University Diploma Project

UNIVERSITY
OF SKÖVDE

VIRTUAL COMMISSIONING
Emulation of a production cell

University Diploma Project in automation techonology
22.5 ECTS
Spring term Year 2016

Authors: Dennis Binnberg, Viktor Johansson

Supervisor Volvo GTO: Michael Larsson, Stefan Berntsson

Supervisor University of Skövde: Mikel Ayani

Examiner: Amos Ng

Attestation

The authors of this thesis attests that guidelines and agreements from the University of Skövde and Volvo Group Truck Operations have been followed. References have been made correctly by the Harvard system, material that is not referenced is the authors own words. Figures and tables are approved for use in this thesis and have been designed by the authors unless otherwise specified.

Dennis Binnberg

Viktor Johansson

Preface

We would like to dedicate this preface to everyone that helped us in this thesis in automation technology.

First of all, we would like to thank Volvo Group Trucks Operation for the great opportunity to work with you in this project.

We would like thank our supervisors at Volvo, Michael Larsson and Stefan Berntsson for helping us throughout this project with important input, answering questions and support.

We would like to thank our supervisor at the University of Skövde Mikel Ayani for helping us with questions regarding the software, PLC-programming and giving us support throughout the project.

We would like to thank PLC-programmer, Daniel Dahlstrand at Volvo for helping us with PLC-related issues.

We would also like to thank our classmates in PRTPG14h at the University of Skövde for support and collaborations during our thesis period.

Last but not least we would like to thank our families.

Abstract

Volvo is continually updating and replacing their equipment and want to investigate the possibility to shorten the time it takes to implement changes and shorten the time in commissioning projects. The use of an emulation model of a production cell can shorten the commissioning time since the equipment and sequence of the cell can be thoroughly tested before implementation. Volvo also wants to investigate the possibility to validate equipment using emulation. The main objectives are to find an emulation software that suits Volvo's needs and build an emulation model of an actual production cell at Volvo called G750. A literature review was performed in which the authors gained knowledge about virtual commissioning, simulation and emulation and the usage of these. A market survey was conducted in order to find emulation software that could handle Volvo's complex production equipment consisting of ABB robots and Siemens PLC. A method for building emulation models of existing production equipment was found during the literature review. The software used to build the emulation model was Simumatik3D. Other software used to make the model as realistic as possible includes RobotStudio, WinCC and PLCSIM. The emulation model handles approximately 350 inputs and outputs. When the emulation model was finished experiments were conducted in order to answer research questions and to reach the main objectives. The experiments validate that the emulation model is representative of the real production cell regarding programming, fail scenarios and movement.

Keywords:

Emulation, Simumatik3D, Virtual Commissioning, RobotStudio, PLC, Simulation, Robot

Table of Contents

Attestation ..... **i**

Preface ..... **ii**

Abstract ..... **iii**

Table of figures ..... **vii**

Table of tables ..... **viii**

Abbreviations and terminology ..... **ix**

1 Introduction **1**

1.1 Short presentation of Volvo 1

1.2 Background 1

1.3 Aim and objective 1

1.3.1 Delimitations 2

1.3.2 The production cell 2

1.4 Sustainable development 2

1.5 Methodology 4

1.6 Disposition 7

2 Frame of reference **9**

2.1 Sensors 9

2.2 Programmable logic controller 9

2.3 Robotics 10

2.4 Emulation 11

2.5 Simulation 11

2.6 Differences between emulation and simulation 12

2.7 Virtual Commissioning 13

3 Literature review **15**

3.1 Virtual Commissioning 15

3.2 PLC emulation and validation 16

3.3 Emulation and simulation 16

3.4 Virtual commissioning of a robot assembly cell 17

3.5 Conclusions 17

3.5.1 Summary 18

4 Market survey 19

4.1 Volvo's requirements 19

4.2 Software 19

4.2.1 Automation Builder 19

4.2.2 Emulate3D 20

4.2.3 Experior 20

4.2.4 Mechatronics Concept Designer 21

4.2.5 Simumatik3D 22

4.3 Market survey summary 22

5 Characterize the system 23

5.1 Sequence 23

5.2 Communication 23

5.3 Inputs and outputs 24

5.4 Layout 24

5.5 Assembly tool 24

5.6 Robot and PLC 25

5.7 Conveyor 25

5.8 Fixture 25

5.9 Pallet conveyor 26

6 Computer aided design 27

6.1 Converting and simplifying 3D models 27

6.2 Building process 27

6.2.1 Layout 28

6.2.2 Conveyors 28

6.2.3 Robot and robot stool ... 29
6.2.4 Fixture with movable parts ... 30
6.2.5 Magazine, light beacon and fence ... 32
7 Virtual environment ... 34
7.1 PLC inputs and outputs ... 34
7.2 Simumatik3D and PLCSIM ... 34
7.3 Simumatik3D and RobotStudio ... 34
7.4 HMI ... 35
8 Testing the virtual model ... 37
8.1 Experiments ... 37
8.1.1 Auto stop ... 37
8.1.2 Replicating robot error ... 37
8.1.3 Missing main cap ... 37
8.1.4 Adding new code - PLC ... 38
8.1.5 Adding new code – Robot ... 38
9 Evaluation of the virtual model ... 39
9.1 Appearance ... 39
9.2 Sequence ... 39
9.3 Simplifications and limitations ... 39
10 Discussions ... 41
11 Conclusions ... 44
11.1 Future work ... 45
11.2 Future work for Volvo ... 45
References ... 47
Appendices ... 49

Table of figures

Figure 1. The production cell G750. ... 2
Figure 2. The method model. ... 6
Figure 3. The disposition model. ... 8
Figure 4. Shows the position of the emulator in the process. Freely interpreted from Koninklijke Bibliotheek (2016). ... 11
Figure 5. Virtual Commissioning, freely interpreted from Reinhart & Wünsch (2007). ... 13
Figure 6. Shows the nodes in the PROFIBUS. ... 23
Figure 7. The robot used in G750. ... 25
Figure 8. Pallets for main caps. ... 26
Figure 9. Comparison of a 3D model before and after the simplification. ... 27
Figure 10. Structure tree of the emulation model. ... 28
Figure 11. The layout. ... 28
Figure 12. The conveyors with cover plate and sensors. ... 29
Figure 13. Robot in position. ... 29
Figure 14. Fixture with and without moving parts. ... 30
Figure 15. Separated moving parts. ... 30
Figure 16. Pink part with translational joint and blue part with rotational joint. ... 31
Figure 17. Movable parts added to the static object. ... 31
Figure 18. The fixture in open and closed positions. ... 32
Figure 19. Robot tool and fixture added. ... 32
Figure 20 The finished emulation model. ... 33
Figure 21. The signals between Simumatik3D and PLCSIM. ... 34
Figure 22. Signals between PLCSIM and RobotStudio. ... 35
Figure 23. HMI Comparison. ... 35
Figure 24. WinCC which simulates the Siemens HMI. ... 36
Figure 25. A simplification of the communication between the software. ... 36
Figure 26. The added test code. ... 38
Figure 27. The change in robot code. ... 38

Table of tables

Table 1. Reader recommendations. 7
Table 2. Summary of the literature review. 18
Table 3. A MoSCoW chart of Volvo's requirements. 19
Table 4. Summary of the market survey (higher number is better). 22
Table 5. Simplifications made in the emulation model. 40

Abbreviations and terminology

Bit	Bit is the basic unit used for handling information in a computer. A bit can only contain one of two values; 1 or 0 represents these values.
Byte	A byte contains 8 bits.
C#	A programming language used for constructing models in Experior.
Double Word	Double word contains 32bits.
FPS	Frames per second, a unit that measures display device performance.
HMI	Human machine interface, a user interface for a human to be able to interact with the machine.
I/O	Input and Outputs, inputs and outputs connects PLC, robot and other hardware such as sensors and conveyors.
MeshLab	A software that can be used for simplifying 3D models.
Node	A connection point in communication networks.
PLC	Programmable logic controller, a controller that stores instructions to control machines, conveyors and more, using inputs and outputs.
PROFIBUS	Process Field Bus – a standard used for communication in automation technology.
RFID	A technology to read information from a distance by transponders called tags.
RobotStudio	A software developed by ABB used for programming and simulation of ABB robots.
Word	A word contains 16 bits.

1 Introduction

This chapter introduces the reader to the background and the purpose. It also informs the reader about sustainable development and the methodology used.

1.1 Short presentation of Volvo

Volvo Group Truck Operations (GTO) is a world-leading manufacturer of trucks and heavy duty machinery. The production in Skövde consists of the engines (13 and 16 litres), camshaft, crankshaft, and cylinder heads. Volvo GTO Powertrain production has about 7300 workers in manufacturing and remanufacturing and about 2700 of them are stationed in Skövde. In the Skövde factory they do casting, processing of all the previously mentioned parts and the assembly of the engines.

1.2 Background

In order to continually improve Volvo's production and the way Volvo build their engines, they are continually updating and replacing their production equipment. This generates high costs in form of test runs and application tests of new production equipment, because of the long time it takes to test the equipment when it is installed. Since these tests often take place after the installation phase the whole test period often occurs during ordinary production. This often affects or stops the production resulting in even higher costs for Volvo. If something is wrong with either the PLC-programming or the Robot-programming this will result in the tests taking even longer time and generating even higher costs. Because of these costs, Volvo wants to investigate the possibility to test production equipment and it´s programming in a virtual environment with the help of emulation before implementation in ordinary production. The vision for Volvo is to use emulation in order to shorten the time to install new production equipment and by those means reduce these unnecessary costs.

1.3 Aim and objective

The aim is to investigate if there are any available emulation software on the market that can handle Volvo's complex production equipment that involves both robot and PLC-programming. If a suitable software is found it will be tested and evaluated, first in a virtual sense and then it will be subject of a live test in a real production cell consisting of both robot and PLC programmable units. The emulation tool has to behave in a representative way compared to the real production equipment when subject to the same tests and fail-provocation as the real production equipment. In order to get this result there are some questions that needs to be answered:

- Which are the most common emulation software available on the market today?
- Which of these emulation software suites the needs of Volvo?
- Can advanced industrial programming of PLC equipment and robots be emulated?
- Will an emulation model find and verify errors in the present production equipment's program code?
- Will an emulation model verify new production equipment's program code?
- Is it possible to implement the Siemens HMI with the emulation model?

The main objectives are:

- Present a minimum of three emulation software that suits the needs of Volvo.
- Choose one emulation software to be subject of a live experiment in Volvo´s real production equipment.
- Present a virtual model in the chosen emulation software that corresponds to real production equipment regarding programming, fail scenarios and movements.

1.3.1 Delimitations

Delimitations are the following:

- The emulation model presented will be a model of the production cell G750.
- Only communication within the G750 production cell will be handled.
- The emulation model presented will only handle programming and sequence of operations.

1.3.2 The production cell

The emulation model that will be created and emulated will be representative of an actual production cell called G750 that is located at Volvo GTO in Skövde. The production cell is an assembly cell consisting of an ABB IRB 6600 robot and the cell is controlled by a Siemens PLC. The robot disassembles main caps from the engine and then mount piston cooling nozzles into the engine block. A drawing of G750 can be seen in Figure 1.

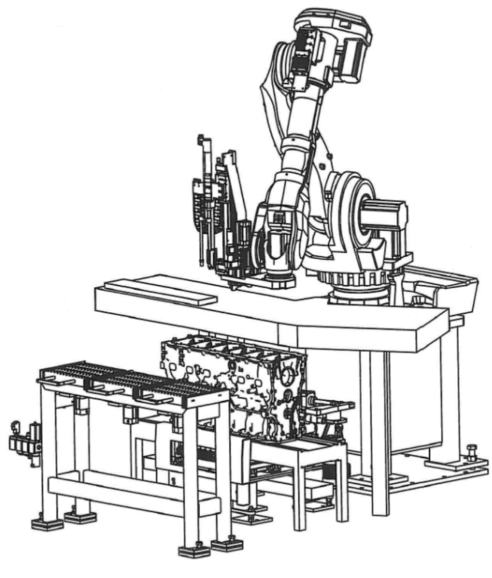


Figure 1. The production cell G750.

1.4 Sustainable development

Humans have always affected the environment surrounding them. One of the first known cases where humans have affected the ecology of an area can be traced back to 6000 BC when people in Mesopotamia built water canals in order to increase the yield from their crops. These canals made settling down in this area possible and when more and more people came to live in this area, more and more complex canals where built in order to meet the increasing demand for food. These new societies also needed resources in form of lumber, stone and metals which was taken from the surrounding areas. When the protection of the surrounding woodlands disappeared, more earth and sand where transported into the area by wind and weather. The sand blocked the canals which led to flooding's and when the water dried out, it left salts that was transported into the soil. The salted soil made the recently fruitful landscape barren, the people starved and were forced to move north to more sustainable landscapes, leaving behind a manmade desert. (Gröndahl & Svanström, 2011)

In the year 1987 the United Nations published a report named "Our Common Future" in which costs in economic terms where put in relations with the environmental problems. In this report sustainable development is defined as following:

"Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs."
(World Commission on Environment and Development, 1987, page 41)

This definition covers more than just the environment question, it also includes other aspects such as the social, economic, ethical and cultural questions (Dahlin, 2014). Dahlin (2014) also mentions that development can be defined as something that changes into the better and the word sustainable is closely related to lasting. Sustainable development first came from the environment problem and has today developed into many dimensions such as climate, social, cultural, economic, political and ethical questions (Gröndahl & Svanström, 2011). Although the definition of sustainable development is easy to understand, how to gain a sustainable development and know what is required to achieve it is much harder (Dahlin, 2014).

According to Dahlin (2014) there are three ways to go in order to solve the problems associated with achieving sustainable development:

- Change of technology
- Streamlining
- Reduced use

Change of technology means to replace a certain technique with another that have a less negative effect on the environment. Streamlining is when pollution and costs are reduced by small, but effective improvement work on the existing process, more effective processes often have a positive effect on both economy and ecology. Reduced use is simply to lessen the extent of a certain process by reducing the usage of it. This can for example be done by adding taxes or fines to the process. (Dahlin, 2014)

An emulation model will be built in order to investigate if it is possible to verify and validate PLC and robotic programs using an emulation software to see if the model is representative of the real production equipment and to examine how long time it takes to build a virtual model. If these objectives are met and the time it takes to build a virtual model is less than the time Volvo spends debugging and validating programs today. Volvo's vision is to use this technique in their commissioning projects in order to reduce the ramp-up time of equipment, the time spent at the supplier's factory for testing and validation and to work more efficient with improvements related to the equipment. If

Volvo manages to reduce the ramp-up time of equipment and the time spent at the supplier's factory, they will save a lot of time and money. This satisfies the needs related to both economic and ecologic development since the visits to suppliers for validation of programs and equipment often requires that Volvo sends people abroad using flight-travel which both affect economy and ecology in a negative way. Although these visits will not be eliminated using virtual validation, it will surely make the visits more effective and hopefully eliminate the need for a second visit.

When working with virtual models for verification the software runs on a personal computer which requires a lot less power than a production cell. Virtual models will also save time, money and energy which all will have a positive effect regarding sustainable development.

Companies that work with virtual models and virtual commissioning will work according to the three ways stated earlier in order to handle problems with sustainable development. They will work with new technology that have a less negative effect on the environment and a positive effect on the economy and they will work in a more efficient way that collaborates with the streamlining approach to handle problems regarding sustainable development. There will also be a less usage of energy with this technique that collaborates with the less usage approach.

1.5 Methodology

Methodology is according to Höst, et al. (2006) the way that the students work with theses. It is the framework of how the work will be done. The method that will be used is dependent on what type of work that will be done and depending on the work an appropriate method or combination of methods needs to be chosen. Based on the chosen method or methods, a concrete plan for the work that needs to be done has to be created. The plan can, for example, consider how to handle data collection, observations and interviews. Literature studies needs to be performed in all theses, but in some cases it can also contribute to the data collection. Data that is gathered in the work with a thesis can be either quantitative or qualitative, or both. Quantitative is data that can be counted or classified, qualitative data is such that is describing and nuanced, which makes it a very detailed form of data collection (Höst, et al., 2006).

A case study was performed by Guerrero, et al. (2014) and was focused on building a virtual model of a small pick-and place equipment that was used for studies at their university. In that report Guerrero, et al. (2014) describes an approach for building virtual models of existing equipment. A methodology that is based on Guerrero, et al. (2014) with a minor change in the last step will be used. The steps that will be used are the following:

- Characterizing the system

This is where an understanding of the system that is to be modelled needs to be achieved. That means that all the inputs and outputs needs to be collected, the flow of the production cell needs to be studied, all the different components of the cell need to be known,3D models over equipment needs to be collected and all the equipment's placements and their relations to each other needs to be documented.

- Computer aided design

In this step the virtual model is to be built. All the 3D models are to be implemented in the software and the cell is to be built in a way that makes the virtual model representative of the real production cell.

- Virtual environment

The different software used will be connected to each other and the PLC- program and the robotic program will be implemented in the software.

- Testing the emulation model

During this step the complete programming will be tested in the emulation model and checked for errors that may occur either in the programs, or in the software. This is also where experiments will be presented.

- Evaluation of the emulation model

This step is done by the end of the project and will evaluate the virtual model, how much time it took to build, if it is representative to the real system and what could and could not be done in the software used. This step differs from what Guerrero, et al. (2014) used in their study, in their study this step was "Virtual model as a monitoring system" instead.

The work that will be performed will follow the method model that can be seen in Figure 2. It shows how the authors first will build a baseline with the purpose, which is represented by emulation software and virtual model in the figure. When the baseline is set a knowledge about the subject in form of theory and case studies will be built. The project will then go into a practical phase where the steps mentioned above will be implemented. The last step is the project ending where evaluations and conclusions will be drawn from the project. In the practical work there is also quantitative and qualitative data collection in the form of observations, discussions, collaborations, documentation and more. The reason for market survey to be present in both forms of data collection is that in the market survey there is a mixture of interviews, documentation and observations.

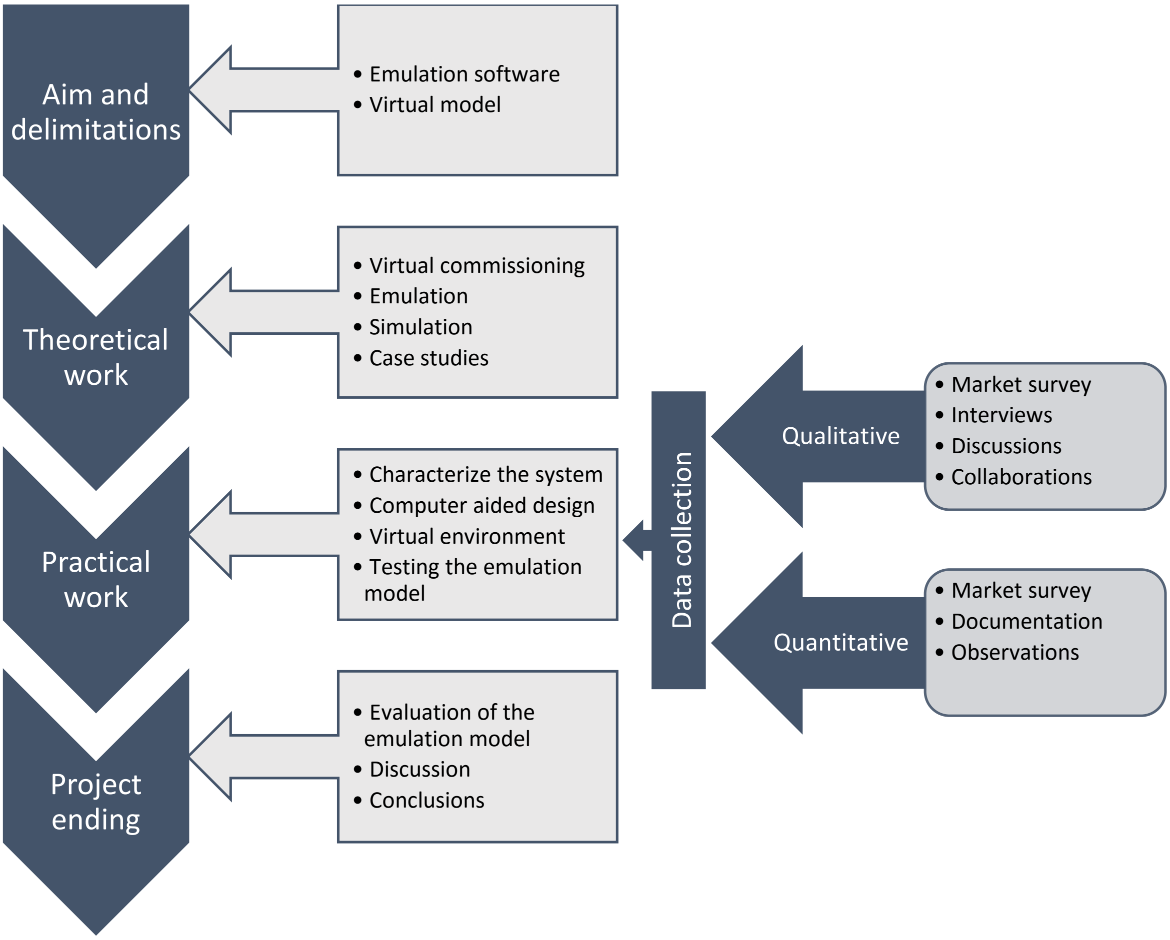


Figure 2. The method model.

1.6 Disposition

The disposition of the report gives an understanding for the reader of how the report is built and also a good overview of what the report will include as can be seen in Table 1.

Table 1. Reader recommendations.

Chapter	Description	Reader recommendation
1. Introduction	Describe the background, what the goals are and delimitations of the thesis.	All readers.
2. Frame of reference	Gives the reader a theoretical understanding of the subjects handled in the report.	All readers.
3. Literature review	Describes similar reports with similar goals.	All readers.
4. Market survey	A survey for software that would suit Volvo's needs.	Readers with a big interest in available software for simulating and emulating production.
5. Characterize the system	Describes the production cell and its characteristics.	Readers who wants to learn more about the production cell.
6. Computer aided design	How the virtual model was build and what was needed to build it.	Readers with an interest about using the software.
7. Virtual environment	The connections between the different software are setup and described in this chapter.	Readers with an interest in how the software communicate.
8. Testing the emulation model	Some tests and experiments have been conducted on the real production cell and the virtual model.	All readers with an interest in the emulation models performance.
9. Evaluation of the emulation model	Evaluating the results of the virtual model.	Readers with an interest in the results and has a knowledge about the subject.
10. Discussion	Discussions and future work.	All readers.
11. Conclusion	Conclusions drawn from the project.	All readers.

Figure 3 visualizes the relations between the different chapters in the report. The first chapter introduces the reader to the aim and objectives, which also defines the following chapters two to nine. Chapter two to three introduces the reader to the field with a theoretical references and what others have achieved with similar methods. In chapter four a market survey on different emulation and simulation software has been done to be able to choose which one to use to build a virtual model. In chapter five to eight the model is built. The ninth chapter evaluates and compares the virtual model to the real production cell. The last two chapters summarizes the work done in the report, conclusions are drawn and some discussion is held.

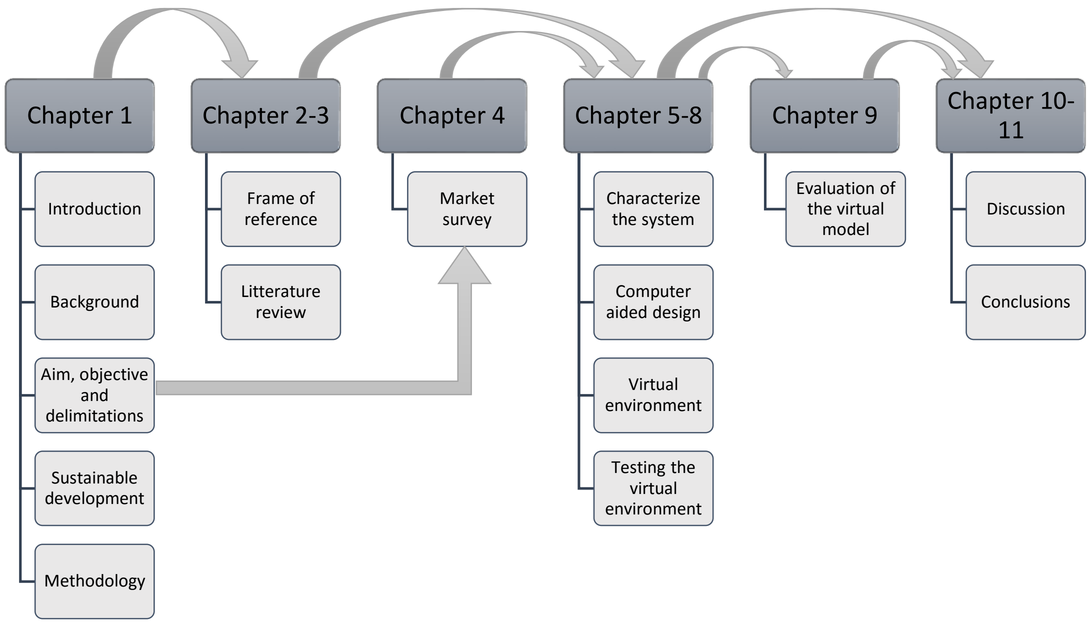


Figure 3. The disposition model.

2 Frame of reference

This section gives a theoretical understanding of the main subjects, virtual commissioning, simulation, emulation, PLC, robotics and sensors.

2.1 Sensors

A sensor is a device that converts a value, like position or magnetic field, to a more suitable format such as a voltage. This voltage can then be used in an analogue to digital converter and then be used in a digital computer. (Groover, 2015)

When building a virtual model in the emulation program it is necessary to add sensors to the program manually in order to get the software to work properly.

In order to control a production cell and the components within that cell, the surrounding equipment and conveyors, sensors are needed to be able to detect changes within the system. According to Groover (2015) there are many types of sensors available for detecting different types of physical values, they are:

- Mechanical: Position, velocity, pressure and mass.
- Electrical: Voltage, current and resistance.
- Thermal: Temperature.
- Radiation: Type of radiation.
- Magnetic: Magnetic fields.
- Chemical: Concentration, PH levels.

2.2 Programmable logic controller

Before the Programmable Logic Controller (PLC) was introduced in the 1970s the controls to a system was hard-wired and used different relays, counters, timers, coils and similar components to control the system. After the introduction of the PLC, older systems were retrofitted to function with the PLC instead and this resulted in the system being much more reliable and increased the production to more than it was capable of compared to when the system was new. (Groover, 2008)

Groover (2008) defines a PLC as a micro-computer that can control several functions in a system. The functions can, for example, include timers, counters, sequences and more. It can also handle signals through digital and analogue input and output (I/O) modules. The functions of the PLC are defined through instructions coded to the memory. A PLC can be used in many different industries and production cells and can control anything from a small conveyor to entire automated storage systems and machine cells. (Groover, 2008)

A PLC consists of a central processing unit, memory and an I/O card. The I/O card is used for sensors, valves, pumps and motors. The system is often build with different modules to be very flexible depending on the complexity of the system. The central processing unit reads the inputs from sensors using the I/O-card and then executes instruction after instruction according to the code loaded in the memory of the PLC. After executing the code, the outputs are calculated and set according to the code before the inputs are read once again. This type of cyclic behaviour is one of the PLCs characteristics. The time it takes to do the I/O readings is determined by how much code there is and how powerful the components of the PLC are. (NE, 2016)

Using a PLC has many advantages according to Groover (2008) the most significant advantages include:

- Easier to program the PLC than wiring the relay control panel.
- Possible to reprogram the PLC instead of rewiring the old system whereas the old systems where often scrapped all together instead.
- The physical size of the PLC is smaller than relay control panels.
- Easier to connect a PLC to a computer than it is for a relay.
- Maintenance of the PLC is easier and the PLC is also more reliable than older systems.
- Greater variability of control functions with the PLC than with relay controls.

2.3 Robotics

If the term robot is mentioned it is a rather broad statement, the definition of robot is according to Bolmsjö (2006) the following:

"Industrial robot or robot is an automatically controlled, reprogrammable universal manipulator programmable in three or more axes, which can be either fixed or mobile for use in industrial automation"

The term programmable in three or more axes is a rather low set goal for an industrial robot, since they are commonly programmable in four or more axes. According to Bolmsjö (2006) the use of industrial robots can be divided in to three main areas and can be described as the following:

- Material handling

In the terms of material handling robots are used for transportation of material or object without the robot performing any processing of the material or object.

- Process operations

Process operations robots are directly involved in the processing, with the aid of an external tool or other outer equipment. Examples of process operations where robots are commonly used is for arc welding, spot welding and spray painting.

- Assembly

Assembly means the composition of components who in different operation stages forms a finished product. A major difference compared to process and material handling applications is that assembly often comes late in the processing chain and therefore the demands on the robot's robustness and reliability often are higher in the assembly application than the others.

2.4 Emulation

Emulation means to imitate, more specific in computer science this means that when the same data and programs are used the same results will be presented in both the real system as well as in the imitation (NE, 2016). This is also claimed by Koninklijke Bibliotheek (2016) who describes emulation as imitating a computer or program platform on another platform. This makes it possible to execute programs on a platform that is not designed to do so originally. The emulator creates a layer between the target, which is to be imitated and the host, who is the imitation and by that means enable compatibility between the two as shown in Figure 4 (Koninklijke Bibliotheek, 2016).

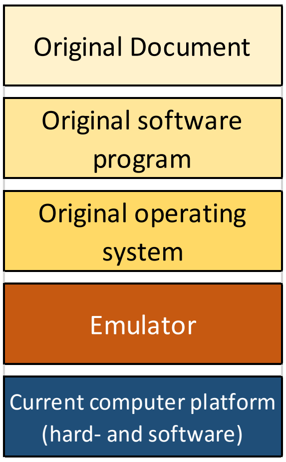


Figure 4. Shows the position of the emulator in the process. Freely interpreted from Koninklijke Bibliotheek (2016).

When using real programming in emulation in order for validation, the system will respond similar to how the real system would do when any input, output or failure event is given. Emulation are mainly a tool for verification of equipment in a virtual environment where programs can be tested, failures can be foreseen and where experiments of equipment can be studied. (Oppelt & Urbas, 2014)

If the emulation model is properly built and is replicating the real life equipment, the model can potentially be used for training operators, as well as how to handle the equipment (Oppelt & Urbas, 2014; McGregor, 2002). Emulation can also study the effects of adding or changing equipment and to experiment with changing sequences, flow and other things to improve the equipment as early as in the design phase (Oppelt & Urbas, 2014).

According to McGregor (2012) there are some scenarios when emulation is particularly useful and economically justifiable:

- When testing is due to be carried out on the critical path of an object
- When full testing before start up is not available
- When the cost of emulated testing is less than the cost of real testing

2.5 Simulation

The easiest way to describe simulation is that it is a replica of a real world system or process over time. A system can be defined as a collection of objects, machines or people that are working together towards a logical end (Law, 2014). The main objective with simulation is always to replicate reality in order to study it, to confirm theories and to answer questions regarding changes to the system. (Banks , et al., 2009)

Simulation can be used if a company is in the process of changing an existing system, updating it with new equipment or want to make another major change to the process or the flow of the process. A simulation model can be a great tool for detecting faults in the system or process already in the design stage and by that means save the company a great deal of time and money. In these scenarios simulation is a perfect tool, but there are some cases when simulation is less appropriate. One of these cases are when the cost of creating the simulation model exceeds the costs that would be saved using the simulation model. There is also the question of time to take into consideration. Sometimes creating a simulation model can be very time consuming, therefore this must be taken into consideration to make sure that the model is finished in due time. Simulation also requires a lot of data in order to be a valid replica of the real world system. If the data does not exist, then simulation is not an appropriate tool to use. (Banks , et al., 2009)

In simulation, two different types of methods are commonly used. The first method is called discrete-event simulation and the other is called continuous simulation. Discrete-event simulation is a form of simulation where the data is not collected at a constant rate, but instead collected when different events, either in time or in the system itself occurs. These events can be, in the case of a production simulation, when there is a change in product, when a product is ready for delivery or when a machine failure occurs. When these events take place it can be described as the system takes a picture of the system and present it to the user. This form of simulation makes the simulation model run smoother because less data is handled. The system only collects data based on events and does not a process a continuous flow of data. This also makes the simulation work faster and require less performance from the computers used. (Banks , et al., 2009)

When continuous simulation is used it collects data all through the entire simulation. This is a form of simulation that can be used when the state of the simulation model needs to be able to be studied at any given point of the simulation. For example, when simulating the upcoming weather or when simulating a body of water to study how it behaves over a period of time. This is a form of simulation that gives the user the possibility to validate theories at any given point in the simulation, but it is a form of simulation that requires a lot of performance from the computers used. (Banks , et al., 2009)

2.6 Differences between emulation and simulation

Simulation and emulation is based on the need to imitate real world systems in order to study them. There are some major differences in the approach to the imitation and the usage of these studies. Simulation uses the same outputs as a real world system in order to find bottlenecks, study plant utilization among other things. In simulation it is not necessary for the processes in the model to behave exactly as the real world system, it is the output that is of the most interest (Erlandsson & Rahaman, 2013). Another way to describe the aim of simulation is that it is used to determine and try different solutions in order to find the best solution. Simulation allows the user to demonstrate and validate functionalities in an easy and cost efficient way that shows results clearly. Emulation on the other hand, is used for more precise operations when executing and verifying real programs is the main goal. Another functionality of emulation is that it can be of use when training operators in a risk-free environment. (McGregor, 2002).

Emulation is also used to study a system, but the difference in the approach compared to simulation is that in emulation the processes behaves similar to the real processes in real life. When using emulation, the real PLC- or Robotic code is used in the emulation model and by that means the model, if built and handled properly will behave as a real life system would. (Danielsson, et al., 2003; McGregor, 2002)

2.7 Virtual Commissioning

The main objective of Virtual Commissioning (VC) is to lessen the time spent on debugging systems and programs in manufacturing equipment using simulation and emulation for verification before the equipment is implemented in the real production (Hoffman, et al., 2010). VC is the manufacturers way of improving and validating production equipment in a virtual environment. This leads to less costs and a more adaptable production. The positive effects of VC can for instance be that the software has a better quality due to more intense testing. The equipment can also be tested in scenarios that would not have been possible in real equipment due to danger and equipment damage (Reinhart & Wünsch, 2007). Furthermore, using VC will make it possible for contractors to discuss and to show the functionalities to the customer in a virtual setting (Stephan, et al., 2012).

Software that is able to handle these rather complicated models often requires a high level of knowledge because building the models is often proven to be quite complicated (Hoffman, et al., 2010). This is also claimed by Lee & Park (2014) who talks about how VC enables the user to validate systems at a programmable level and therefore the virtual model can be used to replicate the real equipment down to a level of sensors and actuators. Due to these factors VC have often been unreachable for smaller companies that lack experience and resources in simulation based environments (Hoffman, et al., 2010).

The profit of using VC instead of regular commissioning is mainly the time-saving aspect. Therefore, in order to have a successful VC project the modelling time of the virtual model cannot be longer than regular commissioning because then the profit with using VC is lost (Reinhart & Wünsch, 2007). The reason that the main positive effect of VC is saving time is due to the multitasking that is required in the beginning of a VC project will lead to an increase in work effort. How VC would be executed is presented in Figure 5 that clearly states that times is saved, but also that the work will be increased in the beginning of a VC project. (Reinhart & Wünsch, 2007)

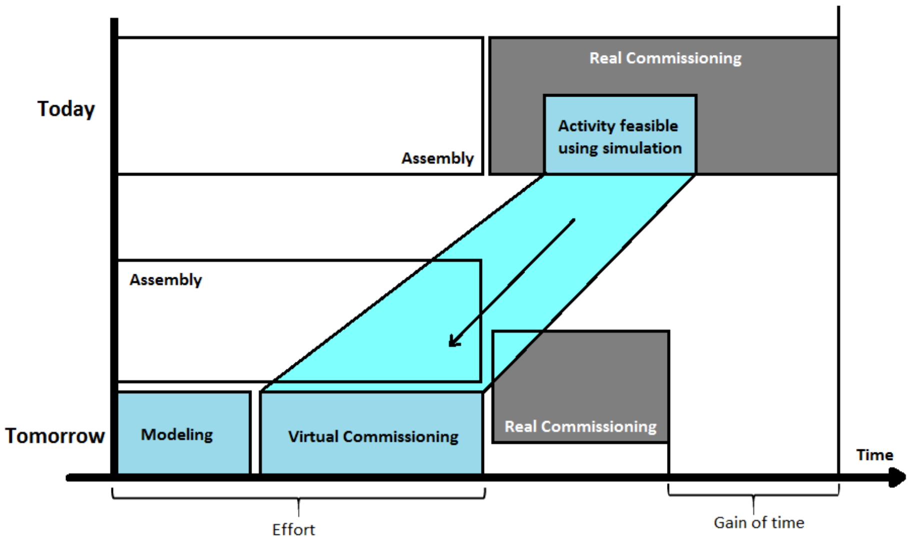


Figure 5. Virtual Commissioning, freely interpreted from Reinhart & Wünsch (2007).

When using VC for avoiding mechanical errors such as collisions it is sufficient with a 3D model with specified mechanical movement and behaviour. When using VC for verification of control programs a simulation of the manufacturing system mechanics at I/O level will be required. To test both mechanical movements and the control programs of a system a simulation that can handle all the movements and programs with sensors and actuators included will be needed. (Hoffman, et al., 2010)

VC is not limited to only mechanical and system verification in an early stage, but also includes process simulation, material flow handling and ergonomics evaluation. (Hoffman, et al., 2010; Siemens, 2016)

3 Literature review

In this section conference papers, reports and theses have been studied to gain knowledge about virtual commissioning, PLC and robot verification. This chapter ends with an analysis and a summary.

3.1 Virtual Commissioning

The major advantage of VC is that, if all the virtual models and simulations are properly built and replicating real life equipment, the adaptability of the company's production increases, the time to implement production equipment is shorten and a more stable and reliable project form is introduced in the company. However, some issues need to be handled in order to design a good virtual model to work with.

One problem with VC is that it requires a lot of the virtual model, it has to be able to be adaptable to the future needs of the company and has to be representative of the real system. To be representative to the real system it is crucial that all the moving parts in the equipment has the proper kinematics. Today these kinematics is programmed and applied to each and every moving part. This is very time consuming and a major drawback to the emulation tools available on the market today. In a study by Lee & Park (2014) the problem with the kinematics programming is presented, the study shows that the programming of kinematics is the major issue that needs to be handled in order to work with VC in a profitable way.

This was also the case when Guerrero et al. (2014) was building a virtual model of a production cell using Process Simulate by Siemens. Guerrero, et al. (2014) had to define the parts in the cell into Dynamic parts, or into static parts and then give each part proper physical attributes. Guerrero, et al. (2014) built a virtual model of an existing pick-and place station and then verified the PLC-program which was built in Siemens Step 7. The experiment was successful and Guerrero, et al (2014) could verify their PLC programming in a virtual environment using Process Simulate and then monitor the existing system with their virtual model. Guerrero, et al. (2014) followed five steps when performing this experiment:

- Characterizing the system.
- Computer aided design.
- Virtual environments.
- Testing the virtual environments.
- Virtual environments as a monitoring system.

A benefit of VC is reduced time spent on programming and more reliable programming. Reinhart & Wünsch (2007) describes an experiment to draw conclusions regarding VC and if time can be reduced in VC projects compared to regular projects regarding PLC programming. In this experiment 30 people built a PLC program for a machine using standard tools for PLC programming and another group of 30 people used a simulation model to perform a VC of their programs before the real implementation. The results of this study showed that commissioning time was reduced by 75% when VC was used and that the fulfilment of requirements was 84% in the group that used the VC, compared to the group that did not use VC that only reached 37% fulfilment of requirements. This experiment clearly shows a benefit in time and quality when using VC, but this experiment was conducted with a rather easy setup and therefore may not be applicable to larger projects (Reinhart & Wünsch, 2007).

3.2 PLC emulation and validation

Several studies in PLC validation using emulation has been conducted. One of these studies were performed by Erlandsson & Rahman (2013) who had the aim of building a virtual model of a Tetra Pak filling machine using Experior and verify PLC-code in hardware in loop simulation. In order to build the model in Experior they used 3D-CAD files of the filling machine and converted them into a format that was compatible with the software. When they were converting the 3D-CAD files, they also had to separate the movable objects and assign them with either motion or static properties. When this was done, they could use these files in Experior and use the “drag and drop” function to assign the details to their rightful places. As they were building the model they did some simplifications to ease the building of the model. One of these simplifications were using motors in the virtual model instead of conveyor belts with hangers as in reality. They used the ActionPoint command in Experior to create imaginary sensors and place them in the virtual model. They also established communication via an ethernet cable between the PLC and PC with the virtual model. The PLC-code was written in RSLogix 5000 and ten inputs and six outputs were used. When they were done they had managed to establish communication between Experior and an Allan Bradley PLC and verify the PLC code, however they had problems with the connection that in some cases were unstable due to the plugins created by Experior to communicate with RSLogix 5000 and Allan Bradley PLC.

To verify PLC programs using simulation based tools Dznic & Yao (2013) built a virtual model of an existing production cell using Experior as a simulation tool. To be able to replicate reality as much as possible they used SketchUp to create their own 3D objects, this because Experiors own library was not sufficient to create the equipment as close to reality as was needed. To create the PLC program, they used Siemens TIA Portal. In order to create a working communication between the simulation software and the PLC a third party software had to be used as a link between the two. This software was NetToPLCSIM. Dznic & Yao (2013) managed to build the model of the production cell using this software and they were able to verify the PLC code. One of the drawbacks with this study was that when 3D object was imported into Experior the objects behaved as solid blocks. When an object had a hole in it then sensors could not detect trough these holes, which lead to the sensors had to be rearranged in order to work properly and therefore the virtual model did not replicate the actual production cell in fully.

3.3 Emulation and simulation

The concept of verification of PLC logic in a virtual environment was tested by Johansson & Nilsson (2015). The programs that Johansson & Nilsson (2015) used was Plant Simulation and Simumatik3D. They built a conceptual model in both software and then compared the results. Johansson & Nilsson (2015) found limitations with both software. Plant simulation were missing functionalities for simulation of certain objects, such as cylinders and sensors. Simumatik3D where under development and had limitations when a larger system was implemented. However, their results show that Simumatik3D where the preferred software for PLC verification and Plant Simulation was more suited for creating more realistic simulation models. Plant Simulation could be used for PLC verification, but not at the same detailed level. (Johansson & Nilsson, 2015)

3.4 Virtual commissioning of a robot assembly cell

In a case study performed by Makris et al. (2012) a production cell consisting of two cooperating assembly robots where used in a VC project. This study had several challenges, the cooperation of two robots adds complexity to the VC project and the elimination of the PLC as the master in the control hierarchy is two examples of the challenges. The robots that were used in the study were two Comau Smart NJ 130 robots. Makris et al. (2012) chose to work with the InVision and WinMOD software to achieve a real-time HIL simulation. Makris et al. (2012) managed to build a representative model of the actual assembly cell using the two software. They managed to validate the entire PLC programs and robotic programs. The cycle time in the virtual model were almost identical to the real cycle time in the assembly cell. One problem they faced during the study where that all the I/O signals had to be manually defined by the programmer which were a very time-consuming task. In the beginning of their project Makris et al. (2012) mention what type of data that is required for a successful VC project and they are the following:

- 3D simulation models of all the equipment that is to be commissioned, including kinematics, electrics and controller program.
- Detailed layout of the production cell with exact placement of resources and relevant equipment.
- Material flow, involving sequence of operations.
- Control systems, either the actual PLC or the emulation software can be used for validation of the virtual prototype.
- Detailed definition of the control system´s I/O signals and their respective mapping.
- Details about extra functionalities such as safety systems.
- IT structure and communication protocols for the networking between the control system and the simulation model.

3.5 Conclusions

In the literature review several interesting findings were found. When a virtual model is to be built, Xcelgos Experior is a software that could possibly be used and therefore it is of great interest that some of the limitations of the software is known in advance. For example, when implementing new 3D models with Experior as was done by Dzinic & Yao (2013) the models may behave as solid blocks is an issue that may cause problems in the building process. In the same study, it was mentioned that they needed a third party software called NetToPLCSIM for communication. In the study performed by Erlandsson & Rahaman (2013) it is mentioned that there is a command in Experior that is called ActionPoint that is used for creating own sensors. In the same study they mentioned that the connection between the software´s sometimes could be unstable, this is also an issue that is good to know in advance. The literature review has also been helpful for establishing a methodology. The methodology that Guerrero et al. (2014) used was applicable to emulation projects where the object that is to be emulated already existed.

When working with emulation which is an important tool in the VC area, there is a need to build a knowledge about VC, the literature review has a focus on this area and some conclusions can be drawn. Reinhart & Wunsch (2007) presents several benefits related to VC, one of which is that the time spent in commissioning can be reduced, the programming will be more stable due to more intensive testing and opens up opportunities to test scenarios that would not be possible in real life.

What is interesting is the drawbacks and dangers working with VC, the reason is that 3D models will be needed in order to replicate G750. Dzinic & Yao (2013), Lee & Park (2014) and Guerrero et al. (2014) all mentions problems implementation of own 3D models and in several of these case studies there is also mentioned that problems occurred with communication between different software. Therefore, it will be crucial to use these case studies and learn from how they solved the problems.

3.5.1 Summary

A summary of the literature review can be seen in Table 2. Conclusions that can be drawn from this literature review are the following:

- Verification of PLC programs using emulation tools are possible.
- Verification of Robotic programming using emulation tools is possible, but not as common.
- Problems when implementing own 3D models are common.
- Adding kinematics to models are time consuming.
- Virtual Commissioning saves time and money in commissioning projects.
- It is important to characterize the system before building the virtual model.
- Communication problems between software´s is common.

Table 2. Summary of the literature review.

Author	**Research area**	**Software used**	**Goal with study**
Erlandsson & Rahaman (2013)	Testing and verifying PLC code with a virtual model	Experior Xcelgo	Build a virtual model and verify PLC-code in hardware in loop simulation
Dzinic & Yao (2013)	Testing and verifying PLC code with a virtual model	Experior Xcelgo, SketchUp, Siemens TIA-Portal	Evaluate the possibility to verify PLC programs by setting up a Virtual Commissioning project
Guerrero et al. (2014)	Testing and verifying PLC code with a virtual model	Process Simulate, Siemens Step 7, NX software	Create and implement virtual environment.
Johansson & Nilsson (2015)	Virtual Commissioning	Plant Simulation, Simumatik3D, Siemens Step7	Verifying of PLC logic in a simulation software
Makris et al. (2012)	Virtual Commissioning	Invision, Winmod	Building a virtual robot cell for offline programming
Reinhart & Wünsch (2007)	Virtual Commissioning	n/a	Economical application of Virtual Commissioning

4 Market survey

A market survey of the existing emulation software that is available on the market today has been conducted. The market survey has been done through interviews with representatives of the companies, studies of documentation, case studies and video demos of the different software. The software presented are the ones where proper documentation, video footage and case studies where available. The companies where an interview could be conducted or where another form of communication could be established are also presented. Note that some of the software found are not included due to absence of references, information and documentation.

4.1 Volvo's requirements

There are several different issues that need to be addressed and several objectives that a software for emulation has to meet in order to work in Volvo's often complex production environment.

MoSCoW is a method commonly used for setting up requirements in projects, MoSCoW is an abbreviation of, Must, Should, Could, Won't. The "Must" is requirements that is obligatory for the project to contain. The "Should" is not obligatory, but good if it was included in the project. The "could" is not a mandatory but it could be included if the time and resources is available. "Won't" is parts that not should be included in the project at all. (Tonnquist, 2014)

To be able to identify different software that meets Volvo's requirements and objectives the MoSCoW method was used and is presented in Table 3.

Table 3. A MoSCoW chart of Volvo's requirements.

Market survey specification	**Must**	**Should**	**Could**	**Wouldn´t**
User friendly	x			
A short learning time to gain basic knowledge about software < 1week		x		
A short learning time to gain basic knowledge about software < 2 weeks	x			
Support	x			
Give a 3D overview that is equivalent to the real machine cell	x			
Time to build < 1 week		x		
Time to build < 2 weeks	x			
Time to build < 3 weeks	x			
Be able to work with both robot-and PLC programmeble units	x			
Be able to work with different brands of equipment (ABB-Siemens, and others)	x			
Possible to update virtual models and emulation over time	x			
Possible to implement own 3D models	x			

4.2 Software

The different software´s found during the market survey is presented below. With a short presentation of the software and the different advantages and disadvantages.

4.2.1 Automation Builder

Automation builder is a software from ABB which can be used for validating both PLC-programmable units, ABB -robots and their programs. The ABB Automation Builder have Codesys integrated in their software and this is where the PLC-programs is incorporated in the emulation software. The Automation Builder also uses ABBs software RobotStudio which is a well-known software for programming ABB´s robots both online and offline. It can simulate the robot movements accordingly to the rapid programming, which makes it a powerful tool when programming ABB robots offline. It provides the user with a 3D overview over the robot cell and the robot behaves exactly as it would in a real production environment. There is an available library including simple conveyors, tools and more, but it is also possible to import your own 3D models. When importing your own 3D models of equipment, kinematics has to be added to sort out the moving objects from the static objects.

Automation builder have the robot programming in focus, compared to the other emulation software on the market that have more focus on PLC-emulation. Automation Builder also uses Codesys which is a program for PLC-programming and it is commonly used by many different companies and University's when it comes to PLC-programming. The major downside of Automation Builder is that ABB has chosen not to have an open software that can handle different brands of PLC. They have excluded the possibility to implement Siemens PLC´s for the benefit of their own PLC´s.

4.2.2 Emulate3D

Emulate3D is a Mitsubishi e-factory alliance partner based in the United States of America who offers a variety of products in the simulation/emulation field. The primary usage of Emulate3D is emulation of material handling systems. Their product for emulation called Emulate3D has been used by the Swedish mail service "Posten AB" to emulate and build 3D models of their sorting centrals. It has also been used by Carter Control system who builds different types of conveyor and sorting systems in order to show the finished product to customers and to test the PLC codes. Both these clients claim that using Emulate3D has helped them save both time and money and that Emulate3D was an easy tool to use. They also claim that they got the support they needed when problems occurred. In Emulate3D it is possible to build a personal library with your own 3D models, or use Emulate3Ds library. In Emulate3D it is possible to import robots and they do have a library consisting of both ABB and KUKA- robots. However, it does not execute the robots program in order to emulate the robot movements. In order to simulate the movements, Emulate3D have their own system where teach and logical points is set to the robot. The program is compatible with a lot of different PLC´s, including Siemens and it is able to connect several PLC´s at the same time. In order for the user to gain a basic knowledge about the software, Emulate3D provides a five-hour introduction course. Support is provided in English or German through e-mail or through online meetings.

4.2.3 Experior

Xcelgo is a company based in Denmark, which provides a virtual automation software and consulting service for 3D modelling. Xcelgos software Experior is able to handle 3D graphics, physics simulation and several different PLC controls and Robots, including ABB robots. Experior has a library containing some basic conveyors, lamps, sensors, mechanical pushers and more. The software also has the ability to import CAD files to make the process of building a machine cell less difficult. Although when importing a CAD file, it may be needed to add some dynamic to the rigid parts of the model to make it able to move. It is also possible to design self-made tools and other models to make the production cell more accurate to the reality.

Experior is according to their CEO, Aksel Jørgensen, very user friendly compared to older solutions like WinMod, AutoMod and Plant simulation. Xcelgo offers support for their users, from web sessions and remote support to introduction training and onsite support. For a C# developer to learn to develop customized objects would take a few days. To build an average equipped machine cell could take up to a week depending on availability of the documentation, structure of I/O-lists and the need to develop new non-standard objects for the software.

The software can emulate multiple PLCs simultaneously and of different brands if needed, as well as running ABB robots simultaneous with the PLCs. Many of the bigger brands of PLCs is supported such as Siemens, Beckhoff and Allan Bradley. Experior is also planned to support more brands in the future.

When done building a virtual model it is fully possible to update the model over time if changes are to be made in the real world machine cell. To make the building process easier it is possible to import 3D CAD files, preferably in the formats COLLADA, STEP, 3DS and WRML.

4.2.4 Mechatronics Concept Designer

Siemens provides a software called Mechatronics Concept Designer (MCD). MCD is a specialized software to create virtual environments, to test PLC programs and to develop offline programs for manufacturing applications. It is a tool used by engineers from the design phase to the implementation phase. MCD can build 3D models where the kinematic behaviours of the different parts can be added. It uses Simatic to handle the communication between the PLC and MCD. When using these tools, it is possible to verify PLC programs in a virtual environment.

The main usage frame for this type of software seems to be in the machine-building segment of industries, that is where this tool would be extremely helpful to gain more a collaborating environment between design and the electrical and automation engineers. This collaboration would help them avoid mistakes due to faulty design or programming mistakes that frequently occur in this type of industries. This according to presentations from Siemens, the information on their webpage about Mechatronics Concept Designer and from tutorial videos found online. Their main commercial, that can be found on their webpage, also focuses on the machine builders. It is possible to implement and to validate PLC-programs, but that is not the main purpose of this software, it is more of a feature. In their brochure it says:

"Mechatronics Concept Designer from Siemens PLM Software is specifically designed to speed up the concept design for machine tools" (Siemens, 2016)

To build an emulation using MCD appears time-consuming, and a lot of work would be needed in order to build a multitasking machine cell consisting of pneumatically, mechanically and robot controlled equipment. This conclusion is drawn upon the fact that each and every movable object or functional object such as sensors, pistons and grippers would have to be given kinematic behaviour, these behaviours are for example: which way to move? How long? And at what speed? To add all these elements to all the objects in an entire robotic cell would be time consuming if validation is the main objective. Industries that are building the machines have great use of this program because it is easier to add all this kinematics in the design phase, then they are already in place when the validation phase start and therefore they would save a lot of time. This conclusion is also drawn upon the fact that all the additional resources on their webpage where they present studies and benchmarking all consist of machine-building companies.

4.2.5 Simumatik3D

Simumatik3D is an educational tool used at the University of Skövde. It is mainly used for learning and validation of PLC-programs. It has many advantages, support for many different brands of PLCs such as Siemens S7, virtual PLC´s and more. Simumatik3D has a library with different objects such as conveyors, lamps, fences, sensors and cylinders that can be implemented in the model. It is also possible to import 3D files. Simumatik3D also has the ability to build specific physics for the imported models. When using these objects there is also the ability to change the dimension, rotation and orientation of the object. It is also possible for Simumatik3D to import and export collections of objects, for example a specific tool with a specific joint or an advanced conveyor system.

The placement of objects in Simumatik3D is relatively easy using a coordinate system where the object coordinates is set and then the object is placed accordingly. Simumatik3D has the ability to import robots from the ABB library and then mirror the movements done by the robot in RobotStudio to Simumatik3D, which makes the actual Rapid programming present in the emulation model.

The major advantage of Simumatik3D is that is an easy tool to use for emulation of both PLC-and robot programmable units. It is open for different brands of PLC´s and the connection between Simumatik3D and the PLC is easy to set up. Using RobotStudio for the robots in Simumatik3D is also an advantage because the model will mirror the actual movements done by the robot which makes the robots trajectory present in the model and therefore collisions will be easy to detect in the emulation model. Simumatik3D's disadvantages is that it is a software under development and therefore minor problems sometimes occur.

4.3 Market survey summary

The conclusion of the market survey is that only two of the software are able to handle all of Volvo's requirements. An estimation has been done through interviews, discussions with representatives of the companies and watching video demos to investigate what and how good the software could handle the requirements. In Table 4 the results of the market survey can be found. A decision is made to use Simumatik3D to create and emulate the virtual model, since it scored the highest in the survey.

Table 4. Summary of the market survey (higher number is better).

Market survey specification	Experior	Simumatik3D	Automation Builder	MCD	Emulate3D
User friendly	3	4	3	2	3
A short learning time to gain basic knowledge about software < 1week	4	5	3	2	4
Avalible support	4	3	4	4	4
Give a 3D overwiew that is equivalent to the real machine cell	4	4	4	4	4
Time to build < 1 week	4	4	4	2	4
Be able to work with both robot-and PLC programmeble units	5	5	5	5	1
Be able to work with different brands of equipment (ABB-Siemens, and others)	4	4	1	3	1
Possible to update virtual models and emulation over time	5	5	5	5	5
Possible to import 3D models (CAD)	4	4	4	4	4
Totalt:	**37**	**38**	**33**	**31**	**30**

5 Characterize the system

In this chapter the production cell G750 and the equipment within the cell is presented to the reader.

5.1 *Sequence*

In order to characterize G750 the first thing that needed to be studied was the function and the sequence of operations in the robotic cell to reach an understanding of how the cell functions.

There are three conveyors which can hold one engine block each. The engine block enters the cell on the first conveyor where it will wait until the previous engine blocks are completed. When this is done all engine blocks move forward to the next conveyor. The middle conveyor is where the assembly is performed, the first thing that happens when the engine block reach the middle conveyor is that the it will slow down and a cylinder places a stopper in front of the engine block. The engine block will then be locked into assembly position by a fixture that holds it firmly into place. The fixture consists of powerful hydraulic cylinders that holds the engine block by adding pressure in six points. When the engine block is in position the assembly begins. The robot first picks up seven main caps from the engine block and then places them on a different conveyor with small pallets that are dedicated to each engine block in order to ensure that the pieces follows their respective engine block through upcoming processes. Then the robot tool ejects the gripper for the cooling nozzles in Z-axis. The gripper takes hold of one nozzle and then screws the nozzle into the engine block. The robot assembles six nozzles into the engine block. When the assembly is completed the engine block moves to the last conveyor and the pallets with the main caps also starts to move forward, the engine block then waits until the operation before is completed and then leaves the cell. All these operations are mainly controlled by the PLC and the robot only performs the operations that the PLC order.

5.2 *Communication*

The communication within the cell is performed by PROFIBUS. There are eleven nodes which can be seen in Figure 6. These nodes handle different types of equipment within the cell. In the figure the node for the robot communication is clearly identified by the icon in the PROFIBUS. There are three nodes for the pneumatic equipment in the robot tool, stoppers and magazine that are called FESTO. There are also three nodes that control three SEW frequency inverters which controls the conveyors. They are called MOVIMOT. One of the nodes controls the RFID in the production cell and it is called U-P6-B. The communication with other PLC´s are controlled by the two nodes called DP/DP. The last node ANYBUS controls the communication with Volvo's production systems.

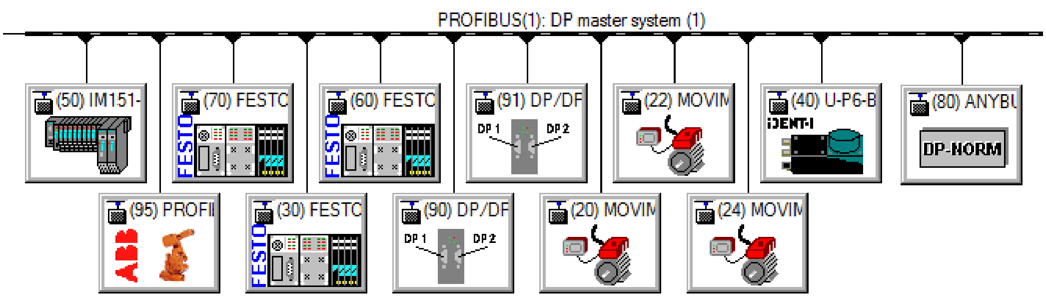


Figure 6. Shows the nodes in the PROFIBUS.

Simulation of the entire PROFIBUS is needed in order to gain all the information needed for the emulation model. This simulation will be done by PLCSIM within the Siemens software.

5.3 Inputs and outputs

There is a substantial amount of signals handled by the PLC. The important signals when building an emulation model are the inputs and outputs for the PLC. These inputs are signals that origins from different sensors throughout the robotic cell, such as proximity, pressure and cylinder sensors. The outputs are those signals that control different equipment in the cell, equipment such as the conveyors, conveyor stops and different types of cylinders. These inputs and outputs are those that will be needed in order to build an emulation model, however, these signals are not the only ones present in the PLC program. The PLC program is stocked with different types of memories, function blocks and sequential programs that also requires a great deal of signal processing. These functions are in most part executed within the PLC program and are therefore not needed in order to build the model, but these signals needs to be identified in order to rule them out. There are also signals that are used for communication with other PLC´s throughout the manufacturing line and for communication within Volvo's production system, these signals also need to be identified for the same reason.

When building the model, a lot of time is spent sorting through these signals to identify and decide whether the signal is needed for the model or not. In Siemens Step 7 there is a feature where it is possible to export all the signals within the PLC program as an Excel file, this file contained all the signals and this file was used in order to identify all the relevant inputs and outputs that were needed in order to build the emulation model. From the start this file contained approximately 1000 signals, but only approximately 110 PLC signals are used in the emulation model. When it comes to communication between the PLC and the Robot there are in addition eight words used for inputs and eight words used for outputs.

The emulation model will handle around 350 digital signals in total.

5.4 Layout

When building an emulation model of an existing production cell it is crucial to know all the equipment placement and the layout of the production cell. Blueprints of the equipment and the layout was provided by Volvo. These blueprints where very accurate in X- and Y- axis, but the blueprints gave no information regarding the placement in Z-axis. This did not become an issue because the placement in Z-axis could be concluded by the position of the robotic tool in the emulation model and with the aid of the 3D models of the equipment.

5.5 Assembly tool

The assembly tool is located on the robot and controlled by the PLC. The tool consists of two grippers, one for the cooling nozzles that is mounted into the engine and one that removes the main caps from the engine. The gripper for the cooling nozzles is located on a plate that moves up and down the Z-axis on the assembly tool. The two grippers also have assembly tools that are located in such a way that the screw on the cooling nozzle or main caps will fit perfectly into the tool when the gripper has hold of the nozzle or main caps. The two grippers have both sensors for detecting if the gripper has a hold of the main cap or nozzle. There are also sensors that detects if the Z-axis is in the upper position, or in the lower position. The gripper for the main caps also have sensors for detecting the width of the grippers. This is due to the fact that one main cap is wider than the other, and therefore the width of the main cap gripper must be controlled.

5.6 Robot and PLC

The robot in the production cell is a ABB IRB 6600 -175-280 and a 3D model of the robot can be seen in Figure 7. It is a robot that reaches 280 cm and that can handle object with a weight up to 175 kg.

The PLC that controls the G750 assembly cell is a Siemens CPU 317-2. The programming language used is Siemens STEP 7 Version V5.5.

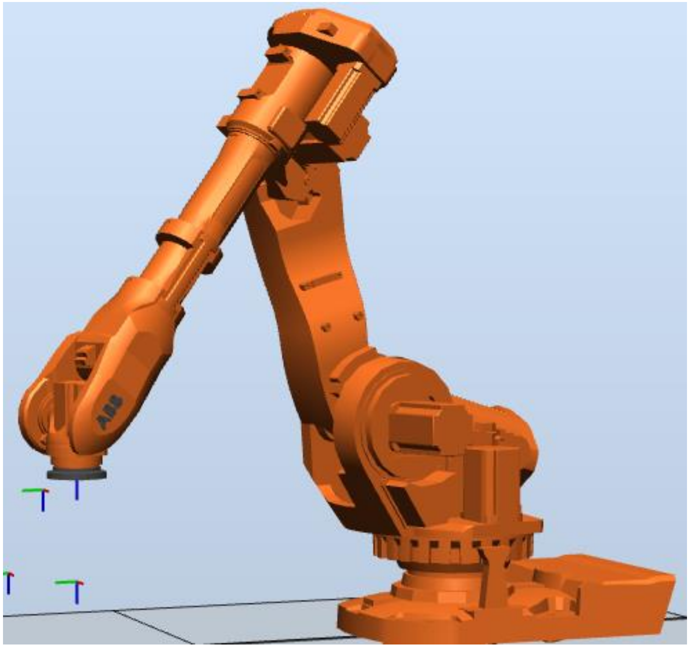


Figure 7. The robot used in G750.

5.7 Conveyor

There are three conveyors that moves the engine block through the production cell. The first one is called “inbana” and is connected with both the production cell G750 and the previous production cell. This conveyor has two sensors for detecting the engine block. The next conveyor is called “station1” and this is where the assembly is performed. This conveyor has three sensors for detection of the engine block and it also has the fixture for the engine block. The last conveyor is called “utbana” and this is where the engine block will wait until the next cell can receive it. Like the previous conveyor it also has three sensors for detection, but it also has a stopper in the middle of the conveyor. This is to ensure that an engine block will not be able to leave the cell until the next conveyor is ready. The stopper is a hydraulic cylinder that has two cylinder sensors.

5.8 Fixture

The fixture is located in conveyor “station1” and the purpose of the fixture is to hold the engine block firmly into place when the assembly begins. The parts that adds pressure to the engine block are controlled by hydraulic cylinders. There are four cylinders that controls the fixture. The cylinders all have two cylinder sensors to detect whether the cylinder is retracted or expanded. The hydraulic system in the fixture is monitored by a pressure-guard to ensure that the fixture is holding the engine block in its position.

5.9 Pallet conveyor

There is a conveyor that carries small pallets for main caps. This conveyor is located between the engine block conveyor and the robot. The conveyor is a friction conveyor which means that it is always running and the pallets are controlled by stoppers. There is a pre-stop in the conveyor, this is the position in which the pallets will wait until the station is ready for them. After the pre-stop there is a main stop. This is where the pallet will be located during the assembly. When the pallet is in the main stop, a cylinder will hold the pallet firmly in place. When the robot has left four main caps, the main stop will release the hold on the first pallet and then lock the second pallet into place. A 3D model of the pallet can be seen in Figure 8. The pre-stop, main stop and the cylinder is pneumatic cylinders that all have two cylinder sensors each.

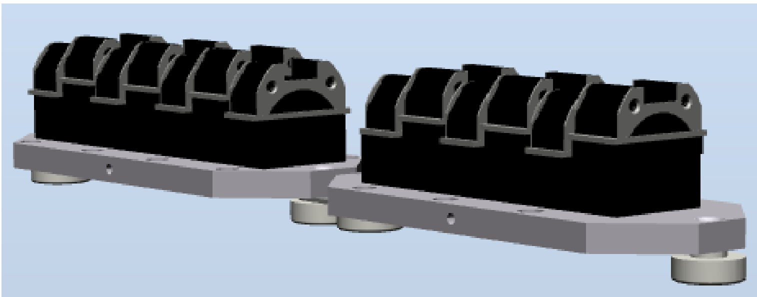


Figure 8. Pallets for main caps.

6 Computer aided design

In this chapter the emulation model is constructed. The 3D models of the equipment are structured into static and moving parts and then simplified and imported into the model.

6.1 *Converting and simplifying 3D models*

The 3D models provided by Volvo were made in Inventor. The models are extremely detailed and accurate because they are the actual 3D drawings of the equipment, tool and products that were made in the design phase. Inventor uses the file format .iam which is a file format that Simumatik3D is not capable to handle.

In order to import the 3D models into Simumatik3D a simplification and a file conversion to the model was needed to be done. To convert the file format from .iam to a .obj, .wrl or .dae RobotStudio was used by first importing the .iam file and then exporting it to a file format compatible with Simumatik3D.

To reduce the stress in Simumatik3D when emulating, all the 3D models had to be simplified. This was done using the software MeshLab. MeshLab reduces the number of faces and vertices on the 3D model but can still keep the geometry very detailed. In Figure 9 an example of a simplification from 15024 faces and 15888 vertices to 5000 faces and 2197 vertices can be seen. The simplified 3D model was saved and then imported into Simumatik3D.

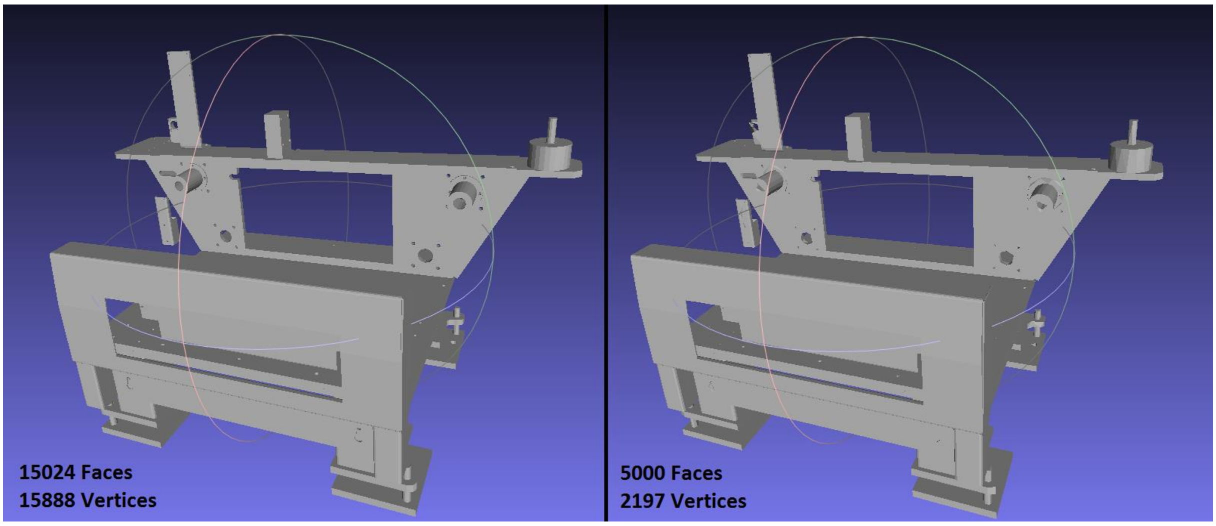


Figure 9. Comparison of a 3D model before and after the simplification.

6.2 *Building process*

The simplified and converted 3D models and objects from Simumatik3D´s own library were used when building the emulation model. In the building process blueprints of the production cell and equipment and electric schematics provided by Volvo were used. Observations and discussions with Volvo’s personnel were also done to clarify complex parts of the production cell.

When building the emulation model the cell was divided in to groups to easily structure the building process. The different main groups can be seen in Figure 10, these groups contain sub-groups of different objects such as motor, pneumatics, the robot and other parts of the system. The expanded structure tree can be seen in Appendix A.

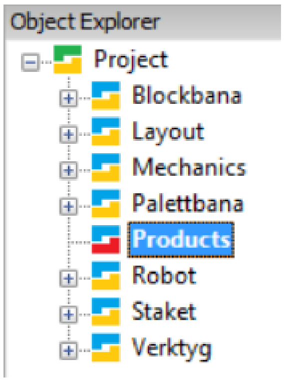


Figure 10. Structure tree of the emulation model.

6.2.1 Layout

When building the emulation model the layout provided by Volvo was imported into Simumatik3D. The imported layout was very helpful when positioning the conveyors, fixture, sensors, robot and magazines containing the cooling nozzles. Since the layout was imported with a scale of 1:1 the positioning of the equipment will be placed in the same way as the real production cell. The layout can be seen in Figure 11.

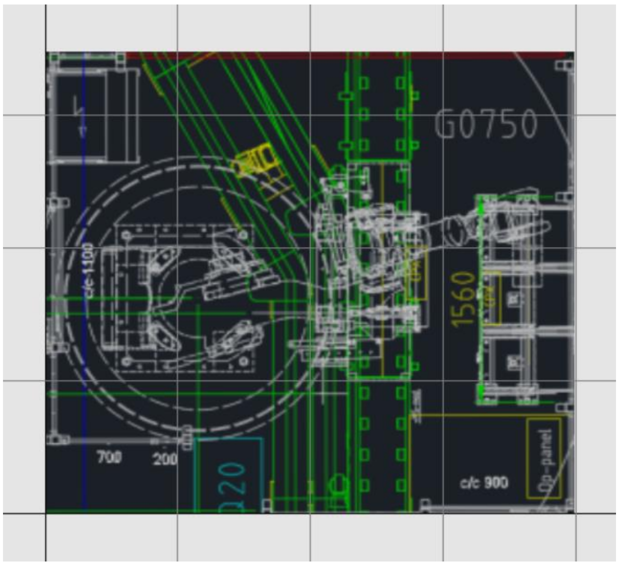


Figure 11. The layout.

6.2.2 Conveyors

In Simumatik3D there is a library that contains basic conveyors. The conveyors can be modified in order to replicate conveyors with belts, chains and rollers. The conveyors in G750 are small wheels that moves the engine blocks. In order to design an emulation model that is representative of the real production cell the conveyors needed to look like the real conveyors. To achieve that a 3D model of the cover plate without any physical attributes was imported and placed on top of a basic conveyor with belt which made the basic conveyor look representative of the real conveyor. This was a decision made in order to save time in the building process and it will not affect the result. The sensors on the conveyors where placed according to the layout picture. There were two sensors that were not present in the layout picture and they were placed according to their placement in the real production cell. Figure 12 shows how the conveyors with cover plates and sensors attached looks like.

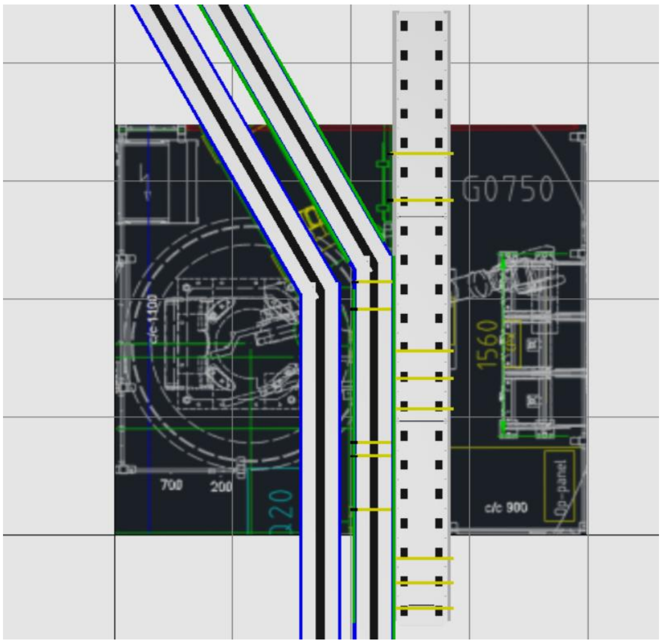


Figure 12. The conveyors with cover plate and sensors.

6.2.3 Robot and robot stool

The robot in the real production cell stands on a stool. The 3D model of the stool went through the previous method described with file conversion and simplification. The stool was then imported into Simumatik3D and placed according to the layout.

Simumatik3D has the ability to import robots from the robot library in RobotStudio. The robot imported to Simumatik3D is a IRB6600, which is the same robot as in the real production cell. This robot was then placed on the stool and can be seen in Figure 13.

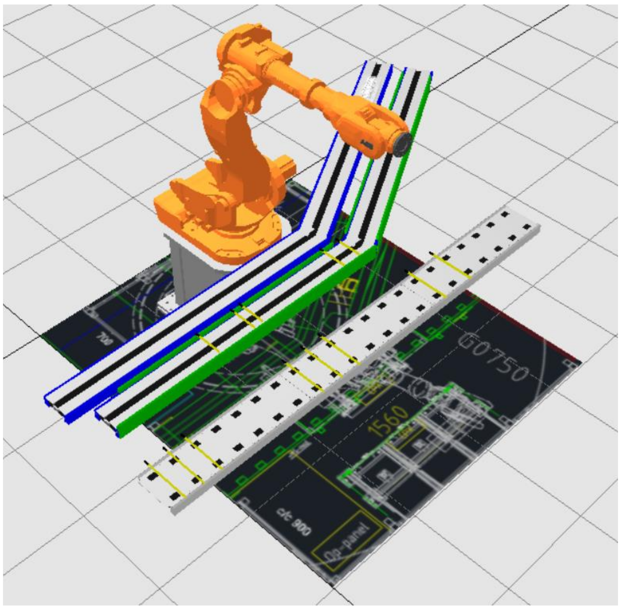


Figure 13. Robot in position.

6.2.4 Fixture with movable parts

To get movable parts in the emulation model they first had to be identified, this was done by observing the real production cell. When the movable parts had been identified they had to be extracted from the 3D models separately. Which can be seen in Figure 14. The separation of parts was done in RobotStudio.

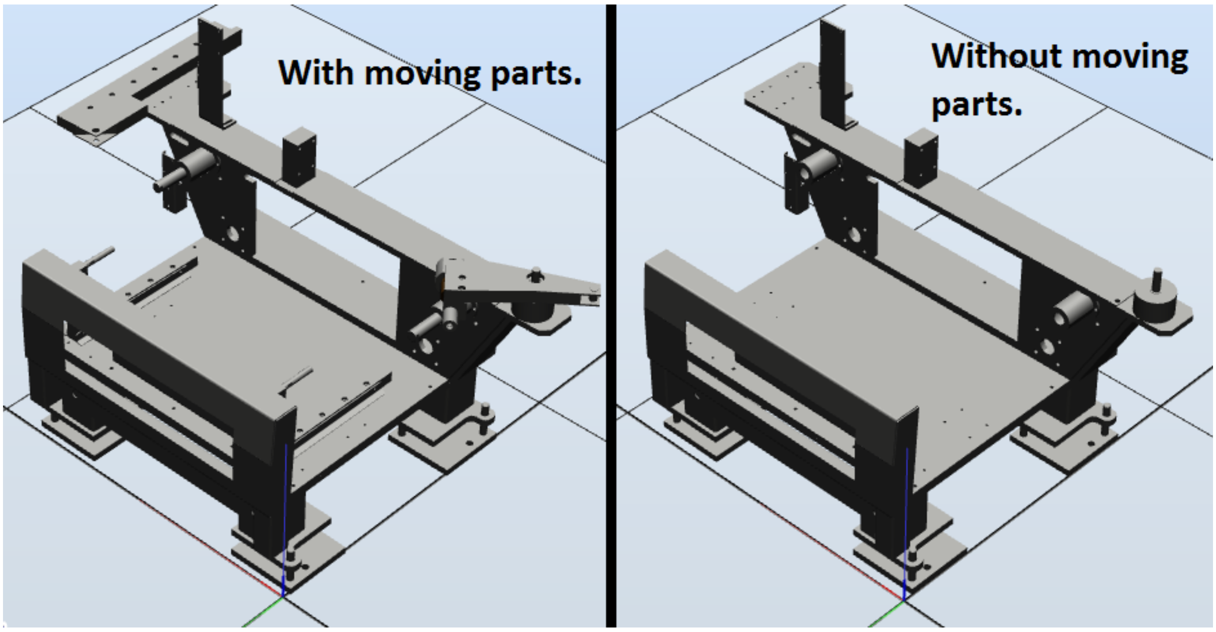


Figure 14. Fixture with and without moving parts.

When all the movable parts from an object had been extracted they could be imported to Simumatik3D. Two of the movable parts can be seen in Figure 15.

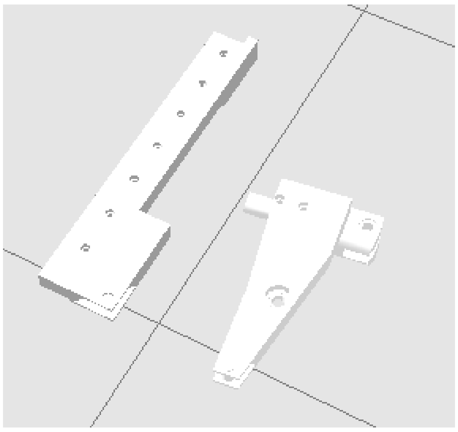


Figure 15. Separated moving parts.

After importing the parts into Simumatik3D they were attached to different joints, what type of joint is dependant of the type of movement in the real production cell. The movement of the objects are from the red axis to the green axis as can be seen in Figure 16. The blue part uses a rotational joint and the pink uses a translational joint. The colour was added to the movable parts to easier distinguish them from static objects.

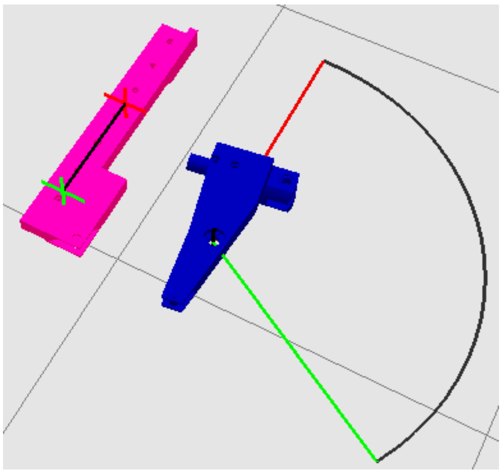


Figure 16. Pink part with translational joint and blue part with rotational joint.

The next step was to place the moving parts at the correct position at the static object which can be seen in Figure 17. In Figure 17 some moving parts have been added in the same way as previously mentioned.

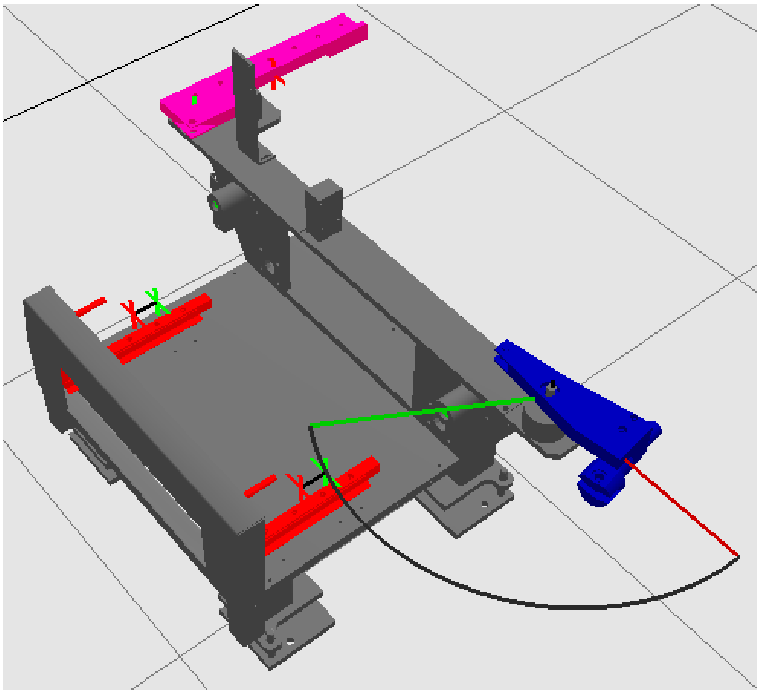


Figure 17. Movable parts added to the static object.

To know what position the moving parts have sensors are needed, these sensors are also present in the real production cell. The sensors might not have the same position in the real cell as in the virtual model, this is due to some limitations in Simumatik3D, but the sensors work and reacts in the same way as the real cell. There are two different types of sensors that can be seen in Figure 18. One type is a joint sensor which can detect the joint link and the other type of sensor is a proximity sensor that senses when something is near.

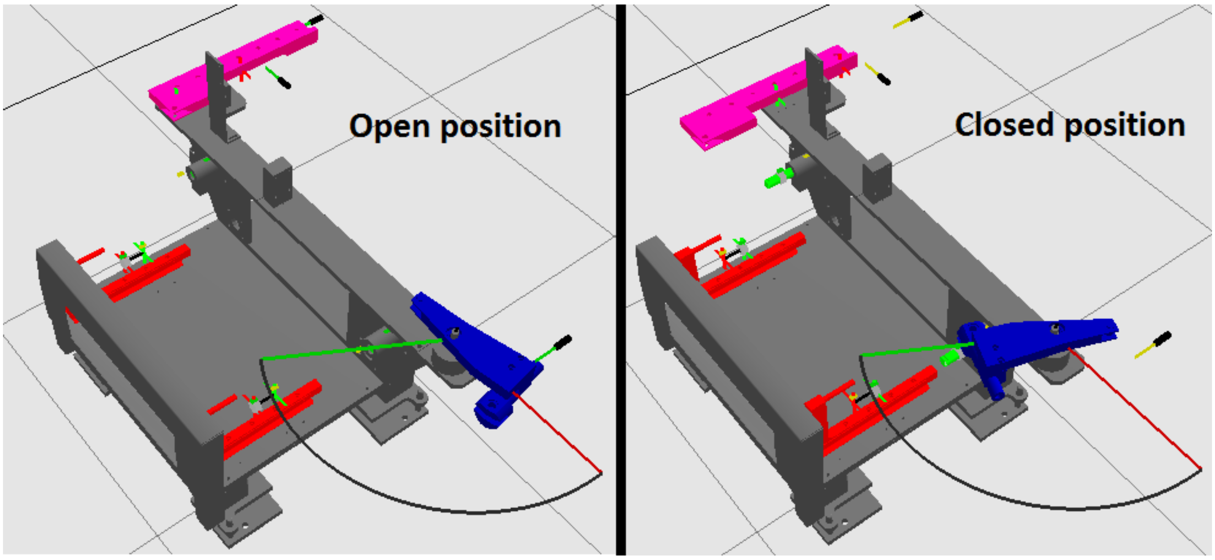


Figure 18. The fixture in open and closed positions.

The same method was used in order to build the robot tool. The emulation model with robot tool and fixture can be seen in Figure 19.

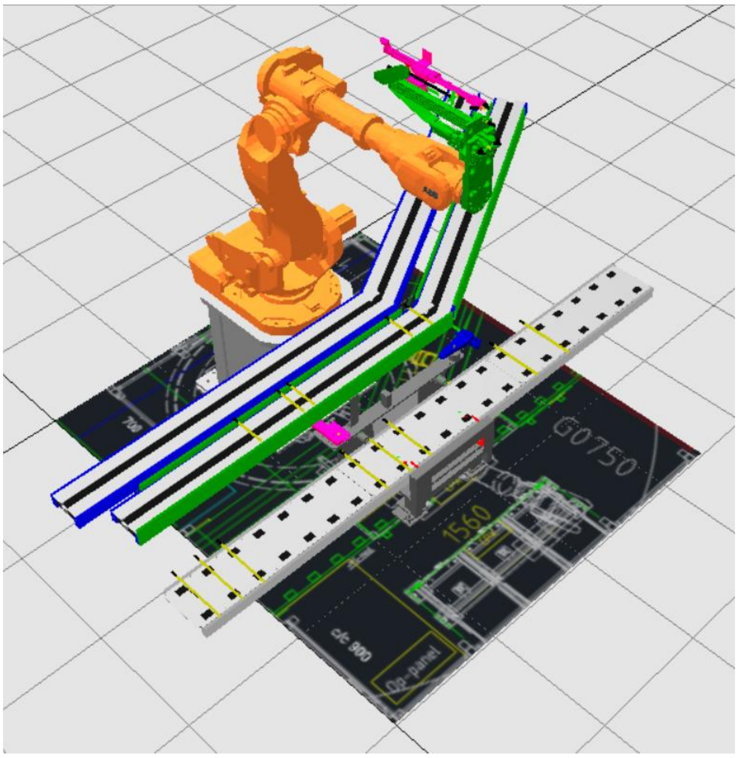


Figure 19. Robot tool and fixture added.

6.2.5 Magazine, light beacon and fence

The last parts to be added to the emulation model were parts and equipment that was either easy to build, like the light beacon, cabinets and fence which are parts from Simumatik3D´s own library, or parts that was not important for the function of the emulation model. They gave the emulation model a look that is more representative of the real production cell.

The magazine has a total of six input signals and three output signals. The inputs are signals that tells the PLC that the magazine is in place and locked. The output signals are for locking the magazines into place. There are three magazines in the production cell. In the model only one of these will be functional and the other two will work under the assumption that they are empty.

The last thing that was added to the emulation model was the light beacon. In Figure 20 the emulation model with fence, light beacon, warning signs, cabinets and magazine is presented and now the emulation model looks representative of the real production cell G750. Some colour standards were also introduced at this point, the red parts are movable parts, green parts are parts that will be moved by the robot. The red colour is used to easier distinguish what could be hazardous for an operator and the green parts are used for easier see the small details in the model.

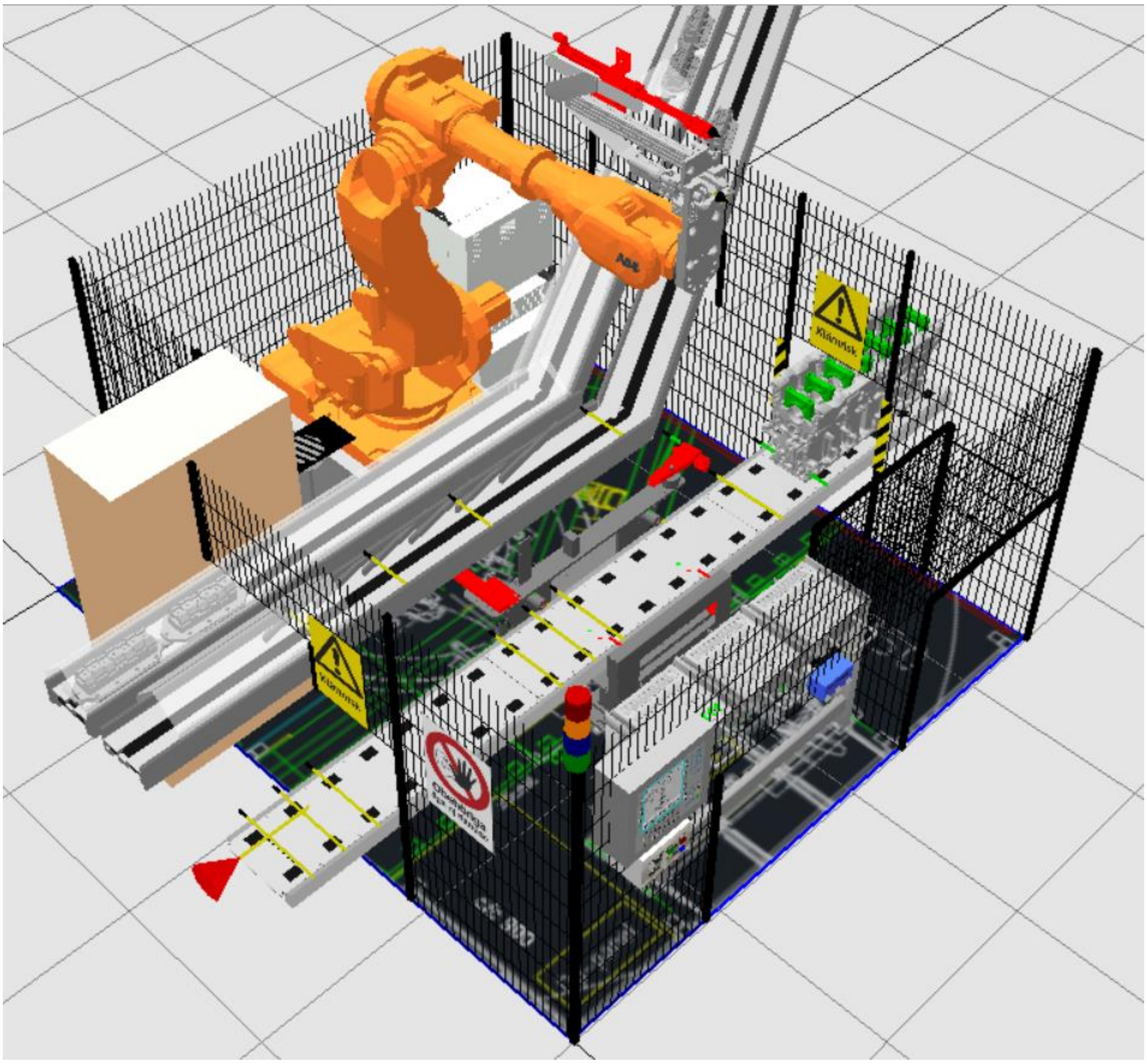


Figure 20 The finished emulation model.

7 Virtual environment

In this chapter connection between the different software´s that are used in order to build the emulation model will be established and explained.

7.1 PLC inputs and outputs

When building the model, a lot of time was spent sorting through the signals to identify and decide whether the signal was needed for the model or not. In Siemens Step 7 there is a feature where it is possible to export all the signals within the PLC program as an Excel file, this file was used in order to identify all the relevant inputs and outputs that was needed in order to build the emulation model, establish communication and control the virtual environment. From the start this file contained approximately 1000 signals, but only approximately 110 PLC signals are used in the emulation model.

7.2 Simumatik3D and PLCSIM

When the building process of the emulation model was completed, communication with the Siemens software PLCSIM was needed in order to emulate the PLC program. When PLCSIM and Simumatik3D is installed on the same computer, Simumatik3D has the ability to connect with the Siemens software PLCSIM. It can also write and read values in the PLCSIM software thus enabling a full emulation of the PLC program. PLCSIM performs a simulation of the entire PLC program and simulates the connection to the different nodes on the PROFIBUS line which makes the emulation of the program as realistic as possible without the real hardware.

Figure 21 shows the signals between Simumatik3D and PLCSIM. An example of how the signals work is that when a sensor is activated in Simumatik3D it will write the new value of the sensor input into PLCSIM. When an output signal is set in PLCSIM Simumatik3D will receive that signal and start a conveyor.

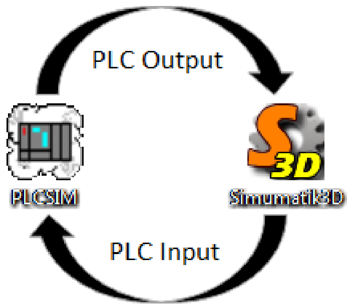


Figure 21. The signals between Simumatik3D and PLCSIM.

7.3 Simumatik3D and RobotStudio

Simumatik3D also has the ability to establish a connection with the ABB software RobotStudio. It is possible to import ABB robots into Simumatik3D and when a robot is imported the robot will mirror the movements done by any robot created in RobotStudio thus enabling the real robot programming to be emulated in Simumatik3D. A video of the emulation model running the real robot code can be seen in Appendix B.

The connection between Simumatik3D and PLCSIM was stable and reliable but the connection between Simumatik3D and RobotStudio created problems in Simumatik3D due to the high amount of signals that was used in the robotic programming. In the robotic program, there was a total amount of 128 signals in and 128 signals out. The high amount of signal processing done by Simumatik3D slowed down the system and created problems. The signals in RobotStudio therefore needed to be changed for the system to work properly. Group signals were added in RobotStudio, which made it possible to group together 8 signals in RobotStudio into one group signal. When this were done and all the signals was grouped then the 128 signals in and 128 signals out became 16 bytes in and 16 bytes out which eased the signal processing in Simumatik3D and made the program work smoother.

The signal processing between PLCSIM and RobotStudio is crucial when emulating the production cell G750 consisting of both a Siemens PLC and ABB robotic equipment. Figure 22 shows how Simumatik3D handles the signals between the PLC and RobotStudio. Simumatik3D does not modify any values between the PLC and RobotStudio, it only forwards the signals. This means that PLC outputs becomes robot inputs and robot outputs becomes PLC inputs.

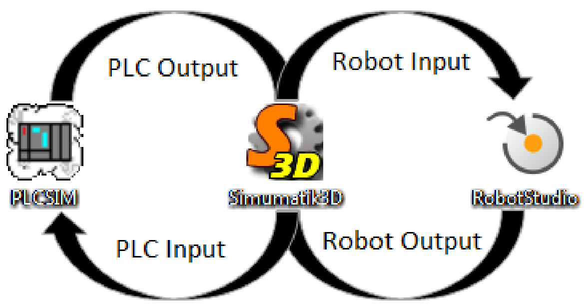


Figure 22. Signals between PLCSIM and RobotStudio.

7.4 HMI

When the emulation model was built, an HMI was added to Simumatik3D. This HMI represents the actual buttons on the HMI cabinet in the real production cell. The real HMI and Simumatik3D´s version of the HMI can be seen in Figure 23.

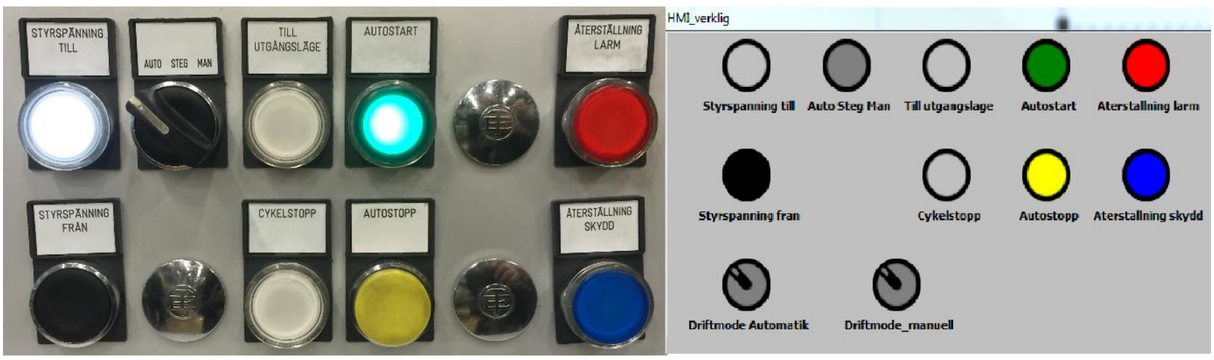


Figure 23. HMI Comparison.

However, there is also an HMI that has more features, where the different equipment in the production cell can be manually controlled, all the sensor values are shown and where the different alarms are presented amongst other things. Instead of actual buttons, this HMI is controlled through a touchscreen. The software used to build this HMI is embedded in Siemens Step7, and it is called WinCC. The WinCC software can be simulated together with PLCSIM and when Simumatik3D is connected to PLCSIM all the different HMI´s in the production cell can be used in the emulation model. Figure 24 shows WinCC that simulates HMI.

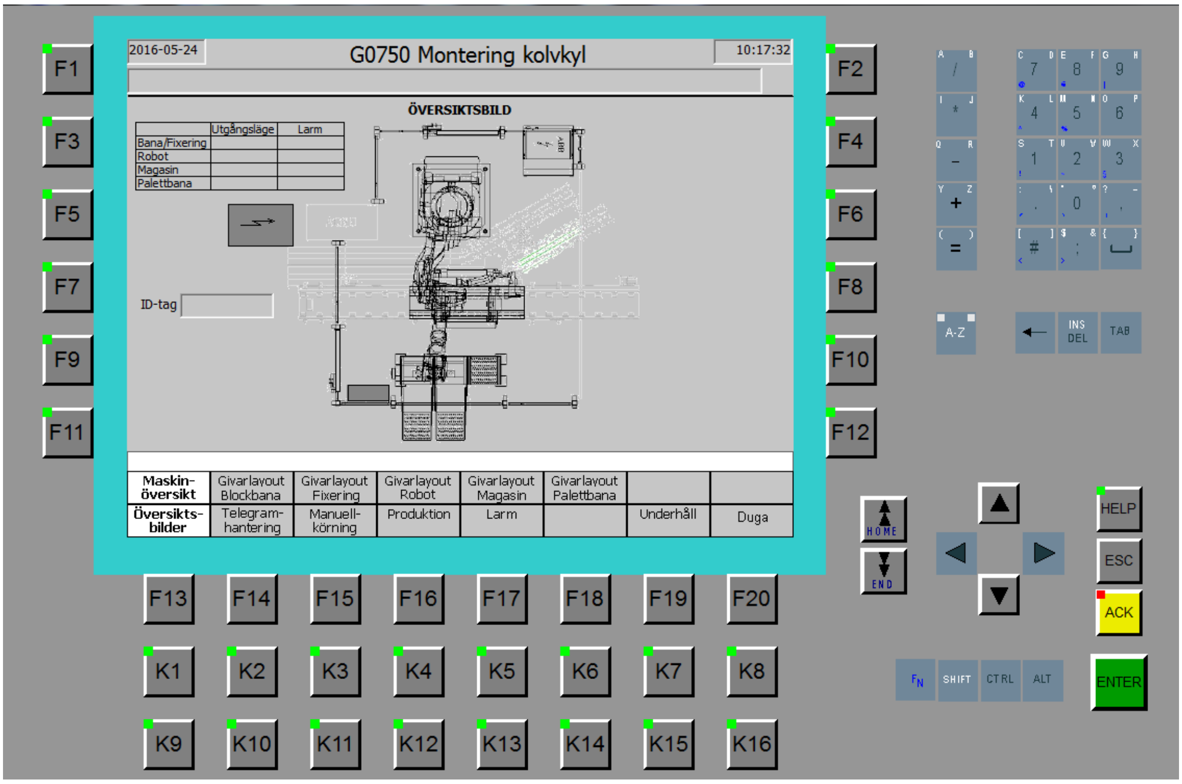


Figure 24. WinCC which simulates the Siemens HMI.

The Siemens HMI communicates to PLCSIM which then affects Simumatik3D depending on what commands are given from Siemens HMI. This makes it possible to control specific parts of the cell manually when needed, a video of this can be seen in Appendix C. The connections between all the different software is visualized in Figure 25.

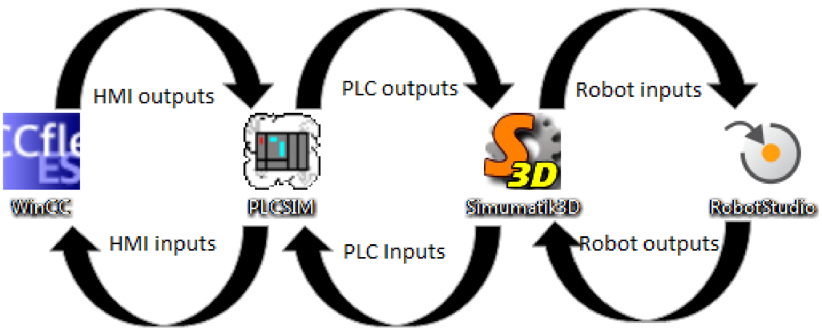


Figure 25. A simplification of the communication between the software.

8 Testing the virtual model

This chapter is dedicated to the experiments that were performed in order to validate the research questions and main objectives.

8.1 Experiments

To investigate if the emulation model reacts to changes in the program, fail scenarios and movements three experiments were conducted. The experiments were performed when the production was stopped for lunch break in order not to affect the production in any way. Personnel from Volvo handled the real production equipment and the authors handled the emulation model.

8.1.1 Auto stop

This experiment was conducted in the emulation model and the real production cell. The experiment was performed by pressing the auto stop button at four different sequence steps in the production cycle and then pressing the go home button. Each of the different stops was performed on different production engines, just to ensure that the tests did not affect each other. The goal of the experiment was to see if both cells reacted in the same way when provoked by this experiment.

The four stops in the production cycle were the following:

- The first stop was right after the fixture locked the engine block in place.
- The second stop was performed when the robot picked up the last main cap.
- Right after the last main cap had been placed on the pallet was the third stop
- When the robot had picked up the first cooling nozzle was the fourth stop.

The results of this experiment were that the real production cell and the emulation model behaved in the same way. In all four tests of the experiment both models placed the robot in home position as well as the fixture and waited for the operator to give it the next instruction. The robot in both cells moved along the same paths and the sequence of which the fixture released the engine block were the same as well.

8.1.2 Replicating robot error

During some testing in the real production cell the robot was unable to automatically move to the home position when the home position button was pushed. The error shown by the PLC were that the robot was outside restart area. This error meant that the robot had to be controlled manually in order to reach the home position.

In order to investigate if such robot errors could be replicated in the emulation model, the emulation model were subjected to the same test as the real production cell. A video of this can be seen in Appendix D.

The result of the replication of the error were that the emulation model reacted in the same way and showed the same error message as the real production cell.

8.1.3 Missing main cap

In order to check if the emulation model reacts to fail scenarios an experiment was performed. The experiment was to remove one main cap and see if the emulation model would react to the missing main cap. The first main cap was removed from the engine block and the emulation was started. The model behaved as usual until the first main cap was to be dropped off at the pallet then an error message was shown on the Siemens HMI. The error message described that the robot did not have the main cap and that it should be placed back in the engine block and then the cycle should be restarted. This experiment on the emulation model can be seen in Appendix E.

This was experiment was also performed in the real production cell. The robot and PLC behaved in the same way as they did in the emulation model. Both the emulation model and the production cell sent the same error code and message and the robot stopped at the same place over the pallet.

8.1.4 Adding new code - PLC

To check if the emulation model could handle changes in the PLC code, a new function was created in the PLC program and downloaded to PLCSIM. The added function was a small bit of code consisting of a toggling contact that switches an output on and off at a rapid speed, the code can be seen in Figure 26. The output was then connected to a new lamp in the emulation model. The emulation was started and a full production cycle was performed. The model behaved exactly the same as before except for the new lamp that was rapidly switching on and off.

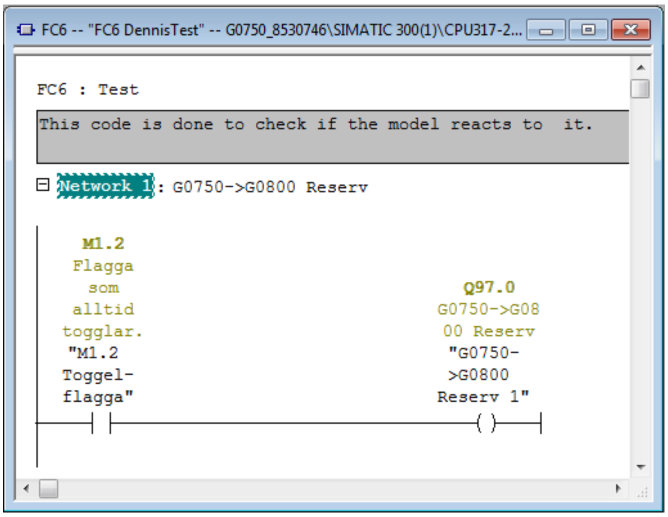


Figure 26. The added test code.

8.1.5 Adding new code – Robot

To see if changes to the robot program affects the emulation model an experiment was performed. The experiment was to add a new line of code which makes the robot wait for a signal that is connected to a key switch in Simumatik3D. The new line of robot code can be seen in Figure 27. The result of this experiment is that the robot waits until the key switch is turned in Simumatik3D. When the key switch is turned the robot will behave as usual, but if the key switch is turned back the robot will finish its procedure and then stop and wait until the key switch is turned again.

```
        WaitDI DP_DI13,1;
```

Figure 27. The change in robot code.

9 Evaluation of the virtual model

The appearance and sequence is evaluated and the simplifications to the virtual model is presented in this chapter.

9.1 Appearance

The emulation model is representative of the real production cell regarding appearance as can be seen in Appendix F where a comparison between the emulation model and the real production cell can be seen. The major difference is that the emulation model lacks doors in the fence. Implementing doors in the emulation model is a feature that Simumatik3D not yet have. What is important when it comes to the appearance in emulation models is that all the equipment that are programmed in the real production cell is present in the model and are representative of the real life equipment. Equipment that are not programmed and not part of the sequence of operations but present in the real production cell are only important to implement in an emulation model if the model will be used to study more than the programming, for instance to validate robot paths and reachability.

9.2 Sequence

The sequence in the emulation model is exactly the same as in the real production cell when the real PLC program is used. The PLC programming in the production cell must be fool-proof in order for Volvo to ensure that no engines are wrongly assembled or that production slows down due to programming malfunctions. This makes the requirements on the emulation model extremely high. The model has to behave exactly as the real production cell in order to work. Otherwise the PLC program will stop the model. What differs in the sequence is the speed of which the engines are assembled in the production cell, the cycle time in the emulation model is 199,6 seconds and the cycle time in the real production cell is 150 seconds. This discrepancy will be discussed later. A video of a full production cycle can be seen in Appendix G.

9.3 Simplifications and limitations

When building the emulation model some simplifications were made. There were also some limitations to the software that needed to be handled. A table of all the simplifications and limitations in the emulation model can be seen in Table 5. In the table it is clearly stated what is simplified, how it was done and why it was done. Within the simplifications there are also limitations stated.

Table 5. Simplifications made in the emulation model.

What was simplified?	How was it simplified?	Why was it simplified?
Rotational devices on robot tool.	Change in program code.	The rotational device uses a separate controller that does not have the ability to communicate with Simumatik3D.
Fuses for safety.	Change in program code.	Saving time.
Hydraulic pressure switch in fixture.	Change in program code.	Simumatik3D does not support a pressure switch function.
Communication with Volvo's production system.	Change in program code.	It is outside the delimitations of this thesis.
Grippers on the robot tool.	Building joints and sensors without 3D model or physics in Simumatik3D.	Saving time.
Sensor for cooling nozzle detection on the robot tool.	Connecting the sensor input to the gripper's closed position output.	Problem with physics in Simumatik3D.
Doors, emergency stops and buttons related to these.	Nothing, these circuits is normally closed and will not affect the model if not implemented.	Simumatik3D does not support emergency stops or doors.
Magazine 2 and 3.	Not given motion or buttons.	The function is fulfilled by magazine 1.
RFID	Change in program code.	Saving time.

10 Discussions

This chapter contains discussions about the building process of the emulation model, the hardships the authors encountered and thoughts about the software.

When the market survey was performed different types of methods were used. The contact with the different software providers was performed in different ways. This makes the market survey somewhat unreliable since the quality of information gained is very dependent on the source. For example, you could gain a lot more information about the software through a personal meeting with a developer compared to information only available on a webpage. To gain the best knowledge about all the software in the market survey personal meetings with representatives of the different software would have been preferable. Also meeting companies that have used the different software for emulation would have given us more objective and reliable sources for information. But also testing the software for a week could have increased the knowledge of how user friendly the software is.

When the emulation model was built we did not have many problems during the computer aided design phase. The software Simumatik3D was known to us and the software is easy to use. The problems that did occur during the computer aided design phase was mainly related to converting 3D models. We spent a lot of time trying to find an easy way to convert 3D files into formats that could be imported into Simumatik3D. When we learned that RobotStudio could be used to import and export 3D files the problem was solved. RobotStudio has the ability to convert all the different 3D models into formats that suited Simumatik3D and when using RobotStudio it is also possible to disconnect the moving parts in the building process. This makes RobotStudio the perfect tool for handling the 3D models when building emulation models in Simumatik3D, since RobotStudio also is used to emulate the robots in Simumatik3D it is also the aspect of using as few software as possible to take into consideration. When the software MeshLab later was found the 3D models exported from RobotStudio could be simplified in order to ease the stress on the laptop used. The time spent building the model was approximately 50 hours this includes; the handling and converting of the 3D models, building the emulation model, mapping of all the PLC and robot signals, and establishing the connection between PLCSIM and Simumatik3D. Since this was the first time building an emulation model we believe that the time will be greatly lessened in a second emulation project. This because of higher knowledge about the different software and also because of the building methods of movable parts that are used in this thesis.

Regarding user-friendliness in emulation software we noticed during the market survey that some of the different software required time consuming manual modelling in C# and other programming languages, this is also mentioned in (Lee & Park, 2014). One of the reasons that we decided to build the emulation model using Simumatik3D is that the software does not require any programming skills in C# or similar programming language. It is easy to build movable objects using the rotational and translational joints.

There were some struggles with the communications between the different software packages. When the emulation model was built and the connection between the different software was established we spent a lot of time sorting through the signals and verifying that the mapping of the signals was the correct. In this phase there where complications with the signals between RobotStudio and PLCSIM. In order to simplify the signal processing, we first tried to use words as signals between the different software. But we found that the signals were mismatched and signals that were passed from RobotStudio into PLCSIM did not affect the right signals in PLCSIM. When we discussed this issue with personnel at Volvo we found out that Siemens does not read words in the same way as RobotStudio. Siemens software reads words backwards compared to RobotStudio, this means that a word that is passed from RobotStudio to PLCSIM with the byte numbers 0-1 will be read in PLCSIM as 1-0. We managed to solve this problem using bytes instead of words for communication, but the solution increased the signal processing thus adding more stress to the laptop. The best solution would have been if PLCSIM would have been able to read words or even double words to decrease the signal processing.

Some simplifications were made in order to make the emulation model work properly. These are presented in chapter 9.3. The reason for making simplifications in the emulation model varied. Sometimes it was to save time in the project, sometimes it was due to limitations in the software and in some cases it was because of the complexity in Volvo's production system. When building an emulation model the main reason for building it must be established. If it is crucial to emulate the robot tool in order to make it behave exactly as in real life, then more time should be spent on the robot tool than on other applications in the production cell. One of the main objectives was to build an emulation model that where representative of the real production cell. In order to make the emulation model representative we focused on replicating the sequence of operations in the production cell and making sure that the real PLC and robotic program could be executed to their fullest extent, with as little simplification as possible. Therefore, the grippers in the robotic tool were not given dynamic behaviour, they were emulated using translational joints instead. This does not affect the sequence of operations in any imperative way. Other things that did not affect the sequence were also simplified, for example the gates in the production cell and the fuses. The reason for simplifying the pressure guard on the fixture was that it was a complex feature to emulate and Simumatik3D could not do this in an easy way. However, this is another thing that does not affect the sequence of operations in any imperative way, the reason that there is a pressure guard is to make sure that the fixture is either closed or open properly. This feature is also controlled by the pneumatic sensors that is implemented in the emulation model, therefore the pressure guard is redundant in the emulation model.

We noticed when building the emulation model that the person that will build emulation models must have a very good knowledge about PLC programming. We have a basic knowledge which helped us a long way in the building process. But in order to solve the complex problems regarding Volvo's production system, personnel from Volvo had to assist in the process. This shows that when building an emulation model a knowledge in PLC programming is required in order to make the process easier and in order to build emulation models in shorter time. In the future Volvo is going to change how the communication with the production system is performed in the PLC program, it will be easier to disconnect the production system from the PLC program thus making building emulation models easier. The personnel at Volvo informed us that there is a simulation computer that is used in order to simulate the production system in commissioning projects. If this would have been known in the beginning the simulation computer would have been used thus probably eliminating the problems surrounding the production system.

There were some issues with the FPS in Simumatik3D. We built the emulation model using a Dell Ultrabook with Intel i5 processor. This laptop had to run four process heavy software of which two requires 3D graphics. This was very demanding for the laptop since it did not have a separate graphics processing unit (GPU). This made the emulation model work slowly. In Simumatik3D the number of FPS is visible on the screen, when the all the different software is running he FPS is reduced to approximately 8-10. A more powerful computer would have made the emulation model run smoother and increased the FPS to approximately 30. Due to the low FPS the cycle time of the emulation model was affected, the cycle time in the emulation model was 400-450 seconds. This is about 300-350 seconds more than the real production cell which is only about 130-150 seconds. Later in the project a new more powerful laptop was used for running the different software and the emulation model. The new computer had a separate GPU and a more powerful processor which made the emulation model run smoother with a FPS of approximately 20. With the new computer the cycle time was reduced to approximately 200 seconds. An even more powerful computer, preferable a stationary computer with high-end specifications could probably handle this task better than the laptops and probably also starting to get very close to the real cycle time of 130-150 seconds.

Finding and verifying errors in program code is one of the main objectives when using emulation tools. During the experiment missing detail in chapter 8.1.2 an error in the PLC program was found. When the gripper picks up the main cap from the engine block it is supposed to check that the main cap is present in the gripper before the robot moves to the pallet. The experiment clearly shows that this is not the case. Our supervisor at Volvo states that this is an error in the program. This means that the experiment validates that errors in the programming can be found with the use of emulation tools.

The usage of emulation models is not limited to commissioning projects. We believe that emulation models can be used to validate improvement suggestions regarding the production cells and validating new equipment. They could also be used for validating sequence changes in production equipment. We spent a lot of time running and re-running the emulation. During this time, we learned how to control the production cell, we gained a knowledge of how the cell works and how to control it using the manual mode on the HMI´s. We also learned how to solve different problems that occurred during the emulation, for example how to handle the production cell if a failure occurred. This indicates that emulation models can be used for training new personnel on how the equipment works.

Sustainable development is an important aspect to take into consideration in industries today. The results of this thesis clearly shows that emulation model can be used for verification and validation of PLC and robot code. This will give a positive effect on the sustainable development. The economical, ecological and social positive effect will be less usage of time, energy and money for companies that use emulation models for verification of PLC and robot code since the commissioning time will be shortened and visits from buyer to seller for verification visits will be lessened. Some of the positive effects of the use of emulation models is presented in chapter 1.4. However, the biggest positive effect that the use of emulation models will provide is the hazard-free environment for testing code, training operators and testing scenarios that would not have been possible in real life due to different type of hazards. There are no limits to the testing that can be done in a virtual model and the production equipment suffers no risk of damage during these tests. The operators can be trained in a safe environment without any risk of harm to either themselves ore the production equipment.

11 Conclusions

In the aim and objectives chapter there are some research questions and objectives that are focused on emulation software. One of the research questions is focused on which is the most common emulation software on the market. The software presented in the market survey are the ones that were found, the most common ones are hard to distinguish from these. The reason for that is that the market for emulation software is very new, and the usage of emulation software is not yet widely spread.

The ones that suit Volvo's needs are Simumatik3D and Experior. The main reason why these software packages suits the need of Volvo the best is that they have the ability to communicate with different brands of equipment. This is a feature that many of their competitors do not have. The market survey clearly shows that there are only two emulation software that suits Volvo's needs. Therefore, the main objective regarding presenting a minimum of three emulation software has not been reached. The reason that this objective was not met is the previously stated issues between different brands of equipment and the relatively new market.

The software chosen to build the emulation model was Simumatik3D. The emulation model that was produced in this software was then used for a live experiment against the real production cell G750. This fulfils the main objective which is to choose one emulation software to be subject of a live experiment in Volvo´s real production equipment. The experiments can be seen in chapter 8.

The real PLC and robotic programs are present in the emulation model, these are advanced industrial programs and the emulation model is run with these programs. This validates that it is possible to emulate advanced industrial programming of PLC equipment and robots. The performed experiments regarding program change in both robot and PLC program clearly shows that the emulation model reacts to the program changes done, which also validates that the programming can be emulated and that the emulation model can be used for validation of program code, this is also verified by the experiments in chapter 8. The experiment regarding missing detail verifies that errors in the program code can be found using emulation models. It also possible to replicate errors from the real production cell in the emulation model which is proven by the replicating robot error experiment. These conclusions answer the questions regarding finding and verifying errors in present production equipment's program code. In chapter 8.1.4 and 8.1.5 new PLC and robot code is added and verified by the emulation model, this concludes that research question concerning verifying new production equipment's program code is proven valid.

The Siemens HMI was implemented in the emulation model with the use of the software WinCC which has the ability to simulate the real Siemens HMI. This proves that it is possible to implement the Siemens HMI with the emulation model. The Siemens HMI is used for presenting errors, manual handling of the cells, checking the states of sensors and home position of the different parts of the cell.

An emulation model of G750 was built in Simumatik3D and it corresponds to the real production equipment regarding programming, as is proven by the fact that the real programs are run in the emulation model. The emulation model corresponds to movements that is proven by the experiments regarding auto stop done and it corresponds to the fail scenarios as is also proven by the experiments done. These facts validate that the last main objective which was to present a virtual model in the chosen emulation software that corresponds to real production equipment regarding programming, fail scenarios and movements has been fulfilled.

According to Reinhart and Wünch (2007) emulation models can be used in order to test real equipment in scenarios that would not have been possible due to danger and equipment damage. The results of the experiments performed in chapter 8 supports this statement.

11.1 Future work

When searching for relevant case studies the authors found that it was hard to find case studies within the emulation area that handled both PLC and robotic equipment. Especially case studies that cover the topic of robotic assembly. The reason for this is probably the complexity of robotic assembly compared to product handling systems that has been subject of emulation studies for years. The complexity of robotic assembly often lies with the physics changing aspect in the emulation software's. There must be features for picking objects, placing them on another object and in some cases changing the objects features. This is an area that should be more studied in order to develop the usage of VC in this area and to further clarify the problems related to emulation of robotic cells.

To further develop the usage of VC and to implement emulation as a VC tool studies should be performed in collaboration with the industries. There must be more studies performed where emulation models are built of real production equipment. This should be studies where different types of production cells are emulated. There is a vast difference in complexity depending on what the production cell does, and the emulation software´s must be put to the test to see how complex equipment they can handle. This should give answers to what kind of production equipment is worth building an emulation model of, and what kind of equipment is too complex to build.

In future work when building emulation models, the handling of the PLC and robot programs must be taken into consideration. When the emulation model of G750 was built simplifications and changes in the program code was written directly into the function blocks of the PLC program. In the future it would preferable to have a standardized work method for handling the simplifications. Instead of writing the changes directly into the PLC program a new function block containing all simplifications could be created. This function block could easily be activated or deactivated, thus enabling an easier transferring of PLC code between the virtual and real environment. With this function the same PLC program would be created for the emulation model and the real production equipment.

11.2 Future work for Volvo

In order for Volvo to continue their work with implementing emulation as an VC tool they must work closely with their suppliers of equipment. Their suppliers must have an understanding about VC and emulation and they must have the resources and knowledge about how emulation models are built. There are some things that are needed in order to build an emulation model. What was needed in order to build the emulation model of G750 were the following:

- 3D models of all the equipment
- Simumatik3D
- RobotStudio
- Siemens Step 7
- WinCC

- MeshLab
- Complete I/O list
- Blueprints of the layout and equipment
- Electrical schematics

Along with this the authors also spent a lot of time at the production cell to gain an understanding of how the cell worked. This will not be possible in the future when building emulation models in commissioning projects. Therefore, it is crucial that the sequence of operations and the function of all the equipment is known. A suggestion on how to achieve this is to use flowcharts to visualize and to gain an understanding of how the cell will work.

When building the emulation model some minor things took longer than it should have. One thing that was time-consuming was to find out the location and address of the sensors on the pallet conveyor. These was not present in the layout blueprint, and there were also sensors missing in the layout on other places. Therefore, the authors suggest that all the sensors and their address is clearly visible on the layout blueprint.

References

Banks , J., Carson, J. S., Nelson, B. L. & Nicol, D. M., 2009. *Discrete-Event System Simulation.* New Jersey: Pearsons Education Inc.

Bolmsjö, G. S., 2006. *Industriell Robotteknik.* 3:2 ed. Lund: Studentlitteratur AB.

Dahlin, J.-e., 2014. *Hållbar utveckling - en introduktion för ingenjörer.* 1:2 ed. Lund: Studentlitteratur AB.

Danielsson, F., Moore, P. & Eriksson, P., 2003. *Validation, Off-line programming and optimisation of industrial control logic,* Trollhättan: University of Trollhättan/Uddevalla.

Dzinic, J. & Yao, C., 2013. *Simulation-based verification of PLC programs,* Göteborg: Chalmers University of Techonology.

Erlandsson, T. & Rahaman, M. M., 2013. *Testing and verifying PLC code with a virtual model of Tetra Pak Filling Machine,* Gothenburg: Chalmers University of Techonology.

Groover, M. P., 2008. *Automation, Production Systems, and Computer-Integrated Manufacturing.* 3rd ed. Saddle River, New Jersey: Pearson Educational Inc..

Groover, P. M., 2015. *Automation, production systems , and computer-integrated Manufacturing.* Harlow: Pearsons Education.

Gröndahl, F. & Svanström, M., 2011. *Hållbar utveckling - en introduktion för ingenjörer och andra problemlösare.* 1:a ed. Stockholm: Liber.

Guerrero, L. V., López, V. V. & Mejia, J. E., 2014. *Virtual Commissioning with Process Simulation.* Online, Taylor & Francis, pp. 11-19.

Hoffman, P., Schuman, R., Maksud, T. M. & Preimer, G. C., 2010. *Virtual commissioning of manufacturing systems a review and new approach for simplification.* Kuala Lumpur, s.n., pp. 175-181.

Höst, M., Regnell, B. & Runeson, P., 2006. *Att genomföra ett examensarbete.* Lund: Studentlitteratur AB.

Johansson, M. & Nilsson, J., 2015. *Virtual Commissioning - Verification of PLC logic in simulation programs,* Skövde: Skövde Högskola.

Koninklijke Bibliotheek, 2016. *Koninklijke Bibliotheek National Library of the Netherlands.* [Online] Available at: https://www.kb.nl/en/organisation/research-expertise/research-on-digitisation-and-digital-preservation/emulation/what-is-emulation
[Accessed 10 03 2016].

Law, A. M., 2014. *Simulation Modeling and Analysis.* New York: McGraw-Hill Education.

Lee, C. G. & Park, C. S., 2014. *Survey on the virtual commissioning of manufacturing system,* Toronto : Toronto University.

Makris, S., Michalos, G. & Chryssolouris, G., 2012. *Virtual Commissioning of an Assembly Cell woth Cooperating Robots,* Patras: Department of Mechanical Engineering and Aeronautics.

McGregor, I., 2002. *The Relationship between simulation and emulation.* Salt Lake City, Brooks-PRI Automation Inc, pp. 1683-1688.

NE, 2016. *Nationalencyklopedin.* [Online]
Available at: http://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/programmerbart-styrsystem
[Accessed 01 04 2016].

NE, 2016. *Nationalencyklopedin.* [Online]
Available at: http://www.ne.se/uppslagsverk/encyklopedi/lång/emulering
[Accessed 25 03 2016].

Oppelt, M. & Urbas, L., 2014. *Integrated Virtual Commissioning an essential Activity in the Automation Engineering Process. From virtual commissioning to simulation supported engineering,* Dallas , TX: IEEE .

Reinhart, G. & Wünsch, G., 2007. *Economic application of virtual commissioning to mechatronic,* s.l.: German Academic Society for Production Engineering.

Siemens, 2016. *Tecnomatix Manufacturing Simulation and Validation.* [Online]
Available at:
https://www.plm.automation.siemens.com/en_us/products/tecnomatix/manufacturing-simulation/human-ergonomics/jack.shtml#lightview-close
[Accessed 15 03 2016].

Stephan, S., Donath, U. & Haufe, J., 2012. *Towards an integrated simulation and virtual commissioning environment for controls of material handling systems.* Dresden, Fraunhofer Institute for Integrated Circuits Design Automation Division.

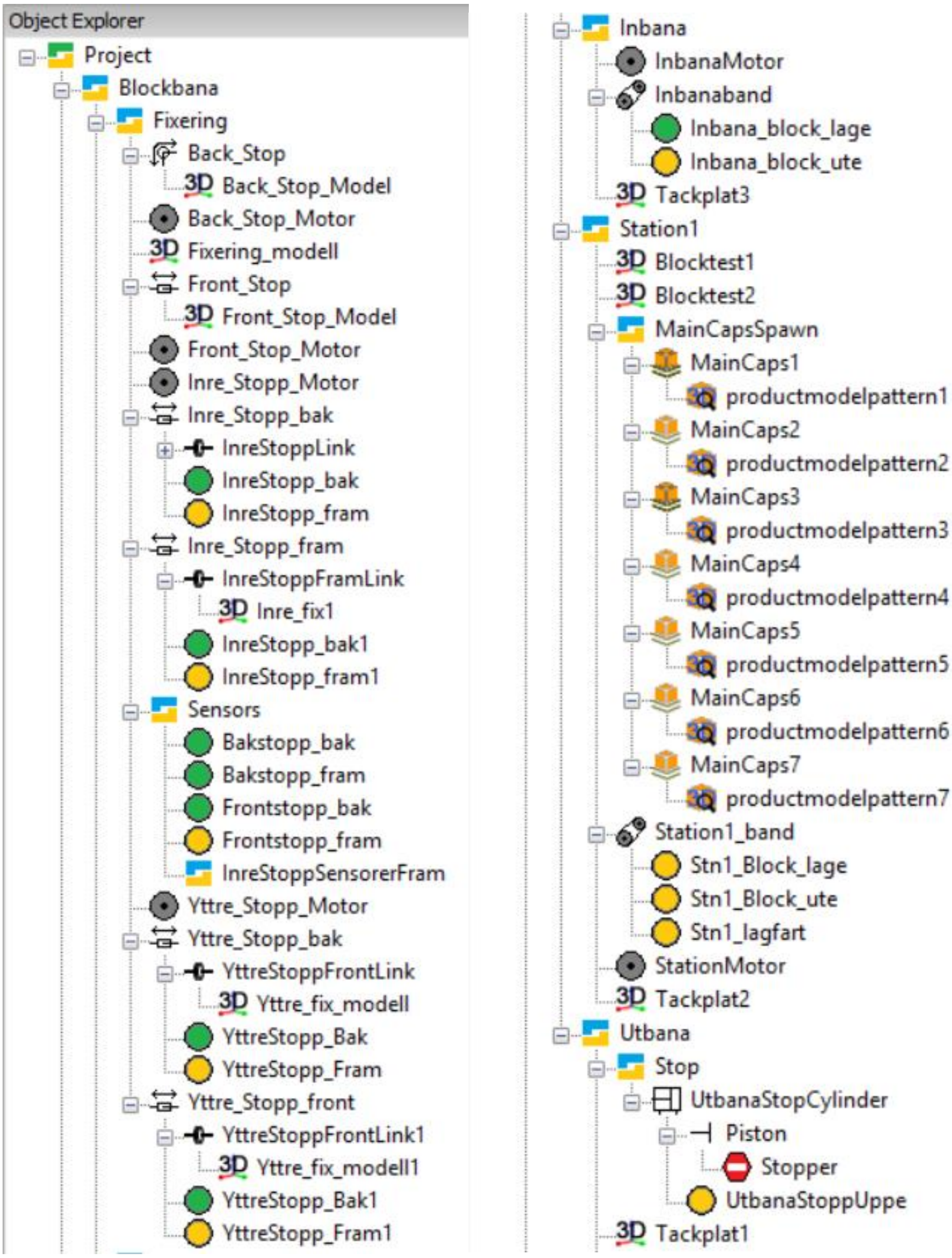
Tonnquist, B., 2014. *Projektledning.* 5th ed. Stockholm: Sanoma Utbildning.

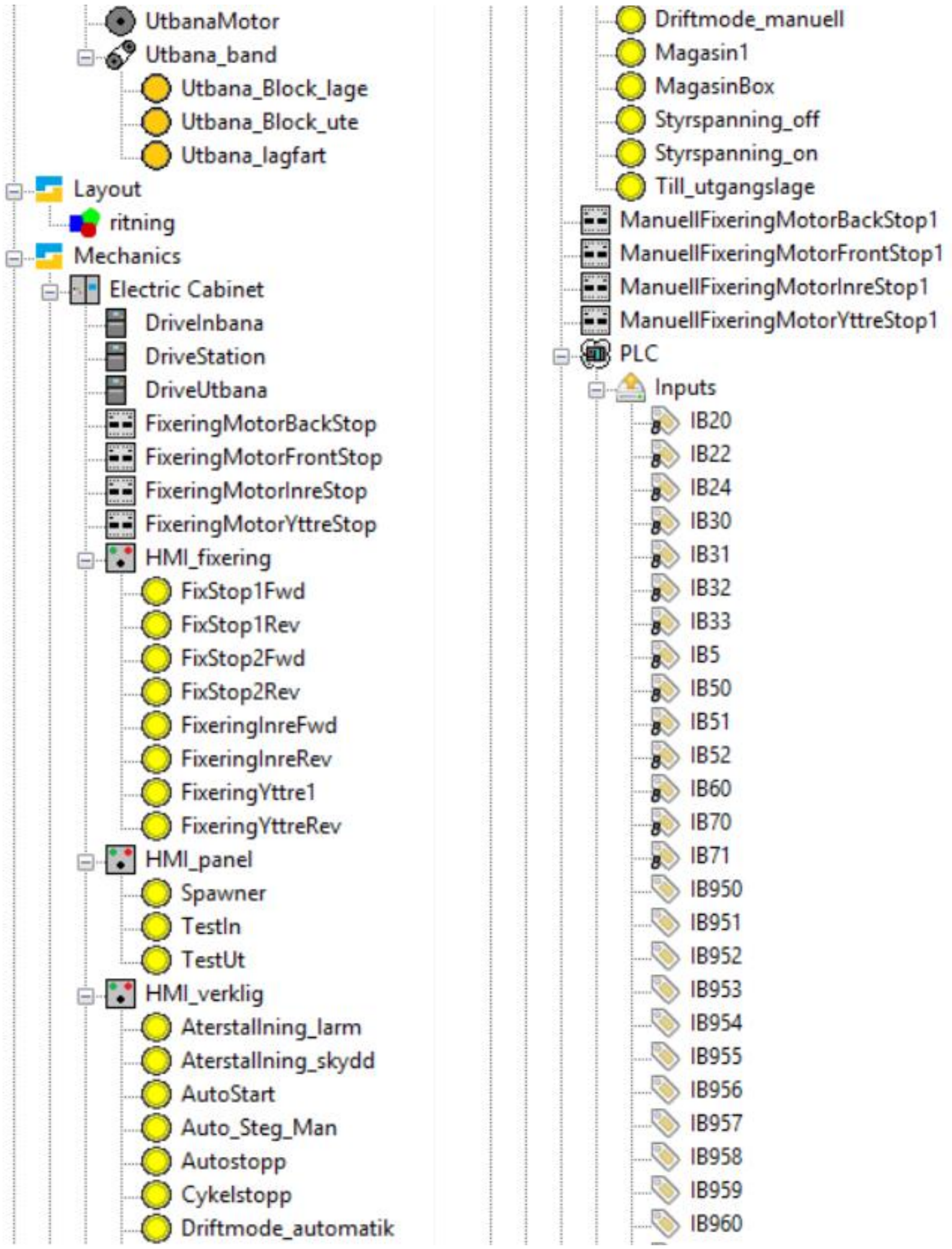
W. C. o. E. a. D., 1987. *Our Common Future,* Oxford: Oxford University Press.

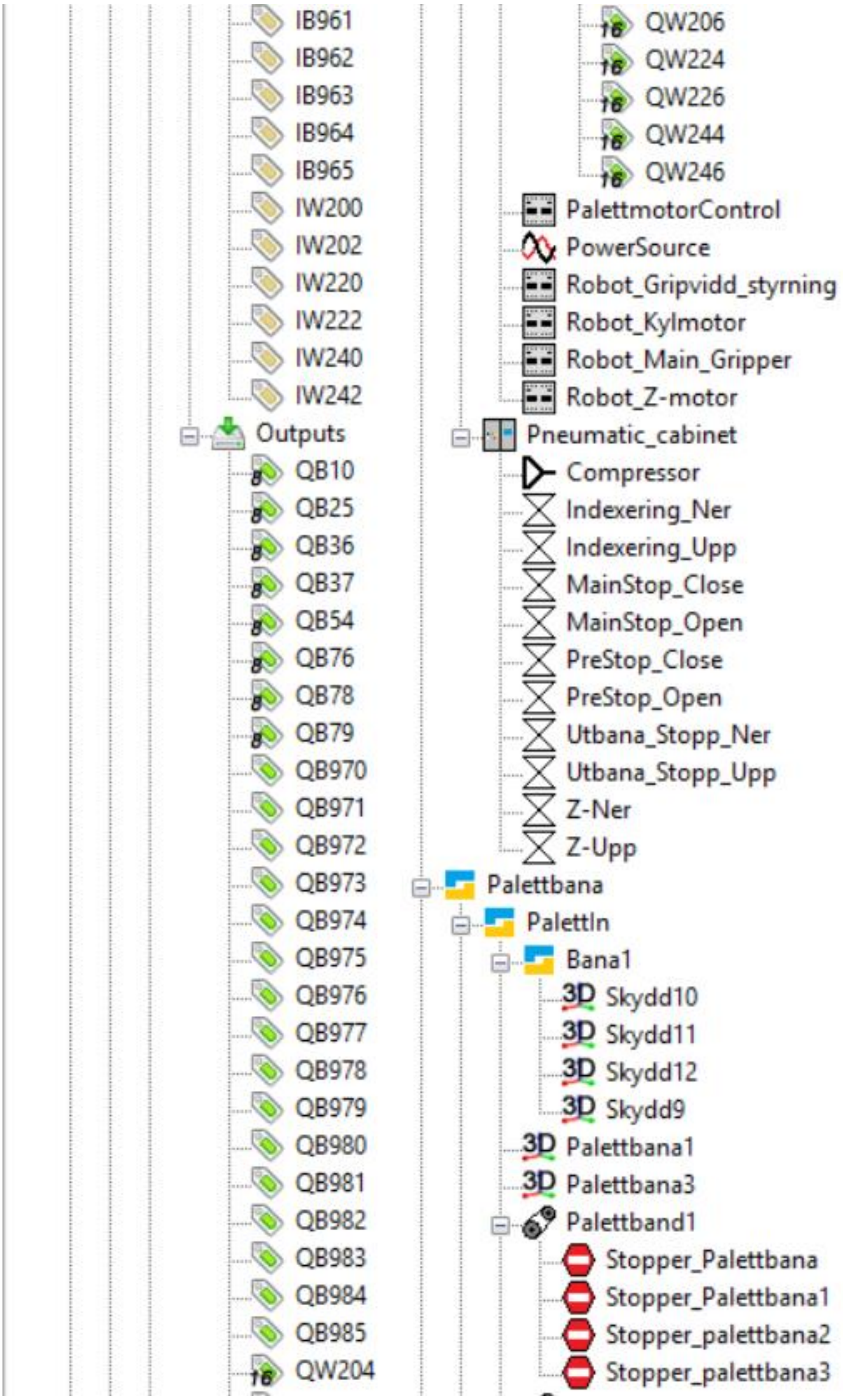
Zäh, M., Wünsch, G., Hensel, T. & Lindworsky, A., 2006. *Nutzen der virtuellen Inbetriebnahme: Ein Experiment,* s.l.: s.n.

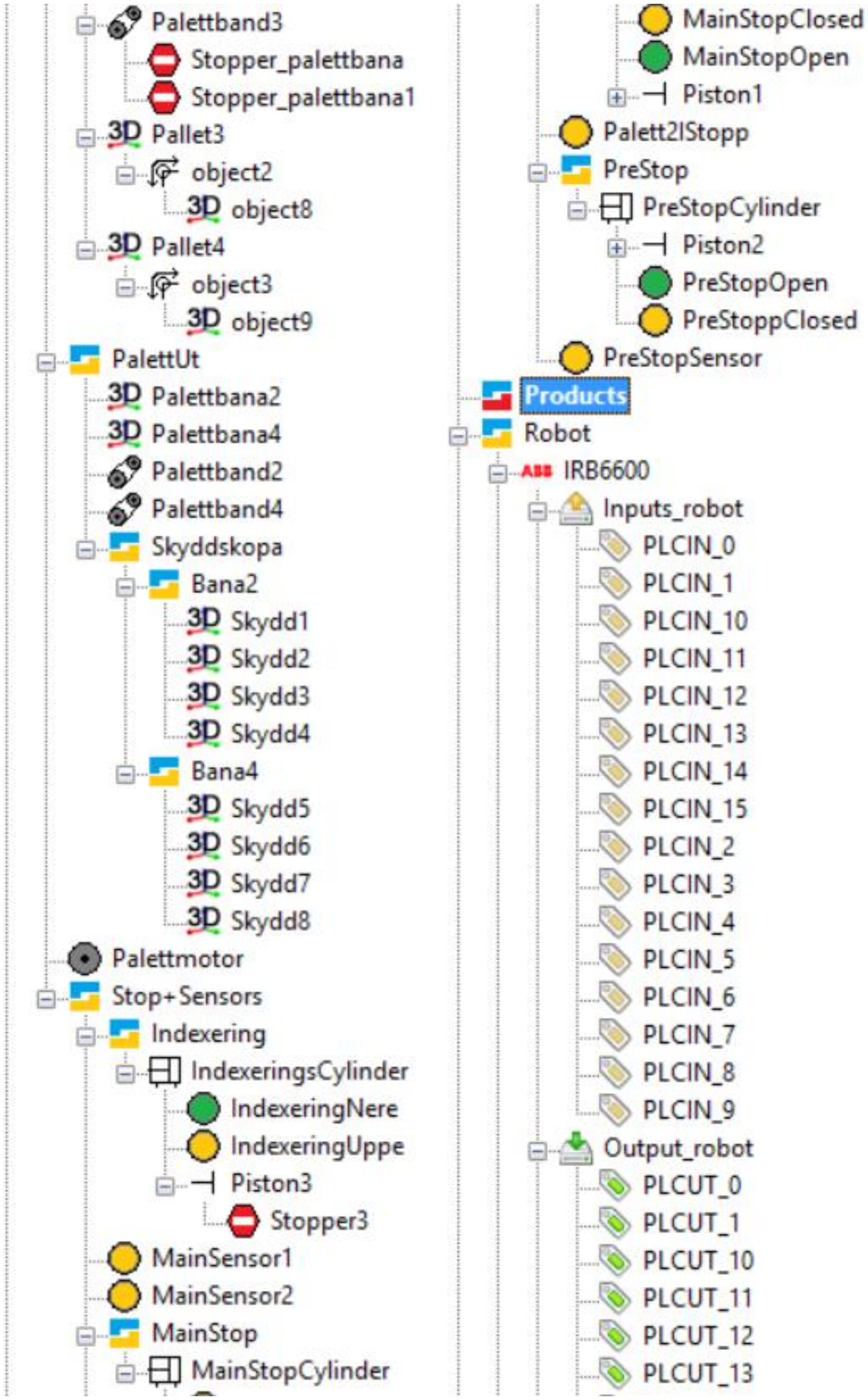
Appendices

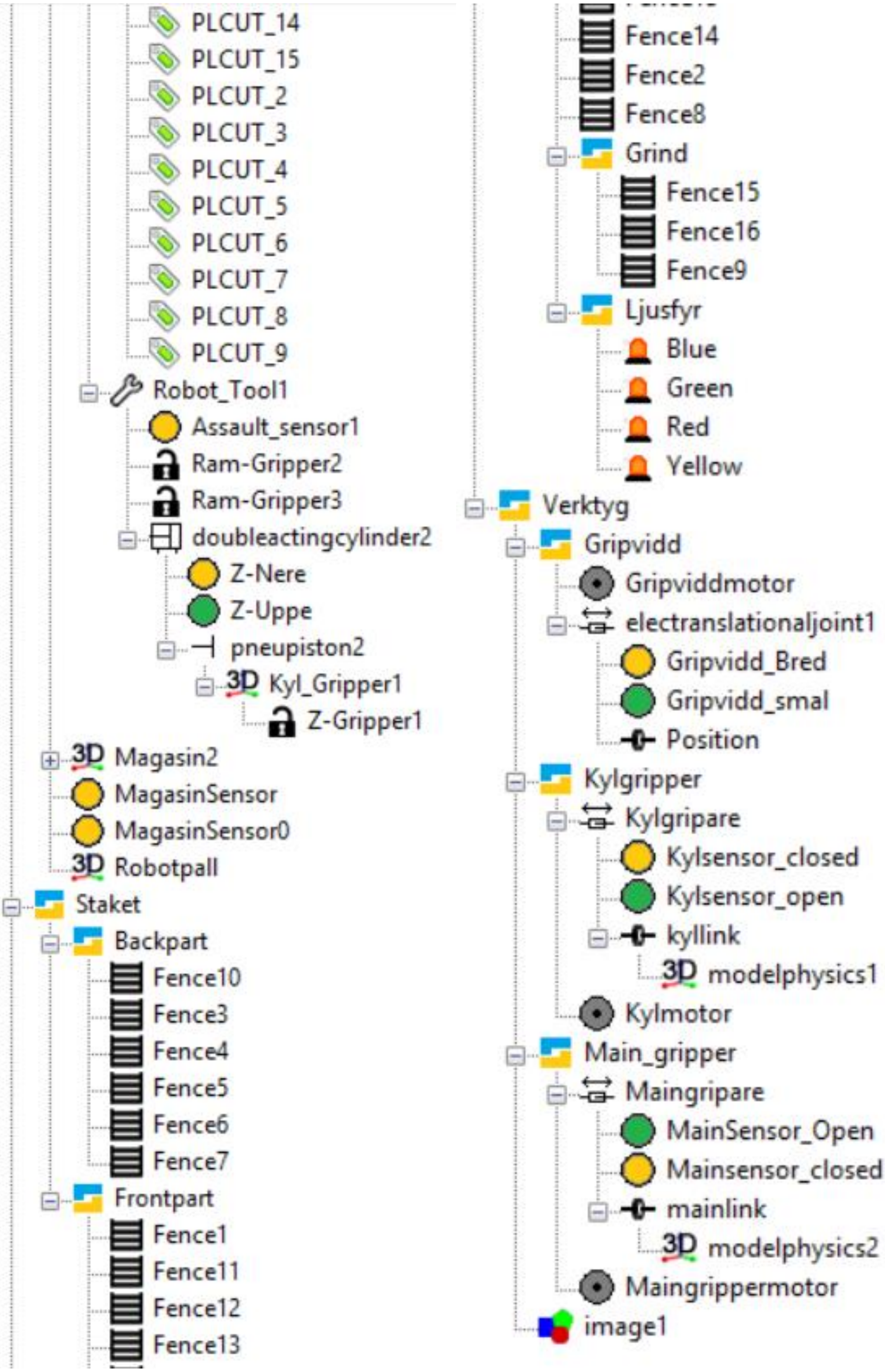
Appendix A– *The structure tree*




UtbanaMotor
Utbana_band
Utbana_Block_lage
Utbana_Block_ute
Utbana_lagfart
Layout
ritning
Mechanics
Electric Cabinet
DriveInbana
DriveStation
DriveUtbana
FixeringMotorBackStop
FixeringMotorFrontStop
FixeringMotorInreStop
FixeringMotorYttreStop
HMI_fixering
FixStop1Fwd
FixStop1Rev
FixStop2Fwd
FixStop2Rev
FixeringInreFwd
FixeringInreRev
FixeringYttre1
FixeringYttreRev
HMI_panel
Spawner
TestIn
TestUt
HMI_verklig
Aterstallning_larm
Aterstallning_skydd
AutoStart
Auto_Steg_Man
Autostopp
Cykelstopp
Driftmode_automatik
Driftmode_manuell
Magasin1
MagasinBox
Styrspanning_off
Styrspanning_on
Till_utgangslage
ManuellFixeringMotorBackStop1
ManuellFixeringMotorFrontStop1
ManuellFixeringMotorInreStop1
ManuellFixeringMotorYttreStop1
PLC
Inputs
IB20
IB22
IB24
IB30
IB31
IB32
IB33
IB5
IB50
IB51
IB52
IB60
IB70
IB71
IB950
IB951
IB952
IB953
IB954
IB955
IB956
IB957
IB958
IB959
IB960


IB961
IB962
IB963
IB964
IB965
IW200
IW202
IW220
IW222
IW240
IW242
Outputs
QB10
QB25
QB36
QB37
QB54
QB76
QB78
QB79
QB970
QB971
QB972
QB973
QB974
QB975
QB976
QB977
QB978
QB979
QB980
QB981
QB982
QB983
QB984
QB985
QW204
QW206
QW224
QW226
QW244
QW246
PalettmotorControl
PowerSource
Robot_Gripvidd_styrning
Robot_Kylmotor
Robot_Main_Gripper
Robot_Z-motor
Pneumatic_cabinet
Compressor
Indexering_Ner
Indexering_Upp
MainStop_Close
MainStop_Open
PreStop_Close
PreStop_Open
Utbana_Stopp_Ner
Utbana_Stopp_Upp
Z-Ner
Z-Upp
Palettbana
PalettIn
Bana1
Skydd10
Skydd11
Skydd12
Skydd9
Palettbana1
Palettbana3
Palettband1
Stopper_Palettbana
Stopper_Palettbana1
Stopper_palettbana2
Stopper_palettbana3

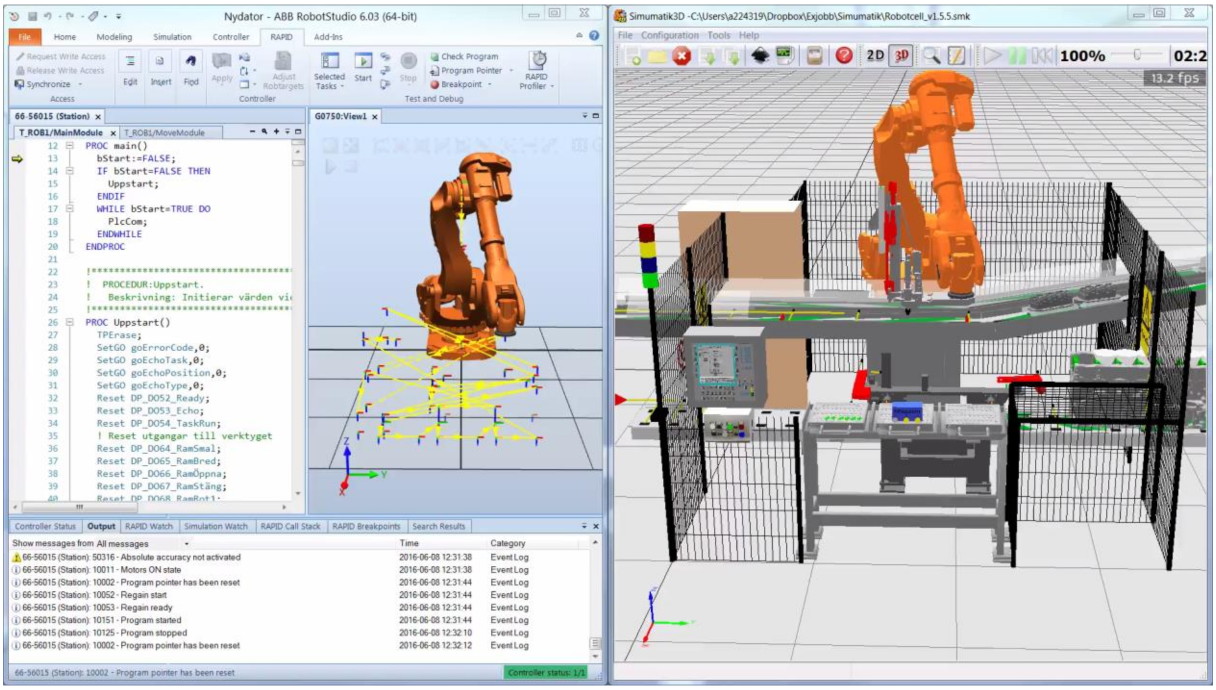

Palettband3
Stopper_palettbana
Stopper_palettbana1
Pallet3
object2
object8
Pallet4
object3
object9
PalettUt
Palettbana2
Palettbana4
Palettband2
Palettband4
Skyddskopa
Bana2
Skydd1
Skydd2
Skydd3
Skydd4
Bana4
Skydd5
Skydd6
Skydd7
Skydd8
Palettmotor
Stop+Sensors
Indexering
IndexeringsCylinder
IndexeringNere
IndexeringUppe
Piston3
Stopper3
MainSensor1
MainSensor2
MainStop
MainStopCylinder
MainStopClosed
MainStopOpen
Piston1
Palett2IStopp
PreStop
PreStopCylinder
Piston2
PreStopOpen
PreStoppClosed
PreStopSensor
Products
Robot
IRB6600
Inputs_robot
PLCIN_0
PLCIN_1
PLCIN_10
PLCIN_11
PLCIN_12
PLCIN_13
PLCIN_14
PLCIN_15
PLCIN_2
PLCIN_3
PLCIN_4
PLCIN_5
PLCIN_6
PLCIN_7
PLCIN_8
PLCIN_9
Output_robot
PLCUT_0
PLCUT_1
PLCUT_10
PLCUT_11
PLCUT_12
PLCUT_13


PLCUT_14
PLCUT_15
PLCUT_2
PLCUT_3
PLCUT_4
PLCUT_5
PLCUT_6
PLCUT_7
PLCUT_8
PLCUT_9
Robot_Tool1
Assault_sensor1
Ram-Gripper2
Ram-Gripper3
doubleactingcylinder2
Z-Nere
Z-Uppe
pneupiston2
Kyl_Gripper1
Z-Gripper1
Magasin2
MagasinSensor
MagasinSensor0
Robotpall
Staket
Backpart
Fence10
Fence3
Fence4
Fence5
Fence6
Fence7
Frontpart
Fence1
Fence11
Fence12
Fence13
Fence14
Fence2
Fence8
Grind
Fence15
Fence16
Fence9
Ljusfyr
Blue
Green
Red
Yellow
Verktyg
Gripvidd
Gripviddmotor
electranslationaljoint1
Gripvidd_Bred
Gripvidd_smal
Position
Kylgripper
Kylgripare
Kylsensor_closed
Kylsensor_open
kyllink
modelphysics1
Kylmotor
Main_gripper
Maingripare
MainSensor_Open
Mainsensor_closed
mainlink
modelphysics2
Maingrippermotor
image1

Appendix B – *Full production cycle with RobotStudio*

This video shows Simumatik3D and RobotStudio running side by side, it also shows that the real robot code is used in the emulation model.

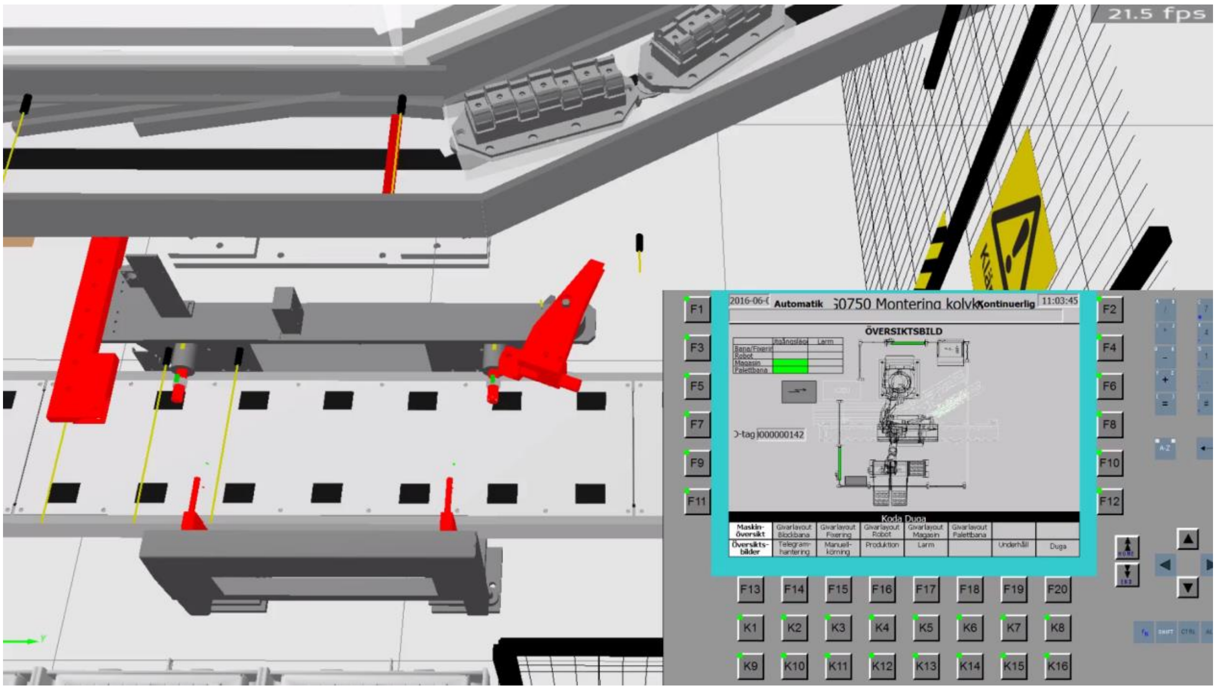
https://www.youtube.com/watch?v=ZvVlIt9ofy8



Appendix C – *Manual control with Siemens HMI*

This video shows how the emulation model can be controlled with the Siemens HMI.

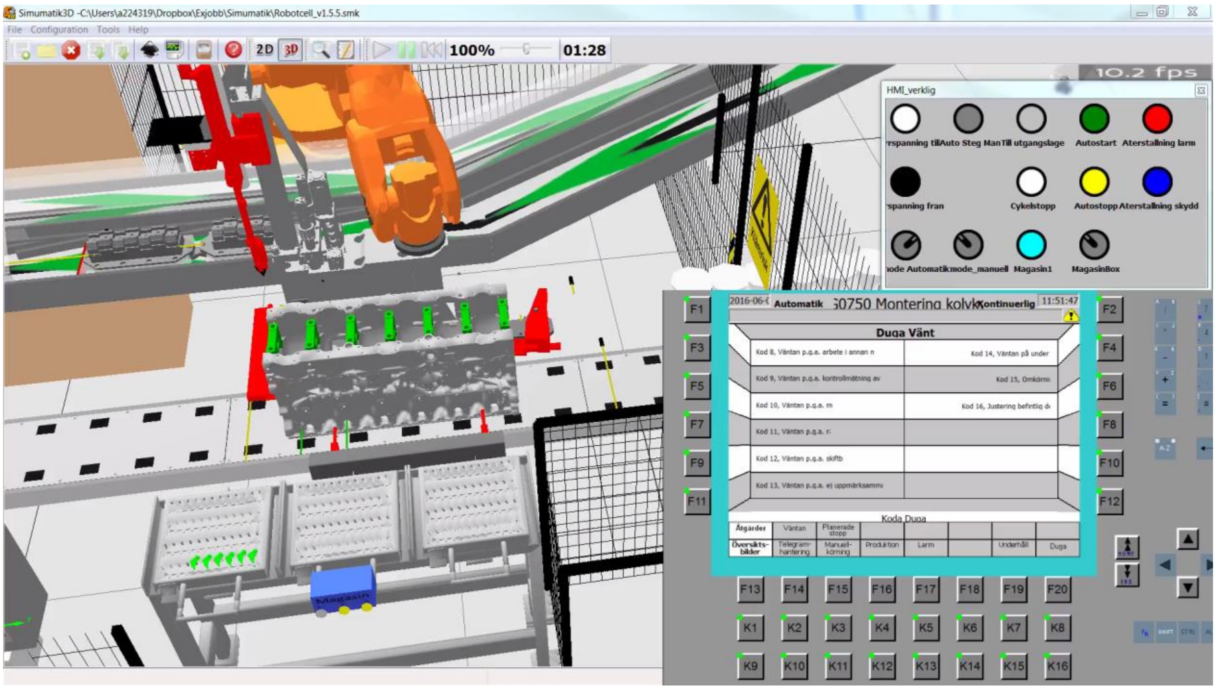
https://www.youtube.com/watch?v=O3t6WPlOGCM



Appendix D– *Replicating robot error*

This video shows the experiment presented in chapter 8.1.2.

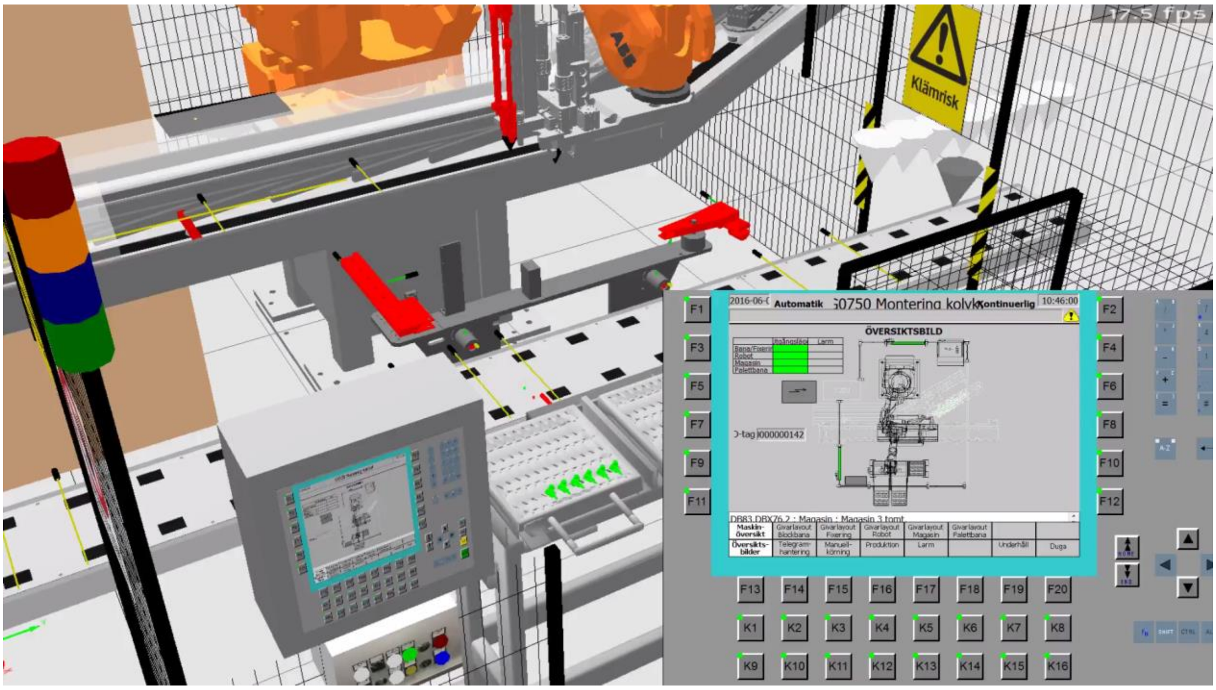
https://www.youtube.com/watch?v=cqc74l1B5mk



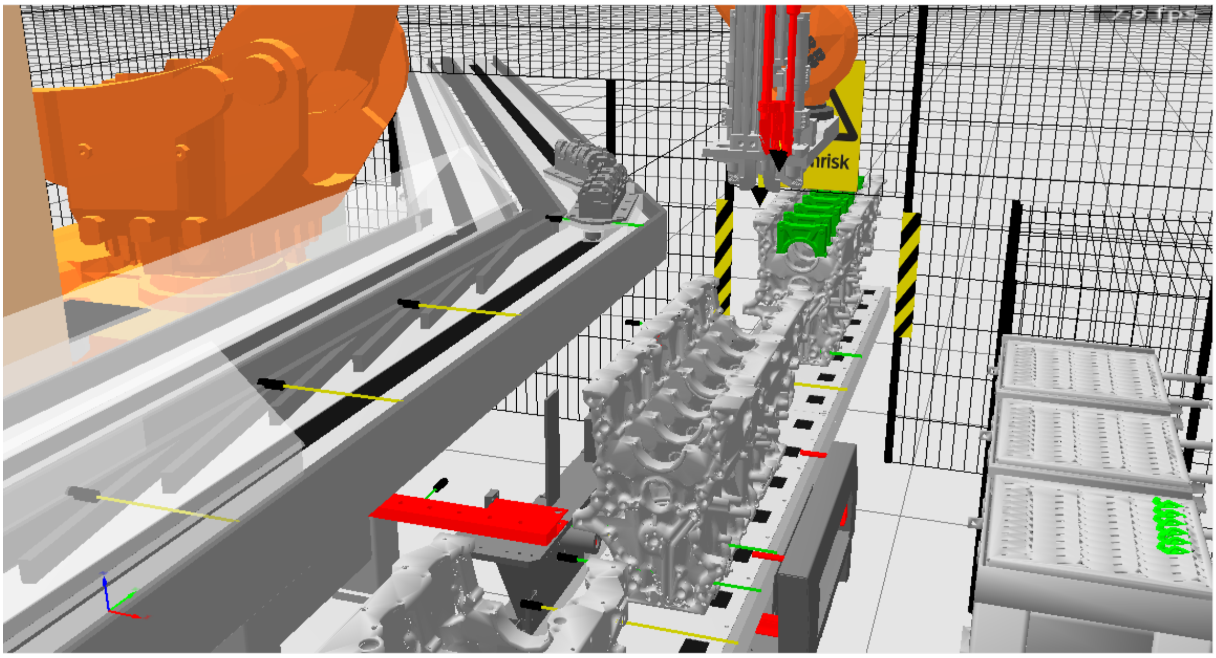
Appendix E – *Missing main cap*

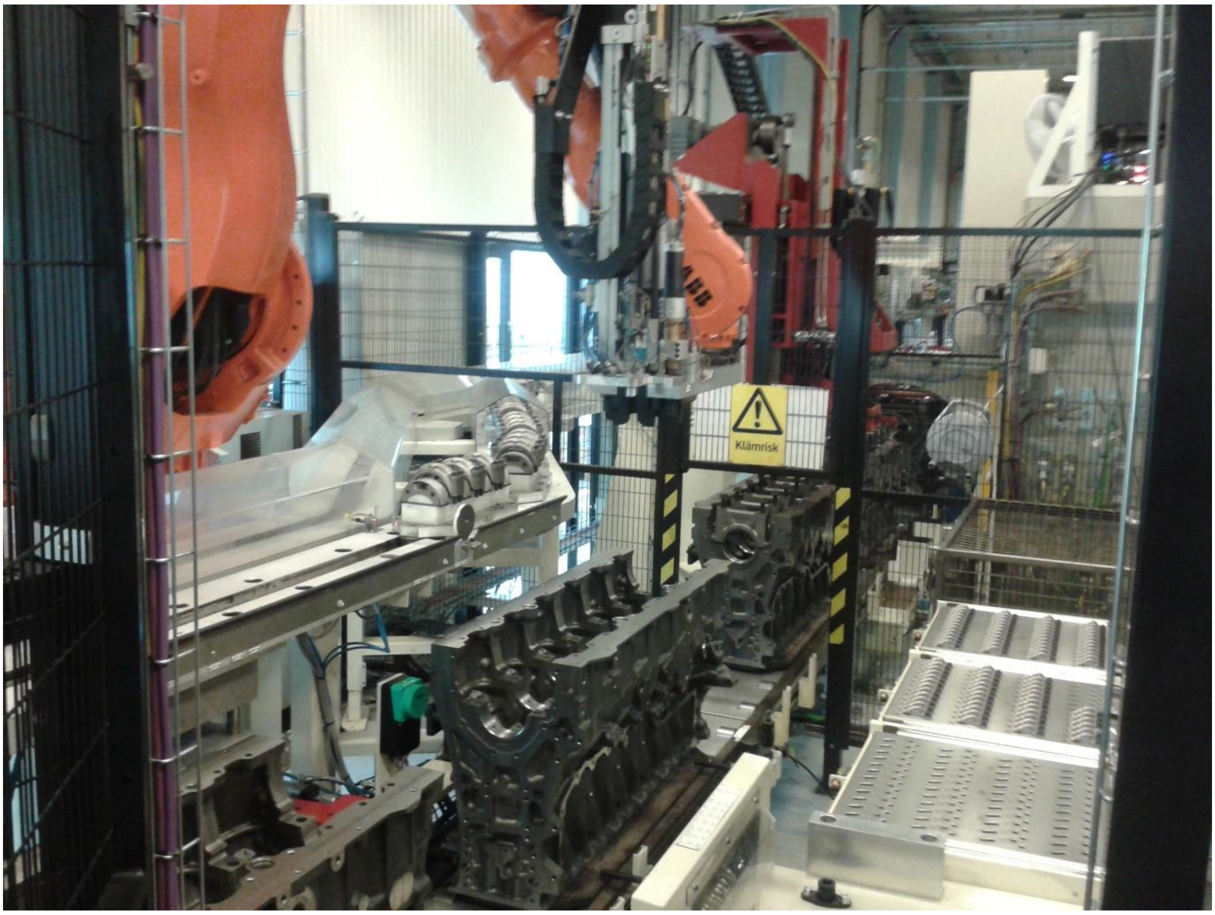
This video shows the experiment presented in 8.1.3

https://www.youtube.com/watch?v=a5ixyyPKle0



Appendix F – *Similarities of real production cell and emulation model*





Appendix G– *Full production cycle*

This video shows a full production cycle of the emulation model.

https://www.youtube.com/watch?v=M0QaNiWBOQ0