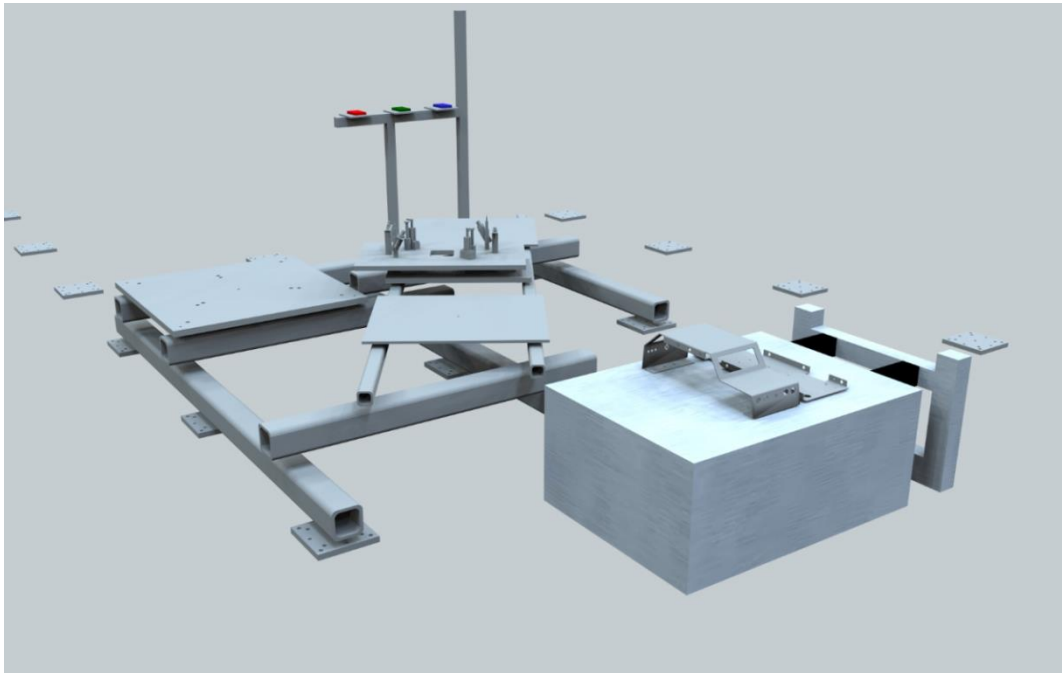


# CHALMERS



## Simulation-based verification of PLC programs

*Master of Science Thesis in Production Engineering*

JASMIN DZINIC

CHARLIE YAO

Department of Signals and Systems  
Division of Automatic Control, Automation and Mechatronics  
Research Group Automation  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2013  
Report No: EX058

REPORT NO. EX058/2013

# Simulation-based verification of PLC programs

JASMIN DZINIC  
CHARLIE YAO

Department of Signals and Systems  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2013

Simulation-based verification of PLC programs

JASMIN DZINIC

CHARLIE YAO

© JASMIN DZINIC, CHARLIE YAO, 2013

Technical report no **EX058/2013**

Department of Signals and Systems

Chalmers University of Technology

SE-412 96 Gothenburg

Sweden

Telephone: + 46 (0)31-772 1000

Cover:

The cover picture shows the Automatic zone in the Production System Laboratory at Chalmers. The view is a screen shot from Xcelgo Experior environment.

Gothenburg, Sweden 2013.

# Abstract

Automation systems have in recent decades increased in complexity and variety in which the demand for safer and faster commissioning has been vital for the industry. The current approach for commissioning is based on a traditional level in which control systems are verified using a trial-and-error based method in a workshop and/or directly in the production facility. The traditional level of commissioning is a time consuming and costly approach and a search for another method has been a must in order to meet the current demand.

Researchers have suggested virtual commissioning which is a way to verify control systems in a virtual environment prior to physical commissioning. Industrial projects based on virtual commissioning have two main approaches for commissioning; Software-in-the-Loop (SIL), where commissioning of control system and the plant is performed in a virtual reality, and Hardware-in-the-Loop (HIL), where commissioning is instead performed on a real control system and a virtual reality. Both SIL and HIL are studied in this project due to the need for the industrial automation company Teamster to verify PLC programs in-house in an early commissioning phase and thereby minimize the commissioning time at the production facility.

In this project, authors have evaluated the possibility to verify PLC programs by setting up a VC project with four main steps; 3D model development, PLC programming, simulation model development and communication between simulation model and the PLC program. Tools used in this project were SketchUp for 3D modeling, Experior for simulation model development and Siemens TIA-Portal V11 for PLC programming.

**Keywords:** *Virtual Commissioning, Programmable Logic Controller, Hardware-in-the-Loop, Software-in-the-Loop, 3D-Modeling, Simulation, Emulation.*

# Acknowledgements

We would like to thank the persons and companies that have contributed with knowledge and thus making this project possible.

- **Kristofer Bengtsson**  
Doctor in the Automation research group at Chalmers University of Technology, Kristofer has been our supervisor at Chalmers and contributed with great ideas for the project. He has also contributed with knowledge and support about the topic in this project.
- **Knut Åkesson**  
Associate professor at Chalmers University of Technology, Knut has been examiner of the thesis report and served us with great ideas and support.
- **Thomas Lydhig**  
Manager for Robot Automation at Teamster AB, Thomas has been our supervisor at Teamster AB and served as a sounding board for all type of ideas and problems.
- **Peter Olsson**  
PLC programmer at Teamster AB, Peter has been our closest contact at Teamster in which he acted as great contribution regarding Experior, PLC programming and the communication between PLC and Experior.
- **Bent Aksel Jørgensen**  
Director at Xcelgo A/S. Bent have been a sounding board between Chalmers and Xcelgo. He has served us with knowledge and expertise about Experior.

Jasmin Dzinic & Charlie Yao, Gothenburg, 2013

# Contents

<b>1. Introduction.....</b>	<b>1</b>
<b>2. Virtual Commissioning.....</b>	<b>5</b>
2.1. Simulation vs. Emulation .....	5
2.2. Concept of VC.....	6
2.3. Data requirements .....	8
2.4. VC and simulation tools.....	8
<b>3. Xcelgo Experior .....</b>	<b>11</b>
<b>4. VC Workflow.....</b>	<b>14</b>
4.1. Data collection .....	14
4.2. 3D model development .....	14
4.3. PLC programming and I/O signals .....	16
4.4. Simulation model development .....	16
4.5. Establishment of communication.....	17
<b>5. PSL Case Study.....</b>	<b>18</b>
5.1. Actual PSL production cell.....	18
5.2. Virtual PSL production cell .....	19
5.2.1. 3D model development .....	20
5.2.2. PLC programming and I/O signals.....	21
5.2.3. Simulation model development .....	22
5.2.4. Establishment of communication .....	22
5.3. Conclusion Case Study .....	24
<b>6. Evaluation on Research .....</b>	<b>25</b>
<b>7. Conclusion &amp; Future Work .....</b>	<b>28</b>
<b>References .....</b>	<b>30</b>
<b>Appendix A.....</b>	<b>31</b>
A.1 Material flow of automatic zone in PSL .....	31

A.2 Inputs/Outputs of the PLC program.....	32
A.3 Plugins used in SketchUp .....	33

# List of Figures

<i>Figure 1.1: Approaches of commissioning: 1: Traditional, 2: HIL, 3: RIL, 4: SIL.....</i>	<i>2</i>
<i>Figure 1.2: Teamster's current workflow .....</i>	<i>3</i>
<i>Figure 2.1.1: Concept of simulation and emulation of systems .....</i>	<i>6</i>
<i>Figure 2.2.1: Software-in-the-Loop .....</i>	<i>7</i>
<i>Figure 2.2.2: Hardware-in-the-Loop .....</i>	<i>7</i>
<i>Figure 2.4.1: Workflow for CAD models .....</i>	<i>9</i>
<i>Figure 3.1: Example of components in Experior component library.....</i>	<i>11</i>
<i>Figure 3.2: Available panes in Experior.....</i>	<i>12</i>
<i>Figure 4.1: VC project setup.....</i>	<i>14</i>
<i>Figure 4.2.1: Swing clamp cylinders before and after model simplification .....</i>	<i>16</i>
<i>Figure 5.1: Actual PSL production cell .....</i>	<i>18</i>
<i>Figure 5.1.1: Representation of the minicar and surrounding equipment.....</i>	<i>19</i>
<i>Figure 5.2.1: Virtual PSL production cell.....</i>	<i>20</i>
<i>Figure 5.2.4.1: Communication between software PLC and Experior.....</i>	<i>23</i>
<i>Figure 5.2.4.2: Communication approach for HIL and SIL .....</i>	<i>23</i>
<i>Figure A.1.1: Flowchart of the Automatic zone in PSL .....</i>	<i>31</i>



# List of Tables

<i>Table A.2.1: Input-list of the PLC program .....</i>	<i>32</i>
<i>Table A.2.2: Output-list of the PLC program.....</i>	<i>32</i>

# Abbreviations

<b>3D</b>	3 Dimensional
<b>AGV</b>	Automated Guided Vehicle
<b>API</b>	Application Programming Interface
<b>C#</b>	C-Sharp
<b>CAD</b>	Computer Aided Design
<b>COLLADA</b>	COLLABorative Design Activity
<b>CPU</b>	Central Processing Unit
<b>HIL</b>	Hardware-in-the-Loop
<b>HMI</b>	Human Machine Interface
<b>I/O</b>	Input/Output
<b>PLC</b>	Programmable Logic Controller
<b>PSL</b>	Production System Laboratory
<b>RIL</b>	Reality-in-the-Loop
<b>SIL</b>	Software-in-the-Loop
<b>TCP/IP</b>	Transmission Control Protocol/ Internet Protocol
<b>TIA</b>	Totally Integrated Automation
<b>VC</b>	Virtual Commissioning

# 1. Introduction

This master thesis has been conducted at Chalmers University of Technology, in cooperation with the industrial automation company Teamster AB. This chapter aims to give the reader a clear understanding of the topic in the thesis. The purpose of the project, the goals to reach and the delimitations are presented.

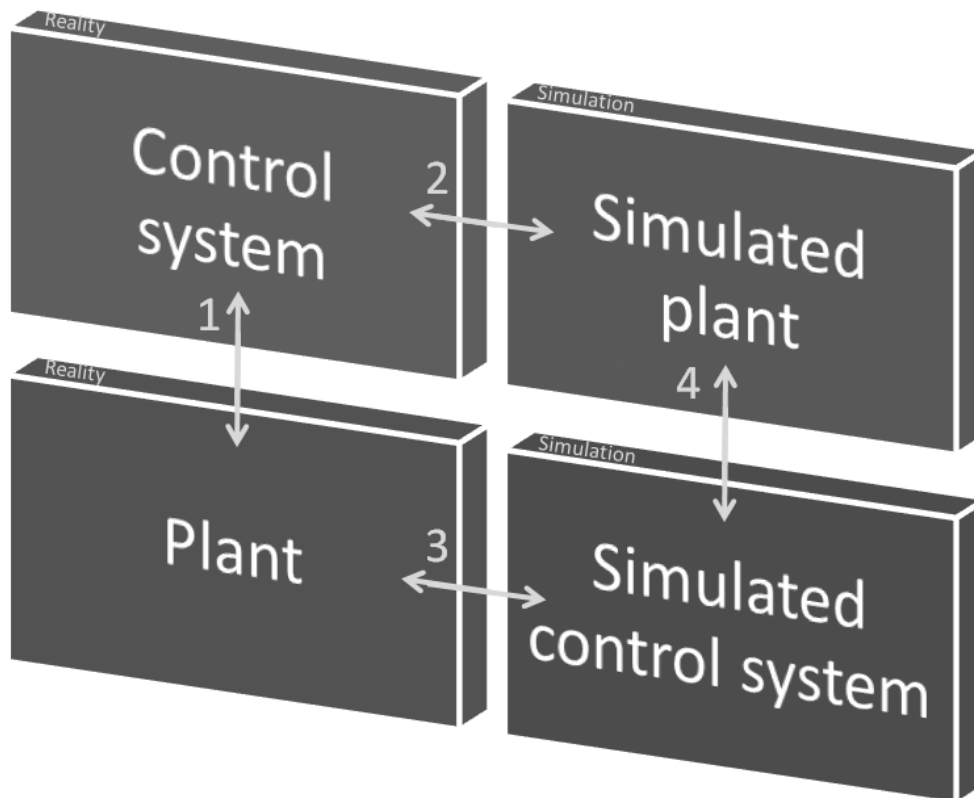
Today, manufacturing plants are well known as being mass production systems with highly automated workstations and fast operating systems. In order to meet customer demands on quality, reliability and delivery; modern manufacturing systems must enable cost effective, fast and flexible production. Hence, manufacturing systems would emphasize higher production rates, increased productivity and efficient use of materials. Modifying and controlling workstations with respect to changing requirements becomes a great challenge.

Traditionally, development of manufacturing systems has been performed sequentially in different phases; facility design, mechanical engineering, electrical engineering and automation engineering [1]. Currently, documentation of all engineering phases is mostly performed separately and partly on-line, where the automation engineer has to wait with the programming and verification of the control programs, Programmable Logic Controller (PLC), until machines and surrounding equipment are deployed in the plant. The complexity and diversity of the line components in term of PLC programs and communication channels requires longer ramp up time for testing and validation of production equipment. This in turn translates into eventual design problems and programming errors which lead to unnecessary time delays and extra costs [1], [2].

Over the last two decades, simulation of automated manufacturing systems has emerged with a purpose to replicate the real plant and virtually verify such systems prior to the installation [3]. A common name for this virtual modeling is virtual commissioning (VC). Intention with VC projects is to give engineers an advantage by working together with the same model at the same time and in an early design phase. One of the most recent and popular features of VC is the ability to include mechatronic behavior of resources and include verification of a PLC program in a virtual environment prior to physical commissioning. From this, one can verify if it is feasible to confirm the required behaviors of machines and the surrounding equipment. As a prominent expectation to combine reality with the virtual world, Smith et al. [4], Hoffmann et al. [6] and Erlandsson et al. [7] explain that PLC programs could be verified with four possible approaches for commissioning (Figure 1.1) and these are further described as:

- (1) Traditional commissioning with both control system and automation system in real world.

- (2) Hardware-in-the-Loop (HIL) with real PLC system and simulated automation system.
- (3) Reality-in-the-Loop (RIL) with simulated PLC system and real automation system.
- (4) Software-in-the-Loop (SIL) with both systems simulated.

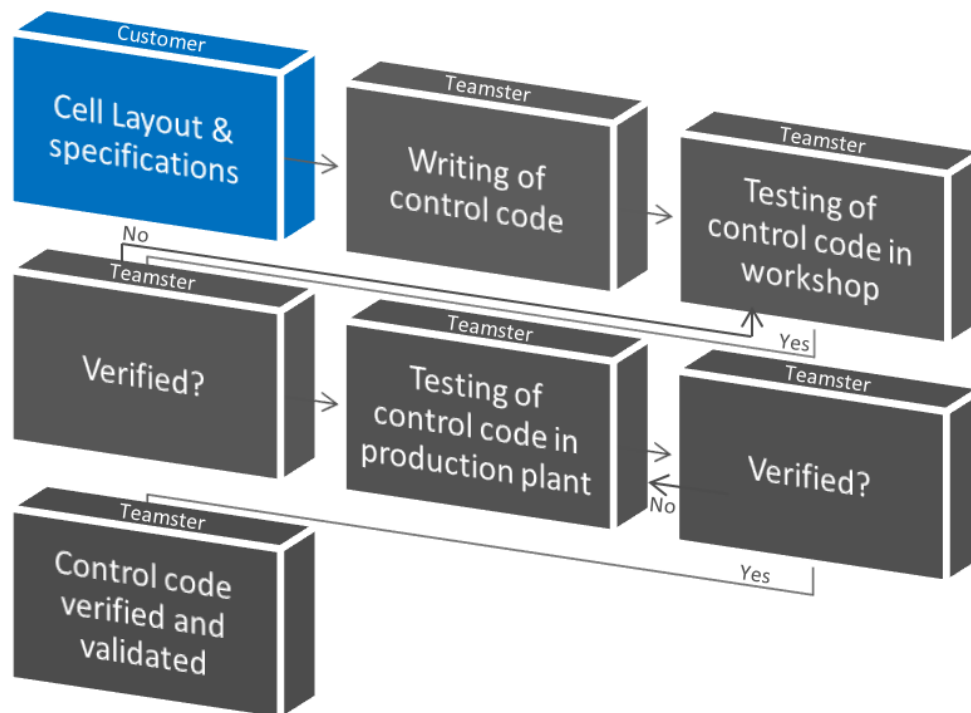


**Figure 1.1:** Approaches of commissioning: 1: Traditional, 2: HIL, 3: RIL, 4: SIL.

Traditional testing (1) involves an error-prone testing method where a PLC controller is directly connected with a physical hardware. The main disadvantage with this approach is that it involves great risk with added costs and system downtime by using a real process system during the verification. The disadvantage is also the case for RIL (3). According to [5] and [6], HIL (2) and SIL (4) are much safer and less costly approaches where virtual reality is used instead of real world systems. These methods entail minimized amount of errors on the PLC program, less risk of damaging equipment, shorter commissioning time and increased quality of the manufacturing system. Therefore, HIL and SIL are of great interest in this thesis.

Teamster AB is a small industrial automation company from Gothenburg, Sweden. Their current workflow process (Figure 1.2) includes a sequential development of manufacturing systems with on-line testing of machines and PLC programs. A PLC program is verified in a traditional way (1), whereby adjustments are performed with a trial and error method. Therefore, Teamster is highly interested in using HIL and SIL in which PLC programs, resources and equipment could be verified in-house prior to the

physical commissioning and thereby also increase the possibility to decrease commissioning time on site.



**Figure 1.2:** Teamster's current workflow.

As an objective to Teamsters needs, this thesis will discuss the challenges and possibilities of verifying PLC programs in a virtual reality. Research questions that need to be answered during this thesis are therefore:

- (1) Can a PLC program be verified by using a simulation software tool such as Expor from the Danish company Xcelgo?
- (2) How to setup a project for verification of PLC programs?
- (3) How many approaches are there for VC and can all be used for verification of PLC programs?
- (4) Can the simulation software tool interact with SketchUp used in Teamsters current workflow?
- (5) What knowledge is required for simulation model development?

Towards these research questions, a complete method of virtual commissioning complemented by various selected and capable software tools are being further discussed. In order to provide as real verification of the PLC program as possible, a case study at Chalmers Production Systems Laboratory (PSL) is presented. PSL consists of a car assembly cell with the purpose to assemble different modules and car bodies into a finished car. The cell consists of a fully automated manufacturing system including PLC, interacting robots, sensors and actuators. Due to the high complexity of the production cell and limited time frame, main focus will be on cell design, sensors,

actuators, PLC program, simulation and verification of the PLC program. Robots and corresponding robot program are therefore not included in this project. The PLC program will be written in same PLC software tool that is used in the PSL, i.e. Siemens TIA-Portal V11 (hereinafter TIA-Portal). The PLC program will then be verified by integrating the TIA-Portal to the simulation software tool Experior. Before the verification can be performed, the simulation model first needs to be developed in Experior. Mostly will the simulation model be developed directly in Experior but a part of it will also be developed in Visual Studio by using the programming language C# which is supported by Experior. Also all 3D models will first be created in SketchUp before being imported into Experior. Finally when the simulation model is developed, the integration between Experior and TIA-Portal will be performed in order to verify the PLC program. The whole verification procedure and the setup for the project are discussed in following chapters:

- (1) **Chapter 2:** Background of virtual commissioning and the corresponding approaches for verification of PLC programs. Also, a VC workflow will be introduced with information about different software tools that are used in this project.
- (2) **Chapter 3:** Extending information about the simulation software tool Experior.
- (3) **Chapter 4:** All steps needed in order to perform a VC are further described in this chapter.
- (4) **Chapter 5:** Verification of a PLC program is based on the case study performed in the PSL at Chalmers. This chapter discusses PSL and the virtual replication of PSL based on different steps in the VC workflow.
- (5) **Chapter 6:** An evaluation on listed research questions is represented in this chapter.
- (6) **Chapter 7:** A conclusion of this thesis with future recommendations is discussed.

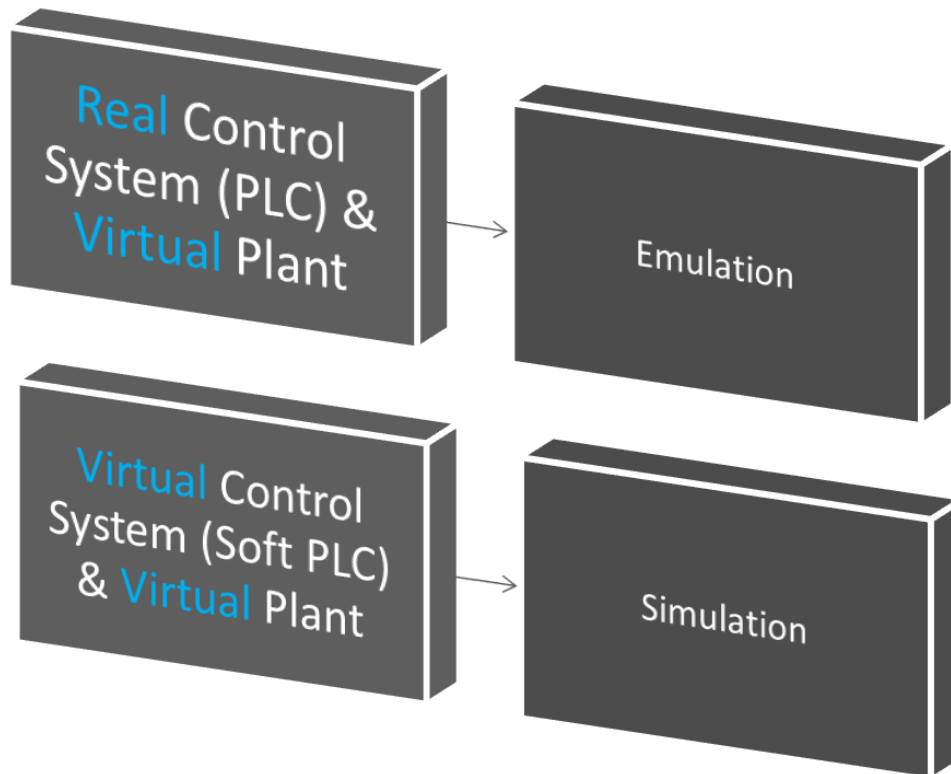
## 2. Virtual Commissioning

This chapter gives a theoretical background about virtual automation systems, detailed description about VC and corresponding approaches, SIL and HIL. This chapter focuses on the procedure of PLC program verification by using the concept of VC. The procedure considers 3D model development, PLC programming and I/O signals, simulation model development and establishment of communication between the control system and the virtual plant.

### 2.1. Simulation vs. Emulation

In the evolution of industrial science, simulation and emulation has become vital for virtual commissioning processes. Testing of real world automation systems could be done virtually, mimicking a real automation system [7]. This project documents current thinking on creating simulation and emulation models to provide feedback to industrial PLC programs in order to test their logical operation prior to commissioning. Both approaches are trying to create the same output as in a real world automation system, but with emulation, testing of PLC programs goes a step further by minimizing the gap between reality and the virtual world. Simulation is more focused on approximations of a real system, where a process of a virtual system is not necessarily similar to a real system, while emulation is focusing on creating a closer interaction between reality and the virtual model by using a part from the real system [8].

As described in chapter 1, there are four methods for testing control systems using a combination of simulation models and real world objects but there are two methods that gain most interest in this project. Method 2 (HIL) describes the emulation and the soft-commissioning and method 4 (SIL) represents the simulation and the in-house commissioning. Figure 2.1.1 describes these two approaches as a combination between real and virtual representation of PLC and production cell equipment.



*Figure 2.1.1: Concept of simulation and emulation of systems.*

## 2.2. Concept of VC

VC projects are performed in order to verify PLC programs in a simulation environment prior to the physical implementation for which all resources are modeled virtually without any risk of endangering lives and equipment. Testing of PLC programs can provide decision making support for different engineering fields and real time communication between PLC and simulation model. According to [1], a VC project involves three distinct factors:

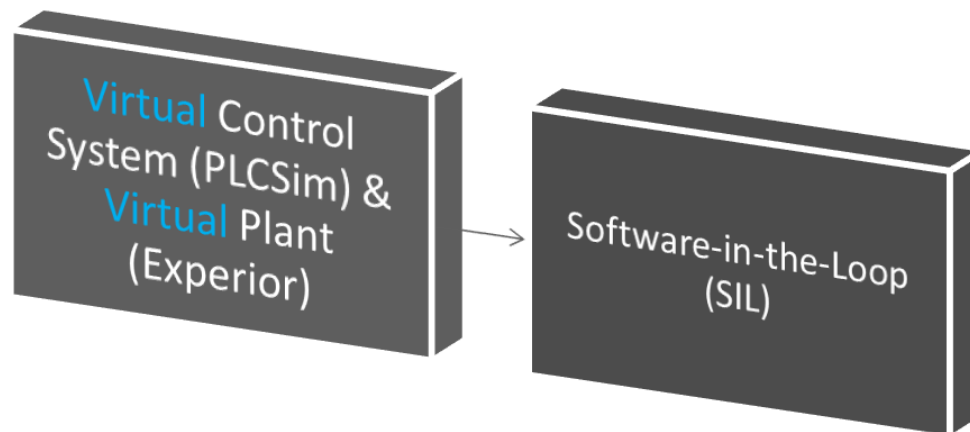
- (1) Mechatronic design including actuators, sensors and functional descriptions
- (2) Machine control including inputs and output signals
- (3) Connection between sensors, actuators and the control program.

All factors need to be identical to the real automation system for controlling the simulated production cell. Mechatronic design provides the best solution for modeling of mechanical and electrical behaviors, whereby a virtual simulation model can represent sensors and actuators in an extensive and comprehensive manner. Still, a virtual model might not behave exactly as real automation system due to; lack of understanding about the real system, inability to correctly model the problem and inability to correctly translate, or code, the conceptual model into a computerized model



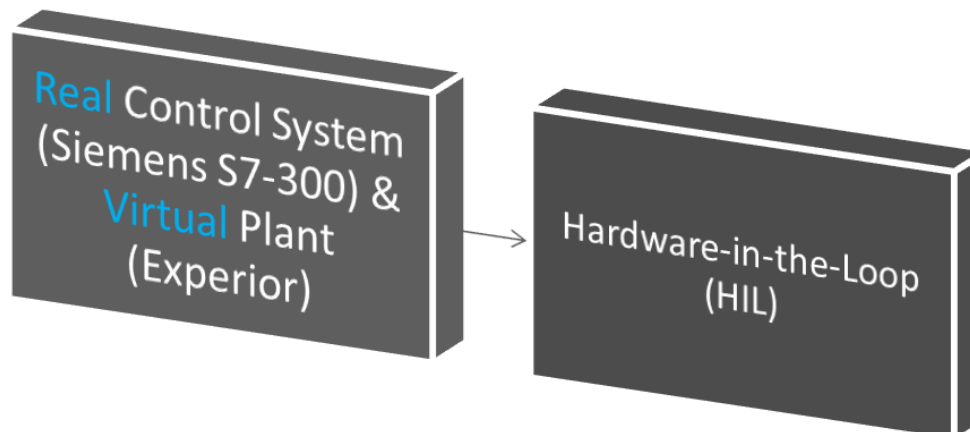
[9]. Therefore, one is guided to first match the intended use of its most typical parts related to the three factors described above.

According to [1], there are two current approaches to build a VC project, SIL and HIL. The first approach, SIL (Figure 2.2.1), also called offline-programming, is based on total virtual modeling where both PLC program and the plant are simulated. The advantage with this method is that a complete VC project can be offered with no hardware required during the design and validation of control programs.



*Figure 2.2.1: Software-in-the-Loop.*

The second approach, HIL (Figure 2.2.2), also called as soft-commissioning, is a real-time simulation method where real PLC hardware is interacting with a virtual plant. This method enables the user to verify and validate more complex control programs for different plant levels, such as line, cell or plant.



*Figure 2.2.2: Hardware-in-the-Loop.*

Both SIL and HIL provide real time simulation with corresponding control programs and equipment by using appropriate communication protocols. Some differences and errors can occur between the real plant and virtual simulator which can lead to far deviations on the result of the simulated model. The results that one might see as good in the simulated model, could be wrong when implementing the system in real world. Therefore, Dominka et al [10] have suggested hybrid commissioning, where

verification of PLC programs are provided either by SIL or HIL, then a mixture of a virtual and real plant in order to finish with a final commissioning of the real plant and the real system. This approach could thereby minimize the risk of having misleading results on the simulation model.

## **2.3. Data requirements**

In order to realize a VC project, one has to follow specific data requirements that involve:

- (1) 3D models of resources and equipment belonging to the production cell including geometries, kinematics, electrics and PLC programs.
- (2) Layout of the production cell including exact placement of resources and equipment.
- (3) Material flow including sequences of operations.
- (4) PLC programs created and verified with software and/or hardware PLC.
- (5) Mapping of Input/Output (I/O) signals.
- (6) Detailed definitions of extra functionalities such as alarms.
- (7) Communication protocols and software drivers, such as TCP/IP protocols for communication between PLC and a simulation model.

As a VC project is setup, further information about project objectives regarding verification of PLC programs are required.

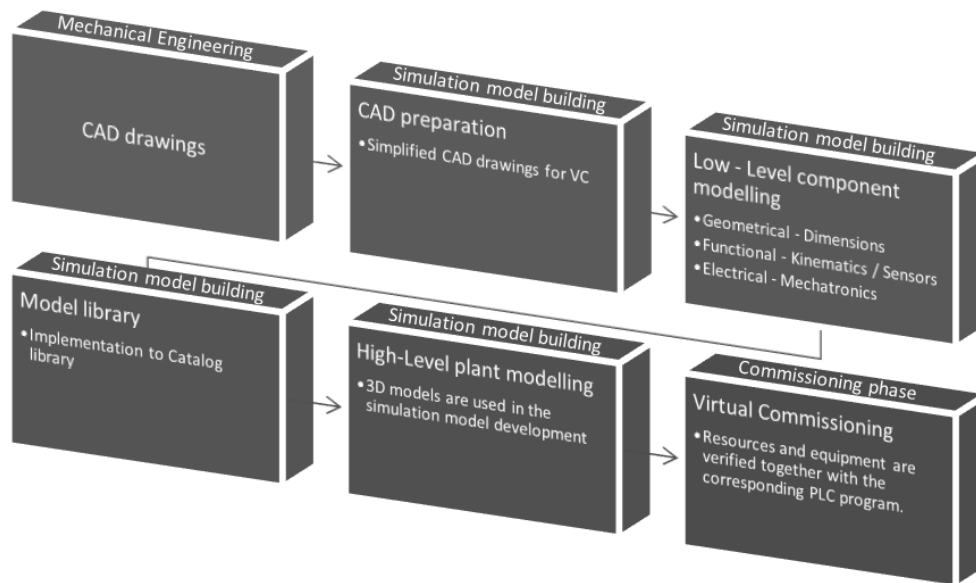
## **2.4. VC and simulation tools**

One of the requirements to start a VC project is to use 3D models in simulation software. According to [3] the simulation model building process is separated into two different modeling tasks:

- (1) Component modeling (Low-level).  
In the beginning of a VC project, all required component models are not always directly available in the simulation tool component library. These components have to be created and later on imported into the simulation tool component library. Remarkable efforts become necessary to build these additional models from available CAD tools.
- (2) Plant modeling (High-level).  
Many simulation tools (e.g. Experior) have as pre-defined component library with various component models available for the user. For these models, it is required to have all needed information in order to create the simulation model needed. It is then relatively easy to set up a mechatronic plant model for VC, but even if high-level plant modeling of a simulation model may be partially based on the pre-defined component library

components: Exact placement and the interconnection of the components require additional effort.

For the mechanical engineering phase, most components that are usual industry components are often received directly from suppliers with corresponding 3D CAD data. This is often not the case for all components and a low-level component modeling procedure (Figure 2.4.1) must be carried out by the user. This scenario complicates simulation model building for VC considerably and requires specific modeling expertise.



**Figure 2.4.1:** Workflow for CAD models.

In this project, both high-level and low-level procedures are of interest. While high-level plant modeling is often included in the simulation software tool, low-level plant modeling has to be provided by an external 3D modeling tool such as Catia, AutoCAD or SketchUp. In this project, SketchUp is chosen for low-level plant modeling due to following reasons:

- (1) SketchUp is included in Teamster for making 3D models because it is a free, easy learning tool with a wide range of resources available in the net.
- (2) SketchUp is supporting various file formats from different CAD software, i.e. models created by other CAD tools can easily be imported and modified in SketchUp.
- (3) A 3D model can easily be simplified in SketchUp with user friendly functions and plugins that later can be exported into various types of file formats such as Microsoft .X and COLLABorative Design Activity (COLLADA).
- (4) Working with PLC resources that has to separate static parts with moving parts, SketchUp is a tool that is able to do this and also to test the dynamics of resources by using a physics engine.

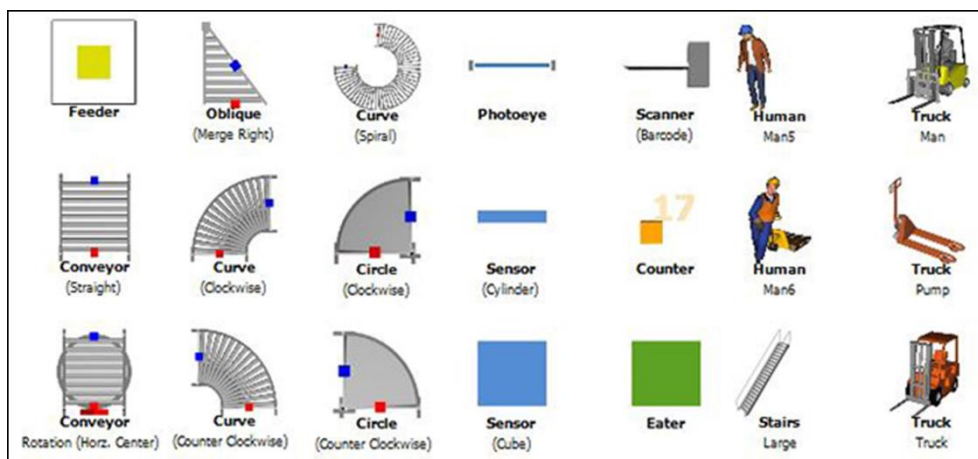
For high-level plant modeling and implementation of a VC project, there are many simulation software tools available on the market. Most simulation software provides a built-in library with high-level plant modeling components. Those models also all have different properties in order to create a fully working simulation model, and some simulation software even use a physics engine to simulate the mechanics of models. Tools that provide most of these behaviors are Delmia, WinMOD, Invision, 3D Emulator, Process Simulate, Experior etc. In this project Experior is chosen due to following reasons:

- (1) Generally, there is a limitation for small and medium-sized companies to use VC solutions because of lacking resources to start solving problems with a VC solution [3]. Xcelgo provides state-of-the-art VC solutions with their software tool Experior, and focuses on all companies including small companies such as Teamster [2], [5].
- (2) Teamster uses SketchUp in their current workflow which gives the opportunity to save CAD models in various types of file formats including .X and COLLADA. Both of these file formats are the main file formats for importing 3D models into Experior.
- (3) Experior gives the opportunity to communicate with Siemens software/hardware PLC which is used in both Teamster and this project case study.

### 3. Xcelgo Experior

Experior is a simulation-based verification tool developed by Danish company Xcelgo. Experior provides virtual automation solutions that for modeling and verifying control systems as well as providing a solid foundation for various applications.

Experior offers a built-in library consisting of components such as; conveyors, sensors, feeders, lamps and switches (Figure 3.1). The library is especially handy for situations when interaction between humans and machines (Human Machine Interface (HMI)) are required. One can, for example, create a control panel in Experior with buttons so that the user can directly interact with a machine during the simulation. The Experior HMI becomes especially useful in the case of SIL, where no actual hardware is available for the verification of PLC programs. Another example of using the library is during simulation of a packaging line where transportation, sortation and packing are vital. Components connected to the packaging line such as conveyors, motors, sensors, lamps, pallets, actuators etc. could then easily be found in the component library.



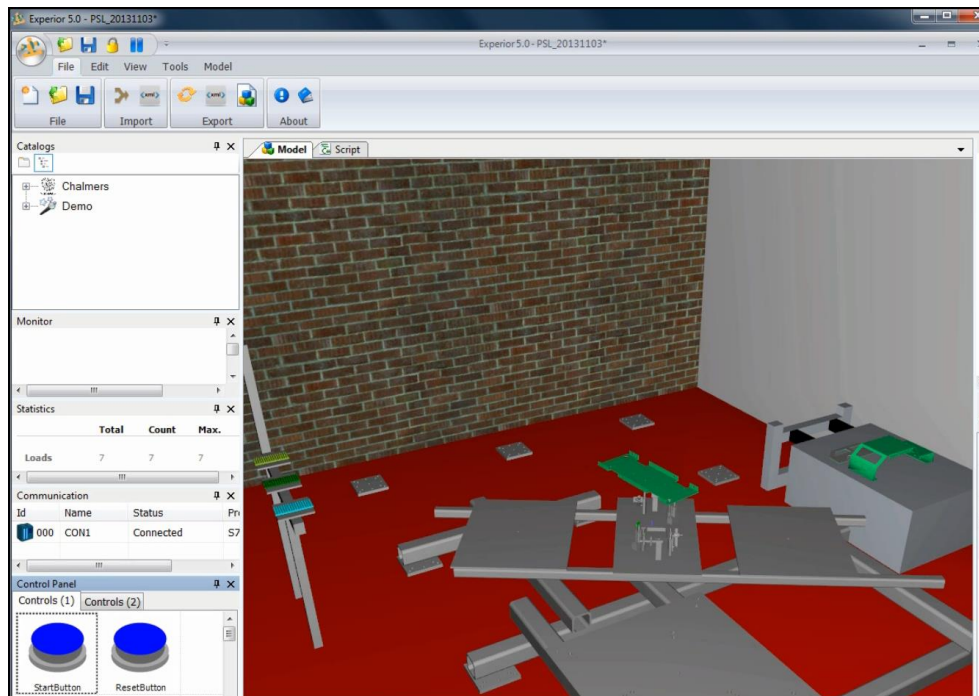
*Figure 3.1: Example of components in Experior component library.*

Experior uses .X and COLLADA file formats when importing 3D models to the simulation environment. Models with those file-formats are able to use 3D rendering which is a process of creating animation in real time simulation. It is also where the dynamics of different rigid parts are appended and required to make these parts possible for moving on a route with a function of collision detection and thereby give faster response in order to withstand with real time environment. In Experior, one can simulate mechanics of rigid parts, soft elements and particles as an intuitive graphical visualization by using NVIDIA PhysX. PhysX is a real-time physics engine that provides an approximate simulation of certain physical system and supports rigid part dynamics [7]. Properties like friction, mass, color and functions like collision detection are a few of the things that PhysX provide.

Experior is very user friendly with most elements provided directly in the main Experior window. But, Experior is also supporting the Visual Studio framework .NET

and the programming language C# for further development of simulation models. The developers are able to create custom components and simulation behavior in C# by using Experiors Application Programming Interface (API). The developer has also the availability to use several industrial communication protocols from which communication with external PLCs can be established.

Exporior offers various panes which are used for displaying different information about the current simulation model. Panes such as the catalogs pane, communication pane and a working area pane (Figure 3.2) are few of many that are available.



*Figure 3.2: Available panes in Exporior.*

All panes used in this project are described as follows:

- (1) **Catalogs:** All parts for building a layout or a simulation model are located. The pane includes both Exporior's own catalog and own created catalog. All models are a part of the library.
- (2) **Working area:** This is the main screen where the construction of the model is taking place and models from the catalogs pane can be added and worked with.
- (3) **Statistics:** One can display information about the current number of loads that has been added to the production system.
- (4) **Communication:** This pane contains all PLC controls and communication protocols.
- (5) **Controls:** This is where all controls such as buttons, switches and lamps are located.
- (6) **Solution Explorer:** All assemblies that are added to the working area appear in Solution Explorer.

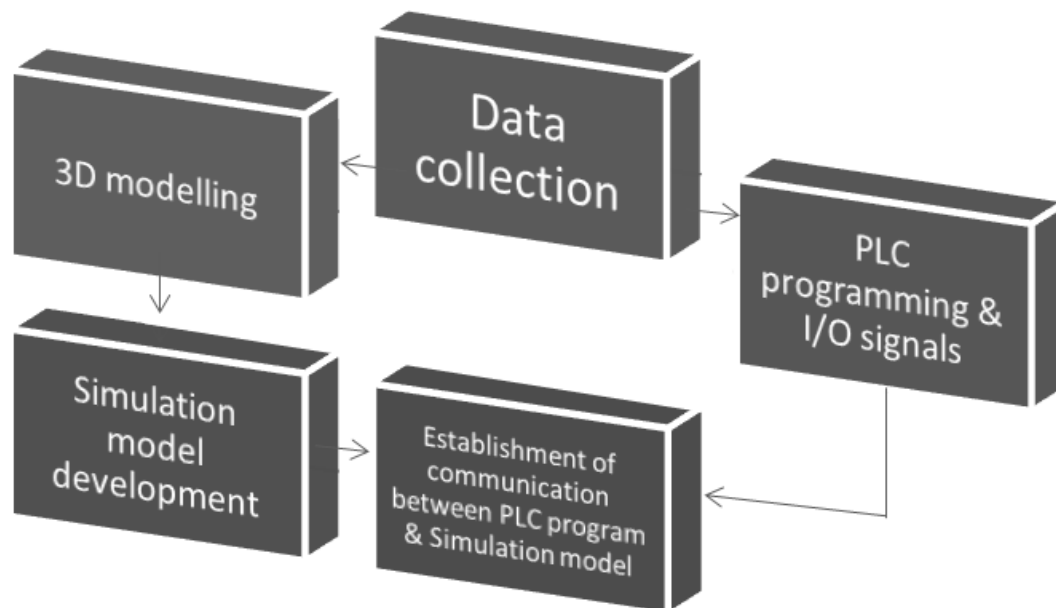
- (7) **Properties:** This pane displays properties available for resources that are included in the simulation model. Some examples of properties are size, orientation, position, visualization, PLC I/O's and behavior.
- (8) **Logs:** This pane gives the opportunity to record everything that is happening during simulation. Here, one can see everything being added to the simulation model and the amount of time the simulation has been running.
- (9) **Inputs/Outputs:** This pane displays all PLC symbols and information that has been assigned to various resources within the model.
- (10) **Alarm:** Here will all alarms in the system be appended with corresponding connections.
- (11) **Monitor:** This pane will display information about I/O assigned to a particular resource. For example, if one wants to follow a motor with its corresponding I/O assignment related to it, one can see the action on the monitor view.

One can also easily view, edit and save their current simulation model and having direct access to Experior wiki in which all information is explained. The menu list which includes all information above is very useful for the user while creating their simulation model. These are described as follows:

- (1) **File:** This section provides basic functions for opening and saving files, but also a quick access to Experior wiki in which all information about Experior and how to create a simulation model is further explained.
- (2) **Edit:** This section provides commands for undoing and re-doing actions, copying and pasting functions as well as selecting items.
- (3) **View:** In this section, one has the ability to choose which panes one needs to show up in the main Experior window. Also, the ability to import 2D files for the working area is available where, for example, a floor of a production system can be added. Beside those two important view options, one can also use different angles by using camera view of the model, setting color range of the model as well as choosing the visibility for various resources and equipment.
- (4) **Tools:** This section consists of various tools in which the user can, for example, record and take screen shots of the simulation model, import static parts directly in main Experior window and also choose different move and rotation options for the mouse.
- (5) **Model:** This section allows the user to run the simulation model in different speed as well as pausing and resuming the process. Another option that one can get benefit from is that options for manually moving loads could be set to allowed or disallowed. One can also choose if loads should act as they stack up in the model or if they just disappear after the corresponding process.

## 4. VC Workflow

This chapter aims to provide a description of VC workflow for realization of a VC project. In [1] and [12] necessary steps for VC is defined. In Figure 4, it is shown how the VC project is set up which includes data collection, 3D modeling, creating PLC programs and belonging I/O signals, developing a simulation model and HMIs and finally use communication protocols in order to establish communication with software/hardware PLC.



*Figure 4.1: VC project setup.*

### 4.1. Data collection

The first step in the process of virtual commissioning is to collect data from previous work or from the vendor. All collected data reflected the first three points in chapter 2.3, i.e. 3D models representing resources in PSL, layout of the production cell and sequence of operations. All these data were then utilized in the PLC programming development phase and the simulation model development phase.

### 4.2. 3D model development

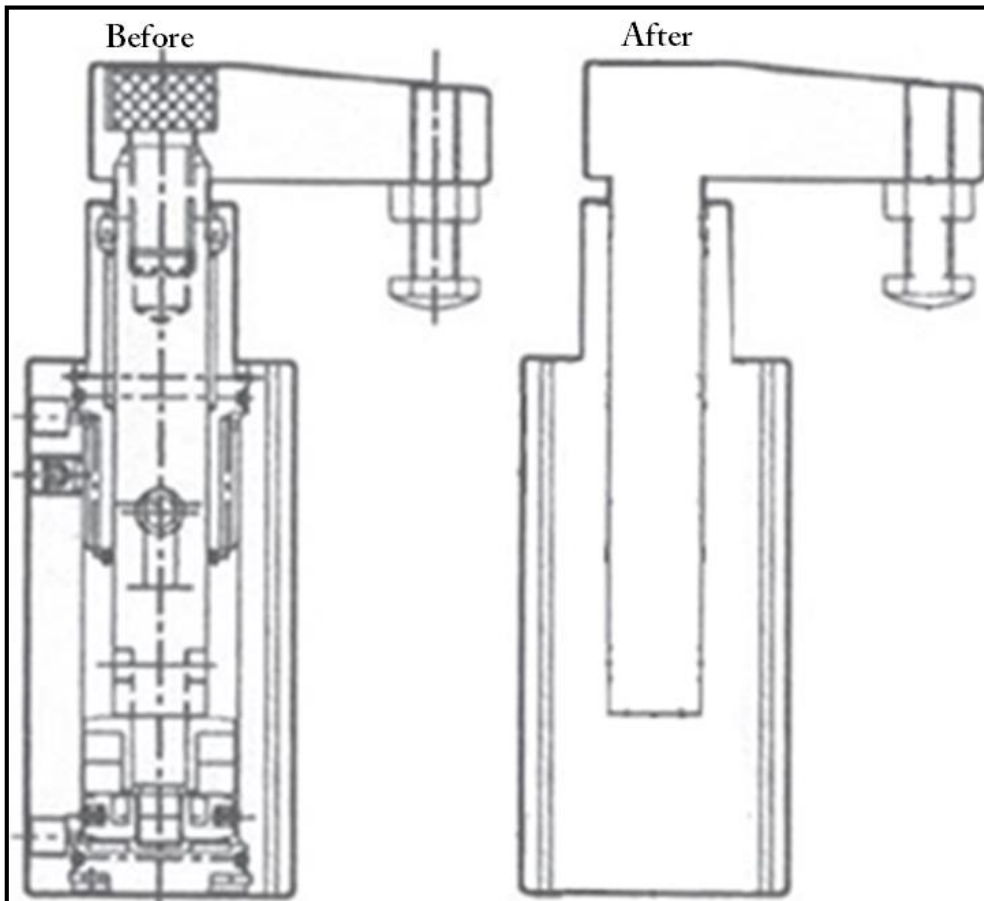
The second step in the process of virtual commissioning is to develop 3D models for replication of the production system. Some components are already provided directly in Experiors' built-in component library (high-level plant modeling) while other components need to be created and imported from external 3D modeling tools (low-level component modeling) such as SketchUp. In general, these components have



already been created in CAD software by its producer with too complex geometrical detail information for use in VC simulation. Consequently, models will increase the CPU usage so that it can affect the fluidity of simulation. This is why thorough model analysis and a simplification are necessary in order to decrease the CPU usage and calculations.

Important data information about 3D models are; model format, geometry, color, the positioning and modeling of resources, definition of degrees of freedom and constraint definition. Other data valid just for rigid parts is friction and collision detection which creates the ability for the model to achieve movements and assembly. These data have to be collected and prepared for use in the simulation model. One must also collect all equipment and resources not available in the built-in library to create and reduce the number of details (geometrical elements) in SketchUp, which in term reduces the number of facets in the simulation model. Beside these requirements for proper 3D model development, one must also separate components with three separate structures; static units, moving units and sensors. These structures are later separated in different files and imported into the simulation software tool.

For example, CAD data of swing clamps cylinders used in this project cannot be used for simulation without simplification. Their complex structures could increase number of facets tremendously and extend the CPU usage. The simplification must be performed by removing features and parts where geometrical elements such as holes, bosses, pockets, hidden invisible parts and small or other selected parts are parts that can be removed. One can also decrease the quantity of polygons/triangular that usually has a great impact on the file size. Figure 4.2.1 explains the difference of a swing clamp cylinder before and after simplification. The route and the outer shell visualization are the only important features for the simulation of the cylinder. This is to predetermine the actual route for the movements in order to avoid collision. Therefore, detailed structure inside the cylinder is not equally important for the simulation process and doesn't have to be included in the virtual environment. By removing all those extra parts in the 3D world, the risk of straining the CPU usage decreases and the simulation will run more effectively.



*Figure 4.2.1: Swing clamp cylinders before and after model simplification.*

### **4.3. PLC programming and I/O signals**

Before the material flow of the production cell can be defined, the PLC program needs to be created and an I/O signals list used by the PLC has to be defined. The actual PLC program is replicating the real system as the I/O signals list. Based on software configuration, signals are setup in terms of required memory for data exchange which is generally grouped either as input or output signals. The I/O signals list is then imported to the simulation software tool so that each signal has the same functionality as in the real system.

### **4.4. Simulation model development**

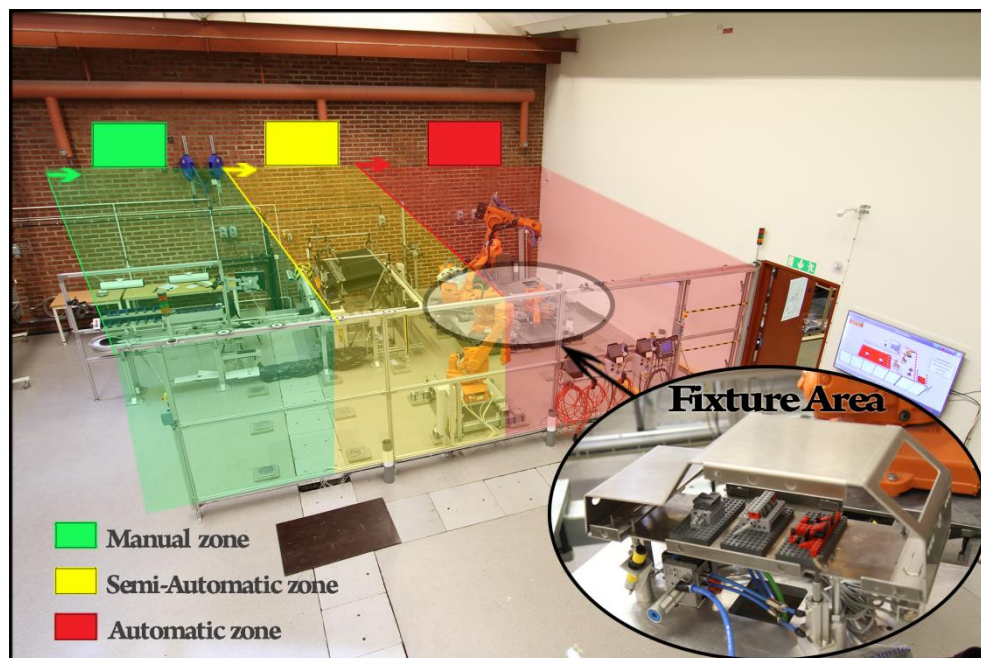
The next step in the VC workflow is to determine the sequence of operations regarding the production in the cell. All signals, inputs, outputs and other entities in the model are carried out in this step. Material flow of the production cell (see Appendix A.1) is used in this step to reflect the behavior of the production system in the simulation model. All system paths, definition of simulation activities, HMI and realization of alarm systems are developed in this step by using Experior API.

## **4.5. Establishment of communication**

The last and final step in VC workflow is to establish a communication between the simulation model and software/hardware PLC, i.e. SIL or HIL. There are different ways to establish communication by using a communication protocol. In this project the Fetch/Write and S7 functions Siemens protocol has been used with an IP address connection. When the communication has been set, the PLC program can finally be verified either as a SIL or HIL approach for testing the PLC program in the virtual environment.

## 5. PSL Case Study

Chalmers Production System Laboratory, PSL, is a research lab focusing on manufacturing and automation processes. It includes a production cell and ergonomic workplaces. In the PSL production cell (Figure 5.1), a fully integrated automation system exists with four robots, one automated guided vehicle (AGV), a conveyor system, fixture and a material façade. Guided by PLC- and Robot-programs, a product can be assembled and/or disassembled with different levels of automation separated into three different zones; automatic zone, semi-automatic zone and manual zone. The product to be produced in this production cell is a mini car consisting of three modules and four car bodies. Modules to be assembled are Lego engine, Lego gearbox and Lego interior. Car bodies to be assembled consist of a roof plate, bottom plate and two side metal plates.

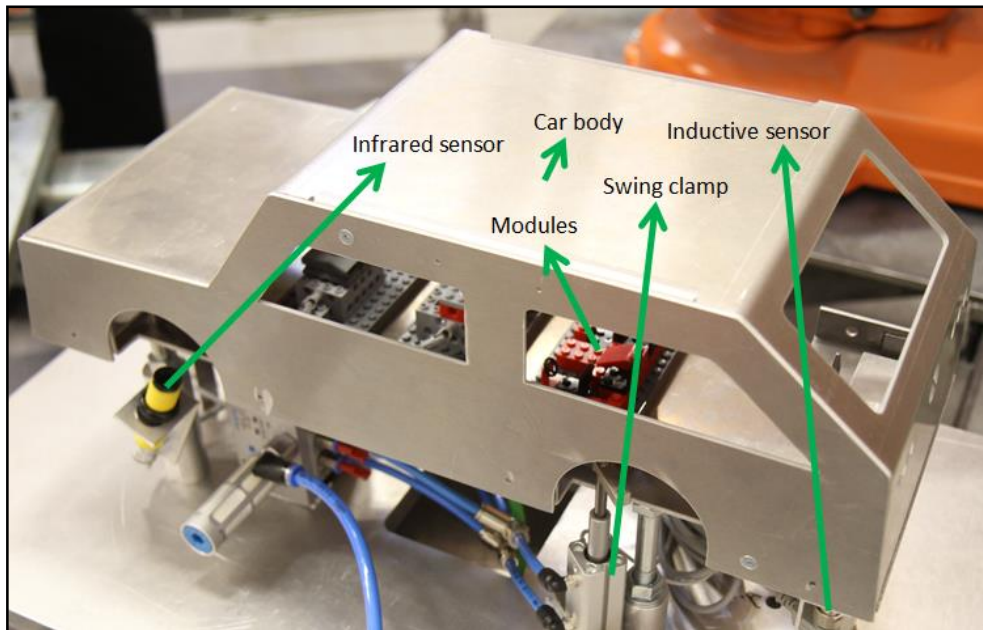


*Figure 5.1: Actual PSL production cell.*

### 5.1. Actual PSL production cell

The VC method has been applied onto Automatic zone and the fixture area. The fixture consists of six sensors (inductive- and infrared-sensors) and two actuators (swing clamp cylinders) which all work together in order to hold together modules and car bodies during the assembly (Figure 5.1.1). Two infrared sensors aim to trigger a response to the PLC when detecting objects from a pre-determined distance. Furthermore, four inductive sensors aim to trigger a response to the PLC when detecting metal objects. Actuators are activated by a PLC program and sensor

triggering response. The actuators are able to move in three different ways; up, down and in a rotation. All movements are pre-determined by two different states where the first state is the initial state and describes the clamps when they are down, while the second state is activated when the car bottom part has reached the fixture and inductive sensors are activated. Both clamps will then move upwards, rotate with a certain degree and finally move downwards again in order to hold the part. When all objects are assembled both clamps will return to its initial states.



*Figure 5.1.1 Representation of the minicar and surrounding equipment.*

Figure A.1.1 in Appendix A.1 describes material flow of the whole assembly process in the Automatic zone and the interaction between modules, car bodies, sensors and actuators. Hardware PLC, Siemens S7-300, is contributing as a main control program between all parts in the production system via a fieldbus communication protocol.

## **5.2. Virtual PSL production cell**

As it can be verified with (Figure 5.1), the model (Figure 5.2.1) is an accurate representation of the automatic zone in PSL with resources such as a fixture, car bodies, modules, sensors and actuators. As mentioned in chapter 1, robots are not included in this project and are thus excluded from the virtual representation. Transportation of modules and car bodies are instead manually programmed in the programming language C# with the purpose to represent the same behavior that robots have, without any difference regarding the visual perspective.



*Figure 5.2.1: Virtual PSL production cell.*

Even though Experior has been used as the main simulation tool, several other tools had to be used together with it in order to create a proper reflection of the actual production cell. These software tools involved SketchUp, Siemens TIA-Portal V11 (TIA-Portal) software/hardware PLC, NetToPLCSim and Visual Studio. As a result, the use of mentioned software tools worked properly together with Experior. From that, a VC project could be performed by following the steps; 3D model development, PLC programming and I/O signals, simulation model development and establishment of communication between PLC and Experior. All the results and discussions regarding these steps are further discussed in the sections below.

### *5.2.1. 3D model development*

First step, 3D model development, involved simplification of 3D models in SketchUp with the purpose to decrease CPU usage and thereby ensure proper simulation running without any lag or delays. Inside extra material, holes and other unnecessary data has been eliminated as well as extra polygons/triangular without hampering the simulation model. Most of the models included tremendous amount of polygons/triangular and it would take a lot of time in order to manually minimize the quantity. Instead, by using a plugin in SketchUp CleanUp (see Appendix A.3), one could easily minimize the file size and quantity of polygons/triangular in a small amount of time.

3D models were then imported into Experior with the file formats .X and COLLADA. The file formats were used for dynamic- respectively static- models. With SketchUp plugin 3D RAD (see Appendix A.3), one could export 3D models directly to .X file format. On the other hand, to export a 3D model to COLLADA file format, one had to use a 3<sup>rd</sup> party tool, in this project AnyCad Exchange. This is because SketchUp doesn't support a fully working COLLADA file [11].

From models that have been created in SketchUp and later on imported into Experior, a custom made catalog was created in C# and then added to the component library. From that point, when custom made models were imported to the catalog, they all inherited property options that Experior API provides for Experior components. Properties such as PLC communication, dimensions and color are a few of the many property options that were included and important for developing the simulation model. In this project, models imported as .X files could also behave as rigid parts with functions such as collision detection and friction. Models imported as COLLADA did not behave as rigid parts and were only used for static parts. One major benefit with separation of static parts and dynamic parts is that the static parts use less CPU than dynamic parts. One can automatically increase smoothness of the simulation process by separating those parts. Together with a strong computer that includes good RAM, process speed, storage capacity and a good graphical card, bigger PLC programs for increased production plants can virtually be verified with less risk for disruptions. In order to run the simulation process in this thesis, it was enough to use a 64-bit Windows 7 computer with a 6 GB RAM, 2,3 GHz processor from Intel and a Geforce GT 640M LE graphical card from Nvidia.

Besides importing custom made models, this project also used the existing Experior component library and sensors belonging to it. These sensors were replicating the inductive and infrared sensors from the real system. As a final result, all models used for the simulation could easily be found in the main Experior window and changing properties in order to replicate the real world system, i.e. all models could thereby be used as high-level plant modeling for VC.

### 5.2.2. *PLC programming and I/O signals*

Next step of VC workflow, PLC programming with an alarm system and I/O-signals list was developed. A PLC program has been developed in order to reflect the exact behavior as in real system. Because this project only considered the fixture area in production cell, the developed PLC program only considered the behavior of sensors, actuators and alarm system belonging to that area. During development phase, corresponding input and output signals were created for sensors, actuators and the alarm. Sensors corresponded the inductive and infrared sensors, actuators the swing clamp cylinders and an alarm. An alarm was added which reflected the real world scenario where production would stop if someone or something entered the Automatic zone during production. By resetting the alarm with a reset button, the production could then continue as normal. With Experior API, one could then use the PLC behavior and prepare I/O values such as tag name, addresses and data type as shown in Table A.2.1 and Table A.2.2 showed in Appendix A.2. Finally when the I/O signals list is imported to Experior, all sensors and actuators will automatically get their PLC values with a requirement of same tag name.

### 5.2.3. *Simulation model development*

In order to fulfill the I/O-mapping, the simulation model had to be developed with all resources and values ready to be mapped with corresponding values from the PLC software. As described earlier, a simulation model has been developed in C# wherein all resources, load transportations and PLC connections were pre-determined by following the steps of PSL workflow. By using different software tools together with Experior, it was possible to virtually represent the real production cell in PSL. There were some challenges with developing the simulation and the available Experior API. Experior wiki provides an introduction to Experior API but further documentation about all methods and classes in Experior API was lacking where expert help from Xcelgo was needed or by using trial and error based method during the development in order to find optimal solutions for the simulation model.

Another challenge during development was the use of imported 3D models from SketchUp. It was found that the coordinate system in SketchUp (X,Y,Z) was viewed as (X,Z,Y) in Experior, and so during development of the simulation model, one had to view the Z coordinate in SketchUp as the Y coordinate in Experior and the other way around. Another thing to keep in mind is that all 3D models imported to Experior are surrounded by an invisible block and acted as one even though the models had another shape. For example, a bottom car body part consists of holes. In the real system, when a body part has reached its position on the fixture, the holes would make it possible for sensors to detect the top car body part appearing behind the bottom car body part by detection through existing holes and thereby trigger a response to the PLC. In the simulation, this scenario would not work because all models are behaving as blocks even though they visually seem not to. Therefore, another solution was required. One could either replace the sensors, or by separate the 3D model in several parts. In this project, sensors were re-placed in order to make the simulation work as it does in the real world.

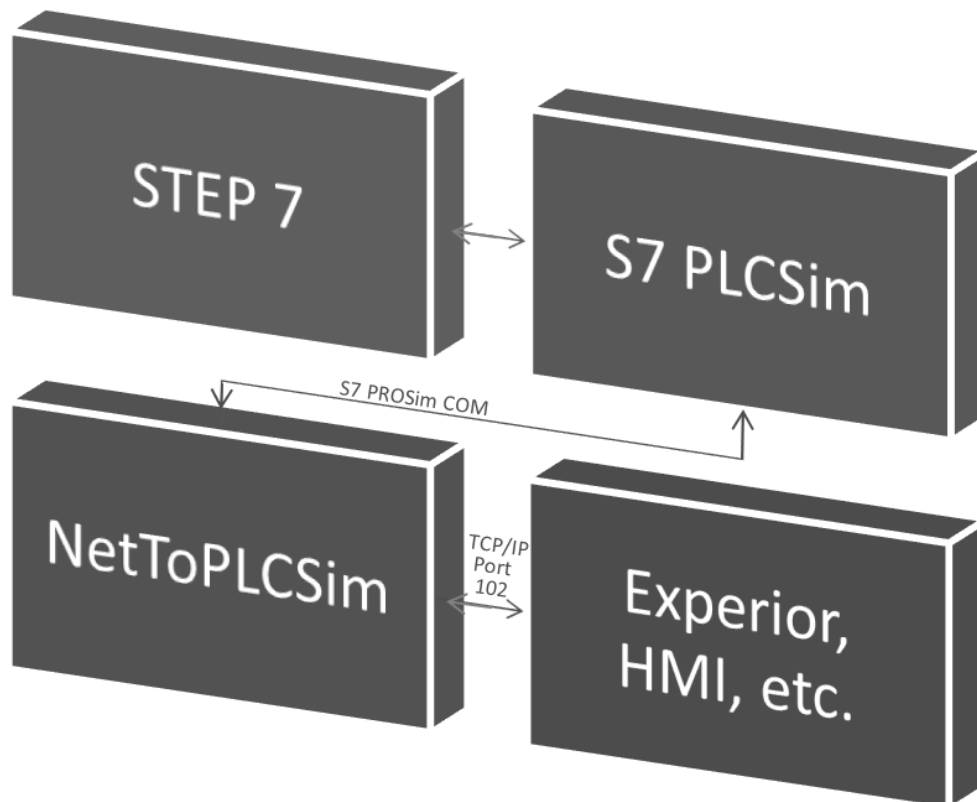
Once the catalog has been developed and then available in the Experior catalogs list, it was easy to fulfill the simulation. As a result, one could find that Experior provides user friendly simulation environment even though the most difficult part of the development phase is the actual creation of custom catalogs. All other features necessary for a proper simulation is very easy to find and use.

### 5.2.4. *Establishment of communication*

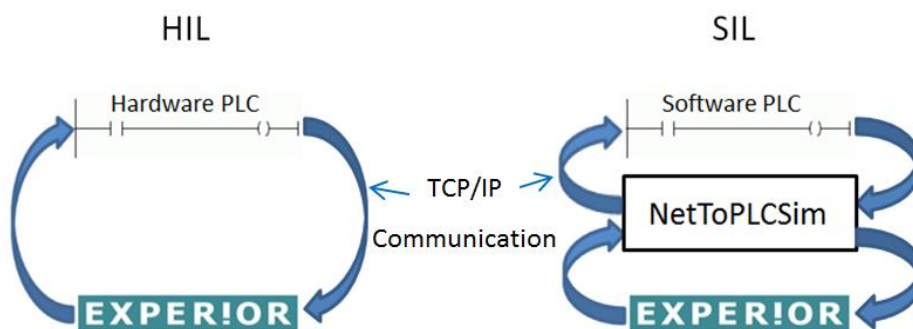
Last step in VC workflow is the verification of the developed PLC program which also was the main objective of this project. The verification process was established with both software PLC (SIL) and hardware PLC (HIL) wherein the differences and similarities also were verified. For communication with TIA-Portal, available communication protocols in Experior were Fetch/Write and S7 functions. For this project, the Fetch/Write protocol has been used for testing PLC signals internally in Experior prior to the real connection with PLC. For HIL, S7 functions were used for



which direct connection with hardware PLC was established by using the same IP address in both ends of the communication. On the other hand, with SIL, one could not establish a direct connection with Experior and instead had to use an external connection tool NetToPLCSim, Figure 5.2.4.1. The external connection tool uses Siemens S7ProSim COM-interface of the PLCSim to read and write data from it. By using the same IP address in both ends of the communication, a connection with Experior was possible. Figure 5.2.4.2 shows the overall establishment of communication for SIL and HIL approach. The communication with S7 functions was tested in various speeds and resulted in delay- and error-free system.



*Figure 5.2.4.1: Communication between software PLC and Experior.*



*Figure 5.2.4.2: Communication approach for HIL and SIL.*

### 5.3. Conclusion Case Study

A sufficient personal computer (chapter 5.2.1) was used for the verification process which included 3D modeling in SketchUp, simulation development in Experior and PLC programming in Tia-Portal. A conclusion for the PLC verification process is that the user is able to verify the PLC program with one or several computers working in parallel, at the same time, with both SIL and HIL approach.

Taking an example of SIL or HIL from the case study, when a button is triggered by the user, the respective I/O signal will then be triggered and transferred to the software/hardware PLC which reads and sends back the signal for instant triggering action in the simulation model. The button in this case is used as a HMI action that is virtually represented directly in Experior. To compare SIL and HIL, the simulation of PSL production cell in Experior has been tested several times and showed same results in both approaches. Functions such as a HMI system and an alarm system were transferred and communicated correctly as well. All signal transfers were automatically logged through Experior and could be viewed directly at the existing monitor pane. Also input and output lists were listed and no error detection could be viewed from respective verification approach during the simulating. The only difference between the approaches for this case study was that a 3<sup>rd</sup> party tool had to be applied in SIL to establish a communication between Experior and the soft PLC.

Even if the simulation process ran flawlessly through the tests, there was still one drawback during the simulation model development which forced the users to reorganize the cell layout in Experior. As described in chapter 5.2.3, all 3D models behave as blocks, therefore models that consist of holes cannot behave as holes even if they visually are. This forced the users to re-arrange the positions of sensors in order to keep the same system functionality as in the reality.

## 6. Evaluation on Research

This project presented the concept of virtual commissioning where several research questions in chapter 1 needed to be answered. To answer the research questions, a VC project was set and corresponding steps represented in the VC workflow were followed: Data collection, 3D model development, PLC programming and I/O signals mapping, simulation model development and establishment of communication between Experior and software/hardware PLC.

The evaluated questions are listed as follows:

(1) *Can a PLC program be verified by using a simulation software tool such as Experior from the Danish company Xcelgo?*

In general, the PLC programmer has to wait until the physical plant is built before testing PLC code. By using Experior, the PLC code can instead be tested in-house and in a virtual environment for safety reasons. Another benefit is that the PLC programmer and the simulation model developer can work in parallel with online changes in the PLC program during the simulation process. The PLC verification is performed either from one computer or separately from several computers. Several computers are often preferable if the PLC- and simulation-developers work in parallel and want to test the PLC program successfully without any further work with document transportations. The only integration needed between software/hardware PLC and Experior are input and output signals in both software. These signals are communicated through a TCP/IP communication protocol. To simplify the work even further, Experior has the ability to connect all inputs and outputs only by listing the same tag name in Experior as in the PLC program. The address and data type will then be integrated automatically in Experior. By importing the I/O list from an Excel file, the whole process will automatically be performed. A little disadvantage is that the I/O list exported from Siemens needs to be converted into an Excel template that Experior can handle before the list can be imported.

(2) *How to setup a project for verification of PLC programs?*

There are several ways to verify PLC programs and few common methods are listed as follows; Virtual verification method in a simulated 3D model environment, formal verification method [13] and trial and error verification method. All three methods can be performed together in sequence or alone but in this project was only the first method used, virtual verification in a simulated 3D model environment. This is mainly because of the project time limit and also because Teamster wanted to test Experior as a PLC verification tool. Formal verification method is also an alternative method for future projects, mainly to secure the quality of the PLC program

even further. Using traditional trial and error method is on other hand an excluded because it is the main reason that production companies want to see alternatives of commissioning processes, including Teamster.

- (3) *How many approaches are there for VC and can all be used for verification of PLC programs?*

There are two main approaches used in order to building a VC project, SIL and HIL. It was confirmed after numerous tests that it is possible to verify PLC programs with both approaches where communication was established without any delays or interruptions (chapter 5.2). One still needs to take in consideration that in this project was only simple inputs and outputs included in the PLC program. No complex PLC functions such as servo motors and frequency converters have been tested which could increase the risk for inequalities between SIL and HIL. The overall conclusion for the communication with Experior is that all communication could be done with a TCP/IP communication protocol. With SIL approach, one still needed a 3<sup>rd</sup> party tool called NetToPLCSim for communication purpose but the SIL approach is preferable unlike HIL in a way that no hardware needs to be connected to a computer. On other hand, the HIL can be directly communicated with Experior through TCP/IP but needs to be connected to a computer. The same PLC can also be directly integrated into the real production facility. In this project, HIL was already connected to another computer located in the PSL which had TIA-Portal installed. The communication between PSL computer and the computer used in this project was simply established with TCP/IP. Both software and hardware PLC are first connected to TIA-Portal and then communicated through TCP/IP to Experior.

- (4) *Can the simulation software tool interact with SketchUp used in Teamsters current work flow?*

Experior can interact with both SketchUp and TIA-Portal. All file formats can be imported to Experior without any property changes. The only difference between the model in SketchUp and the model in Experior is that all imported 3D models act as blocks in Experior. This is a disadvantage which results in a fact that the models are not having the exact model behavior as in reality. For instance car body parts that include holes and have a purpose to help the sensors to see through the parts will not be able to do so in Experior. This issue resulted in a re-arrangement of sensors on the fixture and thereby also failed to mimic the production cell in full.

- (5) *What knowledge is required for simulation model development?*

Experior doesn't require deep knowledge if the use of C# is excluded. C# is in need for more complex production processes that requires more complex functions in Experior API. C# could also be excluded in this project if robot programs and robots were used in this project which automatically would take care of movements and actions. Instead, all

movements and routines had to be programmed in C#. Overall conclusion regarding author's observations of Experior is that the tool is rather a simple tool that gives the user an ability to use drag and drop possibilities for 3D models. Experior has also implementation possibilities of PLC I/O lists and creation of dynamic functions. It is a sophisticated tool with a purpose to simplify the work for the user regardless if the user is an experienced simulation model developer or not. The API functions and methods needs to be further developed because a lot of functions needed to be tested manually in pre-hand to see what functions they had. Some of the functions also needed extra explanation from Xcelgo's Experior developers. The explanation was well informed and easy to understand but still required extra time to implement in the program instead of having the documentation directly available in the API. As a result, Experior can be seen as a flexible tool for both simple and more complex production processes. The tool has a lot of functions available for the users and the main window has a clear view of how to create the simulation process in an easy manner. If the user still has some questions regarding the program, Xcelgo Experior developers are always in helping hand for follow-ups.

## 7. Conclusion & Future Work

Today, verification of PLC programs in a highly automated manufacturing industry is limited in physical level regarding the hardware PLC. Traditional level of PLC program verification is therefore a must with time consuming and costly projects. A demand for a virtual verification of such systems has therefore increased rapidly. Regarding to the demand, a VC method is implemented in this project with the intention to give PLC and mechatronic engineers an advantage by working together with the same model at the same time and in an early design phase. By using Xcelgo Experior interacted with other tools such as Visual Studio, SketchUp and TIA-Portal, a complete VC project could be performed.

In future work regarding Experior, PLC and communication, there are some factors to consider. Working with new software, it is required to have good information accessible about the software and how it works. Xcelgo offered an online documentation that served this project with good help during model development. Xcelgo also served us with an introduction about the software and extra help from software experts which was a great contribution to achieve a SIL and HIL simulation. In future work, a more extensive documentation available for users would make the development of simulation models much more independent from expert help by Xcelgo.

The communication protocol S7 functions provided a communication with software/hardware PLC. Though, for the software PLC, the connection had to be performed through a 3<sup>rd</sup> party tool, NetToPLCSim. In future work, it would be a benefit to access the software Siemens PLC directly from Experior to establish an easier connection. It is also of great interest to investigate systems in which timed synchronization becomes more important, for example communication possibilities for external robots and frequency converters. Therefore, verification of PLC program could in future projects include a larger scale of the production cell in PSL with corresponding robots, conveyor system and AGV.

Another future work would be to find out how the issue in chapter 5.2.3 could be solved. Currently, objects with holes that are imported to Experior could not act as if they have holes because they are all surrounded by an invisible block. Future work is to find out how to overcome this problem without making any re-placement of sensors and then optimize the 3D model development in this project. Another further research question regarding the 3D model development is if it is possible to import both static and dynamic parts in same file and separate those within Experior instead of separating these into two separate files before the importation. This would simplify the development of custom catalogs in Experior tremendously.

When an I/O-list was created in the PLC environment, one could import the total list into Experior main window and thereby give simple PLC properties to resources in the simulation model. Despite the easiness of I/O list handling in the virtual environment, there is still some work to do before implementation. Today, Experior are

using an older version of Excel file format for I/O signal list handling which derives the fact that more work needs to be done prior to the importation. A future recommendation would be to consider direct importation of I/O signal list from TIA-Portal without any need for adjustments.

Teamster currently uses traditional way to verify PLC programs. In order to decrease the project lead time, Teamster is interested to integrate SIL and/or HIL PLC verification approach. As verified from this project, Experior can be integrated with both SketchUp and TIA-Portal, both included in Teamster's current workflow. The commissioning can therefore be performed in-house and errors in the PLC program can also be found and corrected in an early commissioning phase. It means that time used for physical commissioning process and traveling will potentially be decreased.

The time for simulation model development is hard to predict. However, the most things that took time in the simulation developing phase in this thesis was to program in C# which instead could be replaced by a robot software tool as for example RobotStudio from ABB. As a conclusion of this thesis and future work recommendations, the software tool Experior has a potential to decrease the overall project lead time and definitely decrease the physical commissioning time at the production facility if PLC programs were verified in-house in an early stage of the commissioning process.

# References

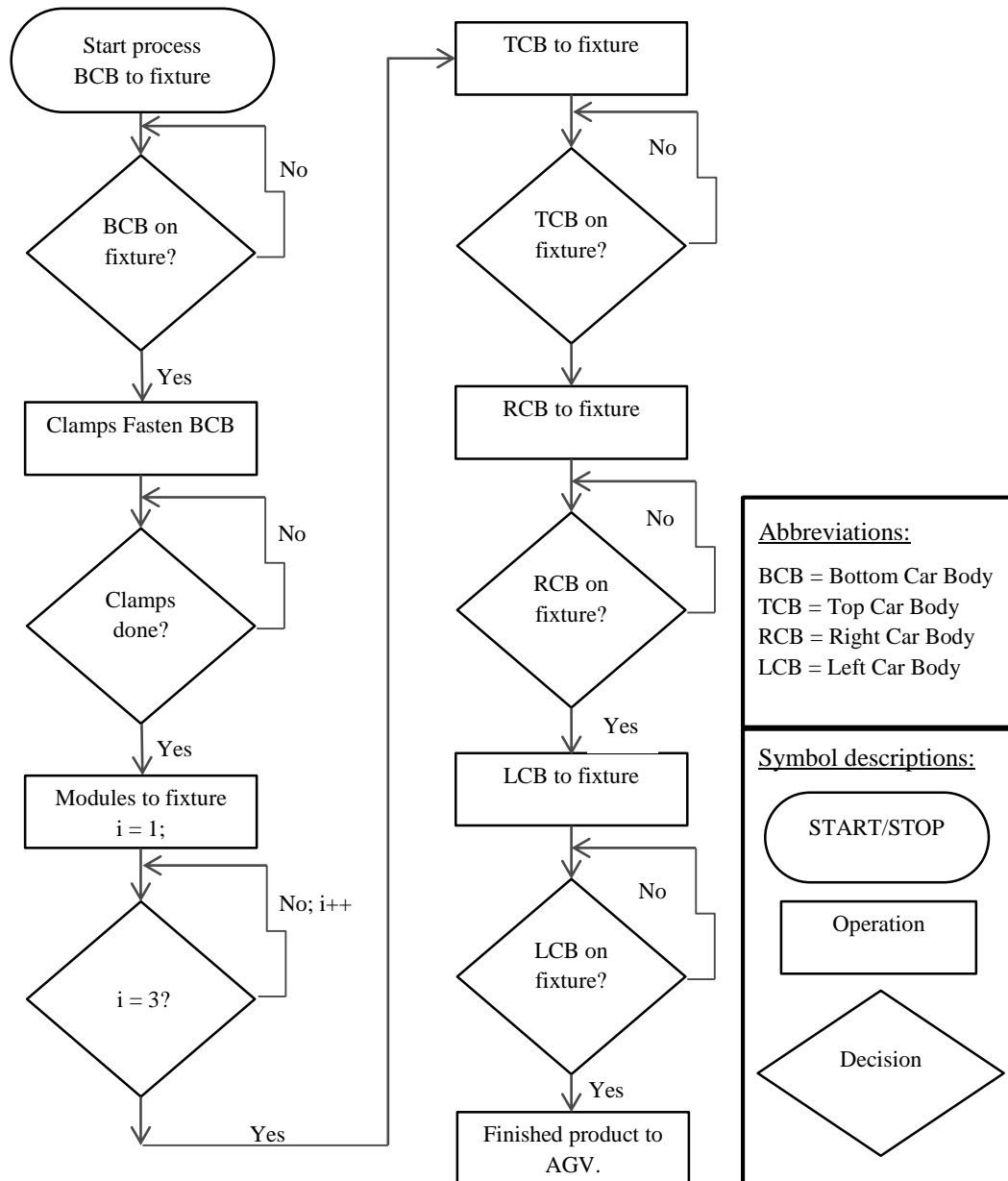
- [1] S. Makris, G. Michalos and G. Chryssolouris. Virtual Commissioning of an Assembly Cell with Cooperating Robots. In *Advances in Decision Science*, Volume 2012, Article ID 428060, page 11, 2012.
- [2] T.Lydhig, P.Olsson. Interviewed by: J.Dzinic & C.Yao. 26 March 2013.
- [3] P. Hoffmann, R. Schumann, T. M. Maksound and G. C. Premier. Research on simplified modeling strategy for virtual commissioning. In *Proceedings of the European Modeling and Simulation Symposium*, pages 293-302, 2012.
- [4] J. S. Smith and Y.Cho. Offline commissioning of a PLC-based control system using Arena. In *Simulation Conference*, pages 1802-1810, 2008.
- [5] Xcelgo, Experior PLC tester (March 2013).  
Available from: [http://www.xcelgo.com/default\\_7E598EF2.htm](http://www.xcelgo.com/default_7E598EF2.htm)
- [6] P.Hoffmann, R.Schumann, T.Maksoud and G.Premier. Virtual commissioning of manufacturing systems - a review and new approaches for simplification. In *Proceedings 24<sup>th</sup> European conference on modelling and simulation*, ibs 41, 2010.
- [7] T. Erlandsson, M. M. Rahaman. Testing and verifying PLC code with a virtual model of Tetra Pak Filling Machine. Place: Chalmers University of Technology, EX016/2013, 2013.
- [8] I. McGregor. The relationship between simulation and emulation. In *Proceedings of the 2002 Winter Simulation Conference*, pages 1683-1688, 2002.
- [9] L. Chwif, M. R. Pereira Barretto and R. J. Paul. On simulation model complexity. In *Proceedings of the 2000 Winter Simulation Conference*, pages 449-455, 2000.
- [10] S. Dominka, F. Schiller and S. Kain. Hybrid commissioning-from hardware-in-the-loop simulation to real production plants. In *The 18<sup>th</sup> IASTED International Conference on Modelling and Simulation*, pages 544-549, 2007.
- [11] SketchUp, 3D Model COLLADA (DAE) Export (September 2013)  
Available from: <http://help.sketchup.com/en/article/114345>
- [12] C.Park, S.Bajimaya, S.Park, G.Wang, K.Han & M.Chang. Development of Virtual Simulator for Visual Validation of PLC program. In *International Conference on Computational Intelligence for Modelling Control and Automation, and International Converfence on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, page 32, 2006.
- [13] H.Barbosa & D.D ħarbe. Formal Verification of PLC Programs Using the B Method. In *Third International Conference, ABZ 2012; Abstract State Machines Alloy, B,VDM and Z*, pages 353-356, 2012.



# Appendix A

## A.1 Material flow of automatic zone in PSL

Flowchart is ideal for processes that have sequential process steps. Steps are executed in a simple order where the arrows follow the steps after various decisions being made.



*Figure A.1.1: Flowchart of the Automatic zone in PSL*

## A.2 Inputs/Outputs of the PLC program

*Table A.2.1: Input-list of the PLC program*

Symbol	Address	Type	Source	No/Nc	PLCid	Description
AlarmSensor	I1.7	BOOL	Input	false	0	Block/Unblock
LeftClamp	I0.7	BOOL	Input	false	0	Output
LeftClamp_Finished	I1.2	BOOL	Input	false	0	Output
Module	I1.1	BOOL	Input	false	0	Output
RightClamp	I1.0	BOOL	Input	false	0	Output
RightClamp_Finished	I1.3	BOOL	Input	false	0	Output
SensorBottom_1	I0.0	BOOL	Input	false	0	Block/Unblock
SensorBottom_2	I0.1	BOOL	Input	false	0	Block/Unblock
SensorLeft	I0.4	BOOL	Input	false	0	Block/Unblock
SensorRight	I0.5	BOOL	Input	false	0	Block/Unblock
SensorTop_1	I0.2	BOOL	Input	false	0	Block/Unblock
SensorTop_2	I0.3	BOOL	Input	false	0	Block/Unblock
StartButton	I0.6	BOOL	Input	false	0	Pushed
AlarmSensor	I1.4	BOOL	Input	false	0	Block/Unblock
ResetAlarmButton	I1.5	BOOL	Input	false	0	Pushed

*Table A.2.2: Output-list of the PLC program*

Symbol	Address	Type	Source	No/Nc	PLCid	Description
CarBottom_Movement	Q0.0	BOOL	Output	false	0	Input
CarFinished_Movement	Q0.6	BOOL	Output	false	0	Input
CarMoveBack_Movement	Q0.7	BOOL	Output	false	0	Input
CarSdeLeft_Movement	Q0.5	BOOL	Output	false	0	Input
CarSdeRight_Movement	Q0.4	BOOL	Output	false	0	Input
CarTop_Movement	Q0.3	BOOL	Output	false	0	Input
ClampClose_Movement	Q0.1	BOOL	Output	false	0	Input
Module_Movement	Q0.2	BOOL	Output	false	0	Input
RunningAfterAlarm	Q1.1	BOOL	Output	false	0	Input
AlarmLampa	Q1.0	BOOL	Output	false	0	Input

Experior Import descriptions:

- Symbol: (Mandatory)  
This is the key for setting up an address in Experior.
- Address: (Mandatory)
- Type: (Optional)
- Source: (Mandatory)
- No/Nc: (Optional)
- PLC id: (Optional)  
(When filling out this column the PLC id and the Symbol is used as the key when trying to find a match.)
- Description: (Optional)

### A.3 Plugins used in SketchUp

Plugins for SketchUp is used in this project to create 3D models that could be imported by Experior.

- (1) **3D Rad:** 3D Rad is a plugin that exports 3D models into .X.
- (2) **CleanUp:** CleanUp is a plugin for optimizing and cleaning up a 3D model. Features included in the plugin are for example; Purging unused items, erasing hidden geometry and duplicate faces etc.
- (3) **Align Tool:** Align tool is a plugin for quickly aligning a specific selection in the 3D model from a plane to a target plane.
- (4) **Angular Dimension:** Angular Dimension tool is a plugin for defining angular dimensions on a 3D model.