



The versatility of virtual models

A practical case of virtual commissioning

MASTER THESIS



Lucas Gardebrand

Faculty of Engineering
Lund University
2022



Examiner LTH

Dr. Charlotta Johnsson, Professor, Dept. of Automatic Control

Supervisors LTH

Dr. Anders Robertsson, Professor, Dept. of Automatic Control

Dr. Volker Krueger, Professor, Dept. of Computer Science

Industrial supervisors Alfa Laval

Jesper Norrman, Operation Development Specialist

Ulf Persson, Automation & Digitalisation Specialist

Copyright © Lucas Gardebrand

Lund, Sweden 2022

Institutionen för Reglerteknik

Institutionen för Designvetenskaper

Lunds Tekniska högskola

Lunds Universitet

Box 118

221 00 Lund

Sverige

Department of Automatic Control

Department of Design

LTH, School of Engineering

Lund University

Box 118

SE-221 00 Lund

Sweden



QR-codes found throughout the thesis
provide deeper insight in forms of video or
in-depth text. (as per 2022-06-23)
They can be scanned with the camera app
in most modern phones.

Abstract

The commissioning phase is the last phase of a construction project when all subsystems come together to be tested as a whole, working as a single entity. This phase is often the most crucial and time-sensitive part of the project as the construction occupies space in the factory and inflicts downtime in processes without adding value. Faults found in this phase can be time-consuming to correct and exhausting for the integrator who has to work intensely over extended periods. A few software changes might solve minor faults, but finding a major design fault at this stage could be catastrophic. The cost of change rises as the project progresses, and at this stage, all parts are manufactured and delivered.

Virtual commissioning could reduce these risks by enabling tests to be made earlier and continuously during the project in a more agile manner. Robots and PLC-code can be run together through emulations, and the cell's behavior can be verified. Any issues that arise can calmly be resolved in the office, far away from the stressful commissioning on a tight schedule and busy factory floor. On the day of the physical commissioning, the tests should give the last confirmation of the coherence with the virtual model.

The models used for the virtual commissioning can be used in earlier project phases. They can be used for layout planning, presentation aid, operator training, and evaluating proposed process changes.

This thesis investigates, analyzes, and practically applies the concept of virtual commissioning and the use of virtual models connected to it.

Keywords: Virtual Commissioning, Virtual models, RobotStudio, Emulation, Simulation, Software, Rockwell, OPC-UA.

Sammanfattning

Driftsättningsfasen är den avslutande fasen i ett automationssprojekt. I många fall är detta första gången alla system körs tillsammans för att fungera som en enhet. Inte sällan är driftsättningen den mest påfrestande delen av projektet då tidspressen kan vara stor för att få igång en maskin som annars står och tar upp plats utan att vara värdeskapande. Fel som upptäcks först i denna fas kan vara tidskrävande att åtgärda och en stor påfrestning för integratören som måste arbeta intensivt under långa dagar. Mindre programmeringsfel skulle troligen kunna lösas med ett par kodjusteringar, men upptäckten av ett större konstruktionsproblem så här sent i processen skulle kunna vara förödande. Allt eftersom projektet fortlöper blir förändringar allt mer kostsamma och vid driftsättningen har alla komponenter redan tillverkats och levererats.

Virtuell driftsättning strävar efter att minska ovan nämnda risker genom att dels tidigarelägga testerna, dels göra dem helt virtuellt i ett mer agilt arbetssätt. Med detta tillvägagångssätt kan testarbetet ske simultant som programmering och konfigurering vilket ger stora besparingar under den fysiska drifttagningen. Problem som upptäcks virtuellt kan fixas på kontoret i lugn och ro, långt från den stressiga, tidspressade drifttagningen och fabriksgolvet bråda tempo.

De modeller som används under den virtuella driftsättningen kan även vara till nytta i flera andra faser under projektet. De kan användas för layoutplanering, presentationshjälpmedel, operatörsträning och processförändringar.

Detta arbete undersöker, analyserar och praktiskt applicerar konceptet virtuell driftsättning tillsammans med de virtuella modeller som används i driftsättningen.

Nyckelord: Virtuell driftsättning, Virtuella modeller, RobotStudio, Simulering, Rockwell,

Acknowledgments

Jenny, thank you for being my wife and best friend.

My two families - thanks for laying the foundation that took me here after 25 years of constant support.

I want to thank my Alfa Laval supervisors, Anna Wenemark, Jesper Norrman, Ulf Persson, and LTH supervisor Anders Robertsson who guided me through these intense months.

I want to thank Richard Jönsson and David Hörstedt for listening to my ideas and not hesitating to consider and implement them in the Viking project. Thanks to Anders, Gustav, Edvin, Thomas, and all of the Alfa Laval employees that have made me feel at home at Alfa Laval.

Finally, Ronja, thank you for meticulously and irreproachably proofreading this paper countless times.

List of abbreviations

Abbreviation	Meaning	Page
AM	Additive manufacturing	10
AR	Augmented reality	33
CAD	Computer-Aided Design	7
CAE	Computer-Aided Engineering	7
CAM	Computer-Aided Manufacturing	8
DCS	Distributed Control Systems	15
DT	Digital Twin	34
FAT	Factory Acceptance Test	37
HMI	Human-Machine Interface	28
I/O	Input/Output	50
OPC-UA	Open Platform Communication -Unified Architecture	30
PLC	Programmable Logic Controller	15
PLM	Product Lifecycle Management	14
PoC	Proof of Concept	87
SAT	Site Acceptance Test	38
SCADA	Supervisory Control And Data Acquisition	28
vFAT	Virtual FAT	38
VR	Virtual Reality	33

Table of Contents

Abstract.....	ii
Populärvetenskaplig sammanfattning.....	Error! Bookmark not defined.
Acknowledgments.....	iv
List of abbreviations.....	v
1. Introduction.....	1
1.1 Company description.....	1
1.2 Background.....	1
1.3 Objective.....	1
1.4 Delimitations.....	2
1.5 Confidentiality.....	2
1.6 Target audience.....	3
1.7 Thesis Outline.....	3
1.8 Methodology.....	3
2. Theoretical background.....	7
2.1 Virtual Models.....	7
2.2 Industrial Automation.....	14
2.3 Digital Factory.....	32
2.4 Virtual commissioning.....	35
3. Literature review and industry research.....	39
3.1 Literature review.....	39
3.2 The industry speaks.....	52
3.3 Virtual Commissioning Software.....	60
4. Viking - From idea to commissioning.....	65

4.1 Current procedure at Alfa Laval.....	65
4.2 Project Viking.....	67
4.3 Collecting and preparing the data.....	70
4.4 Virtual environment setup.....	73
4.5 Programming.....	76
4.6 Safety.....	86
4.7 Virtual commissioning.....	91
4.8 Further uses of virtual models.....	100
4.9 Physical commissioning.....	109
5. Discussion.....	111
Literature review and industry research.....	111
Simulation software and interfaces.....	115
Virtual commissioning of Viking.....	117
Extended use of virtual models.....	119
6. Conclusions and future work.....	121
7. Recommendations for Alfa Laval.....	123
References.....	135
Appendix A. Interviews.....	A1-A18
Appendix B Robotics code.....	B1-B22
Appendix C. Smart components.....	C1-C6
Appendix D RobotStudio Simulated PLC.....	D1-D10

Table of Figures

Figure 1 Industry-as-laboratory, inspiration from Potts (1993) and Muller (2011).	5
Figure 2 Top: Simple milling process with 22 instruction rows. Image reproduced with permission from Rainer Stark. Bottom: AM process with 120000 instruction rows.	8
Figure 3 Handheld 3D scanning using a Revopoint scanner and their software.	9
Figure 4 Different detail levels, file size scale proportionally with the number of facets.	11
Figure 5 Different methods of Additive Manufacturing. Image reproduced with permission from Formlabs.....	12
Figure 6 Automation hierarchy based on ISA-95 with approximate time frames.	15
Figure 7 An old relay panel with plugin relays. Image reproduced with permission under Creative Commons (Signalhead, 2007).....	16
Figure 8 A modular PLC with various modules.....	17
Figure 9 Simple example of an SFC with a parallel branch executing two steps simultaneously.	18
Figure 10 Equivalent FlipFlop-function programmed in FBD, ST, IL, and LD.	20
Figure 11 Relative mentions of terms in literature from 1960-2019 (Michel et al., 2011).....	21
Figure 12 Robot mounted on a linear 7 th axis to extend the workspace.	22
Figure 13 Load diagrams of the ABB robot IRC8700. Right: load diagram for vertical wrist (ABB, 2022b).	23
Figure 14 Different robot types from the four largest robot companies. From left, KUKA 6-axis industrial robot, ABB 6-axis collaborative robot, Yaskawa 4-axis SCARA robot, and Fanuc 4-axis Delta robot.....	24
Figure 15 Coordinate systems used by the robot. a) world, b) task frame, c) base frame, d) user frame, e) object frame, g) tool0, f) TCP.....	25
Figure 16 User frame and object frame defined.	26
Figure 17 ABB's Flexpendant with Workobject Definition Wizard.	27
Figure 18 All eight configurations to reach the same position and orientation of a specific target; each robot is offset 300mm, and two parallel rods are added for clarity.....	27
Figure 19 Award-winning HMI Brückner Maschinenbau GmbH. Image reproduced with permission.	29
Figure 20 Performance level calculator within SISTEMA.....	31

Figure 21 Commissioning configurations, inspired by (Auinger et al., 1999).	36
Figure 22 Flowchart of literature review according to the PRISMA model.	39
Figure 23 Companies' estimates of employee competence concerning VC. Data aggregated from (Alkureidi, 2020).....	42
Figure 24 Virtual commissioning environment by Bysko et al. (2020) Image reproduced with permission from Sara Bysko.	46
Figure 25 Visionary model of the digital process. Image reproduced with permission.	53
Figure 26 Using VR to program the paint program.....	58
Figure 27 A conveyor plant within Emulate3D (Emulate3D, 2022).....	62
Figure 28 Different stages in Alfa Laval's Project Management Strategy PROMAL	65
Figure 29 Hypothetical scenario of a manufacturing project lifecycle, before and after implementing the use of virtual models. Time savings are symbolic estimates.	66
Figure 30 TRL levels used at Alfa Laval (Image reproduced with permission).....	67
Figure 31 Part view of an Alfa Laval heat exchanger (Alfa Laval, 2022).	68
Figure 32 Proof of Technology by Mälardalen Industrial Technology Center.....	68
Figure 33 Overview of Viking, simplified due to confidentiality.	69
Figure 34 The table model used to display the CAD tools.	71
Figure 35 Holes, fillets, chamfers, pockets, and embosses highlighted for removal.	72
Figure 36 Components below 60 mm visualized and selected for removal.	72
Figure 37 Different levels of detail. From left: envelope, feature removal, and original.....	73
Figure 38 Major layout revisions, top-view in RobotStudio.....	74
Figure 39 Example of mechanism creation within RobotStudio.....	75
Figure 40 Options selection during the creation of virtual controller.....	75
Figure 41 Schematic of module and program structure.	78
Figure 42 Structure of simulated PLC in RobotStudio.....	79
Figure 43 Example of a complex smart component, controlling the table.	80
Figure 44 Recording within Signal Analyzer in RobotStudio.....	81

Figure 45 Ladder code of the table.	83
Figure 46 Simulated PLC within RobotStudio.	84
Figure 47 The HMI for Viking in calibration mode.	85
Figure 48 Safe Zones visualized within RobotStudio.....	87
Figure 49 First page of the auto-generated safety configuration report.	88
Figure 50 Excerpt from the auto-generated safety configuration report, showing one of the twelve Safe Zones and its signals.....	89
Figure 51 Swept volume of where the robot would stop if a Cat. 1 stop were triggered.	90
Figure 52 Ghost image of where the robot would stop if a Cat. 1 stop were triggered.....	90
Figure 53 Echo, after importing the Studio 5000 project file.	92
Figure 54 UA-Server and endpoint configuration within FactoryTalk Linx Gateway.....	92
Figure 55 Shortcut setup in FactoryTalk Administration Console.	93
Figure 56 Graphical OPC-UA Client interface within RobotStudio.	94
Figure 57 Setting up the IO-Gateway in Arcestra System Management Console.	94
Figure 58 Configuring the Access Name in InTouch WindowMaker.....	95
Figure 59 Handshakes between PLC and robot in RAPID.	96
Figure 60 vFAT with the virtual environment, HMI, emulated PLC, and Robots.	99
Figure 61 Points marked for removal within Autodesk ReCap.....	101
Figure 62 Pointcloud of the factory imported into RobotStudio.....	101
Figure 63 Virtual meeting using VR in the virtual environment.	102
Figure 64 HoloLens Demo at MITC (Gardebrand, 2021).	102
Figure 65 CAD-model of plate separator with physical data, simplified due to confidentiality (Inventor 2022).	103
Figure 66 RobotLoad set up according to the tool data and payload with the IRB 4600-60 robot.	104
Figure 67 Load diagrams for an insufficient robot (left), and a superfluous robot (right).....	105
Figure 68 Major revisions of grippers, evolving from left to right. Image blurred due to confidentiality.	105
Figure 69 Fracture geometry of gripper after fatigue testing.	106
Figure 70 Generatively designed tool mount used in Viking.....	107
Figure 71 Data of work-related injuries, translated from The Swedish Work Environment Authority (2014).	108
Figure 72 Testing ergonomics and operation procedures using VR.....	109

1. Introduction

1.1 Company description

Alfa Laval is a world-leading manufacturer of products in the field of heat transfer, separation, and fluid handling. At the end of 2020, it had 17000 employees working in more than 150 sites in 100 countries (Erixon, 2021). The headquarters is located in Lund, Sweden, with 2000 employees. The company was founded by Swedish engineers Gustaf de Laval and Oscar Lamm in 1883, initially manufacturing centrifugal separators for the dairy industry. Centrifugal separators still play an integral part in the company's revenue (Alfa Laval, 2022).

1.2 Background

In the past years, Alfa Laval has put great effort into implementing Industry 4.0. Previous work related to virtual commissioning has been conducted in-house (Persson & Norrman, 2018) and by external parties. Alfa Laval wants to better understand how virtual models can further benefit the company to continue on this path. In this effort, the use of complete virtual commissioning is of particular interest as it has shown great potential but has previously been unavailable due to software incompatibility. Therefore, Alfa Laval wants to investigate how it can be deployed in their factory in Lund and potentially globally. Furthermore, Alfa Laval wants to investigate leveraging the virtual commissioning models to extend outside of the commissioning.

1.3 Objective

This study investigates the concept of virtual commissioning in the realisation of production systems, its benefits, limitations, and scalability. The methodology will then be practically tested on an actual project. The study will also investigate how the models built for the virtual commissioning

can be further used to utilize the models' full potential. The main objective will be divided into three parts:

- Perform a literature study regarding cases where the virtual model has been used in connection to commissioning, including academic and industry sources; what benefits it gave, what resources were required, and its limitations.
- Create a virtual commissioning model, which should have the possibility to test and verify low-level PLC code, robotics code, HMI, and mechanisms movements.
- Evaluate different uses of the virtual model and propose a way forward to work with virtual models within Alfa Laval operations based on the findings.

1.4 Delimitations

The main focus of this thesis will be on virtual models in connection to virtual commissioning. Other topics and concepts concerning virtual models, in general, will be covered insofar as they are part of the project.

1.5 Confidentiality

Sensitive information regarding Alfa Laval has been removed or generalised. Parts shown from the production cell have been simplified to the extent that they still carry theoretical value without revealing company secrets. The proposed way forward is only a recommendation made by the author and does not foretell decisions about the implementation of the methods described herein.

1.6 Target audience

The target audience for this thesis is mainly automation engineers, production technicians, project managers, and decision-makers in the manufacturing industry. However, the methods can still be applied in any company within the industry that uses virtual models.

1.7 Thesis Outline

The thesis is divided into seven chapters: Introduction; Theoretical background; Literature study and industry research; Viking – from idea to commissioning; Discussion; Conclusions; Recommendations for Alfa Laval. Chapter 2 will provide theory regarding virtual models, industrial automation, and virtual production, which will equip the reader with the necessary insight to get the most out of this thesis. Chapter 3 presents a literature study and observations from the industry, which will serve as the groundwork for the later development of the virtual commissioning model. In Chapter 4, the knowledge from the previous chapters will be applied to an actual project at the Alfa Laval. The virtual commissioning model will be tested and assessed to validate the knowledge gained during the literature study and model creation, ensuring that the proposition for future commissioning work at Alfa Laval will be robust. In addition, other applications for the virtual model will be explored and analyzed. In Chapter 5, the work is discussed from both an industrial and academic view. Chapter 6 concludes the thesis together with potential future work. Lastly, Chapter 7 presents recommendations for Alfa Laval on how to continue work with virtual commissioning and virtual models.

1.8 Methodology

The purpose of scientific research is to gain knowledge systematically. The methods of doing so are countless and must be suited to the specific task (Wilson, 1990). The following section presents three methods used in this thesis.

Literature survey

The literature study has been undertaken as an integrative literature review, but the selection of papers were made similar to a systematic review. In contrast to systematic reviews, the purpose of the integrative review is not to cover every article written but to draw overall conclusions from past research. Because of this, there is an increased risk of bias if the search strategy is not well-defined. The iterative review methodology may use data from both theoretical and empirical literature (Whittemore & Knafl, 2005).

Interviews

The use of interviews enables fast and efficient data collection. Unlike questionnaires, there is the possibility of follow-up questions and deeper discussions on the subject. The questions asked during the interview depend on the type of data to be collected. For exploratory interviews, an open, flexible, and less structured interview may be suitable as they invite more developed answers and permit follow-up questions. Statistical surveys often deploy stricter interviews as uniformity is required to gather pure and unbiased data. In those cases, the questions must be asked the same way to all participants (Hanington & Martin, 2019).

The flip side of interviews compared to questionnaires is that they are more time-consuming, often resulting in a smaller sample which can be problematic if the data is to be used to generalize across a population (Rowley, 2012). In this case, the data is not used for this purpose but aims to highlight visions and challenges in the industry in more detail.

Industry-as-laboratory

Industry-as-laboratory is a methodology first proposed in the software engineering field by Potts (1993) and later applied to the industrial design field by Muller (2011). The method uses the industry setting as a source of inspiration to define problems. Through research, a new approach is hypothesized and tested through experiments in the actual application.

The results from the experiment are analyzed, and after evaluation, the method can be improved further in an iterative manner (see Figure 1).

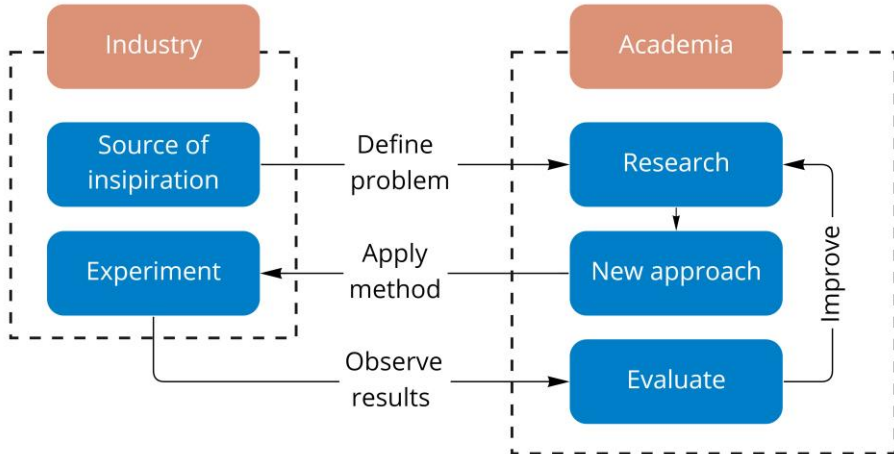


Figure 1 Industry-as-laboratory, inspiration from Potts (1993) and Muller (2011).

The benefit of using Industry-as-laboratory is that it bridges the gap between the well-defined mono-disciplinary academic research and the soft multi-disciplinary problem resolutions needed in the industry. Another benefit of the industry-as-laboratory approach is that it is easier to attribute value to the method with industry metrics such as time, cost, or quality (Muller, 2011).

The Viking project will serve as the testing ground for this thesis (see Chapter 4). The knowledge from the literature review will be incorporated into the interview questions, and the results from both will comprise the basis of the practical application.

2. Theoretical background

This chapter has a bottom-up approach, beginning with the elementary virtual models and their industry applications, how they are used in virtual production, and ending with the virtual commissioning.

2.1 Virtual Models

2.1.1 Computer-aided design, manufacturing, and engineering

Computer-aided design (CAD) is used to create and manipulate virtual objects, provide manufacturing details, and validate basic functions such as movement, strength, and producibility. The virtual objects can be created from scratch using sketches and features, or if the physical object already exist, by using 3D scanning and other reverse engineering methods (Stark, 2022a). The term originated during the 1960s when General Motors, together with IBM, launched the DAC-1 (Design Augmented by Computers). The first CAD software did not process 3D objects but was mainly for electric drawings. Nowadays, however, CAD is used for a variety of applications (Encarnacao et al., 2012).

Computer-aided engineering (CAE) is used to simulate the CAD model to evaluate properties in relation to property requirements such as strength, durability, and frequencies. Traditionally these calculations had to be done by hand, often complemented by a physical prototype that was thoroughly tested. This time-consuming task is significantly sped up with CAE. Within CAE, there are multiple disciplines. These are topology optimization, kinematics simulation, structural analysis, computational fluid dynamics, and manufacturing process simulations, to name a few (Stark, 2022c).

2. Theoretical background

Computer-aided manufacturing (CAM) takes the CAD models a step closer to realization. It is used to create the programs fed to the manufacturing machines. For simple 2D programs, this can be done manually, but programs can quickly become unmanageable if CAM software is not used. A widely used program format is G-code which gives instructions row by row for each movement. Seen in Figure 2 is a manually programmed milling process with 22 rows of G-code and an automatically generated 3D-printing process with over 120000 rows.

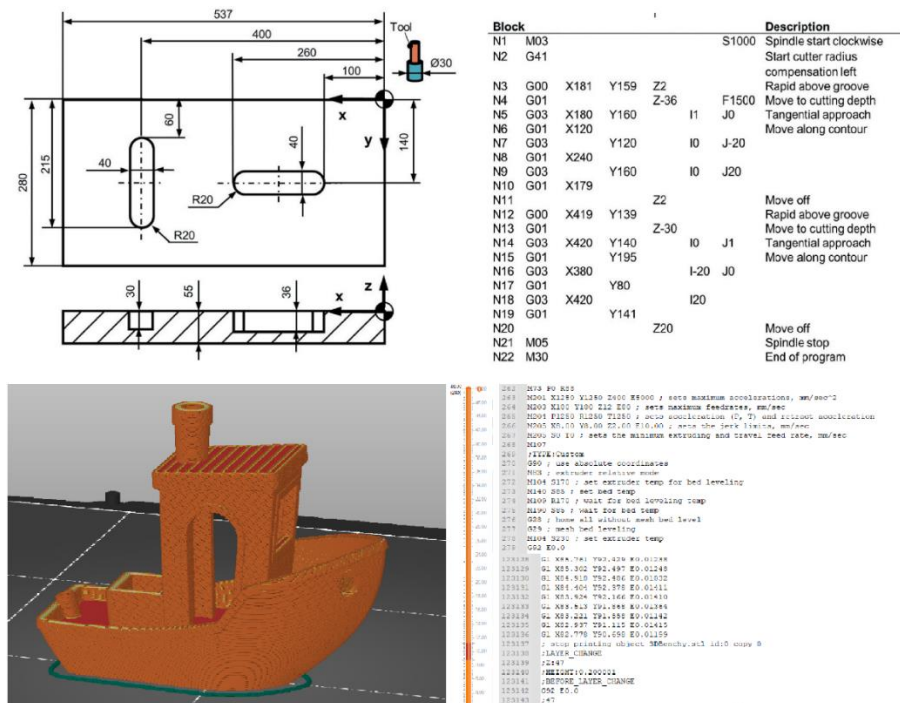


Figure 2 Top: Simple milling process with 22 instruction rows. Image reproduced with permission from Rainer Stark. Bottom: AM process with 120000 instruction rows.

With a graphical CAM system, tool paths can be simulated to give the engineer a visual validation of the program and ensure no unwanted behavior (Stark, 2022b).

2.1.2 3D-scanning

Instead of constructing an object ground up, 3D scanning can be used to create a virtual representation of an already existing physical object which can be especially handy when evaluating if a new component will fit in an existing compartment or reverse engineer legacy parts no longer produced (Helle & Lemu, 2021).

3D scanners are generally divided into contact and non-contact scanners. The contact scanners use a physical probe to scan the object's surface, resulting in a high-resolution scan but is severely limited in speed and flexibility. The non-contact scanners typically probe the object by projecting light and measuring changes in the reflection and can therefore be portable and have a range of several hundred meters. A handheld portable 3D scanner is shown in Figure 3. The resolution of non-contact scans is generally less, especially when scanning larger areas, but they capture much data in a short time (Ebrahim, 2015).

Measuring discrete points in space results in a point cloud floating in space. A typical point cloud can contain millions of points depending on the resolution. Point clouds can be part of virtual commissioning by validating component placement, potential differences, and interference in the physical world (Jakob & Nilsson, 2018).

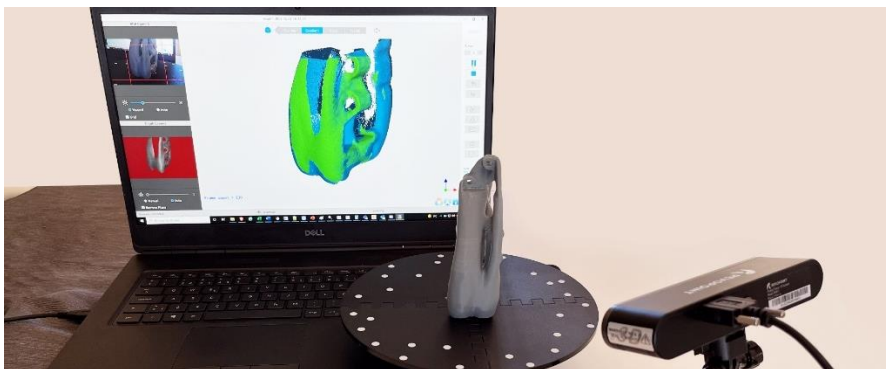


Figure 3 Handheld 3D scanning using a Revopoint scanner and their software.

2.1.3 Additive manufacturing

Additive manufacturing (AM) is the umbrella term for technologies that takes a virtual 3D model and manufacture it by adding material. The concept of Additive manufacturing has been around since 1986 when Charles Hull (1986) filed a patent for an *Apparatus for production of three-dimensional objects by stereolithography*. Since then, the concept has been developed and has become a multi-billion dollar industry with an expected compound annual growth rate of 29% for the next five years. With declining printing costs, the technology has become attainable for everyday people, making the manufacturing technique reach global adoption (Tofail et al., 2018).

Additive manufacturing can produce complex structures that would be impossible to create with the usual subtractive manufacturing methods, such as a milling machine or lathe. As no special tooling is required, the method excels at low volume, high complexity production, giving the customer more freedom of design and personalization. The short setup and throughput time make it excellent for just-in-time production.

AM is the primary driver of *rapid prototyping*. Concept modeling can convey and demonstrate design concepts but does not necessarily function. Functional prototyping does the contrary, focusing on verifying functions of the product and design not crucial for the functionality is excluded. AM can provide both types of models (Gebhardt, 2011).

The first step in using additive manufacturing is creating a virtual model in CAD software. The model is then converted to a file format that the AM software can use. The most common, albeit old format, is STL (stereolithography) which creates a mesh of triangles of a user-specified resolution (see Figure 4). A recently published format, 3MF (3D Manufacturing Format), is much improved with smaller file size, better model quality, and the possibility of specifying material and colors in the file (Diegel et al., 2019).

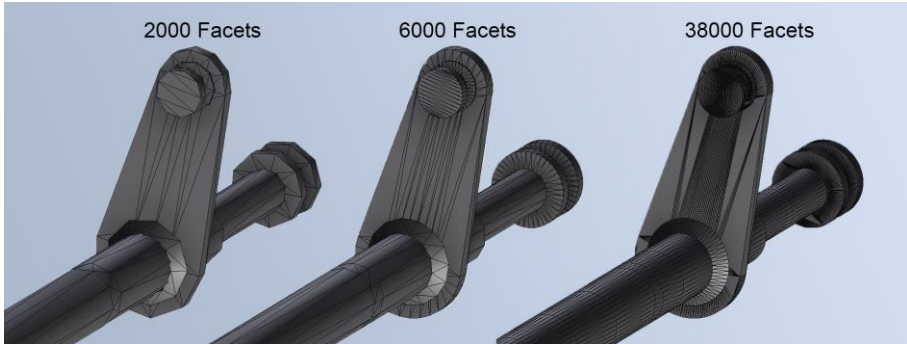


Figure 4 Different detail levels, file size scale proportionally with the number of facets.

The triangulated file is then imported into a *slicer software* . As the name suggests, a slicer is software that slices the model into stacked 2D layers. The thickness of the layers depends on the desired quality, printer hardware, and print time. Popular open-source slicers are PrusaSlicer and Cura, which allow users to change hundreds of parameters to fine-tune the print.

The slicer output is then fed to the machine, which begins building the structure. The technological principle of the print depends on which type of AM machine is used. The three most popular methods are Fused Deposition Modeling (FDM), Selective Laser Sintering (SLS), and Stereolithography (SLA), illustrated in Figure 5. FDM works by melting spooled filament and depositing it through a nozzle in stacked 2D layers; this is the most common method of AM due to its inexpensive components (Diegel et al., 2019). In the SLS process, a fine powder between 20 to 80 microns is spread over the build plate and melted by a laser beam reflected onto the build plate. After the laser has melted the entire layer, the platform is lowered, and a new layer of powder is spread on top. This method is especially used in metal printing (Mueller, 2012). Some 3D printers use photocurable liquid resin, which hardens through a photochemical process where light from a laser (SLA), projection (DLP), or liquid screen display (LCD) causes a polymerization through increased crosslinking.

2. Theoretical background

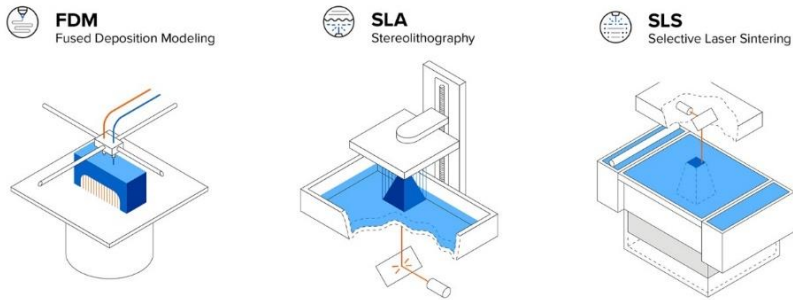


Figure 5 Different methods of Additive Manufacturing. Image reproduced with permission from Formlabs.

The polymerization rate is generally low and is still done in discrete layers, but efforts for a faster continuous curing process are being investigated. Post-processing of printed parts is generally needed where the piece is rinsed in isopropyl alcohol and further cured in a UV chamber (Manapat et al., 2017).

The layers of FDM prints give the part anisotropic properties, being noticeably weaker in the z-direction during tension but not as noticeable during compression. The SLS and resin prints are more isotropic and can provide full strength comparable to a cast or injection molded component, making them suitable for end-use parts (Diegel et al., 2019).

Nowadays, many CAD software come with slicers natively, which lowers the threshold for mechanical designers to begin using AM. It is advised to design the model specifically with AM in mind¹ as benefits such as decreased weight, improved fluid flow, and decreased volume may be lost otherwise. Designing for additive manufacturing is, in many ways, the opposite of conventional designing as traditional manufacturing methods are subtractive. Because of this, it is crucial to have a clear picture of the parts

¹ Often called DfAM, Design for Additive Manufacturing

function and the companies manufacturing abilities before the design process starts. Producing a part designed for conventional manufacturing in an AM machine is rarely beneficial (Diegel et al., 2019).

2.1.4 Generative design

Generative design is a method of iteratively searching for a solution to a design optimization problem. The generative design engine employs processes similar to nature's evolutionary approach using genetic variation and selection, often giving the structures an organic shape. Each iteration is continuously evaluated, and the engine learns from each change to find an optimal solution to the user-defined goal. The user goal can be defined in properties such as weight, strength, stiffness, frequency, material, build orientation, and manufacturing method (McKnight, 2017).



As the methods for manufacturing generatively designed parts is mainly additive, there is little to no waste. Using generative design allows constructions with no weak points and virtually homogenous strain and stress throughout the model. This minimizes the amount of material needed, saving manufacturing time, material cost, and our planet's resources (Briard et al., 2020).

Additive manufacturing is the main enabler for generative design as it is often difficult to manufacture the generated models with traditional manufacturing methods. With additive manufacturing growing to include all kinds of materials and properties, the possibilities are endless (Briard et al., 2020).

2.1.5 Model-based definition

Traditionally the 2D drawing has been the authoritative source of geometrical data for manufacturing. With the emergence of CAD and 3D modeling, creating drawings became more manageable, but the 2D drawing was still the primary document used to communicate the design. Model-based definition (MBD) aims to eliminate redundant documentation and ambiguity by having the 3D model as the sole source of product information. Product Manufacturing Information such as tolerances, design requirements, and material specifications would instead be represented in the CAD model (Bijnens & Cheshire, 2019). This simplification helps lower the complexity of PLM systems by reducing the amount of documentation, which is crucial due to the steadily increasing data sources and interfaces in modern industry (Alemanni et al., 2011).

2.1.6 Product Lifecycle Management

Product Lifecycle Management (PLM) software manages assets from a product during its entire lifecycle, from ideation to disposal. These assets can consist of CAD data, BOM lists, usage documentation, orders, quality control, maintenance information, and disposal information, among other things. The PLM software lets different stakeholders access data about a product and make sure to keep track of changes and relations between the data. PLM objectives can be to increase product portfolio value, increase traceability, or reduce cost from claims (Stark, 2020).

2.2 Industrial Automation

The ISA-95 hierarchy model shown in Figure 6 is commonly used to describe industrial automation's different levels and interfaces. Virtual commissioning generally seeks to validate controllers and their interfaces towards supervisory systems and field equipment; hence only level 0 to level 2 will be described.

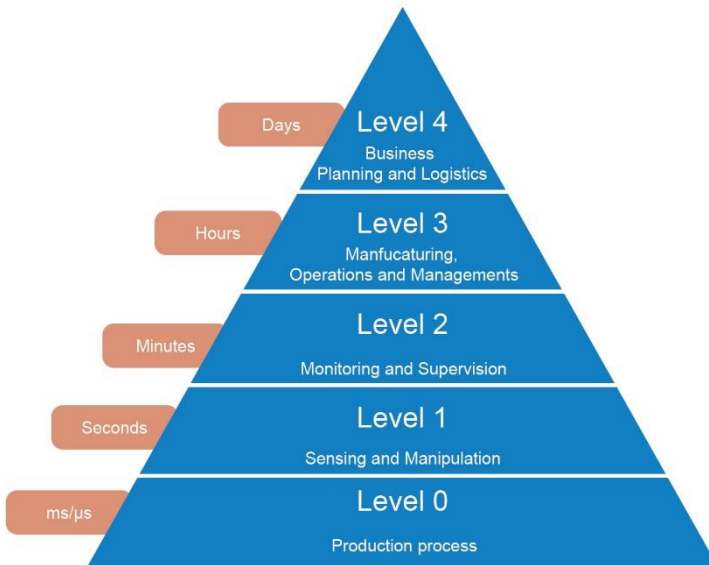


Figure 6 Automation hierarchy based on ISA-95 with approximate time frames.

2.2.1 Sensors and actuators

Field devices are found at the bottom of the automation pyramid and comprise sensors and actuators. Sensors measure manufacturing process parameters such as distance, temperature, reflectivity, vibrations, flow, or pressure. Actuators act on the manufacturing process using pneumatic, hydraulic, or electrical devices. Output signals from the sensors and input signals to the actuators are generally communicated through a controller (Ikram & Thornhill, 2010).

2.2.2 PLC, DCS

PLCs are the backbone of industrial automation as they provide reliable and flexible logic. A PLC is a purpose-built real-time computer that takes an input, makes a computation based on a stored set of programmed rules, and gives an output. Prior to PLCs' invention in 1968, the logic was achieved through relay panels connected with cabling. An example of such a relay panel can be seen in Figure 7. Using relays was time-consuming,

2. Theoretical background

the logic was hard to change, and the relays themselves were large, hot, and power-hungry (Segovia & Theorin, 2012).

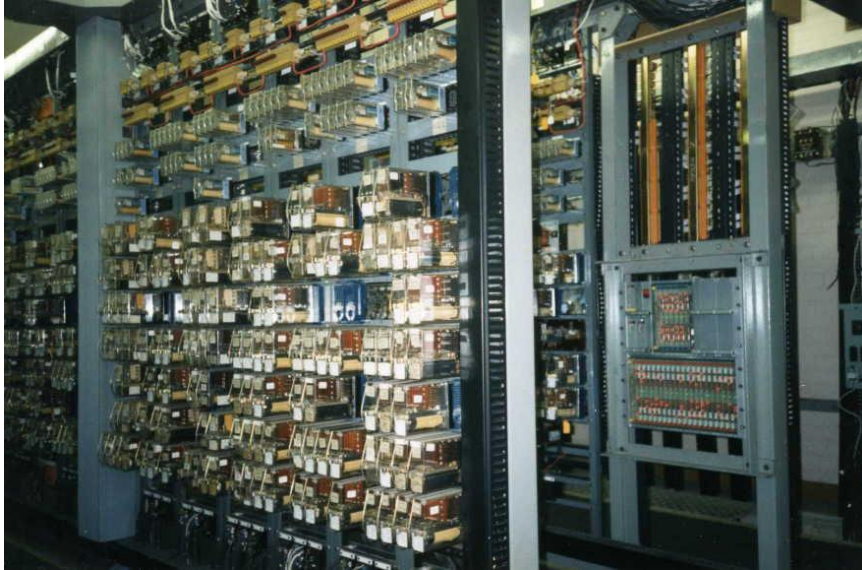


Figure 7 An old relay panel with plugin relays. Image reproduced with permission under Creative Commons (Signalhead, 2007).

The PLC executes programs one rung at a time using scans, where each rung in a program gets checked before the next. The scan time of a PLC refers to the time it takes for the CPU to read input, execute the entire program and give an output. The scan time depends on the program's complexity but is typically at a millisecond-level (Bolton, 2015).

PLCs are often modular, where the modules share the same backplane. Generally, a PLC consists of a power supply module, a CPU module, and various communication modules (Figure 8). The modules are rugged and can operate in demanding environments (Bolton, 2015).

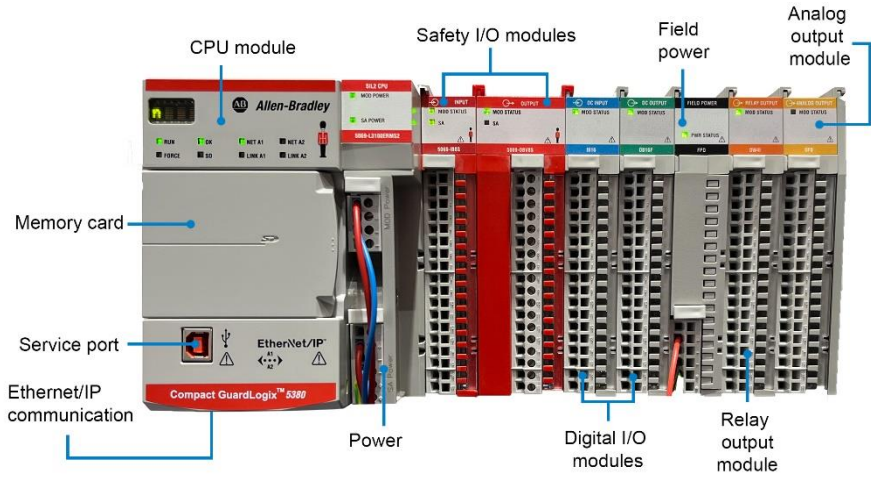


Figure 8 A modular PLC with various modules.

The Distributed ControlLogix System (DCS) was invented around 1975 to handle multiple controllers in large systems or entire plants with better security and robustness than the single point of failure PLC. The differences between the two have lessened over the year, and modern PLC networks perform similarly to a DCS, excluding the DCS integrated supervisory control and data acquisition that would require a separate SCADA system for the PLC. DCSs are suited for large continuous processes geographically distributed across the plant, typically seen in industries such as refineries or food processing plants (Mehta & Reddy, 2014).

Programming languages

The IEC 61131-3 standard (2013) specifies the requirements of PLC systems and defines three graphical and three textual programming languages that are briefly described below. The standard does allow other programming languages to be used as long as they fulfill the base requirements regarding compliance, consistency, and standard functions.

Sequential Function Chart (SFC)

Sequential Function Chart is a descriptive language used to describe the structure of programs in the PLC, visually showing the sequential and parallel execution. It consists of steps programmed in any of the other four IEC 61131-3 languages, and between every step is a boolean transition condition that lets the program continue to the next step when fulfilled (Figure 9). While SFC is both a graphical and a textual language, the textual version is primarily used to simplify backup or information exchanges between systems (Tiegelkamp & John, 2010).

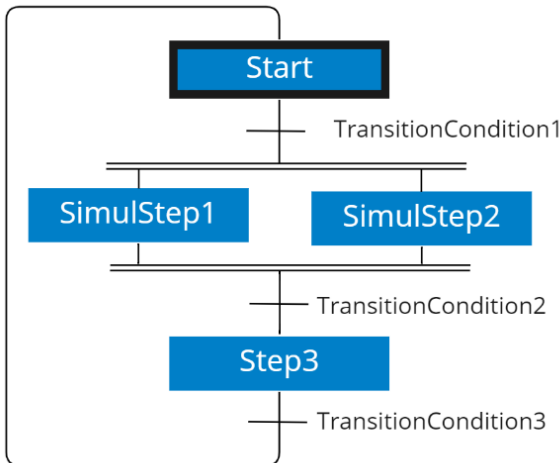


Figure 9 Simple example of an SFC with a parallel branch executing two steps simultaneously.

Ladder Logic Diagram (LD)

The Ladder Logic language is the oldest PLC programming language and was created to replicate the physical relays. The ladder network is built using rungs connected between the left and right power rail, giving it its ladder-like appearance, as seen in Figure 10. Input conditions are put on the rung that must be fulfilled to let the power flow through the output. Present-day ladder logic is generally combined with function blocks to

achieve more advanced operations or to contain programs for reuse (Commission, 2013).

Due to Ladder Logic's popularity during the third industrial revolution, many companies continue to use it because of its familiarity among technicians. It is currently listed as the 50th top programming language (Spectrum, 2021).

Instruction List (IL)

Instruction List is a low-level textual language where each instruction is described in precisely one line. An instruction has an optional label that can jump between lines, an operator or function, zero or more operands, and an optional comment. The example in Figure 10 shows the code for a flipflop where the first column is the operator, and the second is the operand. In the example, the label and comment have been omitted. Being a low-level language, IL has less overhead resulting in very efficient program execution and memory usage. The latter is not as critical in modern PLCs, where memory is less limited.

Function Block Diagram (FBD)

Function Block Diagram originates from the signal processing field and uses graphics that look like electronic circuits interconnected through connecting wires. It is commonly used in DCS and continuous process manufacturing. The function blocks can easily be reused and have the benefit that they can be programmed in any of the other IEC 61131-3 languages (Ramanathan, 2014).

Structured Text (ST)

Structured Text is a high-level programming language, similar to conventional programming languages like Java or Python. As it is a textual language, it has no graphical interface. A considerable benefit of ST is that the general knowledge of computer languages increases, lowering the bar to getting into control logic (Antonsen, 2020).

2. Theoretical background

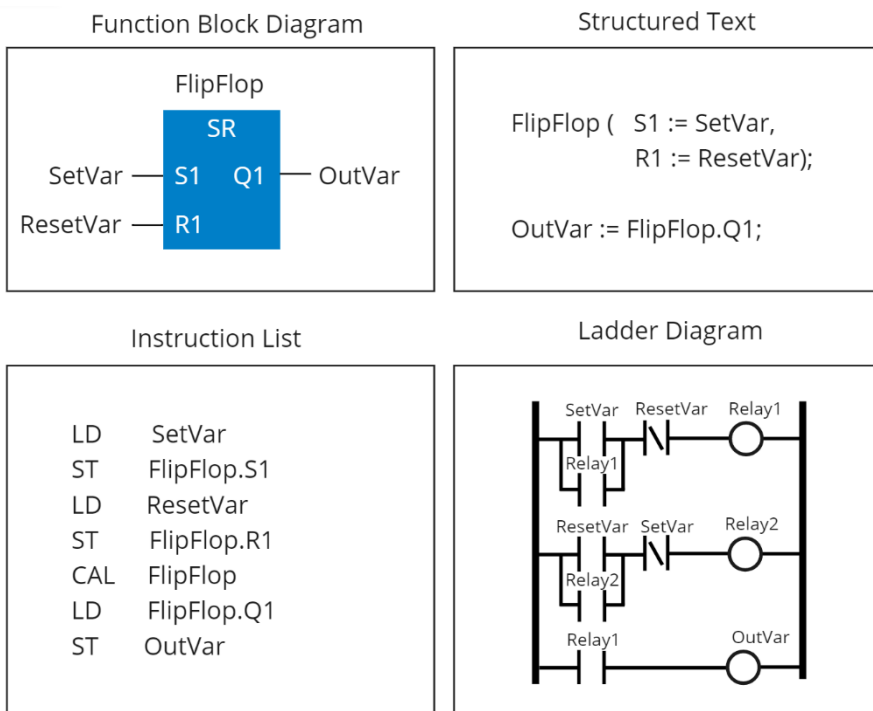


Figure 10 Equivalent FlipFlop-function programmed in FBD, ST, IL, and LD.

Language usage

Which language to use depends on the application, preferences within the company, or geographical location. Ladder logic is preferred in the US, while European industries prefer Structured Text. Function Block Diagram is commonly used in process control applications and Distributed Control Systems (Ramanathan, 2014). An n-gram search² of the languages is presented in Figure 11, which shows that ladder logic and structured text are the most popular in literature.

² The Google Ngram engine counts mentions of terms in millions of books and other literature. The result were checked for term overlapping.

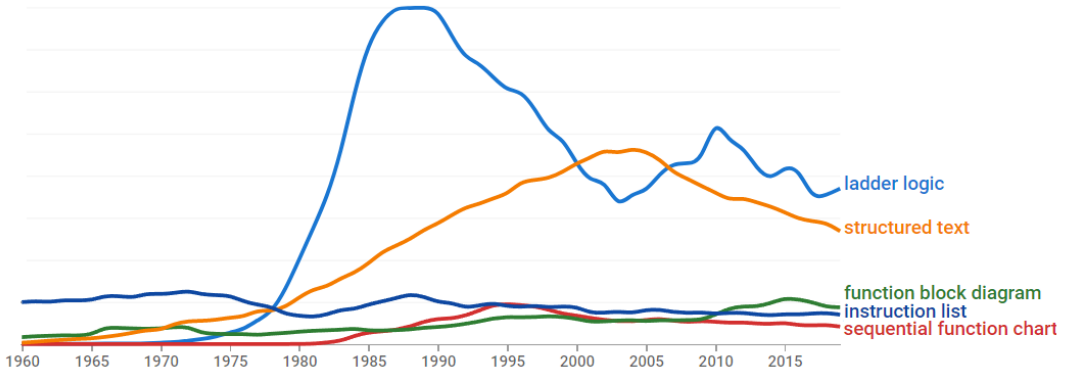


Figure 11 Relative mentions of terms in literature from 1960-2019 (Michel et al., 2011).

2.2.1 Industrial robots

According to the ISO definition, an industrial robot is “*an automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications.*” Multipurpose means that the robot is adaptable to different applications through physical alteration ("ISO 8373:2012(en)," 2012).

Making the manipulator multipurpose is usually achieved by using different end effectors. An end effector is the device mounted on the last link of a manipulator and is usually, but not always³, the part of the robot that interacts with the environment (Bolmsjö, 2006).

A robot consists of actuators making the joints move and sensors to control the movement through a robot controller, similar to the earlier description of how sensors and actuators interact with a PLC. In contrast to PLCs, robot programming is usually done through more conventional text-based languages such as C++ or Python. Many robot manufacturers use their

³ See <https://youtu.be/ASEtz2M1RiY> for a practical example where the YuMi wraps a Christmas gift while holding it in place with the fourth link arm.

2. Theoretical background

own languages; ABB uses RAPID, KUKA uses KRL and Fanuc uses KARREL, all of which are based on the high-level language Pascal (Biggs & MacDonald, 2003).

Degrees of freedom

An object's degrees of freedom correspond to the object's total translational and rotational components. Industrial robots are commonly described by their number of axes, which often correlate with their degrees of freedom.⁴ A typical CNC machine can move in the x, y, and z directions and therefore have three degrees of freedom. Adding a rotational component will result in a 4-axis robot, generally found in *Pick & Place* applications, where objects can be grasped without the need for tilting the end effector (Bolmsjö, 2006).

While there are robots with seven axes or more, the 7th axis generally refers to track motion systems where the entire robot is translated to extend the work envelope (Patel et al., 2014). Figure 12 shows such a system that allows the robot to weld on one station while the other is prepared.

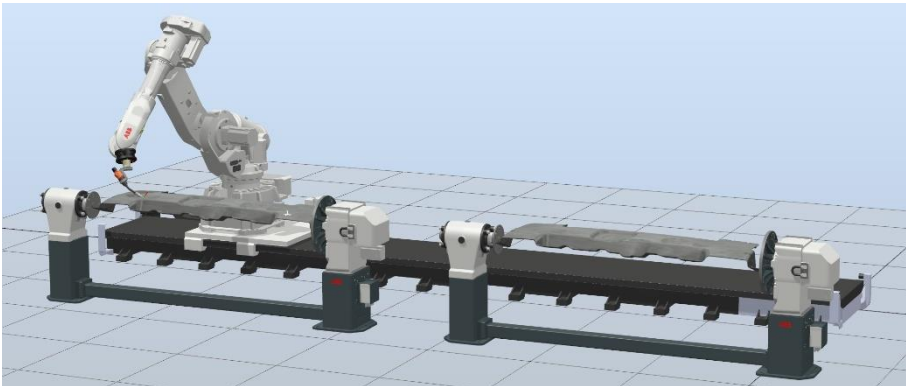


Figure 12 Robot mounted on a linear 7th axis to extend the workspace.

⁴ A robot can have more axes than degrees of freedom, but never the other way around.

Mechanical capacity

Common to all robot applications is the movement of some object from one position to another. Therefore, the main requirements when selecting a robot are reach and payload capacity, which usually are how robot manufacturers present their robots. The payload capacity is only valid for loads with a center of gravity close to the flange, and the capacity is rapidly declining as the distance from the flange increases, but limiting movements so that the wrist is always pointing downwards can increase the capacity as can be seen in Figure 13. A significant moment of inertia can also limit the choice of robots. Robot manufacturers often provide software that aid calculations of robot load, for example, KUKA.load and ABB's RobotLoad, where mass, center of gravity, and moment of inertia can be put in to recommend a robot model (ABB, 2022b; KUKA, 2022).

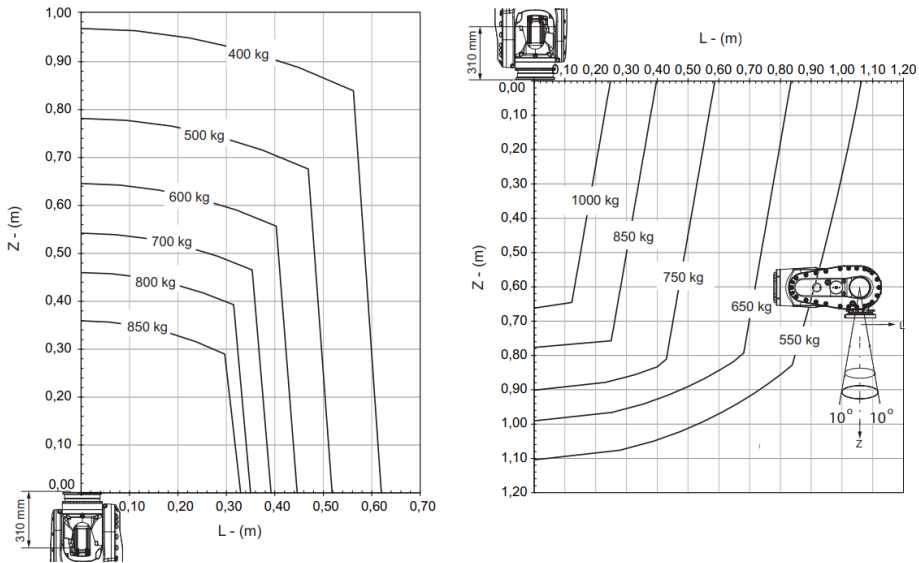


Figure 13 Load diagrams of the ABB robot IRC8700. Right: load diagram for vertical wrist (ABB, 2022b).

Types of robots

Common types of robots are industrial, collaborative, SCARA, and Delta-robots, all of which have different application areas (Figure 14). The rigid

2. Theoretical background

but light arms of the delta robot provide outstanding acceleration of up to 10 g, with typical cycle times of 0.35 seconds, making it perfect for in-line pick and place operations (ABB, 2022a). As the name suggests, the collaborative robot is designed to work collaboratively alongside a human operator. Such robots often use various techniques to mitigate the risk of injuries, such as pinch-free joints, force sensors to detect contact, and laser scanners to sense humans.



Figure 14 Different robot types from the four largest robot companies. From left, KUKA 6-axis industrial robot, ABB 6-axis collaborative robot, Yaskawa 4-axis SCARA robot, and Fanuc 4-axis Delta robot.

Accuracy and repeatability

Specific applications may need high position accuracy and repeatability. Inaccuracies are generally caused by backlash in the motors and joint flexion. Performance testing is usually carried out according to ISO 9283, which specifies testing conditions. Notably, all six axes have to be in motion, and the test must be run at maximum load and reach at varying speeds. ("ISO 9283:1998(en)," 1998) For modern robots, accuracy and repeatability are around ± 0.05 - 0.2 mm. The specified values apply to poses created in the physical application, i.e., targets made virtually might not have as good accuracy.

Coordinate systems

For the robot controller to be able to move the manipulator to the desired position, the target must be defined in space to some reference coordinate system⁵. The coordinate systems described in Figure 15 are used by ABB (2021), but other brands set up their coordinate systems similarly.

The world coordinate system (a) is only utilized in the simulation to relate virtual objects to a fixed position and is not seen by the physical robot controller. The task frame (b) is defined in the world coordinate system and is usually aligned with the base frame when using a single robot and aligned with one of the robots in a system with multiple robots. The base frame (c) is always located at the robot's base. The user frame is defined in relation to the task frame, and the object frame is defined in relation to the user frame. The tool center point (f) is defined in relation to the robot flange, named tool0 (g).

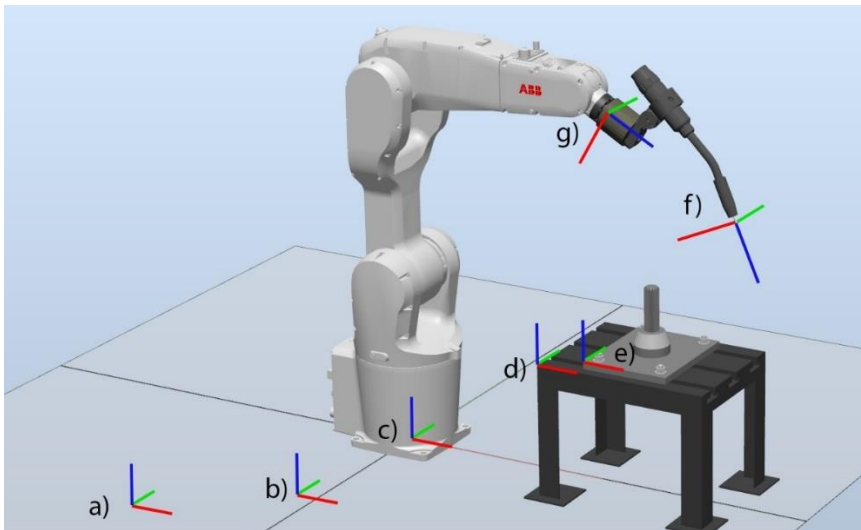


Figure 15 Coordinate systems used by the robot. a) world, b) task frame, c) base frame, d) user frame, e) object frame, g) tool0, f) TCP.

⁵ Sometimes referred to as a frame

Workobjects

When doing offline programming, i.e., not connected to the robot, every object has a well-defined position and orientation in space. It is impossible to get this absolute precision when assembling the physical cell. Workobjects are used to solve this precision issue. A Workobject consists of a user frame and an object frame. The user frame is expressed in relation to world or base coordinates, and the object frame is expressed in relation to the user frame, as visualized in Figure 16. The robot targets can then be related to the object frame (ABB, 2021).

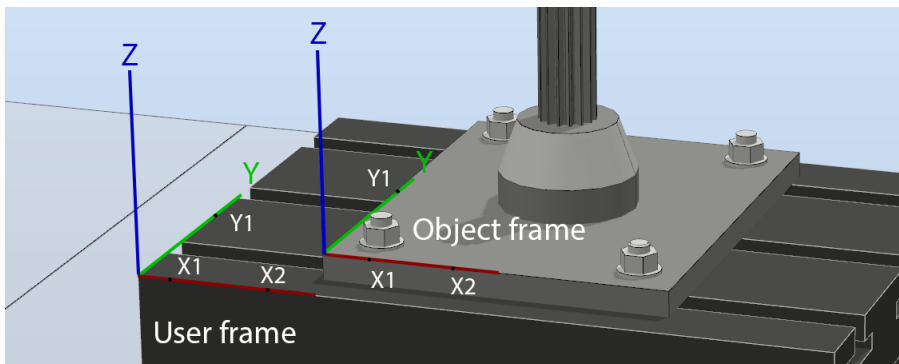


Figure 16 User frame and object frame defined.

After the physical assembly, the work object is defined by providing two points that lay on the x-axis and one point on the y-axis of the coordinate system in the physical application. From these points, the position and orientation can be calculated, which is typically done with the help of the FlexPendant that guides the procedure, as seen in Figure 17 (ABB, 2021). The coordinate system will then be transposed to the new, correct location, and all targets will follow. If the position of a physical object is altered, either by mistake or intentionally, every target need not be reprogrammed; only the work object will have to be redefined.

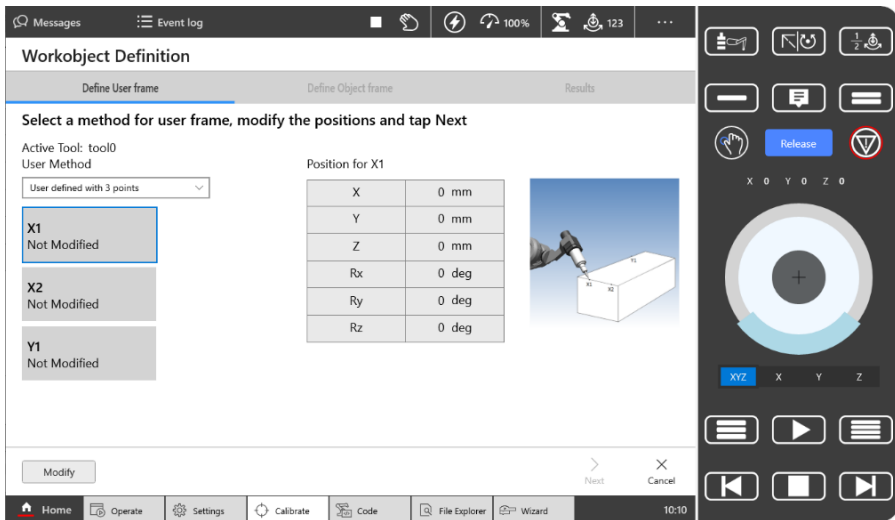


Figure 17 ABB's Flexpendant with Workobject Definition Wizard.

Configurations and singularities

As visualized in Figure 18, a typical six-axis industrial robot can reach a target in multiple ways. Only providing a position and orientation would therefore be ambiguous. This ambiguity could cause undesirable behavior as some configurations might cause collisions with other objects. Because of this, the configuration is defined in robot programs, usually as a set of values describing in which quadrant the axes are located.

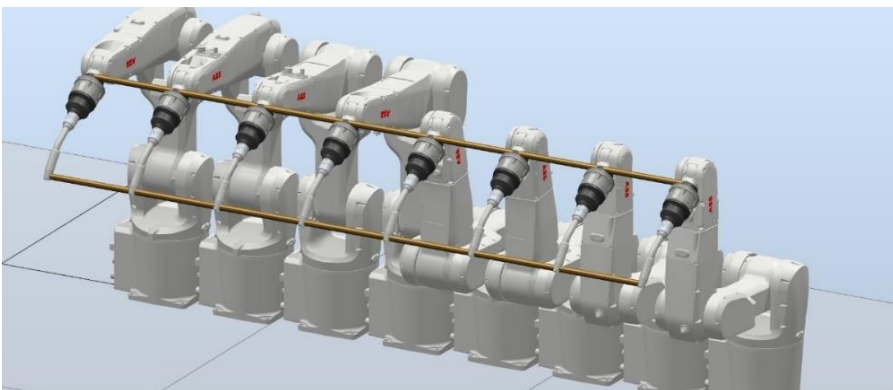


Figure 18 All eight configurations to reach the same position and orientation of a specific target; each robot is offset 300mm, and two parallel rods are added for clarity.

Certain positions in the workspace can theoretically be reached in an infinite amount of ways. These points are often called singularities and can cause undesirable behavior as there is no single solution to the inverse kinematics equations. For a typical 6-axis robot, arm singularities occur when the wrist center⁶ is aligned with axis one or when the wrist center is aligned with the plane going through axis two and three. If not constrained, at least one axis will spin at a high velocity to keep up with the programmed path when going through or close to a singularity. Practically this is solved by putting limitations on axes' movements during movements close to singularities (Bolmsjö, 2006).

2.3.3 SCADA and HMI

Supervisory Control And Data Acquisition, or SCADA, is the system architecture of software and hardware that allow remote control and data collection in real-time. SCADA was created to reduce the need for routine checks by personnel on-site. This becomes especially important in plants scattered over large areas or with a large number of field devices and can typically be found in power plants and food processing facilities. The data from field devices and HMIs are generally fed to the PLC, which transmits it via LAN to the SCADA system. Historical data may be logged in a database or historian software for future use, such as plotting trends or finding bottlenecks. Modern SCADA software is typically vendor agnostic and can connect using a variety of protocols (Boyer, 2009).

The Human-Machine Interface, or HMI, is the communication link between the human and the machine, making it crucial for operations. A well-designed HMI should use natural mapping and provide feedback. The knowledge about the operation of a machine should be put in the world to let non-experts operate the machine, but not hinder efficient use by the knowledgeable user. The system's abilities should correspond to the user's intentions and perception. The system should also provide information

⁶ The intersection of the last three axes in robots with spherical wrists.

about its current state and how subsequent actions will affect it⁷ (Norman, 2013). An example of a good HMI design is shown in Figure 19. The text and numbers have been arranged and sized in a way that allows for much information to be shown without the interface being messy.

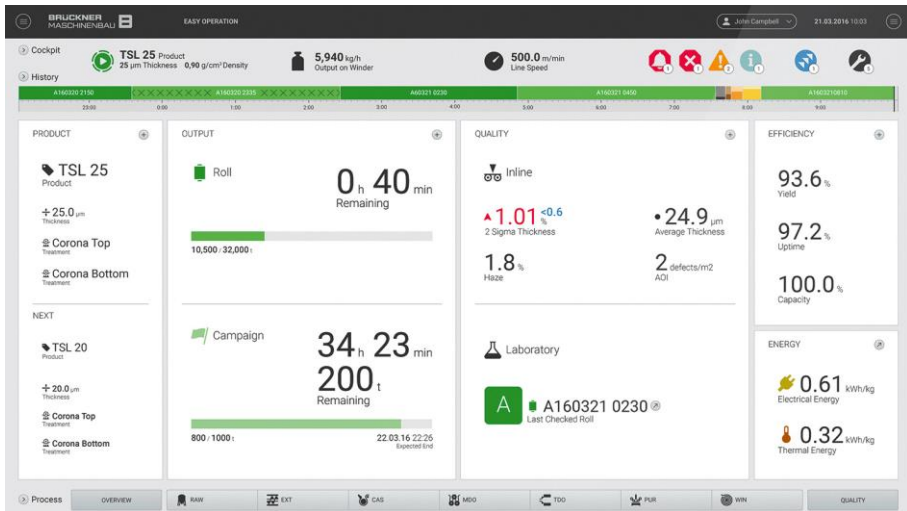


Figure 19 Award-winning HMI Brückner Maschinenbau GmbH. Image reproduced with permission.

It highlights important metrics and uses red to signal alarming values. The bottom bar is sorted into process steps and signifies that additional information can be gathered by clicking on the buttons. By giving information about the current and next order, the operator can be confident that the correct order is being processed and prepare the next one.

2.2.4 Communication between automation levels

There has to be a communication link between automation devices and systems to work in unison. Multiple data exchange protocols such as EthernetIP, Profinet, and OPC are available to aid this communication. The communication hardware often determines which protocol will be used; for

⁷ Within the area of interaction design these concepts are generally referred to as The Gulf of Evaluation and the Gulf of Execution.

example, Allen-Bradley PLCs use EthernetIP while Siemens PLCs use Profinet. OPC is a broader protocol used by various suppliers (Lin & Pearson, 2013). IO-Link is a protocol created for short-distance communications with sensors and actuators. The IO-Link device can then be connected to the control unit via other communication protocols (Heynicke et al., 2018).

2.2.5 Safety

In order to create a satisfactory safety solution, certain requirements must be met. There are various standards to follow to help meet these requirements, such as EN ISO 13849-1. The method for developing personal safety is described in the standard in three steps. The first step is risk assessment, which begins by determining the scope of the machine. All risk sources within the scope of the machine must then be identified and categorized in so-called *Performance levels*, ranging from *a* to *e*, and determine the needed reliability for a particular application. A low-risk process requires only safety level *a*, while a high-risk process requires a higher level of security. The second step is risk reduction through, for example, redesign, protection, or information. If the risk reduction is made through safety devices, the third step is to design and evaluate the safety functions. Safety functions consist of some input, logic, and output. The input can be emergency stops, safety sensors, or light curtains, the logic can be a safety PLC and the output can be switches, safety relays, and contactors. Each device has a probability of dangerous failure per hour (PFH_D), and the sum of the probabilities will determine the safety function's performance level. All safety functions must meet the required performance level ("ISO 13849-1," 2015).

Sistema is a free Safety Integrity tool for determining safety function performance levels and generating technical documentation. Many safety product manufacturers have prepared SISTEMA libraries for their products, making the configuration process quicker (IFA, 2020).

Required Performance Level:

Severity of injury (S)

S1 Slight (normally reversible injury)

✓ S2 Serious (normally irreversible injury or death)

Frequency and/or exposure times to hazard (F)

F1 Seldom to less often and/or exposure time is short

✓ F2 Frequent to continuous and/or exposure time is long

Possibility of avoiding hazard or limiting harm (P)

✓ P1 Possible under specific conditions

P2 Scarcely possible

Figure 20 Performance level calculator within SISTEMA.

ISO 13855-2010 specifies safety distances for body parts approaching a hazardous zone. The ISO standard is comprehensive, with many conditions for each case depending on the hazard location, detection capabilities, and angle of approach.

Standards regarding robot safety can be categorized into three types. The type A standard ISO 12100 contains principle strategies for machine safety. The type B standard ISO 13849-1 explained above specifies guidelines for designing the safety of control systems. The type C standard ISO 10218-1/-2 provides guidelines for the safety of industrial robots. Collaborative robots are addressed in the complementary ISO 15066:2016, containing requirements for hand guiding, speed monitoring, and force limiting.

2.3 Digital Factory

2.3.1 Simulations and Emulations

Even though a simulation model and an emulation model in many ways may look identical, their structure and purposes differ. A simulation is a system that behaves similarly to the physical entity being simulated but is implemented entirely differently. It can give a basic understanding of the system's behavior, functionality, and limits in a flexible and explorational manner. On the contrary, an emulation is a system that behaves exactly as the physical entity and is consequently more constrained and rigorously defined, allowing for credible validation of how the system would perform in the physical application.

Depending on the intention, some parts may be emulated while others are simulated. The robot and PLC programs would be useful to emulate together to verify they behave the same in the virtual environment as in the physical environment. Visuals and non-important objects might not need to be emulated, such as a forklift delivering material, and can therefore be simulated instead (McGregor, 2002).

2.3.2 Virtual and Augmented reality

There are many definitions of Virtual Reality. The definition by Sherman and Craig is prominent within the field. It states that: “*Virtual Reality is a medium composed of interactive computer simulations that sense the participant’s position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation*” (Sherman & Craig, 2018, p. 16). The immersion is generally achieved by using VR headsets, a head-worn device with two screens giving the perception of a 3D world. The headset is tracked using

various sensors either mounted on the headset or in the room, letting the user move around in the environment realistically (Stark, 2022d).

While there is a trend of increasing the availability of VR within companies, its use is still considered low (Stark, 2022d). The most common use case in the industry is evaluating the physical ergonomics such as line of sight, posture, and reachability. It can also work as a way to communicate ideas and convey design proposals without expensive physical prototypes (Berg & Vance, 2017).

Augmented reality is a mixed representation where the physical world and virtual objects are concatenated, typically by overlaying virtual objects on a transparent screen between the user and the physical world. In contrast to virtual reality, the AR-user does not lose awareness of the surroundings, which could be dangerous in a factory setting (Carmigniani et al., 2011).

Typically the augmented data is visualized on a head-mounted see-through display. The user's position and orientation can be determined using optical sensors, gyroscopes, and other tracking technologies. Physical objects are often tracked optically using geometric feature extraction and edge detection (Stark, 2022e).

The industry's interest in augmented reality is increasing overall, but the technology is especially relevant for maintenance and assembly, where tasks vary significantly. Typically observed benefits of using augmented reality in the industry are reduced time spent on tasks and fewer user errors. Other uses are remote support, inspection, and training (Bottani & Vignali, 2019).

2.3.3 Digital twin

The Digital Twin was conceptualized by Grieves during a PLM presentation at the University of Michigan in 2003. In the years after, the concept had many different names until it was finally referred to as a Digital Twin,

coined by NASA employee John Vickers using it for troubleshooting spacecraft (Grieves, 2019).

Since 2021 the term has received an ISO definition which states that a digital twin is a; “*Fit for purpose digital representation of an observable manufacturing element with synchronization between the element and its digital representation.*” The elements are not limited to the equipment and material but include other manufacturing elements such as personnel, facility, environment, and supporting documents (“ISO 23247-1:2021(en),” 2021). Five additional similar standards are currently being developed. Common to the definitions is that they explicitly state that there must be a physical twin for a digital twin to exist. Otherwise, the digital twin would be no more than a digital model (Johnsson, 2021).

The Digital Twin works by capturing real-time data from the physical asset, after which forecast and proactive optimization can be done, after which data can be sent back to the physical twin to act on the results. The algorithms used by the Digital Twin differ between application domains and should be purposely selected for the task at hand (Negri et al., 2021). Production planning, maintenance, and layout planning have been recognized as suitable areas to use a Digital Twin. Examples of this could be replanning and diverting orders during standstill or making statistical analyses of sensor data to predict wear and schedule preventive maintenance (Kritzinger et al., 2018).

In a meta-study by Liu et al. (2021), it was concluded that out of 147 academic publications claiming a digital twin in their papers, only 40% (60) were actual digital twins, with the rest of them being either digital shadows or digital models. With the term becoming an over-used marketing buzzword in the industry, concerns have been raised that the term will lose its actual usefulness (Wright & Davidson, 2020).

2.4 Virtual commissioning

Testing and verification on real plants with real controllers is expensive and time-consuming. The concept of virtual commissioning tries to solve this issue by detecting errors and weaknesses at an early stage without having to disrupt production or unnecessarily occupy resources (Lee & Park, 2014). Field research has shown that the method can reduce the time required for commissioning and increase the quality of the delivery (Zäh et al., 2006).

2.4.1 Level of detail

The level of detail should match the expectations of the model. When simulating on the plant level, metrics of interest could be OEE, inventory, throughput, and material flow. Having detailed process models of each actuator or sensor in such a simulation would be heavy, complex, and unnecessary for the task, i.e., high effort. In the same way, having a too simple model when looking at the machine level might overlook essential features such as failure behavior or kinematic movement, i.e., low benefit (Puntel-Schmidt & Fay, 2015). The dilemma occurs when the system size is on the plan level, but some machines have to be tested closer. Zaeh et al. (2003) proposed a solution to the dilemma through a dynamic level of detail using a *magnifying glass* where certain parts of the processes can be examined in a higher level of detail.

2.4.2 Configurations

By combining the physical plant and controller with their virtual counterparts, four different configurations become possible, each with specific advantages and application areas. The configurations shown in Figure 21 are generally referred to as real commissioning, reality-in-the-loop, hardware-in-the-loop, and software-in-the-loop and are further described below (Auinger et al., 1999).

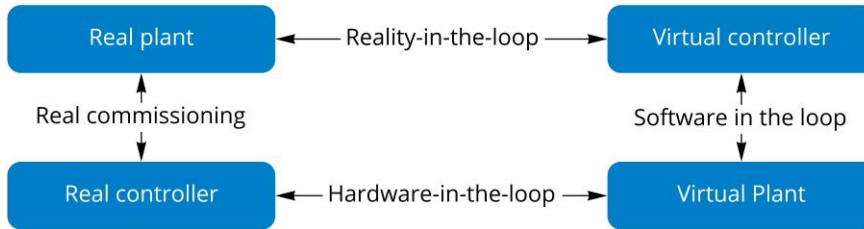


Figure 21 Commissioning configurations, inspired by (Auinger et al., 1999).

Real commissioning

Real commissioning has historically been the most used commissioning configuration involving the real plant and real controller. All testing has to be done on-site, potentially disrupting the production. The number of test cases will be fewer as it takes time to restore the cell, and some would not be tested due to the risk of damaging equipment. As this is the first time all components are tested together, errors introduced by the programming may be found very late, making real commissioning time-consuming and expensive (Schludermann et al., 2000).

Hardware-in-the-loop

Hardware-in-the-loop requires a physical controller, usually a PLC or DCS, that can communicate with the simulated plant. The advantage of this approach is that all code can be written and verified without the need for the actual cell. Having a physical controller ensures real-time, deterministic responses with accurate program execution (Rankin & Jiang, 2011).

Software-in-the-loop

Software-in-the-loop is the configuration used when performing virtual commissioning. In this variant, the PLC is emulated with the benefit that no hardware is needed, which reduces capital tied up without being used. Program bugs and errors can be detected earlier and give a more realistic prognosis of when the unit can be realized. New products or processes can be introduced without production disruption, and cycle times can be measured and optimized beforehand (Rahman & Mohamad, 2016).

Reality-in-the-loop

Reality-in-the-loop utilizes physical machines and equipment with a simulated control system and can be used to simulate and debug control code not yet implemented. Reality-in-the-loop can be used when it is valuable to validate the control code for a swift replacement in processes where equipment is frequently replaced (Auinger et al., 1999; Dahl et al., 2016).

2.4.3 FAT, vFAT and SAT

The commissioning phase usually ends with a Factory Acceptance Test (FAT). The FAT is conducted to ensure that all software and hardware meet the requirements as defined in the safety specification. While not mandatory to meet standard safety specifications, it is highly recommended (Commission, 2003). The scope of a FAT varies between companies and are usually have tailored their needs. Generally, the testing protocol should be comprehensive and verify all safety functions, fault handling, and exceptional cases (Smith & Simpson, 2020).

The virtual factory acceptance test (vFAT) is performed using the FAT protocol on the virtual commissioning model. In addition, the vFAT can include tests that would be too dangerous, damaging, or impractical to perform in a physical cell. Examples of this could be tool detachments and collisions (Ayani et al., 2018). As a reaction to the Covid-19 pandemic, the term has occasionally been used to describe a physical FAT done via videoconferencing (Peiris & De Silva, 2021).

A site acceptance test (SAT) is similar to the FAT except that it is performed at the customer's site after delivery, installation, and configuration. The Standard Operating Procedures, certificates, or other documents related to the equipment are often provided to the product owner after the SAT, marking the project's end (Chow & Walker, 2019).

2. Theoretical background

The physical FAT and SAT are not part of the virtual commissioning but can gauge how well the virtual commissioning was executed. If the virtual commissioning was successful, there should be few issues during the FAT. Even in cases where virtual commissioning does not reduce the time spent on commissioning, the time spent can be shifted from on-site to off-site, effectively reducing the total cost nevertheless (Lilja & Magnusson, 2021).

3. Literature review and industry research

This chapter consists of three parts: a literature review that considers theoretical standpoints, company interviews to capture the industry's practice and visions, and some software that can be used for Virtual Commissioning.

3.1 Literature review

Virtual commissioning is a well-defined term, practically exclusively used in the tech industry. For the literature review, the databases Google Scholar and Scopus were searched for thesis projects, scientific articles, and conference papers with this term. Additional keywords were used to narrow the search results further, such as *manufacturing*, *robot*, and *production*. The search resulted in 328 results on Google Scholar and 122 results on Scopus. Fifty papers from the databases were selected based on their titles, publication date, and relevance. After reading the abstracts of the selected papers, thirteen were chosen to be part of this review.

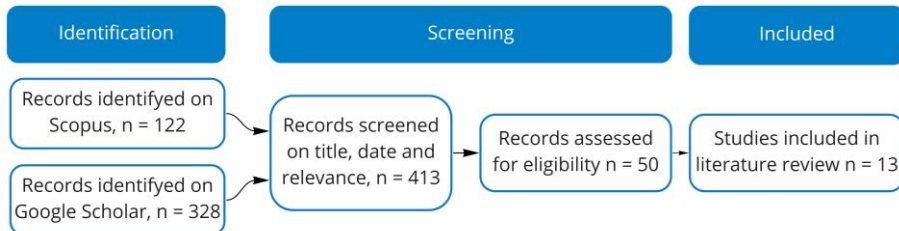


Figure 22 Flowchart of literature review according to the PRISMA model.

The research questions to be answered are:

- RQ1 What are the benefits of using virtual commissioning
- RQ2 Which are the limitations of virtual commissioning
- RQ3 What are the prerequisites of virtual commissioning
- RQ4 What is the current state of the technology

3. Literature review and industry research

Table 1

Selection of literature review papers

	Name	Year	Author	RQ
Master Theses	A deep dive into the current and future conditions for virtual commissioning of production lines.	2021	Anton Lidell	1,3
	Investigation on Industrial Practice of Virtual Commissioning	2020	Alkureidi	4
	Seamless integration of augmented reality into digital workflow of virtual commissioning	2019	O. Havlíček	1,2
	Virtual production line – Virtual commissioning	2018	Persson & Norrman	1,2
Research papers	A virtual commissioning based methodology to integrate digital twins into manufacturing systems	2021	Barbieri et al.	1
	Human-in-the-loop simulation for virtual commissioning of HRC	2020	Rückert et al.	1
	Automotive Paint Shop 4.0	2020	Bysko et al.	1
	A standardization approach to Virtual Commissioning strategies in complex production environments	2020	Albo & Falkman	2,3
	Virtual Commissioning – Scientific review and exploratory use cases in advanced production systems	2019	Lechler et al.	4
	Impact of a virtual twin on commissioning time and quality.	2018	Schamp et al.	1,3
	From virtual commissioning to simulation supported engineering	2014	Oppelt & Urbas	1,3
	Economic application of virtual commissioning to mechatronic production systems	2007	Reinhart & Wünsch.	3
	Nutzen der virtuellen Inbetriebnahme: Ein Experiment	2006	Zäh et al.	1,3

A deep dive into the current and future conditions for virtual commissioning of production lines (Lidell, 2021)

The study aimed to identify critical conditions to virtually commission a production system and compare the identified methods. Furthermore, interviews were carried out to assess the use of standardized methods for virtual commissioning. While Lidell did not mention the study of Digital Twins as an objective, a substantial part of the thesis addressed this topic, probably due to the author's view of the digital twin as an integral part of virtual commissioning. The terminological confusion became apparent through both the interviews with industry representatives and the literature review, with Digital models and Digital shadows used synonymously with Digital Twins (see Chapter 2.3.3). To mitigate this discord, Lidell states the importance of a standard and common definition to avoid further confusion and mentions that ISO 23247-1 is underway (and was published some months after Lidell's thesis was published).

The processes where VC was deemed nonbeneficial were more complex processes such as spray painting or welding. The parameters for these processes would need to be tweaked during the physical commissioning regardless and would often be economically unfeasible to simulate for every commissioning project.

Investigations on Industrial Practice of Virtual Commissioning (Alkureidi, 2020)

This meta-study screened 500 records of Virtual commissioning and selected 50 papers publicized between 2014 and 2020. It was found that 77% of the studies were within automation and robotics, but only 17% in manufacturing and process. This result was attributed to the considerable processing power needed to simulate processes and that VC can contribute more on the automation level.

3. Literature review and industry research

Apart from the literature study, a smaller survey was conducted in which 30 companies responded. The companies represented ten different industry sectors, but the authors did not state how the companies were inquired. The study found that 4% of the companies had a virtual commissioning strategy implemented, 18% were currently implementing the strategy, 9% had formulated a strategy, and 29% were developing a strategy. Notably, only 20% of the respondents said they were not invested in virtual commissioning. The questionnaire also covered data-related questions from which it was concluded that 30% did not collect process data during production, while 53% collected some data and 17% collected all data.

The companies' estimates of virtual commissioning requirements in different fields can be seen in Figure 23 and show a general lack of data security and IT infrastructure competence.

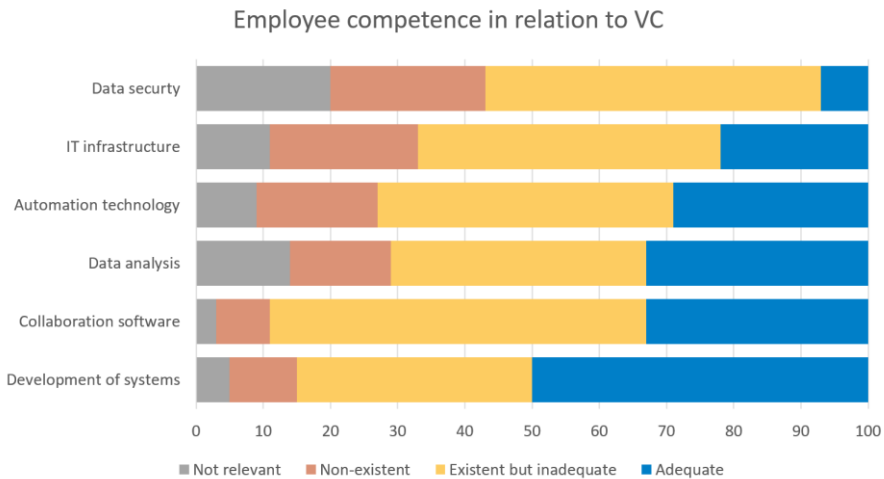


Figure 23 Companies' estimates of employee competence concerning VC. Data aggregated from (Alkureidi, 2020).

Seamless integration of augmented reality into the digital workflow of virtual commissioning (Havlíček, 2019)

The aim was to use Augmented Reality to let the commissioning engineer visualize, move and resize work zones defined by the automation engineer in Siemens' automation portal TIA. Any changes using the Hololense were to be synchronized to the portal software and vice versa.

Havlíček found that there is currently no appropriate framework or workflow for working with augmented reality; seven different software had to be used to achieve the specified functionality. Because of this, the setup time was extensive and complex, lowering the benefits of using the application. Additionally, the operators were unaccustomed to new technology, further lowering the benefits in the short term.

While Havlíček did succeed in making the application work, the lacking software support and the many software introduced many issues such as gimbal lock⁸ and unrecoverable crashes.

Virtual production line – Virtual commissioning (Persson & Norrman, 2018)

This thesis was the first to investigate virtual commissioning at Alfa Laval. The aim was to propose areas where virtual copies would be beneficial to aid product implementation and optimization. Several production lines were evaluated through benefit effort analyses graded by chance of success, value to the company, and potential for further work. Model accessibility was a determining factor as well.

⁸ Gimbal lock occurs when two axes in a three-axis gimbal become parallel, effectively losing one degree of freedom, resulting in erratic behavior.

After selecting the most promising production line, the virtual environment was created using RobotStudio to facilitate the virtual commissioning. The virtual commissioning was highly limited due to multiple factors; the PLC program was created in CoDeSys, but no PLC code from the actual application was tested; the signal handling was primitive, and only boolean values were used in an amount far less than in the physical cell; the actual HMI could not be tested as it was not finished; the commissioning assumed perfect behavior with no faults or errors.

Persson and Norrman found the maturity of virtual commissioning to be low but recognized virtual commissioning as a standard work procedure in the future. Due to the large number of software needed, the learning curve to gain sufficient competence for a VC was considered steep. The authors' vision is that the VC models could be collected in a database to build complete digital factories in 3D. Moreover, they saw the benefits of using the virtual copy as a learning tool after getting positive feedback from the operators after showing the HMI and simulation.

Persson and Norrman recommended against creating virtual environments for existing production units due to the amount of effort of getting drawings, models, and documentation and creating virtual models from these assets.

A virtual commissioning based methodology to integrate digital twins into manufacturing systems (Barbieri et al., 2021)

This study proposes a methodology for implementing and testing digital twin architecture in a virtual environment before the physical commissioning. The authors place virtual commissioning in the pre-digital twin phase and explore how to use VC to interface and synchronize architecture⁹ for

⁹ Note the distinction, no Digital Twin is present as the system has not been physically commissioned yet.

DT, MES, and PLC to verify its functionality before the physical implementation. In that sense, the emulation is used to virtually commission a digital twin by ensuring every layer communicates and processes the data as specified.

A case study was performed to apply the proposed methodology. A production sequence was set up in Excel, acting as MES. CoDeSys acted as PLC, and the production process was set up in Experior. The machines were modeled with breakdown and repair functions, and the goal was for the digital twin to reschedule production automatically and efficiently in case of a breakdown. This was achieved by using a genetic algorithm that generated and tested many sequences to find the minimum makespan.¹⁰

Human-in-the-loop simulation for virtual commissioning of human-robot-collaboration (Rueckert et al., 2020)

With the increase of collaborative robots in the industry, Rueckert et al. investigated a virtual commissioning configuration named Human-in-the-loop (HITL). The method sought to extend the capabilities of virtual commissioning by being able to validate additional requirements of the robotics safety standard ISO/TS 15066. This was done by mapping the operator's behavior to discover hazards arising from unexpected behavior or faults. The disadvantages of the method were that the simulation must be able to run in real-time and that it takes a detailed model to make sure the user is immersed enough to behave naturally. Rueckert et al. conclude that 51% of the requirements within the standard can be checked using virtual commissioning, and an additional 11% of the requirements can be checked by extending the virtual commissioning with a human in the loop.

¹⁰ A video of the process can be found at youtu.be/kRcVmvT-NI8

Automotive Paint Shop 4.0 (Bysko et al., 2020)

The paper aimed to research buffer control systems and sequencing algorithms for a car paint shop where manual flow control was used. As the Car Sequencing Problem had previously only been studied theoretically, the company wanted to use the virtual engineering and commissioning techniques to apply the theory in practice without intervening in the process, causing unnecessary downtime (Figure 24).

A virtual representation of the paint shop was created together with PLC code, after which virtual commissioning was used to evaluate different car buffer configurations to evaluate which one was to be commissioned in the physical production unit. Hardware-in-the-loop was used for this.

Three different sequencing algorithms were programmed and compared to no buffer and manual sequencing. Automatic sequencing was found to be 29% more effective than with no buffer and 22% more effective than manual sequencing. More impressively, the number of changeovers was reduced by 85%, decreasing waste of paint, cleaning agents, and set-up time. Thanks to using hardware-in-the-loop, the new optimized code could directly be transferred to production.

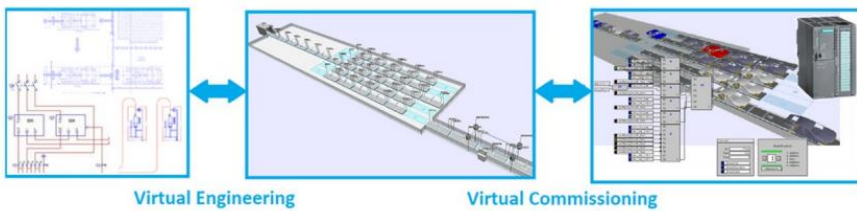


Figure 24 Virtual commissioning environment by Bysko et al. (2020) Image reproduced with permission from Sara Bysko.

A standardization approach to Virtual Commissioning strategies in complex production environments (Albo & Falkman, 2020)

This paper classifies different levels of virtual commissioning regarding size, functionality, complexity, and dynamics. Five levels are distinguished, where one or more classes of functionality further describe each level. The first level is a simple emulation of a control system for logic verification. In the second level, communication protocols and signals give additional capabilities to connect to other devices. Found in the third detail level are these devices that could be sensors, motors, or actuators. These devices are brought to life in the fourth level by adding visualization, kinematics, and dynamic response. Lastly, the fifth level contains higher-order control such as production control connected through factory-network or supervisory functions like SCADA or Historian. Standardization was acknowledged as a critical factor due to the number of subsystems that have to work and communicate together.

Furthermore, the authors explore the economic contrast between greenfield and brownfield within Virtual commissioning and present a weight matrix to aid the decision. It was noted that a high level of virtual commissioning would be more beneficial in larger greenfield cases thanks to its longer lifespan and might become too expensive for minor upgrades in a brownfield case.

Virtual Commissioning – Scientific review and exploratory use cases in advanced production systems (Lechler et al., 2019)

Lechler et al. presented an overview of the latest research on Virtual Commissioning and categorized it on the abstraction levels of automation, robotics, process, and energy. The research was further sorted by the field of

applications logistics, assembly, manufacturing, and testbed. Most research targeted automation in logistics, assembly, or testbeds. Most of the robotics research focused on the assembly process. They found limited research on both the process and energy abstraction levels, an absence attributed to limited real-time capability and processing power. Lechler et al. call for manufacturers to provide better virtual models of their products with more than purely geometric data to aid the creation of the Virtual Commissioning environments making it an established practice.

Impact of a virtual twin on commissioning time and quality.
(Schamp et al., 2018)

The study aimed to investigate the effect of virtual commissioning compared with traditional commissioning. An experiment was conducted where a small conveyor system was to be commissioned. In the experiment, 20 students were divided into two groups; The test group used virtual commissioning, and the reference group used traditional commissioning. The experiment showed a statistically significant positive effect of virtual commissioning on debugging time and requirements fulfillment. While the reference group used an average of 160 minutes for debugging and scored 61 in quality, the group using virtual commissioning used an average of 43 minutes and scored 89 in quality. The authors recognized the limited number of participants and proposed further research to substantiate the result. It was further noted that modeling time was not considered during the test and that having virtual models is “an important requirement for the industrial application of virtual commissioning”.

Integrated Virtual Commissioning an essential Activity in the automation Engineering Process- From virtual commissioning to simulation supported engineering (Oppelt & Urbas, 2014)

Oppelt and Urbas suggest an integrated approach to Virtual Commissioning where the models are used throughout the entire project and not only for the last test step. They emphasize parallelizing the tasks and using the hardware configurations and signal definitions created early in the automation process. By testing a developed function immediately, errors will be found as early as possible.

Several requirements are listed to enable the integrated approach. The simulation tool has to be scalable and support both HiL and SiL configurations. There have to be templates and interfaces set up for the efficient design of the simulation environment, and there has to be a management system for documentation and testing configurations.

The authors anticipate that integrated virtual commissioning will lower the threshold for the industry to adopt virtual commissioning as standard practice because it can be done without large changes to the traditional engineering workflow. Instead of having virtual commissioning as a final step before the physical commissioning, the models grow during the engineering phase and can be used as continuous support.

Economic application of virtual commissioning to mechatronic production systems (Reinhart & Wunsch, 2007)

Virtual commissioning suffers from what Reinhart and Wunsch describe as the *Efficiency Measuring Problem*. A project team will either choose to do virtual commissioning or not, and there are seldom benefits to doing both. Therefore, proper A/B-testing of virtual commissioning is practically only feasible in an academic setting. Because of this, Reinhart and Wunsch propose a four-step evaluation to find if a specific virtual commissioning project is economically viable.

First, the production system is decomposed into subsystems. Closed-loop material flows should be in the same subsystem; this also applies to closely related hardware, such as a PLC and robot controller in the same robot cell. This set of subsystems then constitutes potential projects for Virtual commissioning. In the second step, the subsystems are quantified in effort and benefit. Reinhart and Wunsch define effort as estimating the subsystems' mechanical complexity and connections to other subsystems. The mechanical complexity is estimated by counting I/O-signals, flow connections, and other internal processes. Outer complexity is determined through the amount of communication and material flow. The benefit is calculated by the contribution of control to system functionality, the size of the subsystem and the importance of the subsystem. The third step combines the aforementioned metrics by calculating the products to form an effort-benefit diagram, making the projects comparable. During the last step, previous deliveries are reviewed to identify the appropriate virtual commissioning strategy. Time, cost, and quality are three problem areas mentioned that a strategy could specifically target.

Nutzen der virtuellen Inbetriebnahme: Ein Experiment (Zäh et al., 2006)

This experiment targeted the previously mentioned *Efficiency Measuring Problem* by performing an A/B-test on a tin-can press cell. The cell was specifically designed to compare virtual and physical commissioning, containing typical industry units such as PLC I/O-modules, sensors, actuators, and motors. The virtual simulation environment was prepared beforehand as an exact image of the physical press cell; hence only the control function programming was compared in the experiment. Quality was determined by accessing isolated parts of the code for different functions within the cell. Two groups of 30 programmers were assigned to do physical or virtual commissioning.

The commissioning time within the group using virtual commissioning was 75% lower than the time spent by the group with no virtual tools, albeit the reduction in spent time in total was less notable at 15%. Looking at the quality of the code, the group using virtual commissioning scored 84 out of 100, while the other group only scored 37. Additionally, a waste reduction was observed in the group using virtual commissioning, as faults in the code would not damage machinery, equipment or parts.

The authors propose working with the code in parallel with mechanical and electrical designing, and once the assembly phase starts, virtual commissioning can begin as few changes are made after this point.

Mentioned prerequisites for virtual commissioning are readily available data and project managers that know how to integrate virtual commissioning within their project team optimally. The authors suggest standardizing frequently used parts for reuse to speed up future projects by eliminating cumbersome manual modeling.

3.2 The industry speaks

This section presents the results of interviews with several industrial companies. Complete transcripts can be found in Appendix E.

Theoretical methods have little value to the industry if they cannot be implemented practically. Therefore, it is relevant to understand the progress and possibilities of virtual commissioning from the industry's vantage point. As mentioned in Chapter 1.8, interviews are a suitable form for quick and broad information gathering while still being able to ask follow-up questions.

The company representative was asked the same questions, but the discussion was held open. The interviews aimed to investigate current obstacles and future visions of virtual commissioning. The selection was made by searching for companies that have mentioned virtual commissioning.

The questions asked to all company representatives are as follows:

- In what way does your company use virtual models and virtual commissioning currently? (RQ1, RQ4)
- How has the usage of virtual models and virtual commissioning been historically? (RQ1, RQ2, RQ3)
- What are your immediate aim regarding utilizing virtual models and virtual commissioning? (RQ1, RQ2, RQ4)
- What is your long-term vision regarding utilizing virtual models and virtual commissioning? (RQ1, RQ2)
- What is, in your opinion, the largest obstacle to overcome in order to reach your vision? (RQ2, RQ3)

TechTribe

TechTribe is an integrator that offers complete industrial automation solutions. Their virtual simulation and commissioning foundation is built upon Emulate3D from Rockwell (see Chapter 2.4.4). As TechTribe has noticed a reduction in commissioning time by using virtual models, they want to find a way of incorporating the methods into their existing project model. In their visionary project model (Figure 25), they strive to include the virtual model at an earlier stage, preferably already when the solution is presented, and then to be able to reuse the models later.



Figure 25 Visionary model of the digital process. Image reproduced with permission.

The anticipation is that this will lead to an increased number of virtual commissioning projects that can be used for marketing. A major challenge for TechTribe is that these services are not yet in demand, and it is a demand that TechTribe and other companies will have to create themselves. In the long term, TechTribe hopes the customers themselves will ask for these services in future projects.

Another difficulty is to find an appropriate detail level of the virtual model so that unnecessary features are not included wasting time whilst still not compromising the validity of the commissioning. TechTribe considers it problematic that changes in their CAD models do not carry over to Emulate3D, resulting in extra work.

FlexLink

FlexLink is a conveyor manufacturer specializing in automated and flexible conveyors. Historically they have been product-oriented with few services. Since 2014 Flexlink has had its own desktop application called Flexlink Design Tool, which is based on Visual Components (see Chapter 2.4.4). Currently, it is used mainly by salespeople to prepare quotes for the customer, but they want to make it more publicly available. Using the design tools speeds up sales, provides visual substance, and can quickly generate a Bill of Materials. The reception has been overwhelmingly positive of Flexlink's addition of more virtual models to their sales procedures. However, they want to find ways to easier scale the use of the models rather than doing isolated projects with single customers.

They want to extend the use of their virtual models, layouts, and simulations to enable performance and asset monitoring and make it more accessible directly to the customer. FlexLink acknowledges that its digital readiness has generally been immature and that it is a long road ahead until the complete solutions are in place, especially since they see similar unreadiness when visiting customers. To become more digital, they first need an infrastructure to gather and analyze data.

They have experienced a market drift towards services rather than products, and virtual commissioning would be one of the services that FlexLink may deliver in the future. The customer would not necessarily be paying for this service; it may also be integrated into the standard operating procedure.

When asked whether these services align with FlexLink's otherwise niche business strategy surrounding conveyors, they restate that the decreasing demand for products forces them to rethink their strategies to remain relevant in the future.

Conflicting FlexLink's short-term visions are organizational issues where it is hard to translate needs into a job role that fits the current role and pay structure, especially since some of these profiles require a significantly higher entry-level salary. A new team called Digital Transformation has been put in place to educate and get the organization on board.

AFRY

AFRY (former ÅF) is one of Europe's largest engineering, design, and advisory consultancy firms with 17000 employees divided into six divisions. They have been looking into virtual commissioning since 2017 when a master thesis was conducted on the subject. In the thesis, *Virtual commissioning of smart factory* (Älegård, 2017), an interview with the company was conducted. Virtual commissioning was not used within AFRY at the time, but it was estimated to be standard practice within ten years.

Virtual commissioning is part of the AFRY X division and their new Real Digital Twin concept. When asked about what AFRY means when they say Digital Twin, they mean emulated controllers in some industries and say that a city model could be an example of a digital twin.

Currently, the division is working on building a library of components of their hardware. They were previously more focused on virtual commissioning but now want a better framework to use digital twin afterward. The solutions are both sold as services to customers and used internally. Using SIMIT is preferred thanks to its extensive library; process industries generally do not need any other software than SIMIT.

While building their tools and components, they find it challenging to find information on specific hardware. In addition to this, they have no standard of signal names or how to structure a project to make sure everyone is on board which creates a divergence between projects.

Handovers between disciplines are a common difficulty. Electronics and mechanical designers hand over to the programmers and then to the virtual commissioning team. Standardization is needed that stipulates routines for documentation, namings, interfaces, and models to ensure the communication is seamless and efficient. Without a standard, they risk creating dependencies on specific people.

AFRY acknowledges the need for a change in working practices. Ideally, they would have scheduled testings biweekly with a predetermined focus on specific functions of the cell. This method seldom works due to the industry's unfamiliarity with this way of working, resulting in everything being stressfully tested later.

Evomatic

Evomatic is a Swedish automation integrator company targeting the manufacturing industry. They have used simulations in every project for over 15 years to communicate potential solutions to the customer and generate and validate robot programs. RobotStudio is always used regardless of which robot brand will be used in the final application due to its superior virtual environment and Evomatics programmers' proficiency with the software.

They have completed a few full virtual commissioning in different configurations, both with and without graphics. While they have not noticed any increase in customer demands, they have found the virtual simulation model an excellent tool to communicate concepts with the customer at the beginning of projects. Evomatic still sees great benefit even from only validating the robotics code.

One of their current challenges is to focus the efforts and parallelize the work in order to be able to do commissioning earlier in the process. A step in this direction has been taken where they reorganized the company into

project groups instead of function groups so that designers, PLC programmers, robot programmers, and project managers sit together instead of sitting in their respective offices. This change has facilitated cooperation and made the process more efficient. However, they believe that much more can be done in this area and that there is a great benefit in bringing forward commissioning.

Evomatic's view is that the time spent on the project is about the same, but the time has shifted from being in overtime out in production and instead of being in the office during regular working hours, which saves money. Evomatic sees significant safety benefits in performing parts of the commissioning virtually. Project delays are customary, resulting in a crowded and hazardous environment shared by electricians, mechanics, programmers, and moving equipment working to meet a rapidly approaching deadline. By letting electricians and mechanics work alone on the physical plant, lost time injuries can be reduced.

Väderstad

Väderstad is a family-owned business from Sweden producing agricultural machinery. All their parts are painted in their iconic red color, a task that they now use robots to perform. The painting paths are created by using VR in RobotStudio. It would arguably be a stretch to call this virtual commissioning as it is a simple path validation. However, the workflow shares many similarities with what would constitute virtual commissioning. They receive the part to be painted from the mechanical designer, and through Creo Flex simulates the part's positioning when hung due to gravity. The gravity-aligned part is exported to RobotStudio, where the painter teaches the robot by virtually manipulating the paint gun, as seen in Figure 26.





Figure 26 Using VR to program the paint program.

The production cells at Väderstad have primarily been built by third-party integrators; hence no virtual commissioning has been made on the cell level. Väderstad estimates that utilizing VR, simulation, and validation has halved their time on programming tasks. They have found no other software than RobotStudio to suit their needs during their investigations into simulation software. They have no concerns about being limited to ABB robots.

Scania

Scania is a Swedish commercial vehicle manufacturer with over 1500 dealers and workshops worldwide, delivering over 90000 trucks annually. They see a need to have up-to-date virtual models and scan the factory frequently to ensure the production is always moving. They have tried all four configurations of virtual commissioning; the less common Reality-in-the-loop has been used when migrating PLC programs to new PLC systems before delivery (see 2.4.2 Configurations). Scania uses multiple software for their virtual models. RobotStudio is used in their body-in-white factory, where they download robot code to refine it.

The main focus in the coming years will be to centralize virtual model data to one software, 3DEXperience. The robotics code, as well as all other virtual data, will be in 3DEXperience. A challenge is to connect the models from 3DEXperience with RobotStudio, as it is not an option to set up the

cell in RobotStudio in each project. With 50 product introductions yearly, Scania relies on real-time model data. Scania remarks on the importance of looking into the current purchasing routines and how they should be changed for the future as a decision to exclude virtual models from a contractor would incur a lifelong cost.

Volvo

Volvo began using virtual commissioning in 2016 after evaluating the method in a master thesis project. The PLC connectivity within RobotStudio was poor at the time, and they requested that ABB implements a better way to connect the Siemens PLC to RobotStudio as it would be too time-consuming to make the connections manually. A SIMIT shared memory connection was added shortly after. A principle decision was made that the ABB virtual controller should be used when appropriate due to its reliability, and all retrofit projects must have a simulation done before implementation.

Operators were delighted to be able to go inside the cell using VR and were more positive about the new machines. The vFAT took only half the time of their regular FAT, and they could test in much more detail. Furthermore, they could try products not yet manufactured. With the virtual models becoming even more accurate, Volvo is currently investigating if they can eliminate the physical testing altogether.

Elektroautomatik

Elektroautomatik is Sweden's largest automation integrator that provides turnkey automation systems (Elektroautomatik, 2022). They have been using virtual commissioning since 2018. They see virtual commissioning primary as a risk reducer and have found mechanical issues on multiple occasions during the virtual commissioning that could be fixed before incurring costs.

Elektroautomatik considers the increase in virtual commissioning indisputable, and therefore they focus on methods to create the virtual environment more efficiently. Altogether they estimate the decrease in commissioning time to be 40%.

They use the customer's regular FAT protocol for the virtual commissioning and take note of deviations. Typical deviations are functions that must be physically tested, such as servos and safety functions. Vision guiding is also generally not included in the vFAT as it gives no added benefit. After the virtual commissioning, a backup of the virtual environment is provided to the customer for future use, albeit it is not updated with modifications made during the physical FAT.

3.3 Virtual Commissioning Software

Software profiled towards virtual commissioning has increased in recent years. Just as with CAD software, the software is very similar to one another, and which to use largely depends on the liking and familiarity within the company. The base requirement for virtual commissioning is the capability to connect to control devices and simulate a virtual environment with movement and kinematics in real-time. While real-time is not always necessary, it can cause issues with the PLC execution. In this subchapter, a few of the most common software are presented.

RobotStudio

RobotStudio is ABB's offline programming, visualization and simulating tool. RobotStudio is limited to ABB robots, with the benefit that the virtual controller emulates the physical controller, meaning that all behavior will directly translate from the virtual world to the physical.

Most native CAD file formats are supported, requiring no extra conversion step to import models. The imported model can also retain a link to the CAD model. The model library is smaller than other software vendors,

containing only a few equipment and peripherals. Advanced logic can be created using *Smart Components* (see Chapter 4.5.1) or through self-made C# applications using the software development kit. There are two choices for PLC connectivity, OPC-UA, and SIMIT.

ABB has started incorporating transportation within RobotStudio with its ASTI Mobile Robotics Group acquisition in 2021, which might extend its use outside of the cell level.

Visual Components

Visual Components focus on manufacturing design and simulation technology. The company was founded in 1999 and later acquired by KUKA in 2017. Its latest version, Visual components 4.0, was released in 2016 as a significant remake that took advantage of the advance in computer hardware and 64-bit processors. Furthermore, an extensive API was created, allowing scripts and custom integrations to be made in Python. Visual components, hardware neutral even after the KUKA acquisition, hosts a vast robot library with over 1500 predefined models from over 30 brands.

Visual Components enables plug-and-play transportation controllers, including humans, automated guided vehicles (AGVs), and forklifts which are highly configurable, including speed, picking and dropping patterns, battery levels, and charging time. This capability can come in handy when planning new processes on a plant level where the material is transported between different cells. Data sampling from processes can be gathered into statistics reports to measure utilization, throughput, cycle times, and travel distances, to name a few.

Siemens S7 and TwinCAT PLCs are natively supported, but general support through OPC UA is available. Robot controller connectivity is limited, and thus, robot programs can be simulated but not emulated. Robot program postprocessors can be found in their forum; however, these are not officially released (*Visual Components*, 2022).

Emulate3D

Emulate3D is a simulation software acquired by Rockwell in 2019. It has the advantage that it can connect to Allen-Bradley PLCs natively and can perform the PLC emulation faster than in real-time. It features black-box components for parts not yet designed, experimentation of layout, statistics, and reporting. The robot movements are either run by pseudo-code or by visualizing the joint values from the robot software. The code cannot be used to move the actual robots. Emulate3D has a feature called Cad Is The Model, which can be used in Creo, Inventor, and Solidworks, making it possible to directly link and import models to utilize constraints and motions previously set in the CAD program. Conveyors and buttons can also be predefined within the CAD program. Emulate3D supports OPC and Mitsubishi, Siemens, Rockwell, and Beckhoff PLCs. Emulate 3D supports VR.



Figure 27 A conveyor plant within Emulate3D (Emulate3D, 2022).

Process Simulate

Process Simulate is simulation software that is part of Siemens PLM solutions. The software has modules for Spot Weld, Robotics, Commissioning, Human, and Assembly. The human module can analyze ergonomics and reach during operation. The Assembly module can optimize assembly sequences and make sure the procedure is free of collisions. It natively connects to Siemens' own PLCs, but other brands can be used through OPC-UA.

Simumatik

Simumatik is an online simulation platform. The company with the same name was founded in Skövde, Sweden, in 2018 and currently consists of ten employees. The development of the software itself began in 2010 when Mikel Ayani, the CEO of Simumatik, needed a means of simulation in his research at the University of Skövde. Nowadays, it is entirely cloud-based, bringing both positive aspects such as flexibility and accessibility, but with negative aspects such as needing internet connection and thereby induced latencies. The web-based solution also has issues with large complex geometries, which slows down the application. Simumatik uses cloud-computing credits and charges by the hour.

4. Viking - From idea to commissioning

In this chapter, the theory is applied in practice on an actual project at Alfa Laval to evaluate the practical prerequisites, benefits, and limitations of virtual models within virtual commissioning.

4.1 Current procedure at Alfa Laval

Project management

The project management methodology used at Alfa Laval is called PROMAL (Project Management Alfa Laval) and seeks to secure a project management culture by

- Improving and spreading the culture within the company
- Instituting smart, simple, and supportive processes and tools
- Developing and improving project management training
- Improving alignment of projects with strategy

This project management structure is mandatory for every project with a budget surpassing EUR 30000.

The PROMAL works with a stage and tollgate model presented in Figure 28. At every tollgate, a decision is taken to approve the initiation of the next phase, put the project on hold until tollgate criteria are met, or kill the project.

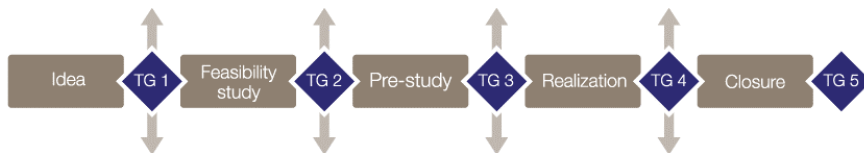


Figure 28 Different stages in Alfa Laval's Project Management Strategy PROMAL

The pre-study includes everything up to the point of realization. In a case of a mechanical project, this includes mechanical and electrical design. The realization begins with the order to manufacture and assemble the components. Both pre-study and realization might be outsourced to a third party, but some projects are done in-house.

In Figure 29, the PROMAL stages are mapped to a typical project timeline. The proposed approach symbolizes hypothetical time savings from performing the activities stated under each part of the cycle.

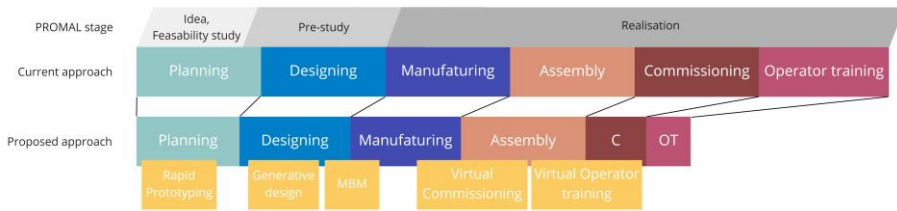


Figure 29 Hypothetical scenario of a manufacturing project lifecycle, before and after implementing the use of virtual models. Time savings are symbolic estimates.

Technology readiness levels

The method of Technology Readiness Levels (TRL) is used within Alfa Laval to systematically assess the maturity of particular technologies. The TRL method was first established by NASA in 1990 (Mankins, 1995). The development to reach TRL levels are made within Technology Development (TD) projects divided into Technology Investigation (level 1 to 2), Proof of Technology (level 2 to 4), and Proof of Concept (level 4 to 6). If the process already has a high TRL, the project is generally a pure investment project (see Figure 30).

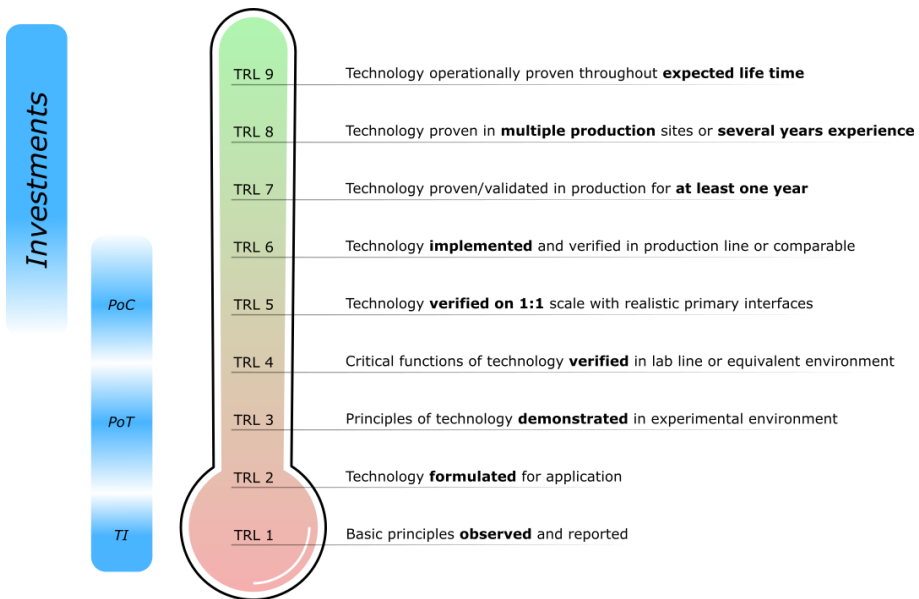


Figure 30 TRL levels used at Alfa Laval (Image reproduced with permission)

4.2 Project Viking

At the Alfa Laval site in Lund, gasketed plate heat exchangers are manufactured. The heat exchanger consists of a number of plates with elastomeric gaskets in between to route the fluid through alternate channels (see Figure 31). This enables heat transfer between two fluids while keeping them separated. Currently, the gaskets are mounted on the plates by hand, which is a monotonous task with repetitive movements taxing on the body in the long run. After a workshop session in 2019 where the plausibility of future projects was evaluated, a team of engineers began investigating automatic gasket mounting. A Technical Development project was initiated with the intention of bringing the TRL level of a gasket mounting method to TRL 3 through a Proof of Technology.

4. Viking – From idea to commissioning

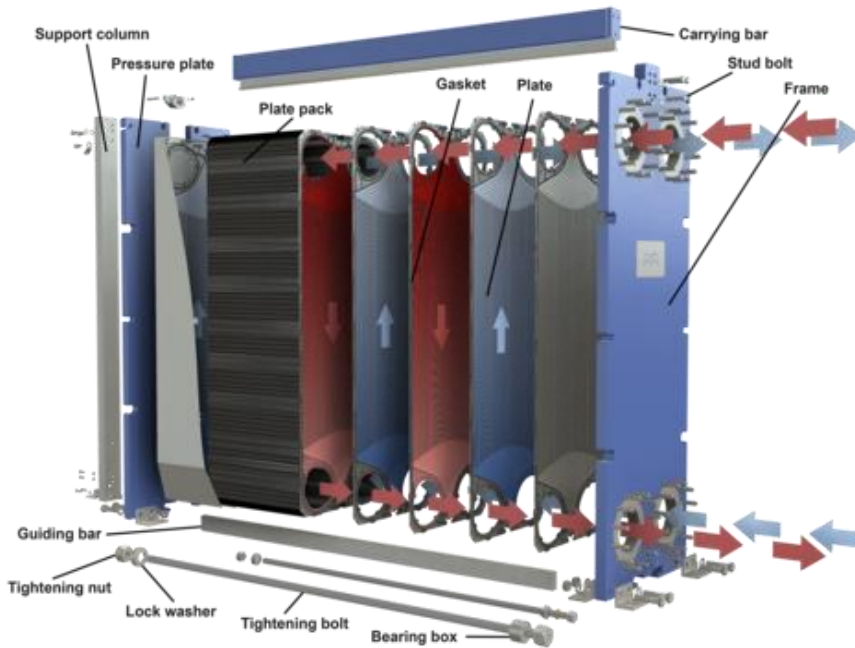


Figure 31 Part view of an Alfa Laval heat exchanger (Alfa Laval, 2022).

A successful proof of technology by Mälardalen Industrial Technology Center (MITC) was presented in 2020 (Figure 32), and it was decided to continue with the Proof of Concept project, which was named Viking.

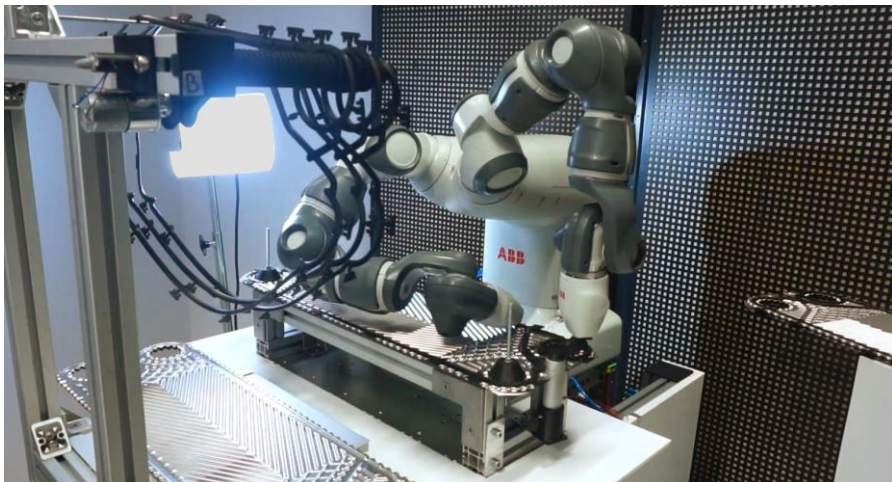


Figure 32 Proof of Technology by Mälardalen Industrial Technology Center.

Operation of the cell

The production cell shown in Figure 33 is described below to provide context and a sense of the project scope. Note that the layout in the figure has been iteratively refined over time by utilizing the virtual models (see Chapter 4.4). The operation of the cell can be described as follows:

- 1a. The gaskets are loaded in bulk by the operator.
- 1b. The plates are loaded in stacks by the operator.
- 2a. A gasket is separated from the bulk by robot 1.
- 2b. A plate is separated from the stack by robot 2.
- 3a. The gasket is picked and put on an alignment mechanism.
- 3b. The plate is picked and put on an alignment table.
4. The gasket is transferred to the plate held down on a table.
5. The plate is gasketed by robot 3 and 4 while held down on a table.
6. The gasketed plate is transferred to the exchanger or pallet.

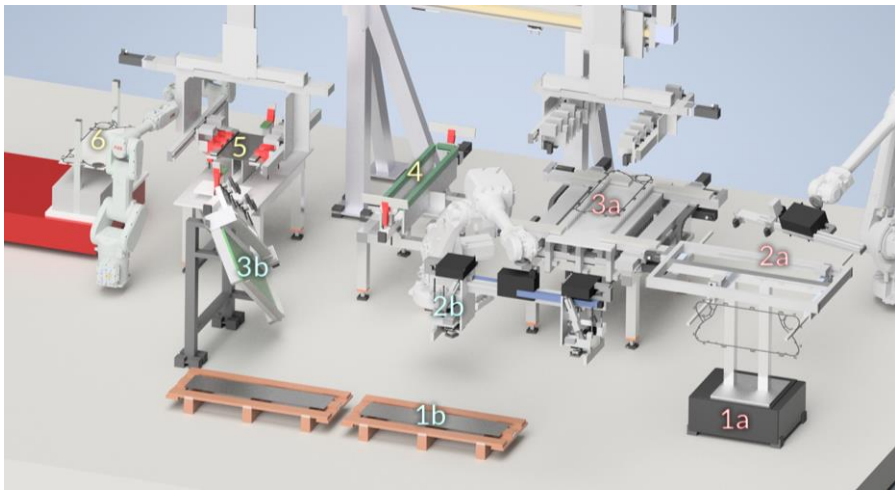


Figure 33 Overview of Viking, simplified due to confidentiality.

Software

The software needed for the virtual commissioning largely depends on which automation brands are used and what the company has competence within. For the Viking project, the following software will be used:

- Inventor – CAD data
- RobotStudio – Virtual commissioning and robot environment
- Studio 5000 – PLC programming environment
- FactoryTalk Echo – PLC emulator
- FactoryTalk Administration Console – Communication setup
- FactoryTalk Linx Gateway – OPC-UA server
- Aveva InTouch WindowMaker – HMI programming
- Aveva InTouch WindowViewer – HMI visualization
- Microsoft SQL Server Management Studio 18 – SQL database
- ArcestrA System Management Console – HMI-OPC configuration

RobotStudio was selected due to prior knowledge about the software, its powerful virtual controller, and its ability to do the robot programming and commissioning within the same software. Studio 5000 and FactoryTalk was selected as Rockwell PLCs are used within the company. Aveva was used as Alfa Laval had templates and know-how in the software.

4.3 Collecting and preparing the data

Building the cell from scratch can be a time-consuming task. In order to avoid unnecessary duplicate work, it is advantageous to use already existing CAD models and assemblies if present. While most simulation software can import STL files, much information is lost during the conversion. RobotStudio supports imports of native CAD formats, with the benefit that the assembly structure is intact, with parts grouped in the same way the mechanical designer grouped them. Moreover, the native format removes the conversion step. RobotStudio also allows a connection between the

simulation model and the CAD model, keeping it up to date on any changes in the latter. The mechanical design of the Viking project was created in Inventor and was directly imported into RobotStudio in the *iam* format.

Heavily detailed CAD models can often be unnecessary for simulation. Instead, they add more complexity and slow down the program. At the beginning of the project, the parts were imported with no simplification. With sufficiently fast hardware, this is no issue. However, it made file sharing hard and time-consuming as files could be larger than 1 gigabyte. Simplifications can be made to combat this issue. For the mechanical designer, it is vital that all details are correct and all parts are left intact as it provides the basis for bill of materials lists, ordering documentation, and drawings, to name a few. It is a balancing act to make sure the model is not too heavy to run smoothly but not too simple, so it loses its purpose. The data simplification also adds a step to the virtual commissioning and should be avoided if the hardware allows it. The data preparation is demonstrated on one of the tables of the cell (Figure 34), but the workflow is generally applicable.

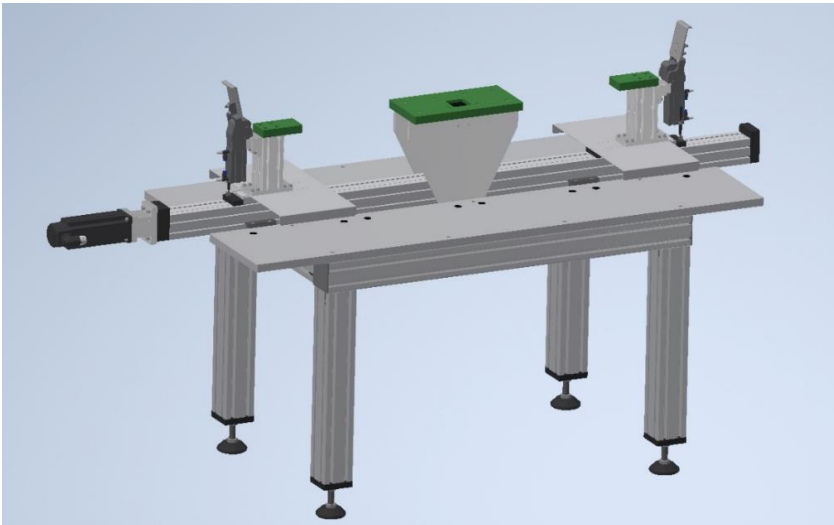


Figure 34 The table model used to display the CAD tools.

4. Viking – From idea to commissioning

In the common CAD software Inventor, the function *Simplify* can perform simplifications to the model by letting the user remove holes, fillets, chamfers, pockets, embosses, and tunnels. When selected, the affected features light up, after which the user can choose to preserve features of importance (Figure 35).

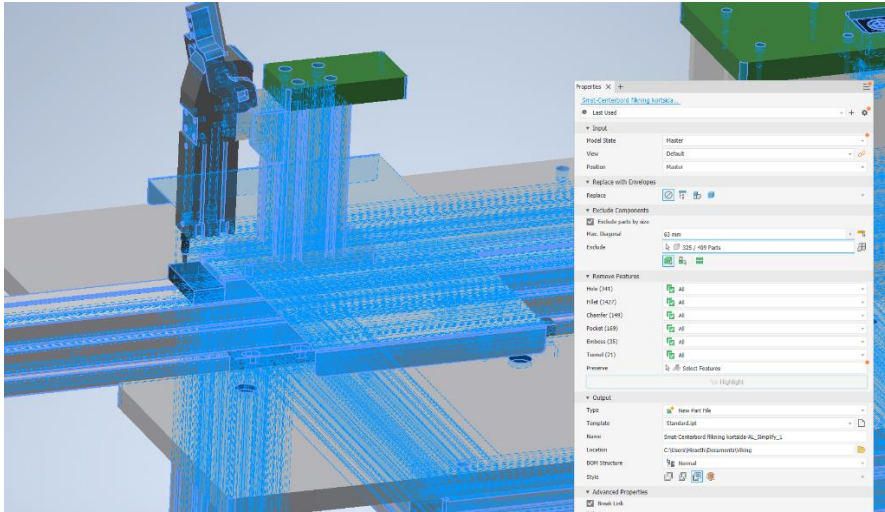


Figure 35 Holes, fillets, chamfers, pockets, and embosses highlighted for removal.

Components below a specific size can be removed to reduce the complexity further. As visualized in Figure 36, 20% of the table components were smaller than 60 mm and entirely unnecessary for the virtual commissioning. For extreme simplification, Inventor can replace parts with envelopes.



Figure 36 Components below 60 mm visualized and selected for removal.

In Figure 37, three tables with different detail levels are shown. The leftmost table has every part replaced by envelopes, bringing the file size down by 93% compared to the rightmost original. The middle table has features and small components removed, reducing the file size by 62%. Using the Statistics module in RobotStudio, it was noted that the gaskets accounted for 74% of the total cell size due to their many vertices; simplifying them made the total file size much smaller.

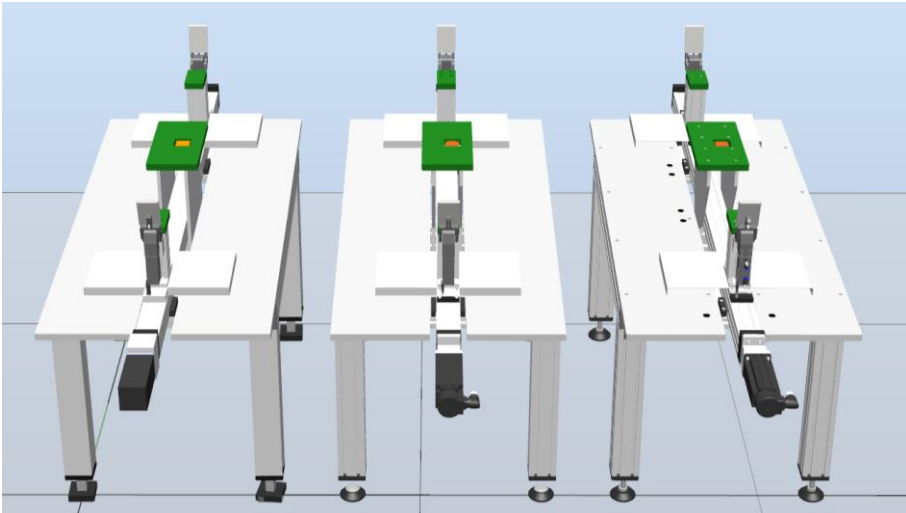


Figure 37 Different levels of detail. From left: envelope, feature removal, and original.

4.4 Virtual environment setup

Layout planning

With the data prepared, the CAD models were imported into RobotStudio. As the layout was not fully determined, RobotStudio aided in finding an optimal layout by quickly being able to move around objects and using robots to check reachability. The major layout revisions can be seen in Figure 38, where the cell got gradually more detailed. The appropriate robot models were selected and ordered after the layout was deemed satisfactory (see Chapter 4.8.3).

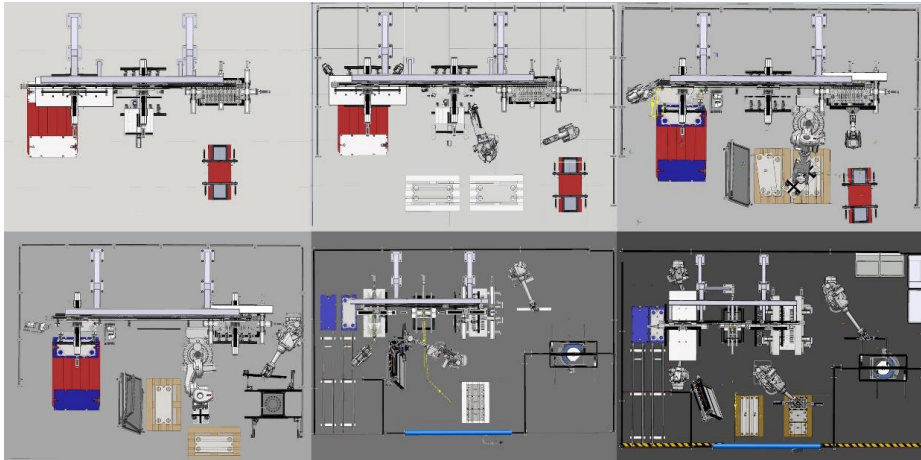


Figure 38 Major layout revisions, top-view in RobotStudio.

During the assembly of the physical cell, some changes had to be made to the positioning of particular objects. The new positions were quickly verified virtually before adjusting them on-site.

Mechanism creation

In preparation for the mechanism creation, all moving parts were separated from the static parts and parts moving together were grouped. RobotStudio has a *create mechanism* tool where links and joints between the links are specified. In Figure 39, the creation of the holding mechanism is shown, with the rotational joint axis represented as a green line.

The mechanisms were controlled using the smart components PoseMover or JointMover (see Chapter 4.5.1). In total, 40 mechanisms, with a total of 120 joints were created.

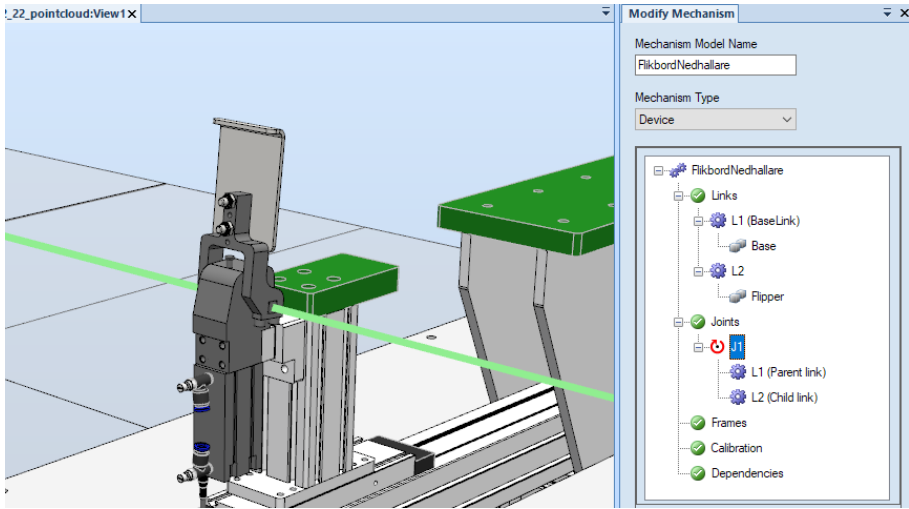


Figure 39 Example of mechanism creation within RobotStudio.

The previously selected robots were then imported and positioned, after which two virtual controllers were created to match the physical system, running two robots each. Options matching the ordered robots were selected to make the virtual installation identical to the physical installation (Figure 40). It would have been possible to run all robots on one controller as it has a limit of four robots, but a fifth robot was anticipated to be added in the future for other tasks.

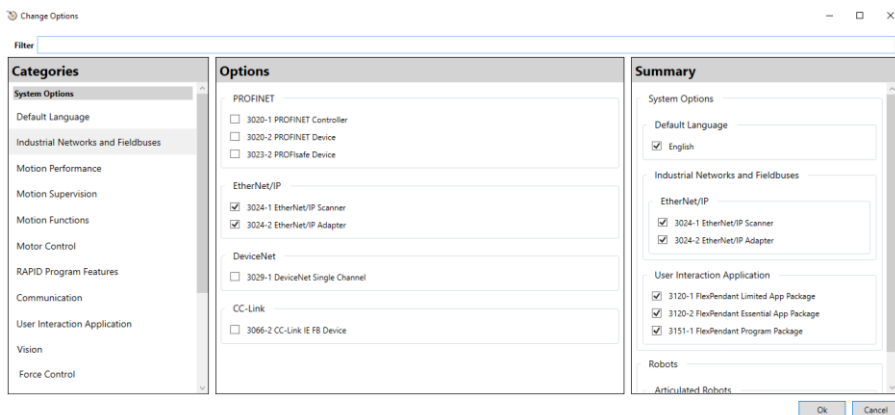


Figure 40 Options selection during the creation of virtual controller.

4.5 Programming

The focus will be on robotics as this is the part not provided by Alfa Laval. As mentioned earlier, ABB robots and software will be used, but the workflow is similar to other software. The PLC and HMI programming will be briefly described as they are programmed by other engineers.

4.5.1 Robot

Tool definition

It is important to have a correctly defined tool even in the virtual world, as it will affect movement, acceleration, and speed in the simulation. The Tool Center Point (TCP) is the point of the tool that will go to the programmed target coordinates. In this application, the TCP is between the teeth of the grippers for the gasket robots and in the middle of the suction cups on the plate handling robot. The gasket robots have one tool for gripping the gasket and one for gripping the plate. The TCP data can be acquired from the CAD or be physically measured (see Chapter 4.8.3). The IRC5 controller has a tool wizard that can define a tool using multiple positions and elongation points for greater calibration accuracy. Comparing the two methods showed that the CAD data is sufficient to make reliable programs, but defining the tool using the tool wizard should be done to account for deviations when manufacturing or assembling the tool.

The tool mass, the center of gravity (CoG), and moment of inertia were determined through the service routine *LoadIdentify*, available on the robot controller as a standard routine. During this routine, the robot moved the axes slightly while measuring changes in motor load. The difference was minimal when comparing the parameters identified by the routine with the CAD data. Both the load and CoG were off by only 2%, and the inertia was off by 7%.

Targets

Targets are defined by position, rotation, and robot configuration. The Alfa Laval robot standard stipulates that targets should be parametrized to accommodate different models set by the recipe. Instead of doing new targets for each new product, the program should adapt itself after specified dimensions using relative or offset movements.

Only ten targets are specified for the entire production unit, making the program easy to read and change. The robot moves relative to these targets depending on plate length, width, and thickness.

Interrupts and traps

An interrupt is a program-defined event that will trigger when the chosen interrupt condition is true. The interrupt can trigger on signal changes, elapsed time, robot positions, data changes, and errors. Interrupts are adequate to use in cases where it is not expected or known when something will happen, such as a water tank overflowing or checking the communication with peripheral equipment. A trap routine is a routine that will be executed when the corresponding interrupt is triggered. When an interrupt occurs, it will immediately suspend the normal program execution and send the program pointer to the trap routine. It will *trap* the program pointer in the routine until the entire routine has been executed, after which it will return to regular operation. As interrupts can occur anywhere in the program, it is crucial to thoroughly test the robot's behavior by triggering interrupts at random. Using the virtual environment for this is superior as it is easier to reset with no risk of damage. In Viking, interrupts will be used to exit the program execution when the robot should go back home.

Task and program structure

Figure 41 shows the module and program structure within the robot controller. The whole set of system and program modules is called the Task,

and the program modules are referred to as the Program. There is no difference in what the system and program modules can do, only how the controller handles them. When saving or copying a program for later use, all program modules but none of the system modules are saved. Instead, the system modules are used to define data and routines specific to the current project, such as tool, load, or work object data.

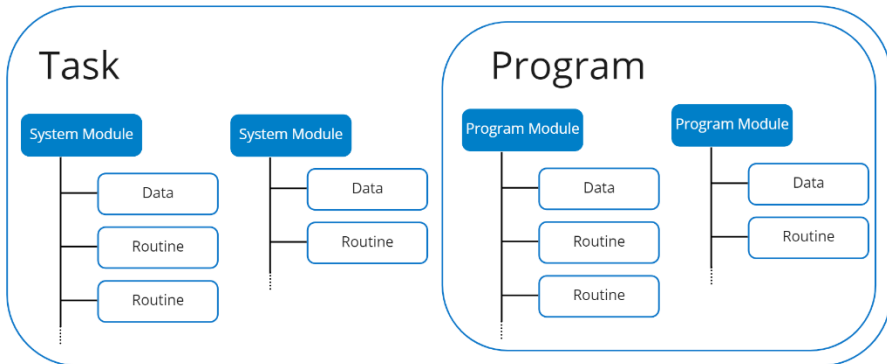


Figure 41 Schematic of module and program structure.

Usually, a controller can only have one task, the motion task, where all movements are programmed. When using two or more robots in a Multi-Move system, multitasking is enabled, allowing up to 20 tasks to run simultaneously. These tasks can serve different purposes, such as checking I/O, timers, alarms, or analyzing data during movement.

In this project, separate tasks will be added to simulate the PLC so that signals and operations can be tested before the virtual commissioning (see Appendix D for complete code). This simulated PLC can also aid the PLC programmer by visualizing the different sequences. The structure seen in Figure 42 is built to replicate the ladder logic structure used at Alfa Laval, where the step procedures contain conditions to be fulfilled before continuing to the next step. This is not necessary for virtual commissioning, but it helps during the development by giving structure and to enable partial virtual commissioning.

<pre> MODULE GasketRobotSteps TASK PERS num stepper:=1; PROC main() init; stepper := 1; WHILE TRUE DO CallByVar "action",stepper; CallByVar "step",stepper; incr stepper; ENDWHILE ENDPROC PROC step1() !Step conditions to move to next step ENDPROC PROC step2() !Step conditions to move to next step ENDPROC </pre>	<pre> MODULE Actions PROC init() !Initialisation, resetting signals, etc. ENDPROC PROC action1() !Actions executed during step 1 ENDPROC PROC action2() !Actions executed during step 2 ENDPROC PROC action3() !Actions executed during step 3 ENDPROC PROC action4() !Actions executed during step 4 ENDPROC </pre>
---	--

Figure 42 Structure of simulated PLC in RobotStudio.

Smart Components

Smart components enable advanced logic, motion, and calculations to be used in the virtual environment, separated from the virtual controller. Smart components can be likened to function blocks in PLC programming as they have underlying code packaged in a block with inputs and outputs. RobotStudio has a library with 73 pre-programmed smart components, and custom ones can be created with the use of RobotStudio’s SDK. Figure 43 shows a smart component containing 13 pre-programmed components, individually wired to their respective input and outputs.



4. Viking – From idea to commissioning

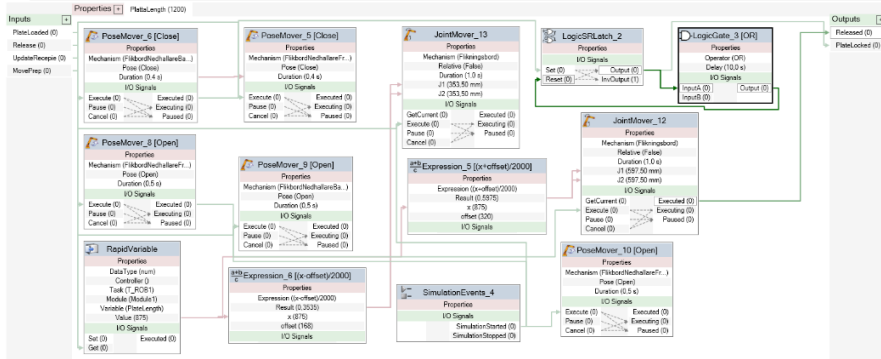


Figure 43 Example of a complex smart component, controlling the table.

Creating smart components is one of the most time-consuming parts of setting up the virtual environment, often containing hundreds of connections with a reasonably hard-to-troubleshoot graphical interface.

I/O signals

In total, 1035 I/O signals were set up for the virtual commissioning within RobotStudio. The distribution of signals is shown in Table 2. The smart components had the most connections overall due to the many mechanisms which need connections for every movement.

Table 2 I/O signals used within RobotStudio

Asset	Digital signals	Group signals
<i>Robot 1</i>	25	29
<i>Robot 2</i>	20	29
<i>Robot 3</i>	31	68
<i>Robot 4</i>	38	63
<i>Simulated PLC</i>	64	118
<i>Virtual equipment</i>	20	4
<i>Smart components</i>	324	202
Total	522	513

Signal analyzer

The signal analyzer can be used to record and display signals from the robot controller, either offline or online. Many data types are available, e.g. tool acceleration, tool changes, maximum linear acceleration, speed, and orientation. In Figure 44, the energy consumption is plotted together with speed, acceleration, and stop estimations in both time and position.

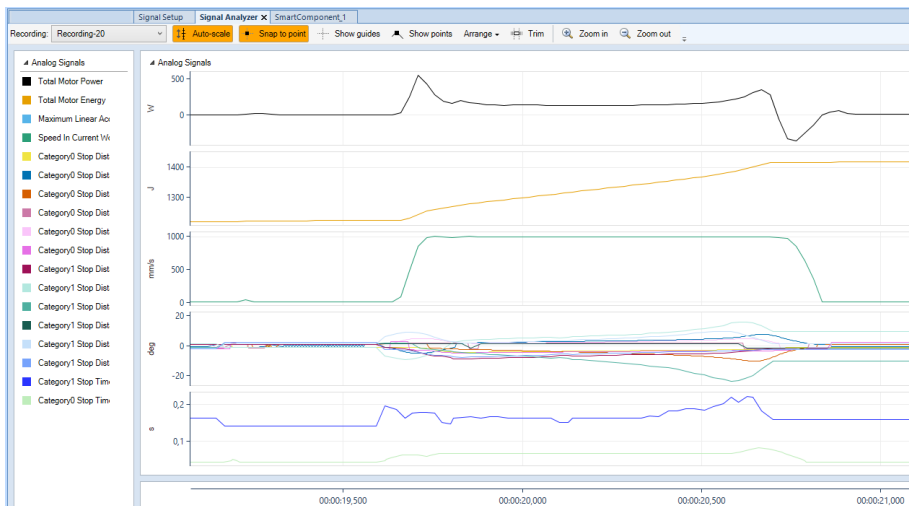


Figure 44 Recording within Signal Analyzer in RobotStudio.

Records, recipes

All data about the current order is stored as recipes in the HMI. The robot programs should, in that sense, be *naive*. The program should not handle any data other than simple calculations if necessary for quicker execution. When building these programs without an HMI or PLC, these recipe parameters need to be input and received from somewhere else. The data type *Record* can be used to achieve this. The record is a composite of other data types, such as strings, numbers, booleans, or other records. A Record can be used to give the same functionality as a class in other programming languages.

First, the record is defined by specifying attributes that should be included in the record. The record `PlateData` is defined below and will be used when creating a new instance of the type `PlateData`.

```
RECORD PlateData
    num PlateLength;
    num PlateWidth;
    num PlateThickness;
ENDRECORD
```

Then the `PlateData` `M10Plate` is created by assigning values to the different attributes.

```
PERS PlateData M10Plate:= [500,200,1];
```

Calling `M10Plate.PlateWidth` in a robot program would then return 200.

A robot standard was drafted during the project. Below is an excerpt of some essential guidelines.

- All non-trivial code should be commented.
- Variable naming should follow the Alfa Laval convention.
- The program should be parametrized to make the program adaptable to different products.
- Speed and acceleration should be able to be modified through parameters.
- An I/O list should be created for all communicative devices.
- Robots should move at appropriate speeds to reduce wear.
- Values should be logged to simplify troubleshooting.
- Handshakes should be used to coordinate programs, and all parameters should be mirrored to verify that all devices have synchronized variables.

4.5.2 PLC

Within Alfa Laval, the ladder logic language is used as it is regarded as easy to troubleshoot for the maintenance personnel. The PLC standard states that the first letter indicates the data type; in Figure 45, this can be seen where N is a counter, I is input, and B is a boolean. The last letter indicates in which program the variable is used. The standard template uses a stepper to step through the program as each step condition is fulfilled.

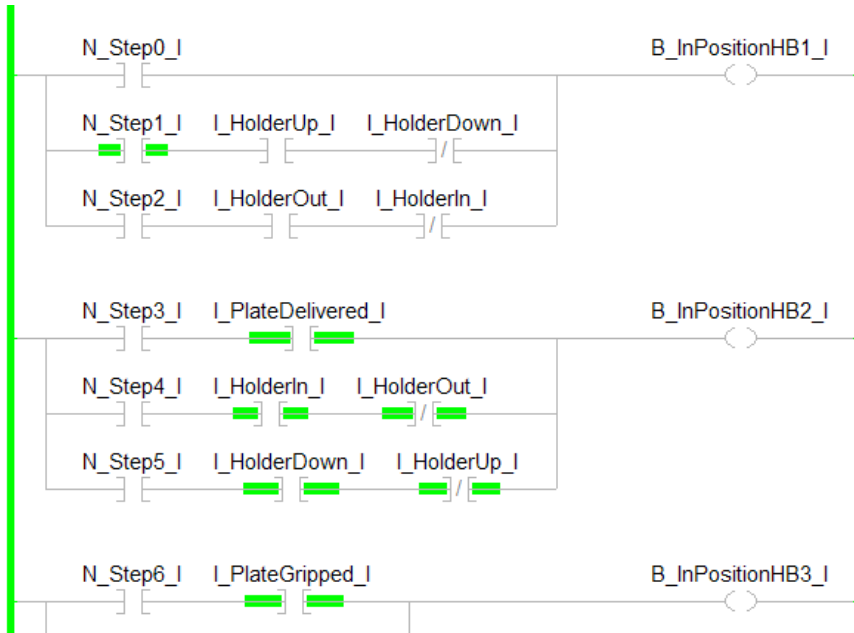


Figure 45 Ladder code of the table.

Once again, the table will serve as a demonstration object. First, the holder should be released to ensure it is not clamping (1). Next, the holder is moved outwards (2) to make way for the next plate delivered (3) by the portal. The holders are now moved in (4) and clamped (5) so that the plate can be gasketed. After the gasketing, the robot signals that the plate can be released (6).

4. Viking – From idea to commissioning

Using the standard template, the above routine can efficiently be converted to the ladder logic diagram shown in Figure 46, where each step corresponds to the described behavior. The PLC programmer will provide the PLC program for the virtual commissioning in the Viking project.

A PLC was simulated within RobotStudio to facilitate partial virtual commissioning (code in Appendix D). The setup of the simulated PLC was done within RAPID as separate tasks in the robot controller. The working principle of the tasks was made to replicate the PLC program.

```
1 MODULE GasketRobotSteps
2 TASK PERM num stepper:=9;
3
4 PROC main()
5   Init;
6   WHILE TRUE DO
7     CallByVar "action",stepper;
8     CallByVar "step",stepper;
9     Incr stepper;
10  ENDWHILE
11 ENDPROC
12
13 !wait for speed to be set
14 PROC step1()
15   Waituntil GRD0Speed = simrgdSpeed;
16 ENDPROC
17 !wait for correct job to be selected
18 PROC step2()
19   Waituntil GRD0JobNo = simGRD0JobNo;
20 ENDPROC
21 !wait for first job started
22 PROC step3()
23   Waituntil GRD0JobStarted = 1;
24 ENDPROC
25 !wait for job to be done
26 PROC step4()
27   Waituntil GRD0JobDone = 1;
28 ENDPROC
29 PROC step5()
30   Waituntil simSCDIgasketGripped =1;
31 ENDPROC
32 !wait for job go to table
33 PROC step6()
34   Waituntil GRD0JobMo = simGRD0JobMo;
35 ENDPROC
```

```
1 MODULE Actions
2 !Initialization step
3 PROC Init()
4   stepper:=1;
5   Reset simSCD0StretchGask;
6   Reset simSCD0GrpPacking;
7 ENDPROC
8 !set speed values
9 PROC action1()
10  SETGr simrgSpeed,100;
11  SETGr simrgAmp,180;
12  Reset simGRD0GasketReleased;
13 ENDPROC
14 !select job 2 "pickGasket"
15 PROC action2()
16  IF GRDIABSide=0 THEN
17    Set simGRD0ABSide;
18  ELSE
19    Reset simGRD0ABSide;
20  ENDIF
21 SETGr simGRD0JobNo,2;
22 SETGr simGRD0GasketLdth,3000;
23 SETGr simSCD0GasketLength,300;
24 SETGr simGRD0GrpOffet,400;
25 SETGr simGRD0GrpLct,11000;
26 ENDPROC
27 !start pickGasket
28 PROC action3()
29   !Set simGRD0AllowedGaskets;
30   Set simGRD0StartJob;
31 ENDPROC
32 !reset job started
33 PROC action4()
34   Reset simGRD0StartJob;
35 ENDPROC
```

```
1 MODULE PhysicalGasketRobot
2 !=====
3 !Module Name: Physical
4 !Version: 0.9.3
5 !Description: Module for the physical application
6 !Date: 27-03-22
7 !Author: Lucas Gardebrand
8 !=====
9 PROC main()
10  InitData;
11  WHILE TRUE DO
12    !Write "waiting for PLC to start job....";
13    WaitDI GRD0StartJob,1;
14    Reset GRD0JobDone;
15    declaration;
16    Set GRD0JobStarted;
17    log"Job "HmToStr(GRD0JobNo,0)+" begin";
18    TEST GRD0JobMo
19    CASE 1: ToHome;
20    CASE 2: PickGasket;
21    CASE 3: DropGasket;
22    DEFAULT:
23      TPRass;
24      SetGr PRGErrNo,21;
25      log"Mot OK JobNo = "HmToStr(GRD0JobNo,0);
26      TPRadFK MnAppVal,"OK","OK","","","";
27    ENDTEST
28    WaitDI GRD0StartJob,0;
29    Set GRD0JobDone;
30    log"Job "HmToStr(GRD0JobNo,0)+" done";
31    Reset GRD0JobStarted;
32  ENDWHILE
33 ENDPROC
34
35 !300 1 move robot to home
```

Figure 46 Simulated PLC within RobotStudio.

To the left in Figure 46 is the step program, which contains a loop that steps through the program after the condition of each step has been fulfilled. In the middle is the *Action* task which can be seen as the output of the PLC program. Whenever a step is active, the corresponding action will be carried out. To the right is the actual robot program, which remains unchanged. This gives the ability to switch between the real or emulated PLC and the simulated PLC without any program changes. It can also be used to run the virtual environment before the PLC code is done.

4.5.3 HMI

Alfa Laval uses Aveva InTouch (former Wonderware) for their HMIs, with a backend SQL database containing all parameters. The parameters are divided into machine parameters and recipe parameters. Machine parameters are the same no matter which product is being handled, such as the positions of stationary objects within the cell. The recipe parameters differ from product to product, such as plate length, width, and thickness. All these parameters need to be accessible to the operator and maintenance personnel. It should be noted that it is possible to make custom HMI screens within RobotStudio as the FlexPendant is able to show both text and images, as well as receive numerical and text data, but due to Alfa Laval standards, this is not an option. The HMI used in Viking (Figure 47) is built on an old template by automation engineers.

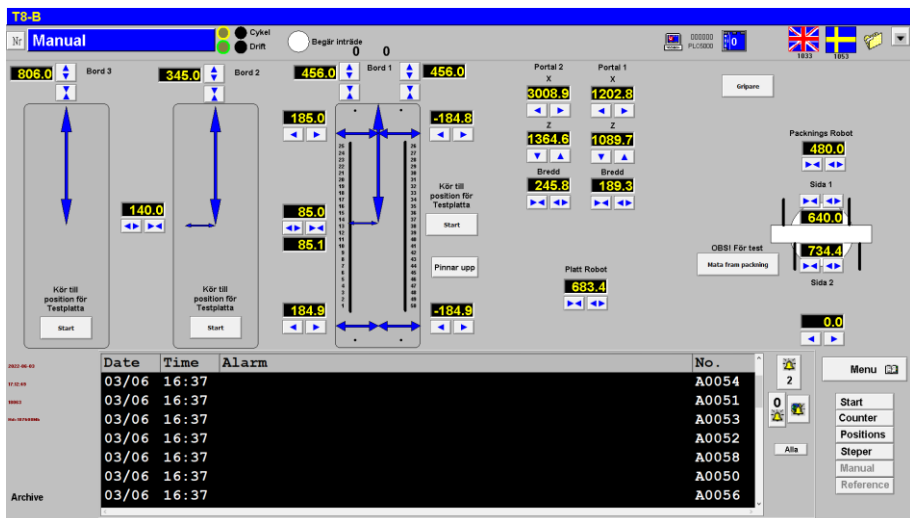


Figure 47 The HMI for Viking in calibration mode.

4.6 Safety

This chapter will focus on the robot safety, as the rest of the safety configuration was set up by other engineers at Alfa Laval. On the PLC side, appropriate safety modules were used, and the safety functions mainly consisted of non-contact sensors on doors, light curtains, roll-up doors, and emergency stops. CE-marking is done together with a third party, and risk assessment will be done within the project group.

Robot safety

SafeMove is a complete safety solution with dedicated safety hardware and software, developed by ABB for their robots and will be used for the robot safety in Viking. SafeMove is included in RobotStudio, and all safety configurations can be made in the virtual environment, which allows for virtual commissioning of the safety configuration.

SafeMove can not control the robot, nor can it in itself prevent a safety violation; it only acts afterward.¹¹ This is desired behavior as the safety program should not affect normal program execution. Because of this, other safety functions are generally needed, such as light barriers or scanners. Multiple types of supervision are available within SafeMove, such as position, speed, and orientation of both tool and axes. Standstill supervision can be used to make sure the robot is still and can be useful when an operator enters the cell or adjusts an object.

For this project, twelve zones, four ranges, and four orientation supervisions were used, as showed in Figure 48. Zones over the tables are there to ensure that the robots are not in the way of the portals and vice versa. Two light curtains and one roll-up industrial door will complement the cell

¹¹ There are other functions, such as *collision avoidance*, that are not within the robot safety module, that can predict if the robot will hit defined objects.

safety. No operators are allowed to be within the cell during normal operation but may enter to refill gaskets or plates. Maintenance personnel will also be in the cell when performing maintenance. The SafeMove configuration allows manual movement of the robots but limits the speed and location to protect the equipment.

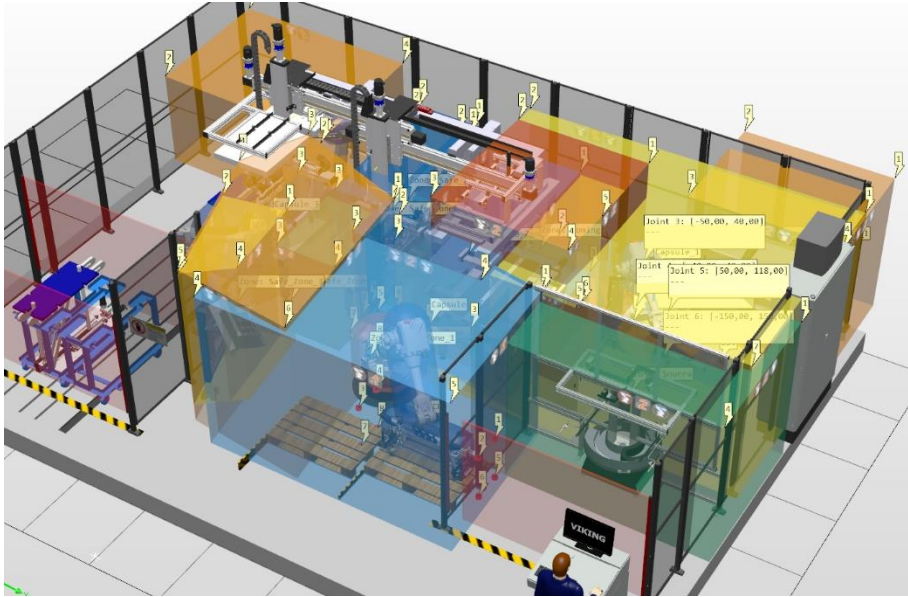


Figure 48 Safe Zones visualized within RobotStudio.

Cyclic brake check

Cyclic brake check is a periodically run routine that will hold the brakes and apply force on each axis until the axis moves. This is done to ensure that the brakes work as intended; any issues will result in a warning or error on the FlexPendant. With the use of Safemove, the cyclic brake check can be enforced and will trigger a stop if the check has not been performed within the prescribed time. If the brake check fails, a brake maintenance routine will be run to slightly burr the brakes for better traction, which might make the brake check pass. Cyclic brake checks were set up to run once every 720 hours, with a pre-warning time at 11 hours to allow the PLC to schedule a brake check at an appropriate time.

Safety configuration

When the safety configuration was written to the robot controller, a safety configuration report was automatically generated containing information about the configuration creator, controller version, Safe Zones, safety signals, and a checksum (Figure 49).

ABB Safety Configuration Report

A detailed description of functions and validation procedures can be found in the SafeMove application manual.

1. General Information.....	1
2. Safety Configuration.....	2
3. Safe I/O Configuration.....	19
4. Combinatorial Logic Configuration.....	22

1. General Information

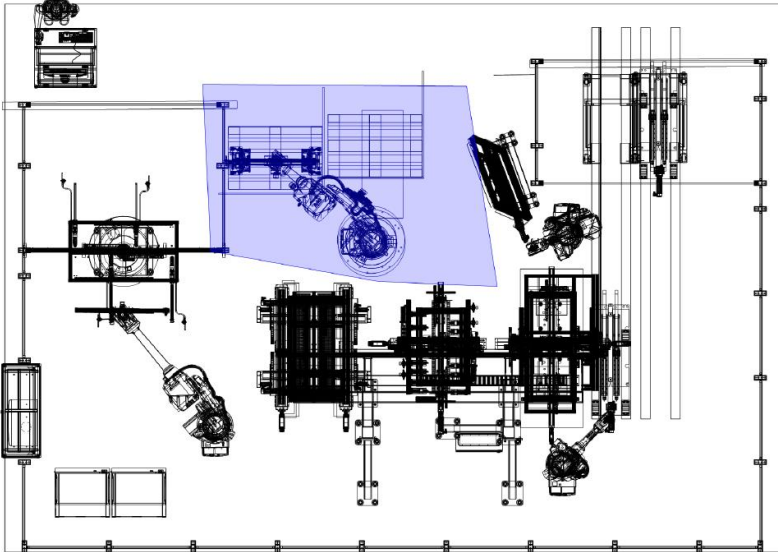
Created by:	Lucas Gardebrand
Creation date:	2022-05-05T13:37:42.069+02:00
System name:	GasketAndPlateRobots
Configuration version:	1.03.01
Controller image version:	1.03.06
Checksum:	489E14EE6B2AFF54F4C071573B2E04A98F85BFBD3784E86AFC613E2930A7DE8
Protected elements checksum:	F1945CD6C19E56B3C1C78943EF5EC18116907A4CA1EFC40A57D48AB1DB7ADFC5

Figure 49 First page of the auto-generated safety configuration report.

A description for each SafeZone, including geometry, activation signals, stop categories, and stop signals, was automatically generated as well, as seen in Figure 50.

When the Viking cell is installed in the factory, the report should be signed by a designated person responsible for the factory's safety and be put in the robot controller cabinet. The 64-character checksum is unique for every configuration and can be used to make sure that the signed configuration is identical to the configuration loaded in the controller.

PlateStackZone - Floor plan



PlateStackZone - Tool Position Supervisions

Tool Position Supervisions - PlateStackZone_TPO

Activation	Function active status	Violation stop category	Violation signal
PRSIAllowedStack	No signal	Category1Stop	No signal
Include upper arm geometry	Allow inside		
true	false		

Figure 50 Excerpt from the auto-generated safety configuration report, showing one of the twelve Safe Zones and its signals.

Stop visualization

A new feature in RobotStudio is the visualization of stop positions recorded in the signal analyzer (see Chapter 4.5.1). The movements can be interpolated to create a swept volume that displays where the robot would end up if the stop occurred at any time during operation (Figure 51).

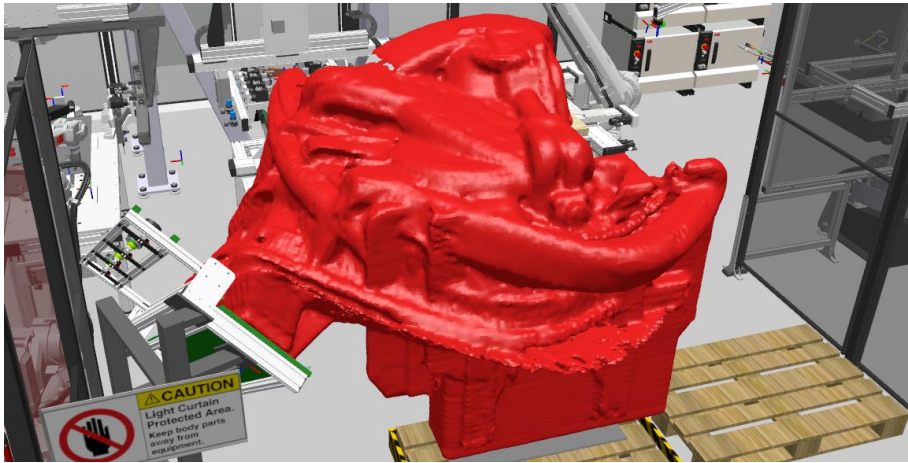


Figure 51 Swept volume of where the robot would stop if a Cat. 1 stop were triggered.

Stop positions for specific moments in time can be viewed in the playback function, where the stop position is shown as a ghost image that can be used for collision detection and measurements between parts (Figure 52).

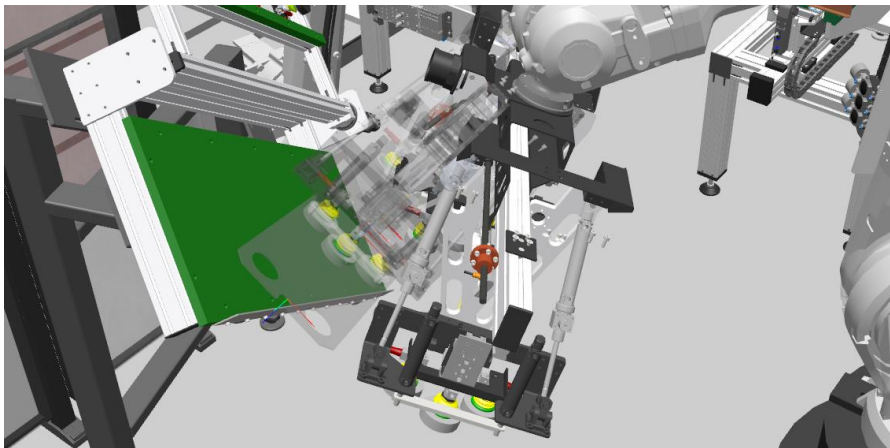


Figure 52 Ghost image of where the robot would stop if a Cat. 1 stop were triggered.

4.7 Virtual commissioning

While the vision system could be virtually commissioned by feeding images through the vision software, it was decided not to include it in the commissioning as the vision interface had been tested before, and the effort of taking virtual images would outweigh the benefit. As a substitute for the vision system, its output was simulated by randomly generating a translational and rotational offset from the set position to still account for the unknown gasket position. By doing so, it is still possible to test the robustness of the robot program.

4.7.1 Interface connection and communication

RobotStudio only has native support for Siemens PLCs, leaving OPC-UA the only option to connect to the Allen-Bradley PLC. The setup below describes the steps to connect the PLC to RobotStudio and the HMI to the PLC using an Allen-Bradley PLC.

Studio 5000 setup

Rockwell recently launched a new PLC emulator called Factory Talk Echo, which replaces the old Logix Emulate software. The setup in Echo is much simplified. First, the Studio 5000 project was prepared for emulation by changing the PLC controller to a 5580 CPU. Echo then imports the project file directly, setting up everything according to the project settings (Figure 53). The PLC emulation was then connected to the OPC server, in this case, FactoryTalk Linx Gateway.

4. Viking – From idea to commissioning

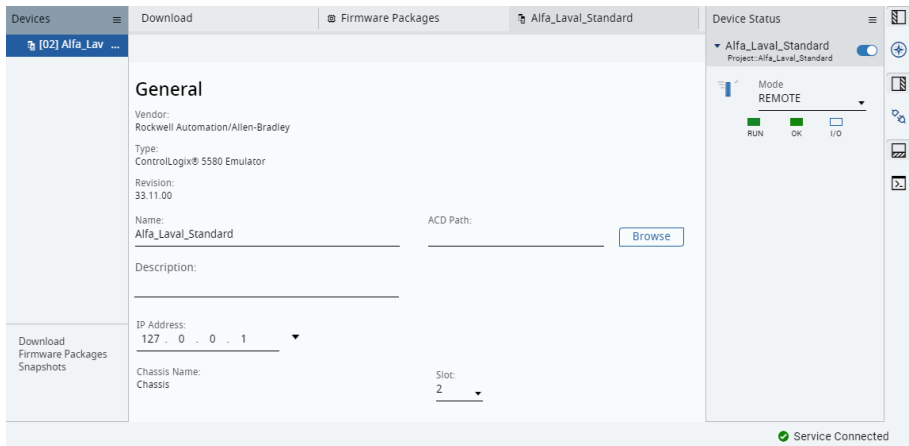


Figure 53 Echo, after importing the Studio 5000 project file.

OPC-UA server setup

The OPC-UA server was configured using FactoryTalk Linx Gateway. A service name was specified, and an endpoint was created with a matching name (Figure 54). As the server was set up locally, the default port 4990 was used for the endpoint. No encryption was used, as the connection could not be completed otherwise.

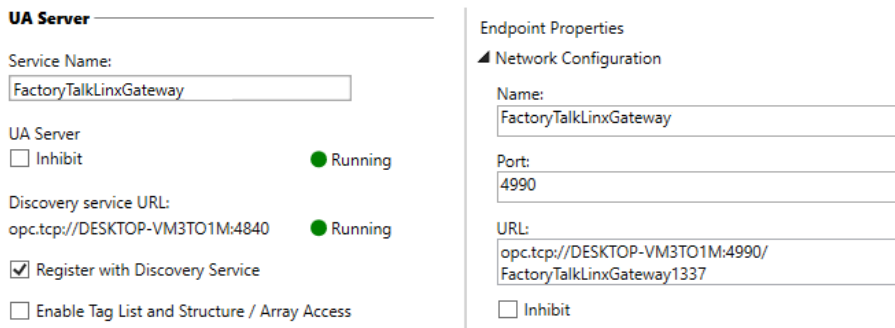


Figure 54 UA-Server and endpoint configuration within FactoryTalk Linx Gateway.

For the tags from the Studio 5000 application to be readable within the OPC-UA server, a shortcut had to be created using FactoryTalk Administration Console (Figure 55). The shortcut takes a snapshot of the tag structure of the PLC and makes it available on the server.

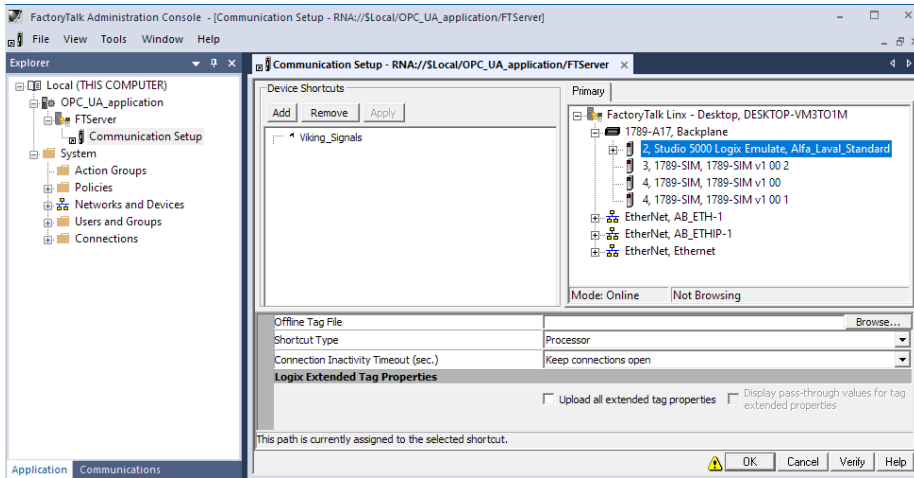


Figure 55 Shortcut setup in FactoryTalk Administration Console.

RobotStudio setup

Connecting Rockwell equipment with RobotStudio has been a considerable challenge in the past, with multiple companies trying and failing due to the complexity. RobotStudio's lack of an OPC client contributed to this hardship, and the OPC server was difficult to use. An OPC-UA client was implemented in RobotStudio 2021 as a Smart Component (see Chapter 4.5.1 Robot). Still, it had limited functionality, and all signals had to be defined in a separate excel configuration file. A Graphical User Interface was added in the second revision, making the process less troublesome. The OPC-UA SmartComponent was used to connect to the server by inputting the endpoint URL from FactoryTalk Linx Gateway. In the graphical configuration interface, all signals from the shortcut could be found on the previously created OPC-UA server. The tags needed for the virtual commissioning were pulled from the server node into the input and output signal window of the OPC-UA client (Figure 56).

4. Viking – From idea to commissioning

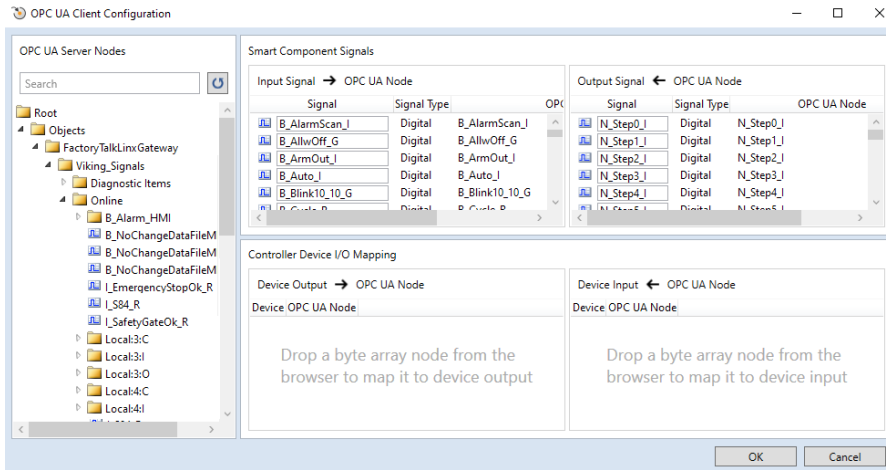


Figure 56 Graphical OPC-UA Client interface within RobotStudio.

HMI setup

First, an IO gateway was created using the ArchestrA System Management Console (Figure 57). Similar to the RobotStudio setup, the endpoint URL was used to connect to the OPC-UA server.

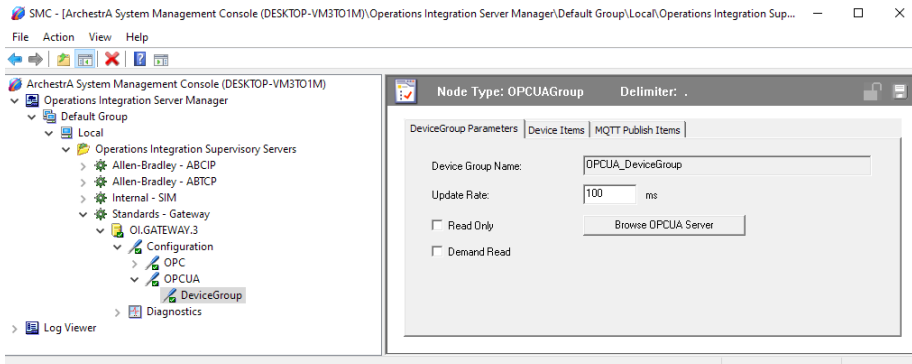


Figure 57 Setting up the IO-Gateway in ArchestrA System Management Console.

The Topic name in InTouch WindowMaker then had to be set to the same value as the Device Group Name in from the System Management Console (Figure 58). The SQL database had to be updated using the same group name.

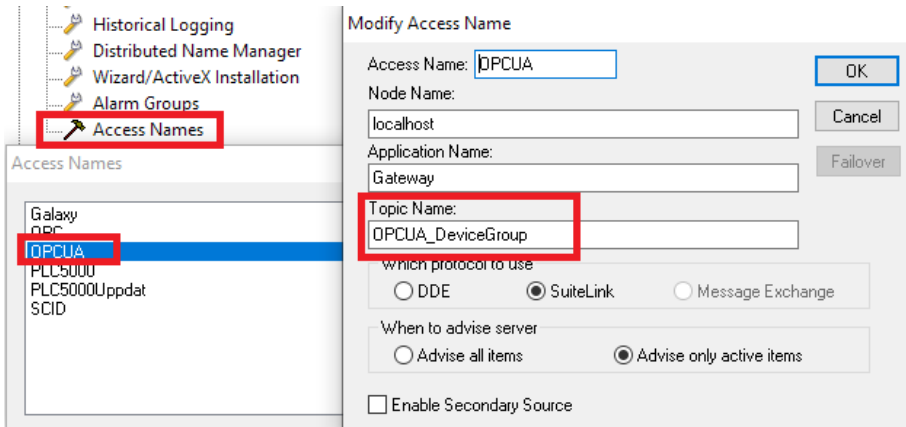


Figure 58 Configuring the Access Name in InTouch WindowMaker.

Connection performance

When trying a simple handshake in the Viking cell, the one-way route time was measured at 46 ms. This delay might be too slow for very time-sensitive applications that rely on fast actions. However, for applications akin to the one presented in this thesis, most delay issues outlined by Carlsson (2012) would be unnoticeable. The above benchmark was performed with a simulation time step of 24 ms in RobotStudio and a PLC scan time of 20 ms. Therefore, the delay is most likely introduced through the software, not the OPC communication protocol.

Handshakes

The PLC is the master in this application, and the robot is the slave working through remote procedure calls with the request-reply method. The handshake procedure and job order can be seen in Figure 59. First, the robot waits for the signal *startJob* to go low, ensuring that the jobs are only triggered on the positive edge. The *jobStarted* and *jobDone* variables are reset, and the robot now waits for the PLC to order a new job. When a new job is called, the value of *giPLC* will be checked, and the corresponding case will be executed. After the execution, *jobDone* is set to 1, signaling the job completion. The procedure is then looped.

```
PROC PLCWaiter()
  WaitDI startJob,0;
  SetDO jobStarted,0;
  WaitDI startJob,1;
  SetDO jobStarted,1;
  TEST giPLC
  CASE 1:
    !Do some job
  CASE 2:
    !Do some job
  CASE 3:
    !Do some job
  CASE 4:
    !Do some job
  CASE 5:
    !Do some job
  CASE 6:
    !Do some job
  DEFAULT:
    TPWrite "illegal choice";
    Stop;
  ENDTTEST
  SetDO jobDone,0;
ENDPROC
```

Figure 59 Handshakes between PLC and robot in RAPID.

In some instances, the robot is pseudo-master as it can control the pneumatics of the tools by the PLC forwarding the robot signals to their respective solenoid.

A spreadsheet of I/O-signals was set up in collaboration with the PLC programmer to communicate how the PLC and robots should be configured. The two robots will share the 128 bytes on the controller. Values sent to the robot will either be one-bit booleans or 16-bit integers. Distance values will be represented in tenths of millimeters for adequate precision. This procedure would have been the same even if no virtual commissioning were to be made.

4.7.1 vFAT

A vFAT protocol was drafted as no vFAT had been performed in-house before. As the premise of virtual commissioning is that it can reduce the physical commissioning, a vFAT protocol was created based on previously used FAT protocols. In Table 3, the test types are listed together with test suitability in vFAT with simulated PLC, vFAT with emulated PLC, and physical FAT. The suitability of each test category is described below.



Table 3

Table of test types and the test suitability of three FAT configurations

Criteria	vF _s	vF _e	F
General			
Conformance with specification	Red	Red	Green
Operators or maintenance personnel present during FAT	Green	Green	Green
Important documents are printed and stored	Red	Red	Green
User Interface			
Easy to understand the presented information	Green	Green	Green
Necessary operation information is shown	Green	Green	Green
Easy input parameter programming	Green	Green	Green
Ergonomics			
Cognitive interaction with the production cell is intuitive	Green	Green	Green
Physical interaction with the production cell is intuitive	Green	Green	Green
Repetitive movements have been reduced	Green	Green	Green
The operator works at the proper height with proper lighting	Yellow	Yellow	Green
Performance			
Full-scale test with recipes	Yellow	Green	Green
Communication between all systems work as intended	Yellow	Green	Green
Capacity according to specification	Green	Green	Green
Start, interrupts and shutdown procedures work as intended	Green	Green	Green

Alarm functions work and show information about faults	Red	Green	Yellow
Exception, signal disturbance, and fault mode testing	Yellow	Green	Yellow
Safety			
Safe Zones trigger and stop execution	Green	Green	Green
Emergency stop triggers and stops execution	Red	Yellow	Green
Emergency stops in proper locations	Green	Green	Green
The design fulfills machine safety regulations	Yellow	Yellow	Green

Typical FAT protocols always have criteria covering physical assets or parts. These criteria can evidently not be fulfilled through a vFAT. The user interface can be tested with any configuration. Ergonomics can be tested through VR and is preferably done before the FAT as changes to the mechanical design is costly at that stage. Lighting is difficult to simulate, and the cost to add lights is generally low; thus, it should be left for the FAT or be tested separately. Most performance tests can be done using the virtual configurations, except for verifying communication when the PLC is not emulated. Alarms are advantageously created and tested in the virtual models as it is easier to simulate faults, and the time to reset the cell for a new alarm test is shorter. The same is true for exception and fault mode testing, where there is a risk of damage to equipment. If the HMI is communicating with the PLC, the simulated PLC method is less suitable for trying alarms and faults. Critical safety functions should always be validated during the FAT, even though testing them virtually can add confidence that the safety functions work when performing the FAT.

Before the PLC program was completed, a partial virtual commissioning was done using the simulated PLC described in Chapter 4.5.2. First, the program was executed according to ordinary operations to verify movement, paths, collisions, capacity, and recipes. Secondly, starting and stopping procedures were tested from various positions to ensure restarts could be made at any time. Lastly, errors and bad input were tested by setting signals to values that should not be possible to find any dangerous behaviors, validating errors thrown by the robot controller.

In preparation for the emulated PLC vFAT, all axes within the PLC program had to be exchanged for virtual axes, which resulted in errors that could only be solved by removing parts of the program as virtual axes do not have the same setup as normal axes. In addition to this, FactoryTalk Echo had no support for safety modules, all of which also had to be removed, resulting in more errors as many segments of the program rely on the safety signals. Many changes had to be done, counteracting the purpose of the virtual commissioning, and while the vFAT with the emulated PLC could be run, as seen in Figure 60, it was decided that it was not worth the effort.

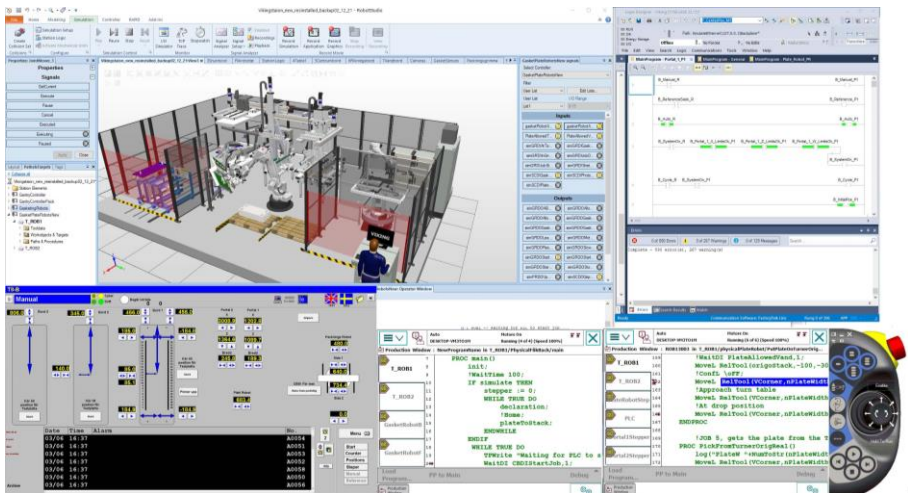


Figure 60 vFAT with the virtual environment, HMI, emulated PLC, and Robots.

4.7.3 Hardware requirements for commissioning

To ensure reliable simulation execution and mitigate time delays causing the simulation and PLC to go out of sync, a 2022 state-of-the-art workstation was used running Windows 10. The system was compared in a series of tests against a 2014 state-of-the-art PC to understand to which

extent hardware advancements enable virtual commissioning. The hardware components in the newer PC were the AMD 5900X 12 core, 4.5 GHz processor, an Nvidia 3070 graphics card, and 64 GB of RAM. The hardware used in the older PC was the Intel i7 4970K 4 core, 4 GHz processor, two Nvidia GTX 970, and 16 GB of RAM.

The two systems did not differ in performance noticeably when running a large simulation of several production cells. Both were able to process the scene in real-time, albeit with a significant drop in frames per second. When inspecting the hardware usage, it became clear that the computer was not efficiently used. Instead, the software held back the performance because it ran on a single core. The presented hardware allows for a simulation time step of 24 ms and a physics time step of 12 ms without any desynchronization in RobotStudio. Running the system at a 12 ms time step results in a 25% slower simulation than in real-time. However, the 24 ms simulation time step is enough for this application.

4.8 Further uses of virtual models

4.8.1 3D Scanning and point cloud

As Viking is a new production line, the plant layout is still in the planning stage. A point cloud was used to aid the layout planners and validate different layout configurations by positioning the production cell at various locations. Autodesk ReCap was used to isolate and remove points of objects where the production line would be, see Figure 61.



Figure 61 Points marked for removal within Autodesk ReCap.

The point cloud was then imported to RobotStudio through the Point-Cloud Add-in, as point clouds are not supported natively. A stand-alone simulation was created with the point cloud included, which was sent to the layout planners for review, as seen in Figure 62.



Figure 62 Pointcloud of the factory imported into RobotStudio.

4.8.2 Virtual reality and Augmented reality

A new feature in RobotStudio is the ability to have virtual meetings. A meeting was set up by generating a unique meeting ID, which was shared with the participants. Both participants with and without VR systems joined the immersive discussion session.

4. Viking – From idea to commissioning



Figure 63 Virtual meeting using VR in the virtual environment.

In the Viking project, there was no opportunity to use AR. However, it could be tested in an AR application set up by MITC, where the operator is guided through the steps in an assembly process (Figure 64). Using the Microsoft HoloLens, information is overlaid on the physical objects tracked using fiducial markers¹² so that the visuals only appear where and when they are usable. As the operator learns the process, more and more virtual objects can be removed until the interaction is smooth and efficient.



Figure 64 HoloLens Demo at MITC (Gardebrand, 2021).

¹² A fiducial is an object used as a point of reference for computer vision.

4.8.3 Robot selection

There are a dozen large robot manufacturers and hundreds of different robot variants. If the factory already has an established robot manufacturer, it is often natural to continue using that brand as it is more familiar, the in-house knowledge is larger, and there might be economically advantageous agreements already in place. Selecting a specific robot model can be a more challenging task. Alfa Laval has a long track record of working with ABB robots. In the Lund factory, ABB robots are used almost exclusively. Thus, we had to use ABB robots for the project.

Having a well-defined virtual model assisted in picking the correct robots by providing the key metrics described in chapter 2.2.1, such as weight, center of gravity, and inertia. In Inventor, the center of gravity and moment of inertia of a model were retrieved from the physical tab of iProperties. As shown in Figure 65, the center of gravity is aligned with the robot tool z-axis, 254 mm away from the base. The moment of inertia around the three axes is shown as well.

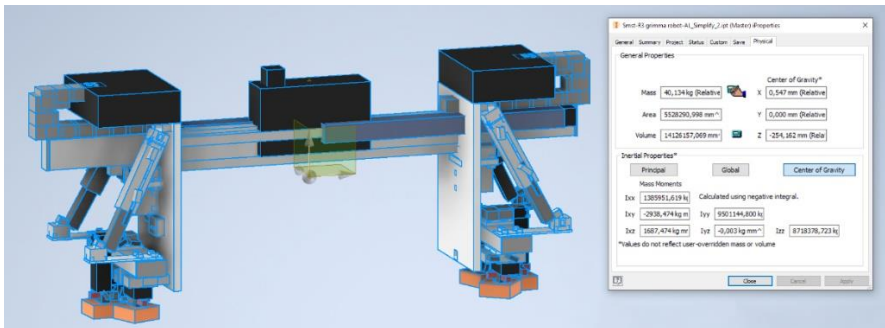


Figure 65 CAD-model of plate separator with physical data, simplified due to confidentiality (Inventor 2022).

These measurements were inputted in the RobotStudio RobotLoad add-in (see 2.2.1 about payload). Within RobotLoad, each load can be specified, such as tool load, load held by tool, and armloads. In this case, loads for a mounting plate and pneumatic valve manifold on axis three were added.

4. Viking – From idea to commissioning

As a smaller robot usually means less cost, safer operation, and easier handling, the goal was to find the smallest robot with sufficient reach and payload capabilities. The choice for this application was the IRB 4600-60 with 60 kg max payload and 2.05 m reach. RobotLoad generated a load diagram and ratings for each axis that was exported as proof of handling capability (Figure 66).

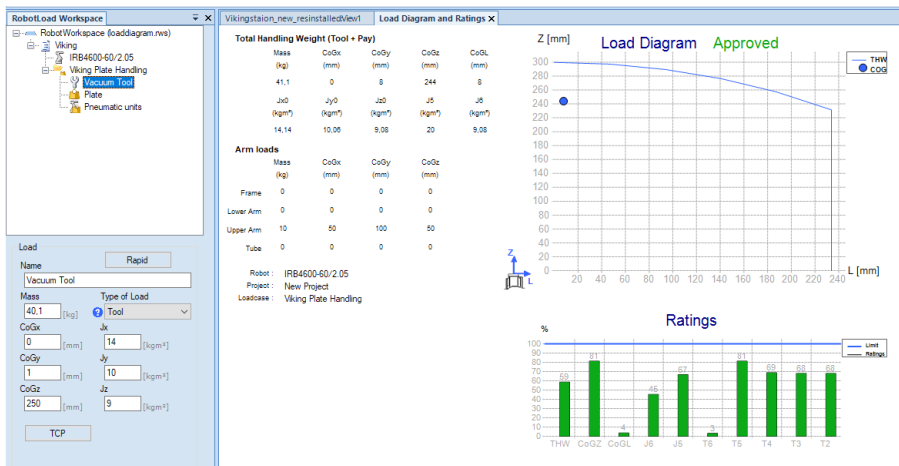


Figure 66 RobotLoad set up according to the tool data and payload with the IRB 4600-60 robot.

For comparison, Figure 67 shows that the IRB 4600 with 45 kg max payload would overload the fifth and fourth axis, and the IRB 6620 with 150 kg max payload would be excessive. The robots that will gasket and move the plates onto the exchanger frame do not need to have a large lifting capacity but need to reach two frame stations. A pair of IRB 2600 robots had sufficient reach and were selected for the gasketing. With CAD models of the cell readily available, it was easy to carry out a preliminary gasketing route to make sure the robot had enough space and reach without hitting other mechanical parts. Being able to estimate which robot model is needed quickly helps the layout design, cost estimation, and purchasing.

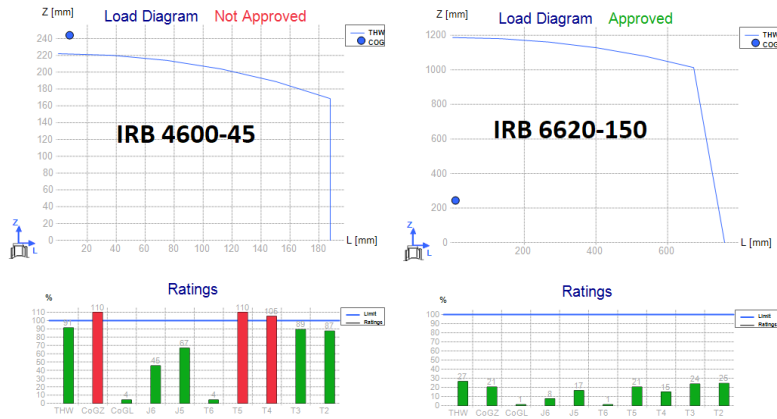


Figure 67 Load diagrams for an insufficient robot (left), and a superfluous robot (right).

4.8.4 Additive manufacturing and generative design

In the Viking project, additive manufacturing played a crucial role in the prototyping phase. Instead of the usual procedure where parts were designed and ordered for manufacturing from a third party, the design could be tested and refined in-house in a much shorter period. This freed up more time to go through more ideas and revisions, ultimately ending up with a better design (see Figure 68).

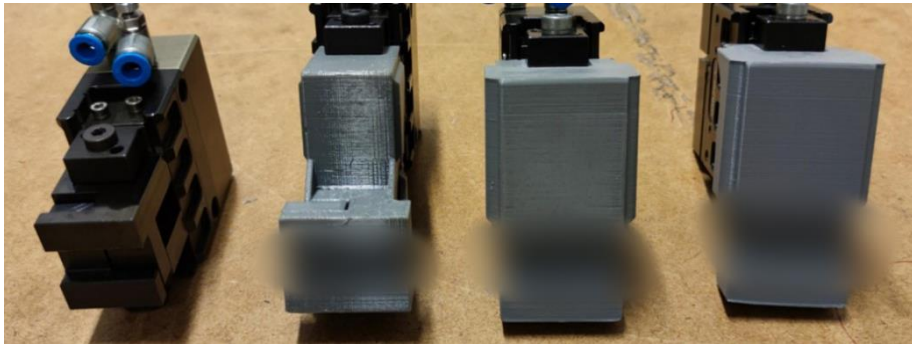


Figure 68 Major revisions of grippers, evolving from left to right. Image blurred due to confidentiality.

The final design spread the gasket fingers 313% wider than the revision before, using a geometry that would have been expensive to manufacture

with traditional methods. With additive manufacturing, it took 128 hours¹³ to print the 16 grippers with a total material usage of 682 g costing 54 EUR and machine wear costing 48 EUR. In contrast, manufacturing the parts with traditional methods was quoted at € 5200.

Due to AM being a new practice within Alfa Laval, extensive stress testing was conducted before including them in the final production unit. The gripper was printed in PETG. A stress test with cyclic gripping was performed in which the gripper sustained 500 000 gripping cycles with a load of 300 N before breaking. The fracture geometry seen in Figure 69 suggests that it was fatigue in a layer that caused the failure. It should be noted that the infill was sparse, and the gripper was printed in the worst orientation to test the worst-case scenario of the print.

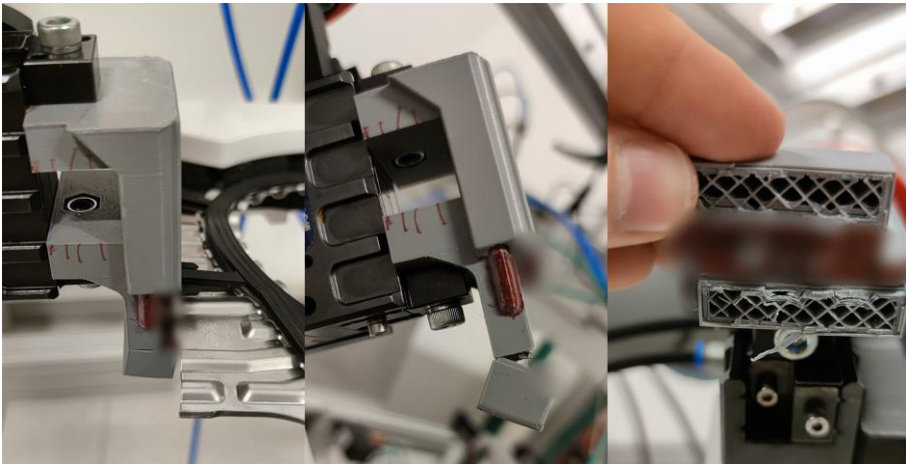


Figure 69 Fracture geometry of gripper after fatigue testing.

A few modifications were made where the lever arm was halved, the infill was set to 50%, and the support erased. As the gripper would only experience 75 N in normal operation, the modified version should hold for years.

The tool mount in Figure 70 was generatively designed, making it very rigid despite being printed in PETG.

¹³ PETG, Infill 40% rectilinear, 5 perimeters, variable layer height 0.07-0.2mm.

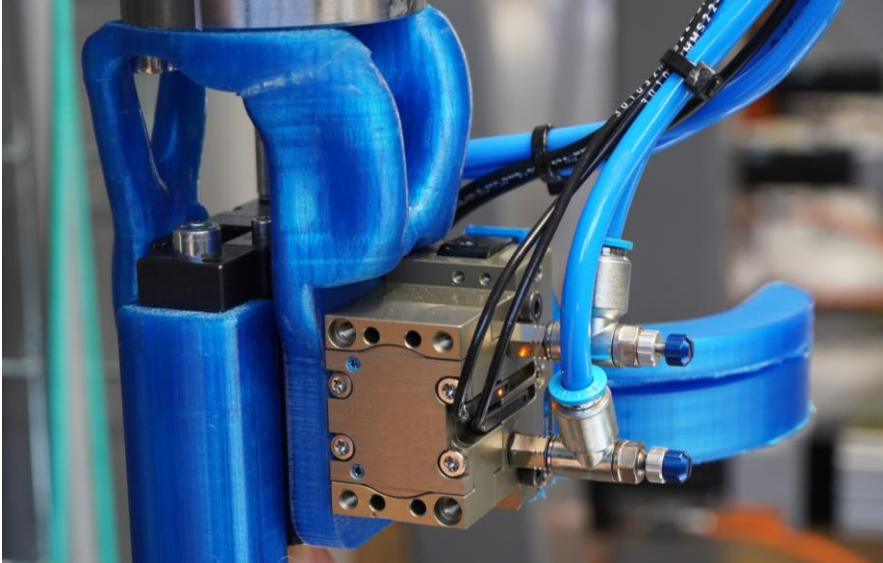


Figure 70 Generatively designed tool mount used in Viking.

4.8.5 Operator ergonomics and feedback

Statistics on occupational injury from the Swedish Work Environment Authority show that heavy lifting and carrying are common causes of stress injuries, see Figure 71. The most common cause of reported occupational injuries is repetitive work and lifting, which accounted for 55% of reports (The Swedish Work Environment Authority, 2014). Thus, it is desirable to lower the volume of heavy tasks and ensure the required task variation. A good overview of the system is needed to lower the number of repetitive tasks. With virtual models, it is possible to mitigate these issues in the early stage of the design process. With the advancement in simulation software and better knowledge and accessibility, more tasks can be automated. In the Viking cell, the operator only interacts with the process when starting a new program or refilling gaskets and plates. This interaction will happen once every hour with the current gasket and plate capacity. In this manner, the cell is already heavily automated and does not call for any

extensive physical ergonomics investigations. This is a considerable improvement from the old process where each gasket was mounted by hand, which was a labor-intensive process with the risk of repetitive strain injury

Reported work-related injuries

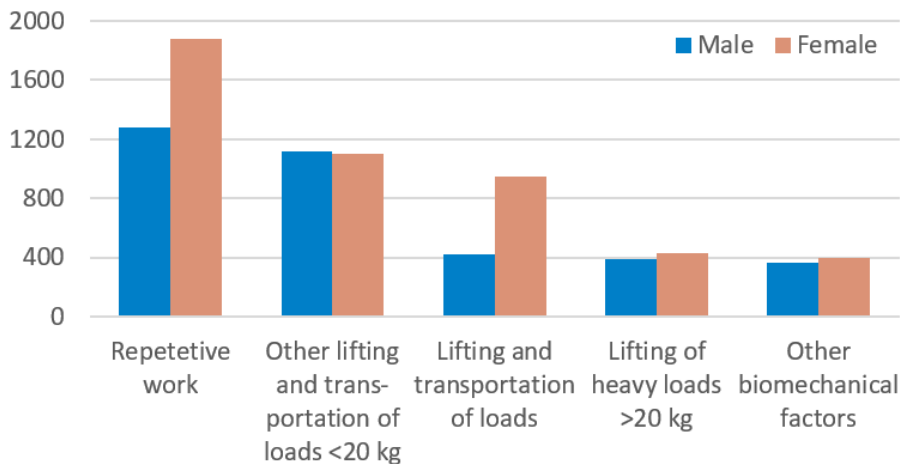


Figure 71 Data of work-related injuries, translated from The Swedish Work Environment Authority (2014).

in the long run. Instead, the focus should be directed toward ensuring the user can keep the process running automatically. The HMI still contains lots of data due to the large number of products to be assembled. Designing an intuitive HMI is outside the scope of this thesis, but it should be noted that having the virtual environment to test in will make it easier to find faults and make sure alarms are communicated in a comprehensible way. The current HMI leaves a lot to be desired with lots of distracting information displayed and vague affordances and signifiers. The menus are hard to navigate, relying on the knowledge of the operator. This increases the risk of mistakes and slips, ultimately affecting the production depending on which operator is on shift.

Getting the operators' view on the ergonomics can improve the process. Here the virtual models can be used to benchmark procedures in terms of

repetitive movements, accessibility, and HMI-intuitiveness. As seen in Figure 72, VR is used to test the physical ergonomics of a drilling process.



Figure 72 Testing ergonomics and operation procedures using VR.

4.9 Physical commissioning

While building the physical production unit, the virtual models were used multiple times to verify last-minute modifications as some parts had to be added for functionality. The production cell was assembled in a testing facility, allowing for a more methodological approach with no overhanging deadlines.

The virtual model should show temporal and spatial coherence with the physical model. If the virtual movements differ too much from the physical movements, the usability of the virtual model lessens as the results can't be fully trusted. To test the spatial and temporal coherence of the current setup a robot program was created in RobotStudio using the virtual model of a calibration sheet. The same calibration sheet was precision machined and put in the physical world. The robot program (Appendix B) was made to include all different kinds of instructions such as fine-points, high-speed movements, timed movements, wait-instructions, and zone targets. The

4. Viking – From idea to commissioning

test showed excellent coherence between the physical model and the virtual model when using RobotStudio and ABB robots.

5. Discussion

In this chapter, I discuss the results from the literature review and industry research and the outcome of the virtual commissioning within the Viking project.

Literature review and industry research

This thesis deploys a lean and agile approach by utilizing virtual models from the very beginning of the project, with continuous improvement and quick exploration of ideas in the simulation environment. The coarse models created after the first project meetings help give an initial sanity check and approximate cycle times. The refinements of models progress gradually as the project develops. In this sense, it is similar to the integrated approach proposed by Oppelt and Urbas (2014). Contrary to their reasoning of having a well-defined separation between automation and simulation, I would argue that the separation would hinder a complete validation of the code. Any additional code needed for the virtual commissioning to work would have to be added no matter if there is a separation; the separation would only add more complexity and become prone to synchronization errors.

Many of the reviewed papers did not account for the time spent setting up the virtual models necessary for virtual commissioning. In the study by Zäh et al. (2006) and Schamp et al. (2018), the virtual environment was already set up and was ready to be programmed. As time spent preparing virtual models for virtual commissioning can be substantial, this may reduce the results credibility of virtual commissioning. Adding kinematics and interface connections to the virtual environment will still consume a notable amount of resources and competence even though readily available virtual models are created during the mechanical design phase, no matter which commissioning method is used. A better benchmark would be to

look at the industry where no simplifications can be made, and all time has to be accounted for. Similar to Zäh et al. (2006) and Reinhart and Wunsch (2007), Evomatic did not notice a significant time reduction but found that the time was spent earlier in the project and the office instead of in the factory, which is cheaper. The consensus of the interviewed companies was that virtual commissioning reduced the cost of the project considerably and, in some instances, saved time too. It is important to remember that virtual commissioning is just a tool and has to be treated accordingly. It will not be viable for every project and, as stated by Reinhart and Wunsch (2007), may take on a different form depending on which resource is to be optimized.

Norrman and Persson (2018) found it possible to gather programming code from existing machines but hard to create a digital model using it. This shows the importance of including the virtual models as a delivery in the request for quotation from the very beginning. Furthermore, they called for implementing a PLM system where all documentation could be gathered. Having such systems would have helped immensely as standard components, reusable code, and layouts would speed up the process; during this project, I had to do a lot of networking to find certain assets.

Norrman and Persson (2018) did not see any possibility of creating an exact virtual copy of a production unit due to the difficulties of handling the sensors of the physical world and concluded that the virtual model will always be a simplification of its physical counterpart. I would argue that it must first be decided what level of detail is wanted (see Chapter 2.4.1). Having an exact copy of every sensor might not be what is sought after, as it may increase the effort with diminishing returns. In that sense, the virtual model *should* always be a simplification of the physical production unit. In this thesis, all external sensors were modeled and included in the virtual environment, as it was decided that was an appropriate detail level. Norrman and Persson (2018) further suggest that creating the virtual en-

environment should begin once most electrical systems and mechanical constructions are completed. The argument is to avoid unnecessary work in case of changes in the construction and electrical system. I want to challenge this idea; building a virtual model of the cell is generally more effortless than construction and routing electronics. The project presented in this thesis is a prime example of where the early creation of the virtual environment gave benefits throughout the project. In many instances, the mechanical or electrical design had to be changed after issues arose in the virtual environment, not the other way around.

An aspect that should not be overlooked is the quality increase observed by Schamp et al.(2018) and Zäh et al. (2006). Finetuning does take some effort, but many production units in the industry are *set and forget*; when the machine is commissioned, there is generally little interest in making minor optimizations. With machines running all day around, small inefficiencies add up.

The driving force to develop the method has come from within the companies. Looking at the integrators, FlexLink, Evomatic, and TechTribe have not yet experienced any demands for virtual commissioning from the customers, while Elektroautomatik has felt an increase. TechTribe believes this is a demand they and other integrators need to create by providing the services without the customer asking to let them see the benefits. The overall interest in Virtual Commissioning seems to be rising. Alkureidis' company questionnaire provided data that support that more companies are looking into using Virtual Commissioning; any deeper analysis is difficult to make due to the openness of the questions. The prerequisite evaluation presumes that the surveyee has enough knowledge about the subject to assess the company's readiness making it vulnerable to cognitive bias.

Evomatic, AFRY, Volvo, and Elektroautomation expressed that changing routines on the organizational level was key to including Virtual Commissioning as standard practice. Some crucial changes were parallelizing tasks and completing them earlier, as well as having specialists working solely with the virtual commissioning. Evomatic made a complete reorganization where they changed from a functional organization structure to a project-based divisional organization structure. FlexLink also sees this need but has not yet been able to make the required changes.

Volvo, Evomatic, Elektroautomation, and Rueckert et al. (2020), see considerable safety-related benefits from using the virtual models to validate cell safety, such as paths, stopping distances, and SafeZones.

The educational value of virtual commissioning should not be overlooked. Hofmann et al. (2017) and Fernandez et al. (2019) studied the possibilities of using virtual commissioning during education to allow more students to gain practical knowledge about logistic control systems. While only one student can implement and verify the developed process design in a full-scale physical model at a time, many students can simultaneously use a virtual commissioning tool. Giving the students this ability resulted in a risk reduction of physical damage to the system and a significantly increased student capacity. Both Alfa Laval and Volvo began investigating virtual commissioning with the help of university students; implementing the methods in coursework will ensure that more competence on the subject is making its way out in the industry.

During the literature review and interviews, it became apparent that the term Digital Twin regularly was misused and incorrectly associated with virtual commissioning. Simumatik, AFRY, Volvo, Scania, Flexlink, and Alfa Laval all used the term when they de facto described digital models. Among the virtual commissioning software providers, the digital twin was described similarly by Experior, Visual components, Emulate3D, and Fast-suite. The only software providers with a somewhat correct definition were

Siemens and ABB. A paragraph about Digital Twins was, therefore warranted to decrease confusion even though it is a separate concept not really within the scope of virtual commissioning (except for when a digital twin is virtually commissioned). I would agree that if a majority agreed on another definition for digital twin, that would eventually be the new meaning; what is problematic is that it seemed like every company had *its own* definition, tailored to suit their use.

It is worth bearing in mind that interviewing only companies that have mentioned virtual commissioning may induce survivorship bias as the reflected views are only from those that are successfully using it. Differentiating between companies that have abandoned virtual commissioning and the uncountable companies that have not tried virtual commissioning is a task too great for this thesis, as companies typically do not advertise failed attempts. Instead, the results have to be interpreted that it is not certain that using virtual commissioning will be fruitful in a particular company, but there is a chance of considerable benefits. Seeing that all companies had similar views on benefits and limitations further validates the results. A similar bias may be present in the literature review as it is a greater chance that research with positive results will be published.

Simulation software and interfaces

All of the software presented in this thesis would have been feasible to use for virtual commissioning as they all share similar functionality. Instead, other parameters had to be considered, such as ease of use, competence within the company, and special features, which are hard to quantify. As it was not within the scope to try every simulation software on the market, RobotStudio was selected due to the history of ABB robots at Alfa Laval and my previous experience in the software.

RobotStudio and some of the investigated software had the functionality to retain the link to the CAD model after import. Changes made to the

original CAD model would then synchronize with the virtual environment, effectively eliminating reimport when making small changes. While this seems like a desirable feature at first, the benefits become drawbacks in assemblies of tens of thousands of parts, where added, subtracted, or changed models will make it impossible for the software to keep track of the part correctly. This can ultimately lead to incomplete models or doubts about if the model is up-to-date. Instead, a revision-based workflow was deployed, where all the models were imported at once and not updated until making a new manual import, making sure that all parts and functions were changed and imported as intended.

In discussions with Simumatik, they expressed the need to simplify CAD models to use them with their software as it could be too heavy otherwise. There are no functions within Simumatik for this, and they recommend using another software, Blender, to simplify the models. Because of these performance limitations and the fact that most of Alfa Laval's models would need to be simplified, Simumatik was not considered viable. Emulate3D's CAD Is The Model plugin could be of interest if the mechanical designers are open to using it, although this feature does not save time but rather shift when the time is spent and by whom. Visual Components have some intriguing features such as Plug and Play mechanisms, statistics, and material transportation. They also possess an extensive library that could be used for quick mockups. A drawback they share with many of the presented software is the inability to validate the final robotics code without additional software from which the joint parameters are read and visually represented.

ABB has an edge through their emulated robot controllers, but they need to keep up their software development as other vendors advance rapidly, adding functionality extending far beyond what is needed for virtual commissioning. Lacking in RobotStudio are material flow, easy mechanism creation, and statistics.

Being able to run the program in real-time is especially important in the process industry with many continuous processes. During the performance tests, no significant difference between mid-end laptops to high-end workstations was found, probably because the software is only utilizing a small part of the capacity. While real-time virtual commissioning was still feasible on a small production cell, the frame rate quickly dropped when adding more complexity and features. This might not be an issue as virtual commissioning currently has most of its benefits on the cell level and not the plant level. The frame drops only affect the user experience, not the results. Simulations of material flow and throughput on the plant level do not rely upon smooth visuals as long as the results are valid.

Virtual commissioning of Viking

FactoryTalk Logix Echo was an excellent upgrade to the outdated Logix Emulate as the old software required manual connections and was error-prone, with the program often crashing. Nevertheless, Echo could not simulate Safety signals, which was a significant downside, practically making virtual commissioning unworkable as all safety modules and code had to be manually removed from the PLC program. By extension, the safety removal made the virtual commissioning lose most of its purpose as the program could not be fully trusted to behave as it should in the physical application. A Hardware-in-the-loop configuration would allow the safety modules to remain, but the safety program would still need to be bridged as the OPC-UA signals would not be considered safe. The issue might be resolved if ABB implemented a way to connect similar to the present Siemens module. The connection to Siemens PLCs from RobotStudio does not require OPC-UA and can handle safe signals. Another benefit of the Siemens PLC is that they do not need to exchange axes for virtual axes, and they can simulate safety even with an emulated PLC.

The communication between software was the most troublesome and uncertain. Most of the software felt outdated in terms of functions. The documentation was either absent or too vast to know where to begin. Setting up the OPC-UA server and connecting it to InTouch and Studio 5000 was characterized by brute-forcing different settings until the communication started working. Many of the issues encountered during this first project will not become issues in the next project as the knowledge now exists within the company. For this reason, it is hard to estimate the actual effort and benefit of virtual commissioning under normal circumstances.

A partial virtual commissioning was performed where the PLC was simulated as tasks within RobotStudio. This setup was much simpler but could of course not validate the PLC or HMI, only the robotics and motion within the virtual environment. The results were satisfactory, and the method could be an option even when the end goal is full virtual commissioning, as issues found during the partial virtual commissioning would not occupy extra resources such as PLC programmers.

A considerable challenge was that the concept of virtual commissioning was new to everyone within the project. This meant that there was minimal experience to build on. Even with the theoretical pillars in place, it is a great work to apply them practically, at least for the first time. What has emerged during interviews with people in the project is that virtual commissioning is fundamentally sound, but we need to work out a methodology that makes it work for Alfa Laval. As the project progressed, it became clear that fully benefiting from virtual commissioning would require a change in the project structure and planning where the virtual commissioning should be a deliverable from the beginning. Among other things, it would be desirable to have closer cooperation between programmers and to have the programming done earlier. The commissioning would have been quicker had there been a standard regarding setup, I/O-addresses, documentation, and connection for the robot, PLC, and HMI in place.

Extended use of virtual models

To utilize the full potential of virtual models, they have to be used continuously throughout the project. By finding additional usage for the models, the cost to develop the virtual models diminishes. Projects typically begin with a brainstorming session. At this time, there is nothing but a blank canvas. Using detailed mockups at the beginning of the design phase can be unnecessarily time-consuming and hinder spontaneous and creative ideas. There was never an explicit request from Alfa Laval to use the models or software for any other purpose than the virtual commissioning. However, it quickly became apparent that making quick mockups using the virtual models in RobotStudio was of immense value. Most CAD software is strictly parametric and requires well-defined objects and dimensions, making it cumbersome to perform changes. While kinematics can be visualized using joints and constraints, it often takes much time to set up correctly. The benefit of making the mockups in RobotStudio was that mechanisms and I/O signals were already set up and ready for the virtual commissioning when the final layout was settled. Most of the robot code was also complete as it had been shaped throughout the project.

Virtual models open up a whole new way of working where operators can be included much earlier and provide feedback that can save both time and money. As Volvo noted, operators that felt they had been part of the development of the production unit were generally more positive about the changes. Alarm handling is another area where operators' insight can prove valuable. Obscure and vague alarm codes are not unheard of, where maintenance personnel has to be summoned to solve issues of such kinds the operator could have solved themselves had they been guided through the troubleshooting.

Additive manufacturing has been an essential part throughout the project due to its quick turnaround and low cost. The ability to test new ideas on demand accelerated the idea generation process. With the plastic parts as

the weakest link, AM avoided disastrous consequences on multiple occasions when the portal or robots crashed into other objects. These crashes could have been avoided altogether by the use of virtual commissioning, but with the VC not being part of the project, the assembly began before the vFAT. Using generative design for the tool mount might have been a bit excessive, but it resulted in a much more stable mount with less resonance and the design time was not that much longer.

6. Conclusions and future work

This thesis contributes to the research of virtual commissioning and lowers the bar for companies to use virtual commissioning by providing a practical example of how it can be implemented and how the virtual assets can be used continuously. At the same time, it highlights limitations to aid the decision to use virtual commissioning.

The literature study outlines various benefits, limitations, and resource requirements. An increase in virtual commissionings is anticipated due to its ability to save time and resources. The main limitation is the lack of standardization and lengthy setup times of the virtual environment and its interfaces.

A virtual commissioning model was created using Allen Bradley, Robot-Studio, and Aveva software. The model was able to test and verify the PLC code, robot code, HMI, and mechanical movement but was severely limited due to Rockwell's incapability to emulate a safety controller. The use of Allen Bradley PLCs is not a viable option for virtual commissioning until they have solved the emulation of safety controllers.

The use of virtual commissioning is becoming more popular, but its viability in the industry relies on software connectivity. Sharing best practices between companies and working tightly with software vendors to provide better connectivity could expedite the process of making virtual commissioning standard procedure.

Further uses of virtual models were evaluated, and it was found that the initial simulations were powerful tools for visualization, cycle time estimation, and concept feasibility. Validation against the physical system showed temporal and spatial coherence between the virtual environment and the physical application, indicating that the results from the virtual commissioning can be trusted.

For future work, it would be of interest to analyze the benefits from more industry-as-laboratory trials, as few studies observe the best practice within projects from start to finish. This would assist in the alignment between academia and industrial practice. Today, there is a skewed distribution towards purely academic studies that easily overlook the organizational parts of virtual commissioning. Organizational obstacles have been outlined, and it would be of value to evaluate which structure would be best for these kinds of projects.

Another topic that has not been studied thoroughly is virtual commissioning's place on the plant level. It is not entirely clear what would be commissioned in such a setup and if it should be called virtual commissioning at all.

Virtual Reality has been around for a while but Augmented Reality solutions are lagging. Augmented Reality could potentially be used to validate the location of the physical assets after the virtual commissioning by visualizing a copy of the geometry on-site, using fiducials to get the proper positioning and scale.

7. Recommendations for Alfa Laval

This chapter is intended for Alfa Laval decision-makers and personnel. With the results from this thesis, I present recommendations for Alfa Laval in their continued work with virtual commissioning and virtual models.

Virtual commissioning

Performing virtual commissioning requires deep and broad knowledge within the project group as it ties together many different fields. If Alfa Laval wants to continue using virtual commissioning, it should be included in the project planning from the beginning. For the time being, it is not viable to do virtual commissioning using a virtual Allen-Bradley PLC because of the inability to simulate safety. Hardware-in-the-loop would be a workaround for this issue and should be investigated. This may preferably be implemented in a lab environment so that future projects have a good starting point with pre-programmed interface connections between HMI, robot, and PLC, lowering the initial effort. Furthermore, it might be of interest to consider Siemens PLCs for comparison as they have better connectivity and support for safety simulation. While RobotStudio is sufficient to perform virtual commissioning, I recommend letting student resources benchmark other software such as Visual Components or Emulate3D for virtual commissioning and plant simulations.

Use of virtual models

I recommend using virtual models early in the process for rapid LoFi prototyping and as a basis for planning workshops and aiding discussions. This will increase the speed at which new technologies can be tested, and TRLs raised. Developing the virtual models further lets us increase predictability by being able to emulate future processes before they are tested physically. Working with the simulation model from the beginning aligns with the Lean and Agile development process, which is an essential part of Alfa Laval's strategy and DNA. The operations development team in Lund

works to serve Alfa Laval globally. Running projects at other sites will be more accessible by implementing tools such as rapid prototyping, virtual reality, and virtual commissioning. Moreover, the cost of travel will be reduced as the project can stay virtual longer.

Standardization and guidelines

Much of the time spent during the practical application of this thesis was on making different interfaces work. An audit of current standards is called for, and a standard for robotics should be drafted and address topics such as communication, programming, and motion. Having a robot standard will speed up future robot projects and ensure they are easy for continuous improvement and product changes. If Alfa Laval wants to continue using the virtual tools, they should be integrated into the current project management structure. I recommend that Alfa Laval compose guidelines on how to work with our virtual models and how Request For Quotations should be formulated to ensure that the virtual models are part of the delivery. Failing to acquire virtual models used in projects with third parties could incur lifelong costs (see the interview with Volvo).

I advise Alfa Laval to continue using ABB robots because of their superior development platforms and widespread use within the company. Focusing on a single robot brand makes it possible to standardize how programs are made and how the robots are set up. This will increase program readability, shorten the installation time, streamline future optimizations, and make it easier to support other sites. In the past, third parties have made robot programming, which is costly and limits optimization and further development. With the increase in robotics projects, I would recommend Alfa Laval to consider having more of this knowledge in-house, as there is enough work to sustain multiple programmers today.

Implementing PLM for production systems

The recommendation for implementing a PLM system was made by Persson and Norrman (2018) during their thesis project at Alfa Laval in 2018. However, it is worth reiterating as there is still a general lack of data handling capability on the automation side of the organization. Currently, the virtual models reside on various servers, personal hard drives, or in teams-groups, making it hard to overview, especially for future users. Optimally the database should be searchable for keywords, and all information related to a specific machine should be accessible in one place. A link to the database location could be put as physical QR codes on the machines or parts.

References

- ABB. (2021). Operating manual Robotstudio.
- ABB. (2022a). Product Specification IRB 360.
- ABB. (2022b). Product specification IRB 8700.
- Albo, A., & Falkman, P. (2020). A standardization approach to virtual commissioning strategies in complex production environments. *Procedia Manufacturing*, 51, 1251-1258.
- Älegård. (2017). *Virtual commissioning of smart factory* Chalmers.
- Alemanni, M., Destefanis, F., & Vezzetti, E. (2011). Model-based definition design in the product lifecycle management scenario. *The international journal of advanced manufacturing technology*, 52(1), 1-14.
<https://link.springer.com/article/10.1007/s00170-010-2699-y>
- Alfa Laval. (2022). *History of Alfa Laval*.
- Alkureidi, H. (2020). *Investigations on Industrial Practice of Virtual Commissioning* University of Skövde. DiVA.
<http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-18658>
- Antonsen, T. M. (2020). *PLC Controls with Structured Text (ST), V3: IEC 61131-3 and best practice ST programming*. BoD-Books on Demand.
- Auinger, F., Vorderwinkler, M., & Buchtela, G. (1999). Interface driven domain-independent modeling architecture for “soft-commissioning” and “reality in the loop”. Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1,
- Ayani, M., Ganebäck, M., & Ng, A. H. (2018). Digital Twin: Applying emulation for machine reconditioning. *Procedia CIRP*, 72, 243-248.
- Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Gutierrez, D., Negri, E., & Fumagalli, L. (2021). A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Springer*, 15(3), 397-412.

- Berg, L. P., & Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. *Virtual reality*, 21(1), 1-17.
- Biggs, G., & MacDonald, B. (2003). A survey of robot programming systems. Proceedings of the Australasian conference on robotics and automation,
- Bijnens, J., & Cheshire, D. (2019). The current state of model based definition. *Computer-Aided Design and Applications*, 16(2), 308-317.
- Bolmsjö, G. (2006). *Industriell robotteknik*. Studentlitteratur AB.
- Bolton, W. (2015). *Programmable logic controllers*. Newnes.
- Bottani, E., & Vignali, G. (2019). Augmented reality technology in the manufacturing industry: A review of the last decade. *IISE Transactions*, 51(3), 284-310.
- Boyer, S. A. (2009). *SCADA: supervisory control and data acquisition*. International Society of Automation.
- Briard, T., Segonds, F., & Zamariola, N. (2020). G-DfAM: a methodological proposal of generative design for additive manufacturing in the automotive industry. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 14(3), 875-886.
- Bysko, S., Krystek, J., & Bysko, S. (2020). Automotive paint shop 4.0. *Computers & Industrial Engineering*, 139, 105546.
- Carlsson, H. (2012). *Reliable Virtual Commissioning* Chalmers University of Technology.
- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia tools and applications*, 51(1), 341-377.
- Chow, P., & Walker, M. (2019). Risk mitigation strategies for emergency power upgrades in critical facilities. 2019 IEEE/IAS 55th Industrial and Commercial Power Systems Technical Conference (I&CPS),
- Commission, I. E. (2003). IEC 61511-1.
- Commission, I. E. (2013). IEC 61131-3.

- Dahl, M., Bengtsson, K., Bergagård, P., Fabian, M., & Falkman, P. (2016). Integrated virtual preparation and commissioning: supporting formal methods during automation systems development. *IFAC-PapersOnLine*, 49(12), 1939-1944.
- Diegel, O., Nordin, A., & Motte, D. (2019). *A Practical Guide to Design for Additive Manufacturing*. Springer.
- Ebrahim, M. A.-B. (2015). 3D laser scanners' techniques overview. *Int J Sci Res*, 4(10), 323-331.
- Elektroautomatik. (2022). Retrieved 19 april 2022
- Encarnacao, J. L., Lindner, R., & Schlechtendahl, E. G. (2012). *Computer aided design: fundamentals and system architectures*. Springer Science & Business Media.
- Fernández, I. A., Eguía, M. A., & Echeverría, L. E. (2019). Virtual commissioning of a robotic cell: an educational case study. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),
- Gebhardt, A. (2011). Understanding additive manufacturing. *ScienceDirect*.
- Grievess, M. W. (2019). Virtually intelligent product systems: digital and physical twins. In *Virtually Intelligent Product Systems: Digital and Physical Twins* (Vol. 256). American Institute of Aeronautics and Astronautics.
- Havlíček, O. (2019). Integrace rozšířené reality do postupu virtuálního uvádění do provozu [Seamless integration of augmented reality into digital workflow of virtual commissioning] Thesis. <https://otik.uk.zcu.cz/handle/11025/39270>
- Helle, R. H., & Lemu, H. G. (2021). A case study on use of 3D scanning for reverse engineering and quality control. *Materials Today: Proceedings*, 45, 5255-5262. <https://otik.uk.zcu.cz/handle/11025/39270>
- Heynicke, R., Krush, D., Cammin, C., Scholl, G., Kaercher, B., Ritter, J., Gaggero, P., & Rentschler, M. (2018). IO-Link Wireless enhanced factory automation communication for

- Industry 4.0 applications. *Journal of Sensors and Sensor Systems*, 7(1), 131-142.
- Hull, C. (1986). *Apparatus for Production of ThreeDimensional Objects of Stereolithography US4575330A*). U. S. o. America.
- IFA. (2020). *Software-Assistent SISTEMA: Safety Integrity Software Tool for the Evaluation of Machine Applications*.
- Ikram, W., & Thornhill, N. F. (2010). Wireless communication in process automation: A survey of opportunities, requirements, concerns and challenges. UKACC International Conference on Control 2010,
- ISO 8373:2012(en). (2012). International Organization for Standardization.
- ISO 9283:1998(en). (1998). International Organization for Standardization.
- ISO 13849-1. (2015). International Organization for Standardization.
- ISO 23247-1:2021(en). (2021). International Organization for Standardization.
- Jakob, M., & Nilsson, P. (2018). Virtual Commissioning and construction of a digital twin for Smarta Fabriker.
- Johnsson, C. (2021). Whitepaper on Smart Manufacturing. *International Organization for Standardization*.
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016-1022.
- Lechler, T., Fischer, E., Metzner, M., Mayr, A., & Franke, J. (2019). Virtual Commissioning–Scientific review and exploratory use cases in advanced production systems. *Procedia CIRP*, 81, 1125-1130.
- Lee, C. G., & Park, S. C. (2014). Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, 1(3), 213-222.
- Lidell, A. (2021). *A Deep Dive into the Current and Future Conditions for Virtual Commissioning of Production Lines* University of Skövde. DiVA.

- Lilja, E., & Magnusson, J. (2021). *Virtual Commissioning of a Projection Based Support System* Chalmers.
- Lin, Z., & Pearson, S. (2013). An inside look at industrial Ethernet communication protocols. *Texas Instruments, White Paper*.
- Liu, M., Fang, S., Dong, H., & Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems, 58*, 346-361.
- Manapat, J. Z., Chen, Q., Ye, P., & Advincula, R. C. (2017). 3D printing of polymer nanocomposites via stereolithography. *Macromolecular Materials and Engineering, 302*(9), 1600553.
- Mankins, J. C. (1995). Technology readiness levels. *White Paper, April, 6*(1995), 1995.
- McGregor, I. (2002). The relationship between simulation and emulation. Proceedings of the Winter Simulation Conference, <https://www.researchgate.net/publication/3997749>
- McKnight, M. (2017). Generative Design: What it is? How is it being used? Why it's a game changer. *KnE Engineering*, 176-181.
- Mehta, B. R., & Reddy, Y. J. (2014). *Industrial process automation systems: design and implementation*. Butterworth-Heinemann.
- Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Team, G. B., Pickett, J. P., Hoiberg, D., Clancy, D., & Norvig, P. (2011). Quantitative analysis of culture using millions of digitized books. *science, 331*(6014), 176-182.
- Mueller, B. (2012). Additive manufacturing technologies–Rapid prototyping to direct digital manufacturing. *Assembly Automation*.
- Muller, G. (2011). Industry-as-laboratory applied in practice: The boderc project. *University of South-Eastern Norway*.
- Negri, E., Pandhare, V., Cattaneo, L., Singh, J., Macchi, M., & Lee, J. (2021). Field-synchronized Digital Twin framework for production scheduling with uncertainty. *Journal of Intelligent Manufacturing, 32*(4), 1207-1228.
- Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
- Oppelt, M., & Urbas, L. (2014). Integrated virtual commissioning an essential activity in the automation engineering process: From

- virtual commissioning to simulation supported engineering. IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society,
- Patel, R., Lehtola, M., & Lemarinier, P. (2014). Integration of an ABB industrial robot with a MRDS enabled mobile robot platform and experiments for track motion applications. ISR/Robotik 2014; 41st International Symposium on Robotics,
- Peiris, S., & De Silva, N. (2021). RE-engineered factory acceptance testing under the new normal. *Built Environment Project and Asset Management, Emerald Publishing Limited.*
- Persson, J., & Norrman, J. (2018). *Virtual Production Line*, Lund University. LUP Student Papers.
- Potts, C. (1993). Software-engineering research revisited. *IEEE software*, 10(5), 19-28.
- Puntel-Schmidt, P., & Fay, A. (2015). Levels of detail and appropriate model types for virtual commissioning in manufacturing engineering. *IFAC-PapersOnLine*, 48(1), 922-927.
- Rahman, A. A., & Mohamad, N. R. (2016). Software-in-the-loop technique: an approach to support reconfiguration of manufacturing system. *J Eng Appl Sci*, 11, 9789-9795.
- Ramanathan, R. (2014). The IEC 61131-3 programming languages features for industrial control systems. *2014 world automation congress (wac)*, 598-603.
- Rankin, D. J., & Jiang, J. (2011). A hardware-in-the-loop simulation platform for the verification and validation of safety control systems. *IEEE Transactions on nuclear science*, 58(2), 468-478.
- Reinhart, G., & Wünsch, G. (2007). Economic application of virtual commissioning to mechatronic production systems. *Production engineering*, 1(4), 371-379.
- Rueckert, P., Muenkewarf, S., & Tracht, K. (2020). Human-in-the-loop simulation for virtual commissioning of human-robot-collaboration. *Procedia CIRP*, 88, 229-233.

- Schamp, M., Hoedt, S., Claeys, A., Aghezzaf, E.-H., & Cottyn, J. (2018). Impact of a virtual twin on commissioning time and quality. *IFAC-PapersOnLine*, 51(11), 1047-1052.
- Schludermann, H., Kirchmair, T., & Vorderwinkler, M. (2000). Soft-commissioning: hardware-in-the-loop-based verification of controller software. 2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165),
- Segovia, V. R., & Theorin, A. (2012). History of Control History of PLC and DCS. *University Lund*.
- Sherman, W. R., & Craig, A. B. (2018). *Understanding Virtual Reality: Interface, Application, and Design*. Morgan Kaufmann.
- Signalhead. (2007). Relay room. In Relay room.jpg.
https://commons.wikimedia.org/wiki/File:Relay_room.jpg
- Smith, D. J., & Simpson, K. G. (2020). *The Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety: IEC 61508 (2010 Edition), IEC 61511 (2015 Edition) and Related Guidance*. Butterworth-Heinemann.
- Spectrum, I. (2021). *Top Programming Languages 2021*.
- Stark, J. (2020). Product Lifecycle Management (PLM). In *Product lifecycle management (Volume 1)* (pp. 1-33). Springer.
- Stark, R. (2022a). Major Technology 1: Computer Aided Design—CAD. In *Virtual Product Creation in Industry* (pp. 113-138). Springer.
- Stark, R. (2022b). Major Technology 3: CAPP, CAM and NC Technology. In *Virtual Product Creation in Industry* (pp. 155-183). Springer.
- Stark, R. (2022c). Major Technology 4: Computer Aided Engineering—CAE. In *Virtual Product Creation in Industry* (pp. 185-221). Springer.
- Stark, R. (2022d). Major Technology 7: Virtual Reality—VR. In *Virtual Product Creation in Industry* (pp. 305-326). Springer.
- Stark, R. (2022e). Major Technology 8: Augmented Reality—AR. In *Virtual Product Creation in Industry* (pp. 327-351). Springer.
- The Swedish Work Environment Authority. (2014). *Belastning i arbete [Stress during work]*.

- Tiegelkamp, M., & John, K.-H. (2010). *IEC 61131-3: Programming industrial automation systems*. Springer.
- Tofail, S. A., Koumoulos, E. P., Bandyopadhyay, A., Bose, S., O'Donoghue, L., & Charitidis, C. (2018). Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials today*, 21(1), 22-37.
- Visual Components*. (2022).
- Whittemore, R., & Knafl, K. (2005). The integrative review: updated methodology. *Journal of advanced nursing*, 52(5), 546-553.
- Wilson, E. B. (1990). *An introduction to scientific research*. Courier Corporation.
- Wright, L., & Davidson, S. (2020). How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1), 1-13.
- Zäh, M. F., Wünsch, G., Hensel, T., & Lindworsky, A. (2006). Nutzen der virtuellen inbetriebnahme: Ein experiment. [The use of Virtual Commissioning: An Experiment]. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 101(10), 595-599.

Appendix A. Interviews

Here the interviews with the company representatives are transcribed non-verbatim. The interviews were done in Swedish. The interviewee consented to have the interview recorded and the information used in this thesis. Text in italics is questions from the thesis author.

FlexLink

Kan du först presentera dig och ditt företag?

Jag heter Danielle Casani och jag är chef för digital transformation på FlexLink, det är ett ganska brett begrepp, jag har jobbat 20 år på Ericsson och har främst jobbat med projektledning i början och försäljning i slutet. Men alltid någon slags transformation och integrationsprojekt och de senaste 10 åren har jag jobbat i automotivesektorn. Först som ansvarig för Volvo Cloud och även det har varit en resa för Volvo Cars att kunna leverera saker as a service som man säger och 2018 blev jag konsult och började jobba med Volvo Trucks och digitala tjänster inom B2B. I september började jag på FlexLink som head of digital transformation. FlexLink är ett väldigt produktfokuserat företag, men vi vet att pengar läggs allt mindre på saker utanför den digitala världen. Då började jag som ansvarig för digitaliseringen end to end, både effektivisering av interna produkter och skapande av nya produkter för att möjliggöra nya revenue streams. Det är kort om vad jag gör.

Flexlink som företag tillverkar conveyor belts, vi har en bred produktkatalog och har börjat jobba i end of line-delen där vi har två palatisers med robotar. När det gäller den digitala världen så är än så länge, tycker jag, ganska omogen. Jag tror att det mest beror på att man fokuserat mycket på sina produkter och har sålt väldigt bra. Nu börjar vi till exempel titta på att sälja mjukvarulicenser och all digital utveckling som man har gjort har varit väldigt intern. Vi har ett verktyg som heter Flexlink Design Tool som är en desktopapplikation baserat på Visual Components och det är vad våra säljare använder för att bygga layouts baserat på kundspecifikation och definiera en Bill of Material. Det jag vill göra istället är att använda makten och möjligheten med ett sånt verktyg för att göra mycket mer än så. Jag vill kunna sälja engineering services till våra kunder, jag vill kunna använda de layout vi skapar för simulering och virtual commissioning, för prestanda monitoring och asset monitoring. Men det kräver en infrastruktur där vi har sensorer och samlar data. Det är det som är mitt uppdrag.

Det är dit flexlink vill? Eller är det din vision?

Dem har tagit in mig just för att jag har den vision, så jag har ett mandat från ledningsgruppen att utveckla denna tjänst. Sen kan naturligtvis inte min vision levereras imorgon.

Min slutliga version är plattform där man har en portal för kunderna där de kommer åt sina produkter och tjänster, en marketplace där kunderna kan köpa nya produkter och tjänster. En partner portal där vi har partners som lägger upp sina produkter och en developer portal där developers och andra företag kan lägga upp sina tjänster. Allt detta hamnar i marketplace som en tillgång för våra kunder och säljare. Flexlink tillhör en grupp som heter Coesia den gruppen har 21 företag som tillverkar olika maskiner som vi säljer till olika industrier och fabriker. Men om du tänker hur du kopplar dessa maskiner med varandra så är det med FlexLink connectors, så FlexLink har en naturlig roll som integratör i hela systemet. Så jag ser det som att FlexLink ska äga platformen.

Så då är tanken med de virtuella driftsättningarna att det är en tjänst som FlexLink skulle leverera?

Exakt, att det är en tjänst som FlexLink ska leverera betyder inte nödvändigtvis att det är något vi får betalt för av våra kunder, men jag vill tjänstifera det så att det mer blir en vanlig del av arbetssättet.

Så när ni drifttar så har ni fortfarande en virtuel modell att testa mot?

Exakt, för att jag se ändå en värld i framtiden när en del av våra kunder är tillräckligt mogna och tillräckligt självkörande, kanske stora kunder med många fabriker. Som då kan komma åt vår portal och hantera sina tjänster själva. Men vi kommer fortfarande ha en mängd små kunder som kanske behöver 10 meter conveyors, thats it. Då är det mer våra säljare som gör arbetet och det blir svårt att monitsera för då är vi där för att sälja mer. Men om kunden säger att jag behöver inte ha stöd, då betalar dem för att komma åt tjänsterna.

Så flexlink skulle gå mer mot tjänster än produkter för att komplettera?

Min bild, och den värld vi lever i, som jag sett i bussinesvärlden sedan många år tillbaka. Den investeringsmängd och kapacitet som våra kunder har är konstant. Men fördelningen och hur de spenderar dessa pengar går mer och mer till fördel för tjänster än för produkter. Anledningen till det är att de har lärt sig på grund av att någon smart i företaget tänkt på det. Att de kan tjäna mer pengar med de prylar de har istället för att köpa fler prylar. Det har hänt i B2B, i automitive till exempel. Lastbilar, kunderna vill inte ha fler lastbilar, de vill ha fler saker att transportera. Om en lastbil är halvtom är det viktigare att hitta någon som fyller upp resten istället för att köpa en ny lastbil. Så företag som volvo trucks och scania är väldigt aktiva i att attrahera sina kunder med digitala tjänster. Jag ser att det blir mer och mer vanligt i annan

industri med. Kurvan där blir inte lika brant. Jag har träffat många kunder och deras digitalmognad är ganska låg. Dem börjar fatta att det finns något som man borde kunna göra men pratar man lite data och digitaltjänster då tittar dem på dig som om du kom från framtiden.

Digitala tool ni använder, det var internt?

Just nu är det mest internt, det finns vissa kunder som använder det, men det är mest internt för våra säljare. Vi har använt verktyget fem till sju år, de började 2014.

Vet du hur ni gjorde tidigare, innan de digitala verktygen?

Nej jag vet inte det, men jag kan tänka mig, för när man pratar med andra företag i gruppen använder de inte alls grafiska 3d metodiker. De använder väldigt mycket 2d ritningar, de använder CAD naturligtvis men mest 2D-ritningar, många har inte så många viktiga vertikala constraints. Även de som har 3D-maskiner bygger sina libraries med maskiner de redan sålt. Så om en ny kund kommer och frågar så tittar de om de har ritat något liknande någon gång.

Så de går mycket på gammal erfarenhet?

Precis, och det var så på flexlink även innan de digitala verktygen.

Vad tycker du är de största fördelarna med det här designverktyget?

Den största fördelen är att man snabbar på försäljning. Man har väldigt lätt att ta fram något som kunden kan se och känna på. Sen att det är modulärt och att det går att rita fort och att man kan skapa BOM-listor faller alltid tillbaka till att Time-to-market är kortare. Den ger också möjlighet att jobba med simulering och hjälpa våra kunder optimera från början och det är också en fördel. Men det har vi svårt att monitsera. Jag tror det är ett värde vi ger bort till våra kunder dagligen. Men möjlighet att ge en kund den bästa layouten från början är viktigt.

Jag vet att Visual Components har mycket sätt att testa olika varianter och optimeringar, som jag förstått det är ert verktyg relativt begränsat i förhållande till Visual Components?

Ja, vi har bara vissa features från Visual components. Vi har en begränsad version av essentials. Vi hade ett möte med Visual Components igår där jag frågade "vad är det ni har som vi inte köpt", för att den här biten som ett internverktyg så har man inte så många krav. Du har begränsat med investering och så vidare. Men om man vill exponera något liknande till kunder så måste man först ha ett värde i det man gör. Vi kanske skulle kunna ge samma verktyg idag, gratis, men med fler features får du betala

så det blir någon premium variant. Men när denna version byggdes så var det bara utifrån att säljarna skulle använda verktyget.

Men det var längesedan vi började samarbeta med VC, nu har dem många nya saker, simuleringar som går utanför fabriken, den här Process modellering, och AGVer. Det kan vara intressant för våra kunder också. Där börjar man tänka mer digitalt. För VC som idag är en supplier skulle kunna bli en partner, eller en konkurrent, beroende på hur vi sätter upp vår struktur. Idén är att försöka expandera detta för att göra bättre affärer.

Tycker du det ligger i linje med FlexLinks verksamhetsområde när ni håller på med conveyors. Att erbjuda tjänster om driftsättning. Det är väl ett helt annat område?

Jag förstår vad du menar, men det finns aspekter i det vi gör redan idag där virtual commissioning är intressant, vi har ju kontrollsystem för våra conveyors och robotar som behöver programmeras. Men som jag sa, att sälja produkter kommer inte att hålla oss levande. Det tror jag alla börjar inse inom flexlink, de senaste 40 åren har vi varit stora i världen men det kommer inte finnas oändligt med behov för conveyors. Då tycker jag, och ledningen, att flexlink ska gå över till att sälja solutions, och inte bara conveyors. Vi har en naturlig fördel, för vi sitter med hela layouten, vi ser inte på enstaka maskiner.

Om du då ser på din vision på kort sikt, vad är närmast?

För mig närmast är att bygga, det finns massor aktiviteter jag försöker att få fram samtidigt, men också för att jag började för två månader sedan. Jag pratar mycket om mina tidigare erfarenheter. Den första är att börja få plattformstanken på gång. Vi har massor verktyg och portaler, vi måste se till att allt detta integreras. Att ge våra kunder och säljare ett naturligt flöde att komma in i plattformen. Vi vill att alla företagen i grupper har sina maskiner i vårt design-tool, och där ska inte vi sitta och rita maskinerna utan vi ger instruktioner och dem som ritat in sina maskiner hur dem vill bara de följer våra APIer. Sen samtidigt bygga en datalake, just nu har vi bara "datapuddles". Jag skulle vilja bygga en integrerad flexlinkdatamodell så att alla produkter och alla aktiviteter pumpas in data i vår datalake. Data som vi samlar från installationer och tester inför leveranser. Det kan vara allt från hastigheter på en motor i en viss fabrik vid en viss tidpunkt. Alla beslut som vår kontroll gör. Idag har vi många splitter och mergers som tar beslut i real time och sen slängs.

Så det är lite digital twin över det?

Både digital twin för produkten men också för organisationer. Vilken layout ritat våra säljare mest, vilka produkter vill kunderna ha mest, på vilka sidor tittar kunderna mest? Jag vill ha olika öar med kundnära data, CRM data, produktdata. Så att vi kan börja korsas den här datan.

Men första början skulle kunna vara att samla produktdata för att bygga en digital twin. Vi har vissa prototyper för de men de är inte industrialiserade.

Vad ser du som den största utmaningen för visionen på kort sikt?

En av de största problemen är att få HR att förstå vilka profiler som behövs för att implementera detta. Jag försökte nyanställa en tjänstutvecklare. För mig är en tjänstutvecklare någon som förstår kundbehovet, hur affärsmodellen kan byggas kring det. Hur man tänker Proof of Concept. Som har en minimal viable produkt, och hur man expanderar den här produkten i en roadmap. Och den här rollen finns inte på Flexlink-kartan. Jag kunde inte ens förklara för HR vad vi ville. De övesatte och satte ut anonsen för en projektledare. Min värsta mardröm är att jag i Januari måste sätta ut en anons för en dataarkitekt. Hur förklarar jag vad en dataarkitekt gör för någon som är van att anställa mekaniska ingenjörer.

Kan du inte anställa själv då?

Jo, jag kan skriva annonsen och det jag vill ha, men sen måste det matchas med de roller som finns i FlexLink, för den som jag anställer måste ha en lön. För om jag behöver ge personen 50000kr i månaden så kommer man tro att det är en senior supervisor när det bara är en ingenjör som kan data.

Men när det handlar om helt nya roller så måste man väl också sätta nya löner som inte är relaterat till andra?

Jo och det är därför mitt team heter digital transformation, jag ser att det ingår i mitt ansvar att utbilda organisationen i det som måste göras. Den andra utmaningen är just att få hela företaget on board. Att utveckla en teknisk lösning är väldigt lätt. Allt går att göra idag. Det finns Proof of Concept på hur vi flyttar data wireless. Varför behöver vi 2021 ett proof of concept på hur vi flyttar data wireless? Det du behöver är en Proof of Concept på den data du samlar är det som kunden behöver. Kan du fånga något värde med den datan? Om våra conveyors går sönder var tionde år? Du kan inte ha preventive maintenance på det vanliga sättet, det är ingen kund som är intresserad att betala för ett meddelande varje månad om att deras conveyer går sönder efter 9 år och 10 månader. Det funkar inte, så du måste gå ut och prata med kunder och förstå vilken deras största huvudvärk är. Är det för att något allting pajar? Du använder datan inte för att hålla reda på när något att gå sönder, men du kan titta före och efter maintenance och säga till kunden "titta här, det hände ingenting, det är något annat som är fel". Jag startar varenda möte med min visionbild, bara för att påminna folk om att "det är dit vi ska"

Är det någon specifik grupp inom FlexLink du känner är mer motståndskraftig mot den här förändringen?

Inte aktivt, men de jag tror kommer vara svårast att vända är säljbolag. De är idag byggda och utbildade för att sälja saker som går att se. De säljer en telefon, det är ingen som skulle kunna sälja en app, för att ge dig ett exempel. Det är det som är utmaningen. Det jag ser att vi måste vända är att övertyga säljarna att även om kunderna köper saker själv i en marketplace innebär det inte att de är en dålig säljare. De får en del av den försäljningen också. Vi ska bygga en incentivemodell så att de har incitament till att sälja mjukvarulicenser eller abonnemang eller vad som helst vi kommer sälja.

Har ni märkt att det är enklare att få kunder när ni nu kan visa de här virtuella modellerna?

Ja ja, kunderna älskar sen vi tog in de här modellerna. Vi har kunder som frågar, ”kan ni hjälpa oss att förbättra vår nuvarande installation”. Där är ett ställe där vi använder digital twin ganska ofta. Ett problem är att det skalar inte upp, vi gör det för enstaka kunder. En av mina mål är att försöka automatisera den här processen så pass mycket att vi kan investera manntimmar bara på det som är värde, det vill säga analysen. Inte att mata in data manuellt. Våra kunder vill se, dem älskar de här 3D-filmerna, ”det här är hur din fabrik ser ut idag, om du gör den här förändringen så kommer din fabrik se ut såhär” och kunderna älskar det, jag förstår inte varför vi ger iväg det gratis.

Många av de företag jag kontaktat har inte riktigt förstått vad jag var ute efter. Tror du att industrin generellt har en låg förståelse för de här begreppen om virtuell driftsättning och digitala tvillingar?

Även jag, jag måste säga att för två veckor sedan hade jag aldrig hört begreppet virtuell driftsättning, bara för att jag kom från en annan värld. Men det räckte att jag googlade för att förstå att det här är något vi måste göra.

TechTribe

Kan du först presentera dig och ditt företag?

Vilka lösningar inom området (virtuella modeller, virtual commissioning, digital model/shadow/twin) erbjuder ni?

TechTribe erbjuder digitala modeller, flödessimulering och virtuell idrifttagning med Emulate3D som verktyg. PLCn kan vara antingen virtuell eller fysisk, och om det är en Rockwell PLC kan den köra programmet snabbare än realtid.

Vad ser ni som en utmaning inom den närmsta tiden inom detta område?

Att mer aktivt börja att använda detta i våra egna projekt utan att behöva ändra för mycket på vår projektmodell, se även presentationen. En av utmaningarna är att hitta rätt nivå på den digitala modellen, så att det går att genomföra en virtuell idrifttagning så effektivt som möjligt. Vidare behöver vi försöka få till att vi påbörjar den digitala modellen tidigare i vår process (redan när vi säljer en lösning) och att då kunna återanvända lösningen på ett effektivt sätt senare i projektet. Idag är det inte optimalt när vi importerar en CAD-modell till Emulate3D och sedan uppdaterar den. Vi behöver göra om en del arbete efter uppdateringen. En annan stor utmaning är att det inom vår bransch inte efterfrågas eller används digitala modeller idag. Detta behov kommer vi behöva vara med och skapa genom att visa på fördelarna genom att vi använder det i våra egna projekt.

Vad har ni för framtidsvision på kort sikt på detta område?

TechTribes korta vision är att vi kommer att börja använda digitala modeller och virtuella idrifttagningar i egna projekt för att kunna säkerställa vår leverans och även bli mycket mer effektiva. Resultatet av dessa projekt kommer vi sedan att använda i vår marknadsföring av Emulate3D. Vidare ser vi en stor potential att nyttja den digitala modellen tillsammans med VR vid operatörsutbildning.

Vad har ni för framtidsvision på lång sikt på detta område?

TechTribes långa vision är att marknaden kommer att efterfråga digitala tvillingar och att vi kan erbjuda detta till marknaden. Både i våra egna projekt och att hjälpa marknaden att själva skapa digitala tvillingar genom att erbjuda tjänster inom detta område.

AFRY

Kan du först presentera dig och ditt företag?

Jag jobbar på AFRY som utvecklare inom vårt koncept som heter RDT, som står för real digital twin. Vi har precis gått in i en nystartad organisation som heter AFRY X som satsar på digitala lösningar. Vårt team blev en del av det med vår produkt inom digitala tvillingar. Vi är åtta som jobbar med det just nu, förra året var det ett gäng som gjorde exjobb som anställdes efter det. Jag jobbar mycket med vårt bibliotek och bygger upp komponenter av vår hårdvara. Vi sitter mycket med SIMIT och standardiserar logik och hjärnan i komponenterna. Jag har inte jättemycket fokus på CADen utan det lämnas till mekanikingenjörerna i projekten. Det har varit mycket fokus på virtuell drifttagning sedan start. Att kunna testa av en station och all kod innan drifttagning för att spara tid och pengar i slutändan. Nu försöker vi hitta andra spår så man kan få

nytta av tvillingen i efterhand, eller även i tidigare faser som designfasen så att hela projektet går in och utvärderar design och hittar om någon mekaniskt måste ändras. Då kanske man behöver vissa rörelser eller kod redan gjord för att lättare hitta sånt. Sen i efterhand kan det handla om träning av operatörer eller underhåll.

Vilka lösningar inom området (virtuella modeller, virtual commissioning, digital model/shadow/twin) erbjuder ni?

Vi säljer det som en tjänst eller digital produkt, men använder det också internt för AFRY-projekt för att bli effektivare. Men den själva digitala tvillingen säljs som en produkt. Vi har nästan alltid SIMIT som grund, det måste inte vara det, men då vi byggt upp ett stort bibliotek är det en bra grund. Är det en robotcell med t.ex en ABB-robot så använder man förmodligen redan robotstudio som 3D-miljö och då gör vi det också. Ibland jobbar vi med processindustri där man mer är intresserad av flöden och beräkningar och då kan det räcka med biblioteket i SIMIT, då är det inget som rör sig som behöver visualiseras utan beräkningar för själva modellen.

De här driftsättningsprojekten, är det något ni gör varje projekt?

Det har nog varit en grund i alla projekt hittills, sen är det de här andra delarna som kommer till då. Men som sagt det är ingen nödvändighet, man kanske vill bygga upp en digital tvilling av något som redan finns för att man vet att man vill bygga ut i framtiden.

Många företag definierar digital tvilling olika, vad är det AFRY menar med digital tvilling, vad är det ni gör med den digitala tvillingen?

När vi pratar inom AFRY om digital tvilling så pratar vi om emulerade styrsystem inom någon typ av industri. Nu kommer jag inte med någon solklar definition tyvärr, men ofta emulering av signaler. Man skulle ju kunna kalla en modell av en stad för en digital tvilling, där man kan gå runt bland hus och så.

I ditt examensarbete skrev du om utmaningar kring avsaknad av standardisering, är det något som fortfarande är utmanande eller har det blivit bättre?

Det är en utmaning att hitta information om hur olika hårdvara fungerar. Det kan vara företagshemligheter från tillverkaren. När det gäller standardisering är det en utmaning att det inte finns någon standard kring signalnamn eller hur man ska strukturera upp ett projekt som gör att alla parter har samma syn på allt. Det blir väldigt olika från projekt till projekt. Även hur man ska testa saker.

Det är ofta upp till kunden vilken standard som ska användas, och även lite branchspecifikt. I processindustrin finns det ofta väldigt tydliga och välutvecklade bibliotek redan. I de fallen brukar det gå smidigare. Inom AFRY finns det en standard till viss del, men den räcker inte hela vägen.

Vad ser ni mer som utmaningar den närmsta tiden?

Det är mycket utmaning med överlämnande mellan olika dcipliner. Det är elektronikkonstruktörer och mekanikkonstruktörer som ska lämna över till programmerare. De ska kunna sammarbeta på ett bra sätt så att all kommunikation uppfattas på samma sätt åt båda håll. Sen måste allt det här överlämnas till väran del (virtuell driftsättning) det är just där man behöver en standardisering från start. Hur man benämner saker och hur mycket info man ska överlämna så att alla förstår det de behöver förstå. Annars blir man lätt personberoende. Det har inte riktigt behövts tidigare på samma sätt.

Görs programmeringen simultant med den mekaniska och elektriska konstruktionen?

Oftast görs det simultant. De behöver ju en viss del av elen för att man ska kunna starta med någon programmering. Ofta behöver man nog starta innan det är färdigt för att hinna med.

Använder ni driftsättningsmodellen till att testa PLC och robot-kod efterhand, eller får ni all kod efteråt och testar den först då?

Idealt sett skulle man haft en testplan för att testa av efterhand i tvåveckors-sprintar och att man bestämmer att nu fokuserar vi på den delen och testar den. Det är också en utmaning då detta oftast inte fungerar. Det här arbetssättet är man inte van vid i branchen överhuvudtaget så ofta blir det att man i efterhand testar allt väldigt stressigt. Där måste vi förändra arbetssätt och få in en testplan i en vanlig tidsplan.

Scania

På vilket sätt använder ni virtuell driftsättning?

I våra helautomatiska liner använder vi alla konfigurationer av driftsättning. I de semi-automatiska fabrikena så ser vi att Hardware in the loop blir lite svårare eftersom vi blandar in manuell hantering vilket gör det lite mer komplierat. De manuella linerna så är det verklig driftsättning, men också reality in the loop då vi vill ta den virtuella koden och testa den mot vår verkliga fabrik. Vi byter ju bland annat utrustning frekvent, en maskin eller en dragare. I de fallen vill vi kunna köra vår PLC-kod virtuell, stega fram till en viss sekvens och säkerställa att signalerna går fram till den processen som ska slutföras.

Vilka fördelar ser ni med virtuell driftsättning?

Vi ser en relevans i att kunna bereda vår fabrik och våra processer tidigt, men också kunna jämföra dessa mot verkligheten för att den virtuella modellen ska efterlikna så som verkligheten är. Vi jobbar till största delen med externa partners när det kommer till förändringar i ävra liner, det är väldigt sällan vi gör förändringar själv. När det gäller body in white delarna så jobbar vi med de stora drakarna som EBZ och EDAG och med mindre förändringar är det ofta ARFRY eller liknande företag.

Vad ser ni som utmaningar den närmsta tiden?

Om vi kollar till beredningen så använder vi många olika mjukvaror, IPS, Delmia, enovia, catia och autodesk. För MES-systemet har vi Apriso. Sen har vi utrustningssimuleringsverktyg där RobotStudio är en stor del. Vi använder Dürr för målning och Universal robots för plock. Just för att få den exakta digitala kopian av vår utrustning, vi tror att den är väldigt viktig när vi kommer till det skedet. Sen har vi vår styrtutrustning som till stor del består av Siemensprodukterna. Det projektet vi håller på med nu, och det som kommer ta större delen av vår tid de kommande åren är att vi vill byta ut så många system som möjligt mot 3D-Experience. Det kommer vara vårt huvudalternativ när det kommer till att skapa vår virtuella data, både när det gäller produktion, service, R&D och sales. Även inom alla discipliner, oavsett om vi pratar robotsimulering, monteringsimulering, logistiksimulering, verktygskonstruktion och alla de här olika disciplinerna som finns som underkategorier kommer också införlivas i 3DExperience.

Så ni ser ett behov av att centralisera och minska antalet mjukvaror?

Vårt budskap är att vi kommer ha vår produkt och vår beredning i 3DExperience. Vår robotkod kommer ligga i 3DExperience. RobotStudio kommer vara det program som bäst hanterar den koden, där vi förädlar den. Men för att kunna göra det måste vi också ha den virtuella produktionsutrustningen i RobotStudio och vårt stora problem är hur vi får över den från 3DExperience, eftersom modellerna kommer mastras där, det finns inget alternativ till det. Vi behöver kunna plocka över hur vår fabrik ser ut vid ett givet tillfälle och positionera den utrustningen i robotstudio för att sen göra det vi behöver göra för att förändra en cell.

Hur mycket förändringar sker hos er?

Vi har 48 introduktionstillfällen varje år, då introducerar vi nya produkter i vår produktion. Då måste vi säkerställa att de nya produkterna passar i vår nuvarande fabriksstruktur. Flyttar man en komponent eller ersätter något måste det simuleras så vi säkerställer att vi har den åtkomsten vi behöver för våra svetsrobotar, för att kunna bekräfta att ändringen går att göra, eller vad som behöver byggas om för att kunna

göra detta. Denna typen av analyser gör vi tidigt idag, men det finns ett utrymme att bli ännu mer effektiv.

Vi kommer uppdatera våra inköpsrutiner kring hur vi får till oss modellerna blir ett viktigt redskap och också att om köp väljer att utesluta modellerna från leveranserna så är det inte enbart en besparing utifrån projektet utan det är en livslång kostnad vi kommer behöva hantera.

Hur använder ni era virtuella modeller idag?

Vi håller på att utforska hur den här informationen kan användas på bästa sätt. Idag använder vi den virtuella modellen främst för attkollisionschecka nya svetspunkter på nya plåtar men ser att det finns ytterligare användningsområden. Vi har en hel del PowerPoints som i praktiken beskriver produktflöden och varje gång det sker en förändring i ett program eller svets så behöver den här dokumentationen uppdateras. Det skulle vi kunna addera som ett lager i vår virtuella fabrik.

Har ni märkt några utmaningar när ni jobbar med integratörer?

Vi har många gånger efterfrågat att integratörerna jobbar direkt i våra system och göra utvecklingsarbetet direkt hos oss, men det vill dem inte. Anledningen till att de inte vill det är att de naturligtvis har en massor IP om hur de utvecklar en line eller en robotcell. Den Ipn vill inte dem släppa ifrån sig eftersom det finns en chans att konkurrenter kommer över den Ipn. När det gäller utbyte av data mellan partners och OEM så är det inte bara att digitaliseringen möjliggör att alla jobbar i samma låda, utan det finns mycket kopplat till företags egen vilja att äga sin egen del och skicka ut den när den är klar vilket komplicerar.

Sen ska sägas att i många av våra projekt så gör ju våra underleverantörer virtual commissioning, men de har sina egna setuper i hur de gör det. Så i de projekten så har underleverantörerna driftsättningsansvaret. Vi ser mer virtual commissioning när vi gör små förändringar för att säkerställa att de går smärtfritt. Men att vi för leverantörerna ska specia exakt hur de ska utföra driftsättningen kommer förmodligen bli mycket dyrare för oss än att leverantörerna skulle få göra det enligt sin egna metodik och processer.

Vi pratade lite om att ni ska mastra allt i 3Dexperience, är det fortfarande 2D-ritningar som är mastern eller har ni funderat på att köra Model based definition eller liknande?

Vi har MBD-program men de riktar sig främst mot bearbetade parter, det är där vi börjat. Anledningen till det är att det är där vi konsumerar kraven som är inuti artikeln. Där har vi kommit ganska långt, inom 16 månader tror vi att vi ska kunna byta ut vårt existerande beredningssystem för smitt och gjutet gods till ett system baserat på MBD och 3Dexperience. Det är absolut en fördel att få bort 2D-ritningarna. Tyvärr förlorar vi små leverantörer som inte är rustade att jobba med MBD, så vi kommer fortsätta behöva generera 2D-ritningarna. Det är snarare en fråga om vilken modell som ska hantera masterinformationen.

Volvo

När började ni med virtuell driftsättning?

Vår resa började 2016 när vi gjorde ett arbete tillsammans med några studenter från högskolan i Skövde. Vi ville veta hur virtuell drifttagning kunde hjälpa oss med fokus på när vi köper ny utrustning. Vi tittade lite på olika verksamheter, hur gör dem? Vi tittade mycket på body in white och vår egen verksamhet i Umeå. Vi märkte ganska fort att vi är inte där, vi har inte de behoven som bygglinerna har. Vi har mekaniskt komplexa utrustningar där vi i vissa fall har robotar, andra inte. Vi har mest kontinuerliga förbättringar i produktion. I våra fall när vi har en produktförändring innebär det i princip alltid en mekanisk ombyggnad av cellen. Så redan från start ville vi hitta en kostnadseffektiv lösning för att verifiera kod från robot, PLC, visionkameror eller vad det nu än kan vara. I det här exjobbet så testkörde vi den digitala tvillingen som vi kallade den då, för att se hur nära verkligheten vi kom. När vi körde start-stop sekvenser så såg vi att det började larma. När vi såg det började vi förstå kraften som ABBs virtuella kontroller har, så vi gick ut och testade detta i produktionscellen, och den betedde sig precis på samma sätt. Det var där och då beslutet togs att ABBs virtuella kontroller ska finnas med, på ett eller annat sätt, vad vi än gör. Den kommer vi inte släppa. Målet var då att vi skulle ha en industrialiserad lösning 2018 och använda det i riktiga anskaffningsprojekt.

Hur använder ni virtuell driftsättning idag?

Vi hade mycket diskussioner med ABB för att hitta en lösning som passade våra processer. Den lösning vi fastnade för var att köra Siemens PLCer och via SIMIT kör vi RobotStudio. RobotStudio kör roboten och även modellerna. Vi har krav på att alla givare som finns i utrustningen som påverkar logiken ska triggas i RobotStudio. Vi hade även ett annat spår när det kommer till NC-maskiner där det kan vara betydligt mer komplext med mekanik och då kör vi med NX istället, men finns det en robot i NX miljö så kör ändå RobotStudio roboten via SIMIT. Vi använder SIMIT för att

simulera elskåpen. Vi bygger beteendemoduler utifrån de komponenter som finns i elskåpen och det som inte triggas från robotstudiomodellerna. Vi har möjlighet att både köra hardware och software in the loop. Vi sa till ABB att vi behövde något enkelt för att koppla ihop RobotStudio med SIMIT. Det var framförallt kostnadsmässigt, vi kan inte sitta och göra manuella koppel. Tar det längre tid att göra simuleringen än att göra det i verkligheten så kommer det bli ifrågasatt. Men i det här skedet utvecklades en smart komponent som gjorde kopplingen snabb och enkel. Vi har krav på att retrofit projekt ska ha en simulering gjord först. Nu är vi uppe på nivån att vi testkör hela liner i RobotStudio.

Vad ser ni för fördelar med virtuell driftsättning?

Det är inte alltid vi kan få fram provmaterial. Det är inte alltid produkten är färdigutvecklad i det skedet. I den här lösningen kan vi debugga cykler och kommunikation utan att ödsla något material alls. En annan fundering vi haft är huruvida vi behöver köra en riktig provkörning om cellen är enkel eller om det räcker med en vFAT. Vi har också blivit mycket bättre på att använda SafeMove-funktionalliteten och testa säkerhetszoner och funktioner. Vi vill säkra kvalitén och minska risken för fördröjningar med kort och snabb uppstart. En bonus är att vi kan tidigarelägga aktiviteter i tillverkningsprocessen. I princip skulle integratörerna utveckla meket, elen och mjukvarumässiga delen samtidigt och parallellt, istället för som det är idag. Integratörerna presenterar ett övergripande koncept för oss. När de gjort det förväntar vi oss att framtagandet av mek och mjukvara sker parallellt.

Evomatic

Kan du först presentera dig och ditt företag?

Magnus Spåre heter jag, jobbar med PLC programmering i huvudsak och lite övergripande med automation. Har jobbat här i 20 år snart och Evomatic har funnits sedan 1998 och har växt mycket på de åren. Vi jobbar ju mycket med robotar, både Yaskawa och ABB, vi har gjort några försök med virtuell driftsättning med ABB robotar där vi har kört hela biten. Vi har också använt virtuell driftsättning i specialmaskiner och styrsystemsbyten.

Vilka delar är det då ni virtuellt drifttagit?

Vi har kört PLC, HMI och motorer. Vi har köpt in ny PLC och nya servomotorer och kopplat upp det mot vad vi hade och simulerat det I/O vi inte har. Man bytte I/O i skåpen och då kommer det inget till dem I/O-noderna så vi simulerade dem istället. Då kunde vi sätta ett nödstopp, manöver till, några extra knappar som manuell och gränslägen. Det var ingen grafik alls utan vi ville bara testa av funktionen. Ofta kommer man rätt så långt med det och bara se att det fungerar. Där hade man ju också kunnat ha med en robot fysisk eller virtuell.

Då ni jobbar med ABB, har ni använt något från RobotStudio för själva driftsättningen eller hur används det?

Det har gjorts en gång med robot och PLC där hela systemet sattes upp virtuellt och provkörde det. Det gick ganska bra, knytningen med roboten och PLCn var lite omständlig. Vi har ju en växel däremellan, SIMIT, mellan robot och PLC.

Är det några lösningar kring detta ni använder just nu ofta vid projekt, t.ex virtuella modellere eller virtuell driftsättning?

Vi kör ju alltid robotsimulering, men ofta stannar det där. Man har simulerat roboten men simulerar inte vidare där utan då körs det på riktigt, men vi tittar på att köra det mer virtuellt. Ofta simulerar man ju bara robotbanorna och gör ett offlineprogram.

De här simuleringarna, är det något ni delar med er av till kund eller är det bara internt?

Ofta visar vi dem för kund i alla fall, det används ofta redan i säljfasen. Man visar en programmerad modell i RobotStudio med banor och hur det plockas och så. Det har vi gjort väldigt länge, säkert över 15 år.

Vid användande av RobotStudio är man ju bunden till ABB-robotar, hur gör ni när ni använder Yaskawa?

Som jag förstått det så används då MotoSim, men den är inte alls lika duktig som RobotStudio. Där har vi ingen SIMIT-koppling idag så där har vi inte simulerat gentemot PLCn. Det är väl där ABB är bäst. Valet är ofta kundens och där har man vant personalen vid olika märken.

Vilka utmaningar ser ni på detta området?

Den största utmaningen är att ändra arbetssättet. Nu jobbar man väldigt linjärt. Vi börjar driftta när linjen är uppbyggd elektriskt och mekaniskt. Om man börjar jobba mer parallellt istället så kan man göra mer innan den är klar och göra en virtuell drifttagning. Det viktiga där är ju att programmeringen är i det läget att man kan börja göra en virtuell drifttagning, den måste läggas tidigare.

Hur ser du att denna parallellisering skulle ske?

Vi har gjort en omorganisation nu där vi delat upp företaget i verksamhetsområden så att mekaniker, automationare och robotprogrammerare sitter tillsammans. Det kan vara en del att man har lite närmare till varandra så att man kan ha snabbare kontakter

och mer samarbete inom gruppen inom projektet. Innan hade vi automation som satt och jobbade med sina grejer och sen satt mekanikerna och jobbade på sitt ställe. Nu är vi mer uppdelade i projekt snarare än funktion.

Har ni någon framtidsvision om vad som är nästa steg inom detta området?

Vi har börjat använda det mycket i specialmaskiner, även med SCADA-system där vi vill testa mycket. Sålänge vi håller oss till PLCn har vi kommit längre, men vi behöver kunna baka in robotar smidigare för att kunna knyta ihop den säcken. Jag ser stora säkerhetsmässiga fördelar, för ofta när vi börjar driftta så är ju elektriker och mekaniker inte klara, ofta ligger ju projektet efter, det är en klassiker. Då ska alla jobba samtidigt på slutet för att slutdatumet står fast. Där finns ju en säkerhetsrisk att vi startar grejer och folk ska springa in och skruva med grejer vilket kan bli farligt. Om vi då kan göra det virtuellt istället så har vi kunnat testa av en del medan de bygger klart. Det är ju inte heller så effektivt att gå på varanda. Det blir avbrott hela tiden i verksamheten för alla inblandade.

Har ni märkt någon förändring från kunder i deras beställningar, om det ställs andra krav till exempel?

Vi har inte märkt av det där jättemycket men det har väl dykt upp nu att någon har tänkt att det varit bra om vi kunnat simulera det innan. Det är också rätt intressant för det är ju ett mervärde för kunden att ha en virtuell modell för om man ska göra förändringar så kan man lägga in en funktion i den färdiga modellen vilket inte alls är mycket jobb.

Har ni någon långsiktigare vision om du ser några år framöver?

Där har vi inte kommit mycket längre, man kanske borde prata om ett visionsarbete där, var man ska lägga sig. Sen jobbar vi ju på den nivån att vi har robotceller från en till fem, tio robotar. Vi gör ju inga fabriker så att säga. Nästa steg hade kunnat vara att göra flödesanalyser i fabrik men där är vi inte riktigt ännu.

Har ni någon uppfattning kring hur mycket ni sparar på det här i tid eller pengar?

Vi har väl inget specifikt egentligen, men det vi kan se är att vi lägger mer kontorstid. Vad jag hört är totaltiden den samma, men man ligger mer tid 7-16 på kontoret än på övertid ute i fabriken vilket sparar pengar.

Väderstad

På vilket sätt använder ni virtuella modeller hos er i Väderstad?

Det är för att simulera lackprogram, skapa nya program framförallt så laddar vi in CAD-modellen och sen använder vi den för att spara tid. Det är ett mycket enklare sätt att arbeta på än tidigare.

Hur länge har ni använt denna metod och hur gjorde ni innan?

Ungefär fyra år. Innan dess var arbetssättet liknande men vi använde en annan mjukvara, UltraArc. Det var en sämre mjukvara där ett runt rör var åttakantigt och så vidare, men principen var den samma. Robotkoden genererades inte i det programmet utan det behövde scriptas i tre steg. Man fick kopiera det man gjort och klistra in till en server.

Kan du förklara er process?

Vi får en fil från konstruktörerna som tas in i Creo Flex där vi har fysikberäkning där lackvagnen finns. Där sitter en koordinat i ett av hålen så att produkten alltid kopplas till den i rätt position. Då tar vi konstruktörens ramdel, tillsätter kedjor så den hänger så som vi vill och exporterar i STEP. Sen komprimerar vi dem i RobotStudio och gör om dem till ett library där vi har ett arkiv så vi alltid kan plocka upp saker vi tidigare jobbat med.

Efter det har vi alla program på en server så att vi kan ladda ner programkoden och testa och göra ändringar.

Är det alltid VR ni använder?

Nej är det enkla detaljer, till exempel rör eller liknande så använder vi paintstrokes i paint-powerpaket. Då får man automatiskt påslag. Mer komplexa delar kör vi i VR eftersom vi då kan kontrollera orienteringen enklare vilket är viktigt i en lackeringsapplikation. Det var när vi bytte robotar som vi passade på att hitta den här funktionaliteten.

Har ni någon uppfattning om hur mycket resurser ni sparar genom detta?

Rent tidsmässigt är det lätt halva tiden att få något körbart med hjälp av VR. Användandet av RobotStudio har också sparat otroligt mycket tid. Den är mycket mer driftsäker och snabbare, även där lär tiden ha minskats med hälften.

Har ni någon framtidsvision om vad som är nästa steg inom detta område?

Just nu är det kvalitetshöjande med visionsystemet, vi har ju kollisioner och sånt och det påverkar ju roboten väldigt negativt. Eftersom det är en EX-klassad miljö så får vi

inte ha elektriska brytare på fästen så då kan man slå sönder leder och sånt på roboten. Annars känns det inte som vi har så mycket frammåt just nu, VR var ett stort steg för oss.

Sen har vi ju det här i bakhuvudet kring hur smidigt det är att arbeta med RobotStudio och VR:en när vi ska bygga nya system och bygga in dem från början och då se vad marknaden har för nytt att erbjuda. Det har ju garanterat kommit både nya funktioner och förbättringar och där har vi en förstudie just nu som rätt grovt dumsuger marknaden på hur det går att förbättra det hela.

Elektroautomatik

Kan du först presentera dig och ditt företag?

Staffan Lundgren heter jag och är anställd som automationsingenjör på elektroautomatik. Just nu jobbar jag nästan uteslutande med PLC och HMI programmering tillsammans med virtuella driftsättningar. Jag gör en del robotprogrammering men det har fått stå tillbaka en del den senaste tiden. Jag gör mycket systemeringar. På vår avdelning i Skövde är vi ungefär 20 anställda, men har för det mesta inhyrda konsulter då vi inte hinner.

Vilka mjukvaror använder ni vid virtuell driftsättning? Är det upp till kunden?

Vi jobbar väldigt mycket med robotceller och den stora majoriteten är ABB, därav att vi har RobotStudio så att vi får robotsystemet. Det är lite knutet till vilket robotsystem vi använder. Får vi välja själva kör vi Siemens-PLC, men vi är inte bundna till det. Vi använder också Allen-bradley, ABB, Beckhoff osv. Är det något vi inte kan så brukar vi lära oss det med.

Säljer ni Virtuell driftsättning som en tjänst eller är det bara användning internt?

Det är allt fler kunder som önskar en virtuell drifttagning, då brukar vi lägga med det som en option som kunden kan välja att köpa till. Sen har det hänt att vi haft projekt där vi känt själva att ”det här verkar lite svajigt” och då har vi gjort en virtuell drifttagning för egen del men då inget vi presenterat för kunden. Vi bygger upp cellen virtuellt och provkör allt själva i lugn och ro, sen har vi en vFAT där kunden kommer hit till oss och kör en regelrätt FAT fast i den virtuella miljön, där kunden har med sina FAT protokoll.

Har ni någon framtidsvision om vad som är nästa steg inom detta området?

Att det kommer bli mer och mer virtuella drifttagningar är det inget snack om, det kommer det bli. Det som vi behöver jobba på är att bygga våra virtuella stationer snabbare för det tar tid. Både och bygga den virtuella 3d-miljön i RobotStudio men också utnyttja SIMIT-miljön. Kanske också undersöka andra verktyg istället för RobotStudio som är smidiga och bättre att jobba med för oss.

Har ni någon uppfattning om hur mycket ni sparar på virtuell driftsättning?

Det beror såklart på komplexiteten på anläggningen, men totalt sett sparar vi tid. Sen är det lite olika men är det en enklare robotcell så kanske vi sparar upp till två veckor, där den vanligtvis skulle ta ungefär 5 veckor. Det tar ju tid att bygga den virtuella modellen, men det är ju inte bara att vi minskar igångkörningstiden, för det blir den helt klart, och säkrare då vi lusat av en hel del. Men hela den virtuella driftsättningen är ju en riskminimering för oss totalt sett, i och med att vi gör det här innan vi börjar bygga den fysiska cellen. Det som blir dyrt är när något inte funkar med mekaniken, något som slår i eller problem med räckvidden. Vi har i ett par projekt upptäckt mekaniska problem under den virtuella driftsättningen där vi behövt konstruera om mekaniken innan den tillverkats och det sparar oss pengar och tid framförallt. Sen gör vi såklart alltid räckviddsanalyser, det gör våra robotprogrammerare i RobotStudio och kollar så de når överallt. Just nu vet du ju inte heller när nya grejer kommer med oroligheterna i världen.

När börjar ni sätta upp den virtuella miljön i ett projekt?

Nu för tiden krävs det ett annorlunda tänk, för det är så många olika kategorier av oss som måste vara färdiga med sitt arbete förhållandevis tidigt i förhållande till om du inte gör en virtuell driftsättning. Våra el och mek-konstruktörer brukar oavsett vilket gå igång ungefär samtidigt. Sen har vi ganska så standardupplägg så våra elkonstruktörer kan gå igång väldigt tidigt med grunderna. I det läget jobbar ofta systemeraren på elen med systemeraren på PLC-sidan där vi specar upp IO-listor och funktionsspecar så att mjukvarugruppen kan gå igång så fort som bara möjligt. Det gör vi ju redan innan el och mek-konstruktion går igång. För det är ju så att när vi ska börja bygga vår virtuella station så ska ju PLC och robotprogram och övriga mjukvaror vara färdiga också.

Appendix B Robotics code

In this appendix, the robotics code is presented. The robot code is grouped by robot and module.

PlateRobot

MODULE MainProgramPlate

```
!*****!  
!Module Name: MainProgram  
!Version: 1.0.0  
!Description: Module for the physical application of the plate Robot  
!Date: 06-06-22  
!Author: Lucas Gardebrand  
!*****!  
PROC main()  
!Initialize  
init;  
!Go home  
Home;  
!If simulate is true we will simulate the PLC in the SimulationProgram module  
IF simulate THEN simulationProg; ENDIF  
!This is the main program loop  
WHILE TRUE DO  
    TPWrite "Waiting for PLC to start job...";  
    WaitDI PRDIStartJob,1;  
    Reset PRDOJobDone;  
    !Update values from PLC  
    declaration;  
    Set PRDOJobStarted;  
    log("Job " + NumToStr(PRGJobNo,0) + " begin");  
    !Check which job was ordered from PLC  
    TEST PRGJobNo  
    CASE 1: Home;  
    CASE 2: ScanForStack;  
    CASE 3: ToStackOrig;  
    CASE 4: PutPlateOnTurnerOrigReal;  
    CASE 5: PickFromTurnerOrigReal;  
    CASE 6: PickFromVacuum;  
    CASE 7: ToTable2;  
    DEFAULT:  
        TPErase;  
        SetGO PRGOErrNo,21;
```

```

        log("Job number " + NumToStr(PRGJobNo,0) + " was received and is invalid,
        exiting");
        TPReadFK nKnappVal,"(OK)","OK","","","";
        EXIT;
    ENDTEST
    WaitDI PRDIStartJob,0;
    Set PRDOJobDone;
    log("Job " + NumToStr(PRGJobNo,0) + " done");
    Reset PRDOJobStarted;
ENDWHILE
ENDPROC

!JOB 1 Moves the robot to home position
PROC Home()
    !If robot is within SafeZone HomingPlate it will move directly to the HomePos
    IF HomingPlate =1 THEN
        log("PR in platezone");
        MoveJ RelTool(origoStack,-116,320,-
1100\Rz:=130),v200,z200,tGrimmaOrigID\WObj:=wStackOrig;
        !If robot is within SafeZone HomingTurner it will first move upwards and then to
HomePos
        ELSEIF HomingTurner=1 THEN
            log("PR in TurnerOrigzone");
            MoveL RelTool(CClearOfAll,0,0,0),v200,z200,tGrimmaOrigID\WObj:=wT2Orig;
            MoveJ RelTool(origoStack,-116,320,-
1100\Rz:=130),v200,z200,tGrimmaOrigID\WObj:=wStackOrig;
            !If robot is within SafeZone HomingTable it will first move above the table, out from the
table and then to HomePos
            ELSEIF HomingTable=1 THEN
                log("PR in Tablezone");
                p1:=CROBT(\Tool:=tGrimmaOrigID\WObj:=wT2Orig);
                MoveL RelTool(p1,0,0,-T2ApproachDist-
p1.trans.z),v200,z0,tGrimmaOrigID\WObj:=wT2Orig;
                MoveL RelTool(CClearOfAll,0,0,0),v200,z200,tGrimmaOrigID\WObj:=wT2Orig;
                MoveJ RelTool(origoStack,-116,320,-
1100\Rz:=130),v200,z200,tGrimmaOrigID\WObj:=wStackOrig;
                !If the robot is not in any of the above SafeZones it has been jogged and must be jogged
within the zones to home correctly.
                ELSE
                    TPErase;
                    SetGO PRGOErrNo,20;
                    log("Robot was not in zone and had been jogged away, can not go home");
                    TPWrite "Not in a position to go home, please jog robot above plates";
                    TPReadFK nKnappVal,"(OK)","OK","","","";
                    ExitCycle;
                ENDIF
            log("PR was homed");
        !
    ENDIF
ENDPROC

```

```

!JOB 2, moves robot above the stack for measuring
PROC ScanForStack()
  !Saves parameters AB and rotation for sanity check in Job 3
  sanityAB := nAB;
  sanityRotate := nRotated;
  log("nXPos "+NumToStr(nXPos,1)+" nYPos "+NumToStr(nYPos,1)+" nZPos
"+NumToStr(nZPos,1));
  log("nRot "+NumToStr(90*nRotated,0)+" nAB "+NumToStr(180*nAB,0)+" nAngle
"+NumToStr(nAngle,1));
  !Moves above stack area in a safe manner
  MoveJ RelTool(origoStack,-500,-300,-1000\Rz:=90-90*nRotated-
180*nAB),v2000,z150,tGrimmaOrigID\WObj:=wStackOrig;
  !Moves above specific stack
  MoveJ RelTool(origoStack,-nXPos,-nYPos,-800\Rz:=90-90*nRotated-
180*nAB),v2000,fine,tGrimmaOrigID\WObj:=wStackOrig;
ENDPROC

!JOB 3, moves robot to the stack with PLC values
PROC ToStackOrig()
  !Checks against parameters from job 2 to make sure they are the same
  IF sanityAB <> nAB OR sanityRotate <> nRotated THEN
    TPWrite "The rotation or AB have been changed since last job, fatal error";
    SetGO PRGOErrNo,22;
    EXIT;
  ENDIF
  log("nXPos "+NumToStr(nXPos,1)+" nYPos "+NumToStr(nYPos,1)+" nZPos
"+NumToStr(nZPos,1));
  log("nRot "+NumToStr(90*nRotated,0)+" nAB "+NumToStr(180*nAB,0)+" nAngle
"+NumToStr(nAngle,1));
  !Goes down to the plates
  MoveL RelTool(origoStack,-nXPos,-nYPos,nZPos-200\Rz:=90-90*nRotated-
180*nAB),v2000,z200,tGrimmaOrigID\WObj:=wStackOrig;
  !Angle is not yet used, but can be used to angle the tool if the plate stack is angled
  MoveL RelTool(origoStack,-nXPos,-nYPos,nZPos-10\Rx:=nAngle*(1-
nRotated)\Ry:=nAngle*nRotated\Rz:=90-90*nRotated-
180*nAB),v2000,z50,tGrimmaOrigID\WObj:=wStackOrig;
  MoveL RelTool(origoStack,-nXPos,-nYPos,nZPos\Rx:=nAngle*(1-
nRotated)\Ry:=nAngle*nRotated\Rz:=90-90*nRotated-
180*nAB),v50,fine,tGrimmaOrigID\WObj:=wStackOrig;
ENDPROC

!JOB 4, moves the plate to the Turner
PROC PutPlateOnTurnerOrigReal()
  log("nXPos "+NumToStr(nXPos,1)+" nYPos "+NumToStr(nYPos,1)+" nZPos
"+NumToStr(nZPos,1));
  log("nRot "+NumToStr(90*nRotated,0)+" nAB "+NumToStr(180*nAB,0)+" nAngle
"+NumToStr(nAngle,1));
  !Lower acceleration while holding plate
  AccSet nAcc,nRamp;
  !Above stack

```

```

    MoveL RelTool(origoStack,-nXPos,-nYPos,nZPos-100\Rz:=nAngle*(1-
nRotated)\Ry:=nAngle*nRotated\Rz:=90-90*nRotated-
180*nAB),v2000,z100,tGrimmaOrigID\WObj:=wStackOrig;
    MoveL RelTool(origoStack,-nXPos,-nYPos,nZPos-800\Rz:=90-90*nRotated-
180*nAB),v2000,z200,tGrimmaOrigID\WObj:=wStackOrig;
    !Move to rotation pos
    MoveJ RelTool(origoStack,-615,-200,-1000\Rz:=90-90*nRotated-
180*nAB),v2000,z200,tGrimmaOrigID\WObj:=wStackOrig;
    !WaitDI PlateAllowedVand,1;
    MoveL RelTool(origoStack,-100,-300,-
1000\Rz:=130),v2000,z200,tGrimmaOrigID\WObj:=wStackOrig;
    MoveL RelTool(VCorner,nPlateWidth/2+VandOffset*3,-nPlateLength/2-
VandOffset*4,-200\Rz:=180),v2000,z50,tGrimmaOrigID\WObj:=wTurnerOrig;
    !Approach turntable
    MoveL RelTool(VCorner,nPlateWidth/2+50,-nPlateLength/2-50,-
50\Rz:=180),v2000,z10,tGrimmaOrigID\WObj:=wTurnerOrig;
    !At drop position
    MoveL RelTool(VCorner,nPlateWidth/2+50,-nPlateLength/2-50,-
20\Rz:=180),v400,fine,tGrimmaOrigID\WObj:=wTurnerOrig;
    ENDPROC

!JOB 5, gets the plate from the TurnerOrig
PROC PickFromTurnerOrigReal()
    log("PlateW "+NumToStr(nPlateWidth,1)+" PlateL "+NumToStr(nPlateLength,1)+"
PlateT "+NumToStr(nPlateThickness,1));
    !Go to position to pick from turner
    MoveL RelTool(VCorner,nPlateWidth/2,-nPlateLength/2,-
100\Rz:=180),v2000,z0,tGrimmaOrigID\WObj:=wTurnerOrig;
    MoveL RelTool(VCorner,nPlateWidth/2,-nPlateLength/2,-
nPlateThickness/3\Rz:=180),v600,fine,tGrimmaOrigID\WObj:=wTurnerOrig;
    ENDPROC

!JOB 6, gets the plate from the vacuum row
PROC PickFromVacuum()
    log("PlateW "+NumToStr(nPlateWidth,1)+" PlateL "+NumToStr(nPlateLength,1)+"
PlateT "+NumToStr(nPlateThickness,1));
    !Go to position away from turner so it can turn
    MoveL RelTool(VCorner,400,-600,-
500\Rz:=180),v2000,fine,tGrimmaOrigID\WObj:=wTurnerOrig;
    !Signal PLC that robot is at turning position
    Set PRDORobotAtTT;
    !Wait for turning procedure
    WaitDI PlateAllowedVand,0;
    WaitDI PRDIPickPlateTT,1;
    WaitDI PlateAllowedVand,1;
    Reset PRDORobotAtTT;
    !Move to pick position on turner
    MoveL RelTool(VCorner,850-nPlateWidth/2+200,-nPlateLength/2,-200-
vacuumOffset\Rz:=180),v2000,z50,tGrimmaOrigID\WObj:=wTurnerOrig;

```



```

MoveL RelTool(VCorner,850-nPlateWidth/2+200,-nPlateLength/2,-nPlateThickness-
vacuumOffset\Rz:=180),v2000,fine,tGrimmaOrigID\WObj:=wTurnerOrig;
WaitDI PRDIPickPlateTT,0; !Checks IO-reset
ENDPROC

```

!JOB 7, moves the plate to table 2

```

PROC ToTable2()
  WaitDI PRDIGrimmaStreight,0; !Checks IO-reset
  log("PlateW "+NumToStr(nPlateWidth,1)+" PlateL "+NumToStr(nPlateLength,1)+"
PlateT "+NumToStr(nPlateThickness,1));
  AccSet nAcc,nRamp;
  MoveL RelTool(VCorner,nPlateWidth/2+VandOffset+300,-nPlateLength/2-
VandOffset*4+200,-600\Rz:=180),v2000,z200,tGrimmaOrigID\WObj:=wTurnerOrig;
  MoveJ RelTool(CClearOfAll,0,50,0),v2000,z200,tGrimmaOrigID\WObj:=wT2Orig;
  WaitDI PlateAllowedTable,1;
  MoveL RelTool(origoT2,0,0,-
T2ApproachDist),v3000,z150,tGrimmaOrigID\WObj:=wT2Orig;
  MoveL RelTool(origoT2,0,0,-
nPlateThickness),v200,fine,tGrimmaOrigID\WObj:=wT2Orig;
  SetDO PRDOPlateRobotAtT3,1;
  WaitDI PRDIPlateReleased,1;
  SetDO PRDOPlateRobotAtT3,0;
  MoveL RelTool(origoT2,0,0,-
T2ApproachDist),v3000,z150,tGrimmaOrigID\WObj:=wT2Orig;
  MoveL RelTool(CClearOfAll,0,0,0),v3000,z200,tGrimmaOrigID\WObj:=wT2Orig;
  MoveJ RelTool(origoStack,-116,320,-
1100\Ry:=20\Rz:=130),v2000,z200,tGrimmaOrigID\WObj:=wStackOrig;
  WaitDI PRDIPlateReleased,0; !Checks IO-reset
ENDPROC
ENDMODULE

```

MODULE DeclareValue

```

!*****
!Module Name: DeclareValue
!Version: 1.1
!Description: Here are all values defined from PLC.
!Date: 06-06-22
!Author: Lucas Gardebrand
!***** PARAMETERS FROM PLC *****!
!
! SPEED PARAMETERS
! num PRGIAcc - Acceleration percentage
! num PRGIRamp - Ramp of acceleration percentage
! num PRGISpeed - Speed percentage
! bool PRDISlowFirstRun - First run will run at reduced speed
!
! PROGRAM PARAMETERS
! bool PRDIStartJob - Starts a new job

```

```

! num PRGIJobNo      - Selected job number
!
! RECEPIE PARAMETERS
! bool PRDIABSide    - Rotation of plate, 0 is normal, 1 is rotated
! bool PRDIAngle90   - Rotation of pallet, 0 is non rotated
! bool PRDIUpDownSide - Rotation of plate upside down
! num PRGIaPosPlate  - If pallet is slightly angled, does not need to be used

!***** PARAMETERS TO PLC *****!

! bool PRDOJobDOne    - Goes high when job is done, resets when new job started
! bool PRDOJobStarted - Goes high when a job is active, resets when job is done.
! bool PRDIPickPlateTT - OK for plate turner to move
! num PRGOErrNo       - Sets error number if exec error
! bool PRDOABSide     - Validation
! bool PRDOAngle90    - Validation
! num PRGOJobNo       - Validation
! num PRGOPlateLength - Validation
! num PRGOPlateWidth  - Validation
! num PRGOPlateThickness - Validation
! num PRGOSpeed       - Validation
! num PRGOxyzPosOffsets - Validation
!
!***** SAFETY SIGNALS *****!
! bool
!***** INTERNAL PARAMETERS DO NOT TOUCH *****!
VAR bool simulate := false; !Select if to use real PLC or simulated PLC
VAR num sanityRotate; !Holds value of rotation
VAR num sanityAB; !Holds value of AB
VAR num VandOffset:=50; !Offset from corner when dropping on turner
VAR num T2ApproachDist:=200; !Height above table when approaching T2
VAR string sanityerror; !String with errors
VAR wobjdata PlateStackTemp; !Temporary workobject for transformation
VAR num nAcc; !Value carrier
VAR num nRamp; !Value carrier
VAR num nSpeedPerc; !Value carrier
VAR num nXPos; !Value carrier
VAR num nYPos; !Value carrier
VAR num nZPos; !Value carrier
VAR num nRotated; !Value carrier
VAR num nAngle; !Value carrier
VAR num nPlateWidth; !Value carrier
VAR num nPlateThickness; !Value carrier
VAR num nPlateLength; !Value carrier
VAR num nPlateSensorHeight := 256; !Height between suction cups and sensor
VAR num nMeasureHeight := 800; !Height where we measure from
VAR num nAB; !Value carrier
VAR num nKnappVal; !Value carrier
VAR num vacuumoffset:=120;
VAR robtarg p1; !Temporary target

```

```
!Initialization step, resets signals
```

```
PROC init()  
  Reset PRDOJobStarted;  
  Reset PRDOJobDone;  
  Reset PRDORobotAtTT;  
  SetGO PRGOErrNo,0;  
  log("initialisation procedure completed");  
ENDPROC
```

```
!Declares values from PLC
```

```
PROC declaration()  
  AccSet 100,100;  
  ConfJ\On;  
  ConfL\On;  
  !Values from Real PLC  
  IF NOT simulate THEN  
    nXPos:=PRGIxPosPlate/10;  
    nYPos:=PRGIyPosPlate/10;  
    nYPos:=nYPos-430;!Avstånd pall 1 pall 2 Y.  
    nZPos:=PRGIzPosPlate/10;  
    nZPos := nZPos-800-256;  
    nSpeedPerc:=PRGISpeed;  
    nRotated:=PRDIAngle90;  
    nPlateLength:=PRGIPlateLength/10;  
    nPlateThickness:=PRGIPlateThickness/10;  
    nPlateWidth:=PRGIPlateWidth/10;  
    nAcc:=PRGIAcc;  
    nRamp:=PRGIRamp;  
    nAB :=PRDIABSide;  
    nAngle := PRGIaPosPlate;  
  
    !Values from simulated PLC  
  ELSE  
    nXPos:=615;  
    nYPos:=615;  
    nYPos:=nYPos-430;!Avstånd pall 1 pall 2 Y.  
    nZPos:=855;  
    nZPos := nZPos-nMeasureHeight-nPlateSensorHeight;  
    nSpeedPerc:=100;  
    nRotated:=0;  
    nPlateLength:=1150;  
    nPlateThickness:=3;  
    nPlateWidth:=250;  
    nAcc:=100;  
    nRamp:=100;  
    nAB := 1;  
    nAngle := 0;  
  ENDIF  
  setSpeed;
```

```

    log("declaration procedure completed");
    sanitycheck;
ENDPROC
!Sanity checks values from PLC
PROC sanityCheck()
sanityerror:="";
IF nXPos>2000 OR nXPos<10 THEN sanityerror:="xPos"; ENDIF
IF nYPos>1000 OR nYPos<-700 THEN sanityerror:="yPos"; ENDIF
IF nZPos>1000 OR nZPos<-600 THEN sanityerror:="zPos"; ENDIF
IF nPlateLength>1300 OR nPlateLength<200 THEN sanityerror:="length"; ENDIF
IF nPlateThickness>10 OR nPlateThickness<1 THEN sanityerror:="thickness"; ENDIF
IF nPlateWidth>1000 OR nPlateWidth<200 THEN sanityerror:="width"; ENDIF
IF pPlateOutOfTable = 0 AND PRGIJobNo <> 1 THEN sanityerror:="Robot are over
Table 2, job "+ numtostr(PRGIJobNo,0) + " is not allowed"; ENDIF
IF StrLen(sanityerror)>1 THEN
    log("SanityCheck failed, "+sanityerror+" had illegal value");
    TPRReadFK nKnappVal,"(OK)","OK","","","";
ENDIF
log("sanity check completed");
ENDPROC
PROC setSpeed()
IF PRDISlowFirstRun=1 THEN
    VelSet 15,600;
    AccSet 60,60;
ELSE
    Velset nSpeedPerc,4000;
ENDIF
ENDPROC
ENDMODULE

```

MODULE SimulationProgram

```

!*****!
!Module Name: SimulationProgram
!Version: 1.0.0
!Description: Module for the simulated application of the plate Robot
!Date: 06-06-22
!Author: Lucas Gardebrand
!*****!
PROC simulationprog()
    WHILE TRUE DO
        declaration;
        nAB := 0;
        nRotated := 0;
        ScanForStack;
        WaitTime 0.2;
        ToStackOrig;
        WaitTime 0.2;
        PutPlateOnTurnerOrigReal;
    ENDWHILE
ENDPROC

```

```

    WaitTime 0.2;
    PickFromTurnerOrigReal;
    WaitTime 0.2;
    ToTable2;
  ENDWHILE
endproc
ENDMODULE

```

GasketRobot

MODULE MainProgramGasket

```

!*****!
!Module Name: MainProgramGasket
!Version: 1.0.1
!Description: Module for the physical application
!Date: 06-06-22
!Author: Lucas Gardebrand
!*****!
PROC main()
  initData;
  declaration;
  toHome;
  IF simulation THEN simulationProg ENDIF
  WHILE TRUE DO
    TPWrite "Waiting for PLC to start job....";
    WaitDI GRDIStartJob,1;
    Reset GRDOJobDone;
    declaration;
    Set GRDOJobStarted;
    log("Job "+NumToStr(GRGIJobNo,0)+" begin");
    TEST GRGIJobNo
    CASE 1: ToHome;
    CASE 2: PickGasket;
    CASE 3: DropGasket;
    DEFAULT:
      TPErase;
      SetGO PRGOERRNo,21;
      log("Not OK jobNo = "+NumToStr(GRGIJobNo,0));
      TPReadFK nKnappVal,"(OK)","OK","","","";
    ENDTEST
    WaitDI GRDIStartJob,0;
    Set GRDOJobDone;
    log("Job "+NumToStr(GRGIJobNo,0)+" done");
    Reset GRDOJobStarted;
  ENDWHILE

```

ENDPROC

!JOB 1 move robot to home

PROC ToHome()

```

    IF HomingGeneral = 1 THEN
        MoveJ RelTool(origoScrew,-pickOffset,500,-
screwapproachdist\Rz:=90),v200,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
        ELSEIF HomingT1 = 1 THEN
            MoveJ RelTool(origoScrew,700,500,-
screwapproachdist\Rz:=90),v100,z100,GasketToolOrigID\WObj:=GasketInfeedProd;
            MoveL RelTool(origoScrew,-pickOffset,500,-
screwapproachdist\Rz:=90),v200,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
            ELSEIF Homingscrew = 1 THEN
                HomingTemp := CRobT(\tool := GasketToolOrigID\WObj:=GasketInfeedProd);
                HomingTemp.trans.z := origoScrew.trans.z-screwApproachDist;
                MoveJ HomingTemp,v200,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
                MoveL RelTool(origoScrew,-pickOffset,500,-
screwapproachdist\Rz:=90),v200,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
            ELSE
                TPErase;
                SetGO GRGOErrNo,20;
                log("Robot was not in zone and had been jogged away, can not go home");
                TPWrite "Not in a position to go home, please jog robot above plates";
                TPReadFK nKnappVal,"(OK)","OK","","","";
                ExitCycle;
            ENDIF
            log("PR was homed");
        ENDPROC

```

!JOB 2 move robot to pick position

PROC pickGasket()

```

    log("gasketDepth "+NumToStr(gasketDepth,1)+" PickOffset
"+NumToStr(PickOffset,1)+" middlePos "+NumToStr(middlePos,1));
    log("OffsetScrewX "+NumToStr(simScrewOffsetX/10,1)+" OffsetScrewY
"+NumToStr(simScrewOffsetY/10,1)+" OffsetScrewZ
"+NumToStr(simScrewOffsetZ/10,1));
    WaitDI gasketRobotAllowedGaskets,1;
    MoveJ RelTool(origoScrew,-pickOffset,-gasketDepth,-
screwapproachdist\Rz:=90),v100,z200,GasketToolOrigID\WObj:=GasketInfeedProd;
    AccSet 1,1;
    MoveL RelTool(origoScrew,-pickOffset,-gasketDepth,-
pickheight\Rz:=90),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
    ENDPROC

```

!JOB 3 bring gasket to table

PROC dropGasket()

```

    Accset 15,15;
    !Move away from screw
    MoveL RelTool(origoScrew,-pickOffset,-gasketDepth,-pickheight-
20\Rz:=90),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;

```

```

MoveL RelTool(origoScrew,-pickOffset,-gasketDepth,-pickheight-20\Rz:=90-
TweakAngle),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
MoveL RelTool(origoScrew,-pickOffset+36,-gasketDepth,-pickheight-20\Rz:=90-
TweakAngle),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
MoveL RelTool(origoScrew,-pickOffset+36,-
gasketDepth+Tan(TweakAngle)*middlePos*2,-pickheight-
20\Rz:=90),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
MoveL RelTool(origoScrew,-pickOffset+36,-gasketDepth+200,-
pickheight\Rz:=90),v50,fine,GasketToolOrigID\WObj:=GasketInfeedProd;
MoveL RelTool(origoScrew,-pickOffset+15,-gasketDepth+200,-
screwleavedist\Rz:=90),v150,z200,GasketToolOrigID\WObj:=GasketInfeedProd;
Set GRDOGasketPicked;
!Move to waiting position for
MoveJ RelTool(origoScrew,-pickOffset+800,-gasketDepth+700,-
screwleavedist\Rz:=90),v100,z200,GasketToolOrigID\WObj:=GasketInfeedProd;
WaitDI GRDILeaveGasket,1;
Reset GRDOGasketPicked;
WaitDI gasketRobotAllowedTable,1;
!Moves to position above borstbord
IF gr=-1 THEN
MoveJ RelTool(origoT1Temp,gr*600,-gr*200/2,-
495\Rz:=90),v100,z200,GasketToolOrigID\WObj:=T1Orig;
ENDIF
MoveJ RelTool(origoT1Temp,gr*(60+GasketWidth/2),-gr*(middlePos),-495\Rz:=90-
gr*90),v100,z200,GasketToolOrigID\WObj:=T1Orig;
!Approaches borstbord from above
MoveL RelTool(origoT1Temp,gr*(60+GasketWidth/2),-gr*(middlePos),-
GasketWidth\Rz:=90-gr*90),v100,z0,GasketToolOrigID\WObj:=T1Orig;
MoveL RelTool(origoT1Temp,gr*(60+GasketWidth/2),-gr*middlePos,-
GasketWidth\Rz:=90-gr*90),v100,z5,GasketToolOrigID\WObj:=T1Orig;
WaitTime 1;
!Circular motion putting down gasket
MoveC RelTool(origoT1Temp,-gr*sin(45)*GasketWidth+gr*(60+GasketWidth/2),-
gr*middlePos,-GasketWidth*cos(45)\Ry:=gr*45\Rz:=90-gr*90),RelTool(origoT1Temp,-
gr*sin(80)*GasketWidth+gr*(60+GasketWidth/2),-gr*middlePos,-
GasketWidth*cos(80)\Ry:=gr*dropangle/1.8\Rz:=90-
gr*90),v100,fine,GasketToolOrigID\WObj:=T1Orig;
!Release gasket
Set GRDOAtTable1;
WaitDI GRDIGasketReleased,1;
!Move Away perpendicular from table
AccSet 100,100;
MoveL RelTool(CRobT(\Tool:=GasketToolOrigID\WObj:=T1Orig),0,0,-
50),v80,z50,GasketToolOrigID\WObj:=T1Orig;
Set GRDOSTretchOK;
!Move to position outside of borstbord
IF gr=-1 THEN
MoveJ RelTool(origoT1Temp,gr*600,-gr*200/2,-
495\Rz:=90),v100,z200,GasketToolOrigID\WObj:=T1Orig;

```

```

    !MoveJ RelTool(origoT1Temp,-200,-gr*1200,-100\Rz:=90-
gr*90),v100,fine,GasketToolOrigID\WObj:=T1Orig;
    ELSE
    MoveL RelTool(origoT1Temp,-1000,-gr*middlePos,-100\Rz:=90-
gr*90),v100,z200,GasketToolOrigID\WObj:=T1Orig;
    ENDIF

```

```

Reset GRDOAtTable1;
MoveJ RelTool(origoScrew,700,500,-
100\Rz:=90),v100,z20,GasketToolOrigID\WObj:=GasketInfeedProd;
MoveJ RelTool(origoScrew,-pickOffset,500,-
100\Rz:=90),v100,z20,GasketToolOrigID\WObj:=GasketInfeedProd;
ENDPROC
ENDMODULE

```

MODULE DeclareValueGasket

```

    !***** PARAMETERS FROM PLC *****!
    !
    ! SYSTEM PARAMETERS
    ! bool GRDIMotorsActOn - Turns motors on
    ! bool GRDIStartMainPgm - Starts program from beginning
    ! bool GRDIStartPgm - Continues program at pointer
    ! bool GRDIResetEStop - Resets Emergency stop
    ! bool GRDIStopRobot - Stops the robot program execution
    !
    ! SPEED PARAMETERS
    ! num PRGIAcc - Acceleration percentage
    ! num PRGIRamp - Ramp of acceleration percentage
    ! num PRGISpeed - Speed percentage
    ! bool PRDIShowFirstRun - First run will run at reduced speed
    ! PROGRAM PARAMETERS
    ! bool PRDIStartJob - Starts a new job
    ! num PRGIJobNo - Selected job number
    !
    ! PROGRAM PARAMETERS
    ! bool PRDIStartJob - Starts a new job
    ! num PRGIJobNo - Selected job number
    ! RECEPIE PARAMETERS

    !***** PARAMETERS TO PLC *****!
    ! bool simJobDoneRob2 - Signals job has been completed successfully
    ! bool simInGripPosition - Signals that robot is in position to grab a new gasket
    ! bool simJobStartedRob2 - Signals ordered job has been started
    ! bool sInTable1 - Robot is in table1 area (safety signal)
    ! bool simInRestPosRob2 - Robot is in resting position
    ! num simError - Error has occurred with corresponding number
    !
    !***** PARAMETERS INTERNAL *****!
    !Which angle to drop the gasket on the BorstBord

```



```

VAR NUM dropangle:=60;
!Offset when approaching from above + is up
VAR NUM screwApproachDist:=100;
!Distance from borstbord
VAR NUM borstBordApproachDist:=30;
!Offset from screw when approaching from the front + is back
VAR NUM screwleavedist:=90;
VAR num pickheight := 0;
!***** DONT TOUCH*****!
VAR num gr;
VAR num gasketDepth := 28;
VAR num gasketWidth := 500;
VAR num TweakAngle := 6;
VAR string sanityError := "";
VAR num nKnappVal;
VAR num PressedKey;
VAR num nSpeedPerc:=100;
VAR robtargt
c5:=[[0,0,0],[1,0,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR robtargt
origoT1Temp:=[[0,0,0],[1,0,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR num PickOffset;
VAR num middlePos;
PERS bool simulation := FALSE;
VAR num abside;
VAR num toolwidth := 60;
VAR num screwWidth := 40;
TASK PERS robtargt HomingTemp:=[[-67.026,-65.5129,-100],[0.707113,1.09459E-06,-
8.06662E-06,0.7071],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

!*****!
PROC initData()
  Reset GRDOInGripPos;
  Reset GRDOGasketPicked;
  Reset GRDOAtTable1;
  Reset GRDOJobStarted;
  Reset GRDOSStretchOK;
ENDPROC

PROC declaration()
  IF NOT simulation THEN
    screwWidth := 40;
    Reset GRDOSStretchOK;
    GasketWidth:=GRGIGasketWidth/10-30;
    pickOffset := GRGIPickOffset/10;
    PickOffset := PickOffset+toolwidth/2-screwWidth/2;
    middlePos := GRGIMiddlePos/10;
    middlePos := middlePos - toolwidth/2;
    nSpeedPerc:=GRGISpeed;
  
```

```

    abside:= GRDIABSide;
ELSE
    Reset GRDOSstretchOK;
    GasketWidth:=GRGIGasketWidth/10;
    pickOffset := 80;
    PickOffset := PickOffset+toolwidth/2-screwWidth/2;
    middlePos := 495;
    middlePos := middlePos - toolwidth/2;
    nSpeedPerc:=100;
    abside:= 1;

ENDIF

IF abside=1 THEN
    gr:=-1;
    origoT1Temp:=origoT1;
    origoT1Temp.robconf:=[-1,0,0,0];
ELSE
    gr:=1;
    origoT1Temp:=origoT1;
    origoT1Temp.robconf:=[-1,0,-2,0];
ENDIF
GasketInfeedProd:=GasketInfeedOrig;
GasketInfeedProd.oframe.trans.x:=simScrewOffsetX/10;
GasketInfeedProd.oframe.trans.y:=simScrewOffsetY/10;
GasketInfeedProd.oframe.trans.z:=simScrewOffsetZ/10;
ConfJ\On;
ConfL\On;
sanitycheck;
setSpeed;
ENDPROC

!*****Performs sanity check on values*****!
PROC sanityCheck()
    sanityError:="";
    IF middlePos>700 OR middlePos<100 THEN
        sanityError:="gripCC";
    ENDIF
    IF GasketWidth>400 OR GasketWidth<50 THEN
        sanityError:="GasketWidth";
    ENDIF
    IF strlen(sanityError)>1 THEN
        log("SanityCheck failed, "+sanityError+" had illegal value");
        TPWrite "SanityCheck failed, "+sanityError+" had illegal value";
        TPReadFK nKnappVal,"(OK)", "OK", "", "", "", "";
        ExitCycle;
    ENDIF
    log("Sanitycheck conducted successfully");
ENDPROC

```

```

PROC setSpeed()
  IF GRDISlowFirstRun=1 THEN
    VelSet 15,250;
    AccSet 30,30;
  ELSE
    Velset nSpeedPerc,4000;
  ENDIF
ENDPROC
ENDMODULE

```

ClipRobots

Only one of the programs will be shown as they are mostly identical

MODULE MainProgramCF

```

!*****!
!Module Name: MainProgramCF
!Version: 1.0.0
!Description: Main module for front robot
!Date: 25-03-22
!Author: Lucas Gardebrand
!

```

```

!*****!

```

```

VAR num ZOffset := -80;

```

```

PROC main()
  init;
  Home;
  IF simulate THEN simulateProg; ENDIF
  WHILE TRUE DO
    TPWrite "Waiting for PLC to start job...";
    WaitDI CFDIStartJob,1;
    Reset CFDOGasketingDone;
    Reset CFDOJobDone;
    declaration;
    Set CFDOJobStarted;
    log("Job " + NumToStr(CFGIJobNo,0) + " begin");
    AccSet 20,20;
    TEST CFGIJobNo
    CASE 1: Home;

```

```

CASE 2: Gasket;
CASE 3: ClearImage;
CASE 4: GripPlate;
CASE 5: plateToStack;
CASE 6: plateToStack;
DEFAULT:
  TPErase;
  !SetGO PREROR,21;
  log("Not OK prgNo = " + NumToStr(CFGIJobNo,0) );
  TPReadFK nKnappVal,"(OK)","OK", "", "", "", "";
ENDTEST
WaitDI CFDIStartJob,0;
Set CFDOJobDone;
log("Job " + NumToStr(CFGIJobNo,0) + " done");
Reset CFDOJobStarted;
ENDWHILE
ENDPROC
PROC home()
  TempTarget_Rob2 := CRobT(\tool:=tGasket\WObj:=FT3Origo);
  TempTarget_Rob2.trans.y := WaitTarget_rob2.trans.y;
  MoveL TempTarget_Rob2,v800,z100,tGasket\WObj:=FT3Origo;
  MoveL WaitTarget_ROB2,v800,fine,tGasket\WObj:=FT3Origo;
  log("Robot " + NumToStr(CFGIJobNo,0) + " homed");
ENDPROC
PROC gasket()
  log("Actual X Pos:" + numToStr(nActualX-calibPlateWidth/2,1)+"Actual Y Pos:" +
numToStr(nActualY-calibPlateLength/2,1)+"Actual Angle:" + numToStr(nActualAngle,1));
  log("Set X Pos:" + numToStr(nActualX+calibPlateWidth/2,1)+"Set Y Pos:" +
numToStr(nActualY+calibPlateLength/2,1)+"Set Angle:" + numToStr(nSetAngle,1));

! Set simJobStarted;
ConfL\Off;
ConfJ\Off;
!Transform the actual positions
wGasketPos:=FT3Origo;
wGasketPos.oframe.trans:=[nActualX,nActualY,ZOffset];
wGasketPos.oframe.rot:=OrientZYX(nActualAngle,0,0);
wTestPos := wGasketPos;
!Move above the gasket
!WaitTime 10;
MoveL RelTool(Origo,-20,0,-80),v2000,z20,tGasket\WObj:=wGasketPos;

!Move to level with gasket
MoveL RelTool(Origo,-20,0,0),v2000,z20,tGasket\WObj:=wGasketPos;
!Approach gasket
MoveL RelTool(Origo,-2,0,0),v2000,fine,tGasket\WObj:=wGasketPos;
!Close gripper
GasketGrip;
!Transform the Set position
wGasketPos.oframe.trans:=[nSetX,nSetY,nSetZ+ZOffset];

```

```

wGasketPos.oframe.rot:=OrientZYX(nSetAngle,0,0);
!Move behind Set position
MoveL RelTool(Origo,-5,0,-1\Ry:=-10),v600,z0,tGasket\WObj:=wGasketPos;
MoveL RelTool(Origo,-8,0,8\Ry:=-20),v600,z0,tGasket\WObj:=wGasketPos;
MoveL RelTool(Origo,-2,0,4),v600,fine,tGasket\WObj:=wGasketPos;
!Open gripper
GasketRelease;
!Nudge gasket
MoveL RelTool(Origo,-4,0,4),v600,z1,tGasket\WObj:=wGasketPos;
MoveL RelTool(Origo,-4,0,0),v600,z1,tGasket\WObj:=wGasketPos;
MoveL RelTool(Origo,2,0,0),v1000,z1,tGasket\WObj:=wGasketPos;
!SetDO C
MoveL RelTool(Origo,-20,0,0),v1000,z10,tGasket\WObj:=wGasketPos;
Set CFDOGasketingDone;
!Move above plate
MoveL RelTool(Origo,-20,0,-80),v1000,z20,tGasket\WObj:=wGasketPos;
!Set flikad

!Get next Set pos from PLC and log values
log("FlikSequence (2) finished with parameters");
ENDPROC

PROC ClearImage()
!Move to a position where an image can be captured
MoveJ WaitTarget_ROB2,v2000,z10,tGasket\WObj:=FT3Origo;

ENDPROC
PROC gripPlate()
!Move to gripping position at table 3
MoveJ RelTool(Origo,0,-nPlateLength/2-80,-
250\Rz:=180),v2000,z10,Plate\WObj:=wPlatePos;
PlateRelease;
MoveL RelTool(Origo,0,-nPlateLength/2-
80,0\Rz:=180),v2000,z10,Plate\WObj:=wPlatePos;
MoveL RelTool(Origo,0,-
nPlateLength/2,0\Rz:=180),v2000,fine,Plate\WObj:=wPlatePos;
PlateGrip;
ENDPROC

PROC plateToStack()
AccSet 20,20;
SyncMoveOn sync1_task_list_;
MoveC RelTool(Origo,0,-Sin(88)*nPlateLength/2,-
nPlateLength/2*Cos(88)\Rx:=2\Rz:=180),RelTool(Origo,0,-Sin(86)*nPlateLength/2,-
nPlateLength/2*Cos(86)\Rx:=4\Rz:=180)\ID:=10,v500,z10,Plate\WObj:=wPlatePos;
MoveL RelTool(Origo,0,-Sin(86)*nPlateLength/2,-nPlateLength/2*Cos(86)-
100\Rx:=4\Rz:=180)\ID:=20,v800,z10,Plate\WObj:=wPlatePos;
!MoveAboveAxlar
ConfL\OFF;

```

```

    MoveL RelTool(Origo,1000,-Sin(86)*nPlateLength/2,-nPlateLength/2*Cos(86)-
100\Rx:=4\Rz:=180)\ID:=30,v800,z10,Plate\WObj:=wPlatePos;
    MoveC RelTool(Origo,1000,-Sin(88)*nPlateLength/2,-nPlateLength/2*Cos(88)-
100\Rx:=2\Rz:=180),RelTool(Origo,1000,-Sin(90)*nPlateLength/2,-nPlateLength/2*Cos(90)-
100\Rx:=0\Rz:=180)\ID:=40,v800,fine,Plate\WObj:=wPlatePos;
    SyncMoveOff sync2_;
    WaitTime 0.1;
    PlateRelease;
    WaitTime 0.5;
    MoveL RelTool(Origo,1000,-Sin(90)*nPlateLength/2-100,-nPlateLength/2*Cos(90)-
100\Rx:=0\Rz:=180),v800,fine,Plate\WObj:=wPlatePos;
    !Reset doGripPlateF;
    PlateGrip;
    WaitTime 0.5;
    MoveL RelTool(Origo,1000,-nPlateLength/2,-
160\Rx:=0\Rz:=180),v500,z10,Plate\WObj:=wPlatePos;
    MoveL RelTool(Origo,1000,-nPlateLength/2,-
100\Rx:=0\Rz:=180),v100,fine,Plate\WObj:=wPlatePos;
    MoveL RelTool(Origo,1000,-nPlateLength/2,-
160\Rx:=0\Rz:=180),v500,z10,Plate\WObj:=wPlatePos;
    MoveL RelTool(Origo,1000,-nPlateLength/2-400,-
160\Rx:=0\Rz:=180),v1000,z100,Plate\WObj:=wPlatePos;
    MoveL RelTool(Origo,1000,-nPlateLength/2-400,-
160\Rx:=0\Rz:=180),v1000,z10,Plate\WObj:=wPlatePos;
    Home;
ENDPROC

PROC GasketGrip()
Reset Local_IO_0_DO5;
    Set Local_IO_0_DO6;
WaitDI Local_IO_0_DI5,0;
WaitTime 0.16;
IF Local_IO_0_DI6 = 1 THEN
    setgo CFGOErrNo,2;
ENDIF
ENDPROC
PROC GasketRelease()
Reset Local_IO_0_DO6;
Set Local_IO_0_DO5;
WaitDI Local_IO_0_DI5,1;
ENDPROC
PROC PlateGrip()
Reset Local_IO_0_DO7;
    Set Local_IO_0_DO8;
WaitDI Local_IO_0_DI7,0;
WaitTime 0.16;
IF Local_IO_0_DI8 = 1 THEN
    setgo CFGOErrNo,3;
ENDIF
ENDPROC

```

```

PROC PlateRelease()
Reset Local_IO_0_DO8;
      Set Local_IO_0_DO7;
WaitDI Local_IO_0_DI7,1;
ENDPROC
ENDMODULE

MODULE DeclareValue
!***** PARAMETERS FROM PLC *****!
!
!   SYSTEM PARAMETERS
! bool CFDIMotorsActOn - Turns motors on
! bool CFDIStartMainPgm - Starts program from beginning
! bool CFDIStartPgm - Continues program at pointer
! bool CFDIResetEStop - Resets Emergency stop
! bool CFDIStopRobot - Stops the robot program execution
!
! SPEED PARAMETERS
! num CFGIAcc - Acceleration percentage
! num CFGIRamp - Ramp of acceleration percentage
! num CFGISpeed - Speed percentage
! bool CFDIShowFirstRun - First run will run at reduced speed
!
! PROGRAM PARAMETERS
! bool CFDIStartJob - Starts a new job
! num CFGIJobNo - Selected job number
!
! RECEPIE PARAMETERS
! num CFGIPVaPos - Process value Rotation of gasket, vision
! num CFGIPVxPos - Process value position of gasket, vision
! num CFGIPVyPos - Process value position of gasket, vision
! num CFGISetPointaPos - Set value Rotation of gasket,
! num CFGISetPointxPos - Set value position of gasket,
! num CFGISetPointyPos - Set value position of gasket,
! num CFGISetPointzPos - Set value position of gasket,
!
!***** PARAMETERS TO PLC *****!
!
! num CBGOErrNo - Sets error number if exec error
! num CBGOPVaPos - Validation
! num CBGOPVxPos - Validation
! num CBGOPVyPos - Validation
! num CBGOSetPointaPos - Validation
! num CBGOSetPointxPos - Validation
! num CBGOSetPointyPos - Validation
! num CBGOSetPointzPos - Validation
!
!***** INTERNAL PARAMETERS *****!

```

```

! CalibPlateLength      - Length between the two corner holes on calib plate
! CalibPlateWidth       - Width between the two corner holes on calib plate
!
!***** PARAMETERS INTERNAL DO NOT TOUCH *****!
VAR bool simulate := false;
PERS wobjdata wFlikPos;
VAR syncident sync1_;
VAR syncident sync2_;
TASK PERS wobjdata
    VAR num calibPlateWidth := 380;
    VAR num calibPlateLength := 1280;
!***** PARAMETERS FROM PLC *****!
VAR BOOL bFirstRun:=TRUE;
var bool bStartJob:= FALSE;
VAR bool bSlowRun := FALSE;
VAR num nJobNumber := 1;
VAR num nAcc := 100;
VAR num nRamp := 100;
VAR num nSpeedPerc := 100;
PERS tasks task_list_2:=["T_ROB1"],["T_ROB2"];
VAR num nPlateLength := 830;
VAR num stackDistance_ := 730;
VAR num nSetX; !Value carrier
VAR num nSetY; !Value carrier
VAR num nSetZ; !Value carrier
VAR num nSetAngle; !Value carrier
VAR num nActualX; !Value carrier
VAR num nActualY; !Value carrier
VAR num nActualAngle; !Value carrier

!***** PARAMETERS TO PLC *****!
VAR bool bPlateincenter:=FALSE;
var bool bJobStarted:= FALSE;
var bool bJobDone:= FALSE;
VAR bool bFlikningDone := FALSE;
VAR num nErrors := 0;
VAR bool bInRestPos := FALSE;
VAR bool table3Ready := FALSE;
VAR string sanityerror; !String with errors
VAR num nKnappVal;

PROC init()
    Reset CFDOJobStarted;
    Reset CFDOJobDone;
    SetGO CFGOErrNo,0;
    GasketRelease;
    log("initialisation procedure completed");
ENDPROC

PROC declaration()

```



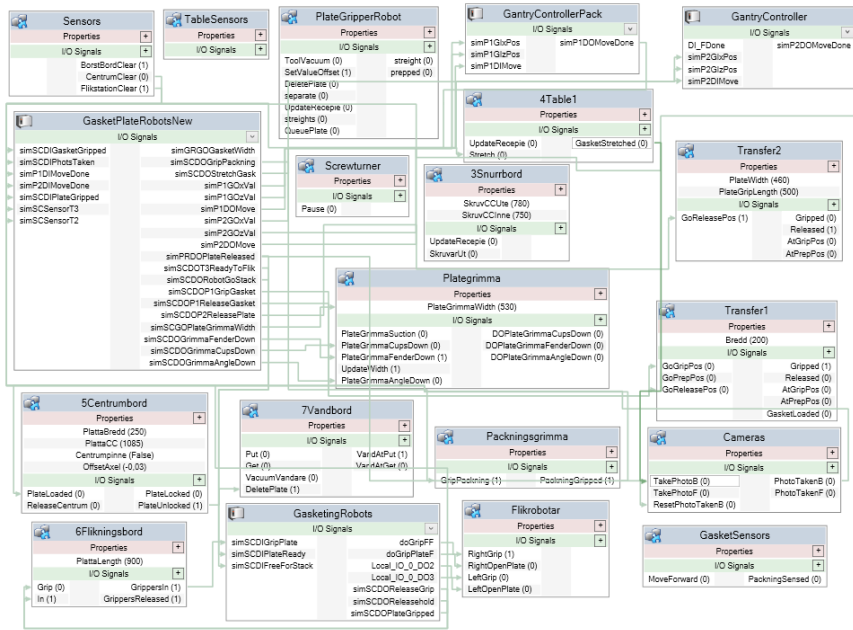
```
AccSet 100,100;
ConfJ\On;
ConfL\On;
IF NOT simulate THEN
!Get PLC values
nSpeedPerc:=CFGISpeed;
nAcc:=CFGIAcc;
nRamp:=CFGIRamp;
nActualX := CFGIPVxPos;
nActualY := CFGIPVyPos;
nActualAngle := CFGIPVaPos;
nSetAngle := CFGISetPointaPos;
nSetX := GInput(CFGISetPointxPos);
nSetY := CFGISetPointyPos;
nSetZ := CFGISetPointzPos;

!!!This part handle negative values!!!
IF nSetX > 32768 THEN
nSetX := nSetX - 65536;
ENDIF
IF nSetY > 32768 THEN
nSetY := nSetY - 65536;
ENDIF
IF nSetZ > 32768 THEN
nSetZ := nSetZ - 65536;
ENDIF
IF nActualX > 32768 THEN
nActualX := nActualX - 65536;
ENDIF
IF nActualY > 32768 THEN
nActualY := nActualY - 65536;
ENDIF
IF nActualAngle > 1800 THEN
    nActualAngle := nActualAngle-3600;
ENDIF
IF nSetAngle > 1800 THEN
    nSetAngle := nSetAngle-3600;
ENDIF
!The actual position
nActualX := nActualX/10+calibPlateWidth/2;
nActualY := nActualY/10+calibPlateLength/2;
nActualAngle := nActualAngle/10;
nSetX := nSetX/10+calibPlateWidth/2;
nSetY := nSetY/10+calibPlateLength/2;
nSetZ := nSetZ/10;
nSetAngle := nSetAngle/10;
nPlateLength := CFGIPlateLength/10;
ELSE
nSpeedPerc:=100;
nAcc:=100;
```

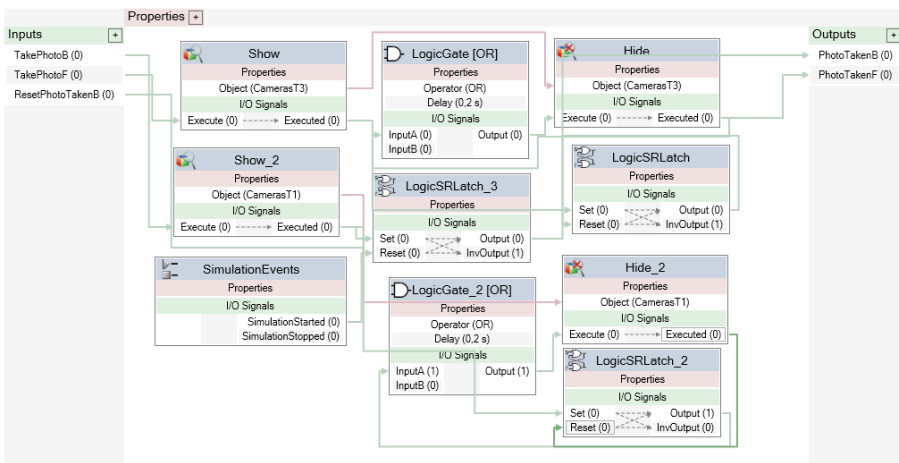
```
nRamp:=100;
nActualX := 200;
nActualY := -600;
nActualAngle := 50;
nSetAngle := 20;
nSetX := 220;
nSetY := 650;
nSetZ := 0;
nPlateLength := 1075;
ENDIF
log("declaration procedure completed");
wGasketPos:=FT3Origo;
wGasketPos.oframe.trans:=[nActualX,nActualY,ZOffset];
wGasketPos.oframe.rot:=OrientZYX(nActualAngle,0,0);
wPlatePos:=FT3Origo;
wPlatePos.oframe.trans:=[calibPlateWidth/2,calibPlateLength/2,0];
sanitycheck;
setSpeed;
ENDPROC
PROC setSpeed()
  IF CFDISlowFirstRun=1 THEN
    VelSet 15,600;
    AccSet 60,60;
  ELSE
    Velset nSpeedPerc,4000;
  ENDIF
ENDPROC
ENDMODULE
```

Appendix C. Smart components

In this appendix, the Smart components are presented
Station logic



Vision system



Gasketingrobots

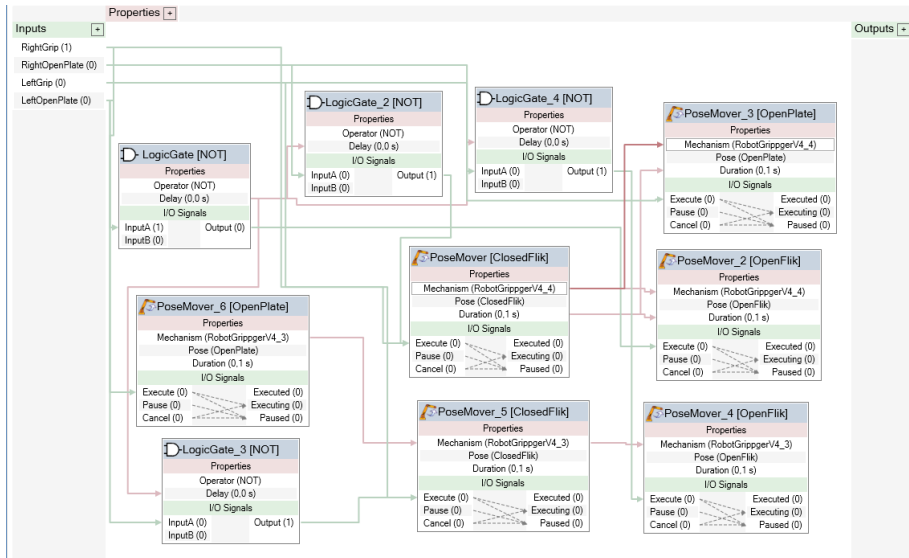
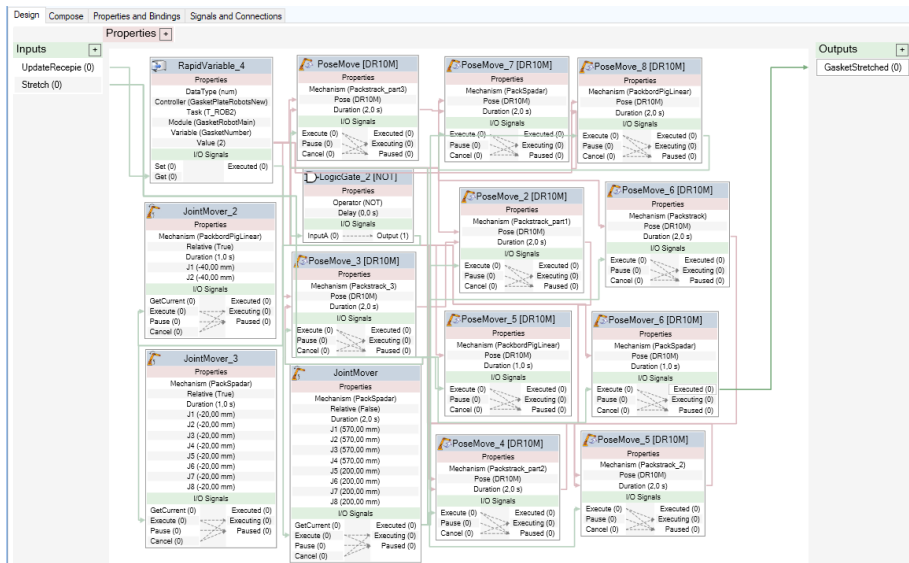


Table 1



Gasket infeed

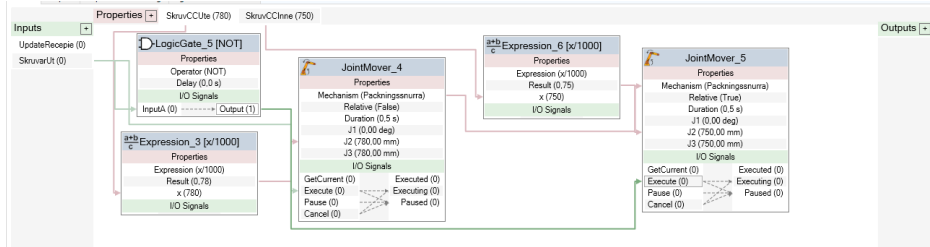
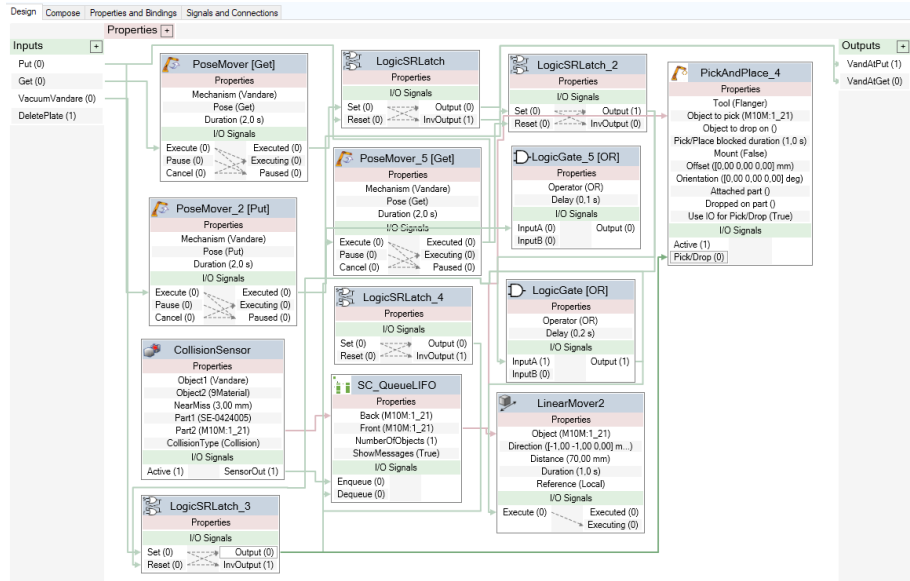
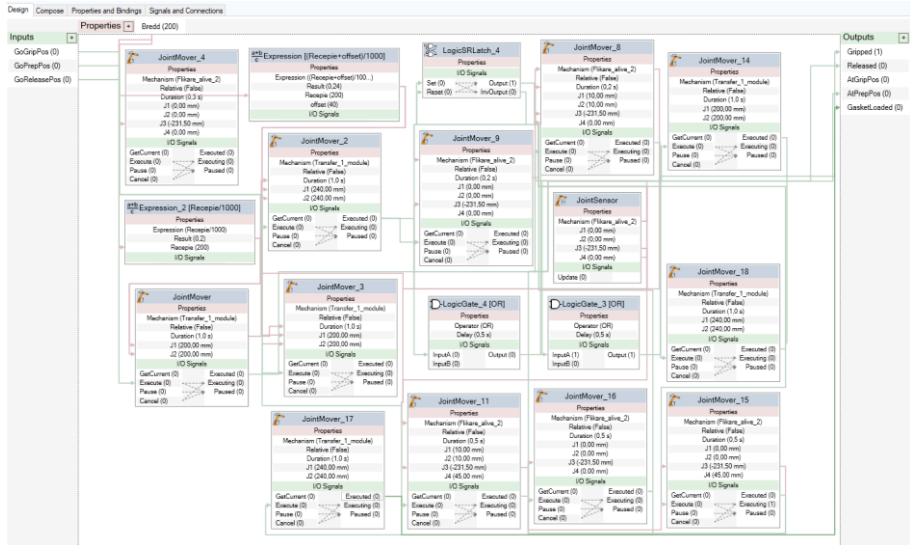


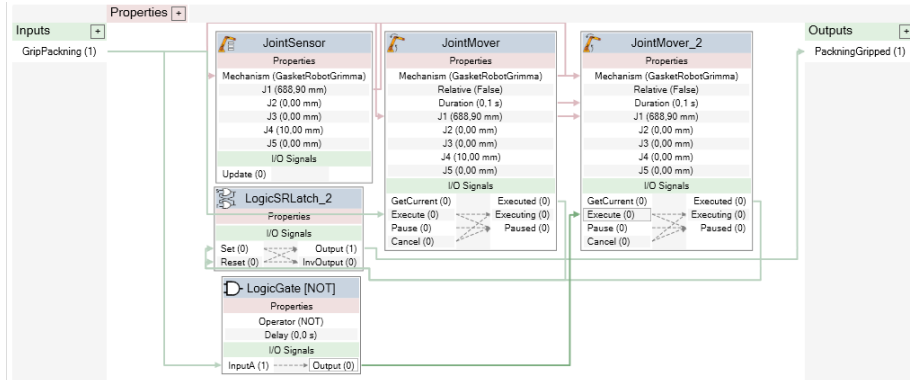
Plate turner



Portal 1

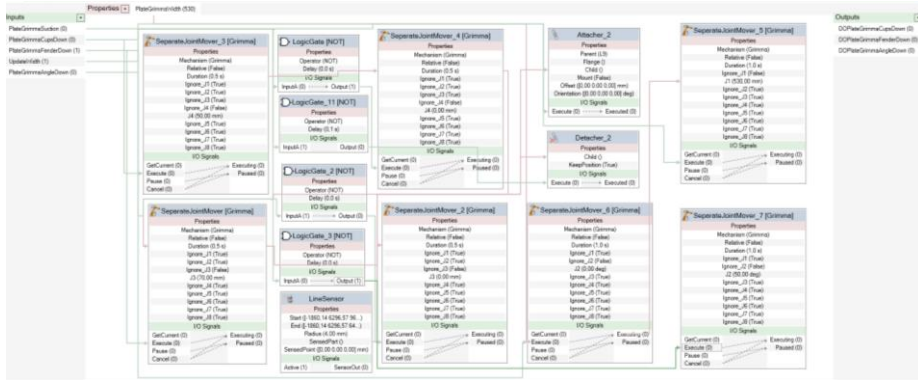


Gasket robot tool

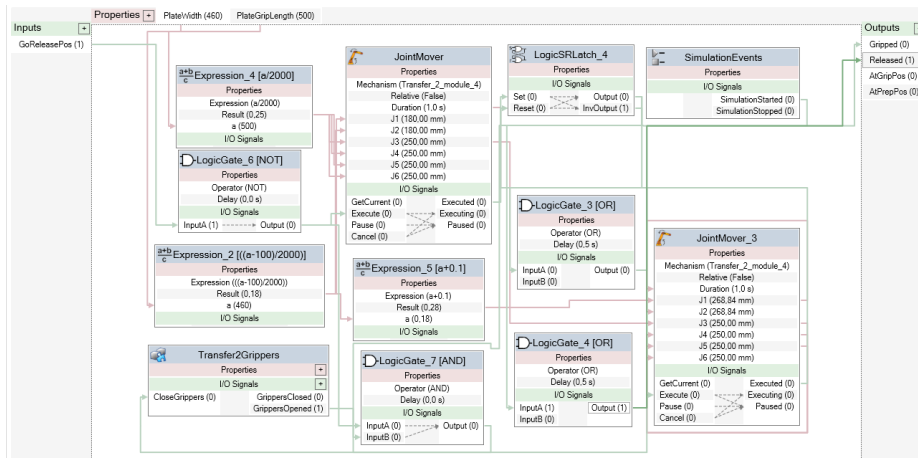


Appendix C. Smart components

Plate robot tool



Portal 2



Appendix D RobotStudio Simulated PLC

In this appendix, the Simulated PLC code is presented

Stepper and Actions for GasketRobot

```
MODULE GasketRobotSteps
  TASK PERS num stepper:=1;
  PROC main()
    init;
    !Loop that goes through the program
    WHILE TRUE DO
      CallByVar "action",stepper;
      CallByVar "step",stepper;
      incr stepper;
    ENDWHILE
  ENDPROC
  !Wait for speed to be set
  PROC step1()
    WaitUntil GRGOSpeed =
simgrgoSpeed;
  ENDPROC
  !Wait for job 2 "pickGasket" to be
  selected
  PROC step2()
    WaitUntil GRGOJobNo =
simGRGOJobNo;
  ENDPROC
  !Wait for job to start job started
  PROC step3()
    WaitUntil GRDOJobStarted = 1;
  ENDPROC
  !Wait for job to be done
  PROC step4()
    WaitUntil GRDOJobDone = 1;
  ENDPROC
  !Wait for gasket to be gripped
  PROC step5()
    WaitUntil simSCDIGasketGripped
=1;
  ENDPROC
  !Wait for job 3 dropGasket" to be
  selected
  PROC step6()
```

```
MODULE Actions
  !Initialization step
  PROC init()
    stepper:=1;
    Reset simSCDOSStretchGask;
    Reset simSCDOGripPackning;
  ENDPROC
  !Set speed values
  PROC action1()
    SETgo simgrgoSpeed,100;
    SETgo simgrgoRamp,100;
    Reset simGRDOGasketReleased;
  ENDPROC
  !Select job 2 "pickGasket"
  PROC action2()
    IF GRDIABSide=0 THEN
      Set simGRDOABSide;
    ELSE
      Reset simGRDOABSide;
    ENDIF
    SETGO simGRGOJobNo,2;
    SETGO
simGRGOGasketWidth,3000;
    SETGO simSCGOGasketLength,300;
    SETGO simGRGOGripOffset,400;
    SETGO simGRGOGripCC,11000/2;
  ENDPROC
  !Start pickGasket
  PROC action3()
    Set simGRDOSstartJob;
  ENDPROC
  !Reset job started
  PROC action4()
    Reset simGRDOSstartJob;
  ENDPROC
  !Set grip gasket
  PROC action5()
    Set simSCDOGripPackning;
```

<pre> WaitUntil GRGOJobNo = simGRGOJobNo; ENDPROC !Wait for job to start job started PROC step7() WaitUntil GRDOJobStarted = 1; ENDPROC PROC step8() ENDPROC !Wait until robot is in dropping position PROC step9() WaitUntil GRDOAtTable1 = 1; ENDPROC !Wait until gasket is dropped PROC step10() WaitUntil simSCDIGasketGripped =0; ENDPROC !Wait until it is OK to stretch gasket PROC step11() WaitUntil GRDOStretchOK = 1; ENDPROC !Wait until job is done PROC step12() WaitUntil GRDOJobDone = 1; ENDPROC PROC step13() ENDPROC PROC step14() ENDPROC PROC step15() stepper := 0; ENDPROC ENDMODULE </pre>	<pre> ENDPROC !Set job go to table PROC action6() SETGO simGRGOJobNo,3; ENDPROC !Set job started PROC action7() Set simGRDOStartJob; ENDPROC !Reset job started PROC action8() Reset simGRDOStartJob; ENDPROC PROC action9() ENDPROC !Release the gasket PROC action10() Reset simSCDOGripPackning; ENDPROC !Set gasket released PROC action11() Set simGRDOGasketReleased; ENDPROC !Start stretching of gasket PROC action12() Set simSCDOStretchGask; ENDPROC PROC action13() ENDPROC PROC action14() ENDPROC PROC action15() ENDPROC ENDMODULE </pre>
---	---

Stepper and Actions for Plate Robot

<pre> MODULE PlateRobotSteps TASK PERS num stepper:=1; PERS bool T2RdyPlate := FALSE; PERS bool P2RdyPlate := FALSE; PROC main() </pre>	<pre> MODULE PlateRobotAction !Initialization step PROC init() stepper:=1; Set simSCDOGrimmaFenderDown; </pre>
---	--

```

init;
!Loop that goes through the
program
WHILE TRUE DO
  CallByVar "action",stepper;
  CallByVar "step",stepper;
  incr stepper;
ENDWHILE
ENDPROC
!Wait for speed to be set
PROC step1()
  WaitUntil PRGOSpeed =
simPrgoSpeed;
ENDPROC
!Wait for job 2 "ScanForStack" to be
selected
PROC step2()
  WaitUntil PRGOJobNo =
simPRGOJobNo;
ENDPROC
!Wait for job to start
PROC step3()
  WaitUntil PRDOJobStarted = 1;
ENDPROC
!Wait for job to be done
PROC step4()
  WaitUntil PRDOJobDone = 1;
ENDPROC
!Wait for job 3 "ToStackOrig" to be
selected
PROC step5()
  WaitUntil PRGOJobNo =
simPRGOJobNo;
ENDPROC
!Wait for job to start
PROC step6()
  WaitUntil PRDOJobStarted = 1;
ENDPROC
!Wait for job to be done
PROC step7()
  WaitUntil PRDOJobDone = 1;
ENDPROC
!Wait for job 4 "PutPlateOnTurner"
to be selected
PROC step8()

```

```

WaitTime 0.5;
Reset simSCDOGrimmaCupsDown;
WaitTime 0.5;
Reset simSCDOGrimmaAngleDown;
setgo simSCGOPlateGrimmaWidth,530;
SETgo simPRgoSpeed,100;
SETgo simPRgoRamp,100;
SETGO simPRGOJobNo,1;
SETGO simPRGOxPosPlate,5000;
SETGO simPRGOyPosPlate,7300;
SETGO simPRGOzPosPlate,7000;
SETGO simPRGOPlateLength,8700;
SETGO simPRGOPlatewidth,3700;
SETGO simPRGOPlateThickness,30;
Reset simPRDOPlateReleased;
Set simPRDOStartJob;
WaitUntil
PRGOJobNo=simPRGOJobNo;
WaitUntil PRDOJobStarted=1;
Reset simPRDOStartJob;
WaitUntil PRDOJobDone=1;
ENDPROC
!Set speed values
PROC action1()
  SETgo simPRgoSpeed,100;
  SETgo simPRgoRamp,100;
  SETgo simPRGOacc,100;
ENDPROC
!Set rotation of plate and select job 2
"ScanForStack"
PROC action2()
  IF PRDIABSide=0 THEN
    Set simPRDOABSide;
  ELSE
    Reset simPRDOABSide;
  ENDIF
  SETGO simPRGOJobNo,2;
ENDPROC
!Start job 2 "ScanForStack"
PROC action3()
  Set simPRDOStartJob;
ENDPROC
!Reset job started
PROC action4()
  Reset simPRDOStartJob;

```

```

WaitUntil PRGOJobNo =
simPRGOJobNo;
ENDPROC
!Wait for job to start
PROC step9()
    WaitUntil PRDOJobStarted = 1;
ENDPROC
!Wait for job to be done
PROC step10()
    WaitUntil PRDOJobDone = 1;
ENDPROC
!
PROC step11()
    !WaitUntil Givare säger plåt är
nere
    WaitTime 1;
    WaitUntil PRGOJobNo =
simPRGOJobNo;
ENDPROC
!Wait for job to start
PROC step12()
    WaitUntil PRDOJobStarted = 1;
ENDPROC
!Wait for job to be done
PROC step13()
    WaitUntil PRDOJobDone = 1;
ENDPROC
!Wait until the table and portal is
ready for new plate
PROC step14()
    !Wait for suction
    WaitUntil T2RdyPlate =TRUE
AND P2RdyPlate = TRUE;
    WaitUntil PRGOJobNo =
simPRGOJobNo;
ENDPROC
!Wait for job to start
PROC step15()
    WaitUntil PRDOJobStarted = 1;
ENDPROC
!Wait for robot to be at Table 2
PROC step16()
    WaitUntil PRDOPlateRobotAtT3
= 1;
ENDPROC

```

```

ENDPROC
!Select job 3 "ToStackOrig"
PROC action5()
    !SET VALUE FROM SENSORS
    SETGO simPRGOJobNo,3;
ENDPROC
!Start job 3 "ToStackOrig"
PROC action6()
    Set simPRDOStartJob;
ENDPROC
!Reset job started
PROC action7()
    Reset simPRDOStartJob;
ENDPROC
!Start picking procedure and select job 4
"PutPlateOnTurne"
PROC action8()
    Set simSCDOGrimmaAngleDown;
    WaitTime 1;
    Reset simSCDOGrimmaFenderDown;
    WaitTime 0.5;
    Set simSCDOGrimmaCupsDown;
    WaitTime 1;
    Reset simSCDOGrimmaCupsDown;
    WaitTime 0.5;
    SETGO simPRGOJobNo,4;
ENDPROC
!Start job 4 "PutPlateOnTurner"
PROC action9()
    Set simPRDOStartJob;
ENDPROC
!Reset job started
PROC action10()
    Reset simPRDOStartJob;
ENDPROC
!Select job 5 "PickFromTurner"
PROC action11()
    SETGO simPRGOJobNo,5;
ENDPROC
!Start job 5 "PickFromTurner"
PROC action12()
    Set simPRDOStartJob;
ENDPROC
!Reset job started
PROC action13()

```

```

!Wait for suction release
PROC step17()
  !Wait for suction release
ENDPROC
!Wait for robot to not be at table 3
PROC step18()
  WaitUntil PRDOPlateRobotAtT3
= 0;
ENDPROC
!Wait for job to be done
PROC step19()
  WaitUntil PRDOJobDone = 1;
ENDPROC
!Reset the stepper
PROC step20()
  stepper:=0;
ENDPROC
ENDMODULE

```

```

Reset simPRDOStartJob;
ENDPROC
!Select job 7 "ToTable2"
PROC action14()
  !Set suction
  SETGO simPRGOJobNo,7;
ENDPROC
!Start job 7 "ToTable2"
PROC action15()
  Set simPRDOStartJob;
ENDPROC
!Reset job started
PROC action16()
  Reset simPRDOStartJob;
ENDPROC
!Release the suction
PROC action17()
  !Release suction
  WaitTime 1;
ENDPROC
!Set plate released handshake
PROC action18()
  Set simPRDOPlateReleased;
ENDPROC
!Reset plate released handshake
PROC action19()
  Reset simPRDOPlateReleased;
ENDPROC
!Tell portal that table is occupied
PROC action20()
  T2RdyPlate:=FALSE;
  P2RdyPlate:=FALSE;
ENDPROC
ENDMODULE

```

Stepper and Actions for Portal 1

```

MODULE Portal1Steps
  TASK PERS num stepper:=5;
  PERS bool T2RdyPlate := FALSE;
  PERS bool T2PlateGasketRdy :=
FALSE;
  PERS bool P1Moving := FALSE;

```

```

MODULE Portal1Action
  !Initialization step, move P1 above T2
  PROC init()
    stepper:=1;
    T2RdyPlate:=FALSE;
    WaitUntil pPlateOutOfTable=1;
    SetGO simP1GOxVal,0;

```

```

PROC main()
  init;
  WHILE TRUE DO
    CallByVar "action",stepper;
    CallByVar "step",stepper;
    incr stepper;
  ENDWHILE
ENDPROC
!Wait for photos to be taken of gasket
PROC step1()
  WaitUntil simSCDIPhotosTaken = 1;
ENDPROC
!Wait until P1 is above T1
PROC step2()
  WaitUntil simPIDIMoveDone = 0;
ENDPROC
!WaitUntil P1 has moved down
PROC step3()
  WaitUntil simPIDIMoveDone = 0;
ENDPROC
!WaitUntil P1 has moved up
!Wait until robot is gone from T2
PROC step4()
  WaitUntil simPIDIMoveDone = 0;
  WaitUntil T2RdyPlate = FALSE;
ENDPROC
!Wait until P1 has movbed to T2
PROC step5()
  WaitUntil simPIDIMoveDone = 0;
ENDPROC
!WaitUntil P1 has moved down
PROC step6()
  WaitUntil simPIDIMoveDone = 0;
ENDPROC
!WaitUntil P1 has moved up
PROC step7()
  WaitUntil simPIDIMoveDone = 0;
ENDPROC
PROC step8()
ENDPROC
PROC step9()
ENDPROC
PROC step10()
  stepper:=0;
ENDPROC

SetGO simP1GOzVal,65536-2000;
Set simPIDOMove;
Set simGRDOLeaveGasket;
WaitUntil simPIDIMoveDone=1;
Reset simPIDOMove;
ENDPROC
PROC action1()
ENDPROC
!Move Portal 1 above T1
PROC action2()
  SetGO simP1GOxVal,65536-
20000+4070;
  SetGO simP1GOzVal,65536-2000;
  Set simPIDOMove;
  P1Moving:=TRUE;
  WaitUntil simPIDIMoveDone=1;
  P1Moving:=FALSE;
  Reset simPIDOMove;
ENDPROC
!Allow plate robot to go to T2,
!Move down, grip gasket
PROC action3()
  T2RdyPlate:=TRUE;
  SetGO simP1GOxVal,65536-
20000+4070;
  SetGO simP1GOzVal,65536-550;
  Set simPIDOMove;
  WaitUntil simPIDIMoveDone=1;
  Reset simPIDOMove;
  PulseDO simSCDOP1GripGasket;
  WaitTime 3;
  Reset simSCDOSStretchGask;
ENDPROC
!Move up
PROC action4()
  SetGO simP1GOxVal,65536-
20000+4070;
  SetGO simP1GOzVal,65536-2000;
  Set simPIDOMove;
  WaitUntil simPIDIMoveDone=1;
  Reset simPIDOMove;
ENDPROC
!Move P1 above T2
PROC action5()
  SetGO simP1GOxVal,0;

```

ENDMODULE

```
SetGO simP1GOzVal,65536-2000;
Set simP1DOMove;
WaitUntil simP1DIMoveDone=1;
Set simGRDOLeaveGasket;
Reset simP1DOMove;
```

ENDPROC

!Move down, gasket the plate

PROC action6()

```
SetGO simP1GOxVal,0;
SetGO simP1GOzVal,65536-500;
Set simP1DOMove;
WaitUntil simP1DIMoveDone=1;
Reset simP1DOMove;
PulseDO simSCDOP1ReleaseGasket;
WaitTime 3;
```

ENDPROC

!Move up

PROC action7()

```
SetGO simP1GOxVal,0;
SetGO simP1GOzVal,65536-550;
Set simP1DOMove;
WaitUntil simP1DIMoveDone=1;
Reset simP1DOMove;
```

ENDPROC

!Move further up

PROC action8()

```
SetGO simP1GOxVal,0;
SetGO simP1GOzVal,65536-2000;
Set simP1DOMove;
WaitUntil simP1DIMoveDone=1;
Reset simP1DOMove;
T2PlateGasketRdy:=TRUE;
```

ENDPROC

PROC action9()

ENDPROC

PROC action10()

ENDPROC

ENDMODULE

Stepper and Actions for Gasket robot 1 (code of Gasket robot 2 is omitted as it is practically the same but mirrored)

```

MODULE GasketBSteps
TASK PERS num stepper:=1;
TASK PERS num fliknumber:=0;
PROC main()
  init;
  !Loop that goes through the
program
  WHILE TRUE DO
    CallByVar "action",stepper;
    CallByVar "step",stepper;
    incr stepper;
  ENDWHILE
ENDPROC
!Wait for speed to be set
PROC step1()
  WaitUntil
CBGOSpeed=simCBgoSpeed;
  WaitUntil
CBGOAcc=simCBgoacc;
  WaitUntil
CBGORamp=simCBgoramp;
ENDPROC
!Wait for job 2 "Gasket" to be
selected
PROC step2()
  WaitUntil
CBGOJobNo=simCBGOJobNo;
  !Waituntil plate ready for flik
ENDPROC
!Wait for job to start
PROC step3()
  WaitUntil CBDOJobStarted=1;
ENDPROC
!Wait for job to be done
PROC step4()
  WaitUntil CBDOJobDone=1;
ENDPROC
!If gaskets are left to be gasketed,
go back to step 1
PROC step5()
  incr fliknumber;
  IF fliknumber<6 THEN
    stepper:=1;
  ENDIF
ENDPROC

```

```

MODULE GasketBAction
PERS bool BReadyStep10:=FALSE;
PERS bool FReadyStep10:=FALSE;
!Initialization step
PROC init()
  stepper:=1;
  fliknumber:=0;
  SETgo simCBGOSpeed,100;
  SETgo simCBGORamp,100;
  SETgo simCBGOAcc,100;
ENDPROC
!Wait until P2 has delivered plate
PROC action1()
  BReadyStep10:=FALSE;
  WaitUntil simSCDIPlateReady=1;
  set simSCDOReleasehold;
  WaitTime 2;
  set simSCDOReleasegrip;

ENDPROC
!Set which gasket position to be gasketed
PROC action2()
  SETGO simCBGOPlateLength,830;
  SETGO simCBGOJobNo,2;
  TEST fliknumber
  CASE 0:
    SETGO simCBGOSetPointxPos,65536-
1268;
    SETGO simCBGOSetPointyPos,3080;
    SETGO simCBGOSetPointaPos,2642;
    SETGO simCBGOPVxPos,65536-1268;
    SETGO simCBGOPVyPos,3080;
    SETGO simCBGOPVaPos,2642;
  CASE 1:
    SETGO simCBGOSetPointxPos,771;
    SETGO simCBGOSetPointyPos,4375;
    SETGO simCBGOSetPointaPos,2700;
    SETGO simCBGOPVxPos,771;
    SETGO simCBGOPVyPos,4375;
    SETGO simCBGOPVaPos,2700;
  CASE 2:
    SETGO simCBGOSetPointxPos,65536-
587;
    SETGO simCBGOSetPointyPos,3559;
    SETGO simCBGOSetPointaPos,758;

```



```

PROC step6()
ENDPROC
!Wait for job 3 "ClearImage" to be
selected
PROC step7()
    WaitUntil
CBGOJobNo=simCBGOJobNo;
ENDPROC
!Wait for job to start
PROC step8()
    WaitUntil CBDOJobStarted=1;
ENDPROC
!Wait for job to be done
PROC step9()
    WaitUntil CBDOJobDone=1;
ENDPROC
!Wait until plate gripper is opened
PROC step10()
    WaitUntil simSCDIgripPlate=0;
ENDPROC
!Wait for job 4 "GripPlate" to be
selected
PROC step11()
    WaitUntil
CBGOJobNo=simCBGOJobNo;
ENDPROC
!Wait for job to start
PROC step12()
    WaitUntil CBDOJobStarted=1;
ENDPROC
!Wait for job to be done
PROC step13()
    WaitUntil CBDOJobDone=1;
ENDPROC
!Wait until P2 is gone
PROC step14()
    WaitUntil
simSCDIFreeForStack=1;
ENDPROC
!Wait for job 5 "plateToStack" to
be selected
PROC step15()
    WaitUntil
CBGOJobNo=simCBGOJobNo;
SETGO simCBGOPVxPos,65536-587;
SETGO simCBGOPVyPos,3559;
SETGO simCBGOPVaPos,758;
CASE 3:
    SETGO simCBGOSetPointxPos,1417;
    SETGO simCBGOSetPointyPos,4375;
    SETGO simCBGOSetPointaPos,2700;
    SETGO simCBGOPVxPos,1417;
    SETGO simCBGOPVyPos,4375;
    SETGO simCBGOPVaPos,2700;
CASE 4:
    SETGO simCBGOSetPointxPos,65536-
1202;
    SETGO simCBGOSetPointyPos,4375;
    SETGO simCBGOSetPointaPos,2700;
    SETGO simCBGOPVxPos,65536-1202;
    SETGO simCBGOPVyPos,4375;
    SETGO simCBGOPVaPos,2700;
CASE 5:
    SETGO simCBGOSetPointxPos,65536-
459;
    SETGO simCBGOSetPointyPos,4375;
    SETGO simCBGOSetPointaPos,2700;
    SETGO simCBGOPVxPos,65536-459;
    SETGO simCBGOPVyPos,4375;
    SETGO simCBGOPVaPos,2700;
ENDTEST
ENDPROC
!Start Gasket
PROC action3()
    Set simCBDOSTartJob;
ENDPROC
!Reset job started
PROC action4()
    Reset simCBDOSTartJob;
ENDPROC
PROC action5()
ENDPROC
PROC action6()
ENDPROC
!Select job 3 "ClearImage"
PROC action7()
    SETGO simCBGOJobNo,3;
ENDPROC
!Start job 3 "ClearImage"

```

```

ENDPROC
!Wait for job to start
PROC step16()
    WaitUntil CBDOJobStarted=1;
ENDPROC
!Wait for job to be done
PROC step17()
    WaitUntil CBDOJobDone=1;
ENDPROC
PROC step18()
ENDPROC
PROC step19()
ENDPROC
!Reset stepper
PROC step20()
    stepper:=0;
ENDPROC
ENDMODULE

```

```

PROC action8()
    Set simCBDOSTartJob;
ENDPROC
!Reset job started
PROC action9()
    Reset simCBDOSTartJob;
ENDPROC
!Wait for both robots to be done gasketing
PROC action10()
    BReadyStep10:=TRUE;
    WaitUntil FReadyStep10=TRUE;
ENDPROC
!Select job 4 "GripPlate"
PROC action11()
    SETGO simCBGOJobNo,4;
ENDPROC
!Start job 4 "GripPlate"
PROC action12()
    Set simCBDOSTartJob;
ENDPROC
!Reset job started
PROC action13()
    Reset simCBDOSTartJob;
ENDPROC
PROC action14()
ENDPROC
!Select job 5 "plateToStack"
PROC action15()
    SETGO simCBGOJobNo,5;
ENDPROC
!Start job 5 "plateToStack"
PROC action16()
    Set simCBDOSTartJob;
ENDPROC
!Reset job started
PROC action17()
    Reset simCBDOSTartJob;
ENDPROC
!Reset the gasket counter
PROC action18()
    fliknumber:=0;
ENDPROC
PROC action19()
ENDPROC
PROC action20()

```

