BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

DEVELOPMENT OF DATABASE OF MECHATRONIC MODELS IN SIMSCAPE

NÁVRH SADY MECHATRONICKÝCH MODELŮ V SIMSCAPE

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE
Zaid Al-Dailami

SUPERVISOR
VEDOUCÍ PRÁCE
Ing. Michal Bastl

BRNO 2020

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

Specification Bachelor's Thesis

Department:	Institute of Solid Mechanics, Mechatronics and Biomechanics
Student:	**Zaid Al-Dailami**
Study programme:	Applied Sciences in Engineering
Study branch:	Mechatronics
Supervisor:	**Ing. Michal Bastl**
Academic year:	2019/20

Pursuant to Act no. 111/1998 concerning universities and the BUT study and examination rules, you have been assigned the following topic by the institute director Bachelor's Thesis:

Development of database of mechatronic models in Simscape

Concise characteristic of the task:

Simscape is a MATLAB / Simulink tool that enables acausal modeling. The main aim of the thesis is to create a database of simple mechatronic models and test the tool in relation to the mechatronic design of the product. It is supposed to focus on model optimization and automatic code generation for HIL simulation.

Goals Bachelor's Thesis:

1) Create a written introduction that briefly describes the Simscape tool and its options.
2) Create a set of selected mechatronic models. Evaluate the Simscape options for the parameter identification and model optimization.
3) Verify the Simscape options for HIL simulations. Specifically, automatic code generation and fixed step solvers

Recommended bibliography:

MathWorks. [US], Simscape user‘s guide, online: http://ww2.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_ug.pdf.

MILLER, Steve; WENDLANDT, Jeff. Real-time simulation of physical systems using Simscape. MATLAB News and Notes, 2010, 1-13.

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year 2019/20

In Brno,

L. S.

prof. Ing. Jindřich Petruška, CSc. Director of the Institute	doc. Ing. Jaroslav Katolický, Ph.D. FME dean

Abstrakt

Tato práce se zabývá především vývojem databáze modelů mechatronik v nástroji Simscape. Je rozdělena do tří hlavních částí a to následně:

První část popisuje nástroj Simscape a jeho možnosti . Práce poskytuje návrh, jak nastroj použít při vývoji modelů a jak se liší od jiných způsobů modelování. Druhá část obsahuje některé mechatronické modely založené na Simscape. Některé z modelů jsou také vytvořené pomocí standardního modelování Simulinku, a ty lze porovnat s dříve uvedenými modely. Třetí část je zaměřena na jeden z dříve navržených modelů se stejnosměrným motorem. Tento model se používá k vyhodnocení, odhadu a optimalizaci skutečných parametrů stejnosměrného motoru, obsahuje také popis, jak vygenerovat c kód z modelů Simscape, abychom jej použili v HIL.

Abstract

This thesis is mainly about development of mechatronics models database in Simscape. It is divided into three main parts as follows:

The first part describes the Simcspae tool and its options. It gives an abbreviation view on how it can be used in developing models and how it differs from other ways of modeling. The second part contains some mechatronics models based on Simscpae. Some of them are also modeled with standard modeling of Simulink, so they are compared together. The third part focuses on one of the models designed before which is a brush DC motor model. This model is used to evaluate, estimate, and optimize real DC motor parameters. It also contains a description of how to generate c code from Simscape models, so it can be used for HIL.

Klíčová slova

Simscape, Mechatronické modely v Simscape, Model stejnosměrného motoru, Vyhodnocení a optimalizace parametrů modelů, HIL with Simcape

Keywords

Simscape, Mechatronics models in Simscape, DC motor model, Evaluation and optimization of models’ parameters, HIL with Simcape

Bibilographic Citation

AL-DAILAMI, Zaid. *Development of database of mechatronic models in Simscape.* Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/121440. Bachelor's Thesis. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Institute of Solid Mechanics, Mechatronics and Biomechanics. Supervisor Michal Bastl.

Declaration

I hereby declare that I have written my bachelor thesis with the title (Development of database of mechatronic models in Simscape) individually with the instructions of my supervisor. I declare that I have not violated any copyrights and all information used from other sources and literature are mentioned in the Reference page.

Acknowledgment

I would like to express my sincere appreciation for the great effort that my supervisor Ing. Michal Bastl exerted. I would like to thank his cooperation and instructions that helped me to accomplish this thesis.

I also faithfully thank my parents and my family members who always have been the main motivation for me

Finally, I would like to thank all my friends who supported me especially Al-Qubati Ali, and all my teachers in VUT especially Ing. Martin Brablc for his help and support. My thanks and gratitude also for any person who helped me complete this thesis.

Content

1 INTRODUCTION 7

2 OBJECTIVES...8

3 SIMSCAPE TOOL 9

3.1 What is Simscape?..9

3.2 Types of Variables 9

3.2.1 In terms of connectivity when measured...9

3.2.2 In terms of sign (direction) 10

3.3 Elements of Simscape..10

3.4 Differences between Simscape modeling and standard Simulink modeling 11

3.5 Advantages and features of Simscape 11

3.6 Ports and lines in Simscape 12

4 OPTIONS OF SIMSCAPE 13

4.1 Simscape Block Libraries...13

4.1.1 Main libraries...13

4.1.2 Add-ons libraries...14

4.1.3 Accessing the Block Libraries:...15

4.2 Connecting Simscape with Simulink blocks:..15

4.3 Solver Configuration Block 15

4.4 Recommended solvers of Simulink for Simscape..16

4.5 References of Simscape domains...17

4.6 General rules in Simscape Modeling[1] 17

4.7 Additional features in Simscape 18

4.7.1 Logging Simulation Data in Simscape..18

4.7.2 Preparing the model for other users...19

4.7.3 Source code of Simscape 19

4.7.4 Finding causes of slow simulation 20

5 MECHATRONICS MODELS...21

5.1 Brush DC motor model 21

5.2 Brushless DC Motor Model...23

5.3 Mass Spring Damper System Model 24

5.4 Additional Models..27

6 PARAMETERS’ EVALUATION AND MODEL OPTIMIZATION..................................29

6.1 Calculating parameters of DC motor 29

6.2 Evaluation of parameters 33

6.3 Estimation of parameters 34

6.4 Model optimization 37

7 REAL-TIME SIMULATION 39

7.1 Code Generation 39

7.2 HIL Simulation 39

7.3 Verifying Simscape Models for HIL 40

7.4 HIL Experiment with MF624 and PIC Kit 41

7.4.1 HIL experiment to control Angular Velocity 41

7.4.2 HIL experiment to control Position 43

8 CONCLUSION 45

REFERENCES 46

LIST OF ATTACHMENTS 47

LIST OF FIGURES 48

LIST OF TABLES 50

LIST OF ABBREVIATIONS 50

NOMENCLATURE 51

1 INTRODUCTION

Modeling systems in engineering is being wide and crucial in the last 30 years. In fact, there is no industry nowadays that produces its products without going through different stages of modeling until exceeding the final suitable and desirable product. One efficient environment of making mathematical and physical models is MathWorks®. MathWorks® has many tools in MATLAB® and Simulink® that facilitate dealing with different models of different systems. One effective tool in Simulink® is called Simscape™. Simscape™ might be more preferred to use than other tools in Simulink® due to its simplicity, versatility, and efficiency. Besides, it is based on the physical network approach. In other words, no need for such complicated mathematical models and differential equations. Moreover, Simscape is capable of modeling and simulating different systems like Electrical, Magnetic, Gas, Hydraulic, Mechanical, Thermal systems, and so on. All system domains are illustrated in Fig. 1. All these factors have been a motivation for me to put more effort into Simscpae and make a brief analysis study of it and its applications. Also, some models of different systems are being studied and simulated in this thesis.

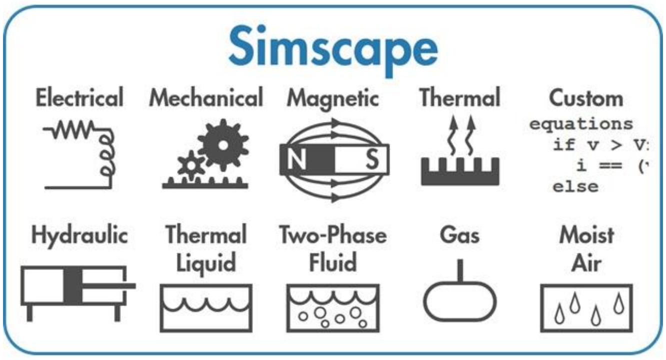


Fig. 1: Simscape domains [14]

After that one model is being chosen to be studied and analyzed by HIL. This model is a DC motor. HIL for the DC motor model is being accomplished by different steps. Firstly, parameters of a real DC motor are being calculated with the help of Double Drive Motor (MF 624) in MechLab by RCP. The DC motor parameters then are reevaluated and estimated with the help of Simulink Design Optimization. Next, these parameters are used in the Simscape DC motor model. Then, the Simscape DC motor model signals are being generated with the help of the MF624 card. At the same time, the controller of the DC motor is being generated to an MCU (PIC EduKit). HIL is being done to the plant and the controller. Finally, the results of MIL and the results of HIL are being compared and analyzed.

2 OBJECTIVES

The three main goals of this project are:

1) Create a written introduction that briefly describes the Simscape tool and its options.

2) Create a set of selected mechatronic models. Evaluate the Simscape options for the parameter identification and model optimization.

3) Verify the Simscape options for HIL simulations. Specifically, automatic code generation and fixed-step solvers

3 SIMSCAPE TOOL

3.1 What is Simscape?

Simscape is one of the add-ons tools in the Simulink environment of MATLAB that can be accessible in the Simulink library browser. Simscape is used to simulate and model real engineering systems in different fields. Each block used in Simscape represents a real physical component. What distinguishes the Simscape from the standard Simulink modeling approach is, that it is based on a physical network approach which means that complex mathematical models are not required in Simscape. Instead, each parameter or component in the model is symbolized and illustrated by a specific physical block. Besides, unlike standard Simulink modeling, Simscape does not use normal connection lines that are based on input and output signals. Instead, their blocks have physical conserving ports, that can only be connected to other specific blocks. These lines are determined by specific colors.

3.2 Types of Variables

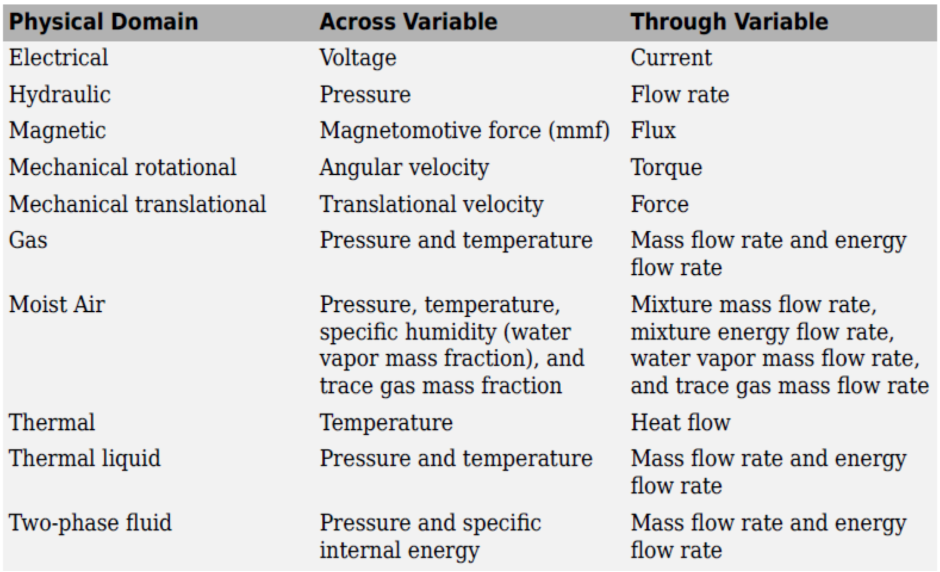
Variables in Simscape are divided into two categories[1]:

3.2.1 In terms of connectivity when measured

A. **Through variables** are variables that are connected in series to an element when they are measured by a sensor, like current speaking electrically.

B. **Across variables** are variables that are connected in parallel to an element when they are measured by a sensor, like voltage speaking electrically.

Fig. 3-1 illustrates the through and across variables for different domains:



Physical Domain	Across Variable	Through Variable
Electrical	Voltage	Current
Hydraulic	Pressure	Flow rate
Magnetic	Magnetomotive force (mmf)	Flux
Mechanical rotational	Angular velocity	Torque
Mechanical translational	Translational velocity	Force
Gas	Pressure and temperature	Mass flow rate and energy flow rate
Moist Air	Pressure, temperature, specific humidity (water vapor mass fraction), and trace gas mass fraction	Mixture mass flow rate, mixture energy flow rate, water vapor mass flow rate, and trace gas mass flow rate
Thermal	Temperature	Heat flow
Thermal liquid	Pressure and temperature	Mass flow rate and energy flow rate
Two-phase fluid	Pressure and specific internal energy	Mass flow rate and energy flow rate

Fig. 3-1: Across and Through variables

The product of the Across and the Through variable of each domain is power in Watt except for magnetic and thermodynamic domains.

3.2.2 In terms of sign (direction)

A. Positive variables:
   - The Through variable is positive when the element direction and its flowing (TV) are in the same direction as in Fig. 3-2
   - The Across variable is determined as $AV = AV_A - AV_B$ so if $AV_A > AV_B$ then the across variable is positive.

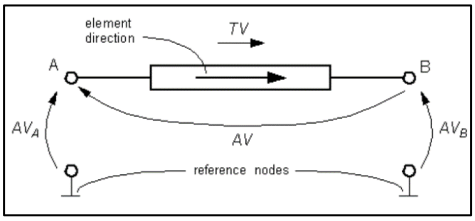


Fig. 3-2: Through positive variable

B. Negative variables:
   - The Through variable is considered to be negative when the element direction and its flowing (TV) are in opposite directions as in Fig. 3-3
   - The Across variable is determined as $AV= AV_A - AV_B$ so if $AV_B > AV_A$ then the across variable is negative.

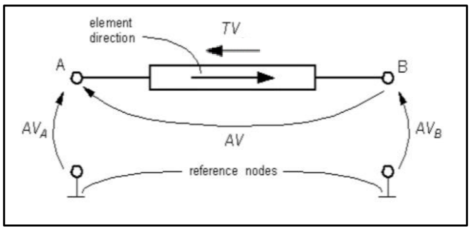


Fig. 3-3: Through negative variable

The importance of this kind of classification comes handy when we want to know and determine whether the element consumes or provides energy. The element consumes energy when the product of the Across variable and the Through variable is positive, while the element produces energy if the product of the Across variable and the Through variable is negative.

3.3 Elements of Simscape

Simscape contains many elements of real-world physical components. Each component differs from the other, but they all share these common features:

- Each element can only be connected to either a conserving port of the same type or to the reference.
- Some elements have one port, some have two ports, and some have more than two.
- Every element has a value and a unit. The value can be changed and the unit has a changeable metric size. For example, converting from milli to micro.
- Each element is either a passive element or an active element. An element is considered to be passive if it sends energy like force and velocity. While an element is considered to be active if it stores energy like dampers, resistors, and springs.

3.4 Differences between Simscape modeling and standard Simulink modeling

Simscape modeling:

1. Based on physical modeling approach.
2. Graphs can be generated automatically from logging simulation data.
3. Each block mimics a real physical component.
4. Complicated mathematical operations and functions are not needed.
5. The connection ports are nondirectional.

Standard Simuilink modeling:

1. Based on the standard mathematical approach.
2. Scopes must be used if graphs are needed to be drawn.
3. Each block is considered as a mathematical opertation that helps to representent the model.
4. Mathematical operations and differencial equations are demanded to most models.
5. Connection ports are directional.

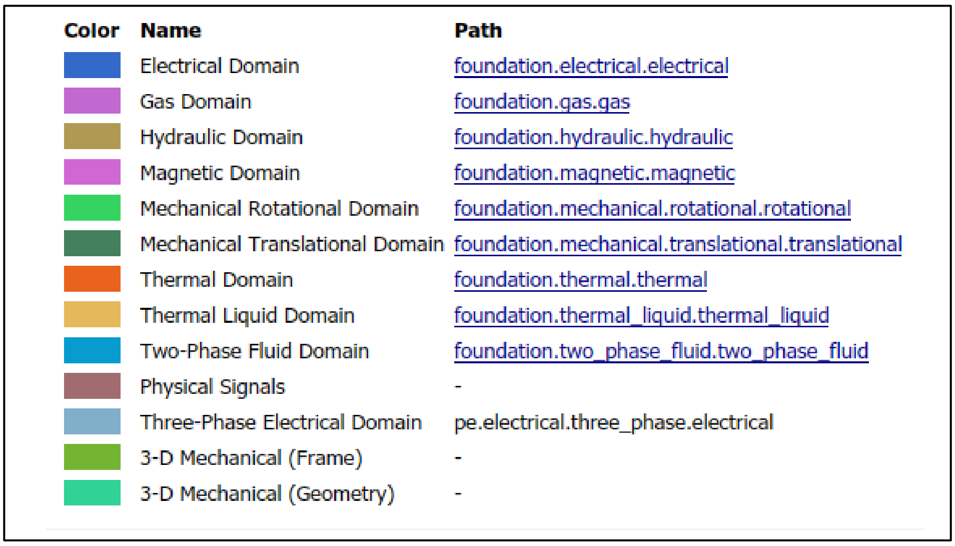
3.5 Advantages and features of Simscape

1. A very effective and fast tool for simulating models.
2. Multiple different systems can be integrated and simulated at the same time.
3. All systems of Simscape can generate code and can later be used for HIL.
4. Simscape can be connected with standard Simulink blocks by specific block converters.
5. All graphs regarding the model are automatically generated and even can be reused in MATLAB form workspace.
6. New blocks can be added to the Simscape Library. The functions and parameters of these elements can be created by MATLAB.
7. Each domain in Simscape is distinguished by a distinct color.
8. Simscape enables 3D simulation. Also, prepared CAD models can be imported to Simscape.

3.6 Ports and lines in Simscape

There are some important notes about ports and lines of Simscape illustrated as following[1]:

- Ports are only connected to other ports of the same type.
- Blocks are not connected by the black standard lines of Simulink. Instead, each domain has a specific line that is distinguished with a specific color as shown in Fig. 3-4



Color	Name	Path
🟦	Electrical Domain	foundation.electrical.electrical
🟪	Gas Domain	foundation.gas.gas
🟫	Hydraulic Domain	foundation.hydraulic.hydraulic
🟪	Magnetic Domain	foundation.magnetic.magnetic
🟩	Mechanical Rotational Domain	foundation.mechanical.rotational.rotational
🟩	Mechanical Translational Domain	foundation.mechanical.translational.translational
🟧	Thermal Domain	foundation.thermal.thermal
🟨	Thermal Liquid Domain	foundation.thermal_liquid.thermal_liquid
🟦	Two-Phase Fluid Domain	foundation.two_phase_fluid.two_phase_fluid
🟫	Physical Signals	-
🟦	Three-Phase Electrical Domain	pe.electrical.three_phase.electrical
🟩	3-D Mechanical (Frame)	-
🟩	3-D Mechanical (Geometry)	-

Fig. 3-4: Color Lines of Simscape domains

- Ports of Simscape blocks cannot be directly connected with standard Simulink blocks. A converter must be used either form Simulink to Simscape or vice versa, depending on the purpose.
- Ports of Simscape blocks of one domain cannot be connected directly to another domain of Simscape. However, some specific blocks normally connect two different domains together like the Rotational Electromechanical Converter in Fig 3-5. It connects between the electrical domain represented by the blue color and the mechanical rotational domain represented by green color.

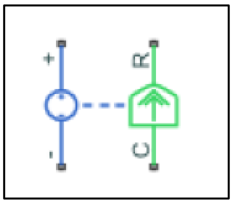


Fig. 3-5: Rotational Electromechanical converter

4 OPTIONS OF SIMSCAPE

4.1 Simscape Block Libraries

4.1.1 Main libraries

Two main libraries belong to Simscape which are[1]:

- **Foundation Library** includes the different principal domains in Simscape that represent different systems with their various comprehensive and extensive components. In total, the foundation library includes nine domains (Electrical, Gas, Hydraulic, Magnetic, Mechanical, Physical Signals, Thermal, Thermal Liquid, and Two-Phase Fluid) illustrated in Fig. 4-1
  Each domain in the Foundation Library includes tens of physical components, sources, and sensors that are used for modeling the system.

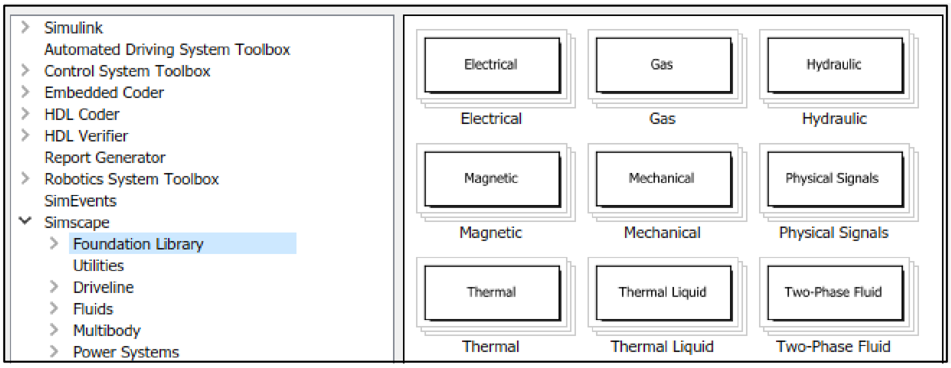


Fig. 4-1: Foundation Library

- **Utilities Library** includes seven blocks which are used to facilitate the simulation in Simscape. Also, some of them are used to connect blocks from standard Simulink to Simscape and vice versa. Blocks are shown in Fig. 4-2

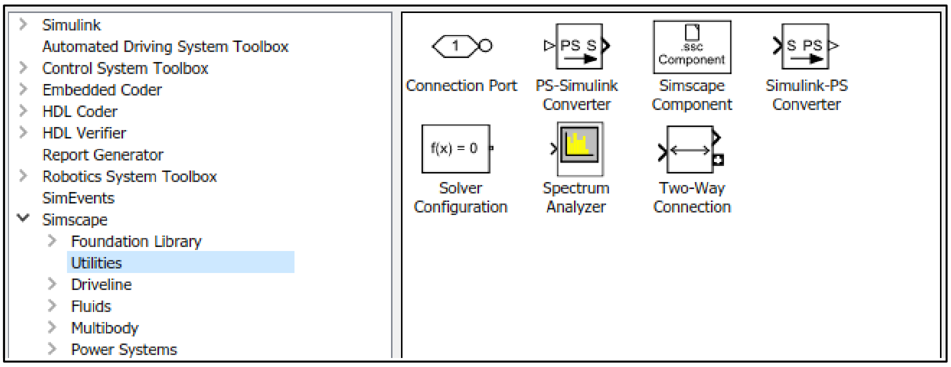


Fig 4-2: Utilites Library

4.1.2 Add-ons libraries

The add-ons Simscape products in Fig. 4-3 include four additional advanced libraries as following:

1. **Simscape Electrical™** (previously known as SimPowerSystems™ and SimElectronics®) provides many components and elements that are used for modeling electronic, mechatronics, and electrical power systems. It consists of different useful components of models representing motors, semiconductors, sources, and other control elements. This library is mainly used when modeling the applications of electromechanical actuation, smart grids, and renewable energy systems.[2]
2. **Simscape Driveline™** (previously known as SimDriveline™) includes many models that are used in rotational and translational mechanical systems. It has different elements of breaks, clutches, couplings, engines, gears, tires, vehicles, and more. It is used in different applications such as industrial machinery and automotive powertrains.[3]
3. **Simscape Fluids™** (formerly known as SimHydraulics®) has libraries' components for simulating fluid systems. It contains different models of actuators, pumps, valves, tanks, accumulators. We can use these components to develop fluid power systems such as front-loader, power steering, and different hydraulic cylinders. The main usage of Simscape Fluids is to enable developing and simulating fuel supply systems, gearbox lubrication, and engine cooling.[4]
4. **Simscape Multibody™** (formerly known as SimMechanics™) is equipped with many components that help the user to generate and run models in a 3D environment. SimMechanics™ library contains many mechanical elements like body elements, constrains, gears, couplings, frames, transforms, and joints. These components are connected together according to the required model and therefore it makes the assembly and the simulation possible in the form of 3D. After that, an automatically 3D animation is generated to analyze and visualize the system dynamics. A useful characteristic of Simscape Multibody™ is that complete and prepared CAD assemblies can be normally imported into the model including all masses, inertias, joints, constraints, and 3D geometry.[5]

Electrical
Driveline
Multibody
Fluids
Simscape

Fig. 4-3: Add-ons products

4.1.3 Accessing the Block Libraries:

The Simscape Library can be accessed in different ways[1]:

1. When typing ssc_new in the MATLAB Command Window, a new untitled Simulink model will be opened including these blocks: Solver Configuration, Simulink-PS Converter, PS-Simulink Converter, Scope, and Simscape Library.
2. When sLibraryBrowser is typed in the MATLAB Command Window, the Simulink Library Browser is opened enabling the user to access the Simscape Library.
3. If Simscape is typed in the MATLAB Command Window, the Library of the Simscape will be opened including all the add-ons products of Simscape.

4.2 Connecting Simscape with Simulink blocks:

When a connection between the Simscape and Simulink blocks is required, a converter must be used. There are two types of converters[1]:

1. **Simulink-PS Converter** is used when a conversion from Simulink blocks to Simscape (physical signal) is needed. See Fig. 4-4-1
2. **PS-Simulink Converter** is used when a conversion form Simscape (physical signal) to Simulink block is needed. See Fig. 4-4-2

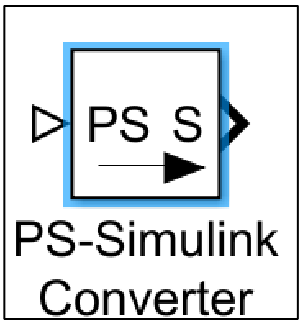


Fig. 4-4-1: PS-S converter

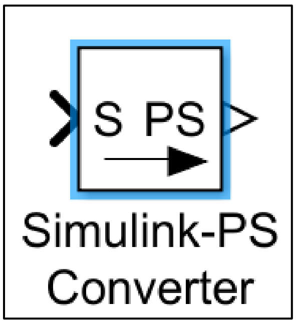


Fig 4-4-2: S-PS converter

4.3 Solver Configuration Block

The following notations summarize the main idea and purposes of using Solver Configuration Block in Simscape[1]:

- Solver Configuration block is a block that must be used in each Simscape model. No model in Simscape can run without it.
- Solver configuration specifies and determines the solver parameters that a model requires.
- It must be used at least once in each Simscape model, but can also be used more than once in different domains.
- It has only one port which can be connected to any Simscape connection line of the model. Look at fig 4-5.
- It is found in the Simscape Library – Utilities

- The default value of consistency tolerance is 1e-9, which is applicable and suitable to most cases in Simscape.
- Three types of local solvers can be used. Backward Euler, Trapezoidal Rule, or Partitioning. The default solver is Backward Euler. More details in Fig. 4-6.
- When doing HIL and generating code, it is recommended to check Use local solver. At the same time, the value of nonlinear iterations is recommended to be 2 or 3 to speed up the simulation and avoid overrun in a Real-time system. In other words, local solvers improve Simulation performance.

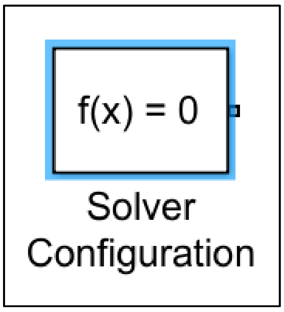
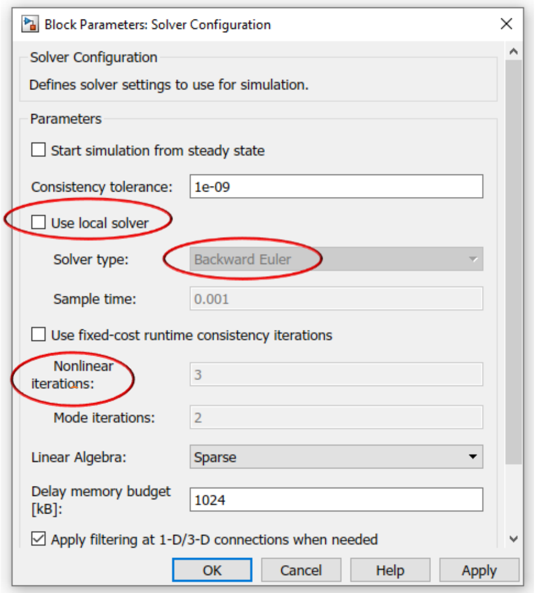


Fig. 4.5: Solver configuration block



Solver Configuration

Defines solver settings to use for simulation.

Parameters

□ Start simulation from steady state

Consistency tolerance: 1e-09

□ Use local solver

Solver type: Backward Euler

Sample time: 0.001

□ Use fixed-cost runtime consistency iterations

Nonlinear iterations: 3

Mode iterations: 2

Linear Algebra: Sparse

Delay memory budget [kB]: 1024

☑ Apply filtering at 1-D/3-D connections when needed

Fig. 4-6: Solver configuration block parameters

4.4 Recommended solvers of Simulink for Simscape

When simulating models in Simulink we have two options in terms of solvers[1]:

- **Variable-step solver** has 8 different types of solvers. The recommended type of variable-step solver for Simscape is ode15s or ode23t.
- **Fixed-step solver** includes also 8 different types of solvers. The recommended type of fixed-step solver for Simscape is ode14x.

4.5 References of Simscape domains

In Simscape conserving ports of the element must be connected to a reference. Each domain in Simscape has a reference and a representation as following:

- **The Electrical Reference block** represents the ground. See fig 4.7.1 left side.
- **The Thermal Reference block** represents a thermal point which is the absolute zero temperature. See Fig 4-7-1 right side.
- **The Mechanical Translational Reference block** represents a point of reference in which all translational elements ports must be connected to (a point fixed in space.) See Fig. 4-7-2 left side.
- **The Mechanical Rotational Reference block** represents a point of reference in which all rotational elements ports must be connected to (a point fixed in space.) See Fig. 4-7-2 right side.
- **The Magnetic Reference block** represents a point for all magnetic conserving ports. See Fig. 4-7-3 left side.
- **The Hydraulic Reference block** represents a connection to the atmosphere. See fig 4-7-3 right side.
- **The Gas Absolute Reference block** represents the absolute reference in a gas network which means that pressure and temperature have zero values. See fig 4-7-4 left side.
- **The Thermal Liquid Absolute Reference block** represents the absolute reference in a thermal liquid network which means that pressure and temperature have zero vlues. See fig 4-7-4 right side.
- **The Two-Phase Fluid Absolute Reference block** represents the absolute reference in a two-phase fluid network which means that pressure and temperature have zero values. See fig 4-7-5

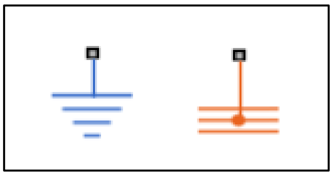


Fig. 4-7-1: ER and TR blocks

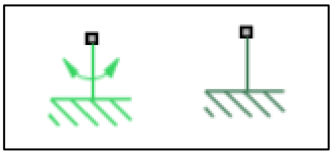


Fig. 4-7-2: MTR and MRR blocks

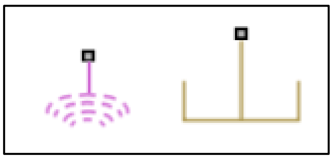


Fig. 4-7-3: MR and HR bocks

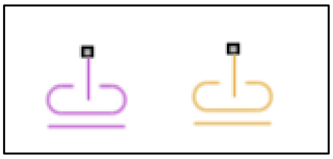


Fig. 4-7-4: GAR and TLAR blocks

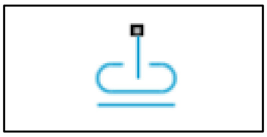


Fig 4-7-5: 2PFR block

4.6 General rules in Simscape Modeling[1]

1. A Solver Configuration Block must be used for each Simscape model. Optionally in the same model, more than one Solver Configuration can be added for each domain.
2. When moving from standard Simulink to Simscape, a converter must be used.

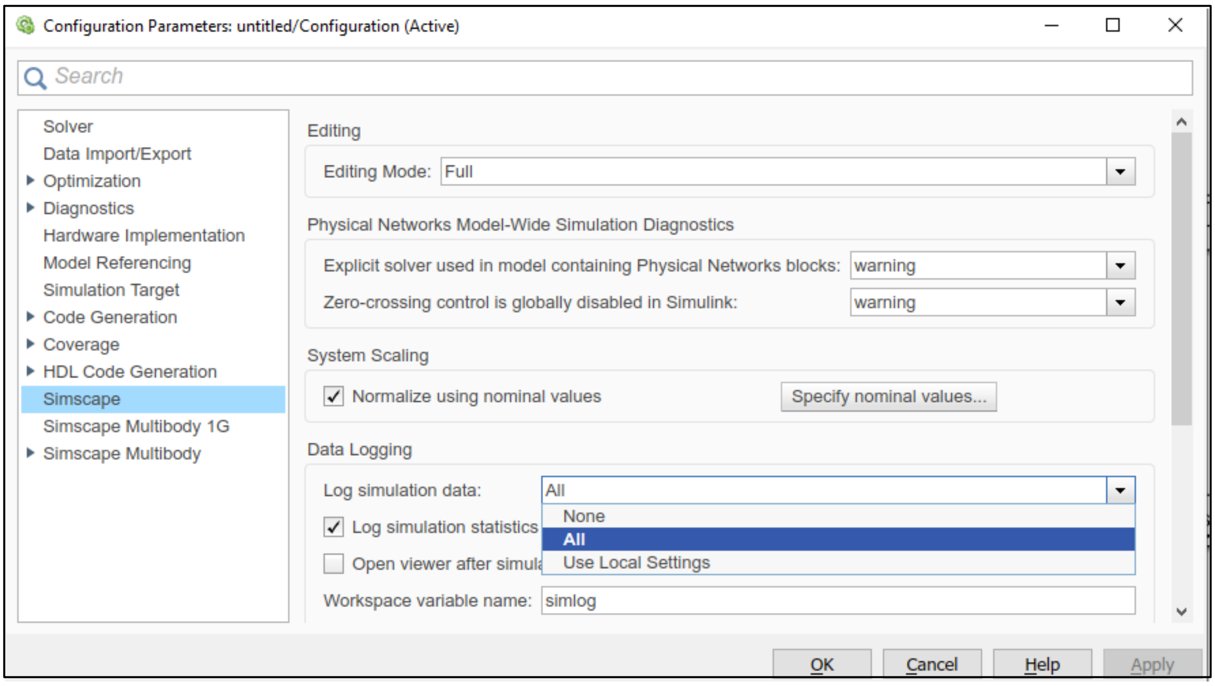
3. It is better to start the model from its easiest possible shape and gradually add the elements to the model.
4. Ports of Simscape can only be connected to other parts of the same type.
5. For each domain in Simscape, there is a reference that must be connected to other ports of the Simscape elements model. There are different types of references for different domains as illustrated in chapter 4.5. More than one reference can be added to the same domain. They would work as one reference.
6. A user of Simscape must be aware of the default setting modes of Simscape with the possibility of changing if necessary.

4.7 Additional features in Simscape

4.7.1 Logging Simulation Data in Simscape

A very useful feature in Simscape that all data regarding the elements in the model can be automatically generated in multiple graphs. This feature enables the user to easily access the data needed and therefore easier and faster analyzing for data.

To activate auto logging simulation data in Simscape some initial steps must previously be done. Firstly, we open the Model Configuration Parameters in Simulink. Then we open the Simscape in the left column. Next, we change the Log simulation data to All and we check for the log simulation statistics as shown in Fig. 4-8. Also, the data can be saved with a name that is later stored in MATLAB workspace variables. In our case, the name is saved within the name of simlog as shown below in Fig 4-8.



Configuration Parameters: untitled/Configuration (Active)

Editing

Editing Mode: Full

Physical Networks Model-Wide Simulation Diagnostics

Explicit solver used in model containing Physical Networks blocks: warning

Zero-crossing control is globally disabled in Simulink: warning

System Scaling

☑ Normalize using nominal values

Data Logging

Log simulation data: All

☑ Log simulation statistics

□ Open viewer after simula

Workspace variable name: simlog

Fig. 4-8: Logging simulation data

Then the data can be accessed by left-clicking any element in the Simscape model → View Simulation data → simlog (which is the saved name of the data) as shown in Fig. 4-9. After that, some graphs will be shown representing each element from different aspects.

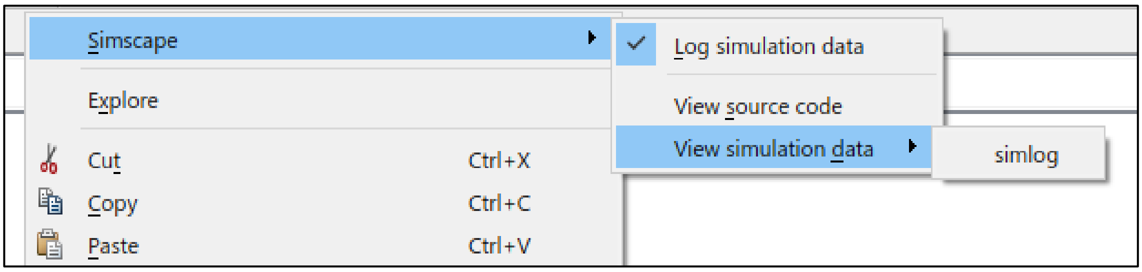


Fig. 4-9: Opening simulation data

Another useful feature of Logging Simulation Data in Simscape is that the data are stored in the MATLAB workspace variable. It means that all data can be manipulated and redrawn with other data to compare, analyze, and make conclusions of the studied case.

4.7.2 Preparing the model for other users

Since Simscape and its products belong to the add-on's products, not everyone can afford it and therefore it is not accessible to each person. Thus, there is an effective feature in Simscape that enables other users, who have access to the Simscape platform but not to its add-ons products, to use prepared models of Simscape. But a proactive step must be submitted before by the designer. The model must be saved in the restricted mode in advance.

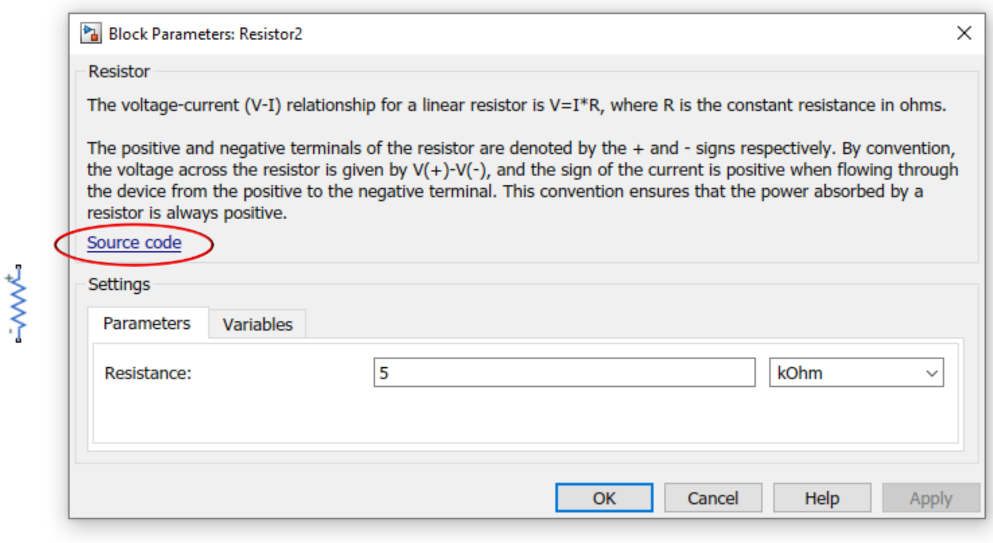
Unlicenced users can do the following with the models in restricted mode:

- Access and simulate the model.
- Parameter values can be changed but not deleted.
- Code generation can be done.
- Access to 3D models.

4.7.3 Source code of Simscape

Each element in Simscape is written in MATLAB functions. The source of the code can be accessed by double-clicking the element and clicking the source code as shown in Fig. 4-10. The figure represents a resistor with its parameters.

Source codes are read-only which means they cannot be modified. However, Simscape enables the user to create new elements by writing new source codes depending on the user's needs.



Block Parameters: Resistor2

Resistor

The voltage-current (V-I) relationship for a linear resistor is V=I*R, where R is the constant resistance in ohms.

The positive and negative terminals of the resistor are denoted by the + and - signs respectively. By convention, the voltage across the resistor is given by V(+)-V(-), and the sign of the current is positive when flowing through the device from the positive to the negative terminal. This convention ensures that the power absorbed by a resistor is always positive.

Source code

Settings

Parameters | Variables

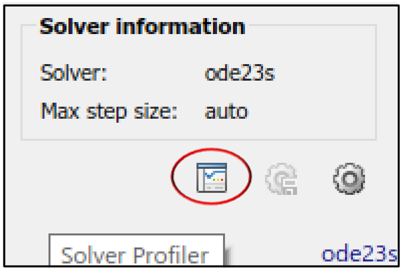
Resistance: 5 kOhm

Fig. 4-10: Resistor element with its block parameters

4.7.4 Finding causes of slow simulation

After choosing the appropriate step-solver or fixed-solver, a problem of slow simulation might occur. If the simulation is a little bit slow, there might be some Zero Crossing. To check whether there is a Zero Crossing or not, we click the Solver on the bottom bar of the Simulink in our case ode23s, and then we open the Solver profiler. See Fig 4-11

After that, an open window illustrated in Fig. 4-12 will be shown. We rerun the Simulink. From the graph, we can determine whether there is Zero Crossing or not and what exactly causes it. If we want to solve the problem, we click the Highlight Block and it directly shows the element that should be adjusted.



Solver information

Solver: ode23s

Max step size: auto

Fig. 4-11: Solver Profiler

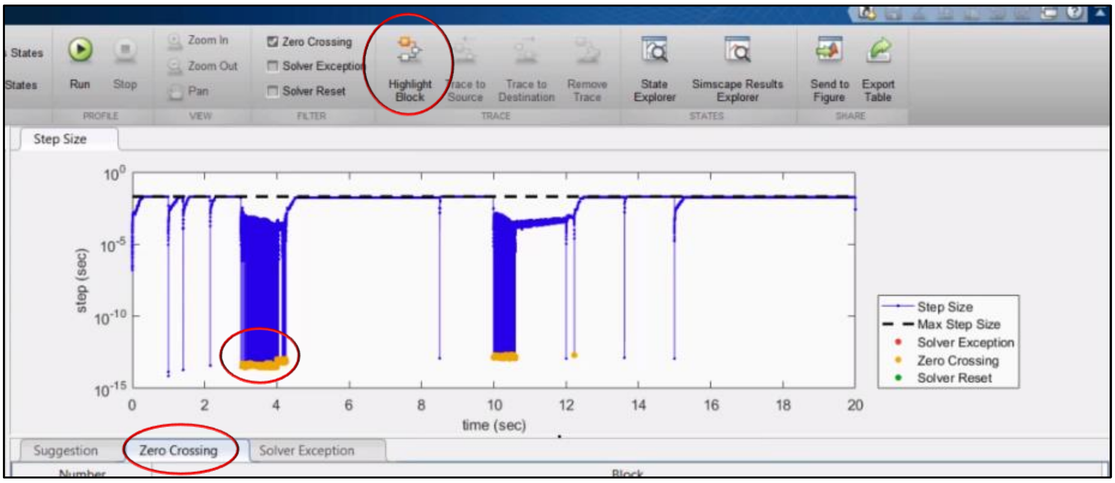


Fig. 4-12: Solving Zero Crossing

5 MECHATRONICS MODELS

5.1 Brush DC motor model

Brush DC Motor is one of the devices that convert electrical energy into mechanical rotary energy. It consists of two parts. The first part is the stator in which the permanent magnet is implemented. The second part is the armature which is connected to a number of coil turns. These turns rotate due to two factors. Flowing of direct electrical current and presence of the stator magnetic field.

Brush DC motors are used in many applications such as electric traction, cranes, lifts, and even in small sizes as in toys. They are preferred to use because they are easy to be controlled and modeled. They are highly recommended when a good speed control characteristic is required.

The DC Brush motor model shown in Fig. 5-1 consists of two parts[6]:

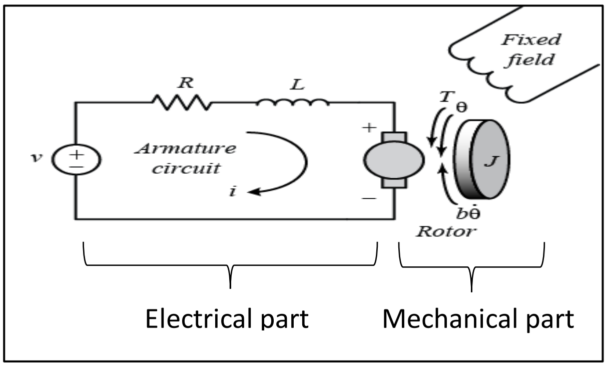


Fig. 5-1: DC motor model

- The electrical domain which represents the electrical components in equations (1,2, and 3) derived from Kirchhoff's second law which states that summing of all voltages around a closed loop equals zero.

The electrical model includes the armature resistance R_a and the leakage inductance L_a and so the Kirchhoff's voltage law can be written as:

$$R_a i_a(\mathrm{t}) + L_a \frac{di_a(t)}{dt} + v_b(t) = v_s(t) \qquad (1)$$

where i_a is the armature current, v_b(t) is the back emf voltage and v_s (t) is the voltage source.

The back emf voltage $v_b(t)$ is proportional to the angular velocity as shown in the following equation:

$$v_b(t) = k_b w(t) \qquad (2)$$

where k_b is the back EMF constant. This constant is proportional to the magnetic flux density B, the length of coil L, the radius of coil r, and the number of coil turns N.

Then we substitute equation (2) into equation (3) and $k_b = k$:

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + kw(t) = v_s(t) \qquad (3)$$

- The mechanical domain in which its components can be represented in equations (4,5, and 6) derived from Euler's second law that states summing of all moments equals the rotational inertia multiplied by the angular acceleration.

$$J_M \frac{dw(t)}{dt} = T_M(t) - B_M w(t) - T_L(t) \qquad (4)$$

where J_M is the rotor moment of inertia, T_M is the motor torque, B_M is the friction coefficient, T_L is the external torque.

The torque of the motor is proportional to the electrical current as shown in the equation below:

$$T_M(t) = K_T i_a(t) \qquad (5)$$

where K_T is the torque constant and it equals $k_b$the back EMF constant so we can rename the two constants (K_T, k_b) as the constant of the motor $K_T = k_b = k$. Then we substitute equation (5) into (4) as follows:

$$J_M \frac{dw(t)}{dt}(t) = ki_a(t) - B_M w(t) - T_L(t) \qquad (6)$$

Simulink Model:

From equations (3 and 6) we can use the following differential equations for modeling DC motor in Simulink, in our case no external torque is implemented:

$$\frac{di}{dt} = \frac{v_s}{L_a} - \frac{R_a i(t)}{L_a} - kw(t) \qquad (7)$$

$$\frac{dw}{dt} = \frac{ki(t)}{J_M} - \frac{B_M w(t)}{J_M} \qquad (8)$$

The model is included with the attachments within the name 1st Model.

Simscape Model:

For creating the DC motor model in Simscape, only these parameters are required to be defined (R_a, L_a, J_M, B_M, k). Input value is v_s and output values will be current and angular velocity. The model is included with the attachments within the name 1st Model. The results of the two approaches of creating DC motor models are compared in Fig. 5-2 and 5-3. This model will be further studied in more detail within a real DC motor in chapter 6.

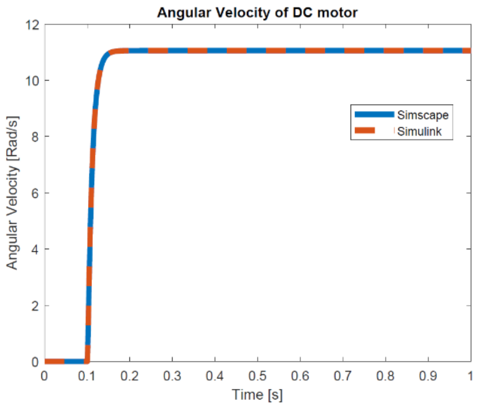


Fig. 5-2: Simulink and Simscape results of angular velocity

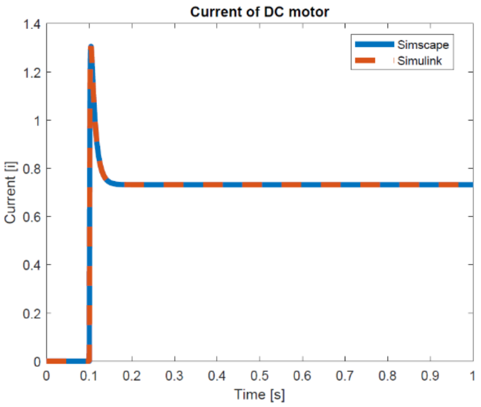
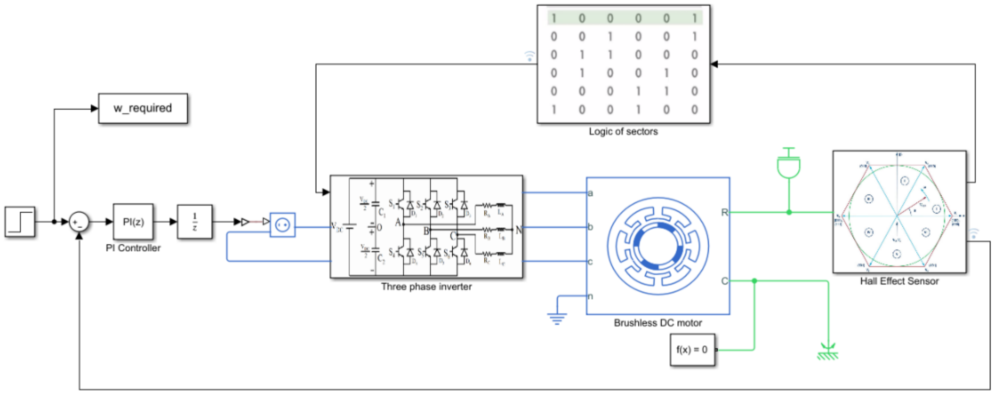


Fig. 5-3: Simulink and Simscape results of current

5.2 Brushless DC Motor Model

BLDC motors are actuators that generate motions from electricity. They consist of a rotor which is a permanent magnet and a stator in which the current flows to create a magnetic field. The main difference between them and Brush DC Motors is that BLDC motors do not include contact commutators. Therefore, BLDC motors are more efficient, more reliable, and less noisy.

BLDC Simscape Model:



1	0	0	0	0	1
0	0	1	0	0	1
0	1	1	0	0	0
0	1	0	0	1	0
0	0	0	1	1	0
1	0	0	1	0	0

Fig. 5-4: BLDC Model

BLDCs can vary from one to another by its pole numbers of stator and rotor. A one-pole rotor and three poles stator have been chosen to be modeled by Simscape. The three-phase inverter which includes six MOSFETs is required to control the flowing current to the stator poles. Also, hall effect sensors are used to determine where the rotor lies. It is divided into six sectors. Each sector ranges from 0^0 to 60^0. Finally, these sensors are used to choose the correct pattern for switching on and off the MOSFETs. A PI controller is used to control the desired speed value.[7] The model is attached within the attachments within the name 2nd Model.

Fig. 5-5 illustrates in which section the rotor is positioned with time. Fig. 5-6 shows the control of the angular velocity of the motor.

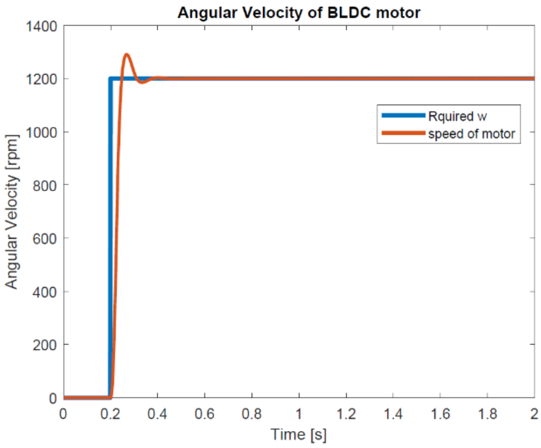


Fig. 5-5: Controlling angular velocity of BLDC

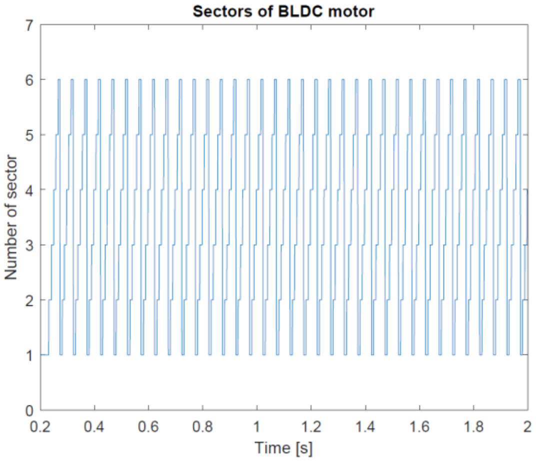


Fig. 5-6: Poistion of the rotor

5.3 Mass Spring Damper System Model

Mass spring damper system from its name consists of a mass, spring, damper, and ropes or rods. The system can also use multiple masses, springs, and dampers depending on the usage. It is used in many applications such as car suspensions, solenoids, doors, bungee cords, and so on.

Mass spring damper model:

For modeling the mass-spring-damper model, Newton's second law of motion is required. It points out that the summing of all forces acting on an object is equal to the mass multiplied by the acceleration. It can be abbreviated in this equation:

$$\sum F = ma \qquad (9)$$

From the characteristics of the spring and the damper, we know that the effect of spring and damper is against the acceleration of the mass, and so the equation becomes :

$$ma = -k_s y - bv \qquad (10)$$

Where k_s is the spring stiffness, y is the displacement, b is the damping constant, and v is the velocity.

Three different cases have been chosen to be simulated with three different approaches as following:

Case 1:

A force applied on a mass-spring-damper system that is on a horizontal surface as shown in Fig. 5-7. This system is studied by Simulink, Simscape, and Multibody. For the Simscape model, only values of (m, k_s, b, F) are needed for the model to be implemented in Simscpae components plus proper connection of lines. For the Simulink model, the differential equation is required to be derived. From equation (10) and based on Fig. 5-7 the differential equation of the system can be obtained as follows:

$$\frac{d^2y}{dt^2} = \frac{F}{m} - \frac{k_s}{m}y - \frac{b}{m}\frac{dy}{dt} - \frac{\mu F_N}{m}\, sgn(\frac{dy}{dt}) \qquad (11)$$

Where μ is Coulomb friction coefficient, $F_N$is the normal force.

For the multibody model, prismatic joint, brick solid, and other components are required. The results of the three approaches match with each other as shown in Fig. 5-8

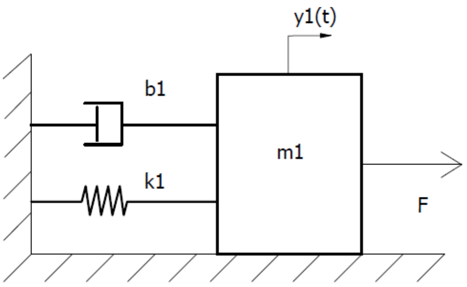


Fig. 5-7: Mass-spring-damper system 1

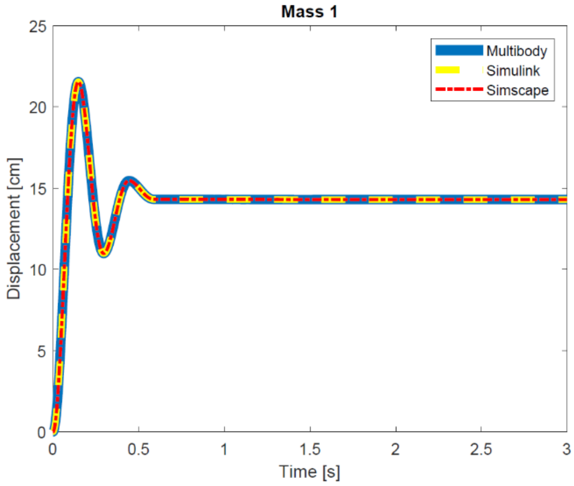


Fig. 5-8: Displacement of m1

Case 2:

A force applied on a mass-spring-damper system that is on incline plane with a specific angle, in our case 60^0 as shown in Fig. 5-9. For Simscape and Multibody models, it's the same as case 1 with few changes. For the Simulink model the differential equation will be:

$$\frac{d^2y}{dt^2} = \frac{F}{m} + -\frac{k_s}{m}y - \frac{b}{m}\frac{dy}{dt} + \frac{F_g}{m}\sin(\alpha) - \frac{\mu F_N}{m}\cos(\alpha)\, sgn\left(\frac{dy}{dt}\right) \qquad (12)$$

where $F_g$is the gravitational force, α is the angle shown in Fig. 5-9. The results of the three approaches of modeling are shown in Fig. 5-10

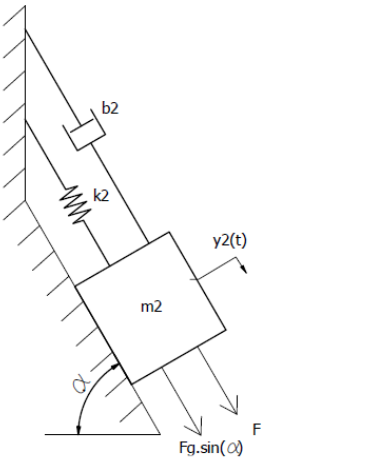


Fig. 5-9: Mass-spring-damper svstem 2

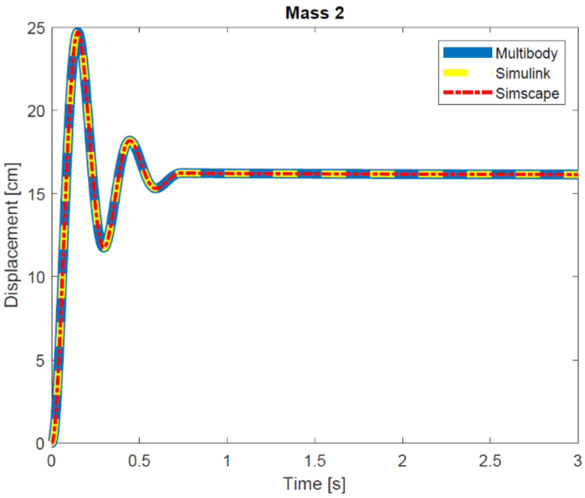


Fig. 5-10: Displacement of m2

Case 3:

In this case, only a gravitational force is applied to the body. In other words, it only moves vertically as shown in Fig. 5-11. For Simscape and Multibody models, the same components are used with minor differences. For Simulink approach the following differential equation is used:

$$\frac{d^2y}{dt^2} = \frac{F_g}{m} + -\frac{k_s}{m}y - \frac{b}{m}\frac{dy}{dt} \qquad (13)$$

Graphs of the displacements of three models are illustrated in Fig. 5-12.

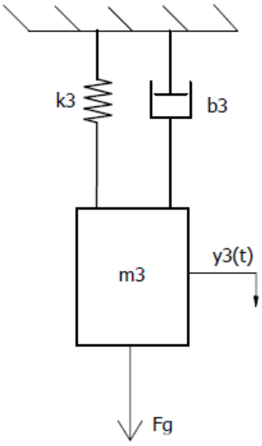


Fig. 5-11: Mass-spring-damper svstem 2

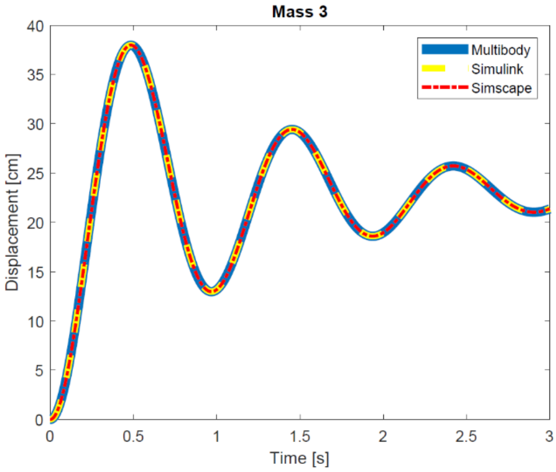


Fig. 5-12: Displacement of m3

The three different approaches of modeling for the three cases are attached with the attachments. The name of the file is 3rd Model.

5.4 Additional Models

Double Pendulum Model:

Double pendulum is a physical simple system that consists of a single one-end fixed pendulum connected with another pendulum as shown in Fig. 5-13. The movement of a single pendulum alone is considered to be regular and simple, but the movement of two pendulums connected together is chaotic and complex. However, it can be described by differential equations.

Double pendulum differential equations can be gained by Lagrange Equations. The last form of deriving of the double pendulum differential equations can be written as:

$$(m_1 + m_2)\,l_1\ddot{\alpha}_1 + m_2 l_2 \ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + m_2 l_2 \dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2)\,g \sin\alpha_1 = 0. \qquad (14)$$

$$l_2\ddot{\alpha}_2 + l_1\ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - l_1\dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2) + g \sin\alpha_2 = 0. \qquad (15)$$

where $m_1 m_2$ are masses of double pendulums, $l_1 l_2$ are lengths of the massless rods 1 and 2, g is the gravitational acceleration, $\alpha_1 \alpha_1$ are the deflection angles of the pendulums, $\dot{\alpha_1}\dot{\alpha_2}$ are the angular velocities of the pendulums, $\ddot{\alpha_1}\ddot{\alpha_2}$ are the angular acceleration of the pendulums.[8] These equations are used in Simulink to model the mathematical model.

For the Multibody model, only a few blocks are needed to be used like Revolute Joints, Solids, and Rigid Transforms. After setting initial deflections of rod 1, rod 2 for both models, comparison of angle deflections of two models can be gained in Fig. 5-14

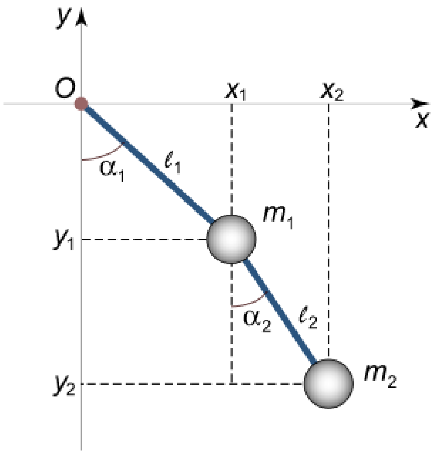


Fig. 5-13: Double pendulum

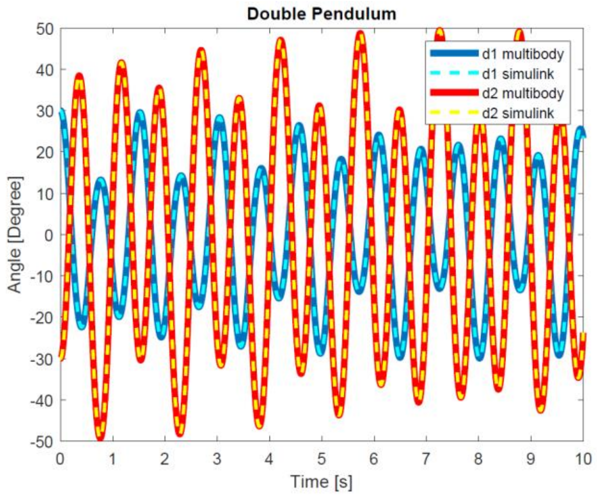


Fig. 5-14: Angle of double pendulum by Simulink and multibody

Both of the models are attached within the name 4th model.

Kinetic Mechanism Model:

This model is done with multibody. It consists of mass points and massless rods as shown in Fig 5.15. It is moved by an external torque.

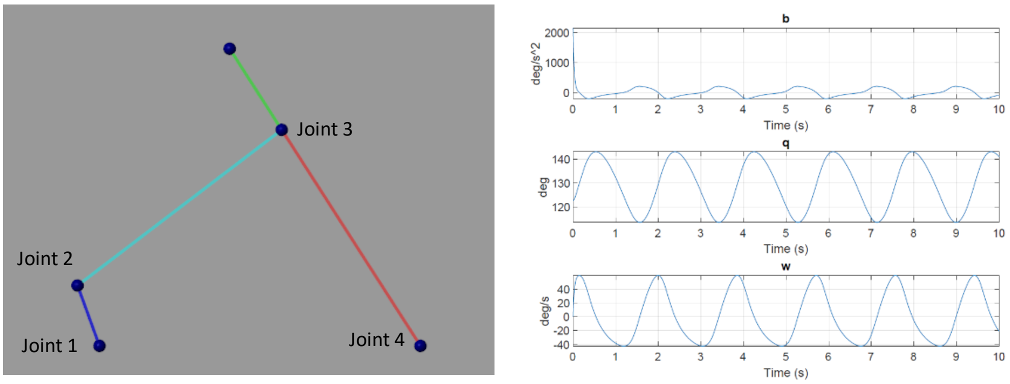


Fig. 5-15: Kinetic Mechanism and Results of joint 4

The model is attached within the name 5th model.

6 PARAMETERS’ EVALUATION AND MODEL OPTIMIZATION

In this chapter, parameters of a real DC motor will be calculated with the help of MF624 that is connected to the double drive motor in Mechlab. After that, these parameters will be evaluated and compared with the Simscape model accomplished in chapter 5.1. Finally, the model will be optimized for the PI control system.

6.1 Calculating parameters of DC motor

Real-Time Windows Target(RTWT)

RTWT is a Simulink tool that enables connection with real-world devices. It allows measuring analog or digital data in real-time as long as the simulation is running.
RTWT library includes many blocks as shown in Fig 6.1. These blocks facilitate the connection and the communication between the simulation and the device in our case is the Mechlab DoubleDrive.[9]

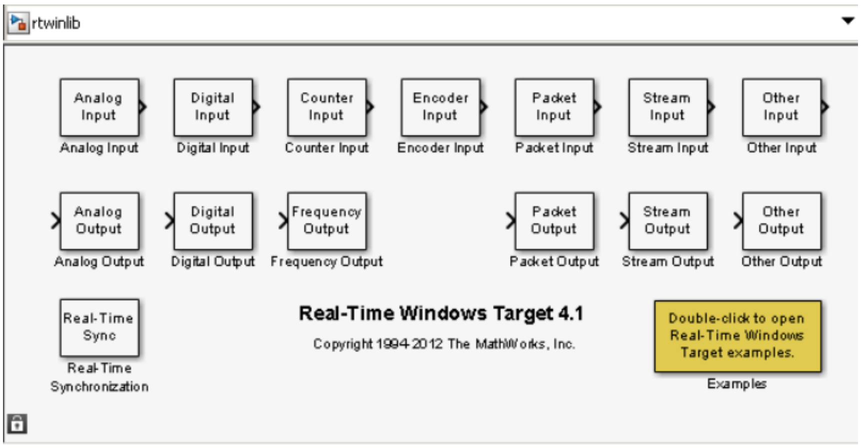


Fig. 6-1: Real-Time Windows Target Library

MechLab DoubleDrive

It is a device that enables communication in real-time with Simulink Real-Time Toolbox. The device includes Double DC motors as shown in Fig. 6-2. The device includes[9]:

- A connection to MF624 card.
- Two DC motors with different parameters.
- H-Brigdge, encoders, LEDs.
- Connection cables and belts.

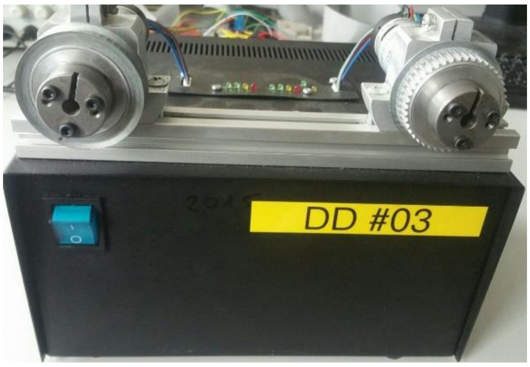


Fig. 6-2: Mechlab DoubleDrive

For calculating DC parameters, two things are required:

- Measured data and graphs of DoubleDrive in real-time. In other words, a model that enables us to set input value for voltage and sending it to DoubleDrive. At the same time, this model should allow us to get data and graphs of outputs which are the angular velocity and current. This model is accomplished within the name Calculating Parameters with the attachments.
- DC Motor Model equations which will enable us to calculate (R_a, L_a, J_M, B_M, k) one by one. The equations that are used are (3 and 6).

Calculating Armature Resistance

For calculating R_a we use equation (3) which is:

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + kw(t) = v_s(t) \qquad (3)$$

An input voltage $v_s(t)$ is being applied, and at the same time, we hold the DoubleDrive motor by hand to stop it from rotating. The graphs obtained are shown in Fig. 6-3 and Fig. 6-4. We can notice that the current is constant and so the derivation of it at this moment is 0. Thus, the equation becomes:

$$R_a i_a = v_s(t) \qquad (14)$$

$$R_a = \frac{v_s(t)}{i_s(t)} = \frac{19.2}{2.7} = 7.1 \quad \Omega$$

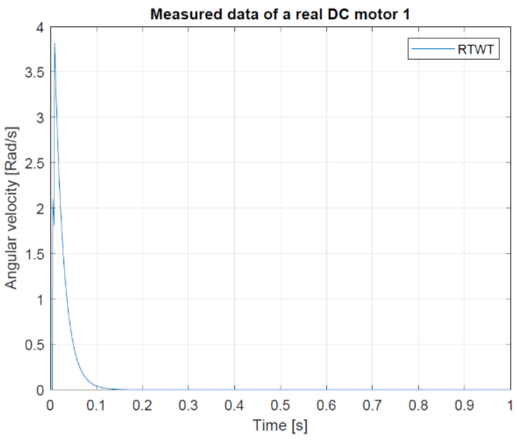


Fig. 6-3: Measured data of Angular velocity

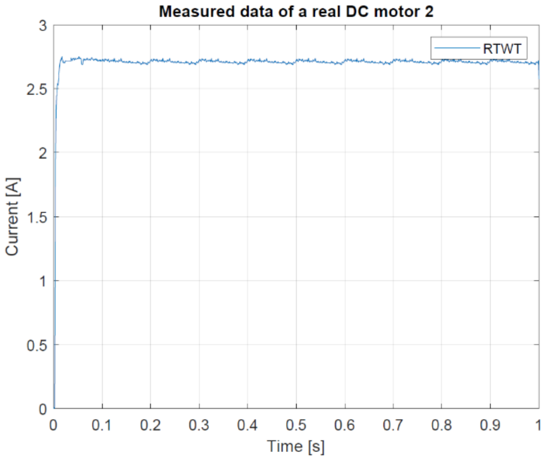


Fig. 6-4: Measured data of current

Calculating Motor Constant

In order to calculate k we can use equation (3). This time we will let the motor rotate so that we will get the value of $w(t)$ as shown in Fig. 6-5. Constant value of i_a while the motor is rotating can be gained from Fig. 6-6. We can observe that value of i_a is constant, so the derivation of it is 0. Value of R_a is already obtained.

Equation (3) after modifying, becomes in this form:

$$R_a i_a(t) + kw(t) = v_s(t) \qquad (15)$$

$$k = \frac{v_s(t) - R_a i_a(t)}{w(t)} = \frac{19.2 - (7.1 * 0.13)}{121} = 0.151 \quad \mathrm{V/Rad.}s^{-1}$$

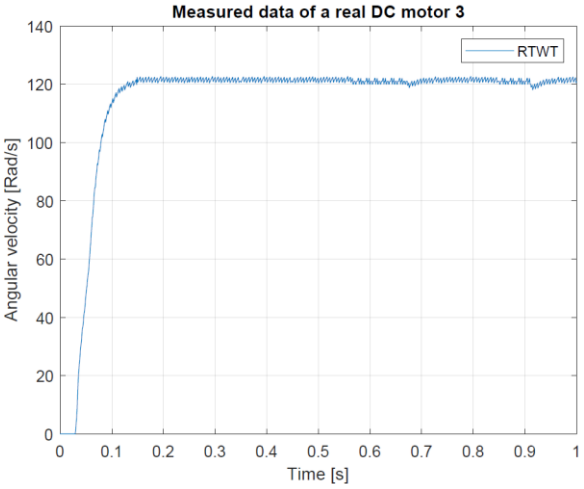


Fig. 6-5: Measured data of Angular velocity

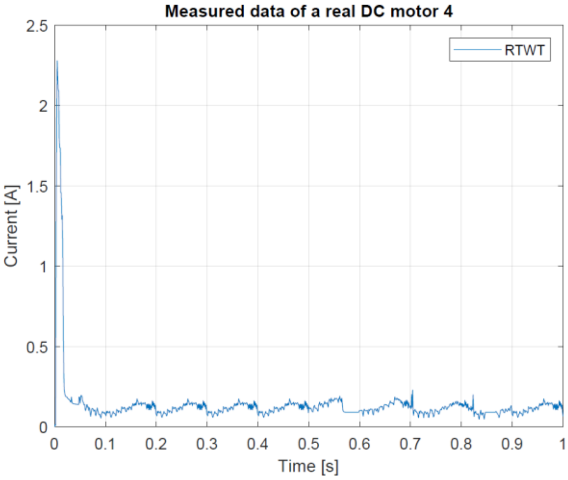


Fig. 6-6: Measured data of current

Calculating Leakage Inductance

For calculating L_a we will use the graph(6-3) that we used when we calculated R_a which means we will apply the input voltage. At the same time, we will hold the motor to prevent the rotating. So equation (3) becomes with this form:

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} = v_s(t) \qquad (16)$$

Since we have the graph of $i_a(t)$ and equation (16) is a differential equation then we can apply the Laplace Transform for equation (16) so it becomes:

$$(R_a + L_a S)I(s) = V(s) \qquad (17)$$

Then the transfer function becomes:

$$\frac{I(s)}{V(s)} = \frac{1}{\tau_e s + 1} \qquad (18)$$

Where $\tau_e$is the Time Electrical Constant. The time constant can be represented by this equation:

$$\tau_e = \frac{L_a}{R_a} \qquad (19$$

This constant can be gained from Fig. (6-7) and R_a is already calculated before. And so we can find the value of L_a as following:

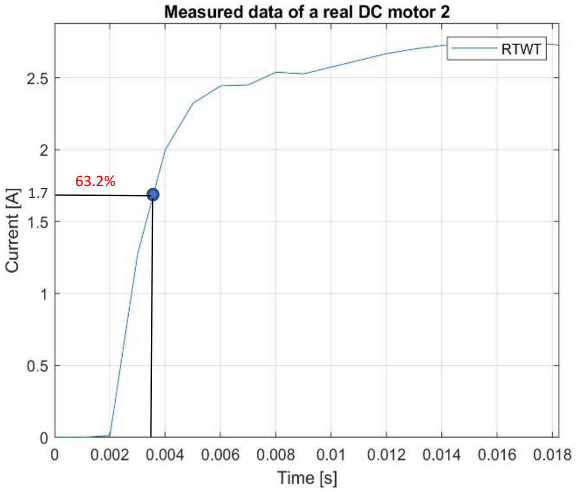


Fig. 6-7: Measured data of Tau Electrical

$$L_a = R_a \tau_e = 7.1 * 1.6 * 10^{-3} 0.03 \qquad = 3.408 * 10^{-4}\ Henry$$

Calculating Friction Coefficient

For calculating the Viscous friction coefficient B_M we use equation (6):

$$J_M \frac{dw(t)}{dt}(t) = ki_a(t) - B_M w(t) - T_L(t) \qquad (6)$$

we let the DoubleDrive motor rotate with constant angular velocity. At the same time, we do not apply any external torque, so the equation (6) turns out into this form:

$$ki_a(t) = B_M w(t) \qquad (20)$$

k is calculated before, $w(t)$ can be gotten from Fig. 6-5, $i_a(t)$ can be gotten from Fig. 6-6, and so we can calculate B_M

$$B_M = \frac{ki_a(t)}{w(t)} = \frac{0.151 * 0.13}{121} = 1.6 * 10^{-4}\ N/ms^{-1}$$

Or we can find Viscous Friction and Stribeck Effect [10] from Fig. 6-8. This graph represents the torque of the motor in the y-axis and angular velocity in the x-axis. This parameter can be entered into the Simscape model within the Rotational Friction block as the following data

Breakaway friction torque = 0.0118 N.m

Breakaway friction velocity = 0.1 rad/s

Coulomb friction torque = 0.00854 N.m

Viscous friction coefficient =

1.236*1e-4 N*m/(rad/s)

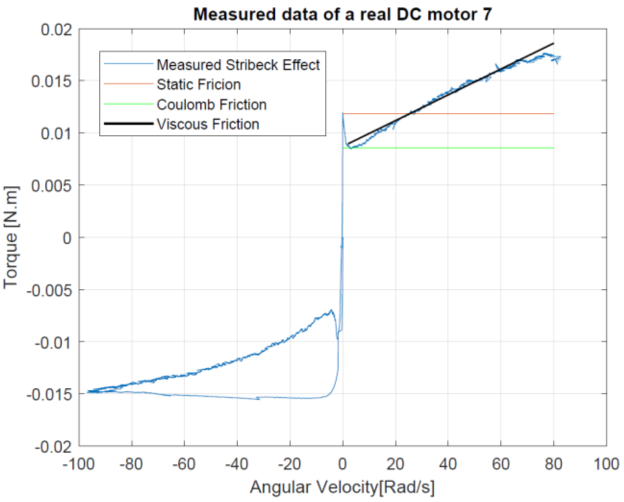


Fig. 6-8: Measured data of Friction

Calculating Moment of Inertia

For calculating J_M we need to use equation (6) without external torque applied to it, so the equations become:

$$J_M \frac{dw(t)}{dt}(t) = ki_a(t) - B_M w(t) \qquad (21)$$

Now we can use Laplace Transform since we have the graph of Fig. 6-9.

$$(J_M S + B_M)W(s) = k\, I(s) \qquad (22)$$

Then the transfer function becomes:

$$\frac{W(s)}{I(s)} = \frac{k}{(\tau_m S + 1)} \qquad (23)$$

Where $\tau_m$is the Time Mechanical Constant. The time constant is represented by this equation:

$$\tau_m = \frac{J_M}{B_M} \qquad (24)$$

This time constant can be gained from Fig. (6-9) and B_M is already calculated from equation (20). And so we can find the value of J_M as following:

$$J_M = B_M \, \tau_m = 1.6 * 10^{-4} * \ 0.03 = 4.8 * 10^{-6} \, Kg.m^2$$

All calculated parameters are illustrated in Tab. 6-1.

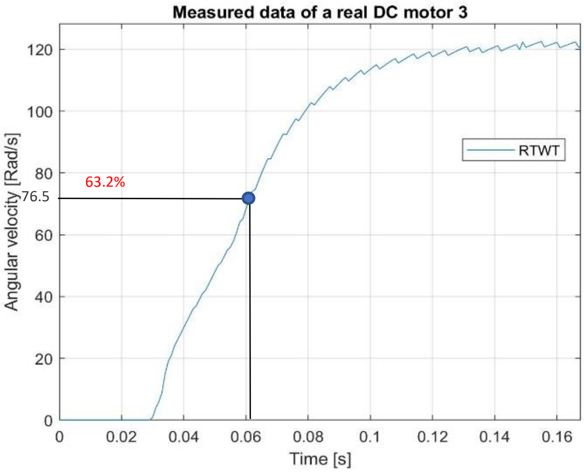


Fig. 6-9: Measured data of Tau Mechanical

Tab. 6-1 Calculated parameters

Parameter	Value	Unit
R	7.1	Ω
L	$3.408 * 10^{-4}$	Henry
K	0.151	$\text{V/Rad}.s^{-1}$
J	$4.8 * 10^{-6}$	$Kg.m^2$
Breakaway_friction_torque	0.0118	N.m
Breakaway_friction_velocity	0.1	Rad/s
Coulomb_friction_torque	0.00854	N.m
Viscous_friction_coefficient	$1.236 * 10^{-4}$	N.m/(rad/s)

6.2 Evaluation of parameters

Evaluation of parameters can be achieved with the help of the Sensitivity Analysis App in Simulink or what is known as Monte Carlo Analysis. It is a way to describe and know how uncertainty or variability in the model output can be attributed to uncertainty in its input. It enables us to know the most important parameters that affect the model and so reducing the number of parameters when doing the estimation later.

Achieving the evaluation of parameters can be reached within the following steps. First of all, opening the Sensitivity Analyzer window. Secondly, selecting the parameters needed to be evaluated. Also, selecting the requirements for the evaluation in our case measured data. Thirdly, Generating the number of samples with its method, setting the type of distribution for each parameter as well as setting lower and upper values. Fourthly, running the Evaluate Model button and waiting for the results. And then plotting the statistics plots to recognize what are the most effective parameters. The statistics plots are shown in Fig. 6-10. Finally, we can use these parameters only for the estimation instead of using all of them, and thus we reduce time, cost, and improve performance.

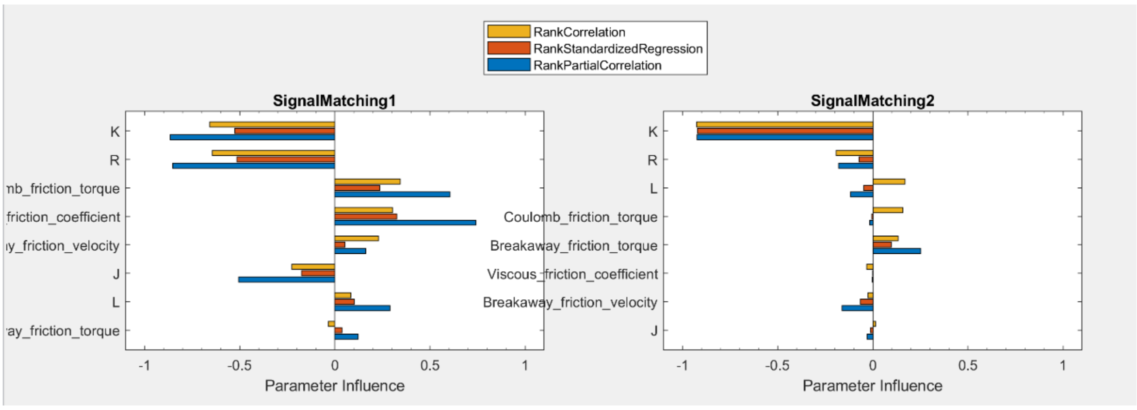


Fig. 6-10: Evaluation of parameters with respect to current (left side) and angular velocity(righ side)

From the total samples generated, the recommended values to start the estimation with are shown in Tab 6-2

Tab. 6-2 Recommended parameters to start the estimation

Parameter	Value	Unit
R	7.607471	Ω
L	0.000339	Henry
K	0.158160	V/Rad.s^{-1}
J	0.000005	$Kg.m^2$
Breakaway_friction_torque	0.010783	N.m
Breakaway_friction_velocity	0.100315	Rad/s
Coulomb_friction_torque	0.007828	N.m
Viscous_friction_coefficient	0.000123	N.m/(rad/s)

6.3 Estimation of parameters

After calculating the DC motor model parameters form the DoubleDrive motor graphs, we then apply these parameters into the DC Simscape model which is explained in chapter 5.1.

After deploying the parameters of the DC motor into the Simscpae model, the measured data and the created model are compared in Fig 6-11

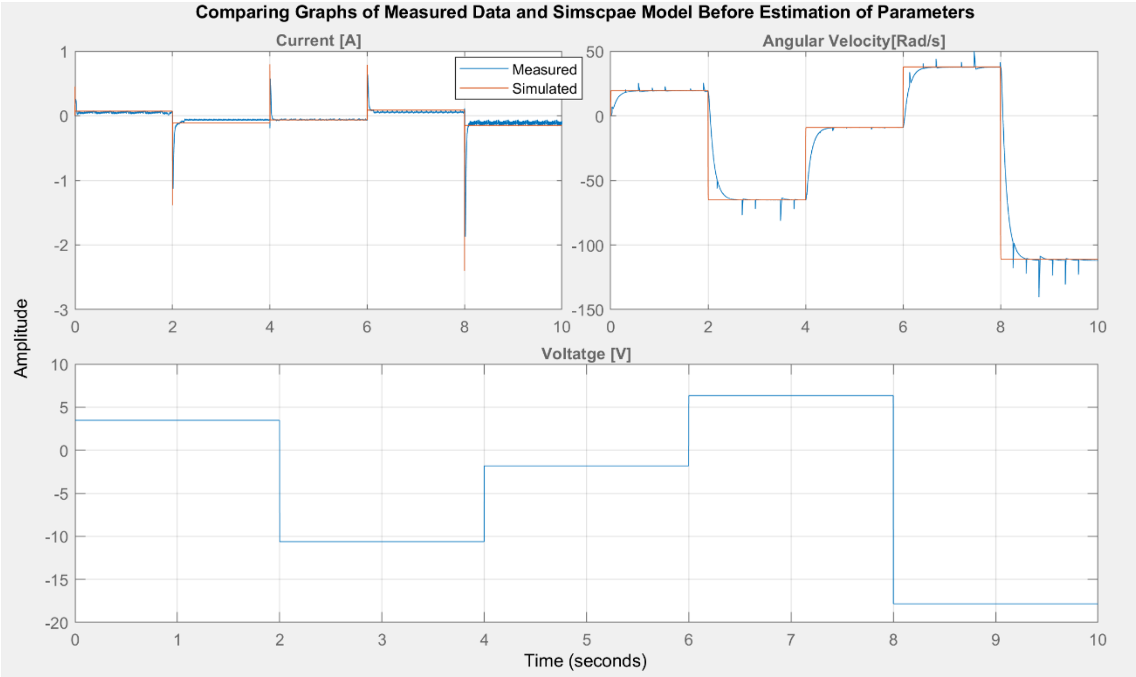


Fig. 6-11: Comparing Measured data graphs of angular velocity and current with graphs od DC Simscpae model before Estimation

For making the model graphs look like measured data, we use the Simulink Design Optimization tool specifically Parameter Estimation App.

Parameter Estimation App is a feature that enables modifying parameters automatically to reach more accurate parameters that match with measured data.

The data that will be needed from DoubleDrive simulation to estimate the parameters are:

- Input data of voltage
- Output measured data of angular velocity
- Output measured data of current

After opening the window of Parameter Estimation we start as following. Firstly, specifying the DC motor parameres that are needed to be reevaluated and selecting the minimum and maximum values of them. Next, choosing a new experiment in which the measured data can be entered. Then, adding the plot of the experiment as well as plotting model response and parameter trajectory. Finally, we view all graphs together, run the estimation button, and wait for the final results. After some iterations, the results change automatically. Graphs of the evaluation are shown in Fig. 6-12

The new parameters of the DC model, which are more accurate, are regenerated to the model, so the Simscape model now is more reliable and compatible than was before.

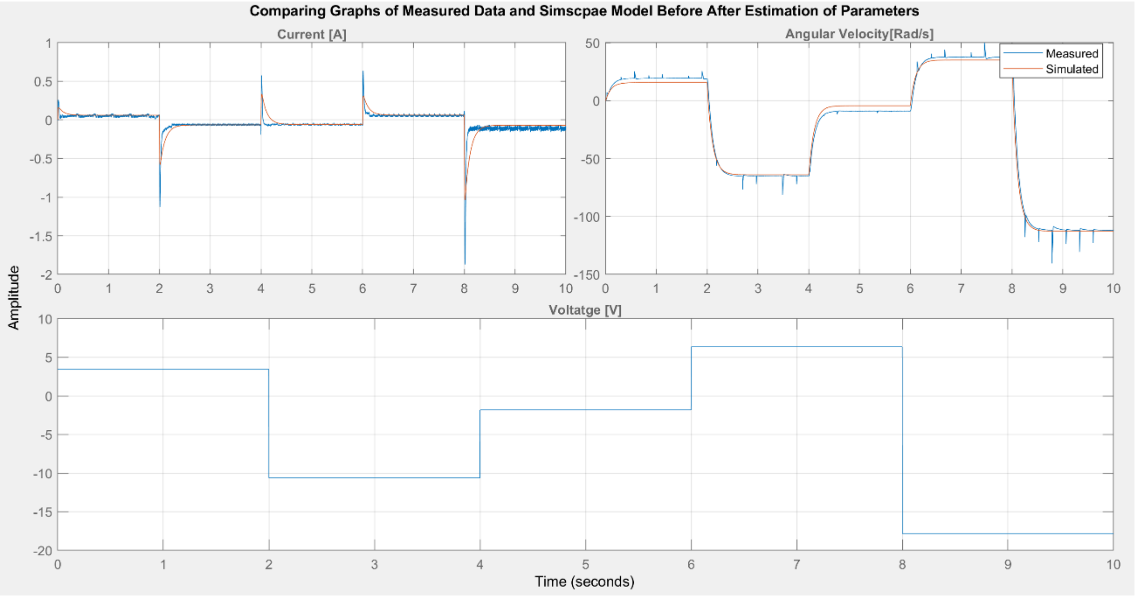
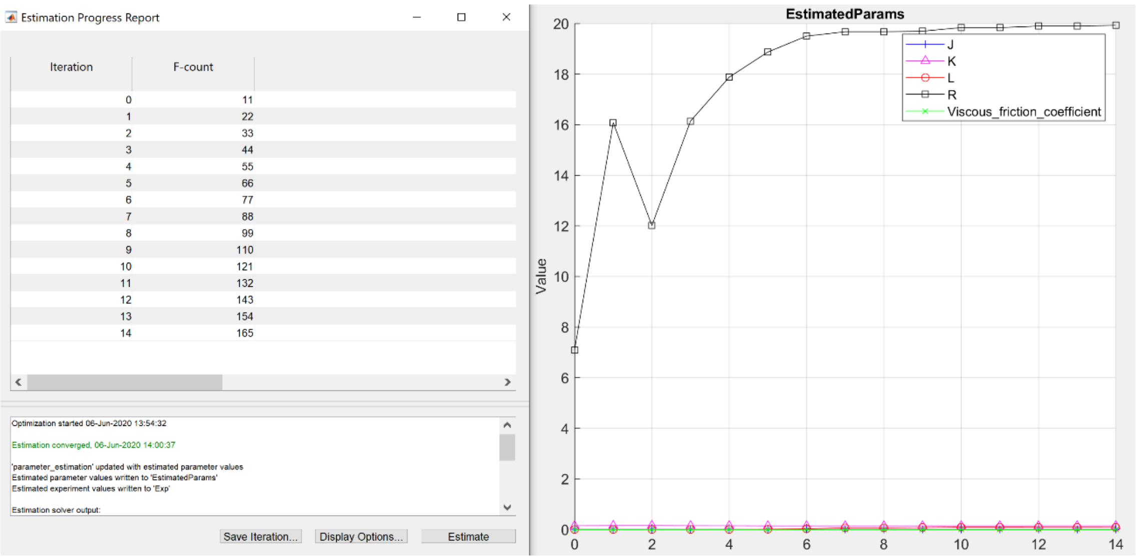


Fig. 6-12: Estimation of parameters

The changing process of the parameters is shown in Fig. 6-13. The Figure also shows the final report of the estimation.



Iteration	F-count
0	11
1	22
2	33
3	44
4	55
5	66
6	77
7	88
8	99
9	110
10	121
11	132
12	143
13	154
14	165

```
Optimization started 06-Jun-2020 13:54:32

Estimation converged, 06-Jun-2020 14:00:37

'parameter_estimation' updated with estimated parameter values
Estimated parameter values written to 'EstimatedParams'
Estimated experiment values written to 'Exp'

Estimation solver output:
```

Fig. 6-13: Final report and parameter trajectory

The final results of the parameters form the trajectory Fig. 6-13 are shown in Tab. 6-3. These parameters now are more accurate and can be regenerated to the model, so the Simscape model becomes more reliable.

Tab. 6-3 Parameters after estimation

Parameter	Value	Unit
R	19.925	Ω
L	0.0886	Henry
K	0.1461	$\mathrm{V/Rad}.s^{-1}$
J	0.00012	$Kg.m^2$
Viscous_friction_coefficient	$1.1672 * 10^{-4}$	N.m/(rad/s)

6.4 Model optimization

To reach a specific angle or a desired speed for the DC motor, a controller such as PI must be used. However, the response of the controller might not be as accurate as required. Therefore, the optimization of a model is an important step for achieving the required results and bettering the performance.

Optimization of the controller parameters can be done within the Response Optimization App. It is a feature in Simulink that enables tuning of the controller parameters with some specific requirements as well.

The steps for model optimization are as follows. Firstly, choosing the suitable blocks from Simulink Design Optimization in Simulink Library. Depending on the purpose, we choose the block such as tracking specific input or step response or bounds of signals. More than one required signal can be used and optimized at the same time. Secondly, opening Response Optimization from the bar above. In Response Optimization, the required graphs which were determined before will be automatically shown. Then we choose the parameters of the controller that are needed to be modified in our case K_p and K_i. Additional requirements can be added as graphs. The required graph for this model case is that the current should not exceed $(\pm 3\ A)$. After that, plotting the parameter trajectory in which the changes of parameters will occur. Finally running the optimization button and waiting until the optimization converges or stopping it when achieving required values. With each time of iteration, the response graphs change automatically showing the new results. Fig. 6-14 shows the graphs of the required optimization for the DC motor model before reaching the required results.

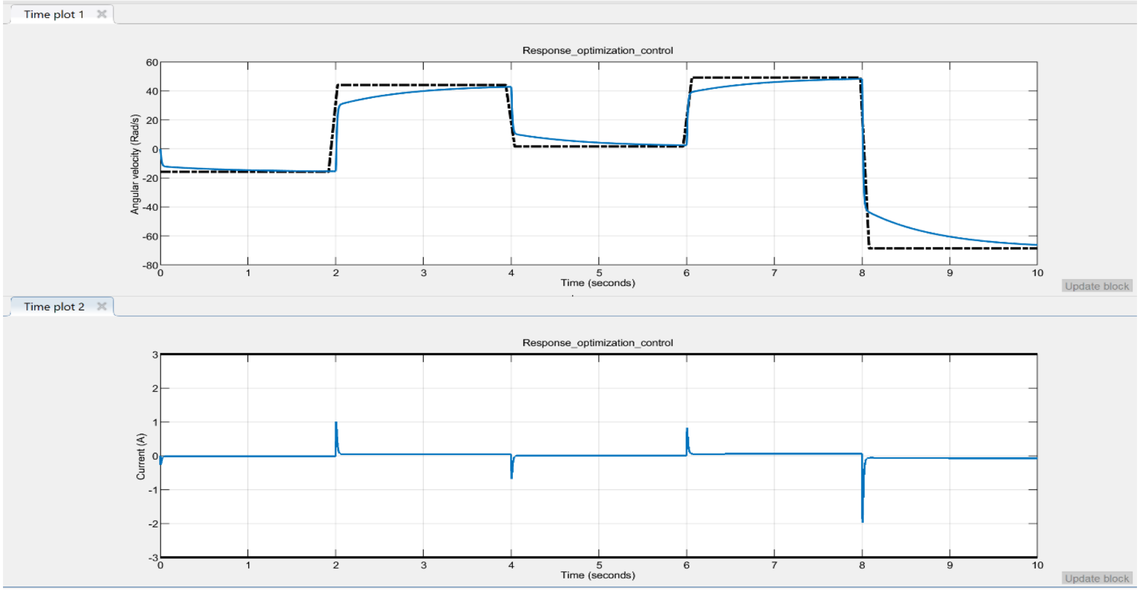
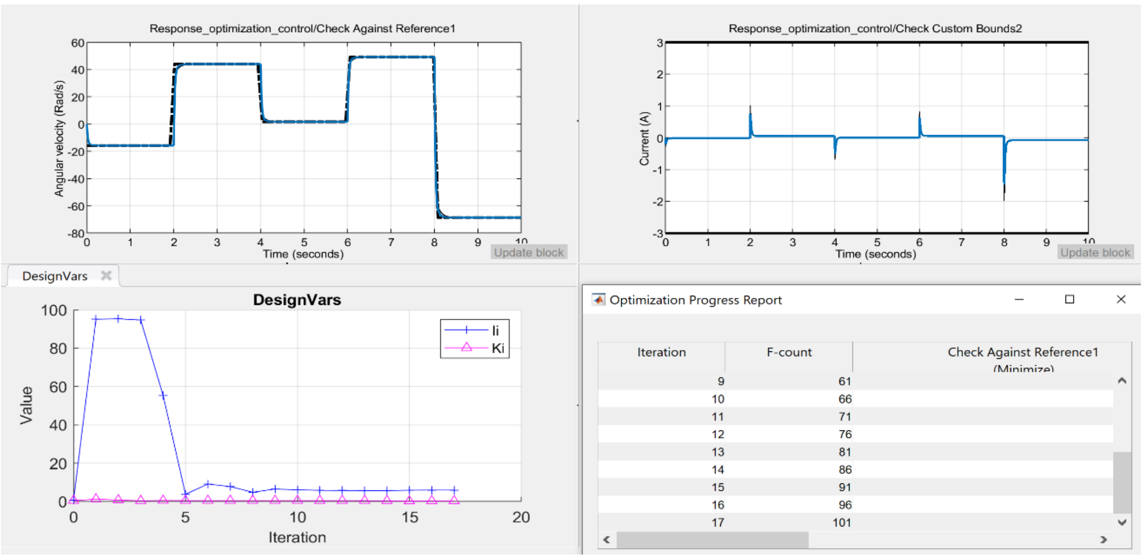


Fig. 6-14: Angular velocity and current before model optimization

Fig. 6-15 shows the graphs with the required results for the DC model with its controller. Fig show number of iteration, counts or running the model, and the bounds.



Iteration	F-count	Check Against Reference1 (Minimize)
9	61	
10	66	
11	71	
12	76	
13	81	
14	86	
15	91	
16	96	
17	101	

Fig. 6-15: Angular velocity, current, Iterationpn plot, and progress report

The results of the two constants of the PI controller are compared with Tab. 6-4

Tab. 6-4 PI values after optimization

	Parameter	Value	Parameter	Value
Before Model optimization	Ki	0.5	Ii	0.8
After Model optimization	Ki	0.34	Ii	5.95

7 REAL-TIME SIMULATION

7.1 Code Generation

It is a feature in Simulink that enables converting blocks which represent the models into c/c++ code that can be used in real embedded systems. In our case, the system target file is Simulink Desctop Real-Time that is used to convert the DC Motor Simcsacpe model to c code.

The importance of code generation is that it increases the speed of the development process and saves time. In other words, instead of writing the code in c from the beginning and losing time, the model can automatically be generated to c/c++ code.

Code generation is one of the steps of the V-Model workflow of automative manufacturing as shown in Fig. 7-1[11]

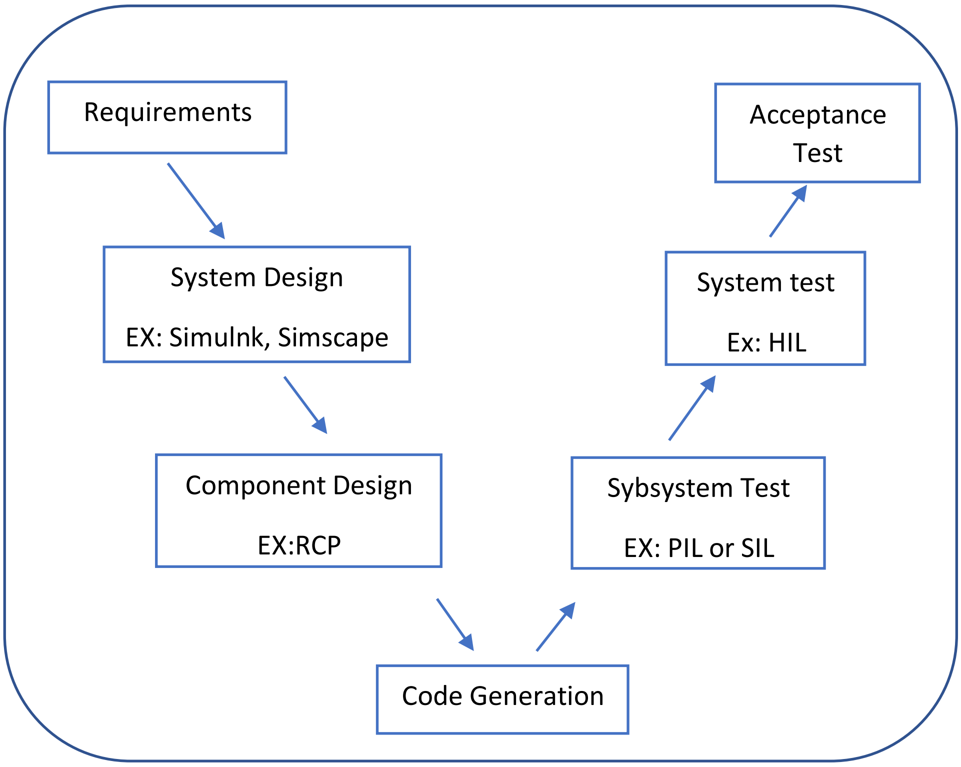


Fig. 7-1: V-model prototype

7.2 HIL Simulation

HIL is an abbreviation of Hardware-in-the-loop. It is a process that is used as the final procedure before making the final product. It can be used to test sophisticated models in real-time embedded systems. Both the plant, which can include many systems, and

the controller can be deployed to MCUs and modeled at the same time as real-time simulation. In other words, the simulation time of HIL matches with the time passing in the real world. HIL simulation can be represented in Fig. 7-2
The benefits of using HIL are [12]:
1- Safe testing for applications that might have some risks in real life.
2- Test for systems and devices that have not been produced yet.
3- Saving time and money.
4- Ability to work constantly.

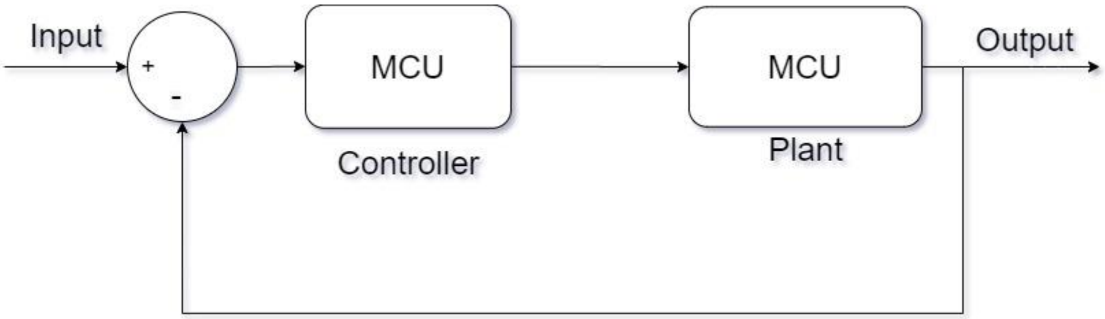


Fig. 7-2: HIL Simulation

7.3 Verifying Simscape Models for HIL

In order to move from Simscape desktop simulation to real-time simulation, the following aspects must be considered and modified until reaching precise results. The aspects are slover choise, number of slover iteration, step size, and model size.
The steps that are necessary to follow for verifying Simscpae models are represented in Fig. 7-3 [12]
For steps number one and two we use variable- step solver for the model and there must not be zero-crossing events.
For step 3 in the chart, an important notice is that:

- If the system is non-stiff, then the explicit fixed-step solver is the recommended one.
- If the system is stiff, then an implicit solver is highly recommended

For step number 4 we manipulate the number of nonlinear iterations and step size until reaching the required results.
For steps number 5 and 6 we deploy the model in a real-time platform and we make sure that it functions properly otherwise, some modifications must be applied.

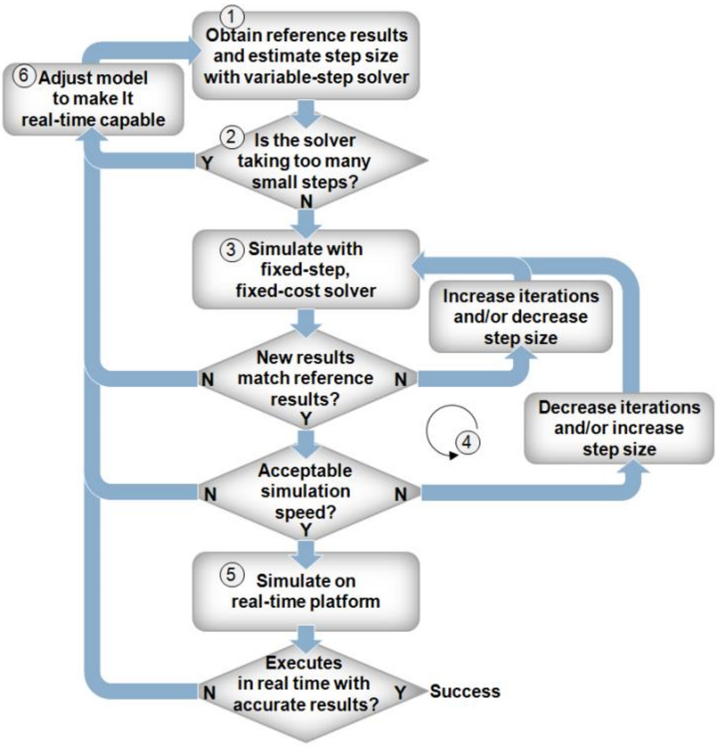

1 Obtain reference results and estimate step size with variable-step solver
2 Is the solver taking too many small steps?
Y
N
3 Simulate with fixed-step, fixed-cost solver
Increase iterations and/or decrease step size
New results match reference results?
N
N
Y
4
Decrease iterations and/or increase step size
Acceptable simulation speed?
N
N
Y
5 Simulate on real-time platform
Executes in real time with accurate results?
N
Y
Success
6 Adjust model to make It real-time capable

Fig. 7-3: Verifying Simscape models for HIL

7.4 HIL Experiment with MF624 and PIC Kit

HIL experiment has been done for two cases:

7.4.1 HIL experiment to control Angular Velocity

MF 624 card has been chosen to represent the plant which is the DC motor model. And the PIC Kit is used to represent the controller[13]. The HIL chart of the experiment is represented in Fig. 7-4. The models are attached to the attachments with the name of the HIL Experiment.

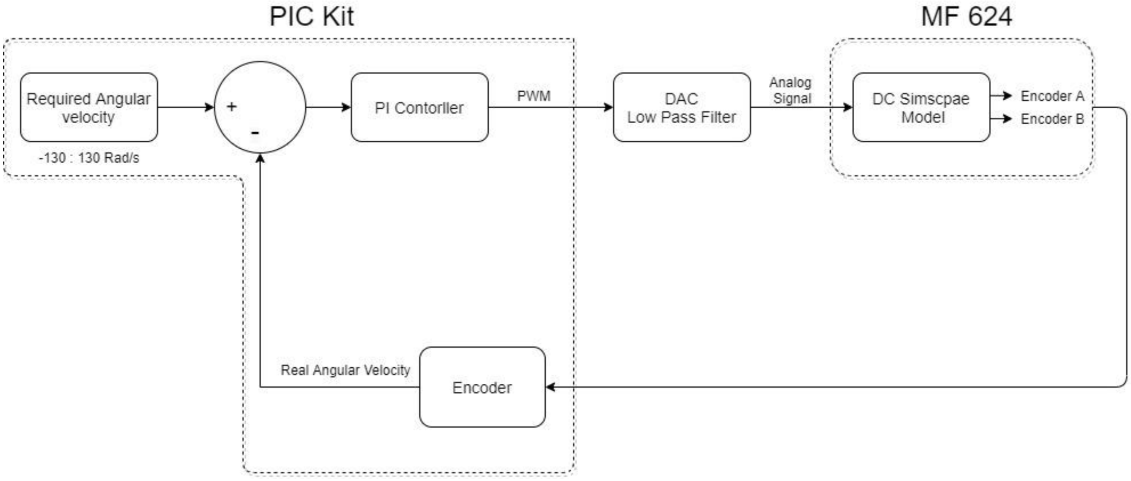


Fig. 7-4: HIL chart with MF624 and PIC Kit for controlling w

The hardware connection between the MF624 card and PIC kit is shown in Fig. 7-5.

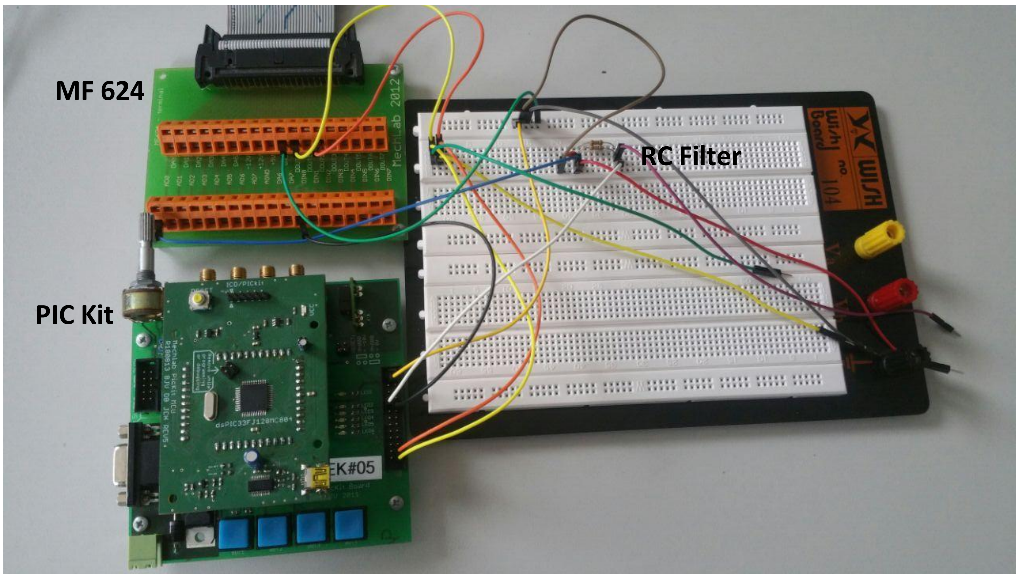


Fig. 7-5: Hardware connections for HIL Experiment

The wire connections between MF624 card and PIC Kit are represented in Fig. 7-6. The figure also includes the low pass filter and its connection to the PIC Kit and MF624 card.

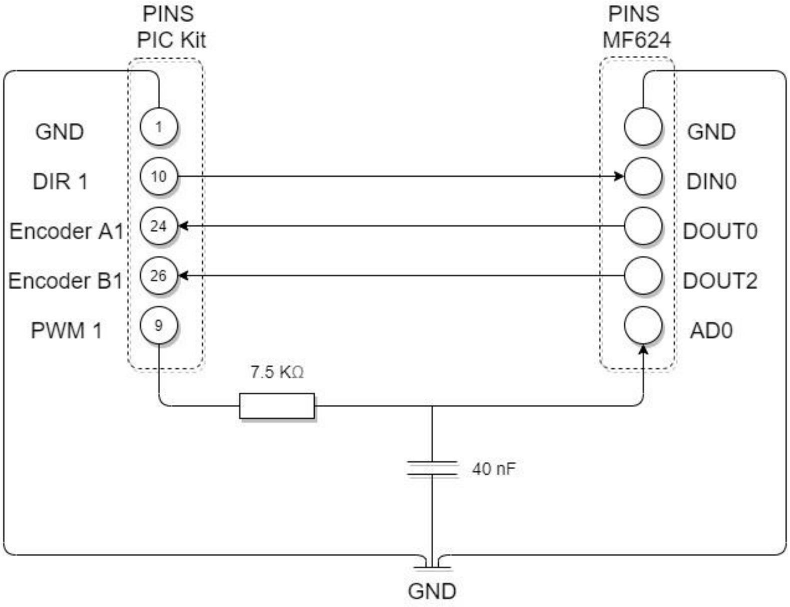


Fig. 7-6: Wire Connections between MF624, low pass filter, and PIC Kit

The filter that is used to convert the PWM signal to an analog signal is a passive RC filter. The frequency of the PWM is 5 kHz so the cut-off frequency has been chosen to be a value between 500 to 1000 Hz. The Value of the resistor is chosen to be 7.5 kΩ and the value of the capacitor is chosen to be 40 nF. The calculation of the cut-off frequency is shown in Equation 25

$$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi\, 7.5\ *10^3 * 40 * 10^{-9}} = 530.5\ Hz \qquad (25)$$

The result of the HIL experiment compared with MIL is represented and compared in Fig. 7-7.

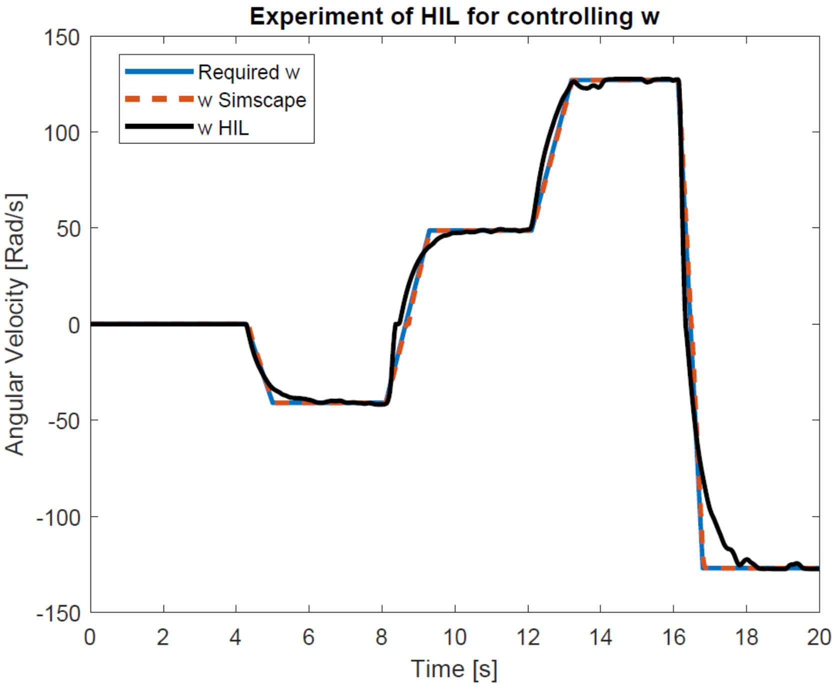


Fig. 7-7: Experiment of HIL with velocity

7.4.2 HIL experiment to control Position

The same hardware and connections are used in this experiment as controlling the velocity, but this time for controlling position. The chart that represents the experiment is shown in Fig. 7-8.

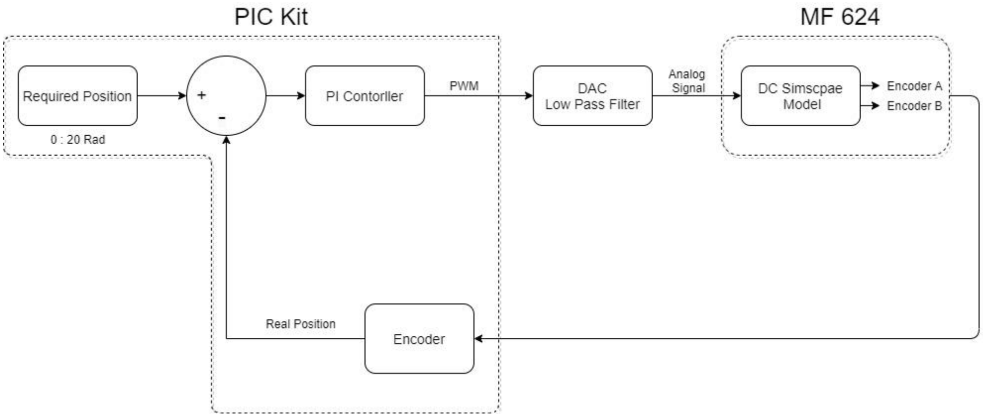
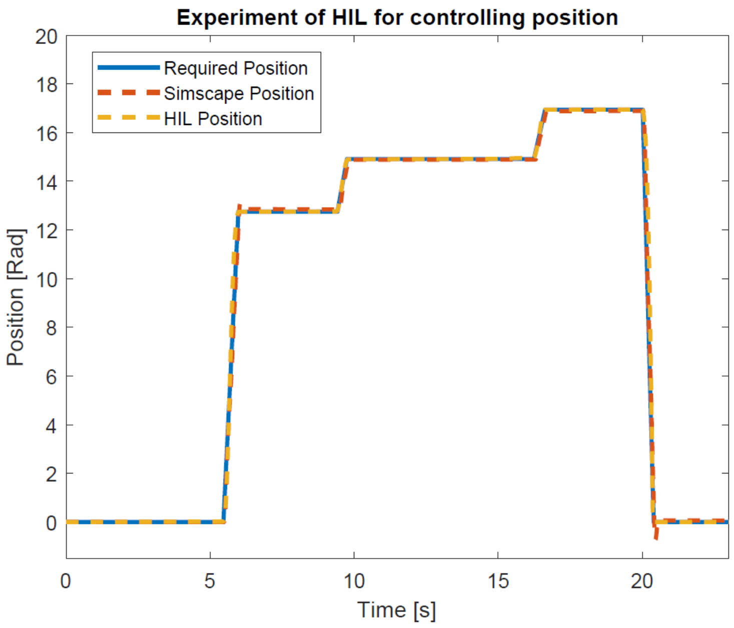


Fig. 7-8: HIL chart with MF624 and PIC Kit for controlling position

The result of the experiment is shown in Fig. 7-9 compared with MIL.



8 CONCLUSION

The main goal of this thesis is to create some mechatronics Models in Simscape and then generating c code in order to perform the HIL experiment.

Firstly, an introduction about Simscpae tool has been written with a description of its options. The Simscape tool is a very strong and effective tool for modeling. It simplifies complex mathematical models into models that represent real-world physical elements by blocks. Simscpae library includes different domains of different systems. Moreover, multiple different domains can be integrated in Simscape as one system, and then generating c/c++ code can be applied. Furthermore, the add-ons libraries, which are Simscape Electrical, Multibody, Driveline, and Fluids, include more advanced blocks to simulate real systems and devices.

Secondly, some mechatronics models have been created and simulated. The models are Brush DC Motor, BLDC Motor, Spring-Damper-Mass systems, Double Pendulum, Kinetic Mechanism. These models are created with Simscape and standard Simulink modeling. Some of them are also modeled with Multibody. The results of the different approaches of modeling are compared in graphs.

Thirdly, one of the models that have been simulated was chosen to be further studied in the real world. The model is a DC motor. The parameters of the DC motor were calculated form DoubleDrive device in Mechlab with the help of the hardware of MF624 and the software of Real-Time windows Target in Simulink. The Parameters, that were calculated are $(R,\ K,\ L,\ J,$ Breakaway_friction_torque, Breakaway_friction_velocity, Coulomb_friction_torque, Viscous_friction_coefficient.) The DC motor parameters calculated from the graphs were then compared with the measured data of the DC motor with the help of the Simulink Design Optimization tool. Firstly, I used the Sensitivity Analysis App so that I can find the most effective parameters in the model. Secondly, Parameter Estimation App is used also to reevaluate the calculated parameters so that the calculated parameters become more identical to the real DC motor. Thirdly, Response Optimization App was used to reevaluate the parameters of the PI controller of the DC motor so that it tracks the required signal and does not go over the limit of the current.

Finally, the DC Simscape model with the new estimated parameters was used as the plant of the HIL experiment. Signals of encoders were taken from the model and sent from MF624 by wire connections to the PIC Kit. The PIC Kit includes the controller that controls the DC motor and then sends the required signal back to the MF624 card to close the loop. The Experiment was done twice. Once for controlling the angular velocity and the second time for controlling the position of the motor. The hardware of MF624 and wire connections might not be the best choices for the HIL experiment especially that there were some delays and missed data sometimes. Maybe if other hardware were used, the results would be more stable and more identical to MIL.

REFERENCES

[1] MathWorks.[US], Simscape user‘s guide, R2019b online: https://ww2.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_ug.pdf

[2] MathWorks, Simscpae Electrical, Retrieved March 15, 2020, online: https://www.mathworks.com/products/simscape-electrical.html

[3] MathWorks, Simscpae Driveline, Retrieved March 15, 2020, online: https://www.mathworks.com/products/simscape-driveline.html

[4] MathWorks, Simscape Fluids, Retrieved March 16, 2020, online: https://www.mathworks.com/products/simscape-fluids.html

[5] MathWorks, Simscape Multibody, Retrieved March 16, 2020, online: https://www.mathworks.com/products/simmechanics.html

[6] Prof. Yon-Ping Chen, 2015, Modeling DC motor, from http://ocw.nctu.edu.tw/course/dssi032/DSSI_2.pdf

[7] Mathworks, BLDC Motor Control, Retrieved April 12, 2020, online https://www.mathworks.com/solutions/power-electronics-control/bldc-motor-control.html

[8] Math24, 2020, Double Pendulum, Retrieved April 5, 2020, online https://www.math24 .net/double-pendulum/

[9] MechLab team, Institute of Solid Mechanics, Mechatronics and Biomechanics, Faculty of Mechanical Engineering, Brno University of Technology, April 4, 2016, Real-Time Toolbox: https://moodle-archiv.ro.vutbr.cz/pluginfile.php/474414/mod_resource/content/1/RealTimeToolbox_DoubleDrive_R2016_04_04.pdf

[10] Ivan Virgala; Michal Kelemen, 2013, Experimental Friction Identification of a DC Motor, from file:///C:/Users/42077/Downloads/10.5923.j.mechanics.20130301.04.pdf

[11] MathWork, V-Model for System Development, Retrieved April 10, 2020, online https://www.mathworks.com/help/rtw/gs/v-model-for-system-development.html?s_tid=blogs_rc_6

[12] MILLER, Steve; WENDLANDT, Jeff. Real-time simulation of physical systems using Simscape. MATLAB News and Notes, 2010, 1-13.

[13] Josef Vejlupek; Barnabas Dobossy, PIC EduKit User Guide, ISMBB, Mechatronics Laboratory, Faculty of Mechanical Engineering, Brno University of Technology, 2018

[14] Mathworks, Simscpae, Retrieved June 23, 2020, online: https://www.mathworks.com/products/simscape.html#:~:text=Simscape%20helps%20you%20develop%20control,components%2C%20domains%2C%20and%20libraries.

LIST OF ATTACHMENTS

1st Model

2nd Model

3rd Model

4th Model

5th Model

Calculating Parameters

Sensitivity Analysis

Parameter Estimation

Response Optimization

HIL Experiment

LIST OF FIGURES

Figure 1: Simscape domains 7

Figure 3-1: Across and Through variables 9

Figure 3-2: Through positive variable 10

Figure 3-3: Through negative variable 10

Figure 3-4: Color Lines of Simscape domains 12

Figure 3-5: Rotational Electromechanical converter 12

Figure 4-1: Foundation Library 13

Figure 4-2: Utilities Library 13

Figure 4-3: Add-ons products 14

Figure 4-4-1: PS-S converter 15

Figure 4-4-2: S-PS converter 15

Figure 4.5: Solver configuration block 16

Figure 4-6: Solver configuration block parameters 16

Figure 4-7-1: ER and TR blocks 17

Figure 4-7-2: MTR and MRR blocks 17

Figure 4-7-3: MR and HR bocks 17

Figure 4-7-4: GAR and TLAR blocks 17

Figure 4-7-5: 2PFR block 17

Figure 4-8: Logging simulation data 18

Figure 4-9: Opening simulation data 19

Figure 4-9: Opening simulation data 20

Figure 4-11: Solver Profiler 20

Figure 4-12: Solving Zero Crossing 20

Figure 5-1: DC motor model 21

Figure 5-2: Simulink and Simscape results of angular velocity 23

Figure 5-3: Simulink and Simscape results of the current 23

Figure 5-4: BLDC Model 23

Figure 5-5: Controlling the angular velocity of BLDC 24

Figure 5-6: Position of the rotor 24

Figure 5-7: Mass-spring-damper system 1 25

Figure 5-8: Displacement of m1 25

Figure 5-9: Mass-spring-damper system 2 26

Figure 5-10: Displacement of m2 26

Figure 5-11: Mass-spring-damper system 2 26

Figure 5-12: Displacement of m3 26

Figure 5-13: Double pendulum 27
Figure 5-14: Angle of double pendulum by Simulink and multibody 27
Figure 5-15: Kinetic Mechanism and Results of Joint 4 28
Figure 6-1: Real-Time Windows Target Library 29
Figure 6-2: Mechlab DoubleDrive 29
Figure 6-3: Measured data of Angular velocity 30
Figure 6-4: Measured data of current 30
Figure 6-5: Measured data of Angular velocity 31
Figure 6-6: Measured data of current 31
Figure 6-7: Measured data of Tau Electrical 31
Figure 6-8: Measured data of Friction 32
Figure 6-9: Measured data of Tau Mechanical 33
Figure 6-10: Evaluation of parameters for current (left side) and angular velocity(right side)34
Figure 6-11: Comparing Measured data graphs of angular velocity and current with graphs od DC Simscpae model before Estimation 35
Figure 6-12: Estimation of parameters 36
Figure 6-13: Final report and parameter trajectory 36
Figure 6-14: Angular velocity and current before model optimization 38
Figure 6-15: Angular velocity, current, Iterationpn plot, and progress report 38
Figure 7-1: V-model prototype 39
Figure 7-2: HIL Simulation 40
Figure 7-3: Verifying Simscape models for HIL 41
Figure 7-4: HIL chart with MF624 and PIC Kit for controlling w 41
Figure 7-4: Hardware connections for HIL Experiment 42
Figure 7-6: Wire Connections between MF624, low pass filter, and PIC Kit 42
Figure 7-7: Experiment of HIL with velocity 43
Figure 7-8: HIL chart with MF624 and PIC Kit for controlling position 43
Figure 7-9: Experiment of HIL with position 44

LIST OF TABLES

Table 6-1 Calculated parameters 33

Table 6-2 Recommended parameters to start the estimation 34

Table 6-3 Parameters after estimation 37

Table 6-4 PI values after optimization 38

LIST OF ABBREVIATIONS

HIL	Hardware-in-the-loop
MCU	Microcontroller unit
DC	Direct current
BLDC	Brushless direct current
SIL	Software-in-the-loop
PIL	Processor-in-the-loop
RCP	Rapid control prototyping
MIL	Model-in-the-loop
CAD	Computer-aided design
PI	Proportional-integral
PWM	Pulse width modulation
DAC	Digital-to-analog converter

NOMENCLATURE

R_a	[Ω]	Resistance
L_a	[Henry]	Inductance
k_b	[V/Rad.s^{-1}]	EMF constant
k	[V/Rad.s^{-1}]	Motor constant
v_b	[V]	Back EMF
v_s	[V]	Voltage Source
w	[Rad/s]	Angular Velocity
i_a	[A]	Current
J_M	[kg.m^2]	Moment of inertia
T_M	[N.m]	Torque
B_M	$[N/ms^{-1}]$	Friction coefficient
K_T	[V/Rad.s^{-1}]	Torque constant
m	[kg]	Mass
a, $\ddot{\alpha}$, $\frac{d^2y}{dt^2}$	[m/s^2]	Accelaration
k_s	[N/m]	Spring stiffness
y, α	[m]	Displacement
b	[N/m.s^{-1}]	Damping constant
v, $\dot{\alpha}$, $\frac{dy}{dt^2}$	[m/s]	Velocity
l	[m]	Length of rods
g	[m/s^2]	Gravitational accelaration
K_p	[-]	Proportional coefficient
K_i	[-]	Integral coefficient