Examensarbete

Modeling and Hardware-in-the-loop Simulations of Contactor Dynamics
- Mechanics, Electromagnetics and Software

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Jon Tjerngren

LiTH-ISY-EX--14/4778--SE

Linköping 2014
Modeling and Hardware-in-the-loop Simulations of Contactor Dynamics

- Mechanics, Electromagnetics and Software

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Jon Tjerngren

LiTH-ISY-EX--14/4778--SE

Handledare: Clas Veibäck
            ISY, Linköpings universitet
            Mattias Rehnman
            ABB AB

Examinator: Anders Hansson
            ISY, Linköpings universitet

Linköping, 18 juni 2014
Nyckelord
Keywords
modeling, contactor dynamics, real-time simulations, hardware-in-the-loop, HIL, dSPACE
Abstract
This master thesis’s subject is to model an ABB contactor’s dynamics and to develop a hardware-in-the-loop simulation environment. The hardware-in-the-loop method utilizes computer models that are simulated in a real-time simulator. The real-time simulator is connected to hardware components.

A contactor is an electrically controlled mechanical switching device and it is used in circuits where large currents can occur. In this thesis, the contactor is divided into three separate subsystems and models are developed for each of them. The three subsystems correspond to the contactor’s mechanics, electromagnetics and electronic components. Computer models are implemented in MATLAB and Simulink to realize the subsystems. The hardware part, of the hardware-in-the-loop simulations, consists of electronic parts that are not modeled. To connect the hardware part to a real-time simulator, from dSPACE, a hardware interface was constructed.

This report focuses on the modeling of the mechanics and the electromagnetics as well as the software implementations. The thesis work was carried out in collaboration with another student. The focuses of his report are the modeling of the electronics and the construction of the hardware interface.

Validation of the hardware-in-the-loop simulations is done by using measurements collected from a real contactor. The conclusion is that the simulations of the contactor’s behavior correspond well with a real contactor.
Acknowledgements

First of all I would like to thank ABB for the opportunity to work with this master thesis. It has been enjoyable to work with a larger project which is both interesting and challenging. I have had to use several aspects from my education at Linköping University.

I am grateful to the ABB employees I have worked with, they have been both easy to work with and helpful. I especially want to thank Mattias Rehnman, my supervisor at ABB, as well as Gunnar Johansson, David Karlen and Ove Coman for their support and expert knowledge.

I am also very thankful to Anders Hansson, my examiner, and Clas Veibäck, my supervisor, from the university.

During the thesis work my family has given me great support, especially Dan Tjerngren. He was the student whom I collaborated with during this project, and our combined work made it possible to complete the thesis. We had a great time and many valuable discussions.

Västerås, June 2014
Jon 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 ............................................................................................................................ 6

2 Contactor Modeling .......................................................................................................................... 7
   2.1 System Description .................................................................................................................... 9
      2.1.1 Subsystems ....................................................................................................................... 10
      2.1.2 Modeling Advice from ABB ............................................................................................ 10
   2.2 ECC .......................................................................................................................................... 11
   2.3 Electromagnet .......................................................................................................................... 12
      2.3.1 Magnetic Circuit Example ................................................................................................. 12
      2.3.2 Lookup Tables .................................................................................................................. 14
   2.4 Mechanics ............................................................................................................................... 16
      2.4.1 Model Simplifications ....................................................................................................... 17
      2.4.2 Theory .............................................................................................................................. 18
      2.4.3 Model 1 ............................................................................................................................. 21
      2.4.4 Model 2 ............................................................................................................................. 22
      2.4.5 Model 3 ............................................................................................................................. 23
      2.4.6 Model 4 ............................................................................................................................. 24

3 MATLAB and Simulink Implementation ......................................................................................... 27
   3.1 Subsystem Blocks ..................................................................................................................... 28
      3.1.1 Simulation Control .............................................................................................................. 28
      3.1.2 ECC ................................................................................................................................. 28
      3.1.3 Electromagnet .................................................................................................................... 28
      3.1.4 Mechanics .......................................................................................................................... 29
   3.2 Parameter Values ...................................................................................................................... 30
   3.3 Non-real-time Simulations and Validation ................................................................................. 30
      3.3.1 ECC .................................................................................................................................. 30
      3.3.2 Electromagnet .................................................................................................................... 32
3.3.3 Mechanics .......................................................... 32
3.3.4 Explanation of the Simulations ................................ 36

4 Hardware-in-the-loop System Setup ........................................ 39
  4.1 dSPACE Simulator ..................................................... 39
  4.2 Constructed Hardware Interface ..................................... 40
  4.3 Extended Simulink Model ............................................. 41

5 Real-time Simulations and Validation ........................................ 43
  5.1 Real-time Simulations .................................................. 43
     5.1.1 ControlDesk ....................................................... 44
  5.2 Measurements of a Real Contactor .................................. 45
  5.3 Validation .............................................................. 46
     5.3.1 PWM Measurement Problems ................................. 46
     5.3.2 ECC ............................................................... 46
     5.3.3 Electromagnet ..................................................... 48
     5.3.4 Mechanics ......................................................... 48

6 Discussion ........................................................................ 53
  6.1 Benefits of the HIL Concept ........................................... 53
  6.2 Model Simplifications ................................................... 53
  6.3 Validations ............................................................... 54
  6.4 Future Improvements ................................................... 54

7 References ......................................................................... 55

Appendix A – Work Process ..................................................... 57
Appendix B – MATLAB Scripts .................................................. 61
1 Introduction

This master thesis was made for ABB, at their Control Products unit in Västerås, Sweden. Control Products is part of the Low Voltage Products division. The division develops products and systems which control, measure and protect electrical devices in industrial applications (ABB, 2014). One of the tasks Control Products work with is to design, produce and market low voltage contactors. A contactor is a controllable electrical switching device, similar to a relay. One of the most common applications for contactors is the startup of electrical motors. ABB has several different contactors depending on the application, Figure 1.1 shows three contactors.

Compared to relays, contactors are used when opening and closing electrical circuits where large currents can occur. An electrical control card (ECC) regulates the contactor to open, close or hold the switch. The switching is implemented by using an electromagnet, controlled by the ECC, to generate a force which affects the contactor’s mechanics.

![Different contactors](image)

**Figure 1.1: Different contactors.**

1.1 Background

The ECC contains a microcontroller and when its software has been developed or updated, it is important to check that the control algorithms result in the correct behavior of the contactor. This can be difficult and time consuming, since it may be needed to setup and perform the tests with real contactors. Currently the testing is done by visually assessing measurements from specific test cases. This is further complicated by that there are several different types of contactors.

ABB is interested in using a hardware-in-the-loop (HIL) simulation setup to speed up the testing and development of the microcontroller’s software. ABB is already doing this for their soft starter products. In a HIL setup, part of a system is implemented as computer models and another as hardware. A real-time simulator runs the computer models and communicates with the hardware. This allows for better control when testing systems, for example model parameters can be changed to reflect specific situations. The microcontroller of the ECC is the main part of the hardware in the HIL setup.
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

Figure 1.2 shows an example of an application using a contactor as well as a sketch illustrating the HIL setup. In the top part of the figure, the contactor is used as a switch for an external circuit (ExC). The contactor is made up by the ECC, the electromagnet and the mechanics. The interactions between the components are also shown.

The ECC’s input voltage may have one or two purposes. The first purpose is to act as power supply and the second purpose, which is optional, is to control the closing and opening of the contactor. The input voltage can be alternating current (AC) or direct current (DC) depending on the ECC used in the contactor. There are several different contactors, and the one to use depends on the input voltage’s amplitude. For example, there are contactors working in the voltage ranges 24 – 60V, 48 – 130V, 100 – 250V and 250 – 500V. If the input voltage is not used as the control signal, a programmable logic controller (PLC) controls the contactor instead.

The shaded areas in the top of the figure represent the separation of the contactor into hardware and computer models for the HIL simulations setup. The setup is illustrated in the bottom part of the figure. Only a part of the ECC is implemented as computer models, the remaining parts are the hardware components of the HIL setup. The electromagnet and the mechanics are both implemented as computer models. The hardware interface block acts as the link between the hardware and the real-time simulator. For example, it passes the generated signals from the simulator to the hardware.
1.4 Purpose
This master thesis has three distinct purposes. The first is to identify which parts of the contactor to keep as hardware and which parts to implement as computer models. The models have to be suitable for real-time simulations. The second purpose is to integrate the software implementation with the real-time simulator in the HIL setup. It also involves constructing the hardware interface necessary for connecting the ECC’s kept hardware with the real-time simulator. The final purpose is to perform real-time simulations and to validate the results by comparing them with measurements from a real test case.

The contactor consists of three subsystems; the ECC, the electromagnetics and the mechanics. The computer models are implemented in MATLAB and Simulink. The reason for this is that ABB has a real-time simulator, from the company dSPACE (dSPACE, 2014), which supports these tools.

The purposes are summarized in the following list:

- 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 project is carried out by two students. The other student is Dan Tjerngren who studied “Master Programme in Engineering Physics and Electrical Engineering” with focus on electronics at Luleå University of Technology.

The thesis work is divided between us in such a way that it is easy to identify each other’s contribution. My main responsibility is the mechanical and the electromagnet models as well as the software implementations. Dan’s main responsibility is the electrical models and the construction of the necessary hardware.

We made two separate reports documenting our own contributions as well as our combined efforts to the project; this was required since we studied at two different universities. This means that not all the details about the Dan’s work will be presented here, it will only be summarized. To get the full overview of the project, both reports have to be read. A description of the work distribution and how the project proceeded can be found in Appendix A – Work Process.

Dan’s report (Tjerngren, 2014) is published at Luleå University of Technology.

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 the whole chapter but it is similar to Dan’s introduction since it describes the same project. Only the literature study was written together.

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

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

- **Chapter 4 – Hardware-in-the-loop Setup**
  - The introduction and 4.1 were written together and are identical in both reports.
  - 4.2 is a summary of Dan’s construction of the hardware interface. We wrote the summary together.
  - In this report, I have written 4.3. We wrote a summary together to be used in Dan’s report.

- **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, more simulation results are presented than in Dan’s report.

- **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 Dan’s since it describes our cooperation.

- **Appendix B – MATLAB Scripts**
  - The whole appendix was written together and it is identical in both reports.
1.6 Method and Material
Several different tools and sources of information were used to perform the thesis. The most useful resource was the ABB personnel, and they were always willing to help. At Control Products they have a lot of experience and knowledge of contactors. They shared this information during meetings, discussions and a few introductory lectures about contactors. Articles, books and the Internet were used to find more information.

The dSPACE real-time simulator supports computer models implemented in MATLAB and Simulink. Therefore it was a requirement, from ABB, to implement the contactor models in these tools. ABB provided a MATLAB and Simulink license on the HIL setup which was used for the implementation. ABB also provided computers, office supplies and workspaces. Access to an electronics lab and a mechanical workshop were provided to be able to construct the necessary hardware implementation. The measurements for the validation phase were collected in a testing room, with help from ABB.

1.7 General Limitations
ABB wants a general model of a contactor that can work with as many different contactors as possible. In this thesis, the simulation and validation processes are limited to only one contactor. It is the ABB contactor AF370 which is controlled by a 100 – 250V AC 50/60 Hz input voltage signal. The simulations are performed in real-time; this means that there are strict demands on the simulation’s calculation time. Therefore, the implemented models cannot be too complex, since the simulation time can then be too long. In this thesis it is more important to develop models that can test the robustness of the microcontroller’s software than to create detailed models of reality. Therefore some model simplifications are acceptable.

There are a few problems that occur when switching an ExC which are conducting large currents. One example is that an electric arc is created in the gap between the contact surfaces when opening and closing the circuit, this can cause the contacts to be welded together (Johansson & Karlen, 2014). This problem and others are outside the scope of this thesis; only the contactor’s internal behavior is studied, modeled and simulated.

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>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 many different contactors, but they are similar in their basic function. Figure 2.1 shows an AF370 contactor as well as several of its components.

![AF370 contactor components](image)

Figure 2.1: Inside of a AF370 contactor.

Sketches illustrating more components of a contactor from ABB are shown 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

If a contactor is in its open state and kept there by the release springs, and no ExC is connected, then the contactor system’s behavior can be described as below.

The control signal to the ECC can come from either the input voltage or in some applications from a PLC interface. When the ECC receives a signal to close the contactor, it regulates the current through the electromagnet’s coil. In turn, this generates a magnetic field around the coil so that a magnetic force attracts the movable core, called the anchor. If this force is large enough to overcome the force from the release springs, it closes the gap between the two cores. The anchor pulls the contact bridge, which connects all the movable masses, along with itself. This will in turn cause the movable contacts to move towards the fixed contacts.

The movable and fixed contacts are the parts of the contactor that makes the actual switching of the ExC. The connection springs are precompressed to avoid too much bouncing at the impact between the contacts when they are brought together. The gap between the cores is larger than the gap between the contacts; therefore the anchor will continue to move towards the fixed core. Bouncing will also occur when the anchor hits the fixed core.

When the contactor is closed the ECC enters a hold state, to reduce the power consumption. In this state the regulated coil current is reduced compared to the current used during closing, which was large because a quick closing is required. When the ECC gets the signal to open the contactor, it demagnetizes the electromagnet’s coil by quickly reducing the coil current. This will lessen the electromagnetic field around the coil. When the electromagnetic force is weak enough, the release springs and connection springs pushes the contact bridge away. The pushing force mainly comes from the connection springs, which are stronger than the release springs.

When the contact bridge moves away, it will drag the anchor with itself and separate it from the fixed core. When the contacts are separated, the release springs return the contact bridge into its opened position. A closing procedure is illustrated in Figure 2.4.

![Figure 2.4: Contactor closing procedure.](image-url)
2.1.1 Subsystems
In this thesis the contactor system is separated into three distinct subsystems. They are called the ECC, the electromagnet and the mechanics.

The signals affecting the subsystems are illustrated in Figure 2.5. The ECC has external control signals which decide if the contactor should be opened or closed. The input voltage is always present and the PLC is optional. Between the ECC and the electromagnet there are the time-dependent coil current, \( i(t) \), and the coil inductance, \( L(i, x) \). The inductance depends on the coil current and the time-dependent anchor position, \( x(t) \). Between the electromagnet and the mechanics there are the anchor position and the magnetic force, \( F(i, x) \). The force depends on the coil current and the anchor position. If the ExC had been included in the thesis, it would have affected the mechanics block.

![System Diagram](image.png)

Figure 2.5: Illustration of subsystem interactions.

2.1.2 Modeling Advice from ABB
ABB’s Corporate Research division, in Västerås, has worked with contactor modeling in Dymola. During a meeting (Andersson & Fransson, 2014) advice was received on how to model the contactor used in this thesis. This included important aspects to consider in the model and where simplifications could be made. Access to some of Corporate Research’s results were provided, which could be used to compare against and validate the Simulink models developed in this project. Corporate Research also had experience with HIL simulations and gave advice about some real-time implementation issues and how to solve them.
2.2 ECC

The ECC’s purpose is to interpret external control signals and to convert these into actions performed by the contactor. More details about the ECC can be found in (Tjerngren, 2014).

The ECC consists of several parts and they are divided into two categories, one for the components realized as computer models and the other for the parts that remain in hardware. The separation is mainly done according to the HIL simulator’s input/output (I/O) board’s limitations. The hardware circuits powered by 12 V or less are kept as hardware. Figure 2.6 shows a block diagram for the ECC.

Simulated functions are represented by the gray boxes. The parts that are kept as hardware are represented by the white boxes. The encircled boxes are called the coil current circuit. Optional functionality is represented by the dashed boxes. The following list shows each block’s category.

- Hardware (white):
  - Microcontroller, controls the contactor.
  - 11.4 V power supply.
  - 3.3 V power supply.
  - Reset circuit, initializes the microcontroller.

- Simulated (gray):
  - Input voltage and rectifier.
  - PLC control, optional control signals for the ECC.
  - Voltage and coil current monitor circuits.
  - Electromagnet, transforms electrical energy into kinetic energy.
  - Pulse width modulated (PWM) switch circuit, regulates the coil current.
  - Discharge/freewheeling circuit, discharges the electromagnet’s coil or limits the coil voltage respectively.

![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.](image-url)
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.

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 Example

To highlight some of the difficulties with the magnetic circuit method, it will be applied on an example. The electromagnet in the example has simpler geometry compared to the ones in the real contactors. The derivations in this section are based on the article (Riba Ruiz, et al., 2010) and equations found in (Sadiku, 2011) and (Nordling & Österman, 2008). The simplified electromagnet is illustrated in Figure 2.7.

![Figure 2.7: Simplified electromagnet.](image)

In Figure 2.8 the electromagnet’s magnetic circuit and an equivalent magnetic circuit, simplified by geometric symmetry, can be seen.

![Figure 2.8: Magnetic circuit (left) and equivalent magnetic circuit (right).](image)
Parameters:

- Cross-section of the core, $S \; [m^2]$ (assumed to be the same everywhere).
- Length of core section $k, l_k \; [m], \; k = 1, 2, 3, 4, 5$.
- Permeability of free space, $\mu_0 \; [H/m]$.
- Relative permeability, $\mu_r \; [-]$.
- Number of turns of the coil, $N \; [-]$.

Variables:

- Position of the movable core, $x \; [m]$.
- Current through the coil, $i \; [A]$.
- Magnetic flux, $\Phi \; [Wb]$.

The reluctances $\mathcal{R}_1, \mathcal{R}_2$ and $\mathcal{R}_3$ correspond to the outer path (excluding the air gap), the air gap and the inner path respectively and they can be expressed as:

\[
\mathcal{R}_1 = \frac{0.5l_1 + l_2 + 0.5l_3}{\mu_0 \mu_r S} = \frac{l_1}{\mu_0 \mu_r S}, \quad (2.3.1)
\]

\[
\mathcal{R}_2 = \frac{x}{\mu_0 S}, \quad (2.3.2)
\]

\[
\mathcal{R}_3 = \frac{l_4}{\mu_0 \mu_r S} + \frac{l_5 + x}{\mu_0 S}. \quad (2.3.3)
\]

The magnetomotive force $\mathcal{F}$ and the total reluctance $\mathcal{R}$ can be expressed as:

\[
\mathcal{F} = Ni = \Phi (\mathcal{R}_1/2 + \mathcal{R}_2/2 + \mathcal{R}_3) = \Phi \mathcal{R}. \quad (2.3.4)
\]

The magnetic field induces an auto-inductance $L$ and together with equation (2.3.4) it can be expressed as:

\[
L = N^2 \frac{\Phi}{\mathcal{F}} = \frac{N^2}{\mathcal{R}}. \quad (2.3.5)
\]

The magnetic force exerted upon the movable core can be calculated by considering the magnetic energy $E_{mag}$ that the coil supplies to the core.

\[
E_{mag} = \frac{1}{2} Li^2. \quad (2.3.6)
\]

If the time interval in the simulation is small, the non-linear characteristics of the electromagnet can be approximated to be linear. Then the magnetic force can be calculated through:

\[
F_{mag} = -\frac{dE_{mag}}{dx}. \quad (2.3.7)
\]
One problem with the magnetic circuit approach is that $\Re$ depends on the permeability. The contactors electromagnets have ferromagnetic cores and this means that the cores’ $B \rightarrow H$ 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.

One way to handle this would be to perform measurements of the magnetic field and then try to fit a model to those measurements.

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 2.9 illustrates fringing and hysteresis.

![Figure 2.9: Illustrations of fringing (left) and hysteresis (right).](image)

### 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 (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 2.10 and Figure 2.11.
Figure 2.10: Lookup table for the electromagnetic force.

Figure 2.11: Lookup table for the electromagnet’s inductance.
2.4 Mechanics

The mechanics’ purpose is to perform the actual switching of the ExC.

Two main components were identified during the planning phase to describe the contactor’s mechanical behavior; these are springs and movable masses. A suggestion from ABB Corporate Research (Andersson & Fransson, 2014) was to use a spring-damper component to model the interaction between movable masses as well as, in some cases, impacts.

A sketch illustrating the contactor’s mechanical system can be seen in Figure 2.12. The figure also points out the implemented models’ position references.

![Figure 2.12: 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.
Among the more important events, in the mechanical process, is the impact that occurs when the masses hit each other. The three most interesting impacts and subsequent bouncing happen between the anchor and the fixed core, between the movable contacts and the fixed contacts and between the contact bridge and the roof of the housing. Two different methods are used to describe the impacts, one uses the coefficient of restitution and the other utilizes the previously mentioned spring-damper component.

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 2.13 (external)

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.

2.4.1 Model Simplifications
The different contactors have a high degree of geometrical symmetry. This makes it reasonable to aggregate both the release springs and the connection springs, into two separate springs for each type. The three movable contacts are also merged into one mass.

Other simplifications are that all the masses are modeled as point masses, with constant mass, and assumed to only move in one dimension. The friction between the components is also not included in this thesis.

In model 1, 2 and 3, the interaction between the anchor and the contact bridge is disregarded and therefore they are merged into one single movable point mass. In model 4 they are separated.

The effects from the ExC are not included in this thesis.
2.4.2 Theory

The contactor mechanics are kept relatively simple because of the real-time simulations. The mechanical components are described by standard linear equations.

Movable Masses

The position, the velocity and the acceleration of the movable masses are calculated by using Newton’s second law of motion (Kraige & Meriam, 2008). That is, the sum of all the internal and external forces affecting a point mass equals its acceleration scaled by its mass.

Parameters:

- Mass, \( m \) [kg].

Variables:

- Position, \( x \) [m].
- Velocity, \( \dot{x} \) [m/s].
- Acceleration, \( \ddot{x} \) [m/s\(^2\)].

Dynamic equation:

\[
m\ddot{x} = \sum F. \tag{2.4.1}
\]

Release and Connection Springs

For both the release springs and the connection springs one end is fixed, while the position of the other end is set in a movable mass. The model used to describe the spring forces is linear according to (Kraige & Meriam, 2008). The release springs and the connection springs are both precompressed. A general case for how the springs are used in the implemented models is illustrated in Figure 2.14.

![Figure 2.14: General case for the spring components.](image)

Parameters:

- Unstretched spring length, \( l_0 \) [m].
- Precompression, \( \delta \) [m].
- Spring constant, \( k \) [N/m].
Variables:

- Offset from the precompression, \( x \ [m] \).

Equation:

\[
F = k(\delta + x). \tag{2.4.2}
\]

**Spring-Damper Components**

The spring-damper components used in this thesis are based on a component called elastogap, from the Modelica Standard Library documentation (Modelica, 2014). The spring part is modeled to be linear. Compared to the springs in the previous section, these are not modeled with any precompression. The force is instead calculated from the unstretched spring length, \( l_0 \). The damper is also modeled to be linear.

The spring-damper component is only active when the corresponding mass’s position is below the unstretched length of the spring. The damper force is limited, to avoid two unphysical events. The first is discontinuities in the force affecting the mass at the moment of impact. The other is pulling forces that can occur when the mass is leaving the spring-damper, for example during bounces. A general case of how the spring-damper components are used in the models is illustrated in Figure 2.15.

![Figure 2.15: General case for the spring-damper components.](image)

Parameters:

- Unstretched spring length, \( l_0 \ [m] \).
- Spring constant, \( k \ [N/m] \).
- Damping constant, \( \lambda \ [Ns/m] \).

Variables:

- Position of a movable mass, \( x \ [m] \).
- Velocity of a movable mass, \( \dot{x} \ [m/s] \).

Equations:

\[
F(x, \dot{x}) = \begin{cases} 
0, & x > l_0 \\
F_s(x) - F_d(x, \dot{x}), & x \leq l_0
\end{cases} \tag{2.4.3}
\]
$F_s(x)$ and $F_d(x, \dot{x})$ are the spring and the damper forces respectively:

$$F_s(x) = k(l_0 - x),$$  \quad (2.4.4)

$$F_d(x, \dot{x}) = \begin{cases} F_s(x), & \lambda \dot{x} > F_s(x) \\ \lambda \dot{x}, & -F_s(x) \leq \lambda \dot{x} \leq F_s(x) \\ -F_s(x), & \lambda \dot{x} < -F_s(x) \end{cases}$$  \quad (2.4.5)

According to equation (2.4.3) the spring-damper only generates a force when it is active, which is when $x \leq l_0$. The damper force in equation (2.4.5) is bounded by the spring force. The upper bound is to avoid the possibility for a pulling force when the damper force is larger than the spring force. The lower bound is to avoid discontinuities in the force at impacts.

**Coefficient of Restitution**

Another way to handle impacts, compared to the spring-dampers, is to use the coefficient of restitution (Nordling & Österman, 2008). All the impacts modeled with this method, in this thesis, are between a movable mass and a fixed mass.

Parameters:

- Coefficient of restitution, $\epsilon [-]$.

Variables:

- Position of the movable mass before the impact, $x^b [m]$.
- Position of the movable mass after the impact, $x^a [m]$.
- Velocity of the movable mass before the impact, $\dot{x}^b [m/s]$.
- Velocity of the movable mass after the impact, $\dot{x}^a [m/s]$.
- Velocity of the fixed mass before the impact, $v^b [m/s]$.
- Velocity of the fixed mass after the impact, $v^a [m/s]$.

Equation:

$$\epsilon = \frac{\dot{x}^a - v^a}{v^b - \dot{x}^b} = / fixed \ mass: v^b = v^a = 0 / \Leftrightarrow \dot{x}^a = -\epsilon \dot{x}^b.$$  \quad (2.4.6)

The impacts are in one dimension and $0 \leq \epsilon \leq 1$. By considering the equation for kinetic energy for the movable mass, $m$, the energy loss after the impact can be calculated as:

$$E_k = \frac{1}{2}m\dot{x}^2 \Rightarrow E_k^a = \epsilon^2 E_k^b \Rightarrow Energy \ loss = 1 - \epsilon^2.$$  \quad (2.4.7)
2.4.3 Model 1
A simplified illustration of the first and second models’ components can be seen in Figure 2.16.

Model characteristics:

- The interaction between the anchor and the contact bridge is neglected. Those masses are merged into one movable mass, called the anchor-bridge.
- The movable masses stop at position limits.
- Impacts at the fixed core and the fixed contact are not handled.

Parameters:

- Gravitational acceleration, \( g \) [m/s\(^2\)].
- Mounting angle, \( \alpha \) [rad].
- Anchor-bridge mass, \( m_{ab} \) [kg].
- Distance from the fixed core to the housing’s roof, \( l_1 \) [m].
- Movable contact mass, \( m_c \) [kg].
- Offset from the fixed core to fixed contacts, \( l_2 \) [m].
- Precompression of the release spring, \( \delta_r \) [m].
- Spring constant of the release spring, \( k_r \) [N/m].
- Precompression of the connection spring, \( \delta_c \) [m].
- Spring constant of the connection spring, \( k_c \) [N/m].
- Damper constant \( \lambda_1 \) [Ns/m], spring constant \( k_2 \) [N/m] and unstretched spring length \( l_{0,1} \) [m] for the spring-damper component handling the interaction between the movable masses.

Variables:

- Position of the anchor-bridge mass, \( x \) [m].
- Velocity of the anchor-bridge mass, \( \dot{x} \) [m/s].
- Acceleration of the anchor-bridge mass, \( \ddot{x} \) [m/s\(^2\)].
- Position of the movable contact mass, \( y \) [m].
- Velocity of the movable contact mass, \( \dot{y} \) [m/s].
- Acceleration of the movable contact mass, \( \ddot{y} \) [m/s\(^2\)].
Auxiliary references:

\[ x' = l_1 - x \Rightarrow \dot{x}' = -\dot{x}, \tag{2.4.8} \]

\[ z = l_2 + y - x \Rightarrow \dot{z} = \dot{y} - \dot{x}. \tag{2.4.9} \]

Equations:

\[
m_{ab}\ddot{x} = -F_{mag} + m_{ab}gsin(\alpha) + k_r(\delta_r + x') + k_c(\delta_c + z) - F_{sd,1}(z, \dot{z}),
\]

\[ 0 \leq x \leq l_1, \tag{2.4.10} \]

\[
m_c\ddot{y} = m_c gsin(\alpha) - k_c(\delta_c + z) + F_{sd}(z, \dot{z}), y \geq 0. \tag{2.4.11} \]

\( F_{mag} \) is the force exerted on the anchor, generated from the electromagnet. \( F_{sd,1}(z, \dot{z}) \) is the force from the spring-damper component which handle the interaction between the anchor-bridge mass and the movable contact mass. The force is calculated according to equation (2.4.3).

2.4.4 Model 2

Model 2 expands model 1 by introducing a method to handle the impacts. No new components are added, therefore Figure 2.16 concurs with model 2.

Model characteristics:

- The interaction between the anchor and the contact bridge is neglected. Those masses are merged into one movable mass, called the anchor-bridge.
- All movable masses stop at the position limits.
- Impacts at the fixed core and the fixed contact are handled by using coefficients of restitution.

Additional parameters to the first model’s parameter list:

- Coefficient of restitution, \( \varepsilon_1 [-]. \)
- Coefficient of restitution, \( \varepsilon_2 [-]. \)
- Coefficient of restitution, \( \varepsilon_3 [-]. \)

The same equations, as in model 1, still describe the basic motion of model 2.

The impacts that occur when the anchor hits the fixed core is using \( \varepsilon_1 \), the impacts when the contact bridge hits the housing’s roof is using \( \varepsilon_2 \), and the impacts when the movable contact hits the fixed contact is using \( \varepsilon_3 \). The dynamics can be described by the following equations. Superscript \( b \) refers to before the impact and superscript \( a \) refers to after the impact.

\[
\dot{x}^a = -\varepsilon_1 \dot{x}^b, \tag{2.4.12} 
\]

\[
\dot{x}^a = -\varepsilon_2 \dot{x}^b, \tag{2.4.13} 
\]

\[
\dot{y}^a = -\varepsilon_3 \dot{y}^b. \tag{2.4.14} 
\]
2.4.5 Model 3

Model 3 expands models 1 and 2 by removing the position limits; it also uses a different method to handle the impacts compared to model 2. An illustration of model 3 can be seen in Figure 2.17.

Figure 2.17: Components and references for model 3.

Model characteristics:

- The interaction between the anchor and the contact bridge is neglected. Those masses are merged into one movable mass, called the anchor-bridge.
- No position limits.
- Impacts are handled by using spring-damper components.

Additional parameters to the first model’s parameter list:

- Damper constants $\lambda_i [Ns/m]$, spring constants $k_i [N/m]$ and unstretched spring lengths $l_{0,i} [m]$ for the spring-damper components $i = 2, 3, 4$ handling the impacts.

Auxiliary references:

$$x' = l_1 - x \Rightarrow \dot{x}' = -\ddot{x}, \quad (2.4.15)$$
$$z = l_2 + y - x \Rightarrow \dot{z} = \ddot{y} - \ddot{x}. \quad (2.4.16)$$

Equations:

$$m_{ab}\ddot{x} = -F_{mag} + m_{ab}gsin(\alpha) + k_r(\delta_r + x') + k_c(\delta_c + z) - F_{sd,1}(z, \dot{z}) + F_{sd,2}(x, \dot{x}) - F_{sd,3}(x', \dot{x}'), \quad (2.4.17)$$

$$m_c\ddot{y} = m_cgsin(\alpha) - k_c(\delta_c + z) + F_{sd,1}(z, \dot{z}) + F_{sd,4}(y, \dot{y}). \quad (2.4.18)$$

$F_{mag}$ is the force exerted on the anchor, generated from the electromagnet. $F_{sd,i}$ are the forces from the spring-damper components $i = 1, 2, 3, 4$. These forces are calculated according to equation (2.4.3).
2.4.6 Model 4
Model 4 expands model 3 by separating the anchor mass and the contact bridge and adding spring-damper components to model their interaction. An illustration of model 4 can be seen in Figure 2.18.

![Figure 2.18: Components and references for model 4.](image)

Model characteristics:
- The anchor and contact bridge masses are separated, and their interaction is included.
- No position limits.
- Impacts are handled by using spring-damper components.

Additional parameters to the third model’s parameter list:
- Damper constants $\lambda_i [Ns/m]$, spring constants $k_i [N/m]$ and unstretched spring lengths $l_{0i} [m]$ for the spring-damper components $i = 5, 6$ handling the interaction between the anchor and the contact bridge.

Variables:
- Position of the anchor mass, $x_a [m]$.
- Velocity of the anchor mass, $\dot{x}_a [m/s]$.
- Acceleration of the anchor mass, $\ddot{x}_a [m/s^2]$.
- Position of the bridge mass, $x_b [m]$.
- Velocity of the bridge mass, $\dot{x}_b [m/s]$.
- Acceleration of the bridge mass, $\ddot{x}_b [m/s^2]$.
- Position of the movable contact mass, $y [m]$.
- Velocity of the movable contact mass, $\dot{y} [m/s]$.
- Acceleration of the movable contact mass, $\ddot{y} [m/s^2]$. 
Auxiliary references:

\[ x_{rel} = x_b - x_a \Rightarrow \dot{x}_{rel} = \dot{x}_b - \dot{x}_a, \quad (2.4.19) \]

\[ x_b' = l_1 - x_b \Rightarrow \dot{x}_b' = -\dot{x}_b, \quad (2.4.20) \]

\[ z = l_2 + y - x \Rightarrow \dot{z} = \dot{y} - \dot{x}. \quad (2.4.21) \]

Equations:

\[ m_a \ddot{x}_a = -F_{mag} + m_a g \sin(\alpha) + F_{sd,2}(x_a, \dot{x}_a) + F_{sd,5}(x_{rel}, \dot{x}_{rel}) - F_{sd,6}(x_{rel}, \dot{x}_{rel}), \quad (2.4.22) \]

\[ m_b \ddot{x}_b = m_b g \sin(\alpha) + k_r (\delta_r + x_b') + k_c (\delta_c + z) - F_{sd,1}(z, \dot{z}) - F_{sd,3}(x_b', \dot{x}_b') - F_{sd,5}(x_{rel}, \dot{x}_{rel}) + F_{sd,6}(x_{rel}, \dot{x}_{rel}), \quad (2.4.23) \]

\[ m_c \ddot{y} = m_c g \sin(\alpha) - k_c (\delta_c + z) + F_{sd,1}(z, \dot{z}) + F_{sd,4}(y, \dot{y}). \quad (2.4.24) \]

\( F_{mag} \) is the force exerted on the anchor, generated from the electromagnet. \( F_{sd,i} \) are the forces from the spring-damper components. These forces are calculated according to equation (2.4.3).
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.

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, 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 electromagnet and the mechanics blocks are described in more detail. The simulation control and the ECC blocks 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 generates signals which represent the ECC’s microcontroller’s control signals, and they are sent to the ECC’s simulated parts. It simulates a sequence representing the closing, the holding and the opening of a contactor and it is implemented by using timed switch blocks to change the output signals.

3.1.2 ECC

The purpose of the ECC block is to interpret the control signals from the simulation control block. From these signals it performs the necessary calculations using two different approaches. One uses a PWM signal’s high and low values and the other uses the PWM duty cycle. The block calculates the coil current which is used by the electromagnet block.

3.1.3 Electromagnet

The purpose of the electromagnet 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. The coil inductance is sent to the ECC block and the electromagnetic force is sent to the mechanics block.

Figure 3.2 shows the inside of the electromagnet block.

![Figure 3.2: Electromagnet Simulink implementation.](image)

To calculate the outputs, the inputs need to be scaled to correspond to the LUTs’ input arguments. The LUTs’ uses cubic interpolation algorithms to calculate the outputs. Before sending them to the ECC and the mechanics blocks, the outputs need to be modified to fit the rest of the model. The force output changes sign, and the inductance output is changed depending on the number of coil turns as well as to the correct unit.
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.

Figure 3.3 shows the second model of the mechanics. The other models have similar Simulink appearances.

The Simulink implementation corresponds to the equations derived Chapter 2.

- **Release spring**, calculates the force from the release springs according to (2.4.2).
- **Connection spring**, calculates the force from the connection springs according to (2.4.2).
- **Spring-damper**, describes the interaction between the two masses by implementing (2.4.3).
- **Anchor and contact bridge**, calculates the anchor-bridge movements by using (2.4.10).
- **Movable contacts**, calculates the movable contact movements and its auxiliary movements by using (2.4.11) and (2.4.9).
- **The green block**, calculates the auxiliary reference from the contactor housing’s roof to the anchor-bridge mass according to (2.4.8).
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 3.4 and Figure 3.5 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.
Figure 3.4: Simulation and comparison with measurements for the ECC PWM approaches.

Figure 3.5: Closer view of the PWM approaches.
3.3.2 Electromagnet
The force and the inductance outputs from the electromagnet block can be seen in Figure 3.6. The two plots in the figure are the results from the cubic interpolated LUTs.

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

Figure 3.6: Simulation of the electromagnet’s lookup tables.

3.3.3 Mechanics
Figure 3.7 - Figure 3.10 show the four different models of the mechanics, and how they correspond to the same measurements. The only available measurements were of the contact bridge’s position.

To be able to compare the measurements with models 1, 2 and 3, they had to be lowered 1 mm since in these models the anchor and the contact bridge are merged into one mass. In model 4, the anchor and the contact bridge are separate masses; therefore the measurements did not need to be lowered for the comparisons.

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.
Figure 3.7: Simulation and comparison with measurements for model 1 of the mechanics.

Figure 3.8: Simulation and comparison with measurements for model 2 of the mechanics.
Figure 3.9: Simulation and comparison with measurements for model 3 of the mechanics.

Figure 3.10: Simulation and comparison with measurements for model 4 of the mechanics.
Figure 3.11 and Figure 3.12 show more details about model 2. The second figure highlights the impact behavior during closing.

Figure 3.11: Simulation of model 2.

Figure 3.12: Impact during closing for model 2.
3.3.4 Explanation of the Simulations

Figure 3.4, Figure 3.6 and Figure 3.8 show the whole contactor behavior. These figures, together with the following equation (Alciatore & Histand, 2007) is used to explain the behavior,

\[ V(t) = \frac{d\Phi}{dt} = \frac{1}{\Phi} \frac{dL}{dt} = L \frac{di}{dt} + L \frac{di}{dt} \]  

(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 3.4. 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 3.6 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.

![Figure 4.1: 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 Constructed Hardware Interface

For the constructed hardware interface, the necessary signals had to be investigated and the pin configuration of the DS2202 HIL I/O board’s connectors had to be assigned. The modified ECC to be used in the real-time simulations is 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 4.3 the inside of the box is shown. For more details see (Tjerngren, 2014).

Figure 4.3: Box containing the constructed hardware interface.
4.3 Extended Simulink Model

The extended Simulink model is shown in Figure 4.4. The Simulation control block has been modified, to only send the necessary control signals. The signals are the supply on/off and the PLC signals. The supply on/off signal turns the power supply on or off for the modified ECC’s microcontroller. It can also act as the control signal, which indicates if the contactor should be opened or closed. Alternatively, the PLC signals control the microcontroller. These signals are transmitted to the microcontroller, which in turn regulates the PWM signal and sets the release signal.

Figure 4.4: Extended Simulink model.

The Simulink model is also extended with the dSPACE interface block, and its implementation is shown in Figure 4.5. It contains the blocks, from the Simulink RTI library, which connects the Simulink model to the physical channels of the DS2202 HIL I/O board when the model has been compiled into a real-time application and been downloaded to the DS1005 board.

Figure 4.5: dSPACE interface block.

The digital-to-analog converters (DACs) of the I/O board has the output range 0 – 10 V but is mapped to 0 – 1 in the RTI blocks’ inputs. For the analog-to-digital converter (ADC), the range is 0 – 60 V but it is mapped to 0 – 1 in the RTI block’s output. Therefore the gains and saturations are included in the interface block. The PWM block’s duty cycle output is between 0 – 1 and does not need any scaling in this model.
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 2.10 and Figure 2.11. 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).
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.

Figure 5.3: Example of a ControlDesk GUI.
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.

The simulation control block in Figure 4.4 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 Contactor
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.

Figure 5.4: Measurement setup.
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. The 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.

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

**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 3.6 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
Comparisons between the measurements and simulations of model 2, 3 and 4 are shown in Figure 5.9 – Figure 5.11. The same procedure was used, as in the non-real-time simulations, to be able to compare the measurements to model 2 and 3.

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.

Figure 5.12 shows a closer look at the closing of the contactor. The closing is fast, compared to the measurements, which is mainly caused by the problems with the PWM signal.
Figure 5.9: HIL simulation and comparison with measurements, for model 2 of the mechanics (good PWM signal).

Figure 5.10: HIL simulation and comparison with measurements, for model 3 of the mechanics (good PWM signal).
Figure 5.11: HIL simulation and comparison with measurements, for model 4 of the mechanics (good PWM signal).

Figure 5.12: Closer look of the closing phase, for model 2 of the mechanics (good PWM signal).
Figure 5.13 shows comparisons between simulations of the mechanics, using model 2, when the PWM signal measurement is good respectively bad. Figure 5.14 and Figure 5.15 illustrate the closing and opening phases. The effect on the mechanics, due to the bad PWM signal, is a delay in the opening phase.

Figure 5.13: HIL simulations of model 2, for two cases of the PWM signal measurement.

Figure 5.14: Closing phase, for two cases of the PWM signal measurement.
Figure 5.15: 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. An analog filter is proposed in (Tjerngren, 2014) to emulate a more realistic signal of the coil current, for the microcontroller’s measurements. 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


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


Appendix A – Work Process

This thesis was mainly done at ABB Control Products in Västerås, Sweden, by me and Dan Tjerngren. Because we were two students doing this thesis, the work needed to be divided in such a way that it was clear who had done what. Therefore we made an extensive plan for the work where the main responsibilities were assigned between us. The thesis was divided into six phases and within each phase we could mostly work in parallel according to our responsibilities. If any problems arose during the parallel work, we helped each other to solve them. Before beginning a new phase, the work converged into a common node. These nodes allowed us to measure the work progress and to control our readiness for the upcoming phase.

The phases and their purposes were:

- Planning, plan and achieve overview of the problem.
- Study and Modeling, study and define models for the contactor and implement them in Simulink.
- HIL Implementation, integrate the models with the HIL setup.
- Simulation and Validation, real-time simulations and control of the result.
- Reserve, extra time.
- End Phase, finalize the thesis work.

Phases

By dividing the work into several different phases, we got a good overview of how the work would proceed and it allowed us to measure the work’s progress.

An illustration of the phases is shown in the flowchart in Figure A.1.

Phase 1 - Planning

The thesis work began with two weeks of planning and studying of contactors, which we did together. The planning resulted in a more accurate description of the problem, a schedule of the work and the mentioned flowchart illustrating the estimated work distribution and timeframe of the different phases.

The studying included, reading an article (Riba Ruiz, et al., 2010) about modeling contactors, disassembling and inspecting a real contactor and attending two short lectures from ABB. The first lecture was about the general behavior of contactors and problems that arise when switching circuits conducting large currents (Johansson & Karlen, 2014). The second lecture was about the ECC (Coman, 2014).

We also defined the I/O signals needed between the different subsystems of the contactor model.
Phase 6: End phase w. 20-23

Phase 5: Reserve w. 18-19

Phase 4: Simulation and Validation w. 16-17

Phase 3: HIL Implementation w. 11-15
  - LiU half-time meeting
  - LTU half-time meeting

Phase 2: Study and Modeling w. 7-10

Phase 1: Planning w. 4-6

Figure A.1: Flowchart illustrating the estimated work distribution and timeframe.
Phase 2 – Study and Modeling
This phase was mostly done in parallel where we worked with our separate tasks.

I performed a more thorough investigation of the mechanical components and the electromagnet of the contactor. I also looked into the RTI provided by dSPACE, which was used when integrating the Simulink models with the real-time simulator. The investigation resulted in four models of the mechanical behavior with different levels of complexity. It was decided that the electromagnet was to be implemented as LUTs because this made it easier to simulate in real-time.

During this, Dan studied the ECC and read about the dSPACE hardware and its I/O channels. He defined the boundary between the ECC and the Simulink models. His work resulted in two different models of the ECC. He also investigated the hardware interface required for the HIL setup.

We developed several models of the subsystems to be able to investigate and find a complete model, which was suitable for the HIL setup. All the models were implemented in Simulink and the complete system was tested together. For the testing, Dan had implemented a simulation of the microcontroller’s control signals. We did an assessment of the Simulink simulation together.

Phase 3 – HIL Implementation
To connect the Simulink model to the real-time simulator’s I/O HIL board’s physical signals I used the RTI provided by dSPACE. The RTI was in the form of a Simulink library containing the necessary I/O Simulink blocks. These were added to the Simulink model.

During this phase, Dan modified an ECC as well as designed and constructed a hardware interface which was used to connect the modified ECC with the real-time simulator’s I/O board. The hardware interface was installed in a 19-inch rack.

We used MATLAB Coder and Simulink Coder to generate C-code, and to compile the code into a real-time application. This application was then downloaded to the real-time simulator. I used the program ControlDesk’s graphical user interface to show the application’s state during real-time simulations and Dan performed measurements on the constructed hardware interface to check that the signals were correct. We had to make a few adjustments of the Simulink model and download it to the real-time simulator 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 validation was done by comparing simulation results with the measurements.

Phase 5 – Reserve
We had planned two weeks which we could use as a buffer against unexpected problems and delays.

Phase 6 – End phase
During the end phase I finalized the report, opposed another master thesis and prepared 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.

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.
% 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
% PWMDutyCycle - PWM signal
% Time - Simulation time
% Threshold - Supply control threshold value
% RMSAmplitude - RMS amplitude
% Frequency - Supply frequency
% Phase - Supply phase
% Power - Maximum supply power
% CoilCurrent - Current through coil
% CoilInputVoltage - Input voltage falling over coil circuit
% InputVoltage - Nominal input voltage

% Duty cycle in decimal form
DutyCycle = PWMDutyCycle;

% When active supply voltage
if On > Threshold
    % Nominal supply voltage
    InputVoltage = sqrt(2)*RMSAmplitude*...
        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*CoilCurrent) > Power
        InputVoltage = sign(InputVoltage)*Power/CoilCurrent;
    end

    % Average supply voltage falls over coil
    CoilInputVoltage = DutyCycle*InputVoltage;

% 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.

%==========================================================================
% 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]
% Corporate research values
% 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]
% Corporate research values
% V = struct('M1', 1.3, ... 'M2', 1.3, ... 'D2', 1.25, ... 'D3', 30);
Listing B.5: Electrical parameter values for the ECC model.

% 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);

% 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);
% breakpoints_2 = force_table(1,2:end)';
% 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);
% breakpoints_2 = inductance_table(1,2:end)';
% InductanceTable = struct('Table', inductance_table(2:end,2:end),...
%                          'Breakpoints1',breakpoints_1,...
%                          'Breakpoints2',breakpoints_2);
Listing B.6: Electromagnetics parameters and LUTs.

```matlab
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 [m] (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 impact 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]
% * UnstrechedLength [m]
% * Damping constant [Ns/m]
% ElastogapType1 = struct('SpringConstant', 5e7, 'UnstrechedLength', 0, 'DampingConstant', 3e3);
% ElastogapType2 = struct('SpringConstant', 2e6, 'UnstrechedLength', 0, 'DampingConstant', 6e2);
% ElastogapType3 = struct('SpringConstant', 3.4e5, 'UnstrechedLength', 0, 'DampingConstant', 6e2);
% ElastogapType4 = struct('SpringConstant', 1.3e6, 'UnstrechedLength', 0, 'DampingConstant', 6e2);

Listing B.7: Mechanics parameter values.
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Jon Tjerngren