Modeling and Hardware-in-the-loop Simulations of Contactor Dynamics

Electronics and Hardware

Dan Tjerngren
2014

Master of Science in Engineering Technology
Engineering Physics and Electrical Engineering

Luleå University of Technology
Department of Computer Science, Electrical and Space engineering
Modeling and Hardware-in-the-loop Simulations of Contactor Dynamics

Electronics and Hardware

Dan Tjerngren

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

June 2014
Abstract

The subject of this master thesis is to develop a hardware-in-the-loop simulation environment representing an ABB contactor’s dynamics. A contactor is an electrically controlled mechanical switching device, used where large currents can occur. The hardware-in-the-loop concept means that hardware is connected to computer models simulated in a real-time simulator.

In this thesis, the contactor is modeled by dividing it into smaller subsystems. The subsystems correspond to the contactor’s electronics, electromagnetics and mechanical components. The subsystems are mainly realized as computer models. The electronics are also partly realized as hardware, and constitutes the hardware in the setup. The models are implemented in MATLAB and Simulink, and a hardware interface is constructed to connect the hardware part to a real-time simulator from dSPACE.

The focus of this report is the modeling of the electronics and the design of the hardware interface. The work was performed in collaboration with another student and his report focuses on the modeling of the electromagnetics and the mechanics as well as software implementations.

Measurements are collected from a real contactor and are used to validate the hardware-in-the-loop simulations. The conclusion from the validation is that the simulated contactor behavior corresponds well with a real contactor.
Acknowledgment

I would like to thank ABB for giving me the opportunity to perform this master thesis. It has been challenging and interesting to work with a larger project, where I have had to use several aspects from my education at Luleå University of Technology.

ABB’s employees have been helpful and easy to work with, and I am thankful to all of them. I especially want to thank Mattias Rehnman, my supervisor at ABB, as well as Ove Coman, Gunnar Johansson and David Karlen for their support and expert knowledge.

I am also very thankful to Jonny Johansson, my examiner and supervisor from the university.

My family has given me great support during the thesis work, especially Jon Tjerngren. He was the student whom I worked with during this project. Our combined effort made it possible to finish the work. We had a great time and many valuable discussions where we helped each other.

Västerås, June 2014

Dan Tjerngren
# Contents

1 Introduction .................................................................................................................. 1
   1.1 Background ........................................................................................................... 1
   1.2 Literature Study ................................................................................................... 2
   1.3 System Overview ................................................................................................. 3
   1.4 Purpose ................................................................................................................ 4
   1.5 Work Distribution ............................................................................................... 4
   1.6 Method and Material ......................................................................................... 6
   1.7 General Limitations ........................................................................................... 6
   1.8 Abbreviations ..................................................................................................... 7

2 Contactor Modeling .................................................................................................... 9
   2.1 System Description ............................................................................................ 11
      2.1.1 Subsystems .................................................................................................. 12
      2.1.2 Modeling Advice from ABB ..................................................................... 12
   2.2 ECC .................................................................................................................. 13
      2.2.1 Model Simplifications ............................................................................... 14
      2.2.2 Input Voltage and Rectifier ...................................................................... 14
      2.2.3 PLC Control .............................................................................................. 15
      2.2.4 Voltage Monitor ....................................................................................... 15
      2.2.5 Coil Current Circuit ................................................................................. 16
      2.2.6 Coil Current Monitor ............................................................................... 19
   2.3 Electromagnet ..................................................................................................... 20
      2.3.1 Magnetic Circuit Approach ...................................................................... 20
      2.3.2 Lookup Tables .......................................................................................... 20
   2.4 Mechanics ......................................................................................................... 22

3 Matlab and Simulink Implementation ........................................................................ 25
   3.1 Subsystem Blocks .............................................................................................. 26
      3.1.1 Simulation Control ..................................................................................... 26
      3.1.2 ECC .......................................................................................................... 26
      3.1.3 Electromagnet ............................................................................................ 27
      3.1.4 Mechanics .................................................................................................. 27
   3.2 Parameter Values ............................................................................................... 28
   3.3 Non-real-time Simulations and Validation .......................................................... 28
      3.3.1 ECC .......................................................................................................... 28
      3.3.2 Electromagnet ............................................................................................ 29
1 Introduction

This master thesis was made for ABB at their Control Product unit in Västerås, Sweden, which is part of ABB’s Low Voltage Products division. A few of the division’s products are low voltage circuit breakers, switches, motor softstarters and enclosures and cable systems which purpose is to protect users from harm. The motor softstarters are used to start and stop motors gradually which increases the motors life expectancy and decreases the cost for unnecessary repairs (ABB, 2014).

One of the products that the Control Products unit designs and manufactures is contactors. A contactor is an electrically controlled mechanical switch that is used in various implementations where large currents can occur, for example one of the largest applications is the starting of electrical motors. ABB has several different contactors depending on the application; see Figure 1.1 for three of them. An electrical control card (ECC) regulates the contactor to close, hold or open the switch. The switching is made by a magnetic force that affects the mechanical system of the contactor. The force is created by an electromagnet that is regulated by the ECC.

![Figure 1.1: Different contactor sizes.](image)

1.1 Background

The ECC contains a microcontroller and when updating old or writing new control algorithms for it, it is difficult and time consuming to validate the function of these changes in a real environment. Also since ABB has many different types of contactors, it is even more difficult to ensure that the changes work for every type. Currently the tests are done by measuring interesting signals and then performing visual assessments.

To simplify and speedup the development process, ABB is interested in using a hardware-in-the-loop (HIL) simulation setup for the contactor product. ABB is currently doing the same for their soft starter products. In the HIL setup, part of a system is realized as hardware and the rest is implemented as computer models. The interactions between these parts are handled by a real-time simulator. In this thesis the ECC microcontroller is the main component of the hardware part. This will allow for easier and faster tests of the contactor’s behavior when the microcontroller’s software has been changed. It will also be easier to setup and control exact test cases when the model parameters can be set to known values.
1.2 Literature Study

A system’s HIL interface can be on different levels depending on the HIL setup’s purpose (dSPACE, 2014). For example an electronic control unit (ECU) controlling an electrical motor can be interfaced at the signal level, the electric power level or the mechanical power level.

The signal level can be characterized by implementing computer models of the mechanical system the motor is attached to, the electric motor and the power electronics. For this, the ECU has to be opened and directly be connected to the real-time simulator. The electric power level implements computer models of the mechanical system and the electric motor. In this case, an intact ECU can be used and connected to the real-time simulator. The mechanical power level only simulates the mechanical system, which means that a real motor is controlled by an ECU in the HIL setup.

There have been several studies about implementations performed with the HIL concept. Three of these are:

Shetty et al. (2001) treat the subject of using the HIL concept to improve mechatronic education at the University of Hartford. They write that the development of automatic controllers can be improved by using the HIL concept. For example, it gives the opportunity to test and optimize system models before committing to real hardware platforms. They conclude that HIL simulations are an effective way to perform system tests with computers.

Wu et al. (2004) investigate about using real-time HIL simulations to achieve a fast and low-cost developing procedure for digital controller design. Their first step was to implement a prototype controller model and a system model in a computer. Non-real-time simulations are then used to evaluate the performance of the controller model. The next step was to replace the simulated controller model with a real hardware controller. This increases the realism of the development of the digital controller. The system model was then simulated in real-time while it interacted with the hardware controller. Their conclusion was that using the HIL concept in development work is fast, safe and reliable.

Lin and Lipeng (2008) focus on how the HIL concept has been applied in electrical vehicle (EV) development in recent years. They state that almost all departments who work in this area have used HIL simulation in their research. For example, the Ford Motor Company has used HIL simulations to evaluate and improve EV control procedures. The French Institute National Polytechnic of Lorraine has made studies about battery and energy management and has developed a HIL system which they have used to improve hybrid power system performance. The Tsinghua University in China has used a HIL system when performing studies on drive systems where they replaced an actual motor with a computer model. The paper concludes that HIL simulations have improved the quality and reduced the cost of the development process of EV systems, they also state that better computer models have to be developed to increase the confidence level of simulation results.

The HIL concept has been used in several development projects in both the industry and the academic world. It is flexible and can be used in many different setups to increase the realism of system simulations.

The modeling of AC contactors has been studied in (Riba Ruiz, et al., 2010). They describe the mechanical and the electromagnetic components of a contactor and create a simulation model. Their model was used to simulate both the transient and steady-state response of the contactor system.
1.3 System Overview

In Figure 1.2 an example of a contactor application and a sketch of the HIL simulation setup are shown. In the top of the figure the contactor is used in an external circuit (ExC), in this example it acts as a switch for a motor. The contactor consists of the ECC, the electromagnet and the mechanical system. It also illustrates how they interact with each other.

The input voltage to the ECC fulfills either one or two purposes. The first purpose is always fulfilled, which is to act as a power supply. Depending on the ECC, the input voltage can either be alternating current (AC) or direct current (DC). There are several different contactors depending on the amplitude of the input voltage; the voltage is divided into the ranges 24 – 60V, 48 – 130V, 100 – 250V and 250 – 500V. The second purpose is optional and it is to act as the control signal for the contactor. If the control is not implemented by the input voltage, the contactor is instead controlled by a programmable logic controller (PLC) interface.

The shaded areas in the figure represent how the contactor system is divided into hardware and computer models for the HIL simulation setup, illustrated in the bottom part of the figure. Only some parts of the ECC are realized as computer models, the rest is used as the hardware of the HIL simulation. The electromagnet and the mechanics are also implemented as computer models. The hardware interface block illustrates the link that passes signals between the hardware and the real-time simulator, for example the control signals generated by the simulator.
1.4 Purpose
The purpose of this master thesis is separated into three parts. The first is to define the contactor hardware to be used in the HIL setup, and implement the remaining contactor as a computer model suitable for real-time simulations. The second is to integrate the model with a HIL simulator as well as creating the hardware interface between the remaining ECC’s hardware and the simulator. The last part is to validate the work by comparing the HIL simulation results against measurements from real tests.

The contactor has three distinct subsystems; these are the ECC, the electromagnetics and the mechanics. The computer models are implemented in MATLAB and Simulink because ABB had a real-time simulator, from the company dSPACE (dSPACE, 2014), designed for a HIL setup that supported these tools.

The purpose is summarized as:

- To investigate and define the contactor hardware to be used in the HIL simulation.
- To investigate and create a Simulink model of a contactor suitable for real-time simulations. The model is divided into:
  - Part of the ECC.
  - The electromagnet.
  - The contactor mechanics.
- To construct a hardware interface between the remaining ECC and the real-time simulator.
- To integrate the Simulink model with the HIL system.
- To perform real-time simulations of the HIL system.
- To validate the result of the HIL simulations by comparing against measurements from a real test.

1.5 Work Distribution
This master thesis is performed by two students. The other student is Jon Tjerngren who studied “Applied Physics and Electrical Engineering”, at Linköping University, with the profile “Control- and Information Systems”.

The work is divided between us throughout the thesis. I had main responsibility of the electrical models and constructing the necessary electronic hardware. Jon had main responsibility of the mechanical and the electromagnet models as well as the software implementations.

Since we studied at two different universities we made two separate reports. My report focuses on the details of my own work while Jon’s report covers his contribution. The shared work is described in both reports. Only summaries of the Jon’s work will be presented here and to get the complete coverage of this master thesis both reports have to be read. The details of our planning and how we distributed the work between us are described in Appendix A – Work Process.

Jon’s report (Tjerngren, 2014) is published at Linköping University.

The two reports were written in such a way that both reports can be read separately, and still give a comprehensive presentation of the work. Also, since the work has been closely interconnected it has not been possible to avoid overlap between the two reports. The overlap means that certain text sections of the reports are identical, similar or summaries of each other’s work.
The main reasons for sections with identical texts are:

- To avoid presenting different implementations of the combined work.
- To avoid ambiguity in the reports concerning the simulation results and the conclusions.
- Take advantages of working together to produce better content.

The same structure and similar headings have been used in the two reports because their contents will be merged into one report for ABB. Most of the figures are also identical since it was a more effective use of our time to only produce one set of figures. A detailed list of the overlapping sections in this report follows below.

- **Chapter 1 – Introduction**
  - In this report, I have written most of the chapter but it is similar to Jon’s introduction since it describes the same project. Only the literature study has been written together.

- **Chapter 2 – Contactor Modeling**
  - In this report, I have written the introduction and 2.1 but they are similar to Jon’s since they describe the same product.
  - In this report, I have written 2.2. We wrote a summary of this section together to be used in Jon’s report.
  - 2.3 and 2.4 are summaries of Jon’s modeling of the mechanics and electromagnetics. We wrote the summaries together.

- **Chapter 3 – MATLAB and Simulink Implementation**
  - The introduction was written together and is identical in both reports.
  - 3.1 contains my own texts as well as summaries.
    - In this report, I have written 3.1.1 and 3.1.2. We wrote summaries for these sections together to be used in Jon’s report.
    - 3.1.3 and 3.1.4 are summaries of Jon’s Simulink implementation work. We wrote these summaries together
  - 3.2 and 3.3 were written together and are almost identical in both reports.
    - 3.3.3 In this report, only parts of Jon’s simulation results are shown.

- **Chapter 4 – Hardware-in-the-loop Setup**
  - The introduction and 4.1 were written together and are identical in both reports.
  - In this report, I have written 4.2, 4.3 and 4.4. We wrote a summary for Jon’s report.
  - 4.5 is a summary of Jon’s Simulink implementation of the HIL software interface. We wrote the summary together.

- **Chapter 5 – Real-time Simulations and Validation**
  - The whole chapter was written together and it is almost identical in both reports.
    - 5.3.4 In this report, only parts of Jon’s simulation results are shown.

- **Chapter 6 - Discussion**
  - The whole chapter was written together and it is identical in both reports.

- **Chapter 7 – References**
  - Similar in both reports since several references are the same in both reports.

- **Appendix A – Work Process**
  - In this report, I have written the whole appendix but it is similar to Jon’s.

- **Appendix B – MATLAB Scripts**
  - The whole appendix was written together and it is identical in both reports.

- **Appendix C – DS2202 HIL I/O Board Pin Configuration**
  - This appendix is only used in this report.
1.6 Method and Material
Several tools and sources of information were used to complete the thesis.

The best available resource was the personnel at ABB, and they were always willing to help. At Control Products they have a lot of experience and knowledge about contactors. They shared this information during meetings, discussions and a few introductory lectures. Articles, books and the Internet were used to find more information.

The dSPACE simulator supports MATLAB and Simulink and it was a requirement from ABB to implement the contactor models in these tools. ABB provided access to a MATLAB and Simulink license, which was used for the implementation.

ABB also provided computers, offices supplies and workspaces. Access to a mechanical workshop and an electronics lab were provided and used to build the necessary hardware implementations. The contactor measurements, for the validation phase, where collected in a test room with help from ABB.

1.7 General Limitations
Due to the many variants of contactors and the different control implementations, the task from ABB was to make a general model of the contactor system. The simulation and validation work in this thesis is only focused on one contactor; it is ABB’s AF370 contactor which is controlled by a 100 – 250V AC 50/60 Hz input voltage signal.

Since the models are simulated in real-time, this imposes a constraint on the simulation’s calculation time. This constraint limits the details of the developed models. To find a suitable solution several contactor models, with different levels of complexity, is developed. It is more important to test the robustness of the microcontroller’s control algorithm than to get an exact representation of the real world. This made it acceptable to make certain model simplifications.

Several problems occur when the contactor is switching an ExC where a large current is conducted. One of the problems is that an electric arc is created between the contactor’s contacts when they are separated (Johansson & Karlen, 2014). An electric arc arises when the voltage over the contacts is large enough to ionize the air to the degree that it begins to conduct a current. This electric arc can weld the contacts together if the opening of the switch is to slow, or if the contacts bounce too much from mechanical impacts at the closing of the contactor. These implications are outside the scope of this master thesis and are not further considered here.
1.8 Abbreviations
Common abbreviations used in this report can be seen in Table 1.1.

Table 1.1: Used abbreviations.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog-to-Digital Converter</td>
</tr>
<tr>
<td>BJT</td>
<td>Bipolar Junction Transistor</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital-to-Analog Converter</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>ECC</td>
<td>Electronic Control Card</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>ExC</td>
<td>External Circuit (to the contactor)</td>
</tr>
<tr>
<td>EV</td>
<td>Electric Vehicle</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HIL</td>
<td>Hardware-In-the-Loop</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>LUT</td>
<td>Lookup Table</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>RTI</td>
<td>Real-Time Interface</td>
</tr>
</tbody>
</table>
2 Contactor Modeling

ABB has a large variation of contactors but their basic structure is similar. Figure 2.1 shows several components of an AF370 contactor.

Figure 2.1: Inside of a AF370 contactor.

Sketches over a general ABB contactor can be seen in Figure 2.2 and Figure 2.3.
Figure 2.2: Sketch illustrating the components of a contactor.

Figure 2.3: Sketch illustrating the contacts and the ExC connections.
2.1 System Description

Assuming that no ExC is connected and that a contactor is in its open state, and is kept there by the release springs, then the function of a contactor system can be describe as below.

A control signal to the ECC can come from either the input voltage or from a PLC interface. If this signal indicates to the ECC to close the contactor, it regulates the current through the electromagnet’s coil. This current creates a magnetic field around the coil which in turn applies a magnetic force on the movable core; this core is also called the anchor. The force acts in such a way that it closes the gap between the two cores. When this force is large enough to overcome the force of the release springs, the anchor will move and drag the contact bridge along with itself. The movable contacts are attached, by the connection springs, to the contact bridge and they will in turn move towards the fixed contacts.

It is the movable and fixed contacts that perform the actual switching of an ExC. When the contacts are brought together, a mechanical impact happens and a bouncing effect occurs. To reduce this bouncing, the connection springs are precompressed. The distances between the contacts is less than the distance between the cores, therefore the anchor continues to move towards the fixed core. When the cores hit each other more bouncing will occur.

When the contactor has been closed, the ECC reduces the coil current to save energy. The reduction is in comparison to the current used for closing, which is large to make sure that the contactor closes quickly. When the control signal indicates to open the contactor, the ECC will demagnetize the electromagnet’s coil by quickly reducing the coil current. In turn this will reduce the magnetic field around the coil and thereby weaken the magnetic force on the anchor. When this happens, the release and the connection spring forces will be large enough to push the contact bridge away. Most of the force comes from the connection springs which are stronger than the release springs.

The contact bridge will pull the anchor along with itself and the cores are separated. When the contacts have been separated, the force from the release springs brings the contact bridge back to its open steady state position. Figure 2.4 illustrates a closing procedure.

![Figure 2.4: Contactor closing procedure.](image-url)
2.1.1 Subsystems
The contactor system is divided into three subsystems for the modeling. These subsystems are called the ECC, the electromagnet and the mechanics.

In Figure 2.5 the signals to and between the subsystems can be seen. The ECC is connected to the external control signals. The input voltage will always be connected in an application, while the PLC input is optional. The interaction signals between the ECC and the electromagnet are the time dependent coil current, $i(t)$, and the coil inductance, $L(i, x)$, which depends on the coil current and the time dependent anchor position, $x(t)$. The signals between the electromagnet and the mechanics are the anchor position and the magnetic force, $F(i, x)$, which depends on the coil current and the anchor position. If effects from the ExC had been included, an external input would have been added to the mechanics block.

![Figure 2.5: Illustration of subsystem interactions.](image)

2.1.2 Modeling Advice from ABB
ABB’s Corporate Research division, in Västerås, has worked with modeling contactors in Dymola for other projects. During a meeting (Andersson & Fransson, 2014) advice was received on how to model the contactor for this thesis. This included important aspects to keep in the model as well as what kind of simplifications had worked for Corporate Research. Access to some of Corporate Research’s results was provided and could be used to compare against and validate the Simulink models developed in this project. Corporate Research also had experience with real-time simulations on a HIL setup, and gave advice on how to solve some real-time implementation issues.
2.2 ECC
The ECC’s purpose is to interpret external control signals and make the contactor to act accordingly. The ECC consists of several parts and are divided into two categories, one for the components which are realized as computer models and the other for the parts that are not. The separation is mainly done according to the limitations of the HIL simulator’s input/output (I/O) board. Circuits that are powered by 12 V or less are kept as hardware. A block diagram illustrating the ECC parts can be seen in Figure 2.6. The gray boxes represent simulated functions and they are described in this chapter. The white boxes represent parts that are kept as hardware and they are described in Chapter 4. The encircled boxes are called the coil current circuit. The dashed boxes represent optional functionality of the ECC. The following list shows each block’s category. The basic electrical equations are found in (Nilsson & Riedel, 2011).

- Hardware (white):
  - Microcontroller.
  - 11.4 V power supply (partially).
  - 3.3 V power supply.
  - Reset circuit.
- Simulated (gray):
  - Input voltage and rectifier.
  - PLC control.
  - Voltage and coil current monitor circuits.
  - Electromagnet.
  - Pulse width modulated (PWM) switch circuit.
  - Discharge/freewheeling circuit.

![Block diagram describing the ECC](image)

Figure 2.6: Block diagram describing the ECC. A white box signifies a function kept as hardware and a gray box signifies a simulated function.
2.2.1 Model Simplifications
Since the complete contactor model will be run in real-time, it has strict computation deadlines and therefore certain model simplifications are necessary. For the ECC this will mostly be to ignore transient behavior, for circuits with the function to switch between two states. Two examples are when the PWM switch turns on or off, and the rectifier is changing the sign of the input voltage.

2.2.2 Input Voltage and Rectifier
The input voltage to the ECC is either AC or DC. It is used to power the ECC and it can also be used as the control signal. If the input voltage is AC the rectifier is used for two purposes. The first is to generate a DC voltage, which is used to supply the microcontroller with power. The second is to make the current through the coil to only increase in one direction. This is important, since then the current through the electromagnet’s coil generates a magnetic field. This field then exerts a force on the anchor in only the closing direction. The rectifier is not necessary if the input voltage is DC.

The input voltage, $u_i$, is modeled by a sinusoidal function for the AC case and for the DC case the function is a constant. The function looks like:

$$
\begin{equation}
    u_i(t) = \begin{cases} 
        \sqrt{2}V_{RMS} \sin \left(2\pi \left(ft + \frac{\alpha}{360}\right)\right), & \text{AC case} \\
        \text{Constant}, & \text{DC case}
    \end{cases}
\end{equation}
$$

(2.2.1)

The simulation parameters that define the AC function and their respective unit are:

- The root mean square value: $V_{RMS}$ [V]. If multiplied with $\sqrt{2}$ it represents the signal amplitude.
- The frequency of the signal: $f$ [Hz].
- The phase shift of the signal: $\alpha$ [°].

The rectifier is emulated by taking the absolute value of equation (2.2.1).

To model a limited power output, $p_i$, for the input voltage source, the following limitation is imposed:

$$
    p_i(t) = i_{coil}u_i \leq p_{i,\text{max}}.
$$

(2.2.2)

That is, the power is simplified as the product between the coil current, $i_{coil}$, and the input voltage. The maximum power, $p_{i,\text{max}}$, is a simulation parameter and is constant during a simulation run. If equation (2.2.2) is not fulfilled the input voltage becomes:

$$
    u_i = \frac{p_{i,\text{max}}}{i_{coil}}.
$$

(2.2.3)
2.2.3 PLC Control

If the input voltage is not used as the control signal, the ECC is instead controlled by its PLC interface. There are two PLC inputs, called ON_N and OFF at the microcontroller pins. The N notation on ON_N indicates that it is inverted in the microcontroller. This is to make the contactor controllable by only one signal when ON_N and OFF are given by the same source. Furthermore there is a CTRL pin that enables the PLC functionality and this is set in hardware. Table 2.1 shows a PLC signal truth table and the corresponding ECC operation.

Table 2.1: PLC control.

<table>
<thead>
<tr>
<th>OFF</th>
<th>ON_N</th>
<th>ECC operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Close contactor</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Do nothing</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Open contactor</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Open contactor</td>
</tr>
</tbody>
</table>

The function of the PLC circuit, on the real ECC, is to set corresponding microcontroller pins to either high or low. The transient aspect of this circuit is not modeled in this thesis.

If the input voltage is used as the control signal, the two PLC signals are set high and the CTRL pin is connected to ground reference to disable the PLC control.

2.2.4 Voltage Monitor

The voltage monitor circuit is used by the microcontroller to measure the input voltage, $u_{i}$. This measurement is used to regulate the contactor. Figure 2.7 shows a schematic of the modeled circuit.

![Voltage Monitoring Circuit](image)

Figure 2.7: Schematic for the voltage monitoring circuit.
The derivation of the measured output signal, \( u_o \), follows as:

\[
    u_i = R_1i_l + u_o \tag{2.2.4}
\]

\[
    i_l = i_1 + i_2 = C \frac{du_o}{dt} + \frac{u_o}{R_2} \tag{2.2.5}
\]

\[
    u_i = R_1C \frac{du_o}{dt} + \left(1 + \frac{R_1}{R_2}\right) u_o \tag{2.2.6}
\]

\[
    u_o = \frac{1}{R_1C} \int_0^t \left( u_i - \left(1 + \frac{R_1}{R_2}\right) u_o \right) d\tau \tag{2.2.7}
\]

2.2.5 Coil Current Circuit

The circuit that affect the electromagnet, and in extension the contactor mechanics, is here called the coil current circuit. It mainly consists of the electromagnet’s coil, the discharge and freewheeling circuit, the PWM switch and the coil current monitor circuit.

In principle, the circuit is represented by the schematic to the left in Figure 2.8. It illustrates the three main paths the coil current can take; the paths are numbered in the figure. When the PWM controlled transistor is turned on and is conducting, the coil current takes path 1. Part of the rectified input voltage, \( u_i \), then falls over the coil, \( u_{coil} \), and the rest falls over the components in path 1. In this state the common diode in path 2 and 3 becomes reverse biased and therefore does not conduct the coil current. When the PWM transistor is turned off, the existing coil current begins to decrease and goes through either the release controlled transistor or the Zener diode. The decrease in current induces a negative coil voltage which is limited by either path 2 or path 3. Path 3 allows a larger voltage and makes the current to decrease faster and is called the discharge circuit. In comparison path 2 makes the current to decrease slower and is called the freewheeling circuit.
Figure 2.8: Principle circuit (left) and the simplified schematic (right) used in the modeling.

On the real ECC, the switching transistors are connected to drive circuits which in turn are controlled by the microcontroller. In this thesis two approaches are considered, the first one model the drive circuits and switch transistors with models for each of their components. The second approach replaces the drive circuits and switch transistors with ideal switches and the transistors’ electrical on state characteristics. The ideal switches are directly controlled by the microcontroller’s signals. The schematic representing the last approach can be seen to the right in Figure 2.8.

For the first approach, an Ebers-Moll (Sedra & Smith, 2004) bipolar junction transistor (BJT) model is implemented in Simulink; this model represents both the active and the saturation state of the BJT. Only the PWM switch’s drive circuit is implemented in Simulink and tested for evaluation purpose.

For the second approach a transistor’s on state characteristics is represented by a DC voltage source and a resistor in series. The diodes are represented by a piecewise linear model. Below a threshold forward bias voltage the diode does not conduct current. Above the threshold the characteristic is approximated by a straight line. The straight line is represented by a DC voltage source in series with a resistor.

The electromagnet’s coil is model by an inductor, \( L_{\text{coil}} \), that depends on the coil current, \( i_{\text{coil}} \), and the anchor position, \( x \). The inductance was modeled by using a LUT which ABB had developed, see Chapter 2.3 for more details. To include temperature dependence for the simulation, the coil resistance, \( R_{\text{coil}} \), is modeled by a linear function (Bentley, 2005) according to:

\[
R_{\text{coil}}(T) = R_0\left(1 + \alpha(T - T_0)\right). 
\]  

(2.2.8)

\( R_0 \) is the resistance at the reference temperature, \( T_0 \), and \( \alpha \) is the temperature coefficient.
Each path is represented by the same circuit setup, see Figure 2.9. The voltage source, $u_i$, was equal to the input voltage for path 1 and zero for path 2 and 3. $u_{coil,L}$ is the voltage over the coil inductor. $R_{path}$ and $u_{path}$ are the equivalent sum of a path’s components’ resistances and voltages respectively.

![Diagram of coil current path](image_url)

Figure 2.9: Schematic for the model of a coil current path.

According to (Alciatore & Histand, 2007) the voltage over an inductor is described by:

$$u_{coil,L} = \frac{d(L_{coil}i_{coil})}{dt}$$  \hspace{1cm} (2.2.9)

The derivation of the coil current follows as,

$$u_i = u_{path} + (R_{coil} + R_{path})i_c + \frac{d(L_{coil}i_{coil})}{dt}$$  \hspace{1cm} (2.2.10)

$$\frac{d(L_{coil}i_{coil})}{dt} = u_i - u_{path} - (R_{coil} + R_{path})i_{coil}$$  \hspace{1cm} (2.2.11)

$$i_{coil} = \frac{1}{L_{coil}} \int_0^t (u_i - u_{path} - (R_{coil} + R_{path})i_{coil}) d\tau$$  \hspace{1cm} (2.2.12)

Two ways to determine a path’s equivalent voltages and resistances are considered. The first makes the decision based on the high and low value of the PWM signal, and the release signal. That is, depending on the indicated path, the voltages and resistances are set to the paths values. The second way is to calculate the average voltages and resistances values based on the PWM duty cycle, and the release signal. That is, a path’s component’s voltage and resistance are weighted by the PWM duty cycle value, to correspond to how long it is active. The components are then summed together to get the average value.

$$x_{path} = dc \cdot x_{path_1} + (1 - dc)x_{path_2 or 3}$$  \hspace{1cm} (2.2.13)

where $dc$ is the duty cycle and $x$ corresponds to either the voltage or resistance for the indicated path.
2.2.6 Coil Current Monitor

The monitor circuit for the coil current, $i_{\text{coil}}$, consists of resistors and a capacitance. The time constant of this circuit is too low for the real-time simulations in this thesis and is instead simplified as a single resistor, $R_m$. This resistor converts the coil current into a voltage, $u_m$, and is simulated as:

$$u_m = R_m i_{\text{coil}} \quad (2.2.14)$$

In the simulation $u_m$ is calculated during the whole simulation run, but on a real ECC this signal is also turned on and off by the PWM transistor. Due to the dynamics of the electronics, this switching gives rise to step responses with transients that are too fast for the simulation to calculate. A sketch is shown in Figure 2.10.

![Figure 2.10: Sketch of coil current measuring transient, amplitude (y) vs. time (t).](image)

In a real application these transients can give rise to incorrect measurements of the coil current and therefore the microcontroller gets the wrong information. To capture this characteristic in the HIL setup, an analog solution could be used as described in Chapter 4.
2.3 Electromagnet

The electromagnet’s purpose is to transform electrical energy into kinetic energy that is used to move the anchor towards the fixed core. More details of the electromagnet modeling can be found in (Tjerngren, 2014).

The original idea was to use theory for magnetic circuits when modeling the electromagnet. However, it was decided during the planning phase that two lookup tables were going to be used instead. The reasons for this were:

- Faster calculations for the real-time simulation.
- Could reuse already existing lookup tables for the contactor AF370.
- Allowed for more time to model the electronics and mechanics.

2.3.1 Magnetic Circuit Approach

There are some problems with the magnetic circuit approach. One problem is that the reluctance, $\mathbf{R}$, of the electromagnet depends on the permeability. The contactors electromagnets have ferromagnetic cores and this means that the cores’ $B - H$ characteristics are non-linear. Because of this, the permeability of the cores is not constant and therefore a way to calculate it must be derived.

Other aspects to consider when modeling the electromagnet are how to handle the fringing that occurs with the magnetic field, especially in the air gaps, and the hysteresis of the magnetic material. Hysteresis is the dependence from previous states. Figure 2.11 illustrates fringing and hysteresis.

![Diagram of electromagnet and hysteresis curve](image)

**Figure 2.11:** Illustrations of fringing (left) and hysteresis (right).

2.3.2 Lookup Tables

ABB had produced two lookup tables for other projects. One describes the electromagnet’s coil’s inductance, and the other describes the magnetic force exerted on the anchor. The tables depend on the anchor position and the current through the electromagnet’s coil. They were made for the AF370 contactor by using the finite element method; see (Sadiku, 2011).

The tables were produced by using a 2D grid with the anchor position on one axis and the coil current on the other, and then calculating the values for the inductance and the force in each point. The calculations were made for a coil with one turn. To use the tables for a coil with an arbitrary number of turns $N$, the input current is multiplied by $N$ and the output inductance by $N^2$. 
The tables did not take any consideration to hysteresis (Sadiku, 2011); this was because the effect from the hysteresis is small for the magnetic material used in the contactors’ electromagnets. The information about the tables came from (Johansson, 2014). Surface plots of the two tables are shown in Figure 2.12 and Figure 2.13.

![Electromagnetic force surface plot](image)

**Figure 2.12: Lookup table for the electromagnetic force.**

![Coil inductance surface plot](image)

**Figure 2.13: Lookup table for the electromagnet’s inductance.**
2.4 Mechanics

The mechanics’ purpose is to perform the actual switching of the ExC. More details about the mechanic’s modeling can be found in (Tjerngren, 2014).

Springs and movable masses are used to describe the mechanics. Also a spring-damper component is used for impacts and the interaction between movable masses. This was by suggestion from ABB Corporate Research (Andersson & Fransson, 2014). A sketch illustrating the contactor’s mechanical system can be seen in Figure 2.14. The figure also points out the implemented models’ position references.

![Figure 2.14: Sketch of the contactor’s mechanics and reference placements.](image)

The different masses and springs, shown in Figure 2.2, in the system are:

- The electromagnet’s movable iron core, called the anchor.
- The electromagnet’s fixed iron core.
- The contact bridge, which connects all the movable masses.
- The movable contacts, which switches the ExC.
- The fixed contacts, part of the ExC.
- The release springs, which return the contact bridge to its initial position.
- The connection springs, which counteract the bouncing between the contacts as well as separates the contacts when switching off the contactor.

The components are set inside a contact housing made of a plastic material.
The forces affecting the mechanical system are:

- Spring forces (internal)
- Spring-damper forces (internal)
- Electromagnetic force (external)
- Gravity, depending on the mounting angle seen in Figure 2.15 (external)

![The contactor’s mounting angle](image)

**Figure 2.15: Illustration of the mounting angle used in this thesis.**

The force caused by the gravity may have a significant effect on the mechanics. This depends on the size of the contactor and the mounting angle, \( \alpha \). The contactors are usually mounted on a wall, and then the gravity does not affect the process as much. In some implementations the wall can move, for example on ships. For the AF370 contactor, considered in this thesis, the recommended mounting angle from the wall is \( \pm 30^\circ \) to ensure proper behavior of the contactor.

The modeling of the mechanics resulted in four different models. They are numbered 1 to 4 and the model complexity rises with the model number. Different models are developed, with the purpose of finding a suitable model for the real-time simulations.
3 Matlab and Simulink Implementation

In this chapter, the implementation of the contactor model is developed and tested in a non-real-time simulation environment.

The models derived in Chapter 2 are implemented in MATLAB R2012b and Simulink. Because the contactor system is separated into three subsystems, they are implemented as three Simulink subsystem blocks. The blocks are called the ECC, the electromagnet and the mechanics. A simulation control block is also added. Figure 3.1 illustrates the Simulink subsystems. The red labels are not used in the non-real-time simulations.

![Figure 3.1: Top level of the non-real-time Simulink implementation.](image)

The simulation control block is based on a simple approximation of a contactor’s microcontroller’s control signals during an on and off switching of the contactor. This means that it generates signals for the closing, the holding and the opening events.

The ECC parts, that are modeled, are implemented using two different approaches. One approach is using a PWM signal’s high or low state to activate different simulated electrical paths in the ECC model. The other approach is using the PWM signal’s duty cycle to calculate an average of the simulated electrical paths. The two approaches are implemented by using a so called variant subsystem block, which can be set to act according to one of the two approaches. In this thesis the approaches are called the PWM signal and the PWM duty cycle respectively.

The electromagnet is implemented in a subsystem block, and it is the link between the ECC and the mechanics.

The four mechanics models are also implemented by using a variant subsystem block. This block can be set to act according to one of the four different mechanics models. The models are numbered from 1 to 4 and the model complexity increases with the model number.

All subsystem blocks are implemented by using standard Simulink blocks.
3.1 Subsystem Blocks
In this section the simulation control and the ECC blocks are described in more detail. The electromagnet and the mechanics are briefly summarized, for more details see (Tjerngren, 2014).

3.1.1 Simulation Control
The purpose of this block is to generate a test sequence for the contactor model when evaluating its behavior. It corresponds to a simple substitute of the hardware part of the ECC. It generates signals which represent the microcontroller’s control signals, and they are sent to the simulated parts of the ECC. The output signals are:

- **Supply On**, signals that the input voltage is on and that the ECC is powered. If the contactor is controlled by the input voltage, this also indicates that the contactor should be closed. If the PLC control is instead used, this signal only indicates that the ECC is powered up.

- **On_N**, not used in the non-real-time simulations.

- **OFF**, not used in the non-real-time simulations.

- **PWM signal**, this signal represents a PWM signal which is only used in one of the two ECC approaches.

- **PWM duty cycle**, this signal represents a PWM’s duty cycle which is only used in one of the two ECC approaches.

- **Release signal**, this signal indicates that the electromagnet’s coil is being demagnetized and that the contactor should open.

- **Simulation time**, this signal represents the time of the simulation.

The sequence representing the closing, the holding and the opening of the contactor model is implemented by using timed switch blocks to change the output signals.

3.1.2 ECC
The ECC block consists of the modeled parts of a real ECC, and its purpose is to link the control signals from the simulation control block to the rest of the contactor model. Figure 3.2 shows the PWM duty cycle approach, the PWM signal approach looks similar. The Simulink blocks represent the implemented functions and equations derived in Chapter 2.

The implementation of the Ebers-Moll transistor was done by using the Simulink Simscape library. After testing the implementation it was decided to not be further considered here. This was because its computations were too time consuming to be used in our real-time simulations.
Figure 3.2: ECC Simulink implementation.

The input signals come from the simulation control block and the electromagnet block. In the non-real-time simulations only the coil current output is used, while the voltage monitor and the coil current monitor blocks’ output is only needed in the real-time simulations. The Simulink implementations correspond to the gray boxes in Figure 2.6, and the blocks’ description are:

- **Coil current circuit parameters**, calculates the equivalent voltages and resistances used in the coil current circuit, except for the coil resistance.
- **Coil resistance**, calculates the coil resistance depending on its temperature which is constant during a simulation.
- **Input voltage**, calculates the input voltage falling over the coil and the voltage monitor circuit.
- **Coil current circuit**, implements equation (2.2.12) to calculate the coil current.
- **Voltage monitor**, implements equation (2.2.7) to calculate the monitored voltage.
- **Coil current monitor**, implements equation (2.2.14) to calculate the monitored coil current.

The coil current circuit parameters and the input voltage MATLAB functions can be found in Appendix B – MATLAB Scripts.

### 3.1.3 Electromagnet

The purpose of the electromagnet block is to act as the link between the ECC and the mechanics blocks. This is done via LUTs, calculating the coil inductance and the magnetic force exerted on the anchor. The block uses two inputs, the coil current calculated in the ECC, and the anchor position calculated in the mechanics block. The coil inductance is sent to the ECC block and the electromagnetic force is sent to the mechanics block.

### 3.1.4 Mechanics

The purpose of the mechanics block is to calculate the dynamic behavior of the contactor. Its input is the electromagnetic force calculated in the electromagnet, and its output is the anchor position, which is sent to the electromagnet.
### 3.2 Parameter Values

ABB provided values for most of the parameters, for example resistor values, number of turns of the electromagnet’s coil, the distances inside the contactor, weights for the masses and spring constants. The coefficients of restitution and the spring-damper values were tuned to give similar impact results as real measurements. The measurements used for this was received during the meeting with Corporate Research (Andersson & Fransson, 2014).

All parameters are initialized by running a MATLAB script before starting the simulations. The MATLAB scripts and the parameter values for the non-real-time simulations are similar to those used later during the real-time simulations; these can be found in Appendix B – MATLAB Scripts.

### 3.3 Non-real-time Simulations and Validation

In this section, plots showing simulation results from the contactor model are presented. The plots illustrate closing, holding and opening phases of the contactor. Measurements, received from Corporate Research (Andersson & Fransson, 2014), of the coil current and the contact bridge position are used for comparisons. At the end of this section the complete simulation sequence is explained.

#### 3.3.1 ECC

Figure 3.3 and Figure 3.4 show simulations of the coil current using the two PWM approaches, described in the introduction to this chapter. The second figure shows that the PWM signal approach gives a more jagged appearance compared to the PWM duty cycle approach.

There are two large disadvantages of the PWM signal approach. The first is that the simulation time step could not be small enough, especially in the real-time simulations, to represent the PWM frequency range used in the real contactor system. The second is that it takes much longer time to run the same simulation setup, compared to the PWM duty cycle approach. Due to these disadvantages the PWM duty cycle approach is used for the rest of this thesis.

![ECC block: Coil current simulations](image)

**Figure 3.3:** Simulation and comparison with measurements for the ECC PWM approaches.
3.3.2 Electromagnet
The force and the inductance outputs from the electromagnet block can be seen in Figure 3.5. The two plots in the figure are the results from the cubic interpolated LUTs.

Figure 3.4: Closer view of the PWM approaches.

Figure 3.5: Simulation of the electromagnet’s lookup tables.
3.3.3 Mechanics

Figure 3.6 shows the spatial simulation result for model 2 of the mechanics. Results from the other models are presented in (Tjerngren, 2014).

At the beginning the contactor is in its open position. At around 0.1 seconds the contactor closes, and the contact bridge moves accordingly. At around 0.12 seconds the holding phase starts and it continues until the contactor opens.

![Mechanics block: Model 2](image)

**Figure 3.6:** Simulation and comparison with measurements for model 2 of the mechanics.

3.3.4 Explanation of the Simulations

Figure 3.3, Figure 3.5 and Figure 3.6 show the whole contactor behavior. These figures, together with the following equation (Alciatore & Histand, 2007) is used to explain the behavior,

\[
V(t) = \frac{d\Phi}{dt} = \Phi = L \frac{di}{dt} = \frac{dL}{dt} + L \frac{di}{dt}
\]  

(3.3.1)

The equation describes the relation between the voltage over an ideal inductor, \( V(t) \), and the total magnetic flux, \( \Phi \), through it. The flux is related to the inductance, \( L \), and the inductor current, \( i \).

In this thesis the electromagnet’s coil can be seen as an ideal coil, which corresponds to the inductor, in series with a resistor. The current is always positive due to the design of the ECC. The inductance is positive since the coil current and the flux changes sign at the same time.

The input voltage is the source for the coil voltage, which is regulated by the ECC. The regulated voltage is divided between the coil and the resistance. The voltage over the resistor is linearly dependent on the coil current, according to Ohm’s law (Alciatore & Histand, 2007). That is, the larger the coil current is, the larger the voltage over the resistor becomes.
For the explanation, three phases are defined from Figure 3.3. The closing phase occurs between 0.1 and 0.2 seconds, the holding phase between 0.2 and 0.56 seconds and the opening phase between 0.56 and 0.75 seconds.

**The Closing Phase**
The phase begins when an input voltage is applied to the ECC. At first, no coil current exists and all voltage falls over the coil. This causes the current to start flowing through the coil, and a magnetic force begins to affect the anchor. When the force is large enough to overcome the release springs the anchor moves towards the fixed core.

When the air gap is close to zero, the force increases fast. This is caused by the corresponding decrease of the magnetic reluctance in the electromagnet, which also means that the flux increases. The inductance also increases, which must mean that the flux increases faster than the coil current. The changes in the force and the inductance can be seen in Figure 3.5 around 0.125 seconds.

The first large decrease in the coil current, in response to the mentioned change in the flux, can be explained by equation (3.3.1). The equation indicates that if the inductance increases greatly, caused by the fast increase in the flux, it has to be compensated by a decrease in the current. This can also be explained by that the coil voltage has to be large, in response to the change in the flux, which means that more of the input voltage falls over the coil. This happens when the coil current, which flows through the resistance, decreases.

At around 0.13 seconds the force stops to increase because the electromagnet becomes saturated. This means that the flux stabilizes at some value, and in turn the coil voltage stabilizes around zero according to equation (3.3.1). The coil current increases until it is limited by the resistance. This also shows up in the inductance plot as a decrease. At around 0.15 seconds the contactor enters an equilibrium, which continues until the holding phase begins.

**The Holding Phase**
The main reason for the holding phase is to reduce power consumption. It starts at 0.2 seconds when the ECC regulates down the voltage over the electromagnet’s coil. This causes the current to decrease, until it reaches a new equilibrium. In turn, the force decreases due to the decrease in the flux. The inductance increases since the current decreases more in proportion to the magnetic flux.

**The Opening Phase**
The opening phase starts at 0.56 seconds, and the ECC disconnects the input voltage from the electromagnet’s coil. The freewheeling circuit, see Figure 2.6, is activated first. When activating the circuit, a negative voltage arises over the ideal coil. The freewheeling circuit consumes energy and causes the coil current to decrease. Equation (3.3.1) then implies that the magnetic flux decreases, which also means that the force decreases. The increase in the inductance implies that the current decreases faster than the flux. At 0.6 seconds, the demagnetizing circuit is activated instead of the freewheeling circuit. This circuit consumes the remaining energy quickly and makes the magnetic flux decrease very fast.

The anchor begins to move upwards when the springs overcome the electromagnetic force. In turn, the inductance also decreases at a rate which makes the current increase for a short duration to keep the momentary coil voltage, according to equation (3.3.1). When the inductance stops to decrease, the current can decrease to zero. The small change in the inductance, at the end of the phase, only shows up in models implementing impacts.
4 Hardware-in-the-loop System Setup

The Simulink contactor model is integrated with the HIL-setup. Figure 4.1 shows a block diagram illustrating the various components in the setup. The first block, from the left, represents a host PC running a program with a graphical user interface which controls the real-time simulations. The second and third blocks represent the simulator hardware. The fourth block represents hardware that is constructed for this thesis. It is needed for transmitting the physical signals to and from the ECC, which is represented by the last block. The ECC was modified to exclude the parts that are simulated.

![Block diagram over the HIL setup.](image)

4.1 dSPACE Simulator

The real-time simulator, from the company dSPACE (dSPACE, 2014), consists of the following hardware and software components (dSPACE GmbH, 2012):

- DS1005 Processor board.
  - PowerPC 750GX, 1 GHz.
- DS2202 HIL I/O board.
- dSPACE Simulink RTI library.
- ControlDesk 3.7.4.

ControlDesk is a software program, running on a host PC, that is used to control the real-time simulations as well as visualizing the simulation results via a graphical user interface (GUI). The DS1005 board provides the computational power needed for the real-time simulations; it also serves as the interface to the host PC and the DS2202 HIL I/O board. The I/O board both generates and measures the required signals that is sent to and received from the constructed hardware. The Simulink RTI library is used to link the Simulink contactor model to the physical channels of the I/O board. When the Simulink model is ready for HIL simulations, C-code is generated from the model via MATLAB Coder and Simulink Coder. The code is also compiled and linked into a real-time application, which is then downloaded to the DS1005 board.

Figure 4.2 shows the hardware setup for the real-time simulations.
4.2 Transmitted Signals

Before constructing the hardware interface, between the DS2202 HIL I/O board and the modified ECC, the necessary signals to send and receive had to be defined. In Figure 4.3 these signals are named and they mainly correspond to the signals to and from the microcontroller in Figure 2.6.
In the Figure 4.3, the signals from the I/O block represents the contactor’s control signals (ON_N, OFF and Supply_ON) and the simulated monitoring circuits’ output (I_MON and V_MON). The I/O board also provides a constant power supply (Supply) and reference ground (GND) for the constructed hardware interface and the modified ECC. The ECC card receives these signals, and outputs its responses (PWM and Release), which are fed back to the simulator. The signals are described as:

- ON_N, one of the PLC signals.
- OFF, one of the PLC signals.
- I_MON, the coil current monitor signal.
- V_MON the input voltage monitor signal.
- Supply_ON, a signal used to turn the ECC supply voltage on and off.
- Supply, constant +12 V power supply.
- GND, the ground reference.
- PWM, the microcontroller’s signal which controls the PWM MOSFET.
- Release, the microcontroller’s signal which controls the release MOSFET.

These signals are assigned, according to their function, to their own pins on the DS2202 I/O board’s D-sub connectors, and the pin configuration is found in (dSPACE GmbH, 2012). The resulting pin assignments are further explained in Appendix C – DS2202 HIL I/O Board Pin Configuration.

### 4.3 Constructed Hardware Interface

The HIL system is mounted in a standard 19-inch rack. The modified ECC to be used in the real-time simulations is therefore installed inside a 19-inch box. The box is equipped with two D-sub socket connectors, as its input interface to the I/O board’s D-Sub plug connectors. Inside the box, the D-sub socket pins are wired to an intermediate electronics prototyping board. On the board the signals are processed before they are transmitted to the modified ECC. In Figure 4.4 the inside of the box is shown and Table 4.1 shows the mapping between the wire coloring and the transmitted signals. The signals are named as in Figure 4.3.

![Figure 4.4: Box containing the constructed hardware interface.](image-url)
### Table 4.1: Mapping between signal name and cable color.

<table>
<thead>
<tr>
<th>Signal name</th>
<th>Physical cable color</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWM</td>
<td>Green</td>
</tr>
<tr>
<td>Release</td>
<td>Blue</td>
</tr>
<tr>
<td>ON_N</td>
<td>Orange</td>
</tr>
<tr>
<td>OFF</td>
<td>Orange/black</td>
</tr>
<tr>
<td>I_MON</td>
<td>Red/black</td>
</tr>
<tr>
<td>V_MON</td>
<td>Blue/black</td>
</tr>
<tr>
<td>Supply_ON</td>
<td>Green/black</td>
</tr>
<tr>
<td>Supply</td>
<td>Red</td>
</tr>
<tr>
<td>GND</td>
<td>Black</td>
</tr>
</tbody>
</table>

#### 4.3.1 Electronics Prototyping Board

The main purpose of the prototyping board is to transmit the signals between the DS2202 I/O board and the modified ECC. It is also used to add a switching circuit to enable to turn the ECC’s access to the constant power supply on and off.

**Switching Circuit**

On a unmodified ECC the input voltage is regulated down to 11.4 V by a circuit consisting of resistors and transistors. This circuit is not kept as hardware since the input voltage was simulated and the I/O board could not output the required voltages. Instead the real-time simulator supplied a 12 V DC output which is used. This voltage is constant and cannot be turned on and off, which is necessary when the input voltage is used to control the ECC. Therefore a switching circuit, controlled by the simulator, is used to enable to turn on and off the ECC’s 11.4 V node access to the power supply.

The schematic of the switching circuit can be seen in Figure 4.5. The control signal, Supply_ON, is generated by the simulator. The resistors are used to restrict the current drawn from the outputs of the I/O board. When Supply_ON goes high the NMOS transistor begins conducting. In turn this lowers the PMOS gate voltage, and the PMOS transistor begins conducting current. Internally, in the modified ECC, capacitors will be charged until the PMOS source-drain voltage becomes too low to conduct current.

![Figure 4.5: Schematic for the controlled switch circuit of the power supply.](image-url)
**ECC Capacitors Release Circuit**

The modified ECC’s 11.4 V node, in Figure 4.5, is connected to a pair of capacitors on the ECC. When switching off the supply voltage these capacitors have to be emptied of their stored energy. On an unmodified ECC this is done by the PWM switch circuit, seen in Figure 2.6, but on the modified ECC a transistor connects these capacitors to ground when Supply_ON goes low.

**Filter Circuit**

As mentioned in Chapter 2, for the coil current monitor circuit, the ECC’s microcontroller monitors the coil current by measuring a voltage over a resistor. The current trough this resistor is turned on and off by the PWM signal which controls a transistor switching circuit. This switching gives rise to step responses with transients that are too fast to simulate. To emulate this in the HIL setup an analog circuit can be used, however a filter circuit was not implemented in this master thesis.

Figure 4.6 shows a conceptual circuit that implements a step response to the steady state I_MON value. Depending on the filter, low-pass in the figure, it can introduce different transients on the ECC’s monitored I_MON voltage. The switch is controlled by the ECC’s PWM signal to get the correct frequency and duty cycle. The current of the I_MON source is restricted by the DS2202 I/O board and to increase its capability a voltage follower, the operational amplifier, is used. The operational amplifier is charging a capacitor to the same voltage as I_MON, and acts to stabilize this voltage level during the step responses.

![Figure 4.6: Schematic for a possible filter circuit concept.](image-url)
**Sketch and Figure of the Board**

A sketch of the prototyping board can be seen in Figure 4.7. The DS2202 I/O board’s wires are connected to the right and the modified ECC is connected to the left. The topmost transistor and resistor circuits are the implementation of the capacitor release and the switching circuits. The bar noted signals are ground sensing pins for the I/O board’s respective signals. The bottommost resistors are connected to the I/O board’s output pins and their purpose is to restrict the drawn current according to (dSPACE GmbH, 2012).

![Diagram of the prototyping board]

**Figure 4.7: Sketch of electronics prototyping board.**

A picture of the electronics prototyping board can be seen in Figure 4.8.
4.4 Modified ECC
A real contactor’s ECC was modified by removing circuits which are either simulated or not needed in the HIL simulator setup. The microcontroller is the core component of the hardware part of the HIL setup. The white blocks in Figure 2.6 represents the circuits that are needed for the microcontroller to run, mainly the 3.3 V power supply and the reset circuit. In addition to these, the circuits that are necessary when programming the microcontroller are also left intact.

4.4.1 Power Supply
The DS2202 I/O board’s 12 V power supply is regulated down to 3.3 V by hardware on the ECC; this voltage level is then used as the power supply for the microcontroller and the reset circuit.

4.4.2 Microcontroller
The microcontroller regulates the coil current by changing the PWM signal’s duty cycle and frequency, and the Release signal. These signals are measured by the I/O board and in turn affect the simulated contactor model. The microcontroller measures ON_N, OFF, V_MON and I_MON, generated by the I/O board, to adjust the PWM and Release signals.

4.4.3 Reset Circuit
The reset of the microcontroller occurs when the input voltage is removed, that is, when the power supply switching circuit is turned off for the HIL setup. The reset’s function is to ensure that the microcontroller’s states have been initialized correctly before the next contactor switching operation.

4.5 Extended Simulink Model
The Simulink model in Figure 3.1 is extended by modifying the simulation control block to only generate signals to the modified ECC’s microcontroller. A block containing the Simulink RTI blocks is also added, the blocks connects the model to the DS2202 HIL I/O board’s physical channels. For more details see (Tjerngren, 2014).
5 Real-time Simulations and Validation

The real-time simulations are performed using the HIL setup described in Chapter 4.

5.1 Real-time Simulations

The real-time application, generated from the Simulink contactor model, is checked during the building and downloading process. This is to determine if the application is suitable to run in real-time.

During the non-real-time simulations a time step of 50 µs was used. This time step did not work for the real-time simulations. The time step had to be changed to around 300 µs before it was accepted by the real-time simulator. This caused problems for model 3 and 4 of the mechanics; they became unstable when using a time step larger than 180 µs.

The Simulink model was investigated for possible improvements, with the purpose to reduce the required time step. The main bottleneck, that was found, existed in the electromagnet subsystem. The LUTs used cubic interpolation and extrapolation algorithms to calculate their outputs. This could be changed to linear algorithms instead; the real-time simulator could then accept a time step of 50 µs.

A problem with the LUTs, when using the linear algorithms, was that they were too coarse. This especially showed up in the coil current simulations, which in some regions diverged significantly from the results in Chapter 3.

The differences between the linear and cubic algorithms’ outputs were reduced by calculating tables with a finer mesh, compared to the original tables in Figure 2.12 and Figure 2.13. This was done by sampling the original LUTs, using the cubic algorithms, in Simulink during non-real-time simulations. Surface plots of the new tables are shown in Figure 5.1 and Figure 5.2.

![Figure 5.1 Lookup table for the electromagnetic force (finer mesh).](image-url)
Figure 5.2 Lookup table for the electromagnet’s inductance (finer mesh).

5.1.1 ControlDesk
The ControlDesk program is used for interacting with the real-time application during runtime. This is done via a GUI, which is illustrated in Figure 5.3.
A layout can be created in the program, and instruments can be added to it. A few of the instruments are sliders, buttons and graphs. Some of the application’s variables can be connected to these instruments. The variables correspond to output values from blocks in the Simulink model.

For example, the value output from a constant block can be connected to sliders and buttons, which then can be changed to affect the simulation. Thereby, among other parameters, the mounting angle, spring constants or resistor values can be altered manually during the real-time simulations to create new test cases. The graphs can be used to plot I/O board measurements or simulation results, for example the PWM signal or the coil current respectively.

A simulation control block in the Simulink implementation is constructed so that the contactor model can be closed and opened manually, or to repeat a sequence of closing and opening. This can be set via the ControlDesk using the method described in the previous paragraph.

5.2 Measurements of a Real Contacto

The measurements, from Corporate Research (Andersson & Fransson, 2014), used in the validation of the non-real-time simulations, were collected from contactors using an older version of the ECC’s software. The modified ECC, used in the HIL setup, is programmed with newer software. Because of this, new measurements were collected for the validation of the real-time simulations.

The measurements were collected in a test lab by mounting a contactor on a wall. The measurement equipment included an oscilloscope, a laser and a current probe. The lasers were used for position measurements. The measurement setup is shown in Figure 5.4.

![Measurement setup](image)

**Figure 5.4: Measurement setup.**

43
5.3 Validation
In this section, plots showing the HIL simulation results are presented. The plots illustrate closing, holding and opening phases of the contactor model. The new measurements of the coil current and the contact bridge position are used for comparisons.

5.3.1 PWM Measurement Problems
The DS2202 HIL I/O board measures the PWM signal generated by the modified ECC. The ECC changes the PWM signal’s frequency and duty cycle depending on the contactor’s state. The changes make it hard for the I/O board, due to hardware limitations, to correctly capture the PWM signal.

For example, when the contactor enters the holding phase the PWM signal stabilizes. The duty cycle and the frequency becomes too short and too high, respectively, in relation to each other for the measurements. This results in that the I/O board’s calculation of the duty cycle becomes a constant value for a fixed time period, until it drops to zero even though the PWM signal is still active. This time period depends on the RTI settings used for the RTI I/O block.

The constant value changes between runs depending on how it was measured before the PWM signal becomes too ill conditioned for the I/O board. In this report, when the value results in a simulated coil current which corresponds well to the coil current measurements it is called good, else it is called bad.

5.3.2 ECC
Comparisons between the measured coil current and the simulated coil current, when the measurements of the PWM signal is good respectively bad during the holding phase, are shown in Figure 5.5 and Figure 5.6. The difference is mainly after 0.15 seconds, during the holding phase, due to the problems with the measured PWM signal. Figure 5.7 shows a closer look of the closing phase.

Figure 5.5: HIL simulation and comparison with measurements, for a good measured PWM signal.
Figure 5.6: HIL simulation and comparison with measurements, for a bad measured PWM signal.

Figure 5.7: Closer look of closing phase, for a good measured PWM signal.
5.3.3 Electromagnet
HIL simulations of the coil inductance and the electromagnetic force are shown in Figure 5.8. Compared to the results in Figure 3.5 the main difference is during the initial rise. This is due to the difference in coil current, mainly caused by the PWM measurement, seen in Figure 5.7.

![Electromagnet block: LUT output during simulations](image)

Figure 5.8: HIL simulation of the electromagnet’s lookup tables, for a good measured PWM signal.

5.3.4 Mechanics
The comparison between the measurements and the real-time simulations of model 2 is shown in Figure 5.9. Results from the other models are presented in (Tjerngren, 2014).

![Mechanics block: Model 2](image)

Figure 5.9: HIL simulation and comparison with measurements, for model 2 of the mechanics (good PWM signal).
Figure 5.10 shows comparisons between simulations of the mechanics, using model 2, when the PWM signal measurement is good respectively bad. Figure 5.11 and Figure 5.12 illustrate the closing and opening phases. The effect on the mechanics, due to the bad PWM signal, is a delay in the opening phase.

![Mechanics block: Model 2]

**Figure 5.10:** HIL simulations of model 2, for two cases of the PWM signal measurement.

![Mechanics block: Model 2]

**Figure 5.11:** Closing phase, for two cases of the PWM signal measurement.
Figure 5.12: Opening phase, for two cases of the PWM signal measurement.
6 Discussion

The result of this master thesis was successful; we have fulfilled all the purposes stated in Chapter 1. We investigated the contactor system, and defined the boundary between the hardware and the software parts used in the HIL setup. We then implemented the three subsystems in MATLAB and Simulink, and integrated them with the dSPACE RTI Simulink library. We also modified an existing ECC and constructed the necessary hardware interface, needed to connect the modified ECC to the dSPACE simulator. Finally, we performed real-time simulations and validated the results by comparing them to measurements.

6.1 Benefits of the HIL Concept

We believe that the HIL setup concept of using computer simulated models in conjunction with real hardware has great potential. A system’s behavior can be tested in well-defined circumstances and can reduce the development time. For example the simulation parameters can easily be set to desired values, to represent several different variants of the same system. Compared to testing a real system, the HIL concept can save time when changing between different setups as well as reproduce test conditions. The conclusion that the HIL concept can be effective concurs with the literature study in Chapter 1.

The HIL concept gives the possibility to run several more tests than is feasible with a real system setup. These tests can be run in a short time period, and can hopefully help to identify problems and improve the system. To help a user, test cases can be generated, run and analyzed automatically while the user performs other tasks. The improvements can, for example, concern power consumption, mechanical design or reliability issues. Another aspect is that the tests can help with gathering knowledge about a system. This can, for example, be used to improve the tests of a real system and lead to sustainable development.

If the power consumption can be lowered for a system, it is beneficial for both the environment and the operating cost of using the system. Improved reliability reduces the probability of failure. Depending on the application, a system failure can have a significant impact. For example if a system malfunctions, and part of a factory has to be suspended for repairs, it can cost a significant amount of money due to delays.

Even though the HIL setup is useful in many ways, it cannot completely replace tests of real systems, for example stress tests and final evaluations of a system. In addition, the models cannot be too complex since the simulations have to be run in real-time. One has to bear in mind that the simplified models make the HIL simulations unable to capture all of a system’s behavior. This agrees with the statement in the literature study that accurate models are needed to increase the credibility of HIL setups.

6.2 Model Simplifications

The models used in this thesis are quite simple, mainly to enable the real-time simulations to run with as small time steps as possible.

For the ECC model this has mainly resulted in neglecting the transistor’s transient behavior. This can also be motivated by considering the PWM signal’s frequency, which the transistors react to. The frequency is in the range of 20 kHz which means that the time period is 50 μs. For example, when the duty cycle is 10% the high period is active for 5 μs. When comparing this to the real-time simulation’s time step, which we got down to 50 μs, the transient behavior is too fast to simulate. It is a drawback that the electronics cannot be thoroughly simulated, but when looking at the comparisons between the
measured and the simulated coil current values in Figure 5.5 the simplifications result in acceptable simulations. Figure 5.7 shows that there is a difference in the beginning of the signal; we believe this is mainly due to problems with the HIL I/O board’s measurement of the PWM signal.

The method of using LUTs for the electromagnet works fine, however new tables have to be calculated for other contactor types. When using the extended LUTs, the linear interpolation algorithm yields good results and low computation time.

The mechanical behavior of the contactor is relatively simple, and the validation indicates that the simplifications of the mechanics are reasonable. Model 2 of the mechanics gives a good tradeoff between simulation results and computation time, compared to model 3 and 4. The spring-damper components’ parameters were tuned to correspond to the measurements, no analysis of the actual materials was performed. One improvement of the mechanics could be to model the friction between the different masses.

6.3 Validations

Only a basic validation of the contactor model’s behavior was performed, by assessing figures showing differences between measurements and simulations. We have only used a few measurement series in the comparisons. To get more reliable validations, a statistical approach should be used with several measurement series.

The setup we used when collecting the measurements, for the validations, was not optimal. One issue was that we could not measure the position of all the interesting movable parts. For example the movable contacts’ position could not be measured without altering the contactor. Another issue was that we did not have a program controlling the contactor’s on and off signal, during the measurements. Therefore we did not have exact timestamps for the changes of the control signal. In extension this means that a real contactor’s signal’s time delays, in response to the closing and opening, are not captured in the validation.

6.4 Future Improvements

We have considered improvements for our HIL setup and the most significant of them are described here.

The constructed hardware interface, in the HIL setup, can be improved by designing a specific printed circuit board instead of using the modified ECC and prototyping card. This will give better solder joints and reduce the physical size of the contactor’s HIL setup. The proposed filter for I_MON or another solution should be implemented to give a HIL setup that is more realistic. The ECC computer model may be implemented by using the Simscape library in Simulink. Simscape can be used to simulate electronic circuits for a more accurate description of the ECC.

We had problems with the HIL I/O board’s measurement of the PWM signal, especially when the duty cycle was low and the frequency high. These problems are explained in Chapter 5. By using an oscilloscope, we determined that the duration of the PWM’s high period was around 1.5 μs; when the simulated contactor had entered its holding phase. According to the specifications of the I/O board’s PWM measurement channels, the board cannot handle PWM signals with a high or a low time period which is lower than 3 μs. This problem might be solved by using hardware dedicated to measuring PWM signals.
7 References

Available at: http://new.abb.com/about/our-businesses/low-voltage-products-division


Coman, O., 2014. Introduction to the ECC. Västerås: ABB.


Available at: http://www.dspace.com/en/pub/home/applicationfields/electric_drives/hil-simulation.cfm

Available at: https://www.dspace.com/en/inc/home/products/systems/ecutest.cfm


Appendix A – Work Process

The work for this master thesis was mainly performed at ABB Control Products in Västerås, Sweden, by me and Jon Tjerngren. Since we were two students working on the same master thesis we had to clearly show each own's contribution. Therefore we began the thesis by making a detailed plan for the work where main responsibilities were assigned; the planning resulted in six phases. Within each phase the work was separated according to our responsibilities and could mostly be performed in parallel. At the end of a phase the parallel work converged and the separate parts were put together and verified. This was to measure the progress of the work and to ensure that we were ready for the next phase. Even though the work was made in parallel we still helped each other when a problem occurred.

The phases and their descriptions were:

- Planning, plan the thesis work.
- Study and Modeling, studying the contactor concept and define models and create them in Simulink.
- HIL Implementation, integrating the models with the HIL setup.
- Simulation and validation, performing real-time simulations and validations.
- Reserve time, extra time.
- End phase, finalizing the thesis work.

Phases

To get a good structure of and to be able to easily measure the progress of the work we planned the work into several phases. A flowchart illustrating the different phases is shown in Figure A.1.

Phase 1 – Planning

This phase was used to plan the master thesis. To be able to make a well-considered plan we first made a short study of the contactor concept. This were done together to get a common understanding, of the problem and goal, for this master thesis. This resulted in the mentioned flowchart of the estimated work distribution and timeframe.

The short study included reading an article (Riba Ruiz, et al., 2010) describing the contactor modeling concept, to dismantle a real contactor to see its mechanical construction and attending two short lectures given by ABB. The first lecture covered the basic contactor concept and problems that arise when a switch is closed or opened where large currents can occur (Johansson & Karlen, 2014). The second lecture went into details about the ECC (Coman, 2014).

We also defined the necessary I/O signals between the various subsystems of the contactor model.
Figure A.1: Flowchart illustrating the estimated work distribution and timeframe.
Phase 2 – Study and Modeling
During this phase we worked separately on our own responsibilities.

I made a more thorough study of the ECC to better understand the various parts and their functions. I read about the dSPACE system to find out what types of I/O channels were available. I considered what would be necessary to build for the hardware interface and where the boundary between the ECC and Simulink models should be for the HIL setup. From there I defined the necessary I/O signals between the ECC and the simulator. The modeling of the ECC went through an iteration process to find two possible models for the HIL setup.

Jon made a more thorough study of the mechanical behavior of the contactor and the electromagnet. He also read about the RTI provided by dSPACE, which was used in the HIL implementation. His studies resulted in four different models, of various levels of complexity, describing the mechanical behavior. The investigation of the electromagnet resulted in the use of LUTs to describe its behavior.

We made several different models of the subsystems with the purpose for investigating the appropriate set to be implemented in the HIL-setup. When the separate tasks were finished and implemented in Simulink we put these together to represent a complete contactor model in Simulink. To be able to test the behavior of a complete model, I made a simple simulation of the microcontroller’s output signals in Simulink. We made the evaluation of the Simulink simulations results together.

Phase 3 – HIL Implementation
During this phase I modified a real ECC to remove parts that instead would be simulated. I also designed and built the necessary hardware interface between the dSPACE simulator and the modified ECC. The hardware interface was installed in a 19-inch rack with two D-Sub connectors. To send signals between the modified ECC and the simulated parts of the contactor system these connectors was connected to the dSPACE HIL I/O board.

Jon added Simulink blocks, from the dSPACE RTI library, to the Simulink models of the contactor. This was done to connect the model to the physical signals of the I/O board.

To be able to download our Simulink models to the dSPACE simulator we used MATLAB Coder and Simulink Coder to generate C-code. The code was also compiled into a real-time application which could be downloaded to the simulator. Jon used the program ControlDesk’s GUI to show the application state during the real-time simulations. I performed measurements on the constructed hardware interface to check that the correct signals were sent and received. We made some adjustments of the Simulink model and downloaded it again before we were ready for the next phase.

Phase 4 – Simulation and Validation
When the integration of the Simulink model with the HIL was complete, we made real-time simulations of the contactor model. We got help from ABB to set up a test of a real contactor and to collect measurements for the validation step. The verification of the HIL system was made by comparing simulated results against measured data.

Phase 5 – Reserve
Two week was planned as a buffer against unexpected problems and time delays.

Phase 6 – End phase
The end phase was used to oppose on another master thesis, finalize this report and prepare for my presentation.
Appendix B – MATLAB Scripts

In this appendix, complementary MATLAB scripts used during the simulations are presented.

ECC Simulink Implementation

There are two MATLAB function blocks used for the ECC, furthermore there are two versions of each block depending on the PWM approach. The two blocks are the coil current circuit parameters block and the input voltage block, their MATLAB scripts are shown in Listing B.1 - Listing B.4.

```matlab
function [R, V] = Parameters(On, PWM_DutyCycle, Release, ...
    Threshold_1, Threshold_2, R201, RM1, RM2, RD2, RD3, VM1, VM2, VD2, ...
    VD3, CoilCurrent)
    %#codegen
    % Parameters calculates resistances and voltages for component equivalents in series with the coil, which values depends on the calculated average by using the PWM duty cycle.
    %
    % On - Determines if the supply is on or off
    % PWM_DutyCycle - PWM duty cycle signal in percent
    % Release - Coil current release signal
    % Threshold_1 - Supply control threshold value
    % Threshold_2 - Microcontroller threshold value
    % R201 - Current sensing resistor
    % RM1 - Transistor M1 resistance
    % RM2 - Transistor M2 resistance
    % RD2 - Diode D2 resistance
    % RD3 - Zener diode D3 resistance
    % VM1 - Transistor M1 voltage
    % VM2 - Transistor M2 voltage
    % VD2 - Diode D2 voltage
    % VD3 - Zener diode D3 voltage
    % CoilCurrent - Current through the coil
    %
    % R - Output resistance
    % V - Output voltage
    
    % Duty cycle in decimal form
    DutyCycle = PWM_DutyCycle;
    
    % When supply is active and Release is high current goes through M1 and R201 for DutyCycle of the time and through M2 and D2 the rest of the time.
    if On > Threshold_1 && Release > Threshold_2
        R = DutyCycle*(RM1 + R201) + (1 - DutyCycle)*(RM2 + RD2);
        V = DutyCycle*VM1 + (1 - DutyCycle)*(VM2 + VD2);
    % When supply is inactive and Release is high current goes through M2 and D2
    elseif Release > Threshold_2
        R = RM2 + RD2;
        V = VM2 + VD2;
    % When Release is low current goes through D3 and D2
    else
        R = RD3 + RD2;
        V = VD3 + VD2;
    end
    
    % When no current goes through the coil the diode equivalent components is off
    if CoilCurrent == 0
        R = 1e8;
        V = 0;
    end
end
```

Listing B.1: Coil current circuit parameters function for the PWM duty cycle approach.
function [R, V] = Parameters(On, PWM, Release, Threshold, ... 
    R201, RM1, RM2, RD2, RD3, VM1, VM2, VD2, VD3, CoilCurrent)
%#codegen
% Parameters calculates resistances and voltages for component
% equivalents in series with the coil, which values depends on the PWM and
% release signals.
% On - Determines if the supply is on or off
% PWM - PWM signal
% Release - Coil current release signal
% Threshold - Logical threshold value
% R201 - Current sensing resistor
% RM1 - Transistor M1 resistance
% RM2 - Transistor M2 resistance
% RD2 - Diode D2 resistance
% RD3 - Zener diode D3 resistance
% VM1 - Transistor M1 voltage
% VM2 - Transistor M2 voltage
% VD2 - Diode D2 voltage
% VD3 - Zener diode D3 voltage
% CoilCurrent - Current trough the coil
% R - Output resistance
% V - Output voltage

% When supply is active and PWM is high current goes through M1 and
% R201
if On > Threshold && PWM > Threshold
    R = RM1 + R201;
    V = VM1;
% When PWM is low and Release is high current goes through M2 and D2
elseif Release > Threshold
    R = RM2 + RD2;
    V = VM2 + VD2;
% When PWM and Release is low current goes through D3 and D2
else
    R = RD3 + RD2;
    V = VD3 + VD2;
end

% When no current goes through the coil the diode equivalent components
% is off
if CoilCurrent == 0
    R = 1e8;
    V = 0;
end
end

Listing B.2: Coil current circuit parameters MATLAB function for the PWM signal approach.
function [CoilInputVoltage, InputVoltage] = ...
  InputVoltage(On, PWM DutyCycle, Time, Threshold, ...
  RMS Amplitude, Frequency, Phase, Power, Coil Current)
%codegen

% InputVoltage calculates the sinusoidal input voltage level. It
% outputs voltage for V_mon and coil, the voltage is the average value
% calculated using the PWM duty cycle. It has a power restriction depending
% on the current drawn by the coil.
%
% On - Determines if the supply is on or off
% PWM Duty Cycle - PWM signal
% Time - Simulation time
% Threshold - Supply control threshold value
% RMS Amplitude - RMS amplitude
% Frequency - Supply frequency
% Phase - Supply phase
% Power - Maximum supply power
% Coil Current - Current through coil
%
% CoilInputVoltage - Input voltage falling over coil circuit
% InputVoltage - Nominal input voltage

% Duty cycle in decimal form
DutyCycle = PWM Duty Cycle;

% When active supply voltage
if On > Threshold
  % Nominal supply voltage
  InputVoltage = sqrt(2)*RMS Amplitude*... 
  sin(2*pi*(Frequency*Time + Phase/360));
  % Supply voltage is limited by power when too much current is
  % drawn, p = i*u (Only coil current is considered)
  if abs(InputVoltage*Coil Current) > Power
    InputVoltage = sign(InputVoltage)*Power/Coil Current;
  end
  % Average supply voltage falls over coil
  CoilInputVoltage = Duty Cycle*Input Voltage;
% When inactive supply voltage: no voltage over coil and voltage for
% V_mon is switched away in V_mon calculation block.
else
  InputVoltage = 0;
  CoilInputVoltage = 0;
end
end

Listing B.3: Input voltage MATLAB function for the PWM duty cycle approach.
function [CoilInputVoltage, InputVoltage] = ...
    InputVoltage (On, PWM, Time, Threshold, RMSAmplitude, ... 
    Frequency, Phase, Power, CoilCurrent)

%#codegen

% InputVoltage calculates the sinusoidal supply voltage level. It outputs
% voltage for V_mon and coil, coil voltage is active when PWM is high. It
% has a power restriction depending on the current drawn by the coil.
% %
% % On   - Determines if the supply is on or off
% % PWM  - PWM signal
% % Time - Simulation time
% % Threshold - Logical threshold value
% % RMSAmplitude - RMS amplitude
% % Frequency - Supply frequency
% % Phase - Supply phase
% % Power - Maximum supply power
% % CoilCurrent - Current through coil
% %
% % CoilInputVoltage - Supply voltage falling over coil circuit
% % InputVoltage  - Nominal supply voltage

% When active supply voltage
if On > Threshold
    % Nominal supply voltage
    InputVoltage = sqrt(2)*RMSAmplitude*...
        sin(2*pi*(Frequency*Time + Phase/360));

    % When PWM is high supply voltage falls over coil circuit.
    if PWM > Threshold
        % Supply voltage is limited by power when too much current is
        % drawn, p = i*u (Only coil current is considered)
        if abs(InputVoltage*CoilCurrent) > Power
            InputVoltage = sign(InputVoltage)*Power/CoilCurrent;
        end
        CoilInputVoltage = InputVoltage;
    else
        CoilInputVoltage = 0;
    end

    % When inactive supply voltage: no voltage over coil and voltage for
    % V_mon is switched away in V_mon calculation block.
    else
        InputVoltage = 0;
        CoilInputVoltage = 0;
    end
end

Listing B.4: Input voltage MATLAB function for the PWM signal approach.
Parameter Values
The MATLAB scripts used when initializing the MATLAB workspace, used for the simulations, are shown in Listing B.5 - Listing B.7.

```matlab
% Set parameters for the electronics
% Components based on document:
% Title:       PCBA AP 370 100-250V AC/DC
% Document number: 10FBR27169G1303
% Date:        2010-12-8
%--------------------------------------------------------------------------
% General electronic values
% * The supply control low voltage level       [V]
% * The supply control high voltage level     [V]
% * The supply control threshold voltage level, high/2 [V]
% * The microcontroller low voltage level     [V]
% * The microcontroller high voltage level    [V]
% * The microcontroller threshold voltage level, high/2 [V]
%--------------------------------------------------------------------------
GeneralElectronicValues = struct('SupplyControlLow', 0, ...
                                 'SupplyControlHigh', 10, ...
                                 'SupplyControlThreshold', 5, ...
                                 'MicrocontrollerLow', 0, ...
                                 'MicrocontrollerHigh', 3.3, ...
                                 'MicrocontrollerThreshold', 1.65);

% Supply voltage values
% * Amplitude (RMS)                          [V]
% * Frequency                                [Hz]
% * Phase offset                             [Degree]
% * Maximum power (p = i*u)                  [W]
%--------------------------------------------------------------------------
InputVoltage = struct('RMSAmplitude', 250, ...
                      'Frequency', 50, ...
                      'Phase', 0, ...
                      'Power', 1.5e3);

% Components resistances
% * Resistor R201                             [Ohm]
% * Transistor M1                              [Ohm]
% * Transistor M2                              [Ohm]
% * Diode D2                                  [Ohm]
% * Zener diode D3                             [Ohm]
%--------------------------------------------------------------------------
R = struct('R201', 0.27, ...
           'M1', 0.43, ...
           'M2', 0.21, ...
           'D2', 0, ...
           'D3', 0);

% Components voltage offset
% * Transistor M1                              [V]
% * Transistor M2                              [V]
% * Diode D2                                  [V]
% * Zener diode D3                             [V]
%--------------------------------------------------------------------------
V = struct('M1', 1.3, ...
           'M2', 1.3, ...
           'D2', 1.25, ...
           'D3', 30);
```
% Structure of electrical components  
%   * Resistances [Ohm]  
%   * Offset voltages [V]  

ElectricalComponents = struct('R', R, ...  
    'V', V);

% Coil resistance depending on temperature (linear)  
% R = R0*(1+Alpha*(T - T0))  
%   * Resistance at temperature T0 [Ohm]  
%   * Temperature [Degree]  
%   * Reference temperature [Degree]  
%   * Linear temperature coefficient [Ohm/Degree]  

CoilResistance = struct('R0', 13.2, ...  
    'T', 27, ...  
    'T0', 27, ...  
    'Alpha', 0.003862);

% Components involved with supply voltage monitoring  
%   * Resistor R7 + R8 [Ohm]  
%   * Resistor R9 + R10 [Ohm]  
%   * Capacitor C6 [F]  

VMonComponents = struct('R1', 860e3, ...  
    'R2', 2.3e3, ...  
    'C', 100e-9);

% Components involved with coil current monitoring  
%   * Resistor R7 + R8 [Ohm]  
%   * Resistor R9 + R10 [Ohm]  
%   * Capacitor C6 [F]  

IMonComponents = struct('R', 0.27);

Listing B.5: Electrical parameter values for the ECC model.

% Load lookup tables for the electromagnetics  
%----------------------------------------------------------------------------------------------------  
% Force table  
%----------------------------------------------------------------------------------------------------  

% Original table (not used at the moment)  
% load('ForceTable.mat');  
% breakpoints_1 = force_table(2:end,1);  
% ForceTable = struct('Table', force_table(2:end,2:end),...  
%    'Breakpoints1',breakpoints_1,...  
%    'Breakpoints2',breakpoints_2);

% Finer mesh (to be able to use linear algorithm in LUT)  
% load('ForceTable_Splined0125.mat');  
% breakpoints_1 = force(2:end,1);  
% breakpoints_2 = force(1,2:end);  
% ForceTable = struct('Table', force(2:end,2:end),...  
%    'Breakpoints1',breakpoints_1,...  
%    'Breakpoints2',breakpoints_2);

% Inductance table  
%----------------------------------------------------------------------------------------------------  

% Original table (not used at the moment)  
% load('InductanceTable.mat');  
% breakpoints_1 = inductance_table(2:end,1);  
% inductanceTable = struct('Table', inductance_table(2:end,2:end),...  
%    'Breakpoints1',breakpoints_1,...  
%    'Breakpoints2',breakpoints_2);
Listing B.6: Electromagnetics parameters and LUTs.

% Finer mesh (to be able to use linear algorithm in LUT)
load('InductanceTable_Splined0125.mat');
breakpoints_1 = inductance(2:end,1);
breakpoints_2 = inductance(1,2:end);
InductanceTable = struct('Table', inductance(2:end,2:end),
    'Breakpoints1',breakpoints_1,...
    'Breakpoints2',breakpoints_2);

% Coil
% * Number of turns on the coil [-]
Coil = struct('CoilTurns', 900);

% Set parameters for the mechanics
%==========================================================================
% Set up variant objects for the variant block
VariantModel1 = Simulink.Variant('VSS_MODE==1');
VariantModel2 = Simulink.Variant('VSS_MODE==2');
VariantModel3 = Simulink.Variant('VSS_MODE==3');
VariantModel4 = Simulink.Variant('VSS_MODE==4');

% Use model 2 initially
VSS_MODE = 2;

% General values
% * The contactor's mounting angle [rad]
% * The gravitational acceleration [m/s^2]
% * Offset between x-y and between x_bridge-y [m]
% * Offset between x_anchor and x_bridge [m] (used in model 4)
GeneralMechanicValues = struct('MountingAngle', 0,...
    'GravitationalAcceleration', 9.81,...
    'OffsetXY', 0.0038,...
    'OffsetAnchorBridge', 0.0005);

% Contact bridge
% * Mass [kg]
% * Distance from x_bridge = 0 to the roof [m] (used in model 4)
% * Initial position (used in model 4)
ContactBridge = struct('Mass', 0.169 + 0.055,...
    'DistanceToRoof', 0.0125,...
    'InitialPosition', 0.0125);

% Anchor
% * Mass [kg]
% * Distance from x = 0 to the roof [m] (used in model 1,2 and 3)
% * Initial position [m] (used in models 1 and 2)
% * Upper bound of position [m] (used in models 1 and 2)
% * Lower bound of position [m] (used in models 1 and 2)
% * Coefficient of restitution [-] (used in model 2)
Anchor = struct('Mass', 0.484,...
    'DistanceToRoof', 0.012,...
    'InitialPosition', 0.012,...
    'PositionUpperBound', 0.012,...
    'PositionLowerBound', 0,...
    'CoefficientOfRestitution', 0.458);

% Movable contacts
% * Mass [kg]
% * External initial position (y0) [m]
% * Upper bound of external position [m] (used in models 1 and 2)
% * Lower bound of external position [m] (used in models 1 and 2)
% * Coefficient of restitution [-] (used in model 2)
% MovableContacts = struct('Mass', 3*(0.055+0.003), ... 
'ExternalInitialPosition', 0.0082, ... 
'ExternalPositionUpperBound', 0.0082, ... 
'ExternalPositionLowerBound', 0, ... 
'CoefficientOfRestitution', 0.5);

% Release spring
% * Precompression  [m]
% * Spring constant  [N/m]
% ReleaseSprings = struct('Precompression', 0.038, ... 
'SpringConstant', 375);

% Connection spring
% * Precompression  [m]
% * Spring constant  [N/m]
% ConnectionSprings = struct('Precompression', 0.0232, ... 
'SpringConstant', 5800);

% Elastogaps (spring-damper component)
% Type 1: is used for the impact between the anchor and the fixed iron core. (Model 3 and 4)
% Type 2: is used for the interaction between the contact bridge and the movable contacts (all models), the impact between the movable contacts and fixed contacts (model 3 and 4), and the spring constant between the contact bridge and the roof (model 4)
% Type 3: is used for the anchor-bridge impact with the roof in model 3.
% Type 4: is used for the interaction between the anchor and the contact bridge in model 4.
% * Spring constant  [N/m]
% * UnstretchedLength  [m]
% * Damping constant  [Ns/m]
% ElastogapType1 = struct('SpringConstant', 5e7, ... 
'UnstretchedLength', 0, ... 
'DampingConstant', 3e3);
% ElastogapType2 = struct('SpringConstant', 2e6, ... 
'UnstretchedLength', 0, ... 
'DampingConstant', 6e2);
% ElastogapType3 = struct('SpringConstant', 3.4e5, ... 
'UnstretchedLength', 0, ... 
'DampingConstant', 6e2);
% ElastogapType4 = struct('SpringConstant', 1.3e6, ... 
'UnstretchedLength', 0, ... 
'DampingConstant', 6e2);

Listing B.7: Mechanics parameter values.
Appendix C – DS2202 HIL I/O Board Pin Configuration

This appendix shows the DS2202 HIL I/O board’s connectors’ pin configuration, which is used for transmitting signals between the real-time simulations and the modified ECC. The board has three I/O connectors (dSPACE GmbH, 2012) named P1, P2 and P3 with 100 pins, 100 pins and 50 pins respectively. The I/O board is delivered with two adapter cables for connectors P1 and P2. These adapter cables convert the 100 pins into two 50 pins connectors with postfixes A and B. For the contactor system only connector P1A and P2A is used. The used pin assignments are shown in Table C.1.

Table C.1: Assigned pin configuration.

<table>
<thead>
<tr>
<th>Signal reference name</th>
<th>Function</th>
<th>DS2202 I/O board</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I/O</td>
<td>Connector</td>
</tr>
<tr>
<td>PWM</td>
<td>Transmit ECC PWM signal for simulator measurement</td>
<td>Input</td>
</tr>
<tr>
<td>Release</td>
<td>Transmit ECC release signal for simulator measurement</td>
<td>Input</td>
</tr>
<tr>
<td>ON_N</td>
<td>Transmit simulator PLC control signal for ECC</td>
<td>Output</td>
</tr>
<tr>
<td>OFF</td>
<td>Transmit simulator PLC control signal for ECC</td>
<td>Output</td>
</tr>
<tr>
<td>I_MON</td>
<td>Transmit simulator coil current monitoring signal for ECC</td>
<td>Output</td>
</tr>
<tr>
<td>V_MON</td>
<td>Transmit simulator voltage monitoring signal for ECC</td>
<td>Output</td>
</tr>
<tr>
<td>Supply_ON</td>
<td>Transmit simulator supply switch control signal for</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>constructed hardware ECC</td>
<td></td>
</tr>
<tr>
<td>Supply</td>
<td>Transmit simulator supply voltage for constructed</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>hardware and ECC</td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td>Transmit simulator ground reference for constructed</td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>hardware and ECC</td>
<td></td>
</tr>
</tbody>
</table>

The names of the connector’s pins correspond to the function of the pins. PWM_IN10 is used for measuring PWM signals, analog-to-digital converter (ADC) measures analog voltage signals and digital-to-analog converter (DAC) outputs digital values as analog voltages. The +12V and GND corresponds to the power supply output and the ground reference respectively for the simulator. Most of the outputted signals are differential between its two pins, the pin column noted as bar is used as a ground sensing pin for the respective signals.
The connectors on the 50 pins side of the adapter cable is in the form of D-sub socket connectors and the outgoing pin numbering can be seen in Figure C.1. Furthermore ABB is using D-sub extension cables with plug-to-plug connectors; therefore the constructed hardware interface has socket connectors as its inputs.

Figure C.1: Sketch off a 50 pin D-sub numbering.