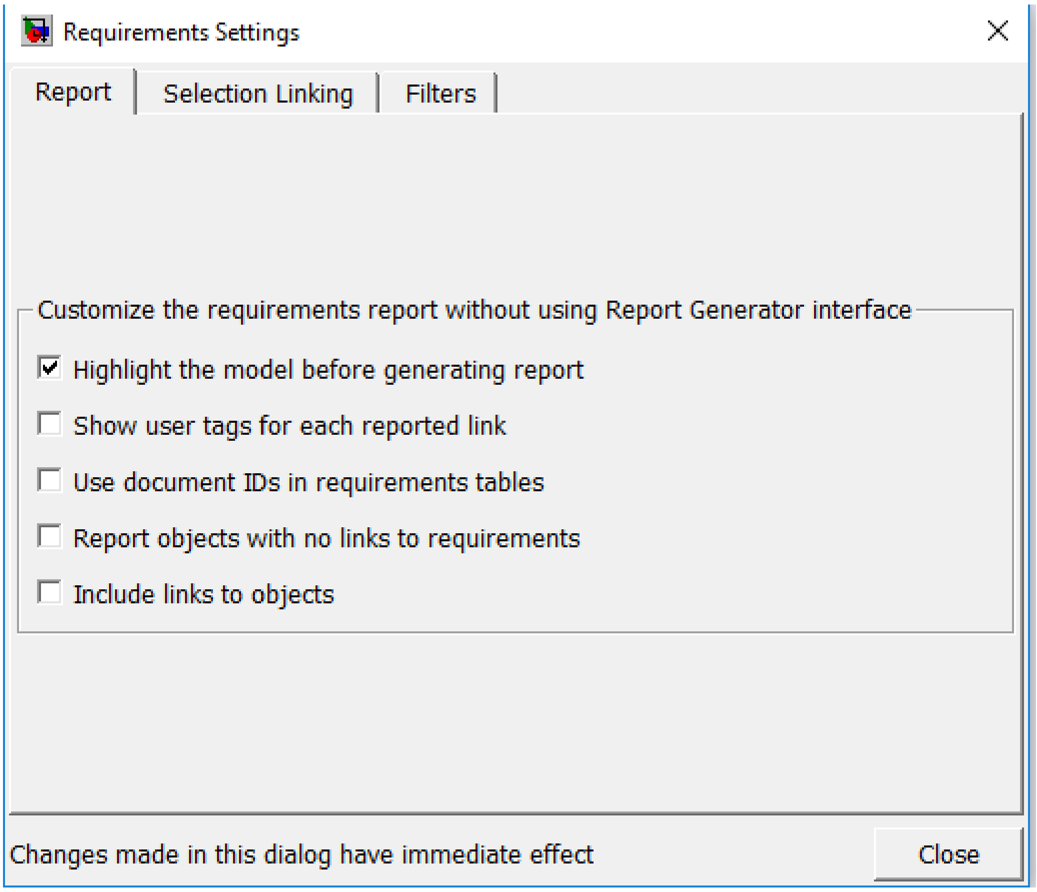

Image 3.17: SIMSCAPE’s Requirements Settings Window

Library Browser

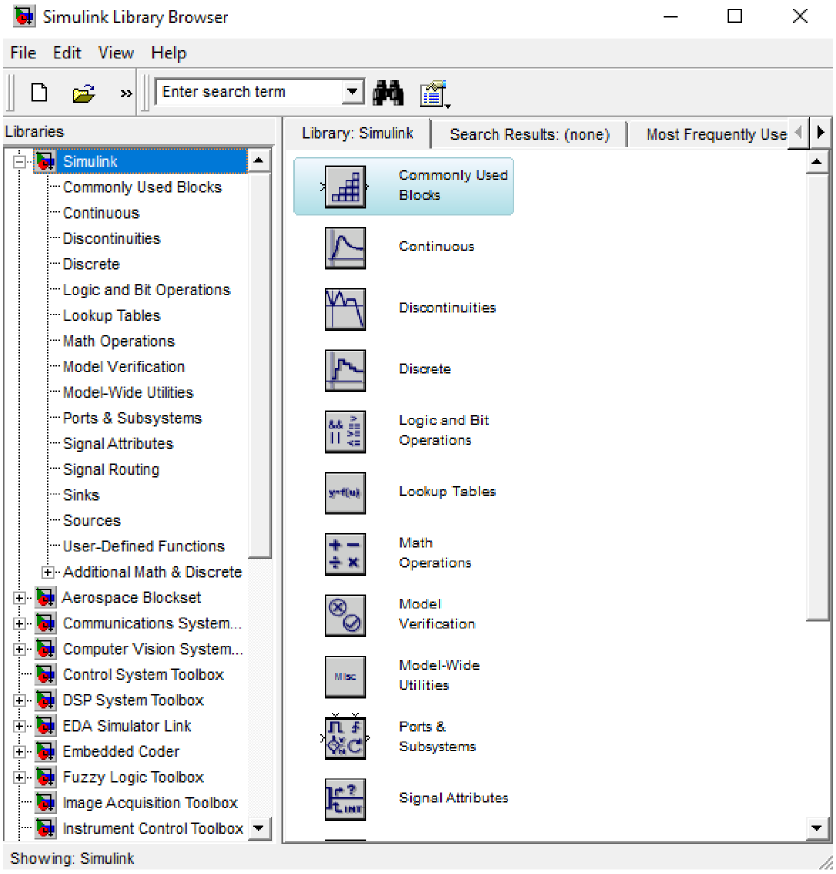

Image 3.18: SIMSCAPE's Library Browser Window

Model Explorer(Ctrl+H)

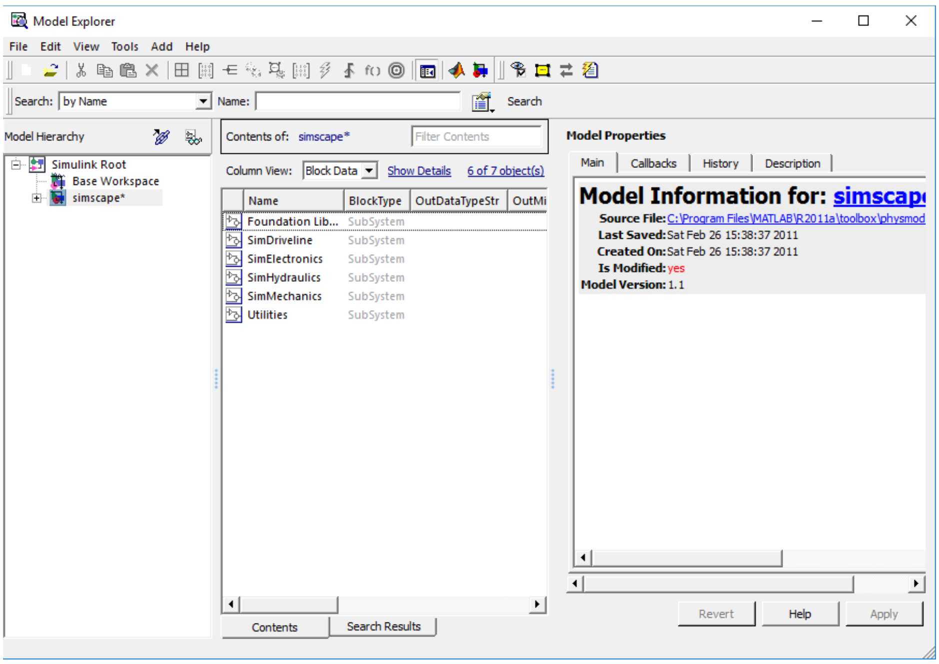

Image 3.19: SIMSCAPE's Model Explorer Window

Sample Time Legend(Ctrl+J) ->
MATLAB Desktop -> Shows MATLAB
Zoom In -> Zooms In
Zoom Out -> Zooms Out
Fit System To View -> Fits the window
Normal(100%)
Show Page Boundaries -> Shows Page Boundaries
Post Values -> Shows the value of the components
Remove Highlighting -> Removes highlighting
Highlight… -> Highlights a selection

Help Section

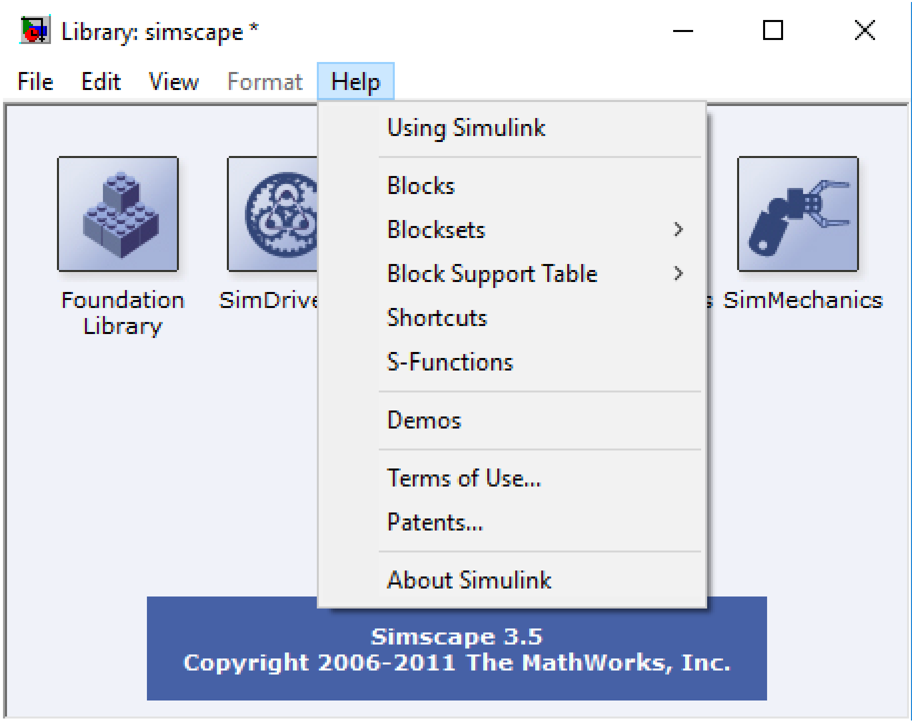

Image 3.20: SIMSCAPE's Help Window

Using Simulink

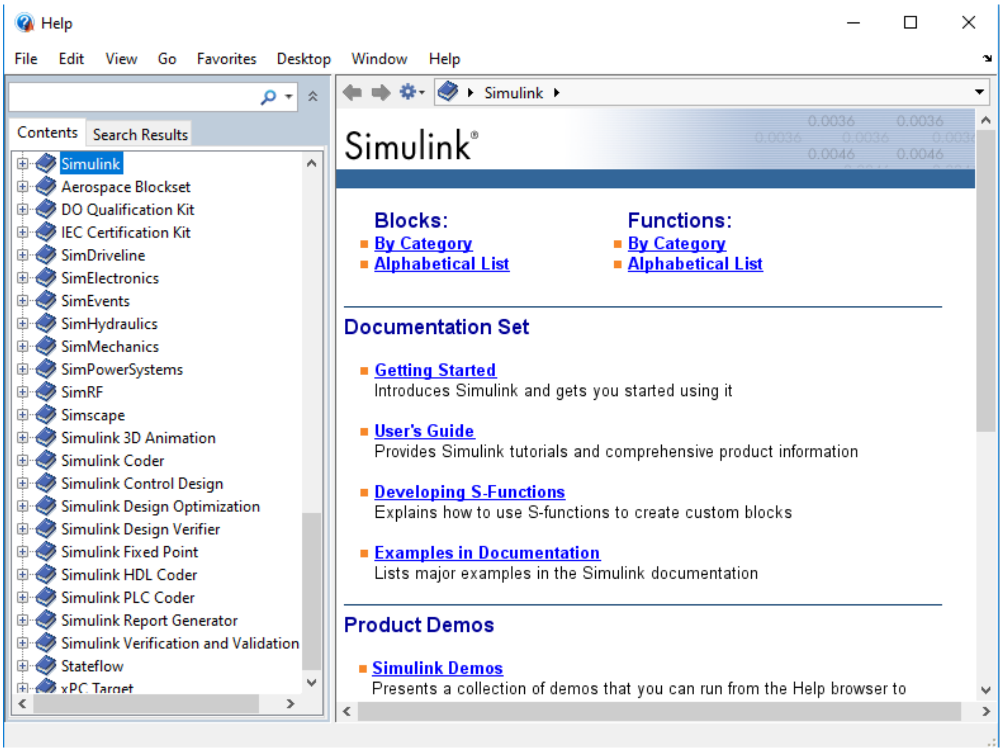

Image 3.21: SIMSCAPE's Help Window

Blocks

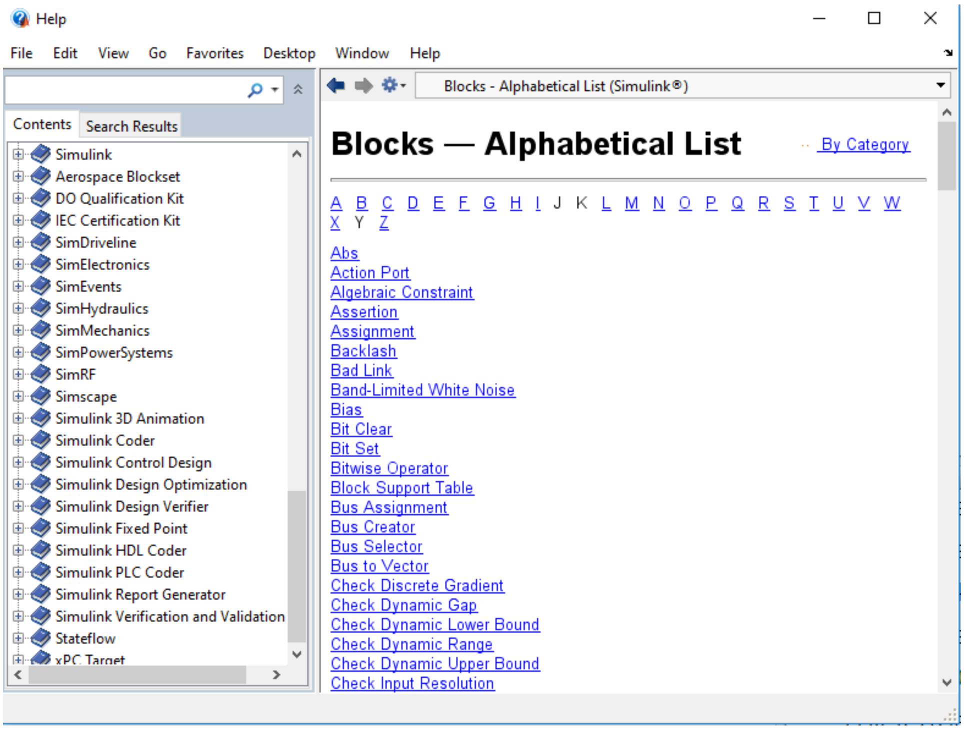

Image 3.22: SIMSCAPE's Blocks Window

Blocksets -> Brings up a list of Blocksets to chose from.

Block Support Table

Shortcuts

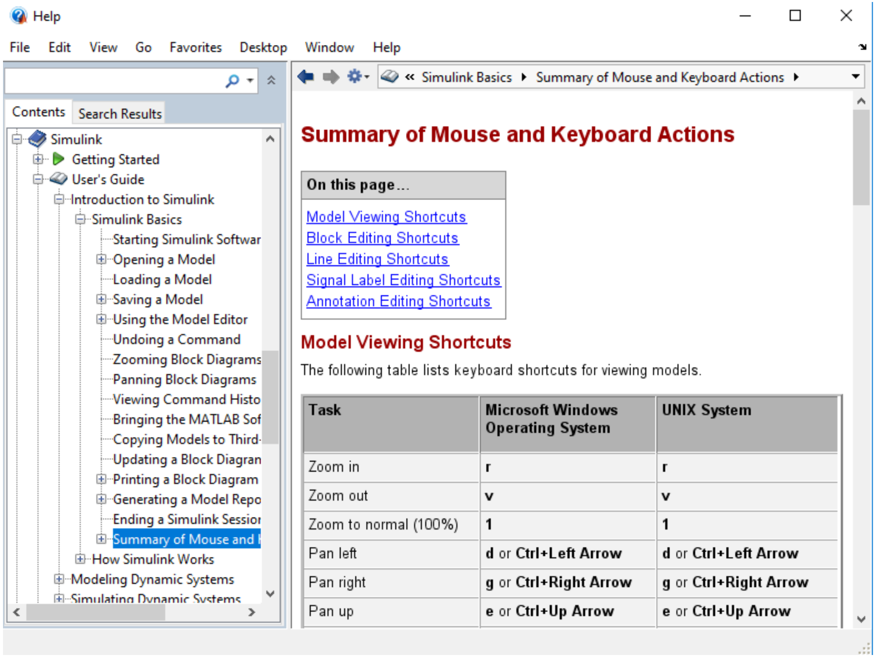

Image 3.23: SIMSCAPE's Shortcuts Window

S-Functions

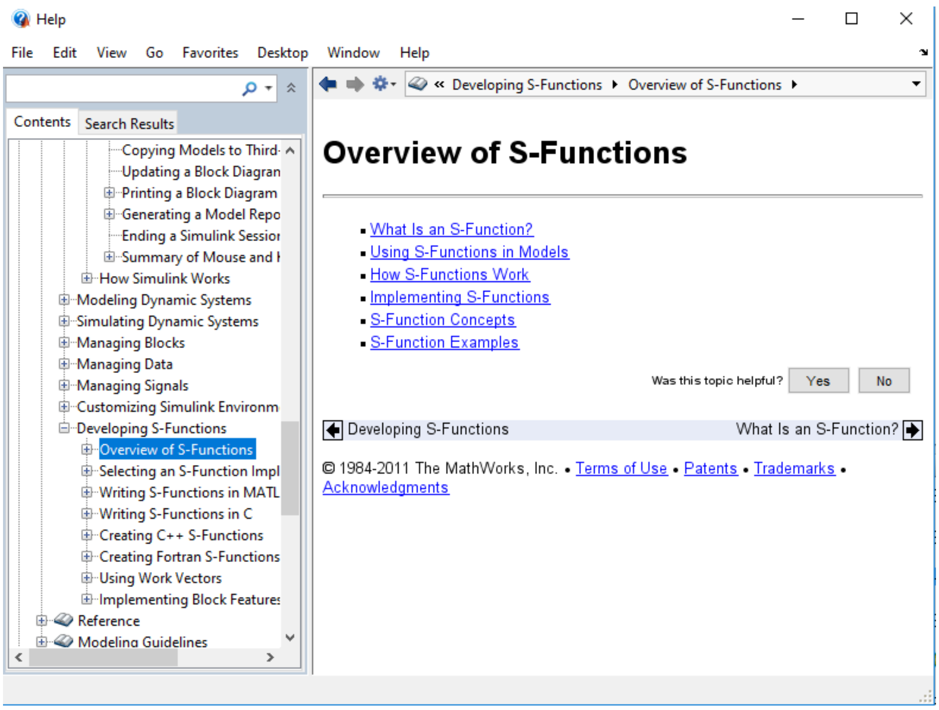

Image 3.24: SIMSCAPE's S-Functions Window

Demos

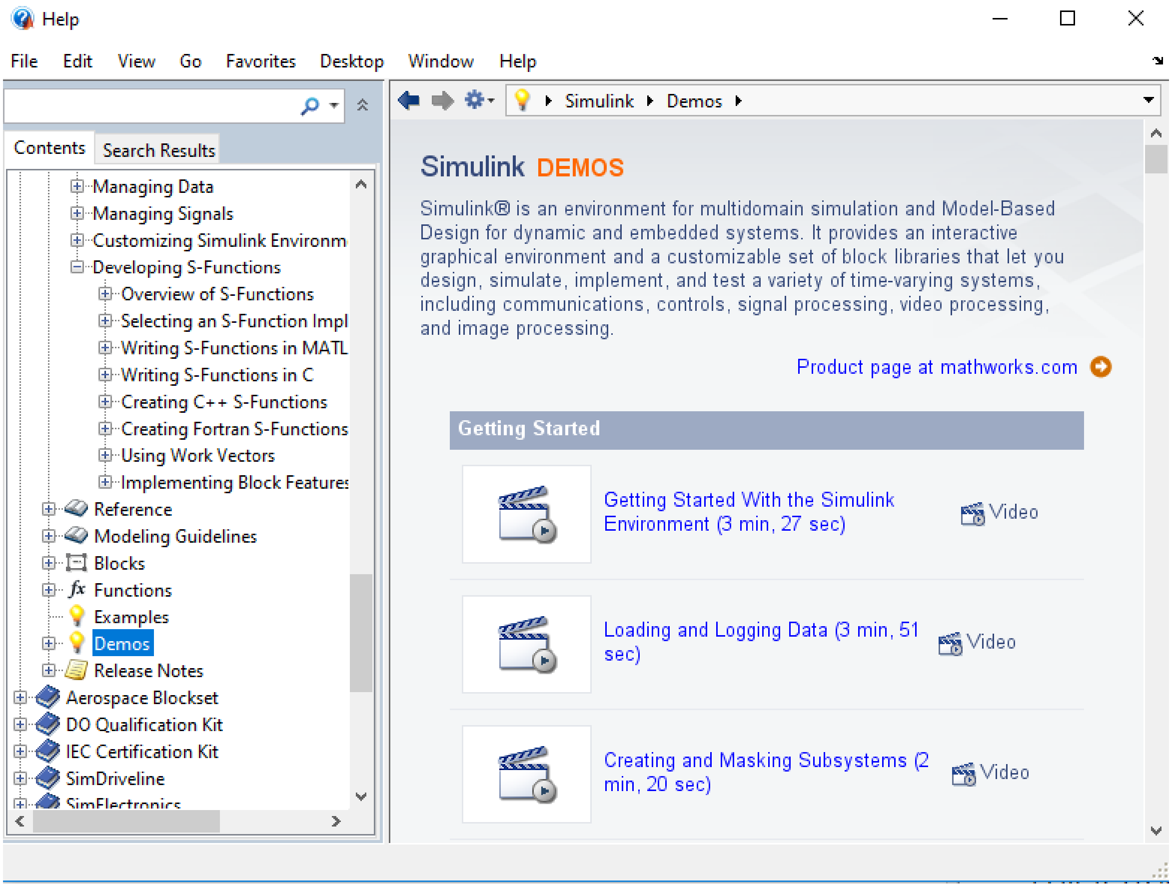

Image 3.25: SIMSCAPE's Demos Window

Terms of Use -> Shows the Software License Aggreement.
Patents… -> Shows MATLAB's Patents.
About Simulink -> Shows information about the Software.

CHAPTER 4

SIMSCAPE COMPONENTS

4.1 Foundation Library

Contains basic hydraulic, mechanical, electrical, magnetic, thermal, thermal liquid, two-phase fluid, gas, and physical signal blocks, organized into sublibraries according to technical discipline and function performed

Image 4.1: SIMSCAPE's Foundation Library

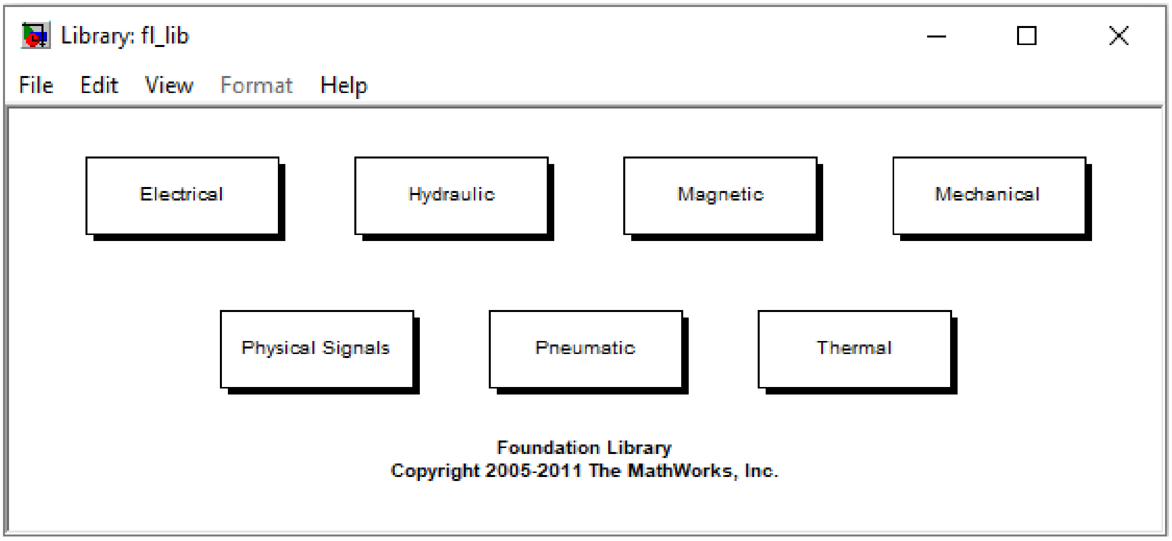

Image 4.2: SIMSCAPE's Foundation Library

4.1.1 Electrical Models

Let's take a closer look at the Electrical Models.

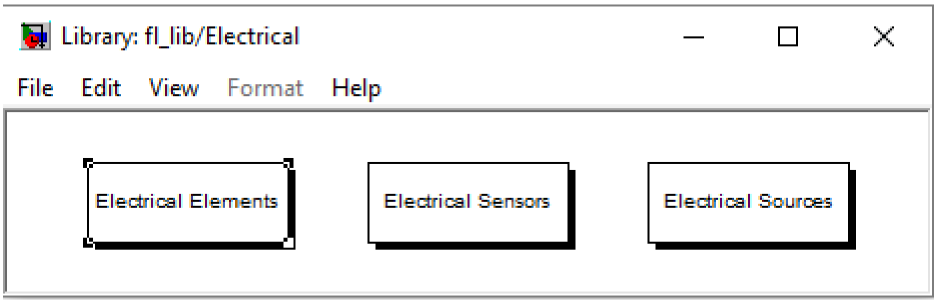

Image 4.3: SIMSCAPE's Electrical Models

Electrical Elements: Electrical building blocks, such as inductors, diodes, capacitors.

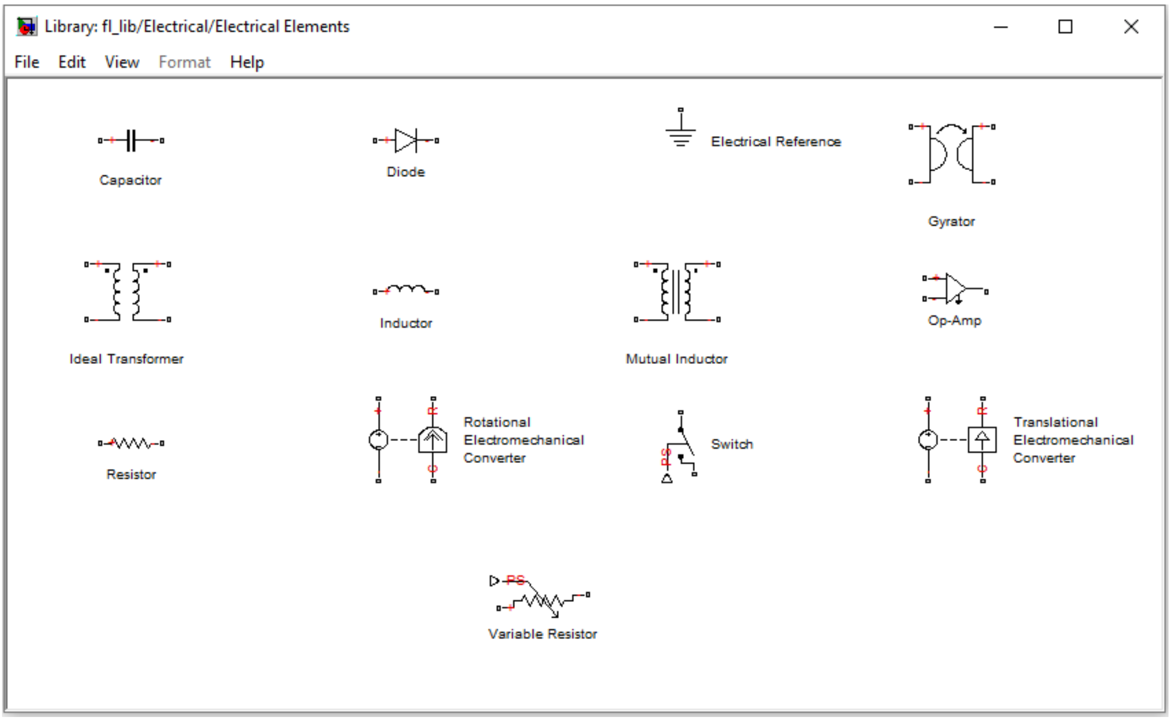

Image 4.4: SIMSCAPE's Electrical Elements

[6]

Capacitor	Linear Capacitor in electrical systems
Diode	Piecewise linear diode in electrical systems
Electrical Reference	Connection to electrical ground
Gyrator	Ideal gyrator in electrical systems
Ideal Transformer	Ideal transformer in electrical systems
Inductor	Linear inductor in electrical systems
Mutual Inductor	Mutual inductor in electrical systems
Op-Amp	Ideal operational amplifier
Resistor	Linear resistor in electrical systems
Rotational Electromechanical Converter	Interface between electrical and mechanical rotational domains
Switch	Switch controlled by external physical signal
Translational Electromechanical Converter	Interface between electrical and mechanical translational domains
Variable Resistor	Linear variable resistor in electrical systems

Array 4.1: Electrical Elements

Electrical Sensors: Current and voltage sensor blocks.

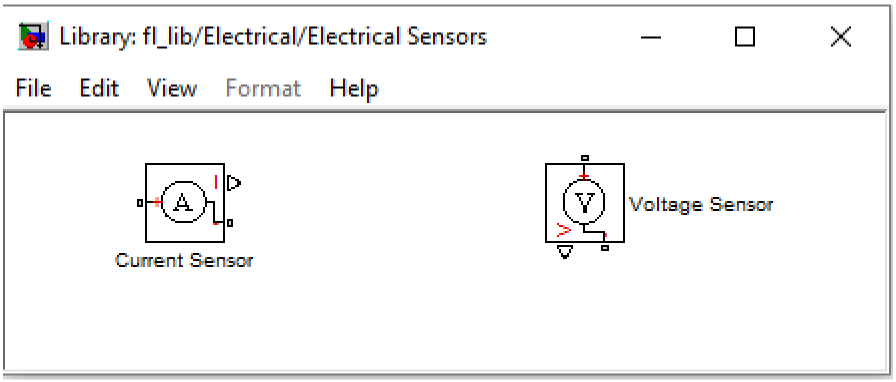

Image 4.5: SIMSCAPE's Electrical Sensors

Current Sensor	Current sensor in electrical systems
Voltage Sensor	Voltage sensor in electrical systems

Array 4.2: Electrical Sensors

Electrical Sources: Current and voltage source blocks.

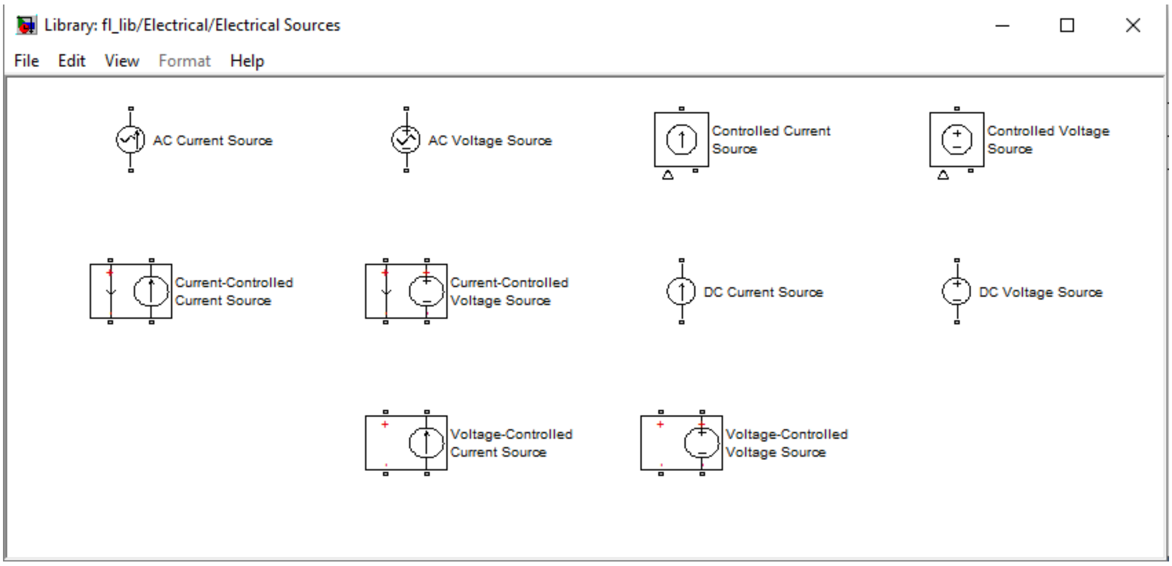

Image 4.6: SIMSCAPE's Electrical Sources

AC Current Source	Ideal sinusoidal current source
AC Voltage Source	Ideal constant voltage source
Controlled Current Source	Ideal current source driven by input signal
Controlled Voltage Source	Ideal voltage source driven by input signal
Current-Controlled Current Source	Linear current-controlled current source
Current-Controlled Voltage Source	Linear current-controlled voltage source
DC Current Source	Ideal constant current source
DC Voltage Source	Ideal constant voltage source
Voltage-Controlled Current Source	Linear voltage-controlled current source
Voltage-Controlled Voltage Source	Linear voltage-controlled voltage source

Array 4.3: Electrical Sources

4.1.2 Hydraulic Models

Let's take a closer look at the Hydraulic Models.

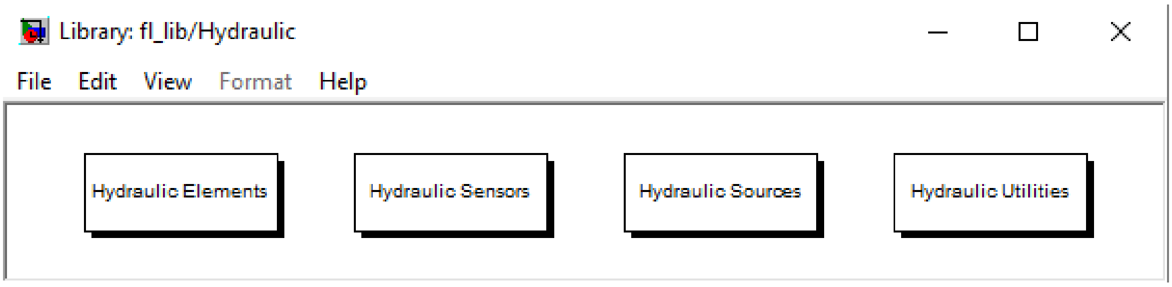

Image 4.7: SIMSCAPE's Hydraulic Models

Hydraulic Elements: Hydraulic building blocks, such as orifices, chambers, hydro-mechanical converters.

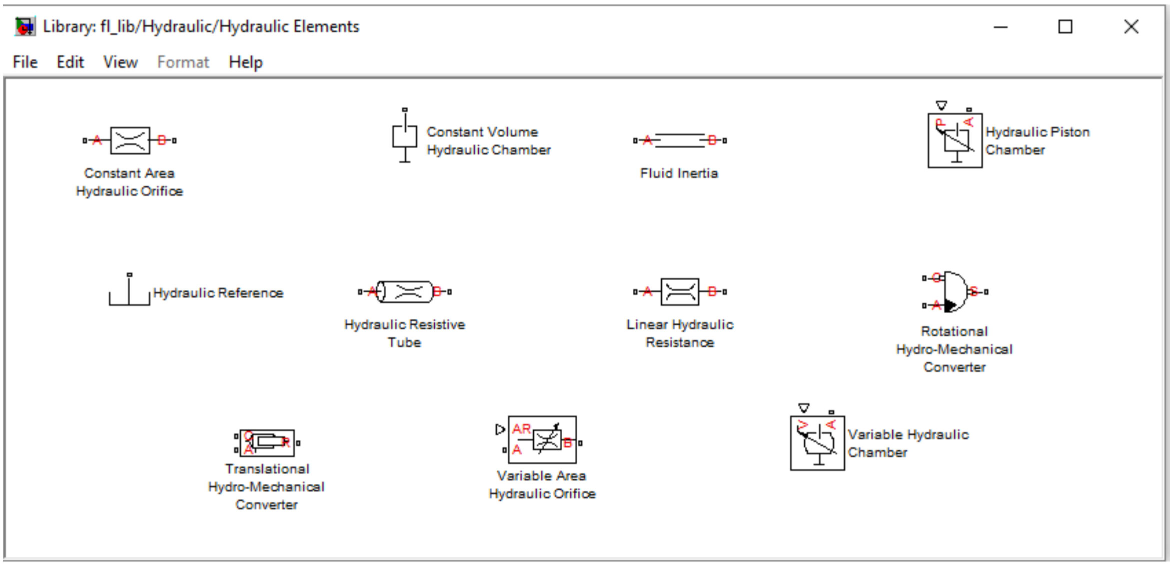

Image 4.8: SIMSCAPE's Hydraulic Elements

Constant Area Hydraulic Orifice	Hydraulic orifice with constant cross-sectional area
Constant Volume Hydraulic Chamber	Hydraulic capacity of constant volume
Fluid Inertia	Pressure differential across tube or channel due to change in fluid velocity
Hydraulic Piston Chamber	Variable volume hydraulic capacity in cylinders
Hydraulic Reference	Connection to atmospheric pressure
Hydraulic Resistive Tube	Hydraulic pipeline which accounts for friction losses only
Linear Hydraulic Resistance	Hydraulic element for setting initial pressure difference between two nodes
Rotational Hydro-Mechanical Converter	Interface between hydraulic and mechanical rotational domains
Translational Hydro-Mechanical Converter	Interface between hydraulic and mechanical translational domains
Variable Area Hydraulic Orifice	Hydraulic variable orifice created by cylindrical spool and sleeve
Variable Hydraulic Chamber	Hydraulic capacity of variable volume with compressible fluid

Array 4.4: Hydraulic Elements

Hydraulic Sensors: Hydraulic flow rate and pressure sensor blocks.

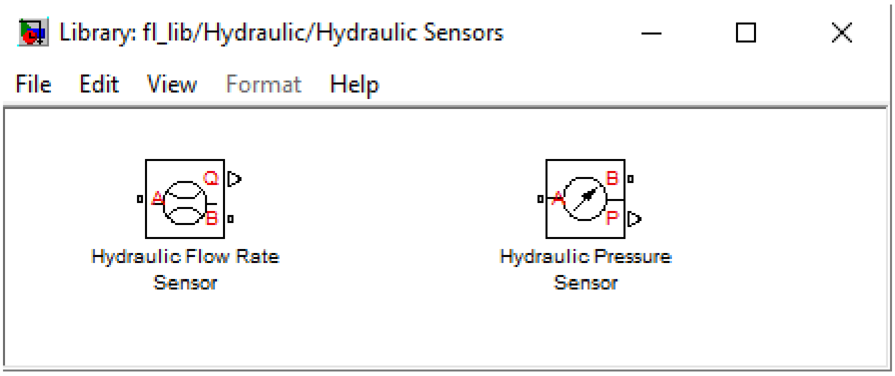

Image 4.9: SIMSCAPE's Hydraulic Sensors

Hydraulic Flow Rate Sensor	Ideal flow meter
Hydraulic Pressure Sensor	Ideal pressure sensing device

Array 4.5: Hydraulic Sensors

Hydraulic Sources: Hydraulic flow rate and pressure source blocks.

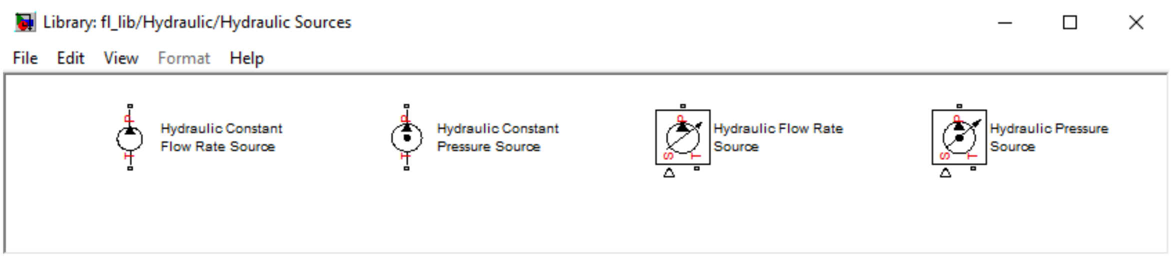

Image 4.10: SIMSCAPE's Hydraulic Sources

Hydraulic Constant Flow Rate Source	Ideal source of hydraulic energy, characterized by constant flow rate
Hydraulic Constant Pressure Source	Ideal source of hydraulic energy, characterized by constant pressure
Hydraulic Flow Rate Source	Ideal source of hydraulic energy, characterized by flow rate
Hydraulic Pressure Source	Ideal source of hydraulic energy, characterized by pressure

Array 4.6: Hydraulic Sources

Hydraulic Utilities: Basic hydraulic environment blocks, such as custom hydraulic fluid.

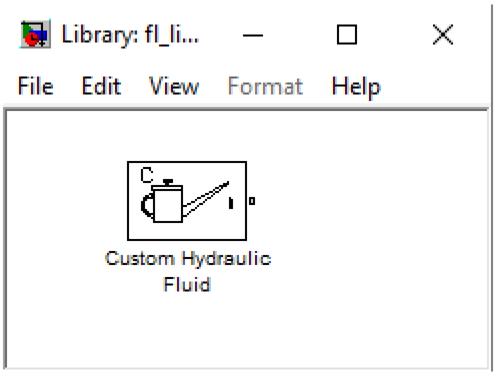

Image 4.11: SIMSCAPE's Hydraulic Utilities

Custom Hydraulic Fluid	Working fluid properties, set by specifying parameter values

Array 4.7: Hydraulic Utilities

4.1.3 Magnetic Models

Let's take a closer look at the Magnetic Models.

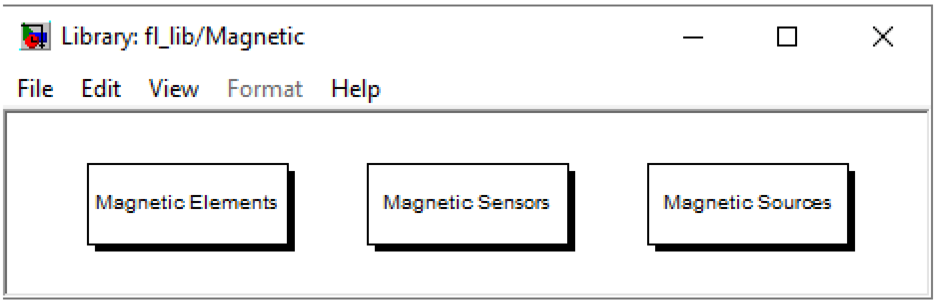

Image 4.12: SIMSCAPE's Magnetic Models

Magnetic Elements: Magnetic building blocks, such as reluctances, electromagnetic converters, actuators.

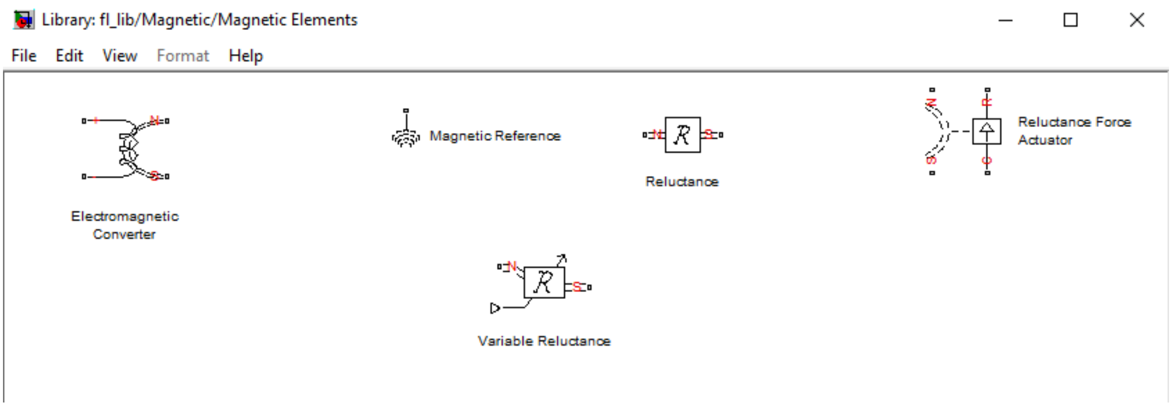

Image 4.13: SIMSCAPE's Magnetic Elements

Electromagnetic Converter	Lossless electromagnetic energy conversion device
Magnetic Reference	Reference connection for magnetic ports
Reluctance	Magnetic reluctance
Reluctance Force Actuator	Magnetomotive device based on reluctance force
Variable Reluctance	Variable reluctance in electromagnetic systems

Array 4.8: Magnetic Elements

Magnetic Sensors: Flux and mmf sensor blocks.

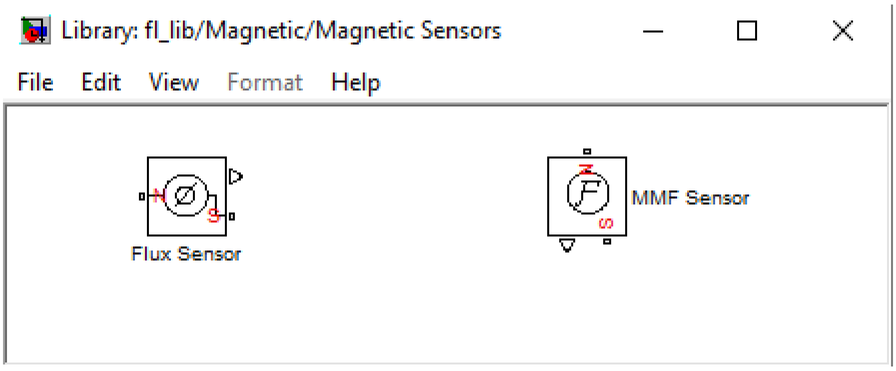

Image 4.14: SIMSCAPE's Magnetic Sensors

Flux Sensor	**Ideal flux sensor**
MMF Sensor	**Ideal magnetomotive force sensor**

Array 4.9: Magnetic Sensors

Magnetic Sources: Flux and mmf source blocks.

Image 4.15: SIMSCAPE's Magnetic Sensors

Controlled Flux Source	Ideal flux source driven by input signal
Controlled MMF Source	Ideal magnetomotive force source driven by input signal
Flux Source	Ideal flux source
MMF Source	Ideal magnetomotive force source

Array 4.10: Magnetic Sources

4.1.4 Mechanical Models

Let's take a closer look at the Mechanical Models.

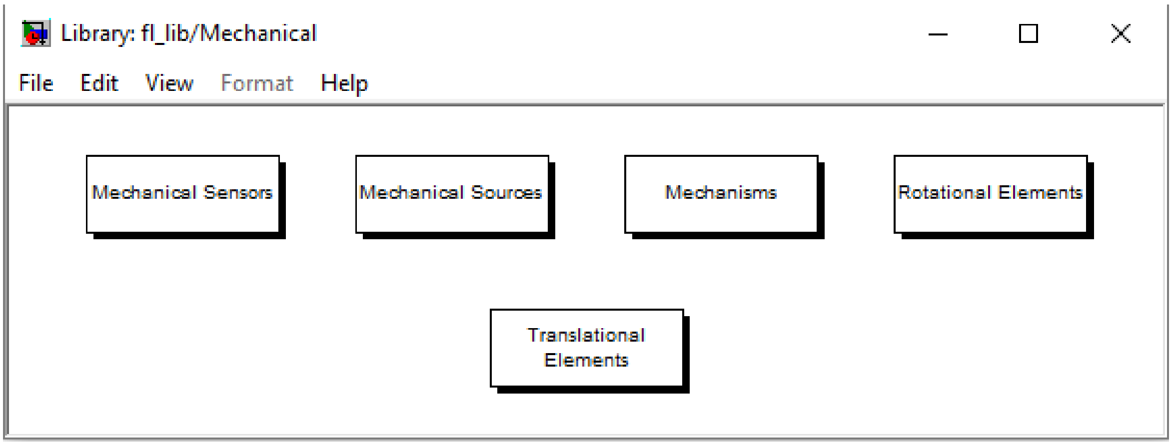

Image 4.16: SIMSCAPE's Mechanical Models

Mechanical Sensors: Rotational and translational motion, torque, force sensor blocks

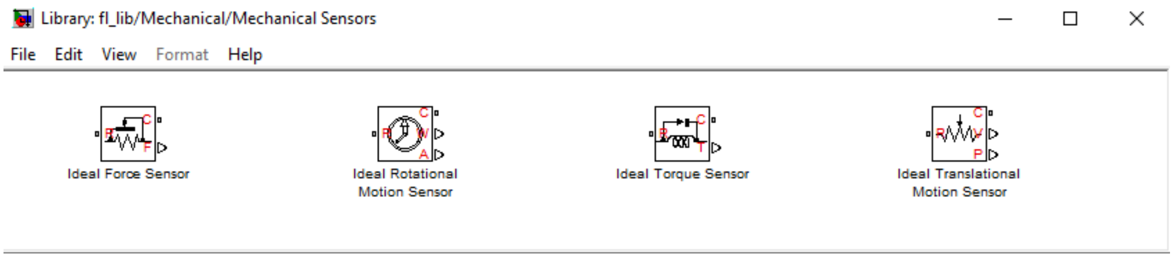

Image 4.17: SIMSCAPE's Mechanical Sensors

Ideal Force Sensor	Force sensor in mechanical translational systems
Ideal Rotational Motion Sensor	Motion sensor in mechanical rotational systems
Ideal Torque Sensor	Torque sensor in mechanical rotational systems
Ideal Translational Motion Sensor	Motion sensor in mechanical translational systems

Array 4.11: Mechanical Sensors

Mechanical Sources: Angular and translational velocity, torque, force source blocks.

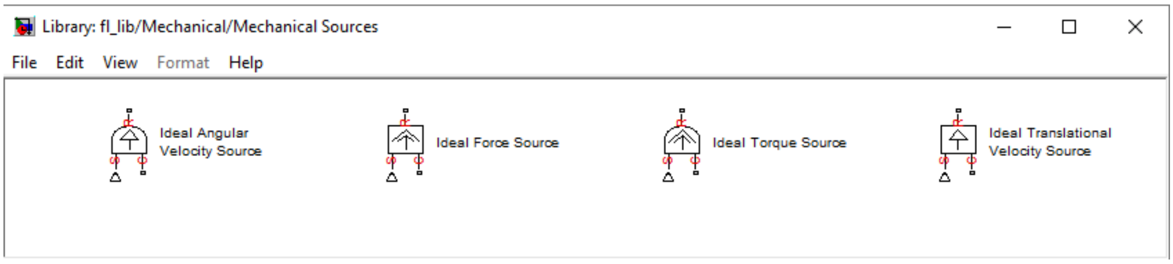

Image 4.18: SIMSCAPE's Mechanical Sources

Ideal Angular Velocity Source	Ideal angular velocity source in mechanical rotational systems
Ideal Force Source	Ideal source of mechanical energy that generates force proportional to the input signal
Ideal Torque Source	Ideal source of mechanical energy that generates torque proportional to the input signal
Ideal Translational Velocity Source	Ideal velocity source in mechanical translational systems

Array 4.12: Mechanical Sources

Mechanisms: Simple mechanisms, such as gear box, lever, wheel and axle.

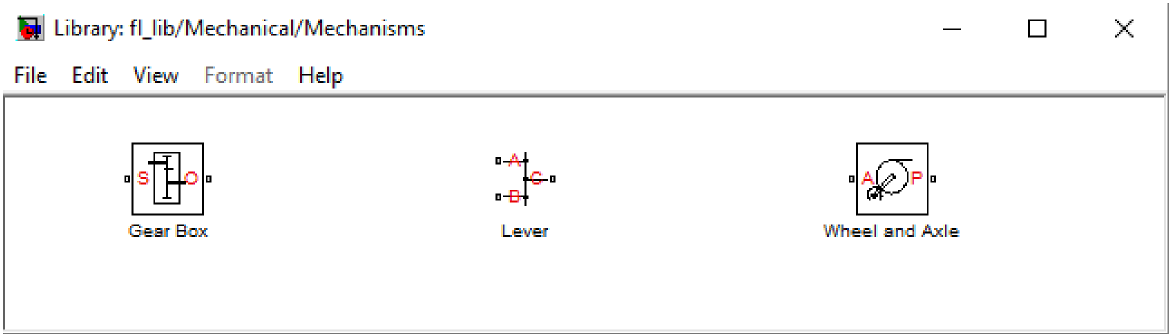

Image 4.19: SIMSCAPE's Mechanisms

Gear Box	Gear box in mechanical systems
Lever	Generic mechanical lever
Wheel and Axle	Wheel and axle mechanism in mechanical systems

Array 4.13: Mechanisms

Rotational Elements: Mechanical building blocks for rotational motion.

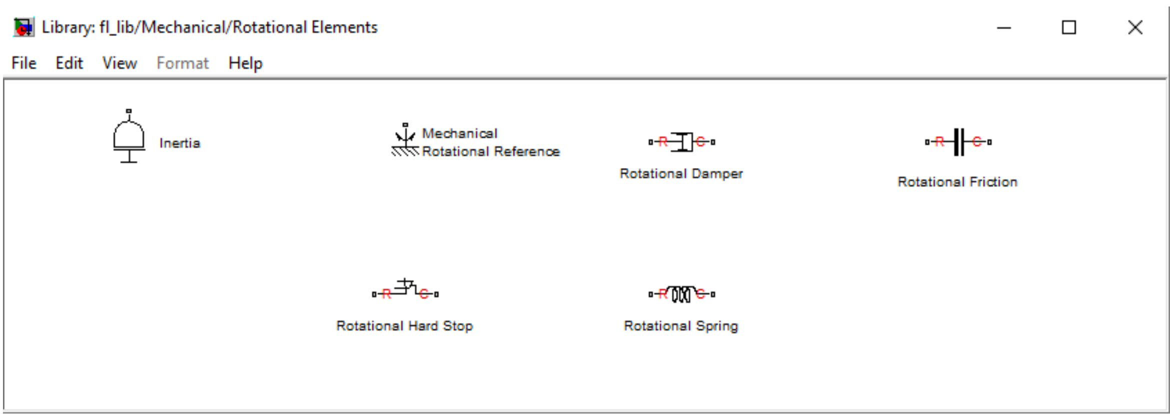

Image 4.20: SIMSCAPE's Rotational Elements

Inertia	Ideal mechanical rotational inertia
Mechanical Rotational Reference	Reference connection for mechanical rotational ports
Rotational Damper	Viscous damper in mechanical rotational systems
Rotational Friction	Friction in contact between rotating bodies
Rotational Hard Stop	Double-sided rotational hard stop
Rotational Spring	Ideal spring in mechanical rotational systems

Array 4.14: Rotational Elements

Translational Elements: Mechanical building blocks for translational motion.

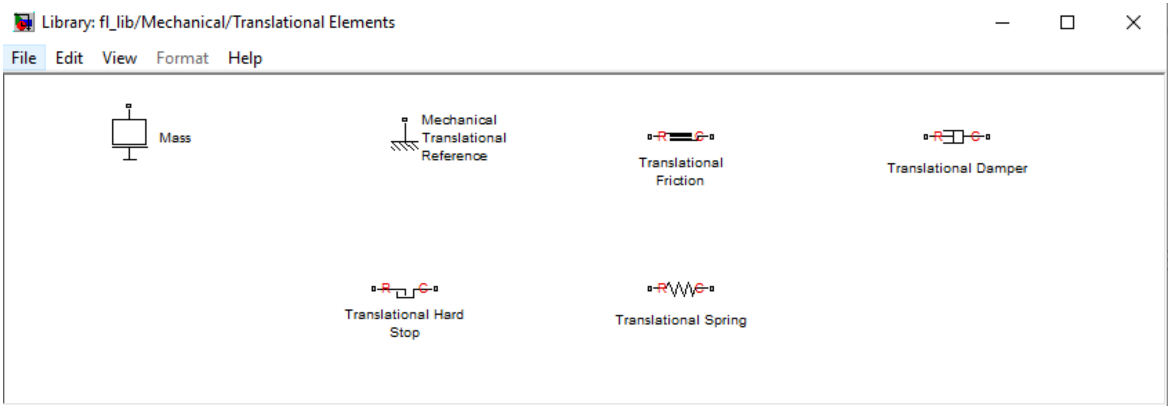

Image 4.21: SIMSCAPE's Translational Elements

Mass	Ideal mechanical translational mass
Mechanical Translational Reference	Reference connection for mechanical translational ports
Translational Friction	Friction in contact between moving bodies
Translational Damper	Viscous damper in mechanical translational systems
Translational Hard Stop	Double-sided translational hard stop
Translational Spring	Ideal spring in mechanical translational systems

Array 4.15: Translational Elements

4.1.5 Physical Signals Models

Let's take a closer look at the Signals Models.

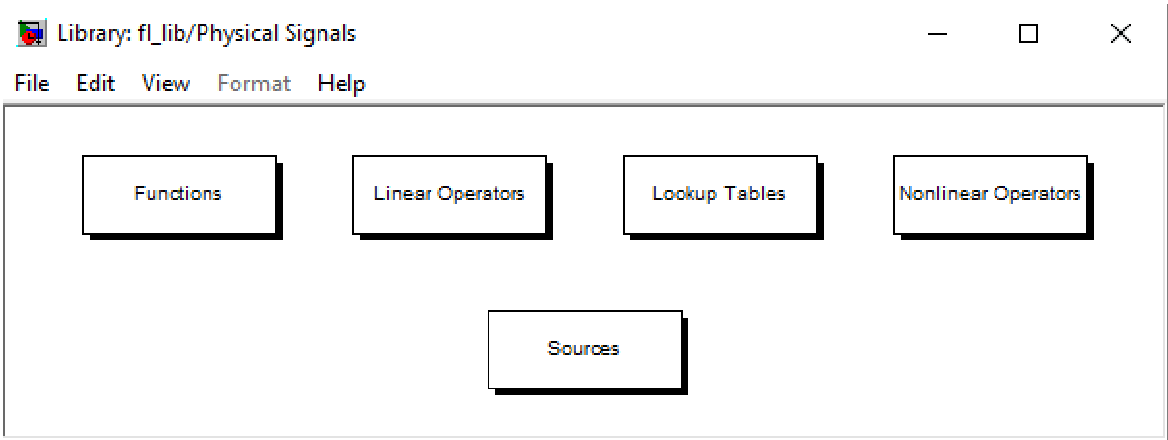

Image 4.22: SIMSCAPE's Physical Signals

Functions: Blocks that perform math operations on physical signals

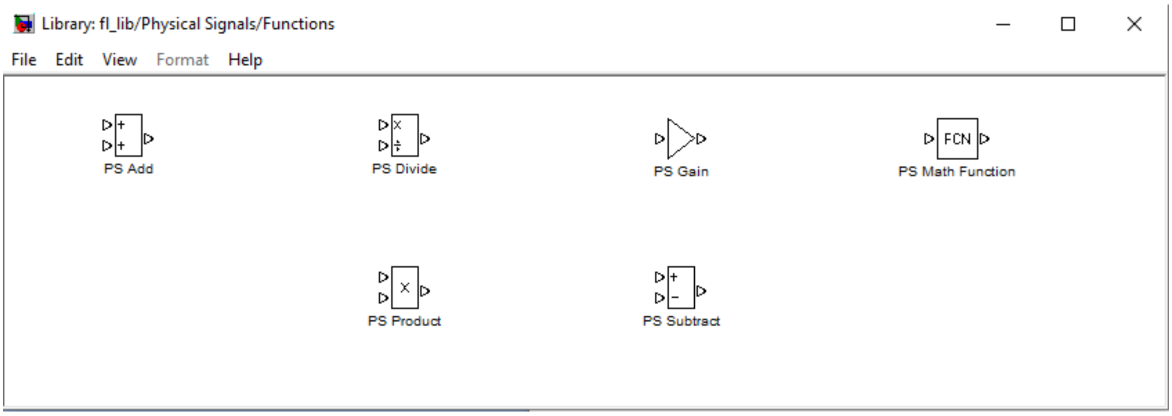

Image 4.23: SIMSCAPE's Functions(Physical Signals)

PS Add	Add two physical signal inputs
PS Divide	Compute simple division of two input physical signals
PS Gain	Multiply input physical signal by constant
PS Math Function	Apply mathematical function to input physical signal
PS Product	Multiply two physical signal inputs
PS Subtract	Compute simple subtraction of two input psysical signals

Array 4.16: Functions(Physical Signals)

Linear Operators: Blocks that simulate continuous-time functions for physical signals.

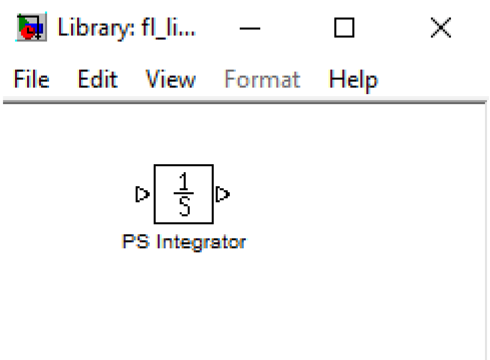

Image 4.24: SIMSCAPE's Linear Operators

PS Integrator	Integrate physical signal

Array 4.17: Linear Operators

Lookup Tables: Blocks that perform table lookup to generate physical signals

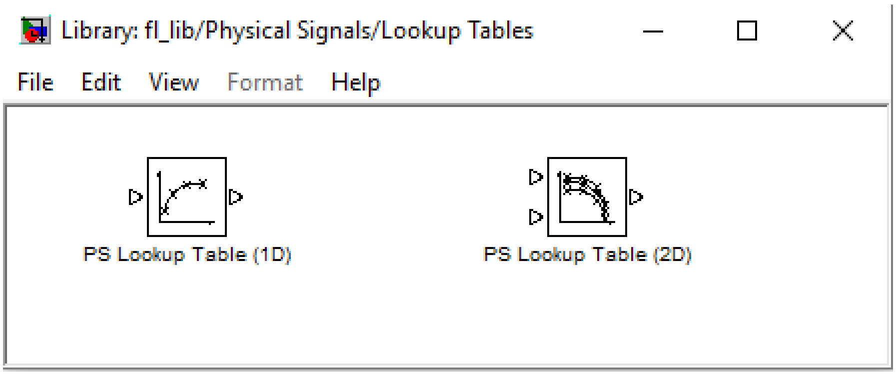

Image 4.25: SIMSCAPE's Lookup Tables

PS Lookup Table(1D)	Approximate one-dimensional function using specified lookup method
PS Lookup Table(2D)	Approximate two-dimensional function using specified lookup method

Array 4.18: Lookup Tables

Nonlinear Operators: Blocks that simulate discontinuities, such as saturation or dead zone, for physical signals

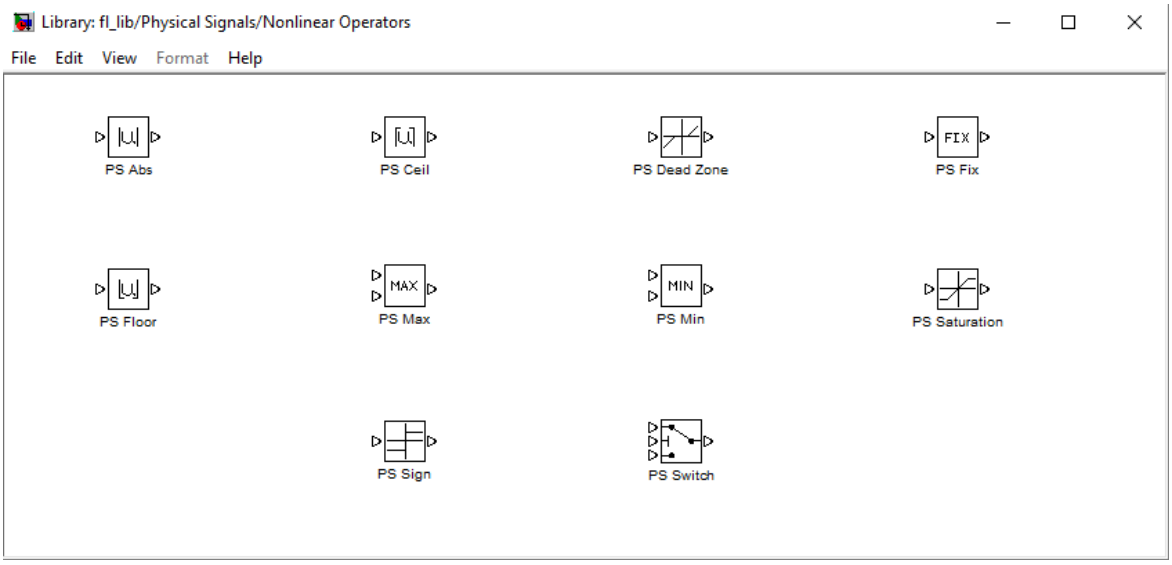

Image 4.26: SIMSCAPE's Nonlinear Operators(Physical Signals)

PS Abs	Output absolute value of input physical signal
PS Ceil	Output the smallest integer larger than or equal to input physical signal
PS Dead Zone	Provide region of zero output for physical signals
PS Fix	Round input physical signal toward zero
PS Floor	Output the largest integer smaller than or equal to input physical signal
PS Max	Output maximum of two input physical signals
PS Min	Output minimum of two input physical signals
PS Saturation	Limit range of physical signal
PS Sign	Output sign of input physical signal
PS Switch	Single-pole double-throw switch controlled by external physical signal

Array 4.19: Nonlinear Operators(Physical Signals)

Sources: Blocks that simulate physical signal sources.

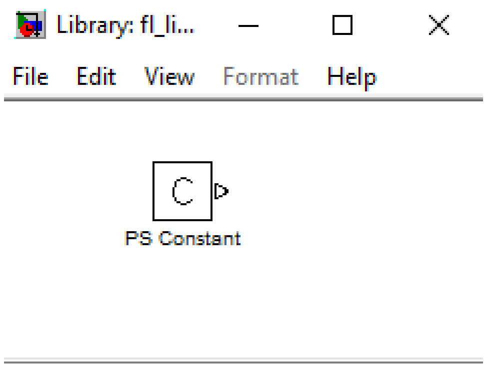

Image 4.27: SIMSCAPE's Sources(Physical Signals)

PS Constant	Generate constant physical signal

Array 4.20: Sources(Physical Signals)

4.1.6 Pneumatic Models

Let's take a closer look at the Pneumatic Models.

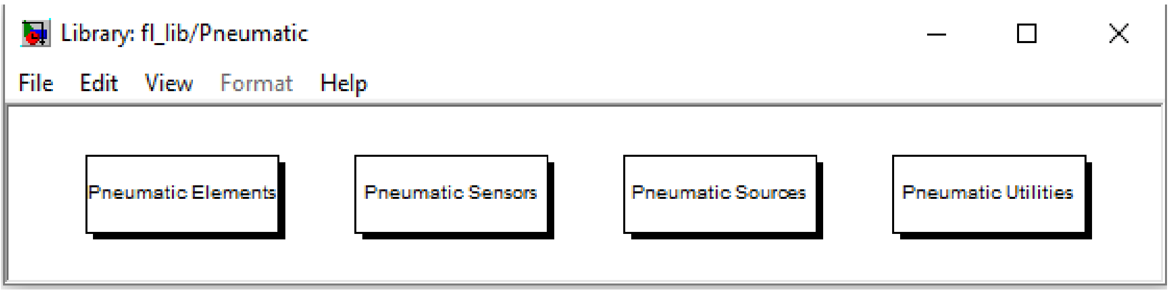

Image 4.28: SIMSCAPE's Pneumatic Models

Pneumatic Elements: Pneumatic building blocks.

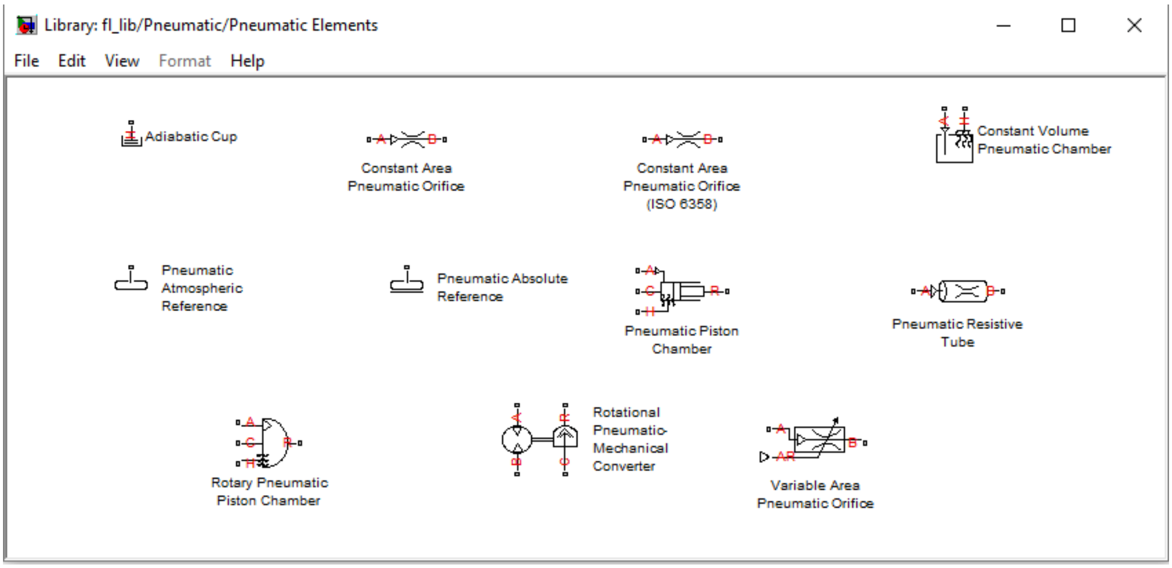

Image 4.29: SIMSCAPE's Pneumatic Elements

Adiabatic Cup	Thermal element with no thermal mass and perfect insulation
Constant Area Pneumatic Orifice	Sharp-edged orifice in pneumatic systems
Constant Area Pneumatic Orifice(ISO 6358)	Fixed-area pneumatic orifice complying with ISO 6358 standard
Constant Volume Pneumatic Chamber	Constant volume pneumatic chamber based on ideal gas law
Pneumatic Atmospheric Reference	Reference connection to ambient pressure and temperature for pneumatic ports
Pneumatic Absolute Reference	Reference connection to zero absolute pressure and temperature for pneumatic ports
Pneumatic Piston Chamber	Translational pneumatic piston chamber based on ideal gas law
Pneumatic Resistive Tube	Pneumatic pipe accounting for pressure loss and added heat due to flow

	resistance
Rotary Pneumatic Chamber	Rotational pneumatic piston chamber based on ideal gas law
Rotational Pneumatic-Mechanical Converter	Interface between pneumatic and mechanical rotational domains
Variable Area Pneumatic Orifice	Sharp-edged variable-area orifice in pneumatic systems

Array 4.21: Pneumatic Elements

Pneumatic Sensors: Pneumatic sensor blocks.

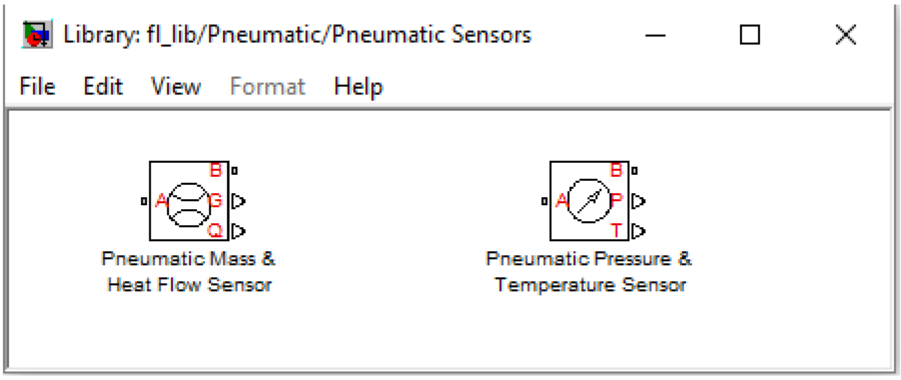

Image 4.30: SIMSCAPE's Pneumatic Sensors

Pneumatic Mass & Heat Flow Sensor	Ideal mass flow and heat flow sensor
Pneumatic Pressure & Temperature Sensor	Ideal pressure and temperature sensor

Array 4.22: Pneumatic Sensors

Pneumatic Sources: Pneumatic pressure and flow rate sources.

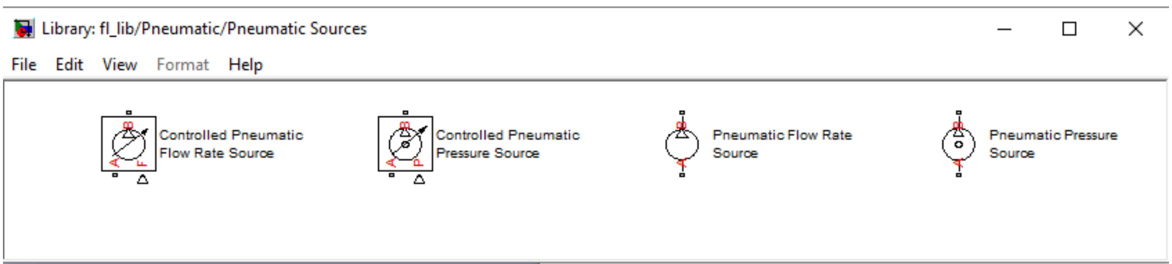

Image 4.31: SIMSCAPE's Pneumatic Sources

Controlled Pneumatic Flow Rate Source	Ideal compressor with signal-controlled mass flow rate
Controlled Pneumatic Pressure Source	Ideal compressor with signal-controlled pressure difference
Pneumatic Flow Rate Source	Ideal compressor with constant mass flow rate
Pneumatic Pressure Source	Ideal compressor with constant pressure difference

Array 4.23: Pneumatic Sources

Pneumatic Utilities: Basic pneumatic environment blocks, such as Gas properties.

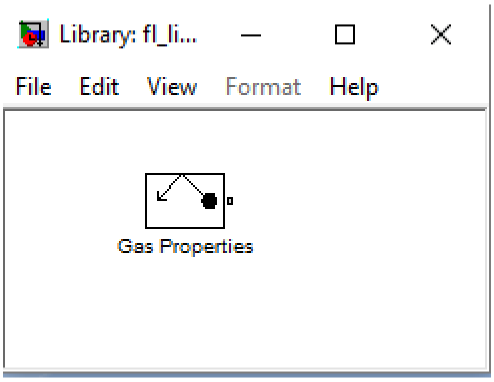

Image 4.32: SIMSCAPE's Pneumatic Utilities

Gas Properties	Global gas properties for attached circuit

Array 4.24: Pneumatic Utilities

4.1.7 Thermal Models

Let's take a closer look at the Thermal Models.

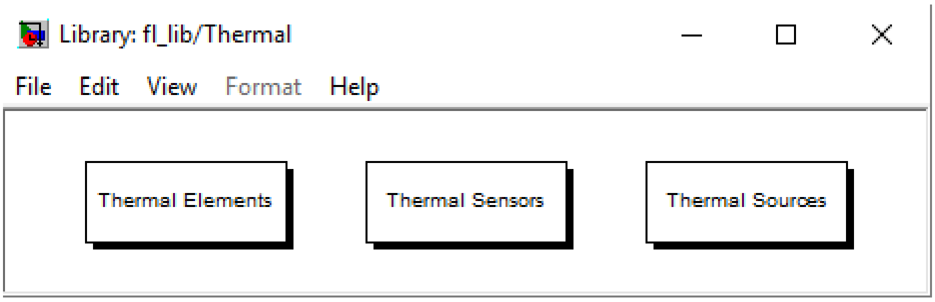

Image 4.33: SIMSCAPE's Thermal Models

Thermal Elements: Thermal building blocks, such as thermal mass, various heat transfer blocks.

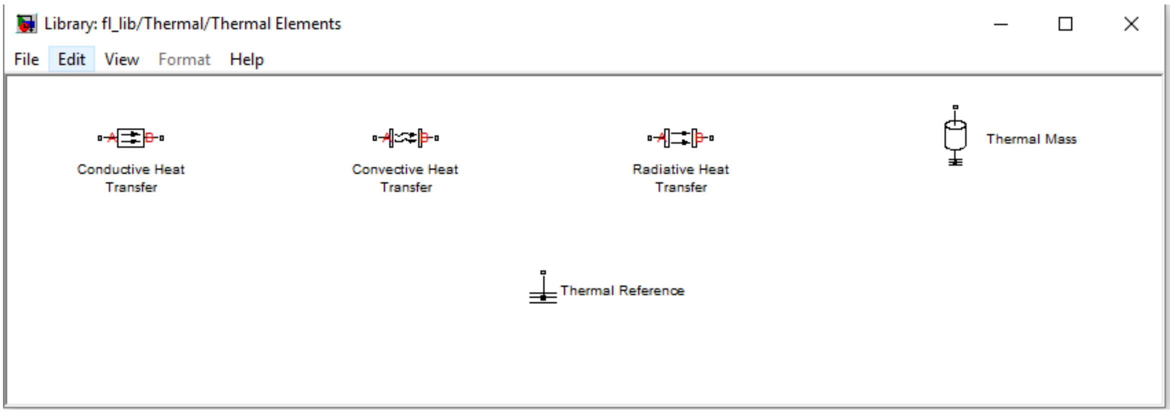

Image 4.34: SIMSCAPE's Thermal Elements

Conductive Heat Transfer	Heat transfer by conduction
Convective Heat Transfer	Heat transfer by convection
Radiative Heat Transfer	Heat transfer by radiation
Thermal Mass	Mass in thermal systems
Thermal Reference	Reference connection for thermal ports

Array 4.25: Thermal Elements

Thermal Sensors: Temperature and heat flow rate sensor blocks.

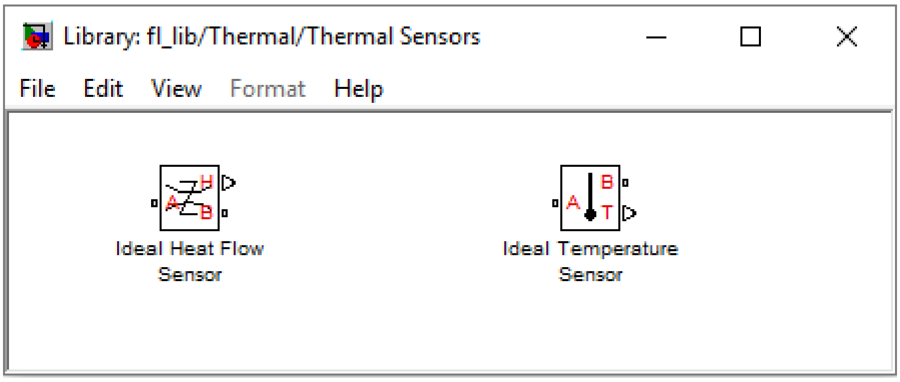

Image 4.35: SIMSCAPE's Thermal Sensors

Ideal Heat Flow Sensor	Ideal heat flow meter
Ideal Temperature Sensor	Ideal temperature sensor

Array 4.26: Thermal Sensors

Thermal Sources: Temperature and heat flow rate source blocks.

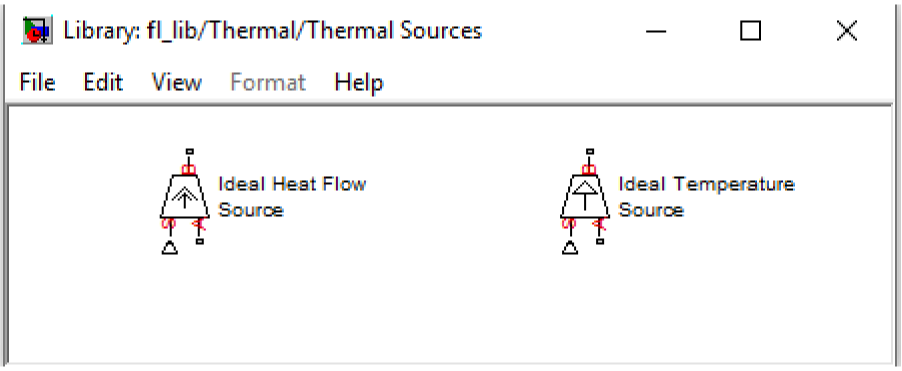

Image 4.36: SIMSCAPE's Thermal Sources

Ideal Heat Flow Source	Constant source of thermal energy, characterized by heat flow
Ideal Temperature Source	Constant source of thermal energy, characterized by temperature

Array 4.27: Thermal Sources

4.2 SimDriveline

Driveline modeling employs a physical network approach, where Simscape blocks correspond to physical elements, such as clutches, gears, and transmissions. The lines that connect these blocks correspond to the physical connections that transmit power. The resulting models let you describe the physical structure of a system, rather than the underlying mathematics.

Image 4.37: SIMSCAPE's SimDriveline

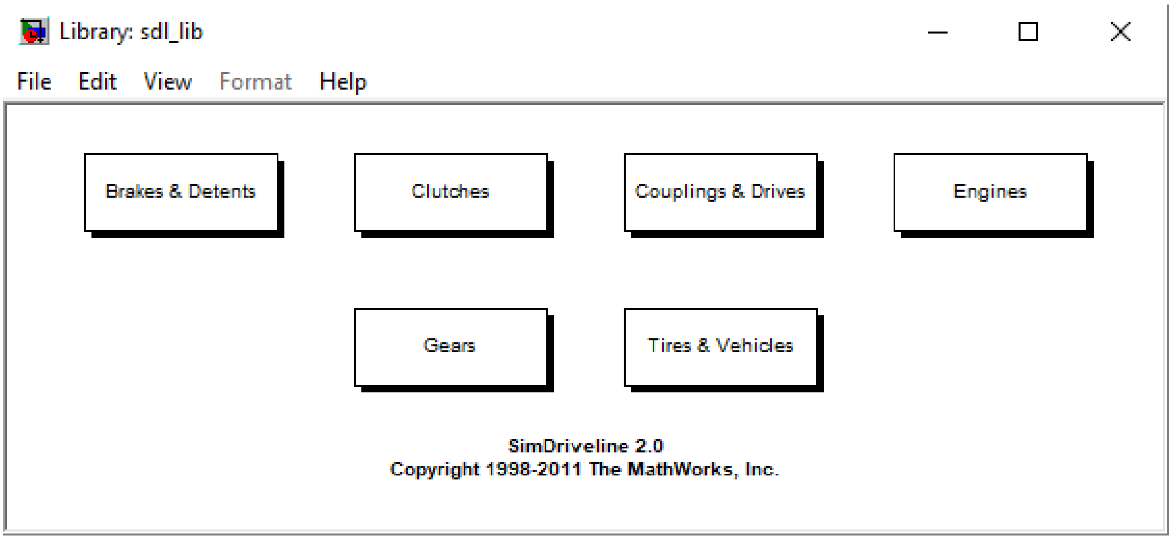

Image 4.38: SIMSCAPE's Foundation Library

4.2.1 Brakes & Detents

Let's take a closer look at the Brakes & Detents. There are two categories, rotational and translational.

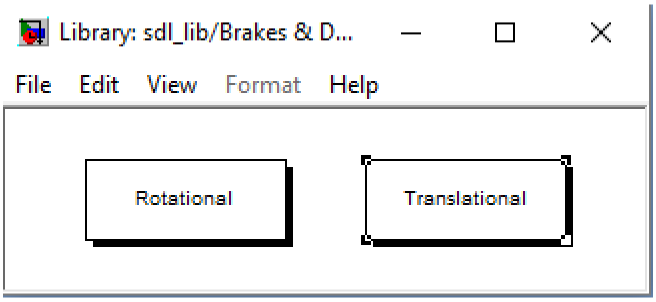

Image 4.39: SIMSCAPE's SimDriveline Brakes & Detents

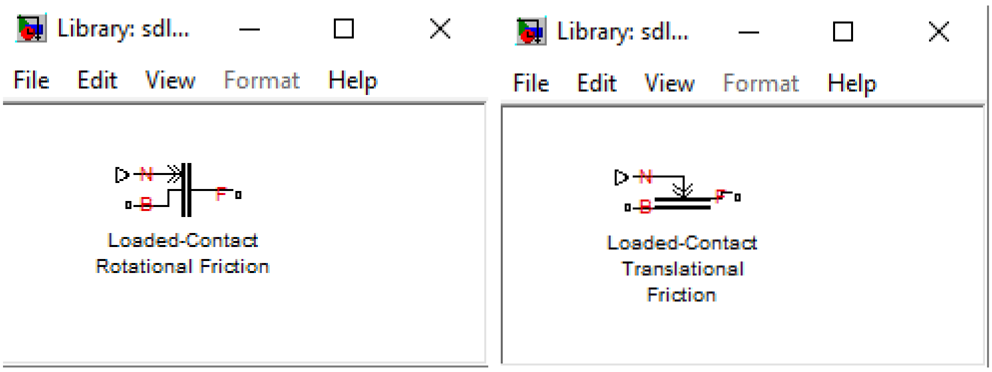

Image 4.40: SIMSCAPE's SimDriveline Brakes & Detents, rotational and translational Friction

Loaded-Contact Rotational Friction	Loaded-contact friction between two sliding bodies
Loaded-Contact Translational Friction	Double-sided, spring-loaded translational detent

Array 4.28: Thermal Sources

4.2.2 Clutches

Let's take a closer look at the Clutches.

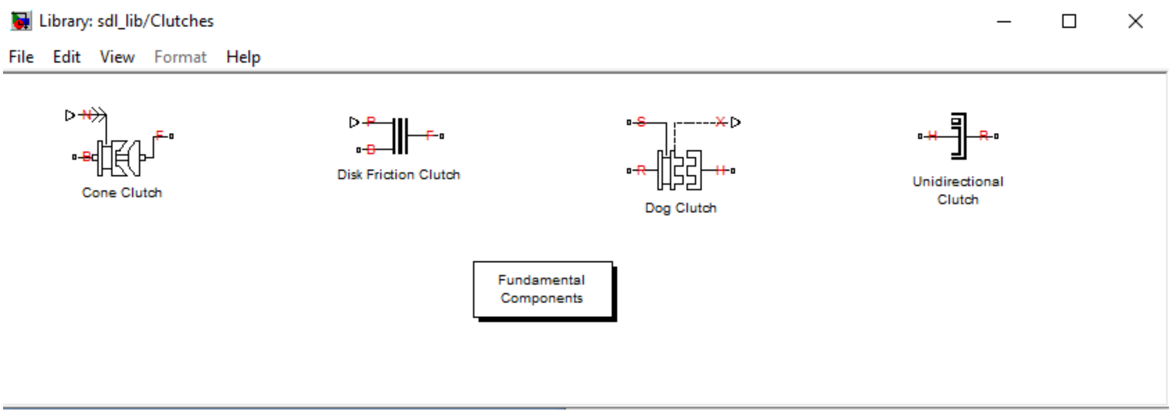

Image 4.41: SIMSCAPE's SimDriveline Clutches

Cone Clutch	Friction clutch with conical plates that engage when normal force exceeds threshold
Disk Friction Clutch	Friction clutch with disk plates that engage when plate pressure exceeds threshold
Dog Clutch	Clutch with toothed plates that engage when plate teeth become enmeshed
Unidirectional Clutch	Clutch that transmits power in a single direction

Array 4.29: SimDriveline Clutches

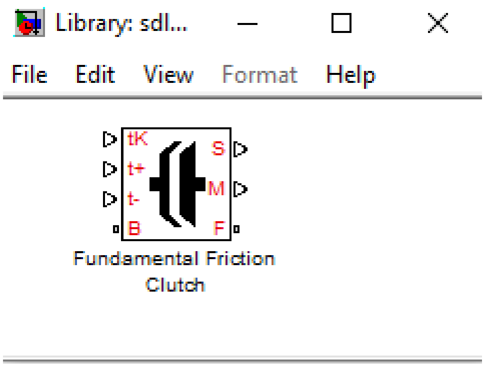

Image 4.42: SIMSCAPE's SimDriveline Clutches Fundamental Component

Fundamental Friction Clutch	Friction clutch with kinetic and static-limit friction torques as inputs

Array 4.30: SimDriveline Fundamental Friction Clutch

4.2.3 Couplings & Drives

Let's take a closer look at the Couplings & Drives.

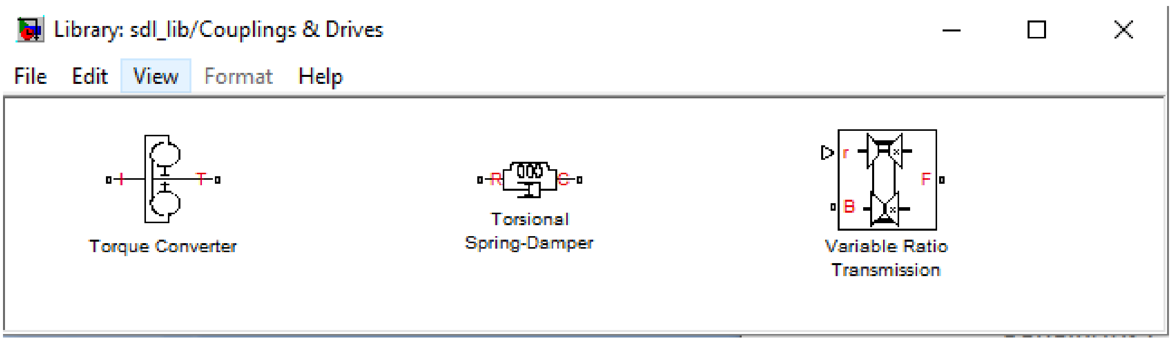

Image 4.43: SIMSCAPE's SimDriveline Couplings & Drives

Torque Converter	Viscous fluid coupling between rotating driveline shafts
Torsional Spring-Damper	Rotational spring and damper coupling, with Coulomb friction, locking, and hard stops
Variable Ratio Transmission	Dynamic gearbox with variable and controllable gear ratio, transmission compliance, and friction losses

Array 4.31: SimDriveline Couplings & Drives

4.2.4 Engines

Let's take a closer look at the Engines.

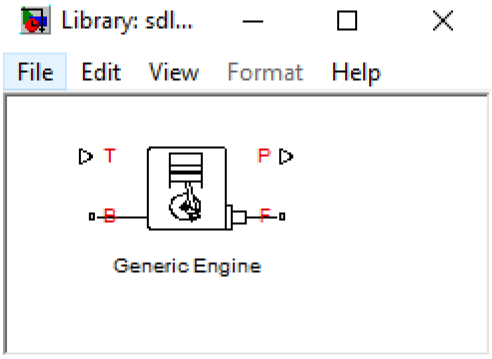

Image 4.44: SIMSCAPE's SimDriveline Engines

Generic Engine	Internal combustion engine with throttle and rotational inertia and time lag

Array 4.32: SimDriveline Engines

4.2.5 Gears

Let's take a closer look at the Gears.

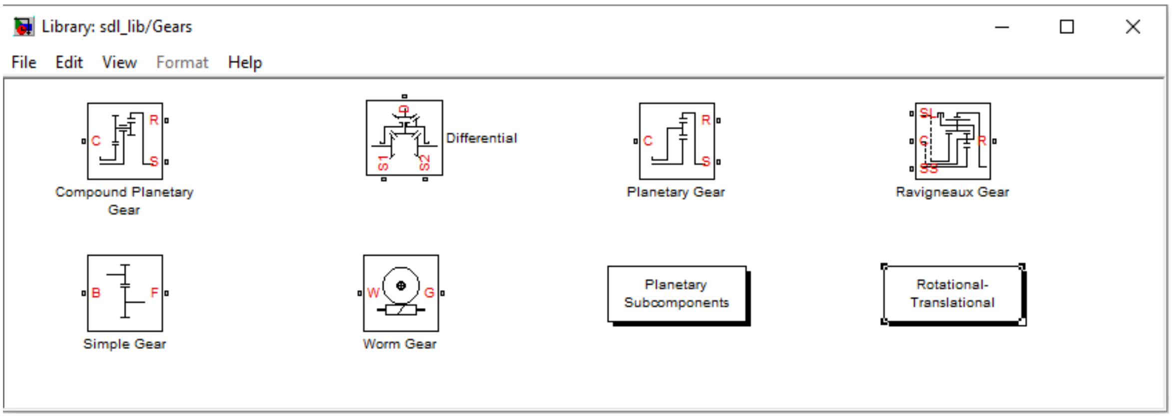

Image 4.45: SIMSCAPE's SimDriveline Gears

Compound Planetary Gear	Planetary gear train with stepped planet gear set
Differential	Gear mechanism that allows driven shafts to spin at different speeds
Planetary Gear	Gear train with sun, planet, and ring gears
Ravigneaux Gear	Planetary gear with two sun gears and two planet gear sets
Simple Gear	Simple gear of base and follower wheels with adjustable gear ratio and friction losses
Worm Gear	Worm gear with adjustable gear ratio and friction losses

Array 4.33: SimDriveline Gears

Planetary Subcomponents

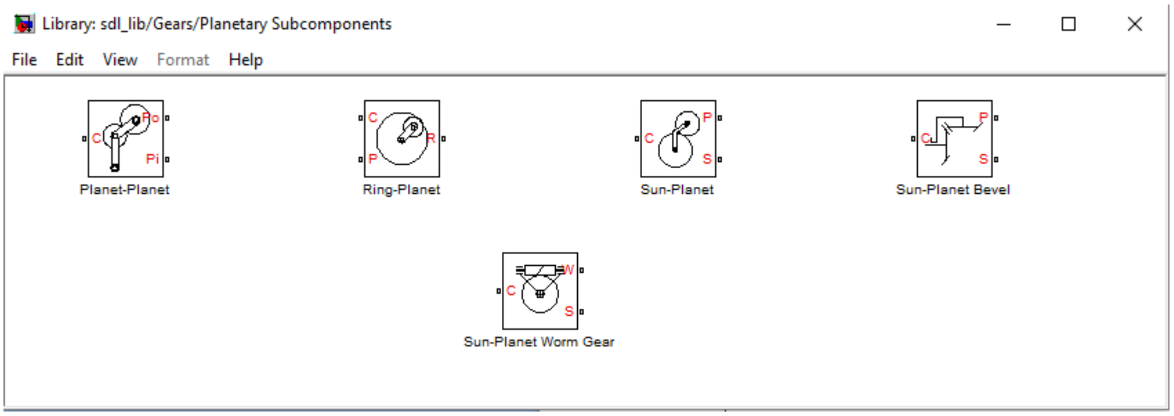

Image 4.46: SIMSCAPE's SimDriveline Planetary Subcomponents(Gear)

Planet-Planet	Planetary gear set of carrier, inner planet, and outer planet wheels with adjustable gear ratio and friction losses
Ring-Planet	Planetary gear set of carrier, planet, and ring wheels with adjustable gear ratio and friction losses
Sun-Planet	Planetary gear set of carrier, planet, and sun wheels with adjustable gear ratio and friction losses
Sun-Planet Bevel	Planetary gear set of carrier, beveled planet, and sun wheels with adjustable gear ratio, assembly orientation, and friction losses
Sun-Planet Worm Gear	Planetary gear set of carrier, worm planet, and sun wheels with adjustable gear ratio, worm thread type, and friction losses

Array 4.34: SimDriveline Planetary Subcomponents(Gear)

Rotational-Translational

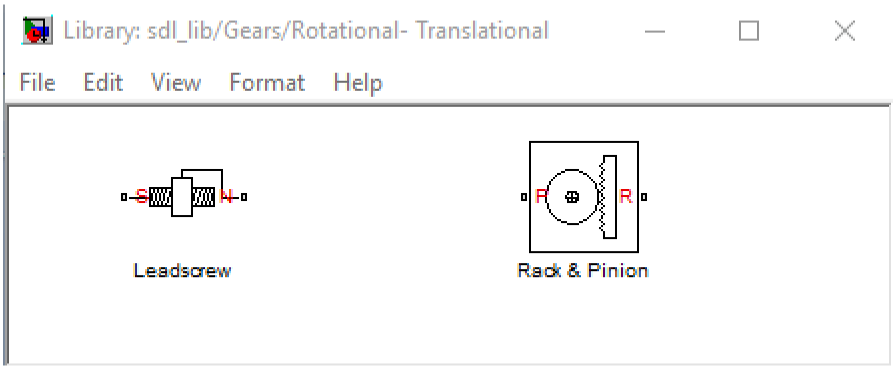

Image 4.47: SIMSCAPE’s SimDriveline Rotational-Translational(Gears)

Leadscrew	Leadscrew gear set of threaded rotating screw and translating nut, with adjustable thread and friction losses
Rack & Pinion	Rack and pinion gear coupling translational and rotational motion, with adjustable pinion radius and friction losses

Array 4.35: SimDriveline Rotational-Translational(Gears)

4.2.6 Tires & Vehicles

Let's take a closer look at the Tires & Vehicles.

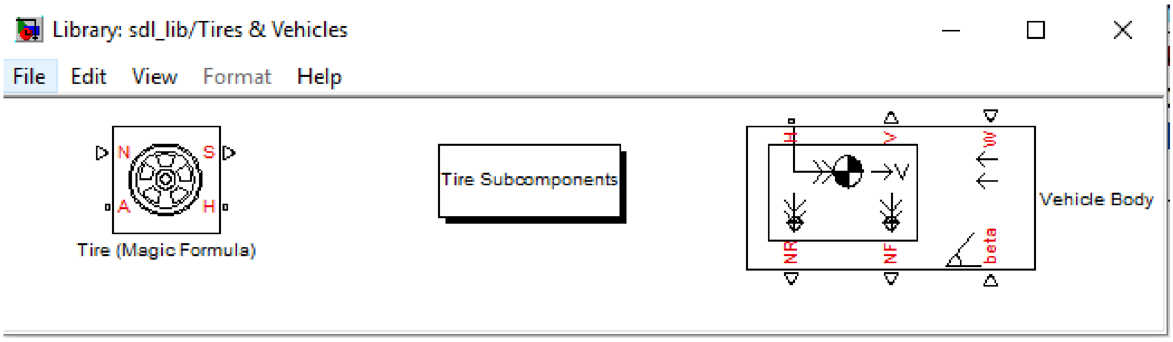

Image 4.48: SIMSCAPE's SimDriveline Tires & Vehicles

Tire(Magic Formula)	Tire with longitudinal behavior given by magic formula coefficients
Vehicle Body	Two-axle vehicle with longitudinal dynamics and motion adjustable mass, geometry, and drag properties

Array 4.36: SimDriveline Tires & Vehicles

Tire Components

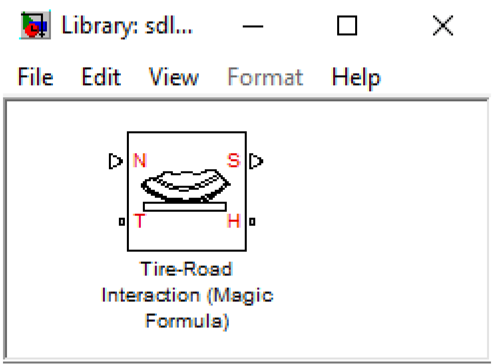

Image 4.49: SIMSCAPE's SimDriveline Tire Components

Tire-Road Interaction (Magic Formula)	Tire-road dynamics given by magic formula coefficients

Array 4.37: SimDriveline Tire Components

4.3 SimElectronics

Provides component libraries for modeling and simulating electronic and mechatronic systems. It includes models of semiconductors, motors, drives, sensors, and actuators. You can use these components to develop electromechanical actuation systems and to build behavioral models for evaluating analog circuit architectures in Simulink®. You can integrate mechanical, hydraulic, pneumatic, and other physical systems into your model using components from the Simscape family of products.

Image 4.50: SIMSCAPE's SimElectronics

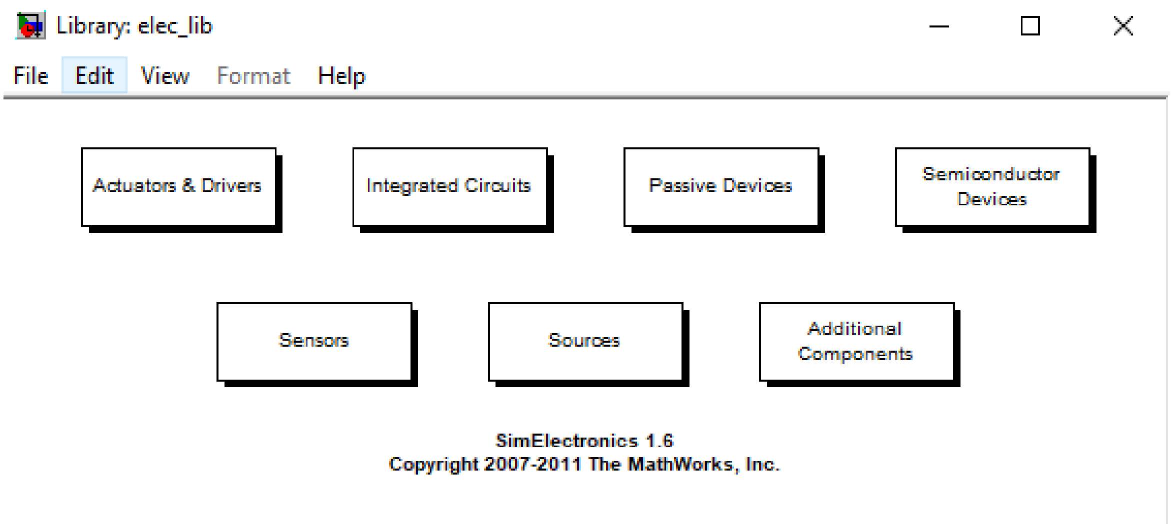

Image 4.51: SIMSCAPE's SimElectronics

4.3.1 Actuators & Drivers

Let's take a look at the Actuators & Drivers.

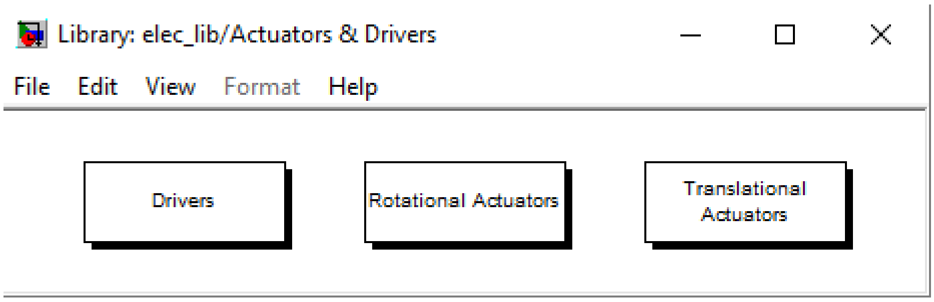

Image 4.52: SIMSCAPE's SimElectronics Actuators & Drivers

Drivers

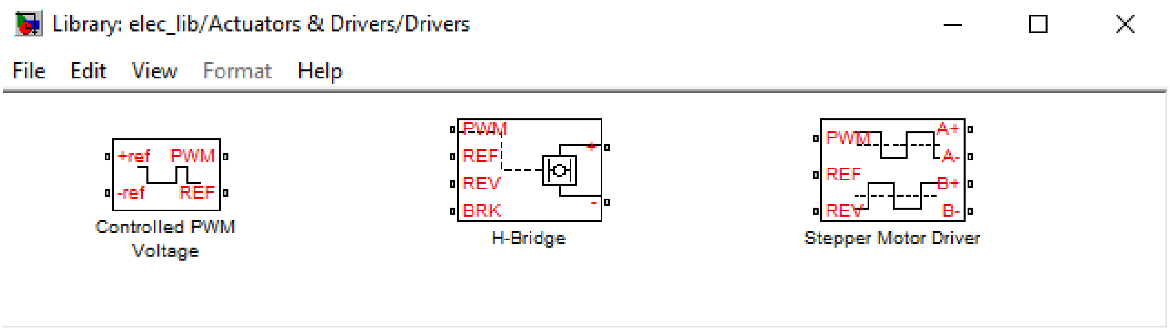

Image 4.53: SIMSCAPE's SimElectronics Drivers

Controlled PWM Voltage	Pulse-width modulated voltage source model
H-Bridge	Model H-bridge motor driver
Stepper Motor Driver	Driver for stepper motor

Array 4.38: SimElectronics Drivers

Rotational Actuators

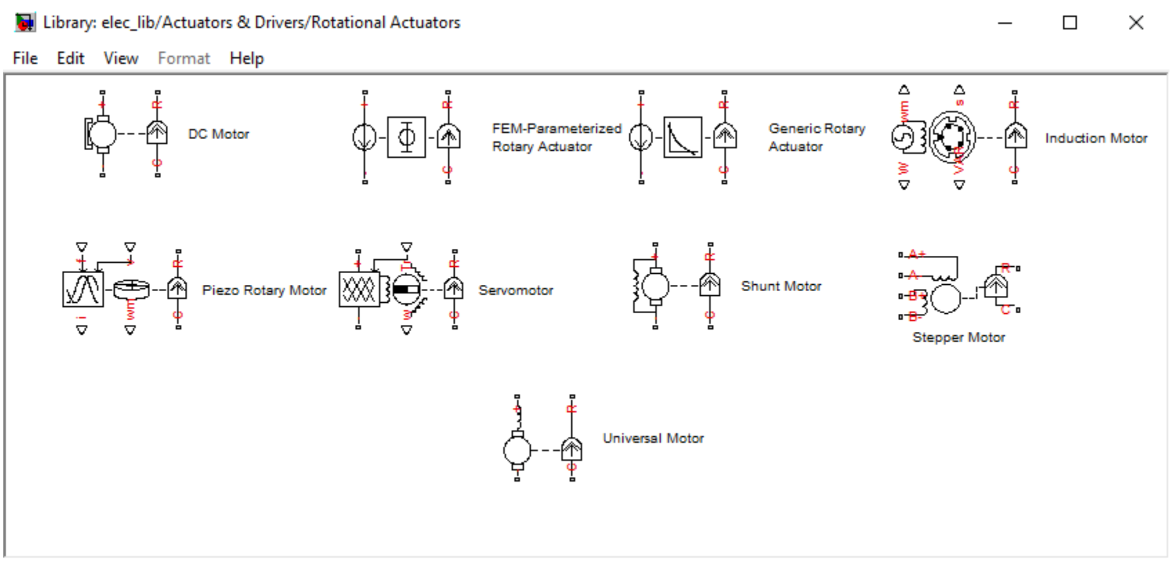

Image 4.54: SIMSCAPE's SimElectronics Rotational Actuators

DC Motor	DC motor model with electrical and torque characteristics
FEM-Parameterized Rotary Actuator	Model rotary actuator defined in terms of magnetic flux
Generic-Rotary Actuator	Model generic rotary actuator driven from DC voltage source or PWM driver
Induction Motor	Model induction motor powered by ideal AC supply
Piezo Rotary Motor	Model torque-speed characteristics of rotary piezoelectric traveling wave motor
Servomotor	Brushless motor model with closed-loop torque control
Shunt Motor	Model electrical and torque characteristics of shunt motor
Stepper Motor	Model stepper motor
Universal Motor	Model electrical and torque characteristics of a universal(or series) motor

Array 4.39: SimElectronics Rotational Actuators

Translational Actuators

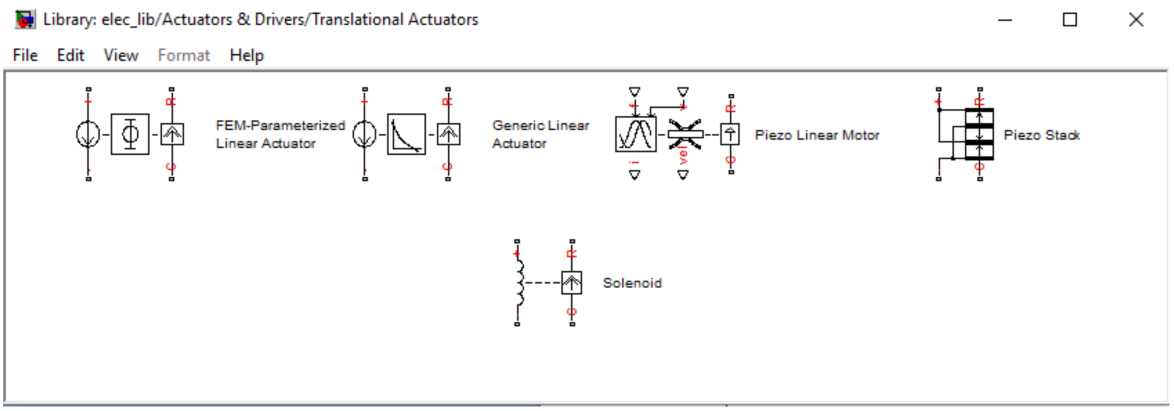

Image 4.55: SIMSCAPE's SimElectronics Translational Actuators

FEM-Parameterized Linear Actuator	Model linear actuator defined in terms of magnetic flux
Generic Linear Actuator	Model generic linear actuator driven from DC voltage source or PWM driver
Piezo Linear Motor	Model force-speed characteristics of linear piezoelectric traveling wave motor
Piezo Stack	Model electrical and force characteristics of piezoelectric stacked actuator
Solenoid	Model electrical characteristics and generated force of solenoid

Array 4.40: SimElectronics Translational Actuators

4.3.2 Integrated Circuits

Let's take a look at the Integrated Circuits.

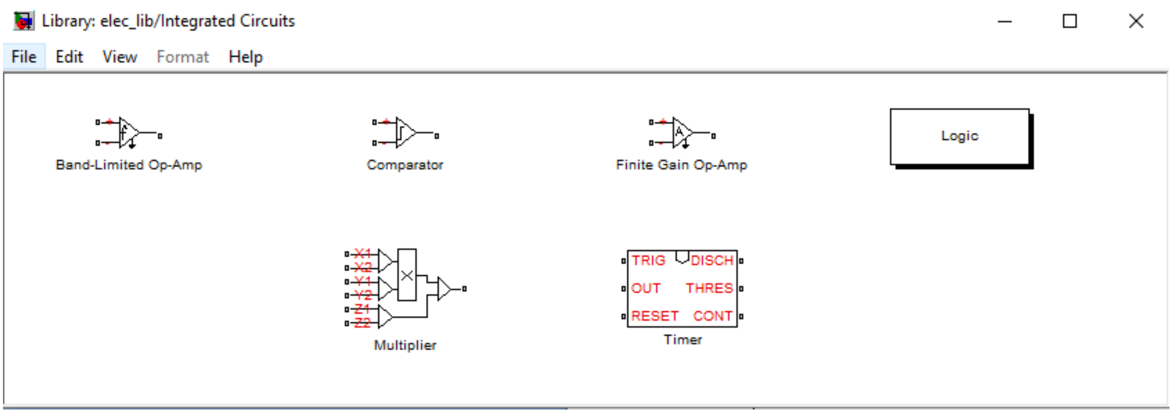

Image 4.56: SIMSCAPE's SimElectronics Integrated Circuits

Band-Limited OP-Amp	Model band-limited operational amplifier
Comparator	Model a comparator behaviorally
Finite Gain Op-Amp	Gain-limited operational amplifier model with optional noise
Multiplier	Model integrated circuit multiplier
Timer	Model timer integrated circuit behaviorally

Array 4.41: SimElectronics Integrated Circuits

Logic

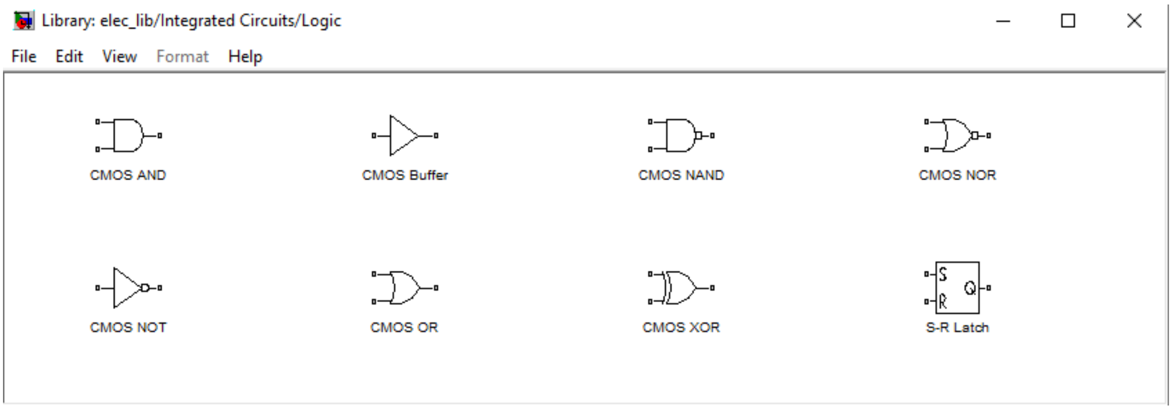

Image 4.57: SIMSCAPE's SimElectronics Logic

CMOS AND	Model CMOS AND gate behaviorally
CMOS Buffer	Model CMOS Buffer gate behaviorally
CMOS NAND	Model CMOS NAND gate behaviorally
CMOS NOR	Model CMOS NOR gate behaviorally
CMOS NOT	Model CMOS NOT gate behaviorally
CMOS OR	Model CMOS OR gate behaviorally
CMOS XOR	Model CMOS XOR gate behaviorally
S-R Latch	Model an S-R Latch behaviorally

Array 4.42: SimElectronics Logic

4.3.3 Passive Devices

Let's take a look at the Passive Devices.

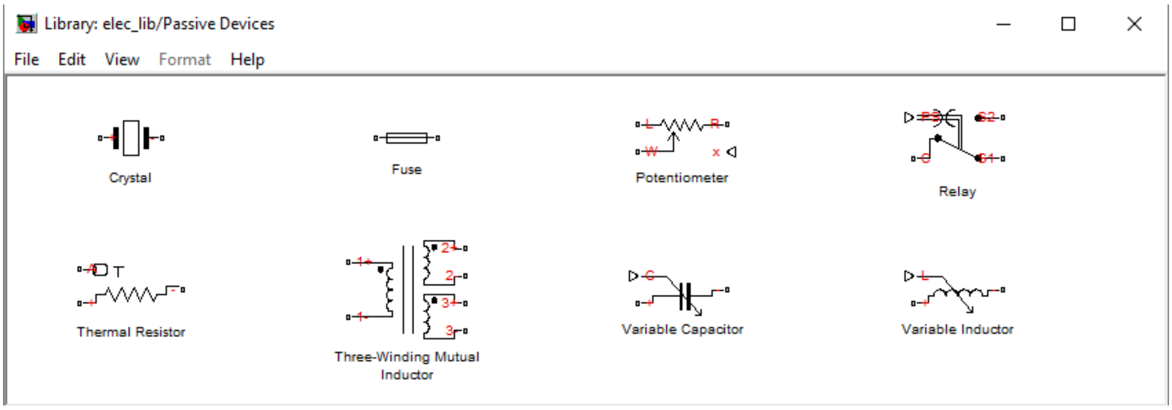

Image 4.58: SIMSCAPE's SimElectronics Passive Devices

Crystal	Model stable resonator
Fuse	Model fuse that protects against excessive current
Potentiometer	Model rotary or linear-travel potentiometer controlled by physical signal
Relay	Model switching and associated delay of relay
Thermal Resistor	Model resistor with thermal port
Three-Winding Mutual Inductor	Model three coupled inductors
Variable Capacitor	Model linear time-varying capacitor
Variable Inductor	Model linear time-varying inductor

Array 4.43: SimElectronics Passive Devices

4.3.4 Semiconductor Devices

Let’s take a look at the Semiconductor Devices.

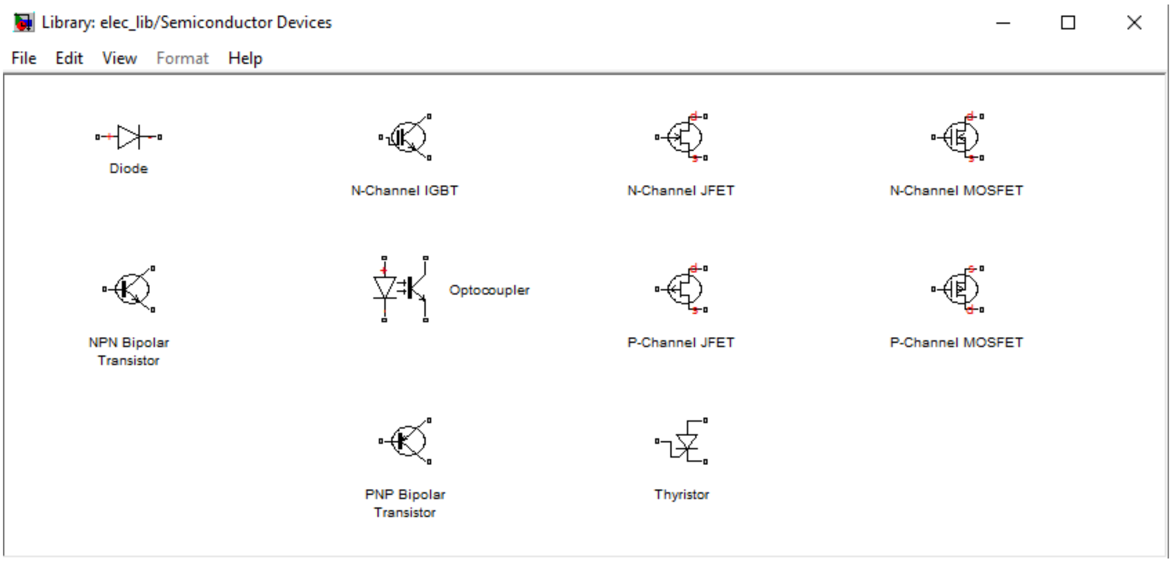

Image 4.59: SIMSCAPE’s SimElectronics Semiconductor Devices

Diode	Diode model; piecewise linear, piecewise linear zener, or exponential diode
N-Channel IGBT	Model N-Channel IGBT
N-Channel JFET	Model N-Channel JFET
N-Channel MOSFET	Model N-Channel MOSFET
NPN Bipolar Transistor	Model NPN bipolar transistor using enhanced Ebers-Moll equations
Optocoupler	Model optocoupler as LED, current sensor, and controlled current source
P-Channel JFET	Model P-Channel JFET
P-Channel MOSFET	Model P-Channel MOSFET
PNP Bipolar Transistor	Model PNP bipolar transistor using enhanced Ebers-Moll equations
Thyristor	Model thyristor using PNP transistors

Array 4.44: SimElectronics Semiconductor Devices

4.3.5 Sensors

Let’s take a look at the Sensors.

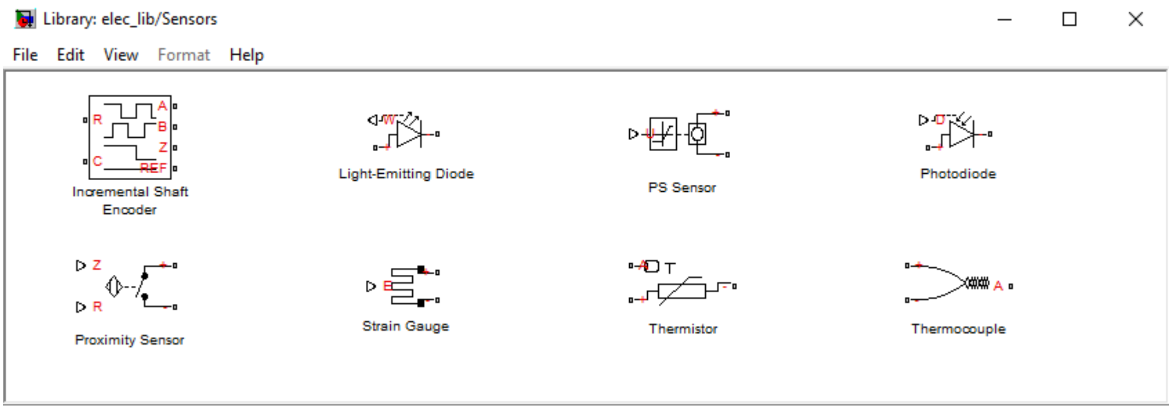

Image 4.60: SIMSCAPE’s SimElectronics Sensors

Incremental Shaft Encoder	Model device that converts information about angular shaft position into electrical pulses
Light-Emitting Diode	Model light-emitting diode as exponential diode and current sensor in series
PS Sensor	Model generic linear sensor
Photodiode	Model photodiode as parallel controlled current source and exponential diode
Proximity Sensor	Model simple distance sensor
Strain Gauge	Model deformation sensor
Thermistor	Model NTC thermistor using B-parameter equation
Thermocouple	Model sensor that converts thermal potential difference into electrical potential difference

Array 4.45: SimElectronics Sensors

4.3.6 Sources

Let’s take a look at the Sources.

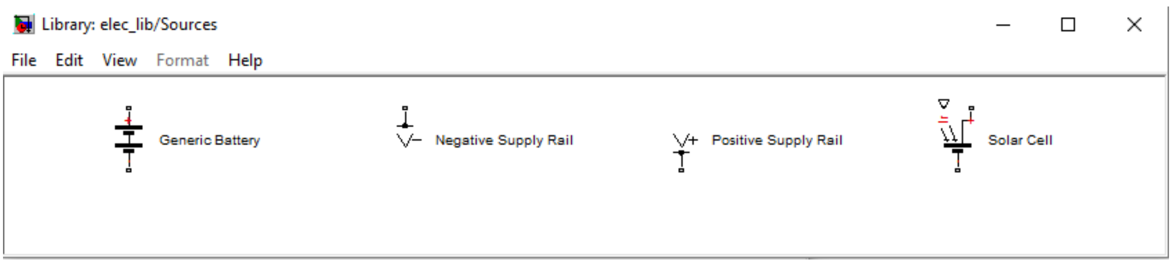

Image 4.61: SIMSCAPE’s SimElectronics Sources

Generic Battery	Simple battery model
Negative Supply Rail	Model ideal negative supply rail
Positive Supply Rail	Model ideal positive supply rail
Solar Cell	Solar cell model

Array 4.46: SimElectronics Sources

4.3.7 Additional Components

Contains SPICE-Compatible components including passive devices, semiconductor devices, sources and utilities.

SPICE(Simulation Program with Intergrated Circuit Emphasis) is an open source analog electronic circuit simulator.

4.4 SimHydraulics

Simscape Hydraulics provides component libraries for modeling and simulating fluid systems. It includes models of hydraulic pumps, valves, actuators, pipelines, and heat exchangers. You can use these components to develop fluid power systems such as front-loader, power steering, and landing gear actuation systems. Simscape Fluids also enables you to develop engine cooling, gearbox lubrication, and fuel supply systems. You can integrate mechanical, electrical, thermal, and other physical systems into your model using components from the Simscape family of products.

Simscape Hydraulics helps you develop control systems and test system-level performance. You can create custom component models with 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 hydraulic system in Simulink®. To deploy models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape Fluids supports C-code generation.

Image 4.62: SIMSCAPE's SimHydraulics

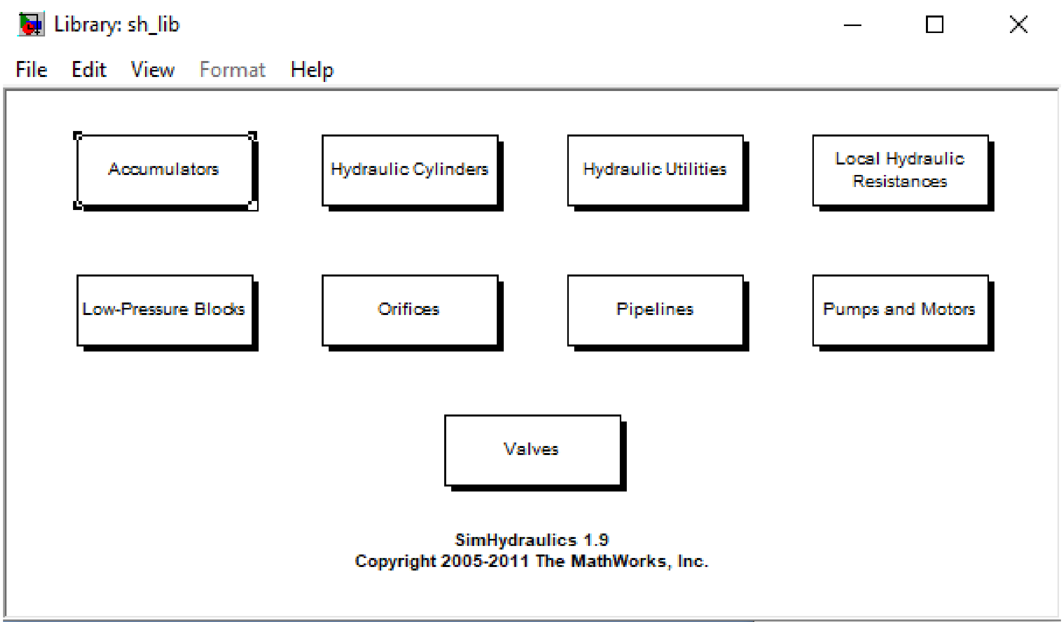

Image 4.63: SIMSCAPE's SimHydraulics

4.4.1 Accumulators

Let's take a look at the Accumulators.

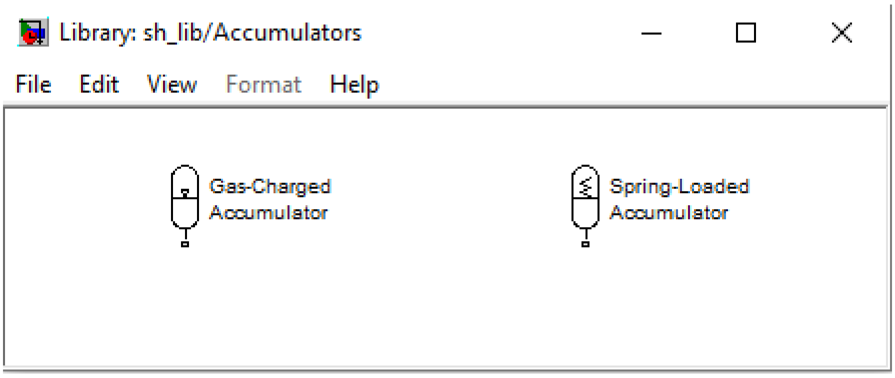

Image 4.64: SIMSCAPE's SimHydraulics Accumulators

Gas-Charged Accumulator	Hydraulic accumulator with gas as compressible medium
Spring-Loaded Accumulator	Hydraulic accumulator with spring used for energy storage

Array 4.47: SimHydraulics Accumulators

4.4.2 Hydraulic Cylinders

Let's take a look at the Hydraulic Cylinders.

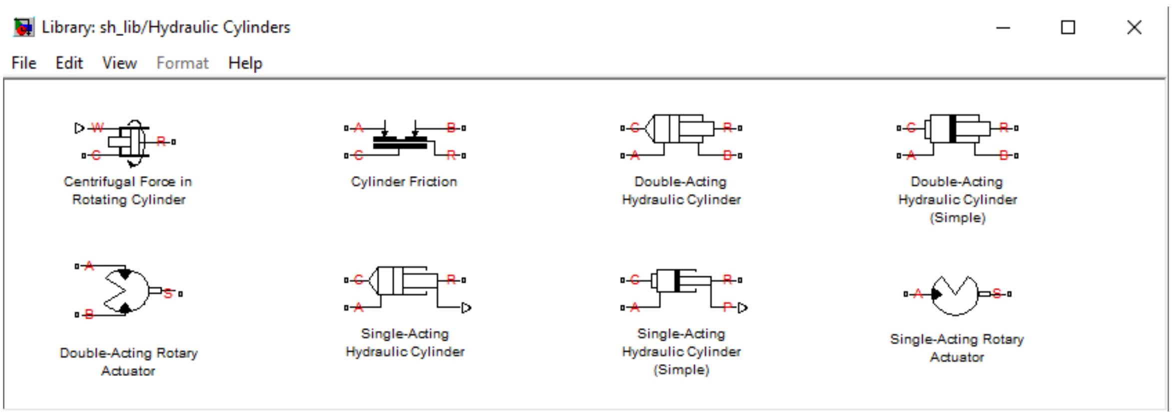

Image 4.65: SIMSCAPE's SimHydraulics Hydraulic Cylinders

Centrifugal Force in Rotating Cylinder	Centrifugal force in rotating hydraulic cylinders
Cylinder Friction	Friction in hydraulic cylinders
Double-Acting Hydraulic Cylinder	Hydraulic actuator exerting force in both directions
Double-Acting Hydraulic Cylinder(Simple)	Basic functionality of double-acting hydraulic cylinder
Double-Acting rotary Actuator	Double-acting hydraulic rotary actuator
Single-Acting Hydraulic Cylinder	Hydraulic actuator exerting force in one direction
Single-Acting Hydraulic Cylinder(Simple)	Basic functionality of single-acting hydraulic cylinder
Single-Acting Rotary Actuator	Single-acting hydraulic rotary actuator

Array 4.48: SimHydraulics Hydraulic Cylinders

4.4.3 Hydraulic Utilities

Let's take a look at the Hydraulic Utilities.

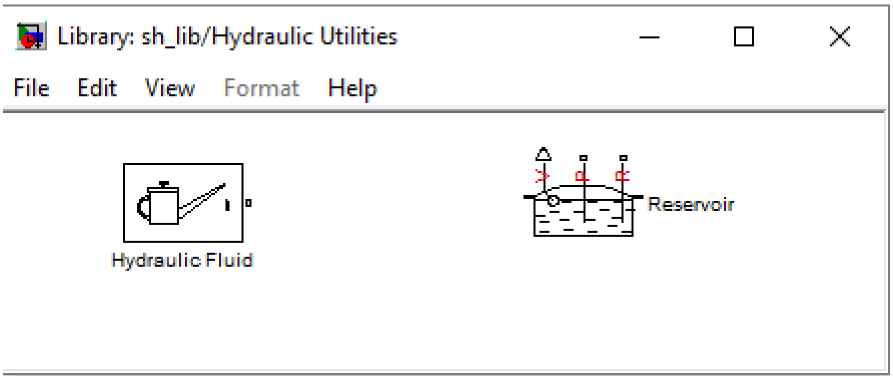

Image 4.66: SIMSCAPE's SimHydraulics Hydraulic Utilities

Hydraulic Fluid	Working fluid properties, set by selecting from list of predefined fluids
Reservoir	Pressurized hydraulic reservoir

Array 4.49: SimHydraulics Hydraulic Utilities

4.4.4 Local Hydraulic Resistances

Let's take a look at the Local Hydraulic Resistances.

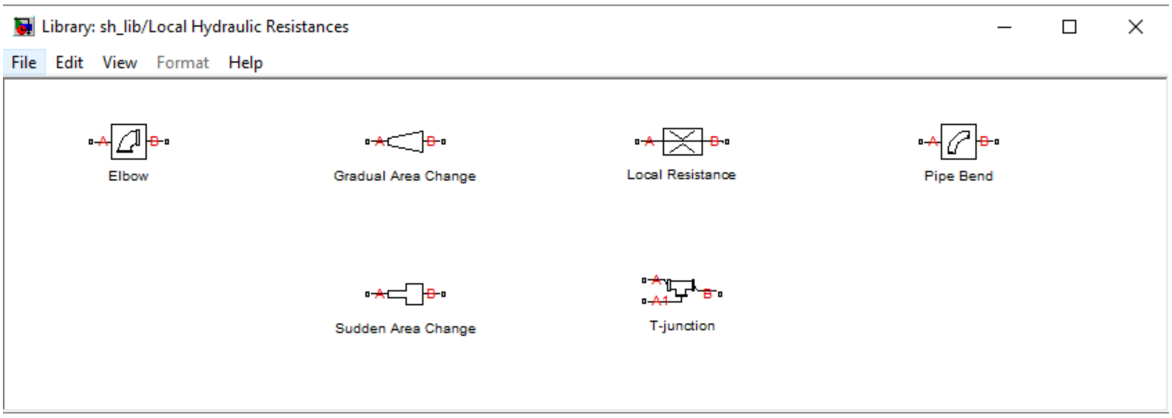

Image 4.67: SIMSCAPE's SimHydraulics Local Hydraulic Resistances

Elbow	Hydraulic resistance in elbow
Gradual Area Change	Gradual enlargement or contraction
Local Resistance	Hydraulic resistance specified by loss coefficient
Pipe Bend	Hydraulic resistance in pipe bend
Sudden Area Change	Sudden enlargement of contraction
T-junction	Hydraulic resistance of T-junction in pipe

Array 4.50: SimHydraulics Local Hydraulic Resistances

4.4.5 Low-Pressure Blocks

Let's take a look at the Low-Pressure Blocks.

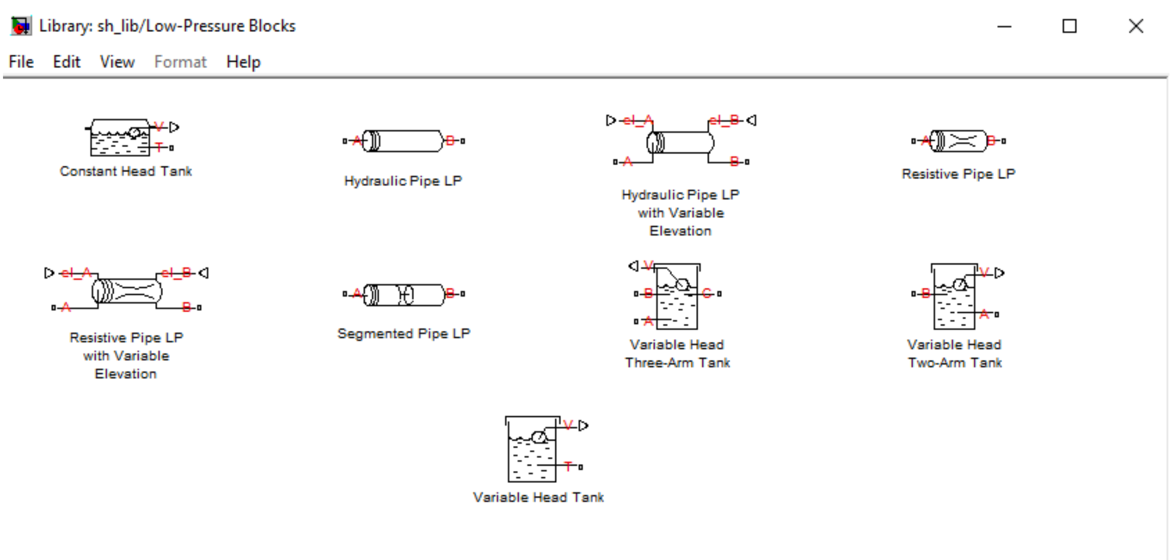

Image 4.68: SIMSCAPE's SimHydraulics Low-Pressure Blocks

Constant Head Tank	Hydraulic reservoir where pressurization and fluid level remain constant regardless of volume change
Hydraulic Pipe LP	Hydraulic pipeline with resistive, fluid compressibility, and elevation properties
Hydraulic Pipe LP with Variable Elevation	Hydraulic pipeline with resistive, fluid compressibility, and variable elevation properties
Resistive Pipe LP	Hydraulic pipeline which accounts for friction losses and port elevations
Resistive Pipe LP with Variable Elevation	Hydraulic pipeline which account for friction losses and variable port elevations
Segmented Pipe LP	Hydraulic pipelines with resistive, fluid inertia, fluid compressibility, and elevation properties
Variable Head Three-Arm Tank	Pressurized fluid container with variable fluid level
Variable Head Two-Arm Tank	Pressurized fluid container with variable fluid level
Variable Head Tank	Pressurized fluid container with variable fluid level

Array 4.51: SimHydraulics Low-Pressure Blocks

4.4.6 Orifices

Let's take a look at the Orifices.

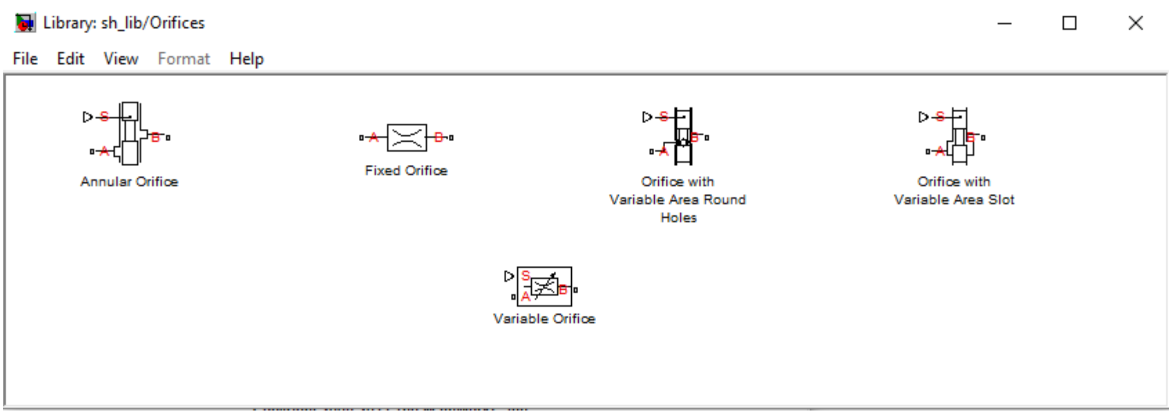

Image 4.69: SIMSCAPE's SimHydraulics Orifices

Annular Orifice	Hydraulic variable orifice created by circular tube and round insert
Fixed Orifice	Hydraulic orifice with constant cross-sectional area
Orifice with Variable Area Round Holes	Hydraulic variable orifice shaped as set of round holes drilled in sleeve
Orifice with Variable Area Slot	Hydraulic variable orifice shaped as rectangular slot
Variable Orifice	Generic hydraulic variable orifice

Array 4.52: SimHydraulics Orifices

4.4.7 Pipelines

Let's take a look at the Pipelines.

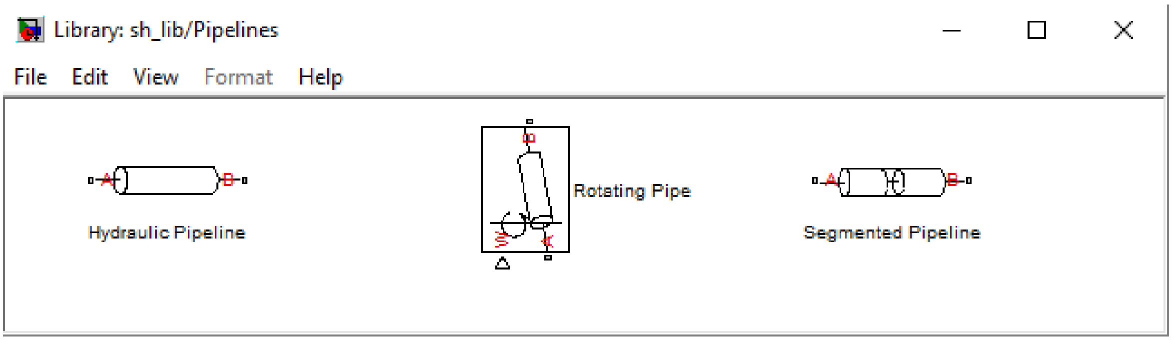

Image 4.70: SIMSCAPE's SimHydraulics Pipelines

Hydraulic Pipeline	Hydraulic pipeline with resistive and fluid compressibility properties
Rotating Pipe	Hydraulic pipeline created by bore in rotating housing
Segmented Pipeline	Hydraulic pipeline with resistive, fluid inertia, and fluid compressibility properties

Array 4.53: SimHydraulics Pipelines

4.4.8 Pumps and Motors

Let's take a look at the Pumps and Motors.

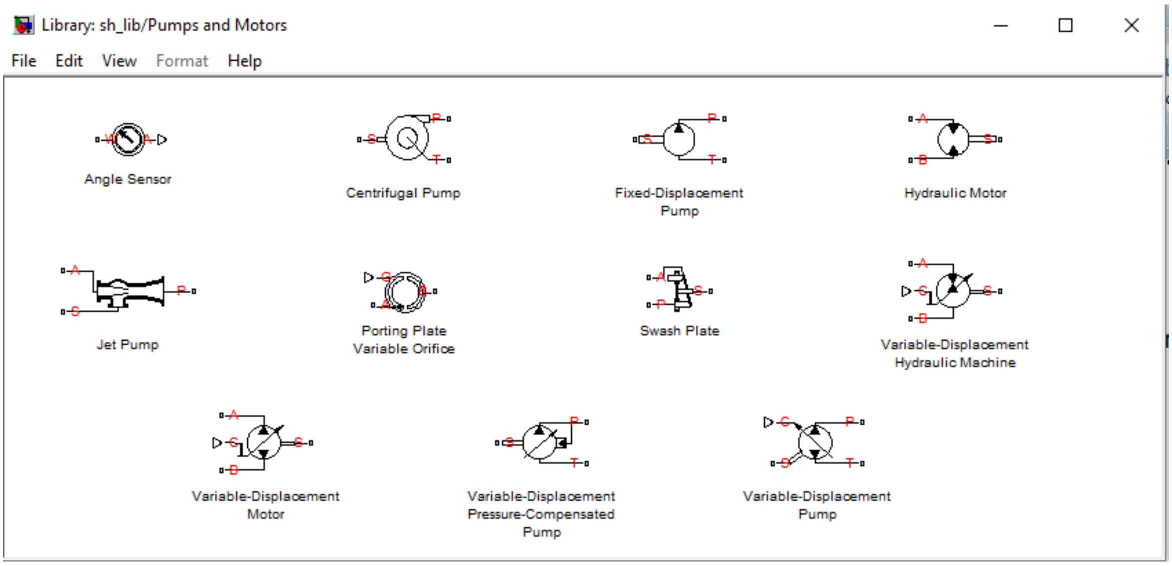

Image 4.71: SIMSCAPE's SimHydraulics Pumps and Motors

Angle Sensor	Ideal angle sensor with measuring range from 0 to 360 degrees
Centrifugal Pump	Centrifugal pump with choice of parameterization options
Fixed-Displacement Pump	Hydraulic-to-mechanical power conversion device
Hydraulic Motor	Generic hydraulic motor
Jet Pump	Jet liquid-liquid pump
Porting Plate Variable Orifice	Variable orifice between piston and porting plate
Swash Plate	Swash plate mechanism
Variable-Displacement Hydraulic Machine	Variable-displacement reversible hydraulic machine with regime-dependable efficiency
Variable-Displacement Motor	Variable-displacement bidirectional hydraulic motor
Variable-Displacement Pressure – Compensated Pump	Hydraulic pump maintaining preset pressure at outlet by regulating its flow delivery
Variable-Displacement Pump	Variable-displacement bidirectional hydraulic pump

Array 4.54: SimHydraulics Pumps and Motors

4.4.9 Valves

Let’s take a look at the Valves.

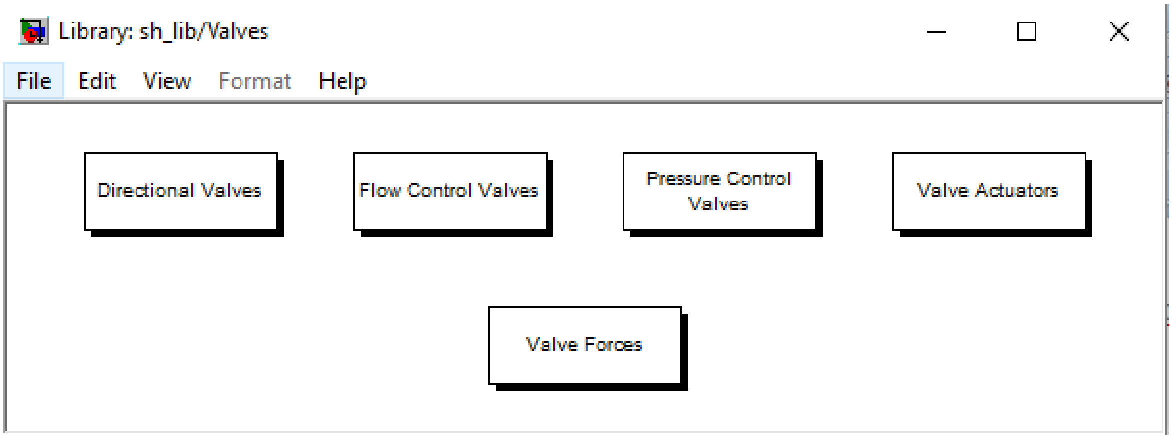

Image 4.72: SIMSCAPE’s SimHydraulics Valves

Directional Valves

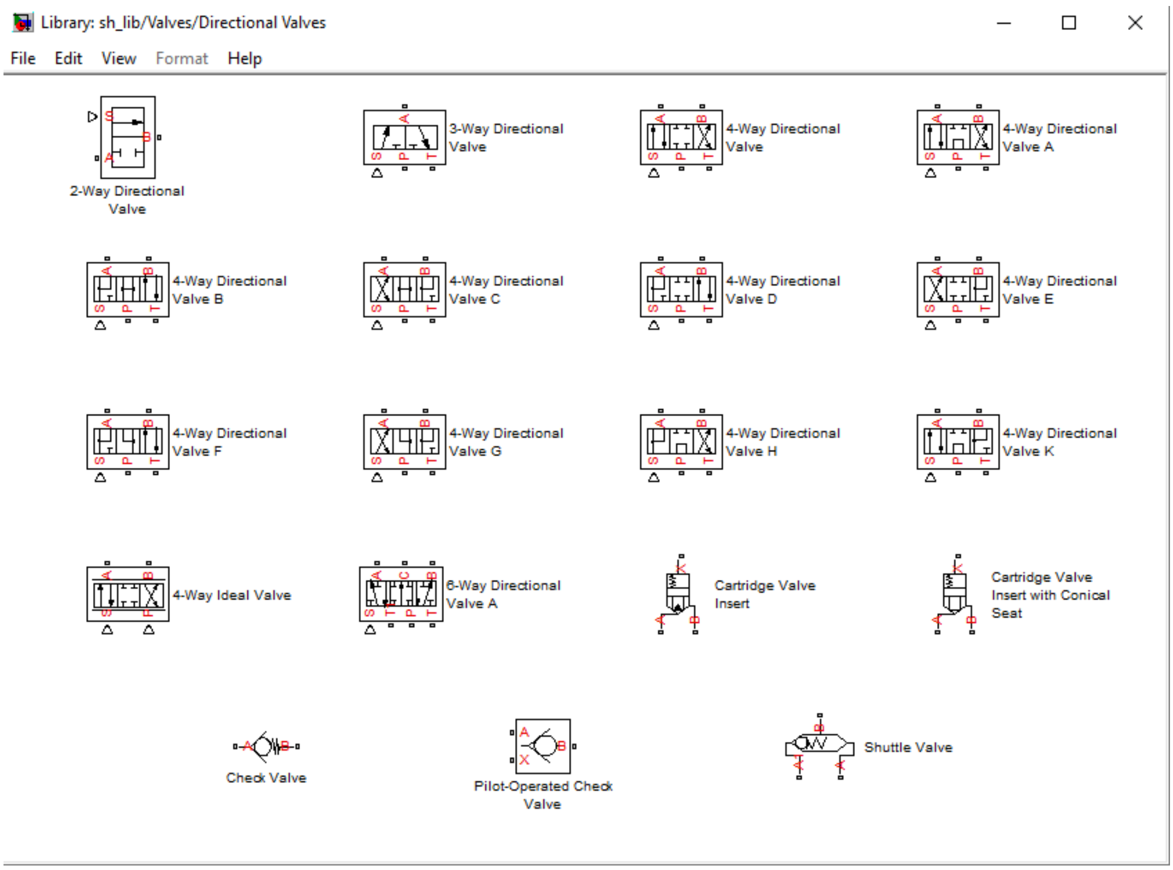

Image 4.73: SIMSCAPE’s SimHydraulics Directional Valves

2-Way Directional Valve	Hydraulic continuous 2-way directional valve
3-Way Directional Valve	Three-port two-position directional control valve
4-Way Directional Valve	Four-port three-position directional control valve
4-Way Directional Valve A	Configuration A of hydraulic continuous 4-way directional valve
4-Way Directional Valve B	Configuration B of hydraulic continuous 4-way directional valve
4-Way Directional Valve C	Configuration C of hydraulic continuous 4-way directional valve
4-Way Directional Valve D	Configuration D of hydraulic continuous 4-way directional valve
4-Way Directional Valve E	Configuration E of hydraulic continuous 4-way directional valve
4-Way Directional Valve F	Configuration F of hydraulic continuous 4-way directional valve
4-Way Directional Valve G	Configuration G of hydraulic continuous 4-way directional valve
4-Way Directional Valve H	Configuration H of hydraulic continuous 4-way directional valve
4-Way Directional Valve K	Configuration K of hydraulic continuous 4-way directional valve
4-Way Ideal Valve	Hydraulic 4-way critically-centered valve
6-Way Directional Valve A	Configuration A of hydraulic continuous 6-way directional valve
Cartridge Valve Insert	Hydraulic cartridge valve insert
Cartridge Valve Insert with Conical Seat	Hydraulic cartridge valve insert with conical seat
Check Valve	Hydraulic valve that allows flow in one direction only
Pilot-Operated Check Valve	Hydraulic check valve that allows flow in one direction, but can be disabled by pilot pressure
Shuttle Valve	Hydraulic valve that allows flow in one direction only

Array 4.55: SimHydraulics Directional Valves

Flow Control Valves

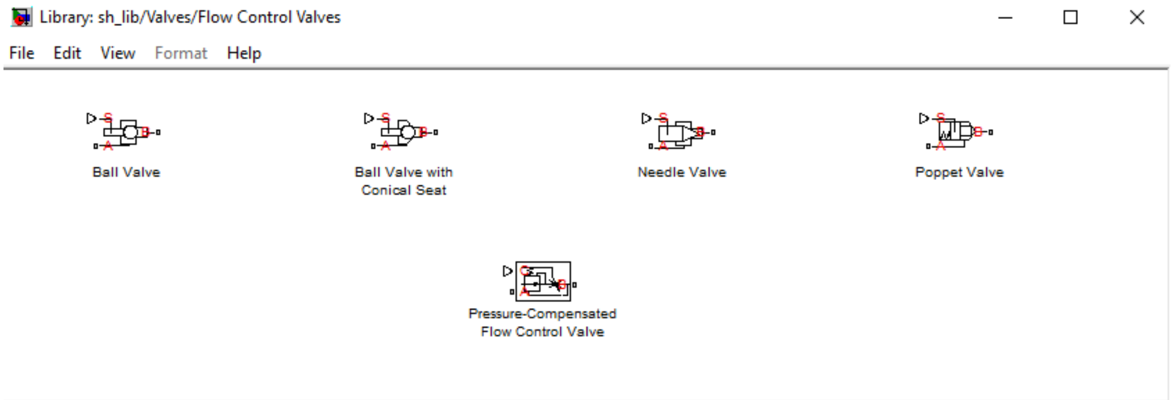

Image 4.74: SIMSCAPE's SimHydraulics Flow Control Valves

Ball Valve	Valve with a sliding ball control mechanism
Ball Valve with Conical Seat	Valve with a sliding ball control mechanism with conical seat
Needle Valve	Hydraulic needle valve
Poppet Valve	Hydraulic poppet valve
Pressure-Compensated Flow Control Valve	Hydraulic pressure compensating valve

Array 4.56: SimHydraulics Flow Control Valves

Pressure Control Valves

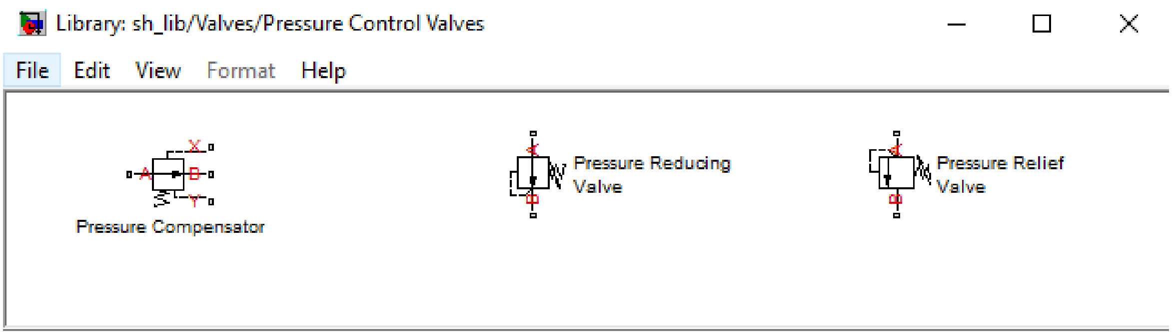

Image 4.75: SIMSCAPE's SimHydraulics Pressure Control Valves

Pressure Compensator	Hydraulic pressure compensating valve
Pressure Reducing Valve	Pressure control valve maintaining reduced pressure in portion of system
Pressure Relief Valve	Pressure control valve maintaining preset pressure in system

Array 4.57: SimHydraulics Pressure Control Valves

Valve Actuators

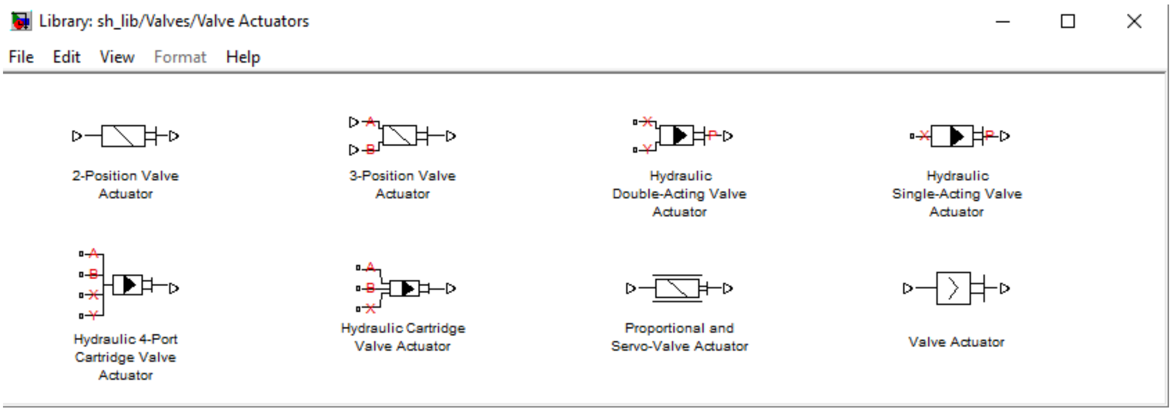

Image 4.76: SIMSCAPE's SimHydraulics Valve Actuators

2-Position Valve Actuator	Actuator for two-position valves
3-Position Valve Actuator	Actuator for three-position valves
Hydraulic Double-Acting Valve Actuator	Double-acting hydraulic valve actuator
Hydraulic Single-Acting Valve Actuator	Single-acting hydraulic valve actuator
Hydraulic 4-Port Cartridge Valve Actuator	Double-acting hydraulic valve actuator driven by four pressures
Hydraulic Cartridge Valve Actuator	Double-acting hydraulic actuator for cartridge valves
Proportional and Servo-Valve Actuator	Continuous valve driver with output proportional to input signal
Valve Actuator	Simplified model of valve driver

Array 4.58: SimHydraulics Valve Actuators

Valve Forces

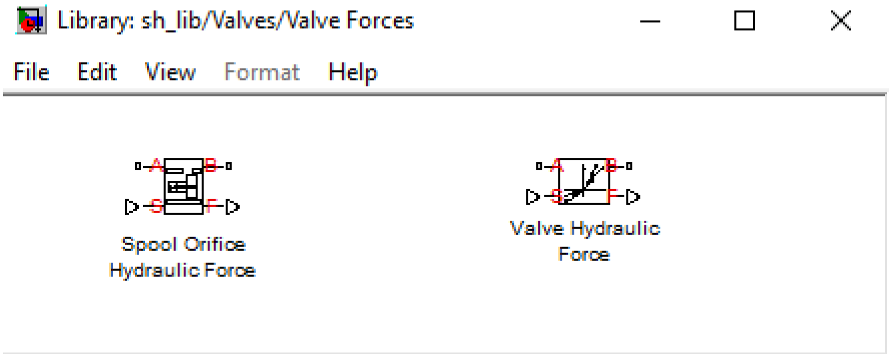

Image 4.77: SIMSCAPE's SimHydraulics Valve Forces

Spool Orifice Hydraulic Force	Axial hydraulic force exerted on spool
Valve Hydraulic Force	Axial hydraulic static force exerted on valve

Array 4.59: SimHydraulics Valve Forces

4.5 SimMechanics

Simscape SimMechanics provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. You can model multibody systems using blocks representing bodies, joints, constraints, force elements, and sensors. Simscape SimMechanics formulates and solves the equations of motion for the complete mechanical system. You can import complete CAD assemblies, including all masses, inertias, joints, constraints, and 3D geometry, into your model. An automatically generated 3D animation lets you visualize the system dynamics.

Simscape SimMechanics helps you develop control systems and test system-level performance. You can parameterize your models using MATLAB variables and expressions, and design control systems for your multibody system in Simulink. You can integrate hydraulic, electrical, pneumatic, and other physical systems into your model using components from the Simscape family of products. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape SimMechanics supports C-code generation.

Image 4.78: SIMSCAPE's SimMechanics

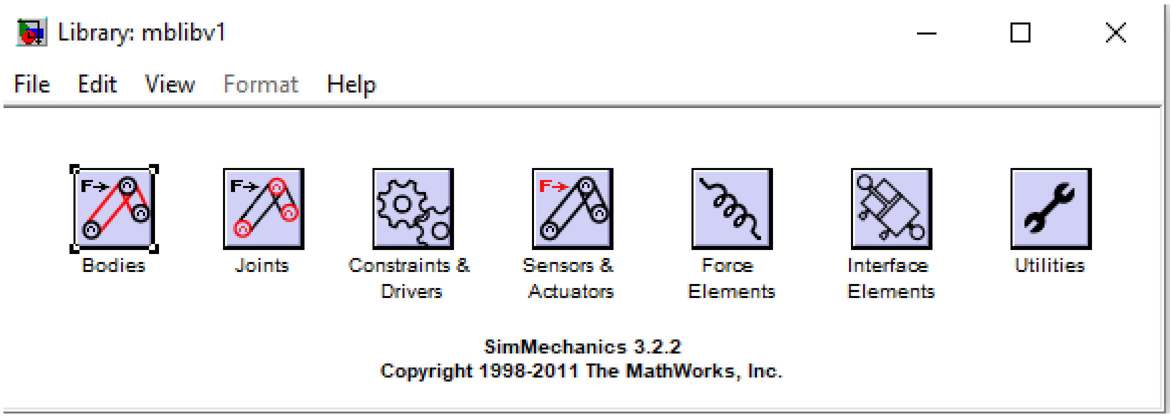

Image 4.79: SIMSCAPE's SimMechanics

4.5.1 Bodies: Represent machines and bodies

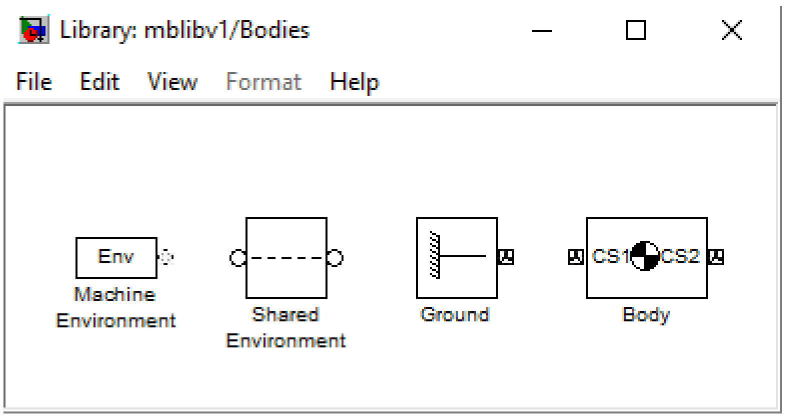

Image 4.80: SIMSCAPE's SimMechanics Bodies

Machine Environment	Mechanical simulation parameters of a machine
Shared Environment	Utility that connects two independent machines in a single mechanical environment
Ground	Fixed point attached to world
Body	Rigid body with frames, inertia and geometry

Array 4.60: SimMechanics Bodies

4.5.2 Joints: Add degrees of freedom

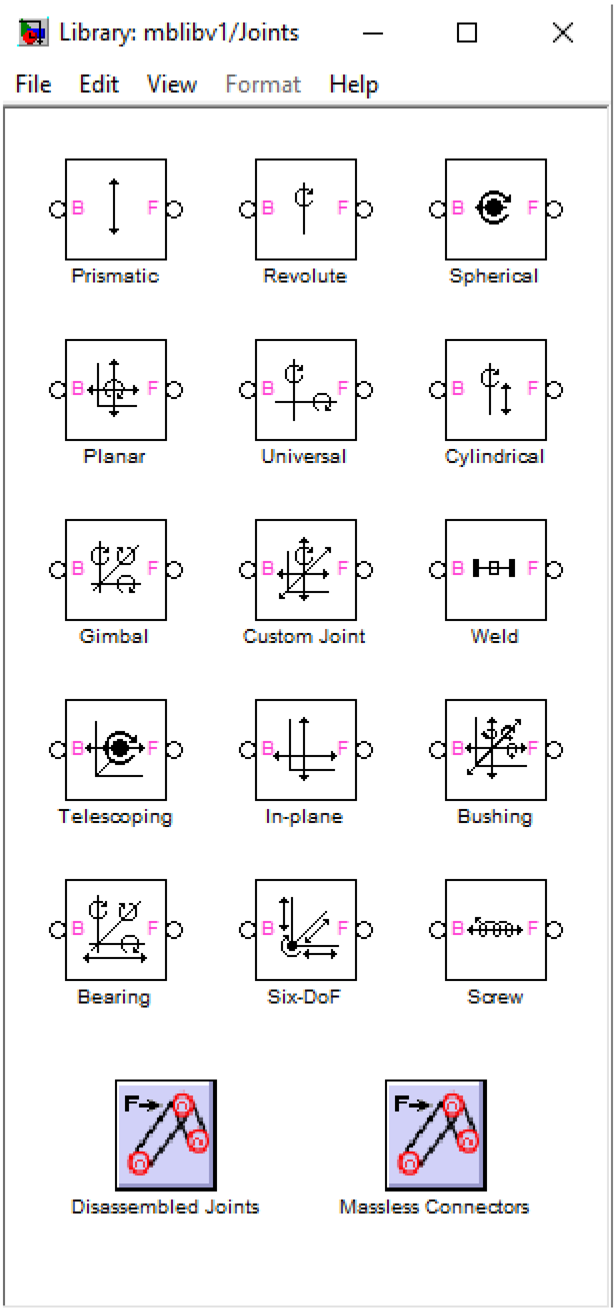

Image 4.81: SIMSCAPE's SimMechanics Assembled Joints

Prismatic	Primitive joint with one translational degree of freedom
Revolute	Primitive joint with one rotational degree of freedom
Spherical	Primitive joint with three rotational degrees of freedom
Planar	Joint with one revolute and two prismatic joint primitives
Universal	Joint with two revolute joint primitives
Cylindrical	Joint with one revolute and one prismatic joint primitives
Gimbal	Joint with three revolute joint primitives
Custom Joint	Joint with custom combination of prismatic, revolute, and spherical joint primitives
Weld	Joint with zero degrees of freedom
Telescoping	Joint with three revolute and one prismatic joint primitives
In-Plane	Joint with two coplanar prismatic joint primitives
Bushing	Joint with three revolute and three prismatic joint primitives
Bearing	Joint with three revolute and one prismatic joint primitives
Six-DoF	Joint with three revolute and one prismatic joint primitives
Screw	Joint with coupled rotational and translational degrees of freedom

Array 4.61: SimMechanics Assembled Joints

Disassembled Joints: Dislocated joints

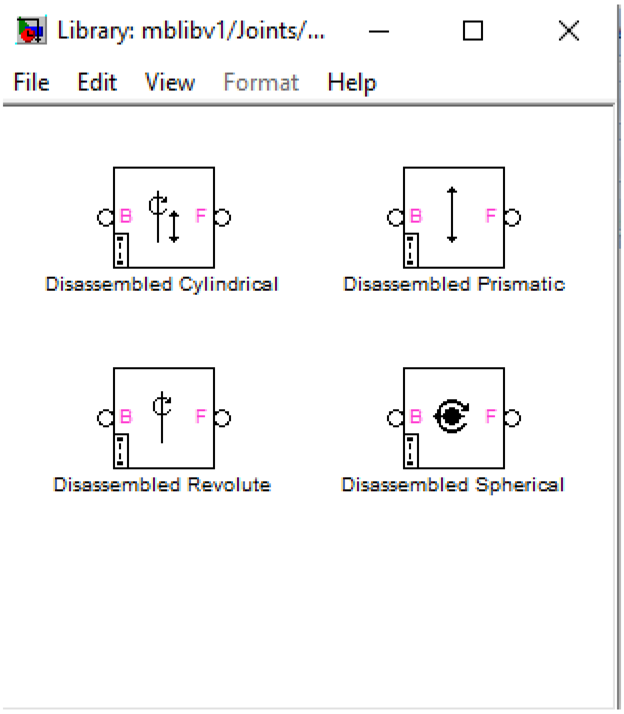

Image 4.82: SIMSCAPE's SimMechanics Disassembled Joints

Disassembled Cylindrical	Joint with misaligned base and follower axes containing one revolute and one prismatic joint primitives
Dissassembled Prismatic	Primitive joint with misaligned base and follower axes containing one translational degree of freedom
Disassembled Revolute	Primitive joint with misaligned base and follower axes containing one rotational degree of freedom
Disassembled Spherical	Primitive joint with misaligned base and follower axes containing three rotational degree of freedom

Array 4.62: SimMechanics Disassembled Joints

Massless Connectors: Rigidly separated joints

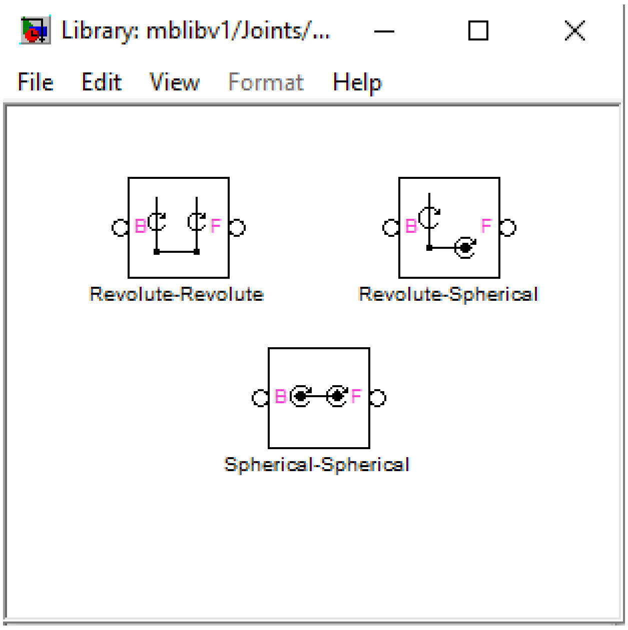

Image 4.83: SIMSCAPE's SimMechanics Massless Connectors

Revolute-Revolute	Constant-length joint connector with two spatially separated revolute axis
Revolute-Spherical	Constant-length joint connector with spatially separated revolute axis and spherical pivot point
Spherical-Spherical	Constant-length joint connector with two spatially separated spherical pivot points

Array 4.63: SimMechanics Massless Connectors

4.5.3 Constraints & Drivers: Remove degrees of freedom

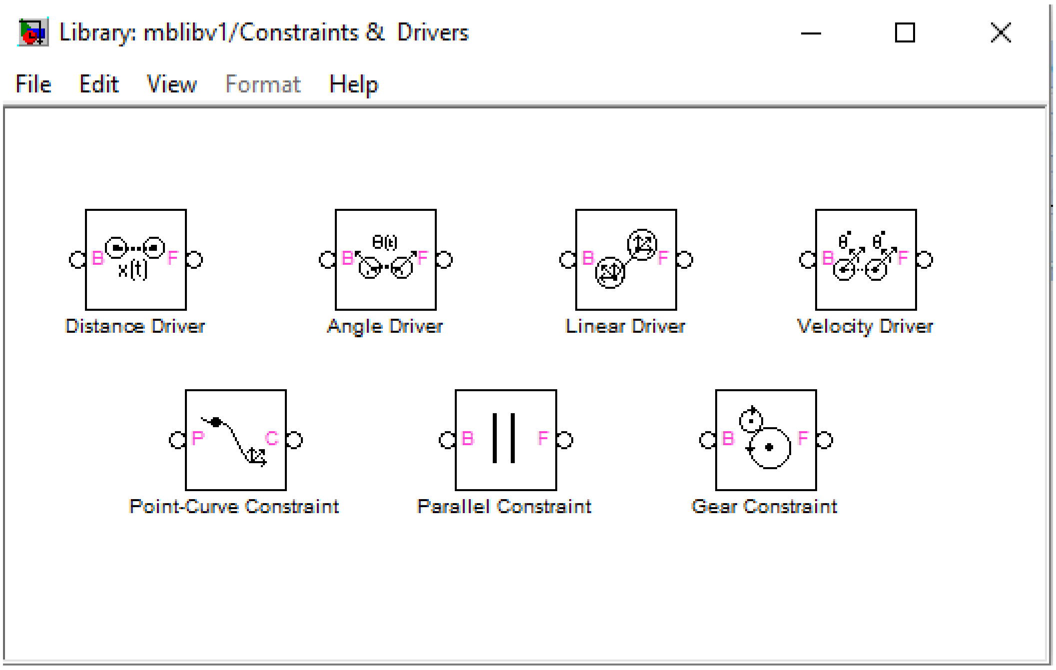

Image 4.84: SIMSCAPE's SimMechanics Constraints & Drivers

Distance Driver	Time-dependent distance between two body coordinate systems
Angle Driver	Driver specifying a time-dependent angle between two body axis vectors
Linear Driver	Time-dependent signal of a vector position component between two body coordinate systems
Velocity Driver	Linear and angular velocity components of base and follower body coordinate systems
Point-Curve Constraint	Constraint that restricts body motion to a specified path
Parallel Constraint	Constant parallel relationship between two body axis vectors
Gear Constraint	Constraint that restricts body motion to rotation along tangent circles

Array 4.64: SimMechanics Constraints & Drivers

4.5.4 Sensors & Actuators: Initiate, impose, and measure mechanical motion

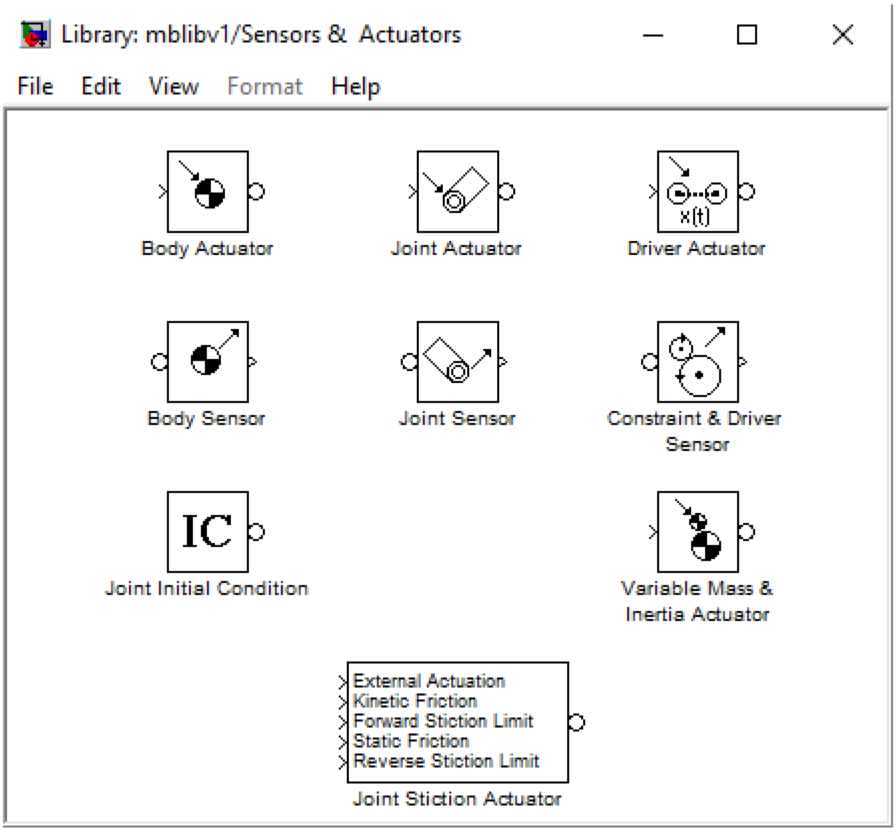

Image 4.85: SIMSCAPE's SimMechanics Sensors & Actuators

Body Actuator	Time-dependent force and torque used to actuate a body
Joint Actuator	Time-dependent force, torque or motion input to a joint
Driver Actuator	Time-dependent motion input for driver blocks
Body Sensor	Body translation and rotation sensor
Joint Sensor	Joint force, torque, and motion sensor
Constraint & Driver Sensor	Sensor used to measure the reaction force and torque between two constrained or driven bodies
Joint Initial Condition	Initial joint position and velocity
Variable Mass & Inertia Actuator	Time-dependent mass and inertia parameters
Joint Stiction Actuator	Joint static and kinetic friction

Array 4.65: SimMechanics Sensors & Actuators

4.5.5 Force Elements: Generate interbody forces

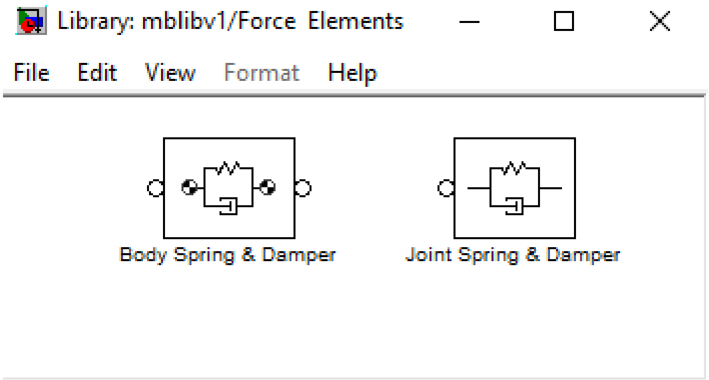

Image 4.86: SIMSCAPE's SimMechanics Force Elements

Body Spring & Damper	Damped linear oscillator force between two bodies
Joint Spring & Damper	Damped linear oscillator force or torque acting on a joint

Array 4.66: SimMechanics Force Elements

4.5.6 Interface Elements: Interface three-dimensional motion with one-dimensional domains in Simscape

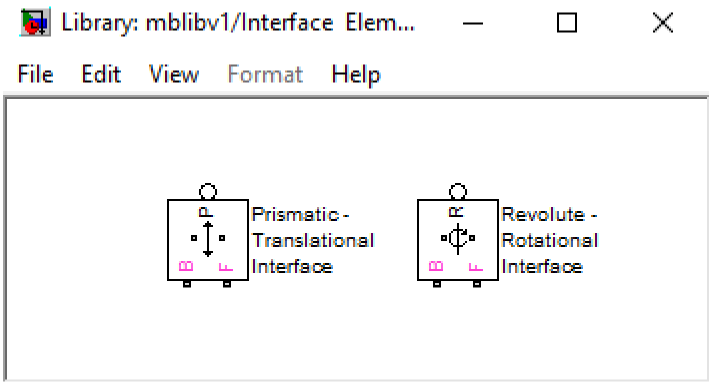

Image 4.87: SIMSCAPE's SimMechanics Interface Elements

Prismatic-Translational Interface	Connection interface between prismatic primitive and Simscape mechanical translational elements
Revolute-Rotational Interface	Connection interface between revolute primitive and Simscape mechanical rotational elements

Array 4.67: SimMechanics Interface Elements

4.5.7 Utilities: Useful Simscape SimMechanics blocks

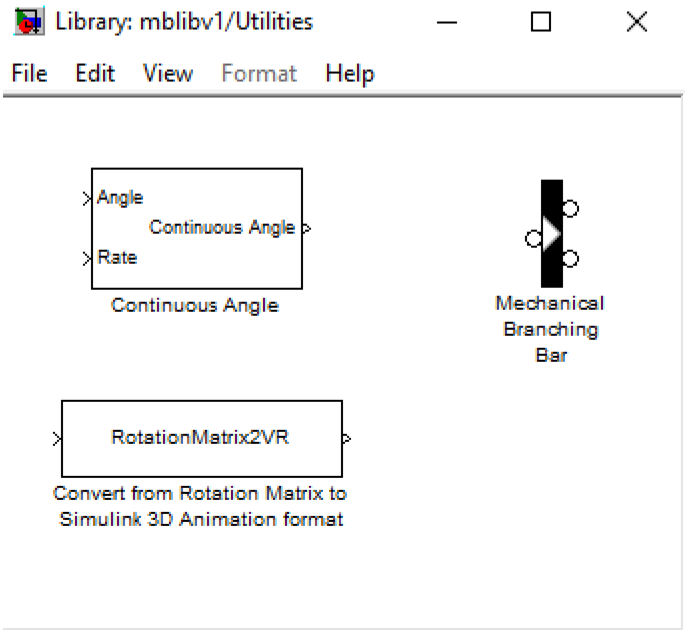

Image 4.88: SIMSCAPE's SimMechanics Utilities

Continuous Angle	Utility that converts a discontinuous bounded angle into a continuous unbounded angle
Mechanical Branching Bar	Utility that maps multiple sensor and actuation signals into a single connection line
RotationMatrix2VR	Utility that transforms 3x3 rotation matrix into rotation axis-angle 4-vector

Array 4.68: SimMechanics Utilities

4.6 Utilities

Required and commonly used blocks for setting simulation environment, interfacing with Simulink models, and generating custom components on-the-fly

Image 4.89: SIMSCAPE's Utilities

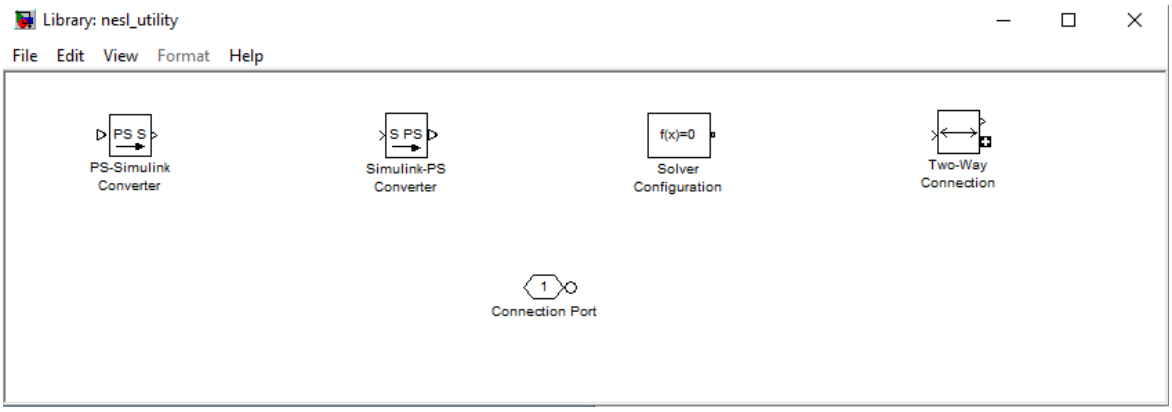

Image 4.90: SIMSCAPE's Utilities

PS-Simulink Converter	Convert physical signal into Simulink output signal
Simulink-PS Converter	Convert Simulink input signal into physical signal
Solver Configuration	Physical Networks environment and solver configuration
Two-Way Connection	Two-way connector port for subsystem
Connection Port	Physical Modeling connector port for subsystem

Array 4.69: Simscape Utilities

CHAPTER 5

SIMSCAPE LANGUAGE

5.1 Simscape Language

The Simscape Language is a MATLAB-based programming language that gives the user the ability to create custom component models. MATLAB is a text based language for coding, mainly used for mathematical problems and control systems. A program therefore written in MATLAB is lines of code. Simulink is a Model-Based system design. MATLAB and Simulink combined offer the user textual and graphical programming in order to design a system in a simulation environment. Simscape, which is a package available in Simulink offers physical systems simulation through various components such as electrical, hydraulic, magnetic, mechanical, thermal, pneumatic etc. that are ready to use through libraries. The Simscape foundation library has basic components in each domain and the user can see the code used for most of them and can also make changes.

Simscape Language is a very powerful tool that gives the ability to the user to create new custom components, however complex they might be in order to match actual physical components to be used in simulation.

5.2 Component Types

In physical modeling there are two types of models, Behavioral and Composite. Behavioral models that are implemented based on their physical behavior and are described by a system of mathematical equations. Composite models on the other hand are constructed out of other blocks and they are connected in a certain way. Simscape language allows the user to create new behavioral and composite models when the pre-existing standard block libraries don’t match the user’s design requirements.

5.3 Creating a new component

New custom built components require their own file with the '.ssc' extension. A component file starts with the keyword 'component' followed by the component name (Component names cannot have spaces in-between) and is terminated by the keyword 'end'. After that there is a comment which names the block in the library. After naming the component and defining the block, there's optional room for more comments describing the custom component (it appears in the description of the block dialog box). Next, component files can have the following sections in any order.

- **Declaration** →Contains declarations for the component : variables, nodes, inputs and outputs.
- **Setup** →Contains assignment statements like *if* and *error.* The setup function is executed once for each component instance during model compilation. It prepares the component for simulation
- **Branches** →Describes the relationship between the component variables and nodes.
- **Structure** →Declares the component connections for composite models.
- **Events** →Event modeling lets you perform changes on continuous variables.
- **Equation** →Declares the component equations for behavioral models.

Each section ends with the keyword 'end'.

5.4 Thermal Resistor

In order to create a new component using the Simscape Language first we have to write the code and describe what the component does. Run MATLAB and go to **File → New → Script** and we type the following code :

Editor - C:\Program Files\MATLAB\R2011a\toolbox\physmod\simscape\library\m\+MyCustomSimscapeLibrary\CustomElements\MyResistor.ssc

```
component MyResistor
% R Therm
% Resistor with temperature dependence defined by V=R(1+alpha*(T-10))
% where R is the nominal resistance at the reference temperature in ohms
% and alpha is the temperature coefficient.

  nodes
    p = foundation.electrical.electrical; % +:left
    n = foundation.electrical.electrical; % -:right
  end
  variables
    i = { 0, 'A' };      % Current
    v = { 0, 'V' };      % Voltage
  end
  parameters
    R = { 1, 'kOhm' };       % Nominal Resistance
    a = { 0.001, '1/K' };    % Temperature Coefficient
    T0 = { 300, 'K' };       % Reference Temperature
    T = {300, 'K' };         % Current Temperature
  end
  function setup
    if R < 0
        error( 'Resistance value must be greater than zero' );
    end
  end
  branches
    i : p.i -> n.i;
  end
  equations
    v == p.v - n.v;
    v == R*(1+a*(T-T0))*i;
    end
end
```

Image 5.1: Thermal Resistor code

Then go to **File → Save As**…→name it MyResistor.ssc ,choose all files and click save. After that is done, go to MATLAB Command Window and type **'ssc_build'**. Now our custom made Thermal Resistor is ready to be used by Simscape.

Let's take a closer look at our code.

Line 1 → The first line in our code defines that we are making a component named MyResistor
Line 2 → In the second line we describe the block name in which case its called 'R Therm'.
Lines 3-6 → Optional short description of the component.
Lines 7-10 → Our component needs two electrical ports, the '+' and '-' of our resistor in order to be connected to other blocks or elements.
Lines 11-14 → Declaration of the component. Two variables, current and voltage.
Lines 15-20 → Declaration of the component. We describe the parameters which the user can change by double clicking the component.

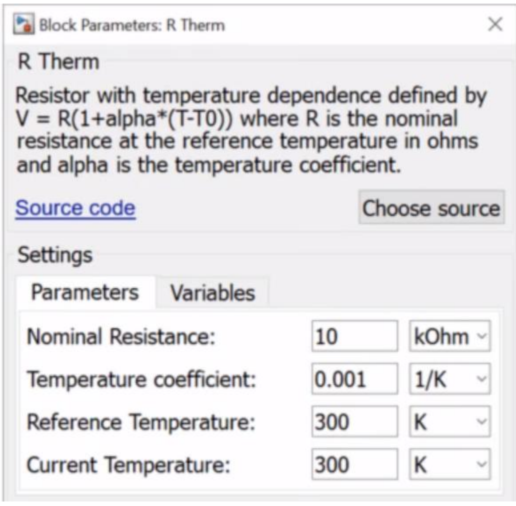

Image 5.2: Block Parameters: R Therm

Lines 21-25 → MATLAB functions and expressions for typical physical modeling tasks such as analyzing parameters and performing computations.
Lines 26-28 → Describes the relationship between variables and nodes.
Lines 29-32 → In this section we define how the component behaves.

CHAPTER 6

A LOOK AT DEMOS AND CREATING A NEW MODEL

6.1 Setting up the Simscape Simulation Results Explorer.

For MATLAB versions earlier than 2015 in order to log and view the simulation data we are going to install an add-on called Simscape Simulation Results Explorer.

Step 1 → Go to https://www.mathworks.com/matlabcentral/fileexchange/28184-simscape-simulation-results-explorer

Step 2 → Click the **download** button

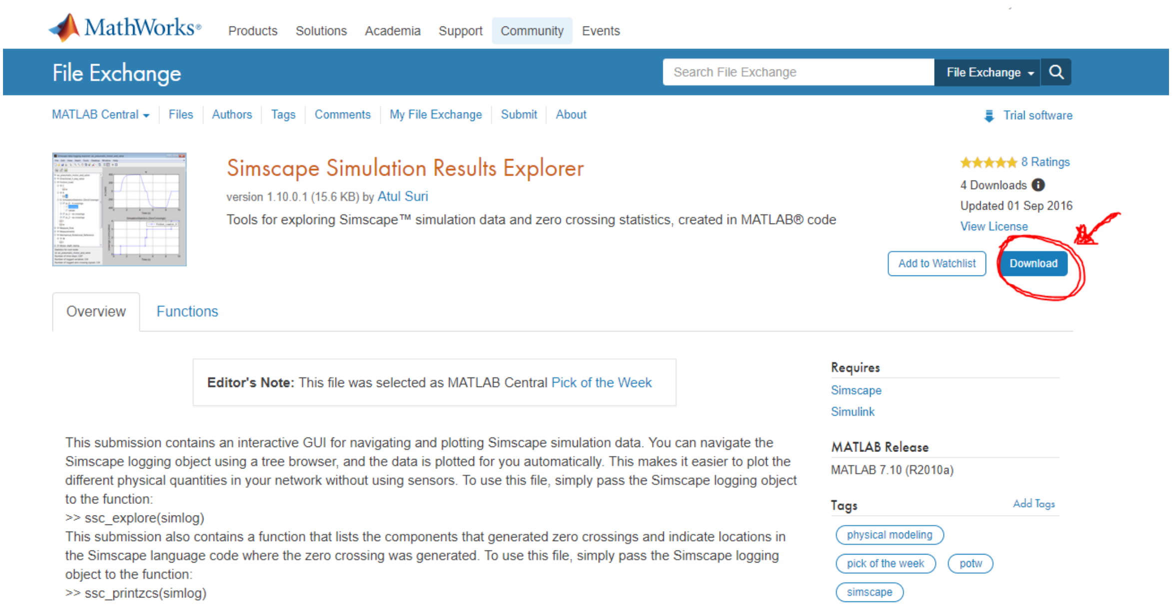

Image 6.1: Results explorer download page

Step 3 → This will download a .zip file called **simscape_logging.zip**. Once the download is finished, locate the file.

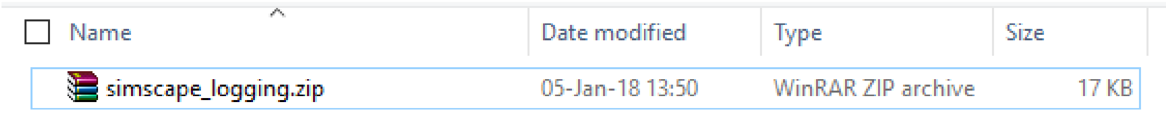

Image 6.2: simscape_logging.zip

Step 4 → Right click the file and chose "**Extract to simscape_logging**"

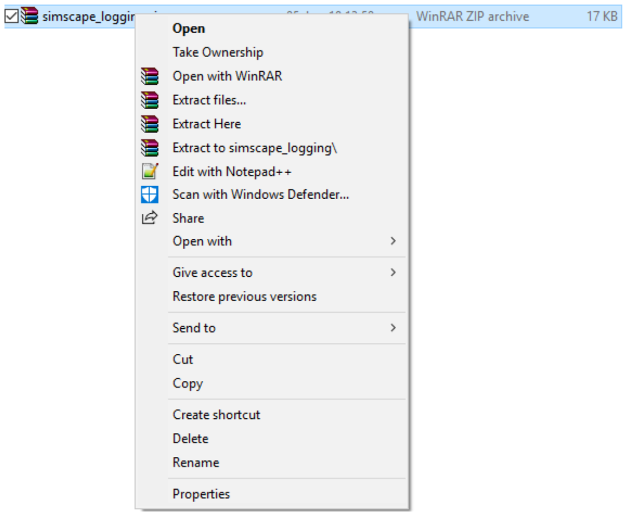

Image 6.3: simscape_logging.zip extraction

Step 5 → This is going to create a folder called "**simscape_logging**". Right click the folder and **Copy.**

Step 6 → Go to C:\Program Files\MATLAB\R2011a\toolbox\matlab\ and right click **→ paste**.

Once this is done we are ready to use the .m file called ssc_explore which is a function that gives the user the ability to log and view the simulation data.

6.2 List of available demos

The available demos in Matlab R2011a are:

Band-limited Op-Amp	ssc_bandlimited_opamp
Nonlinear Bipolar Transistor	ssc_bipolar_nonlinear
Small Signal Bipolar Transistor	ssc_bipolar_smallsignal
Full-Wave Bridge Rectifier	ssc_bridge_rectifier
Circuit Breaker	ssc_circuitbreaker
DC Motor Thermal Circuit	ssc_dc_motor_thermal_circuit
Permanent Magnet DC Motor	ssc_dcmotor
Differentiator	ssc_differentiator
Model Using a Customized Electrochemical Library	ssc_electrochemical_battery
Finite-Gain Op-Amp	ssc_finitegain_opamp
House Heating System	ssc_house_heating_system
Hydraulic Actuator with Analog Position Controller	ssc_hydraulic_actuator_analog_control
Hydraulic Closed-Loop System with Two-Way Valve	ssc_hydraulic_system_2_way_valve
Inverting Op-Amp Circuit	ssc_inverting_opamp
Lead-Acid Battery	ssc_lead_acid_battery
Linkage Mechanism	ssc_linkage_mechanism
Mechanical System with Translational Friction	ssc_mechanical_system_translational_friction
Mechanical System with Translational Hard Stop	ssc_mechanical_system_translational_hardstop
New electronic circuit	ssc_new_elec
Noninverting Op-Amp Circuit	ssc_non_inverting_opamp
Nonlinear Inductor	ssc_nonlinear_inductor
Pneumatic Actuation Circuit	ssc_pneumatic_actuator
Pneumatic Motor	ssc_pneumatic_motor
Pneumatic Motor and Directional Control Valve	ssc_pneumatic_motor_and_valve
Heat Conduction through Iron Rod	ssc_round_rod_heat_conduction
Shunt Motor	ssc_shuntmotor
Simple Mechanical System	ssc_simple_mechanical_system
Solenoid	ssc_solenoid
Solenoid with Magnetic Blocks	ssc_solenoid_magnetic
Switched Reluctance Actuator	ssc_switched_reluctance_actuator
Electrical Transformer	ssc_transformer
Model using a Customized Capacitor Library	ssc_ultracapacitor

Mechanical Rotational System with Stick-Slip Motion	ssc_rot_system_stick_slip

Array 6.1: Simscape Demos

6.3 Permanent Magnet DC Motor(PMDC)

This model is based on a Faulhaber Series 0615 DC-Micromotor. For the 1.5V variant, equivalent circuit parameters are given as:

Rotor resistance R=3.9 Ohms
Rotor inductance L=12mH
Back emf constant K=0.072mV/rpm
Rotor inertia J=0.01gcm^2
Friction torque Mr=0.02mNm

The model uses these parameters to verify manufacturer-quoted no-load speed, no-load current, and stall torque, which are:

No-load speed = 19,100rpm
No-load current = 0.03A
Stall torque = 0.24mNm

When running the simulation, for the first 0.1 seconds the motor has no external load, and the speed builds up to the no-load value. Then at 0.1 seconds the stall torque is applied as a load to the motor shaft. Zooming in on the RPM and Motor Current scopes shows a good level of agreement with manufacturer data.

Often manufacturers do not quote the equivalent circuit parameters, and they must be estimated from information such as no-load speed, stall torque, and efficiency. A test harness such as this model can then be used to test the estimated equivalent circuit prior to using the motor model in a complete system simulation.

Let's go ahead and open it.

Step 1 → Run Matlab and set the Path

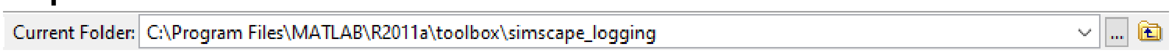

Image 6.4: Matlab Path for simulation data

To be sure that the ssc_explore m file is ready to be used for simulation logging go ahead and type "**which ssc_explore**"

```
>> which ssc_explore
C:\Program Files\MATLAB\R2011a\toolbox\simscape_logging\ssc_explore.m
>>
```

Image 6.5: Matlab path verification

Now, we are ready to run the built in example and take a look at the simulation data.

Step 2 → Type "**ssc_dcmotor**". Press Enter and the Model is going to open.

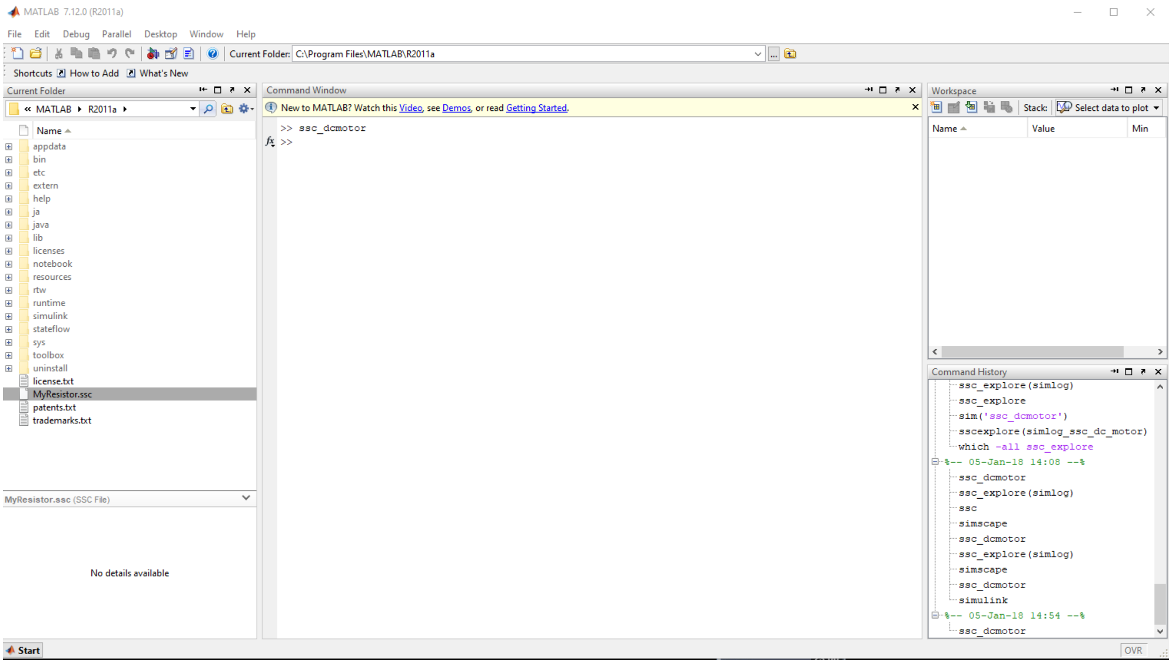

Image 6.6: Running ssc_dcmotor

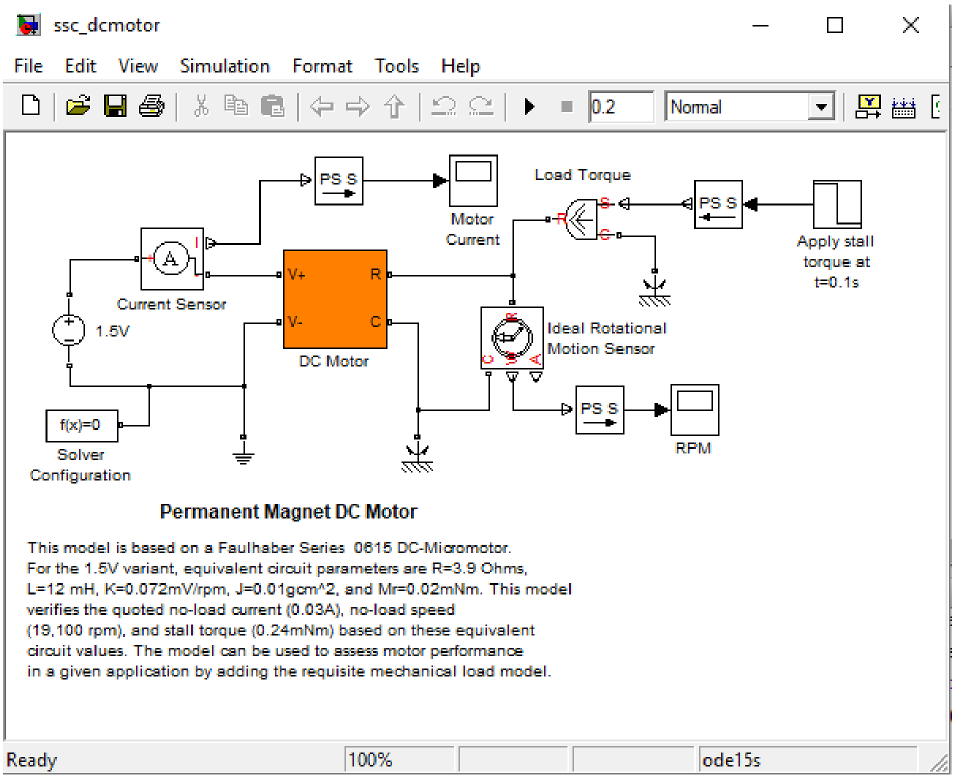

Image 6.7: ssc_dcmotor model

If we double click on the DC Motor, the subsystem is going to open up.

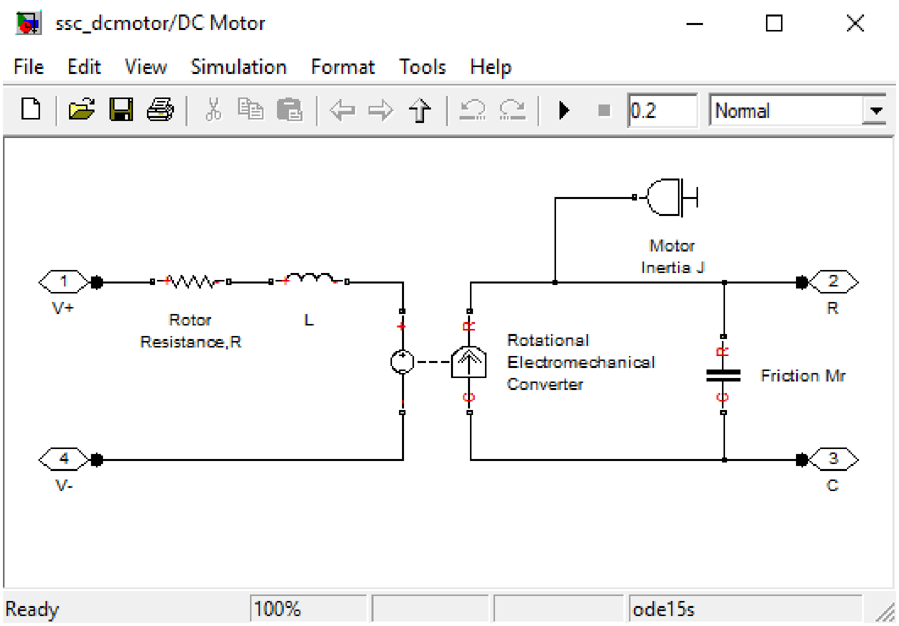

Image 6.8: ssc_dcmotor subsystem

Step 3 → Go to Simulation → Configuration Parameters…

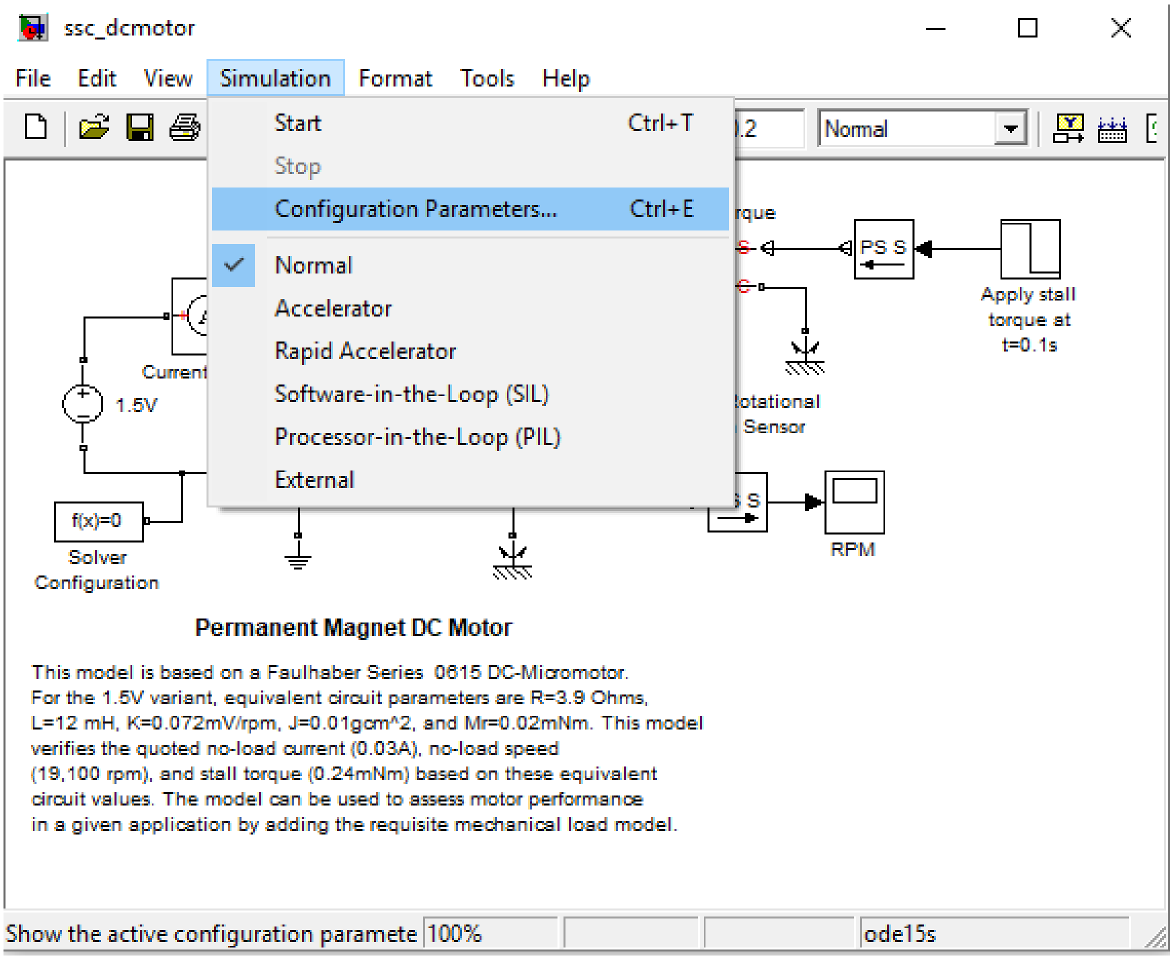

Image 6.9: Configuration Parameters

Go to Simscape and change the Log simulation data from none to **all**.

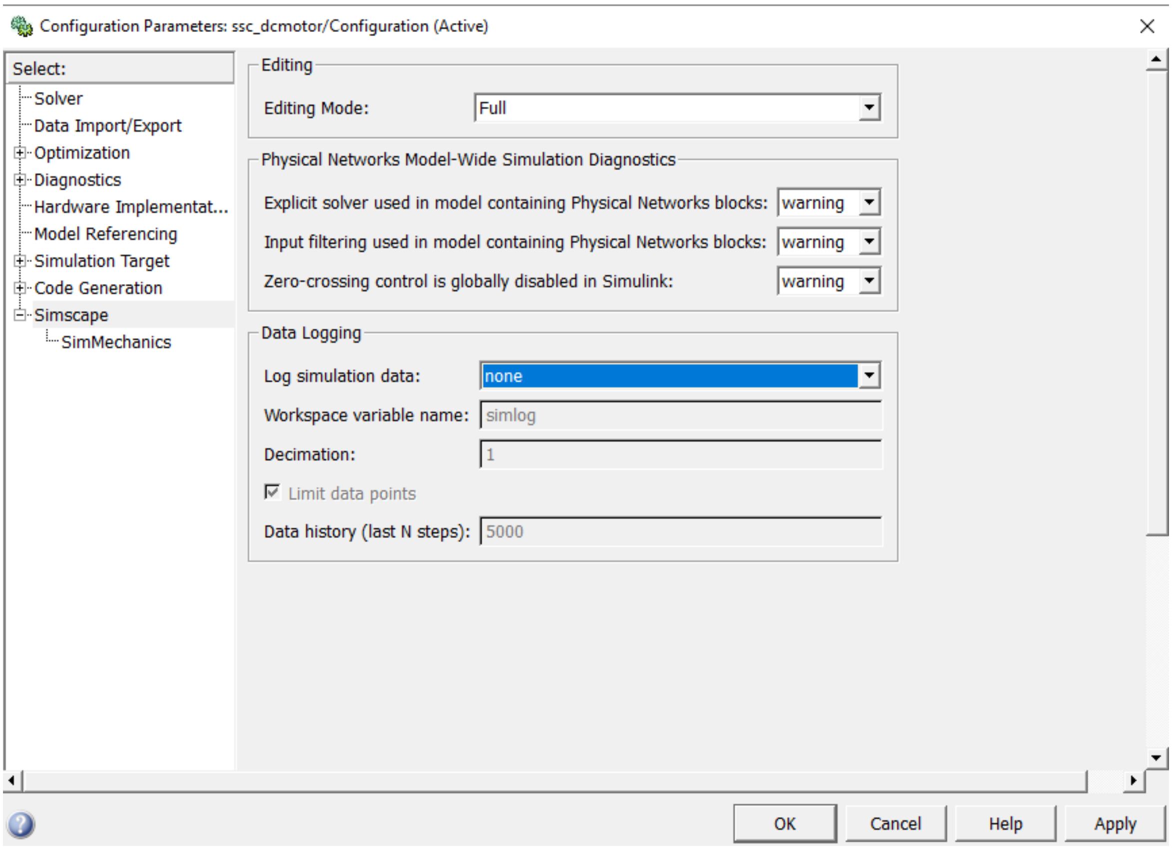

Image 6.10: Configuration Parameters

Click **Apply** and then **OK**. Now all the simulation data is going to be stored in the Workspace variable name “simlog”.

Step 4 → Press **Ctrl+T** to start the simulation.

Step 5 → Go to matlab and type “**ssc_explore(simlog)**”. This is going to open the **Simscape data logging explorer**, the add-on we previously installed. It is a great tool for viewing, logging and sharing the simulation data for our Models.

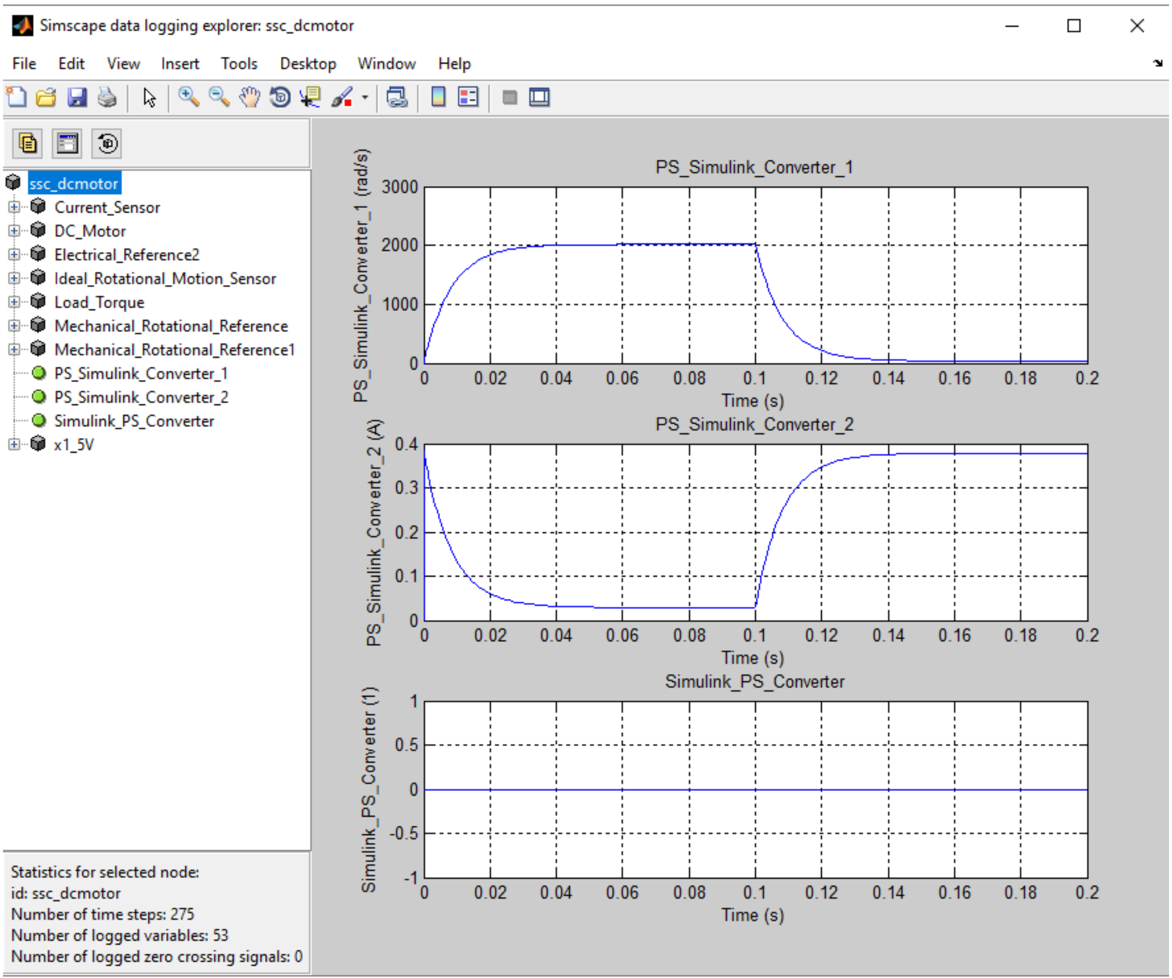

Image 6.11: Simscape data logging explorer

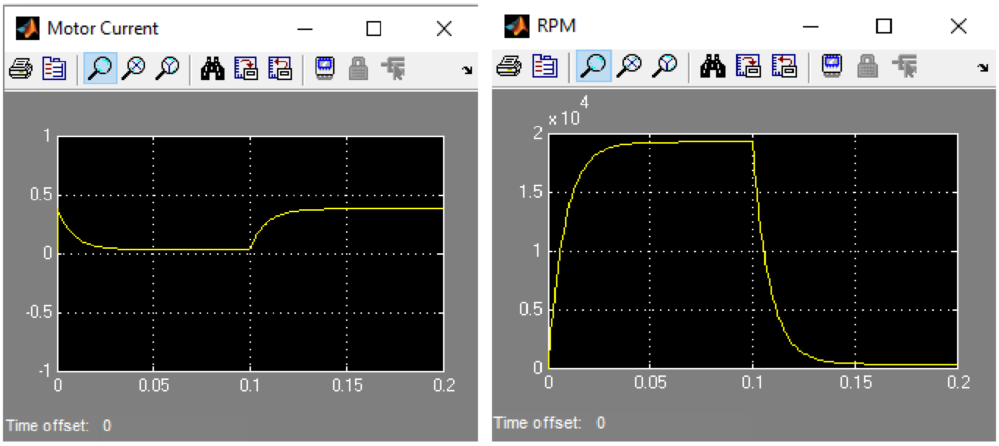

Image 6.12: Motor Current Scope

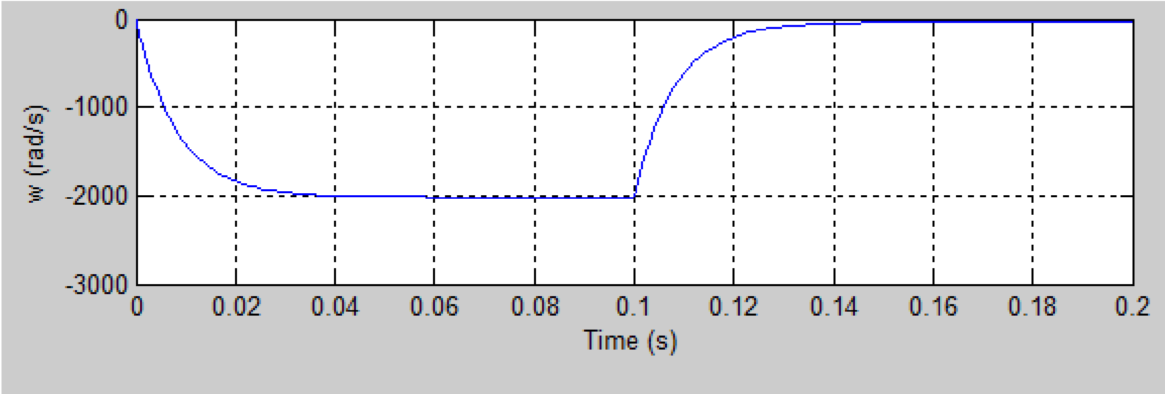

Image 6.13: Load Torque

6.4 Pneumatic Motor

This demo shows how the Simscape Foundation Library Rotational Pneumatic-Mechanical Converter component can be used to approximate the behavior of a pneumatic vane motor. The Pneumatic vane motor block mask parameters are those given on the manufacturer datasheet. The datasheet also states that the motor torque-speed characteristics are linear for fixed supply pressure. To represent this, a linear damping term is placed across the output of the Rotational Pneumatic-Mechanical Converter block. In this simulation, the dynamometer linearly ramps up the speed, and the resulting torque is recorded. The simulation verifies the nominal power as 2.6kW at 3500rpm, and no-load speed as 7000rpm.

Following the steps shown on 5.1 we can run the simulation in order to verify that the nominal power is 2.6kW at 3500rpm and no-load speed as 7000rpm.

Step 1 → Open Matlab and type "**ssc_pneumatic_motor**".

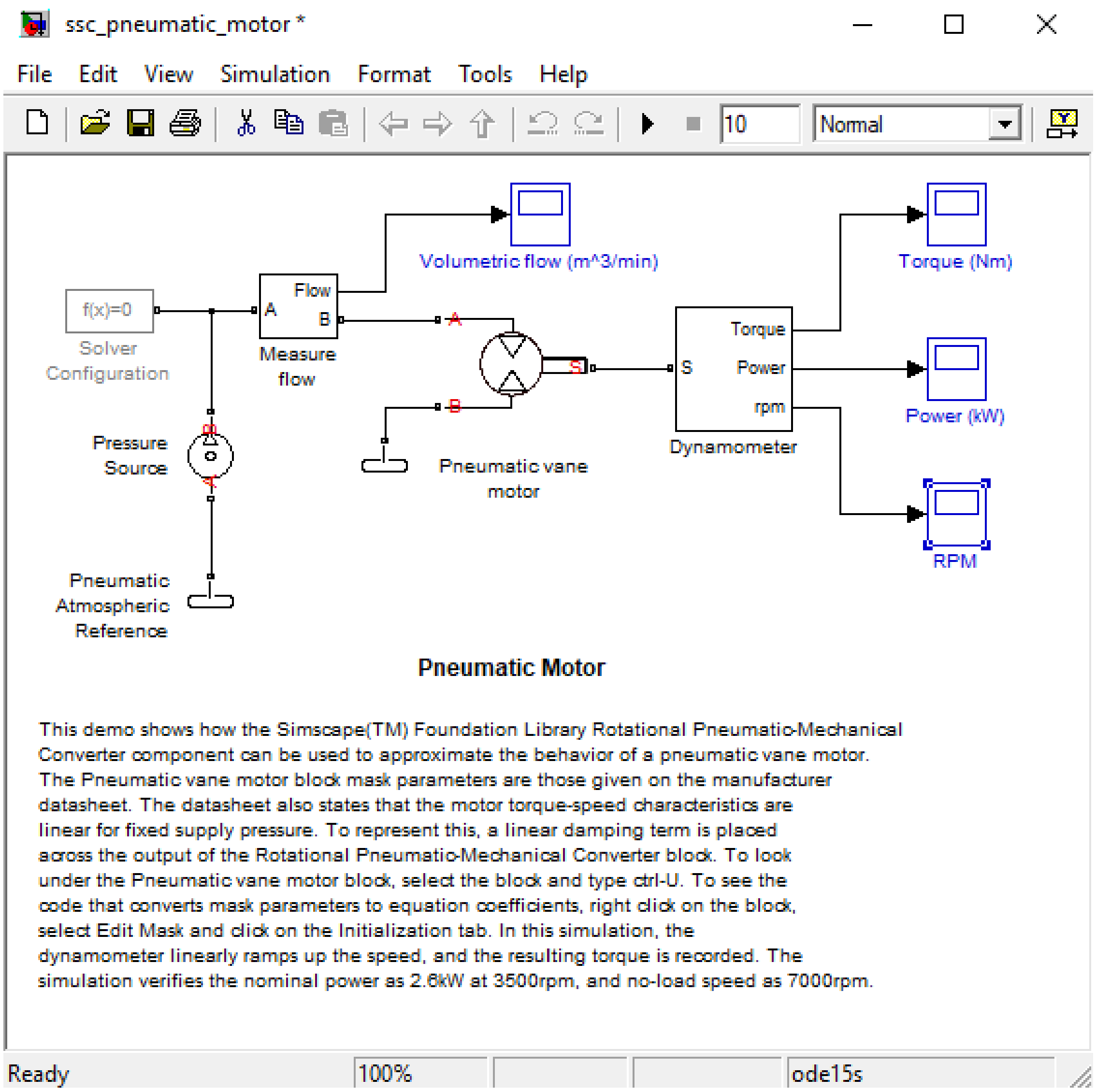

Image 6.14: Pneumatic Motor

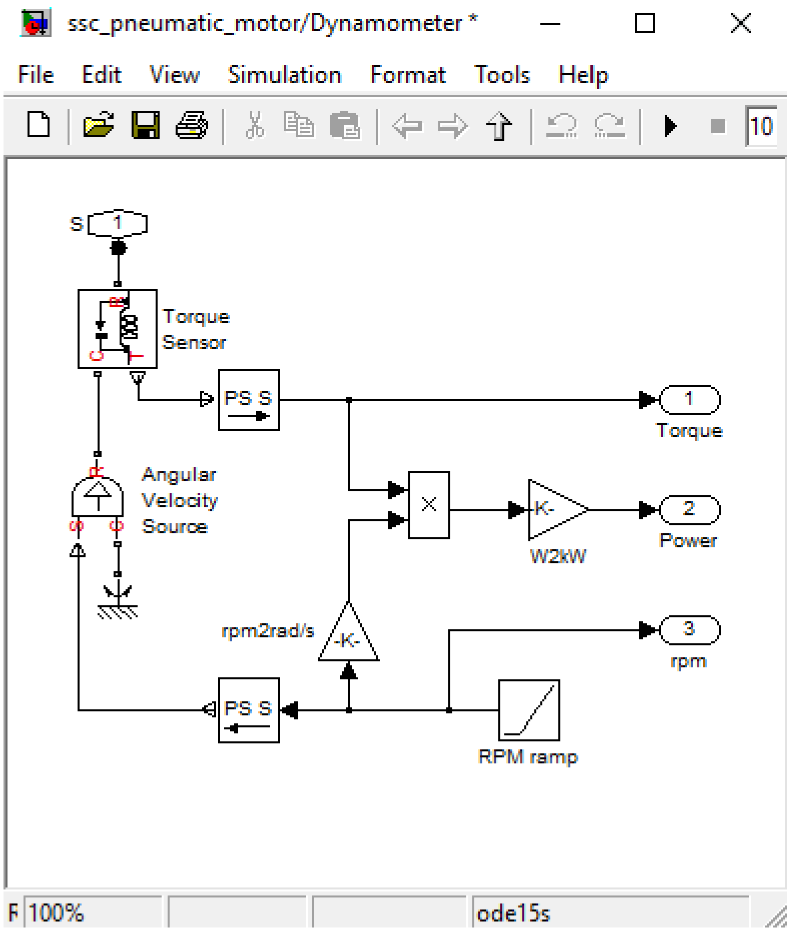

Image 6.15: Dynamometer Subsystem

Step 2 → Go to **Simulation**, **Configuration Parameters**, **Simscape** and change **the Log simulation data** from **none** to **all.**

Step 3 → Run the Simulation(**Ctrl+T**) and then type "**ssc_explore(simlog)**" in the Matlab **Command Window**.

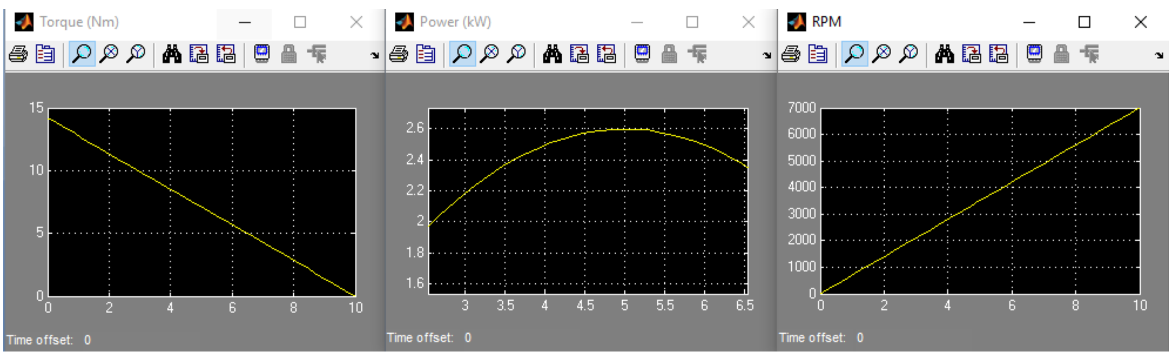

Image 6.16: Pneumatic Motor Scopes

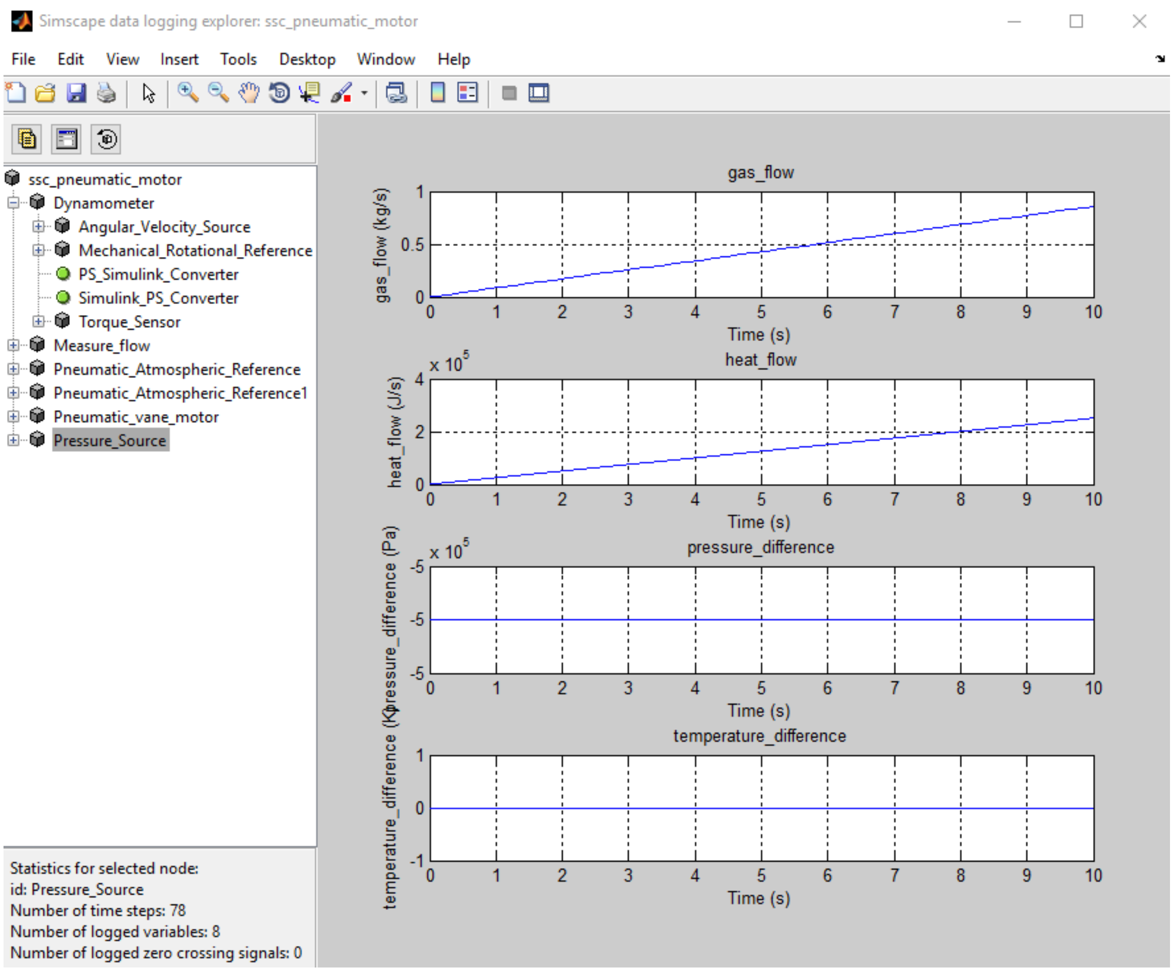

Image 6.17: Pneumatic Motor Pressure Sources

6.5 Modeling a DC Motor

Now, let's create our own DC Motor from start to finish.

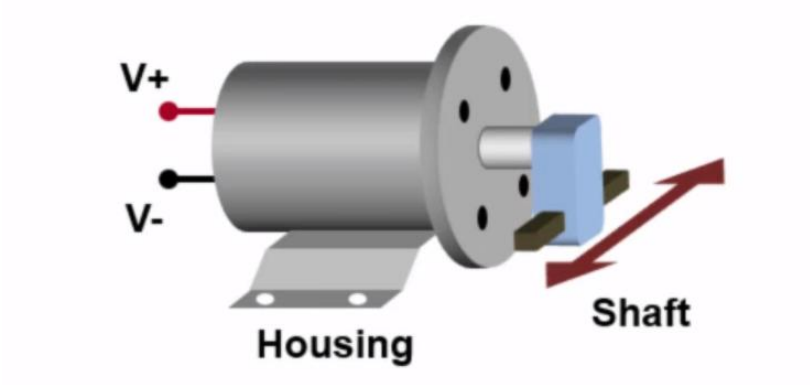

Image 6.18: DC Motor

It's a DC Motor that has two electrical connections and two mechanical connections including one connection that can translate along an axis. Now the problem is Modeling a DC motor with electrical and mechanical effects. Using Simscape we can model this electromechanical system as a physical network. Our model will look like this :

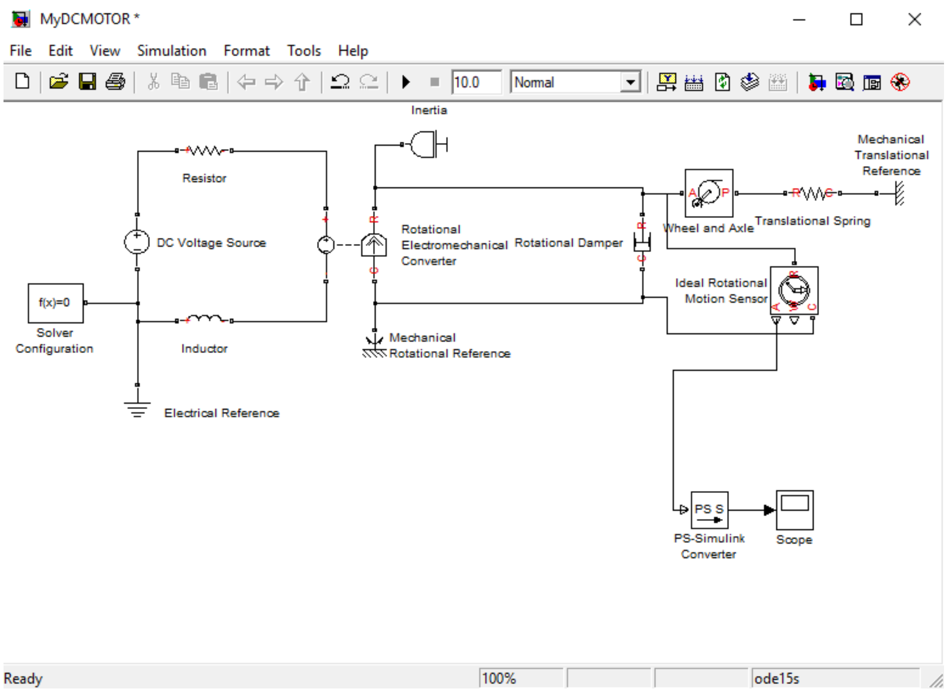

Image 6.19: DC Motor model

When we run the simulation we will see how the motor reacts as it acts against the spring.

Step 1 → Open MATLAB and set the Current folder to : C:\Program Files\MATLAB\R2011a\toolbox\simscape_logging

Image 6.20: Current Folder for Simulation data

Step 2 → Type "**ssc_new**" in the MATLAB **Command Window**. This opens up a Simulink model with the settings recommended for Simscape models. It also adds a few of the blocks commonly used in Simscape. Go ahead and select the upper left Simulink-PS Converter and delete it.

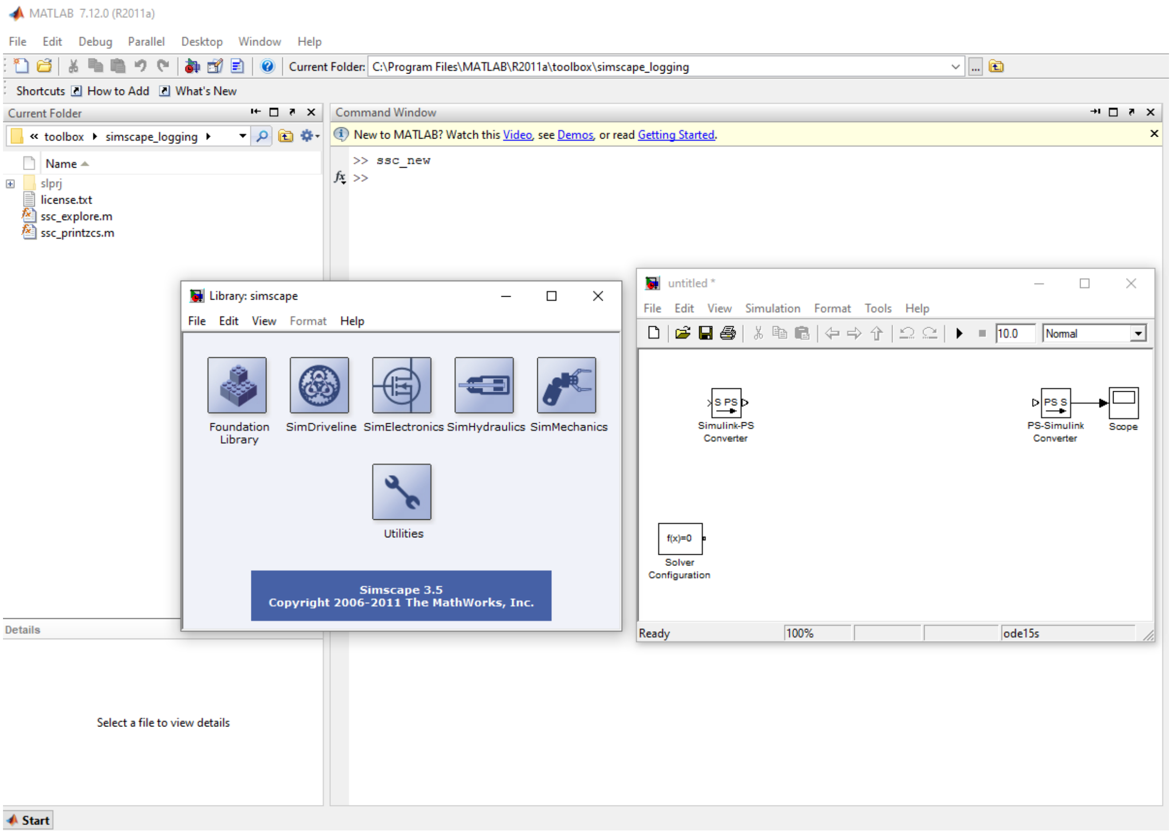

Image 6.21: ssc_new command

Step 3 → The first thing that we'll need to model a DC motor is a **DC Voltage source**. Go to View → Library Browser

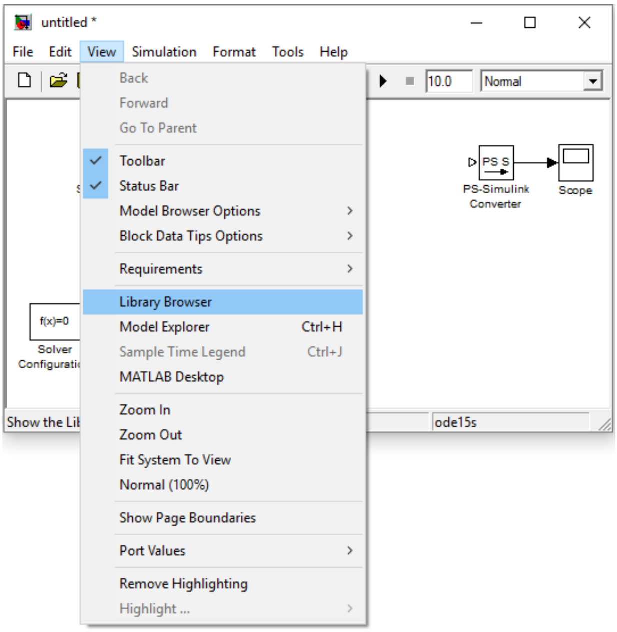

Image 6.22: Opening the Library Browser

Entering a search parameter in the Library Browser will give you a list of models you can use. Go ahead and type “**DC Voltage Source**”.

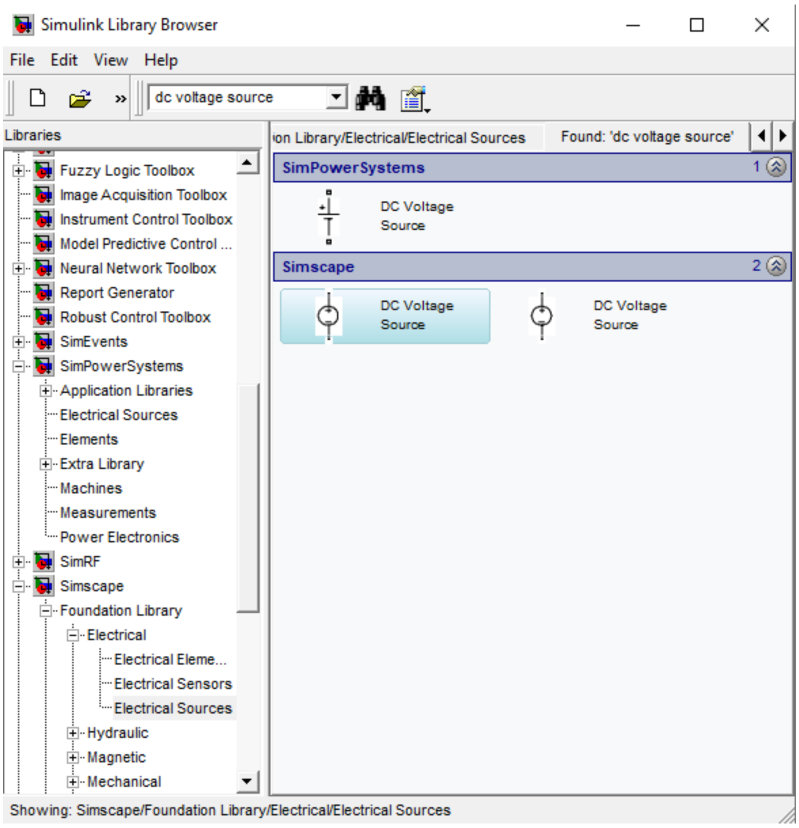

Image 6.23: Library Browser Search

When a selection of the component you want to use is made, in the bottom left corner it shows which Library the selected component is from. In our case, we select the **DC Voltage Source** from the **Simscape Foundation Library**.

Step 4 → To add the **DC Voltage Source** to our model we can either drag the component from the Library Broswer window and drop it to our model or we can left click the component from the Library Browser and add it to the model(Ctrl+I).

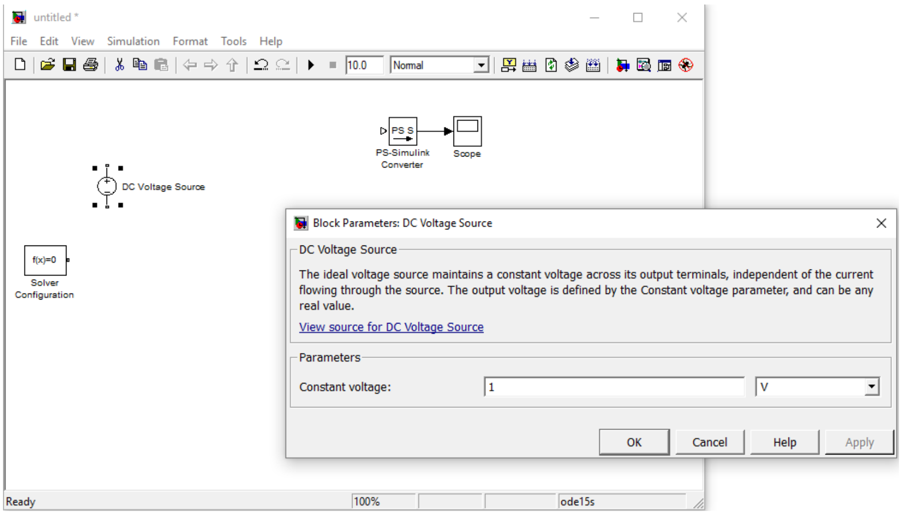

Image 6.24: Adding the DC Voltage Source

After the **DC Voltage Source** is added to our model we can double click on the component and set the **Constant voltage** to **5 Volts** and press OK.

Step 5 → Following the same procedure go ahead and search for a **resistor**, select it and add it to our model.

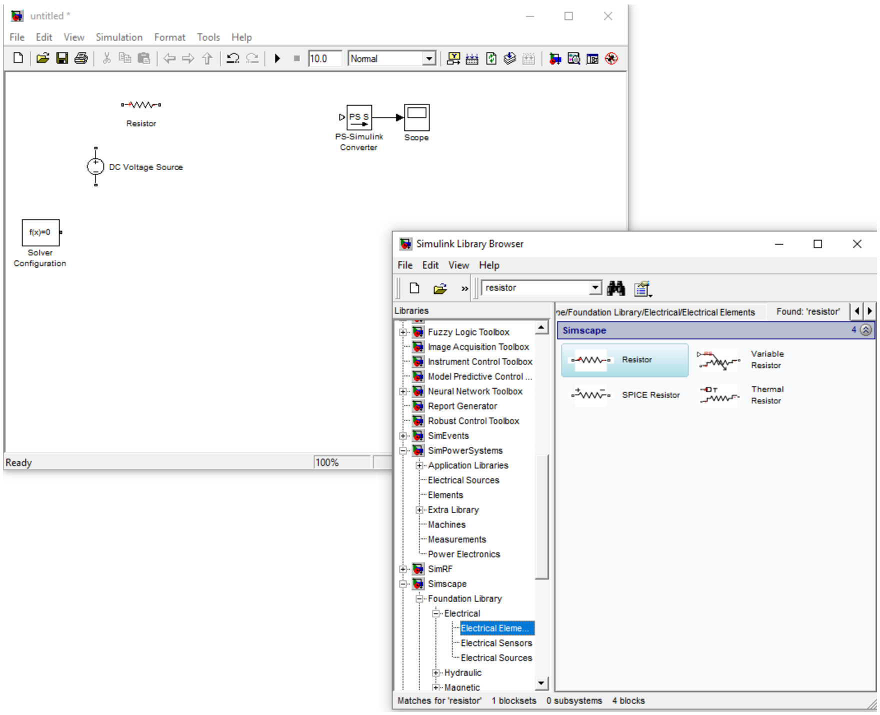

Image 6.25: Adding a resistor

Step 6 → Now we need a block to convert between electrical and mechanical energy. Search for "**electromechanical**" and chose the **Rotational Electromechanical Converter** and add it to the model.

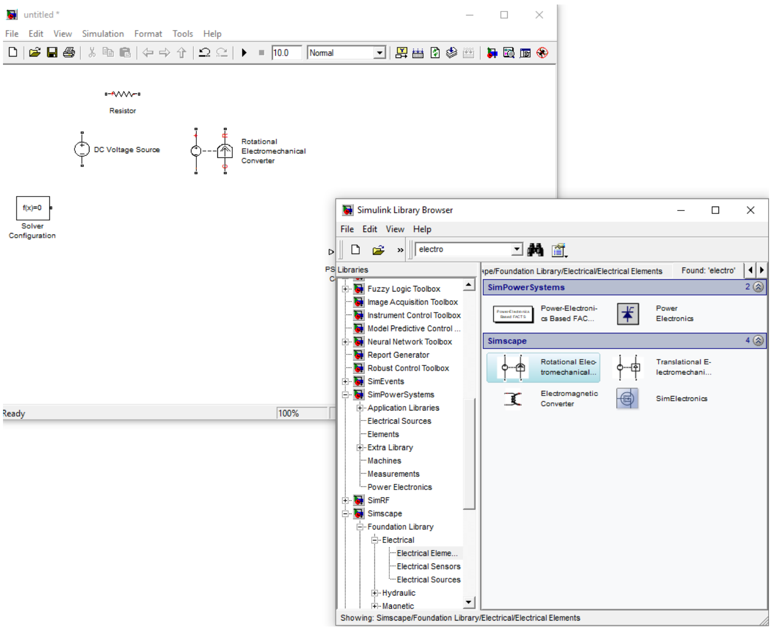

Image 6.26: Adding a Rotational Electromechanical Converter

Step 7 → Adding an Inductor. Search for **Inductor** and chose the one from the Simscape family.

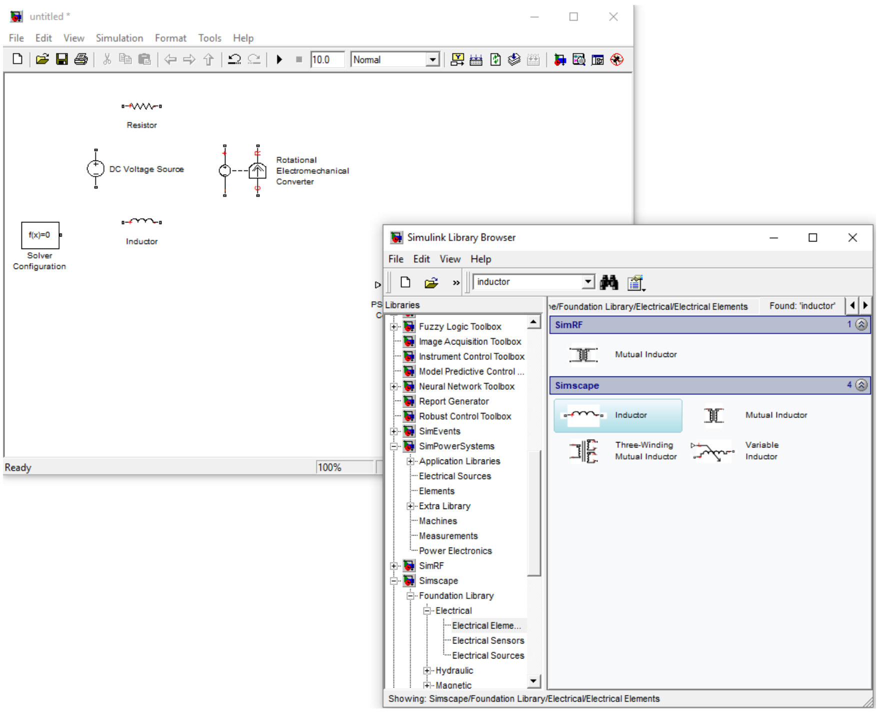

Image 6.27: Adding an Inductor

Step 8 → Adding a ground. Search for **Electrical Reference** and add to the model.

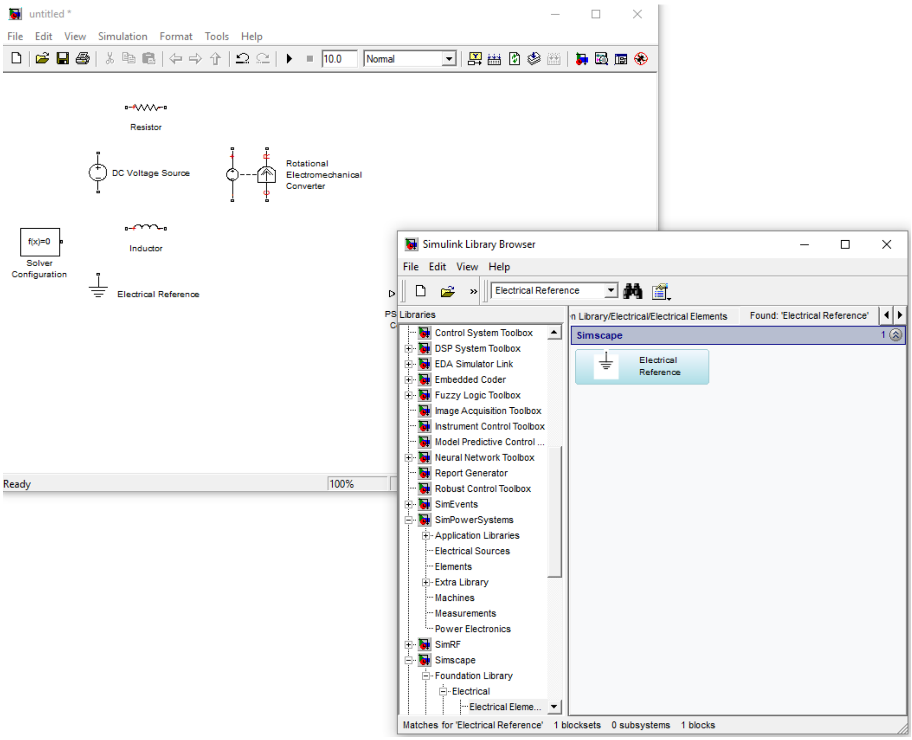

Image 6.28: Adding an Electrical Reference

Step 9 → Lets create the physical connections by wiring the components.

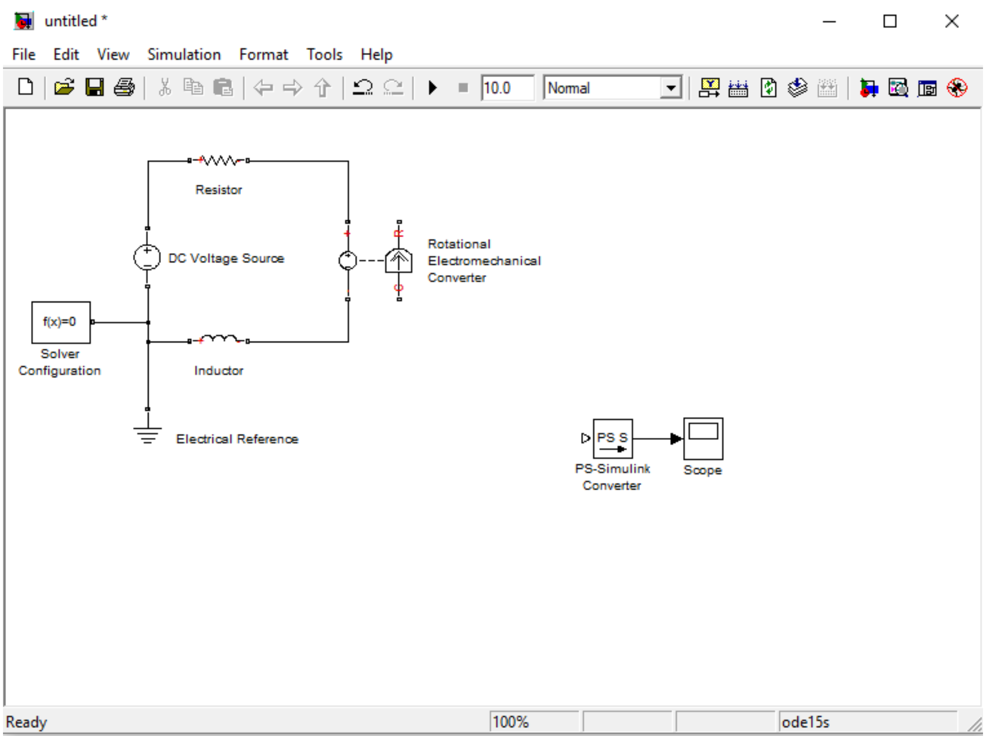

Image 6.29: Creating physical connections

Simscape uses solver technology above and beyond what's available in normal Simulink. To have access to some of those settings we will connect the solver Configuration Block.

Step 10 → Now we are going to add the mechanical portion of the model. We will attach the housing of the motor to a point fixed in space. To do that, search for **Mechanical Rotational Reference** and connect it to the model as seen below.

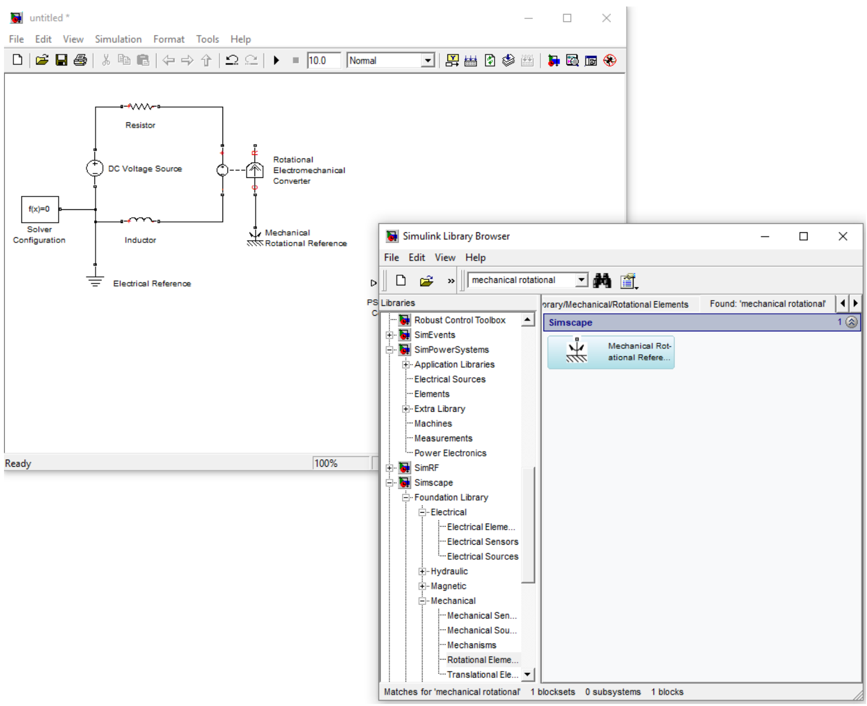

Image 6.30: Adding Mechanical Rotational Reference

Step 11 → Lets connect the shaft with an Inertia. Search for "**Inertia**".

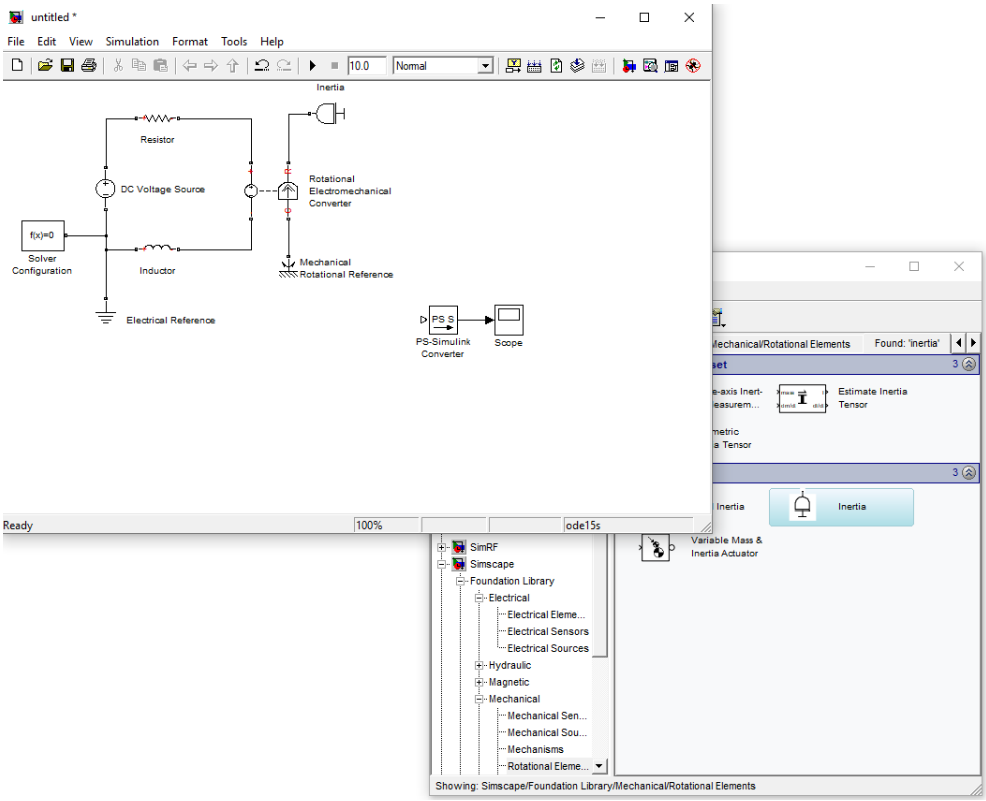

Image 6.31: Adding Inertia to the shaft

Note: To rotate any component, select it and press **Ctrl+R.**

Step 12 → To model the viscous friction at the bearing of our motor we will add a Damper. Search for **Damper** and select the **Rotational Damper**, rotate as shown and connect to the **Converter**.

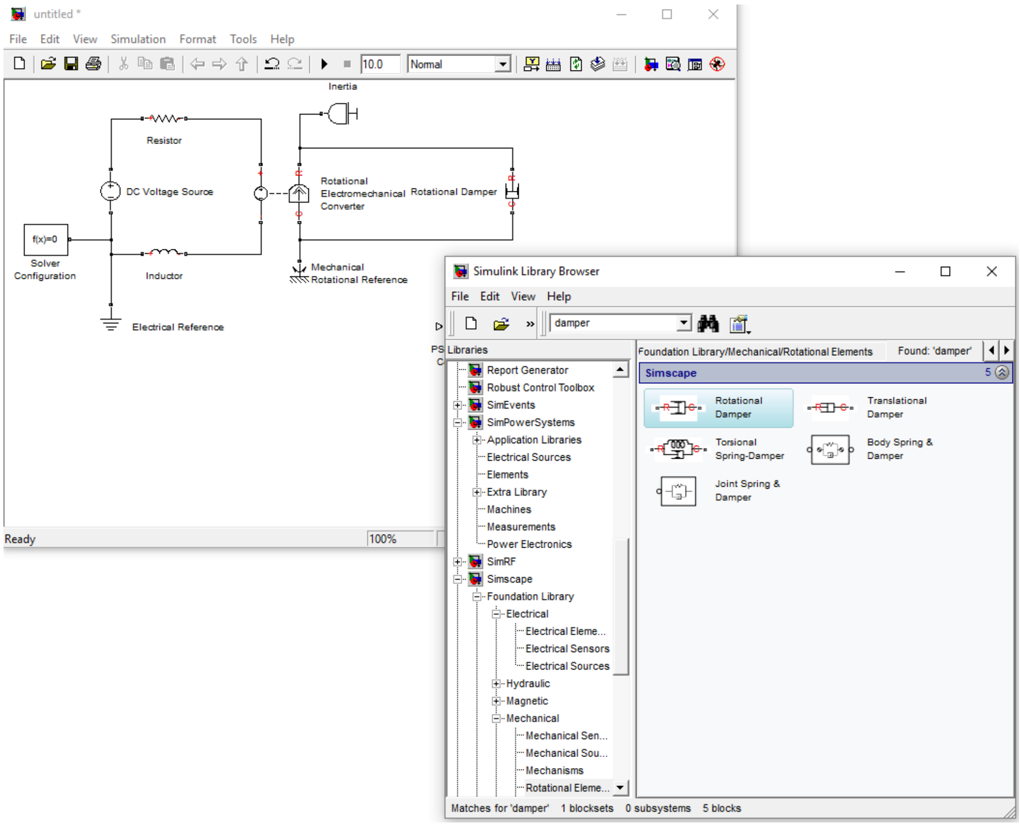

Image 6.32: Adding and connecting a Rotational Damper

Step 13 → To convert the rotational motion of the shaft to translational motion. To do that, search for the **Wheel and Axle** block. Add and connect to the model.

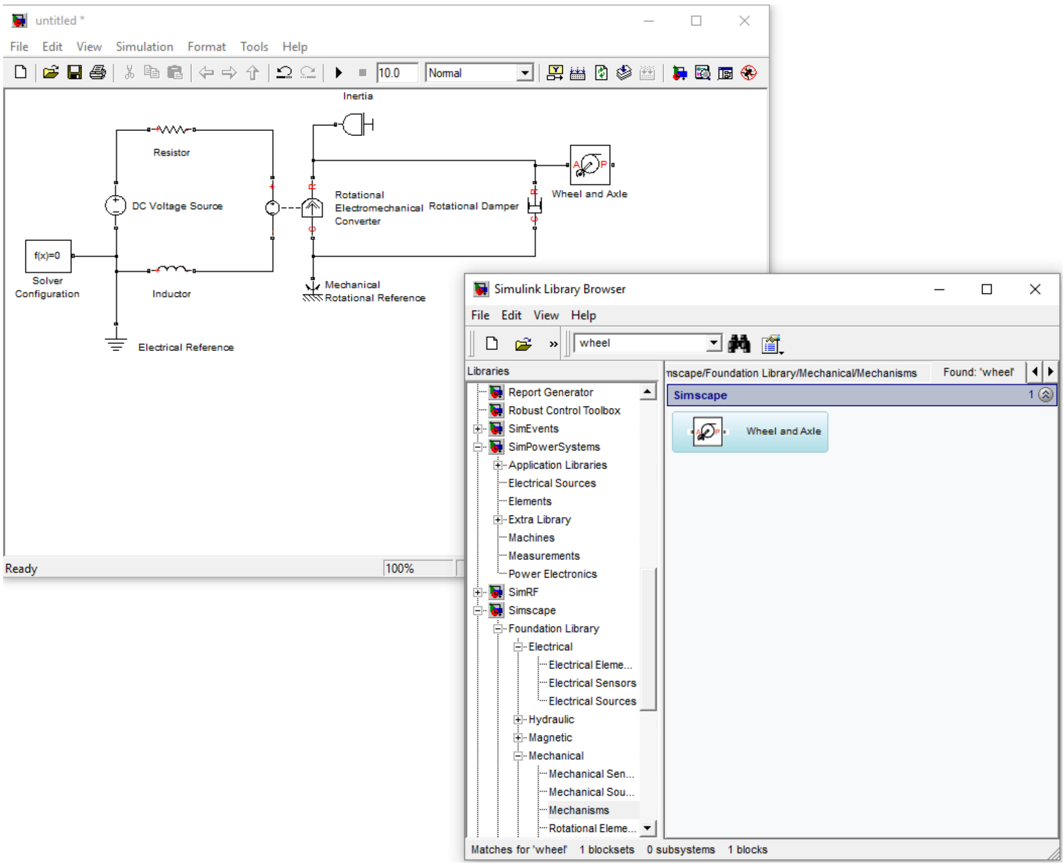

Image 6.33: Adding and connecting a Wheel and Axle

Step 14 → We want the motor to act against the spring so we will insert a Spring. Search for "**spring**" and add the **Translational Spring** to the model.

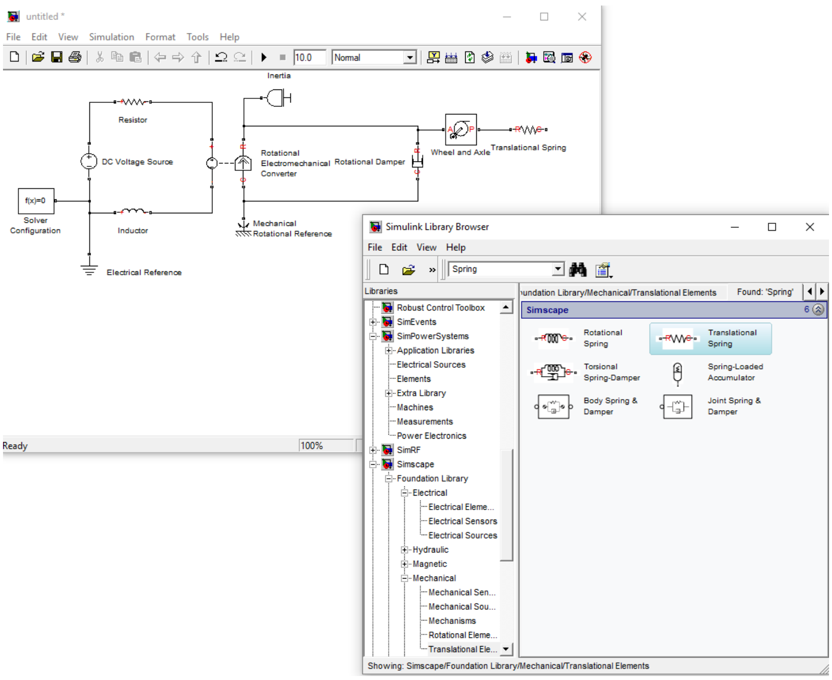

Image 6.34: Adding and connecting a Translational Spring

Step 15 → Connect the Spring to a point fixed in space. Search for **Mechanical Translational Reference** and add it to the model.

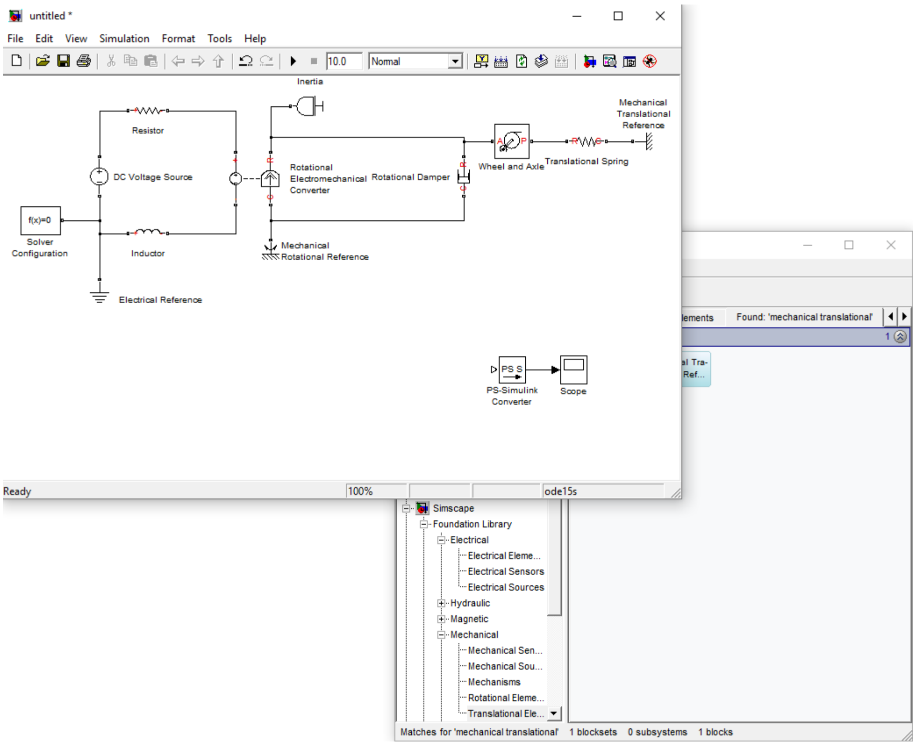

Image 6.35: Adding and connecting a Mechanical Translational Reference

With this the model is complete and we can run the model and simulate.

Step 16 → Save the model by going to **File → Save As** and chose a name for the model. Once we have saved the model we can run the simulations. Go to **Simulation → Configuration Parameters**… From the left section locate **Simscape → Data logging**. Change the Log simulation data from **none** to **all**. Click **apply** and **OK**. Now we can start the simulation.

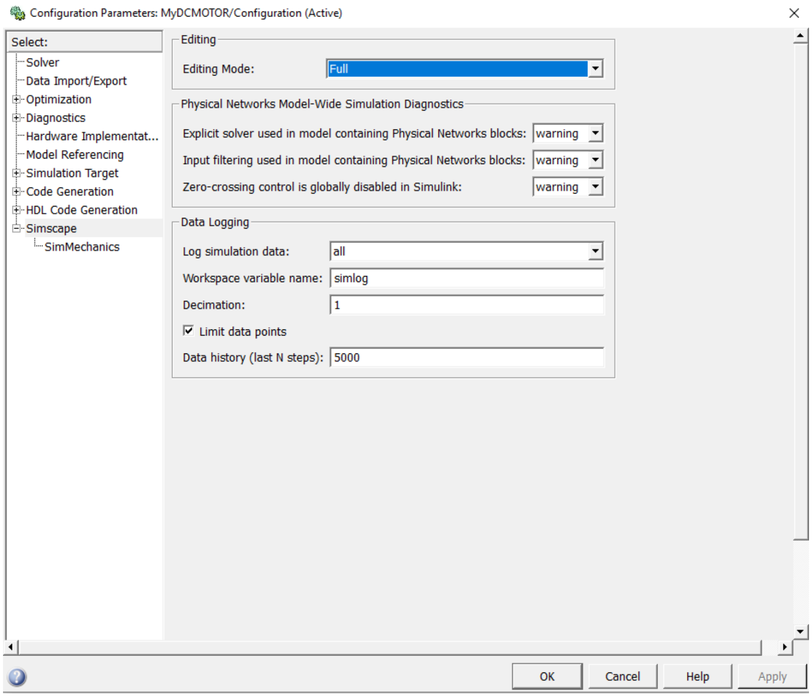

Image 6.36: Simulation data logging

Go to the model and Start the simulation by pressing the start button.

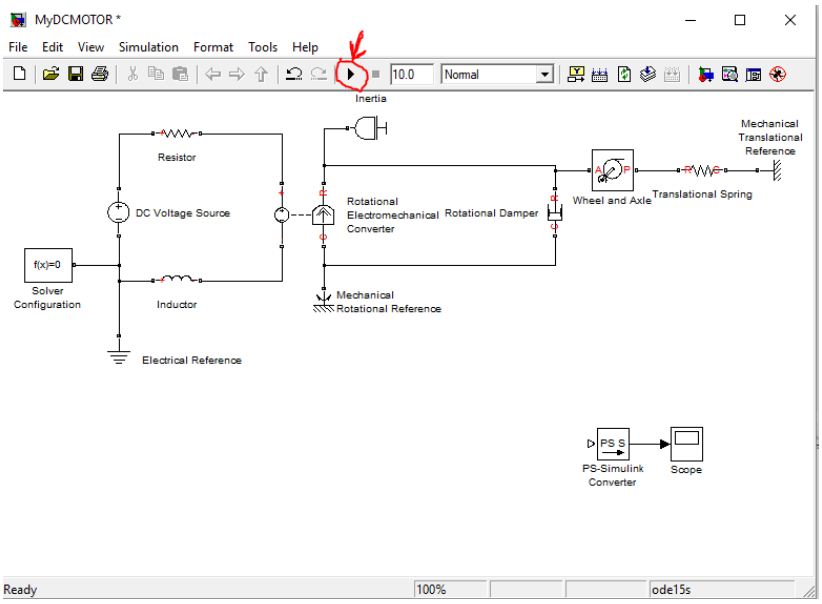

Image 6.37: Starting the Simulation

Step 17 → In order to view the simulation data go to the MATLAB Command Window and type "**ssc_explore(simlog)**". This will open up the **Simscape data logging explorer**. In this window we can explore the results of our simulation through the tree browser.

Here we can see the speed of the shaft as it settles out to 0 rad/s.

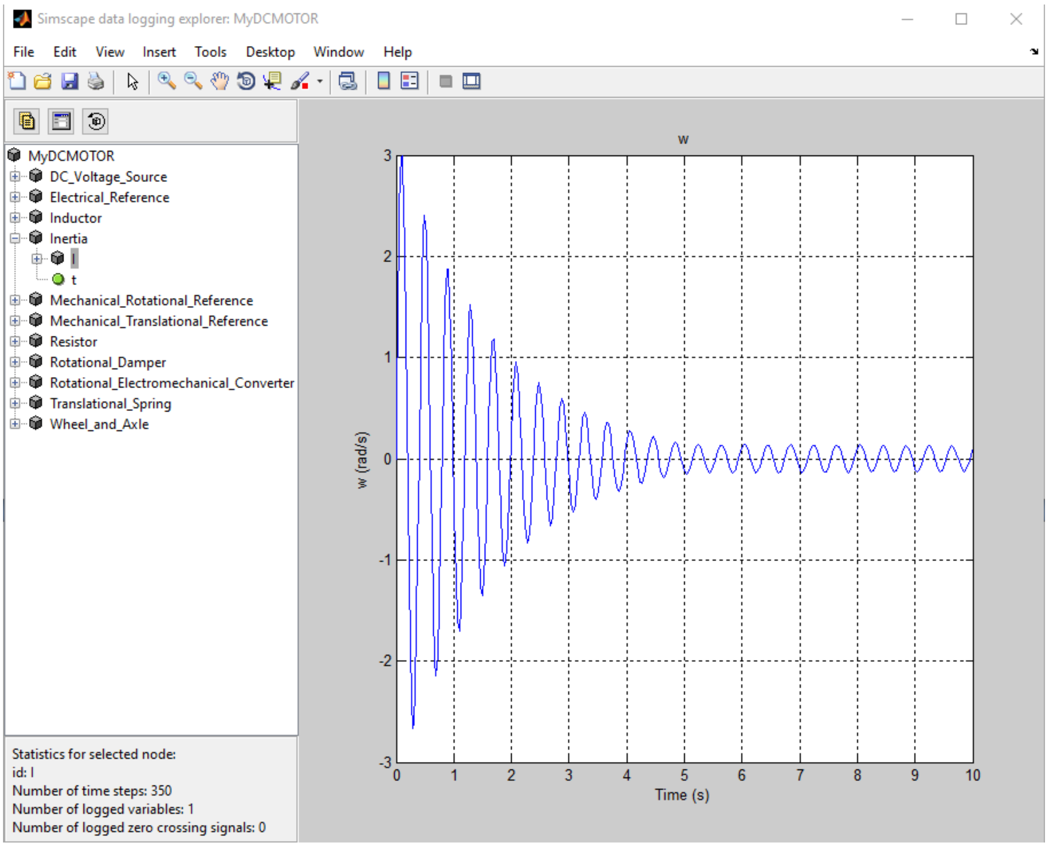

Image 6.38: Simulation Data

We can look at electrical quantities such as the current going through the resistor:

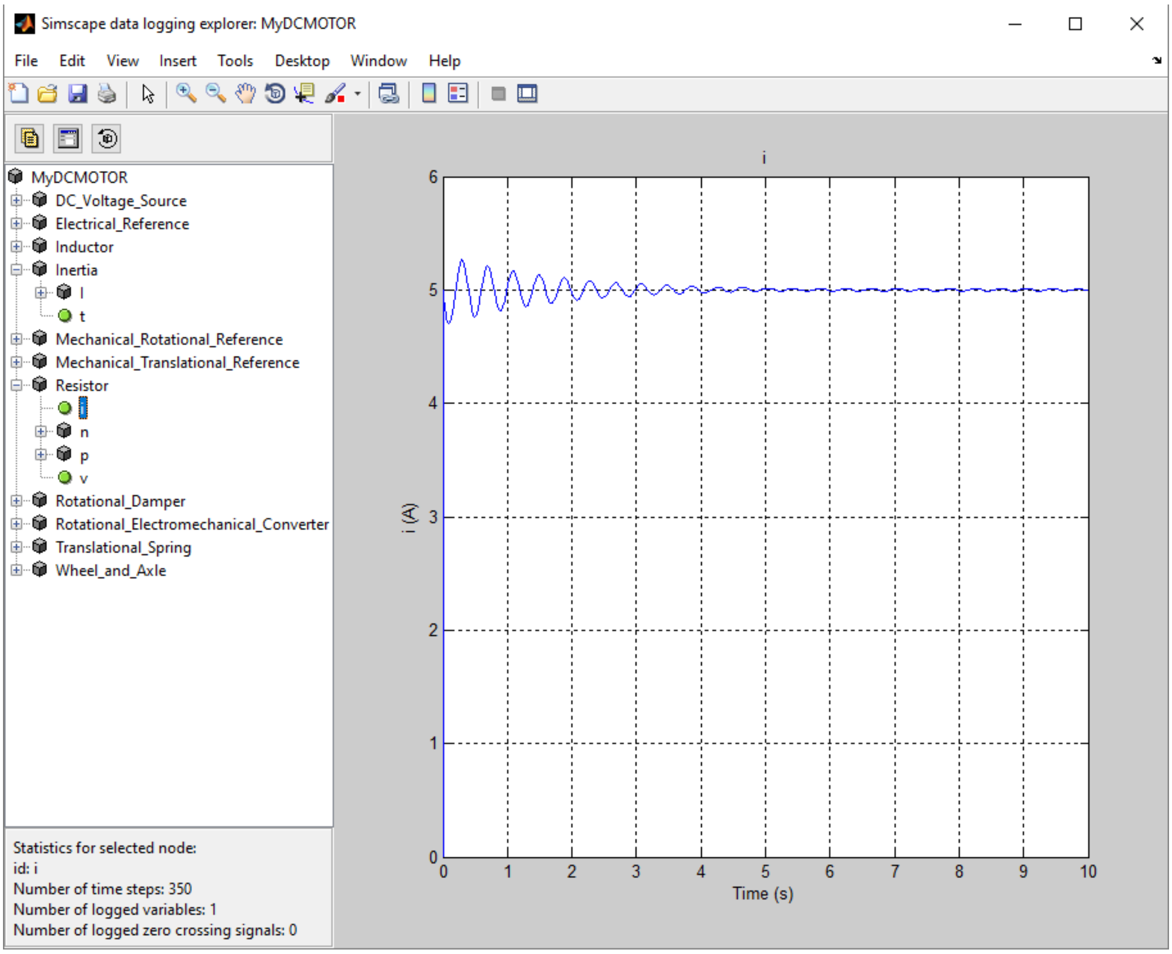

Image 6.39: Simulation Data

We can see by how much the spring was compressed by selecting the **x** variable in the translational spring. It compressed to about 0.01 meters.

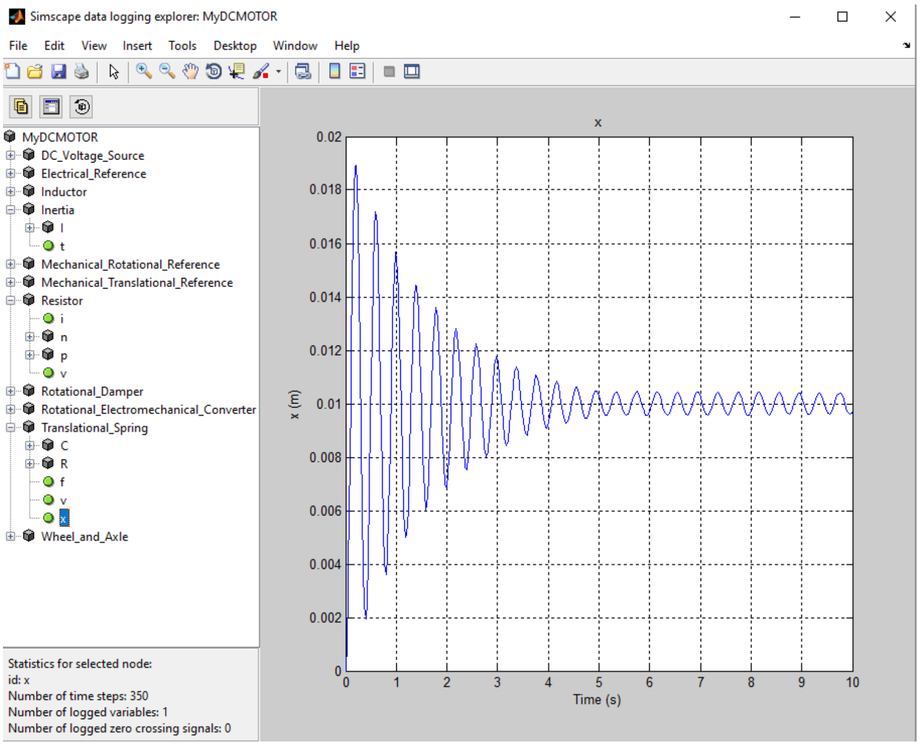

Image 6.40: Simulation Data

Step 18 → We can increase the distance that the motor shaft moves by reseting the stiffness of the spring. Go to the model and locate the **Translational Spring**. Double click it and change the values of the Spring rate to **500**. Click **Apply** and **OK**.

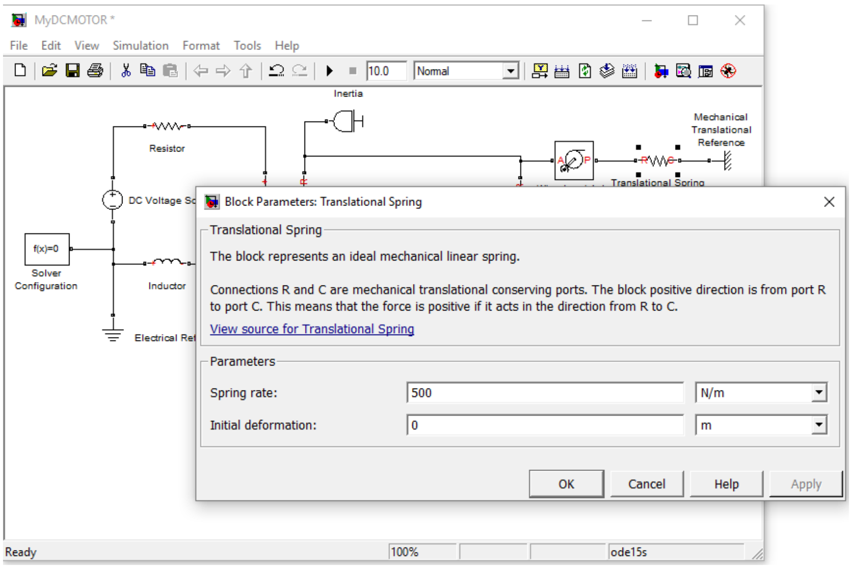

Image 6.41: Changing the Spring rate

Now run the simulation again and reload the logged data.

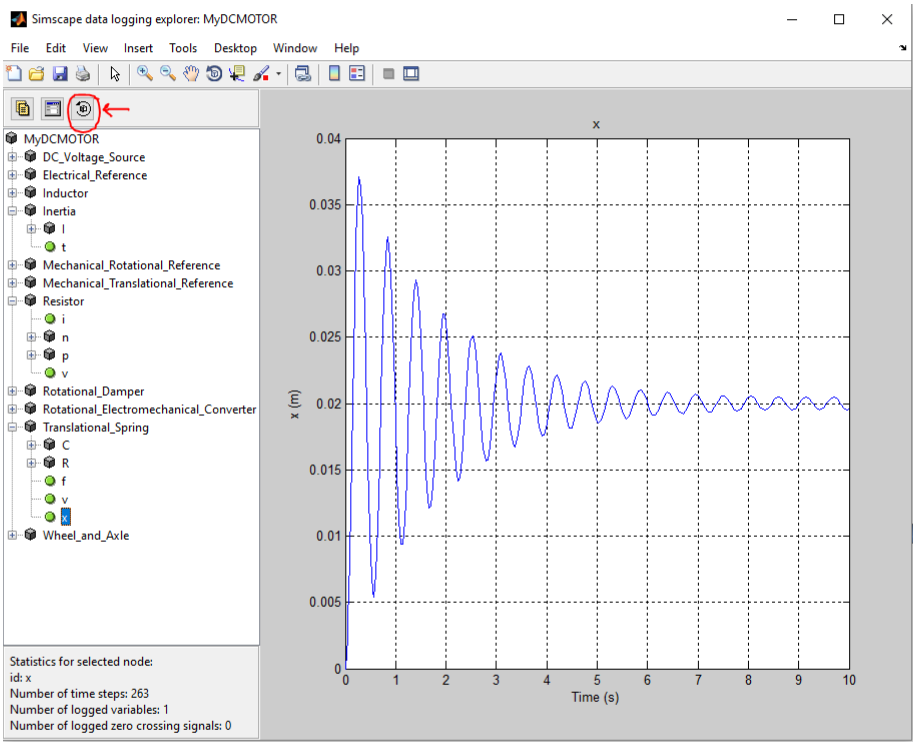

Image 6.42: Reloading logged data

We can see that the spring now compresses 0,02 meters because we halved the stiffness of the spring.

Step 19 → To view the results of this simulation on a Simulink scope we can use sensor blocks. Go to the model and open up the Library Browser. Search for **Motion Sensor** and add the **Ideal Rotational Motion Sensor**. Connect as seen below.

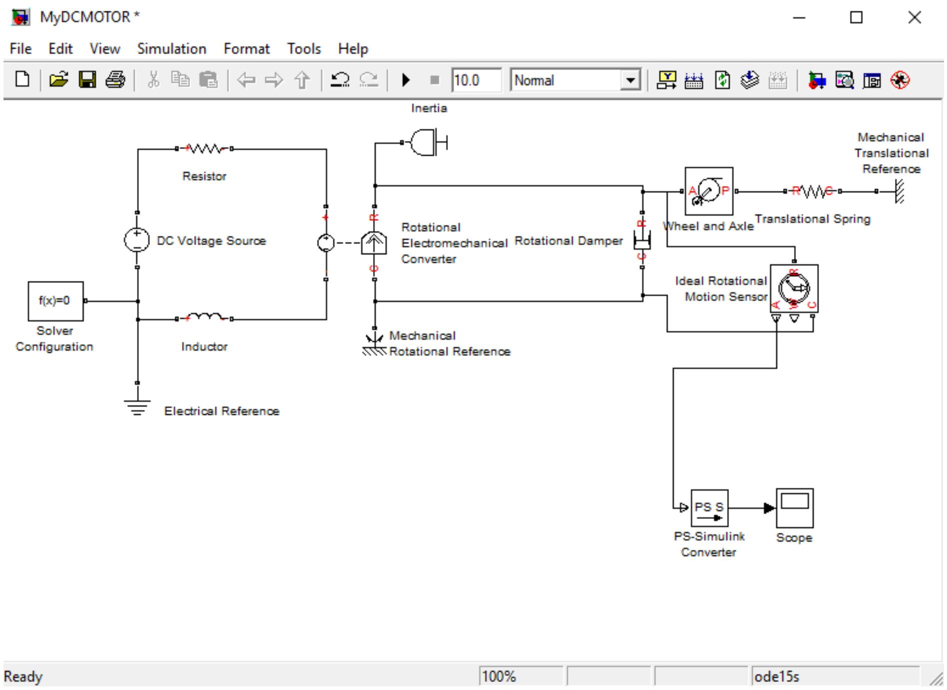

Image 6.43: Simulink Scope connection

Double click on the PS-Simulink Converter and change the Output signal unit to "**deg**".

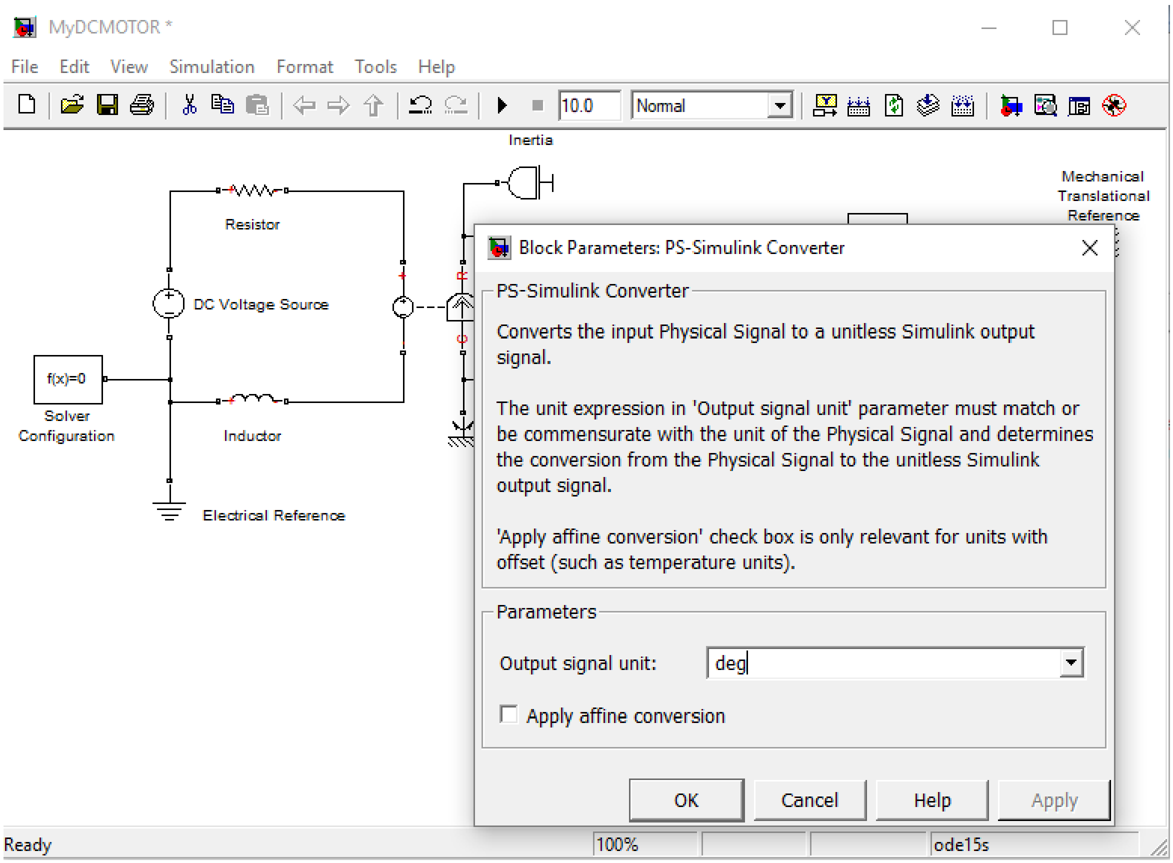

Image 6.44: Changing the output signal unit

Press **Apply** and **OK**. Double click on the Scope and run the simulation again. Once the simulation is done, left click on the Scope window and click **Autoscale**. Here we can see how much the shaft turns, in this case we see that it settles at about 22,5 degrees.

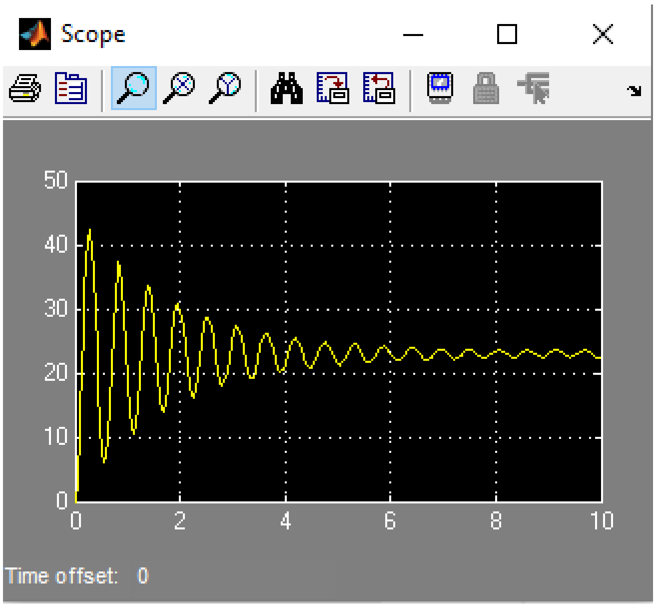

Image 6.45: Simulink Scope

BIBLIOGRAPHY

[1]https://en.wikipedia.org/wiki/MathWorks

[2]https://www.mathworks.com/products/simscape.html

[3]https://en.wikipedia.org/wiki/MATLAB

[4]http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatis matlab.htm

[5]https://www.diva-portal.org/smash/get/diva2:218737/FULLTEXT01.pdf

[6]https://www.mathworks.com/help/physmod/simscape/index.html