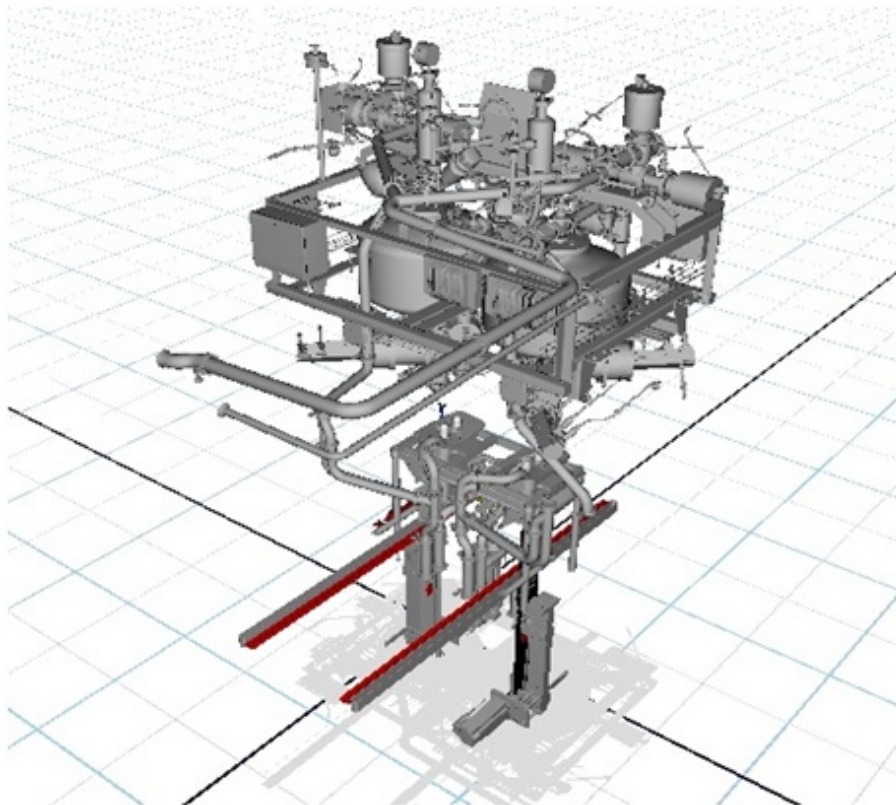


CHALMERS



Master Thesis Report

Testing and verifying PLC code with a virtual model of Tetra Pak Filling Machine

TOBIAS ERLANDSSON

MOHAMMAD MASHIUR RAHAMAN

Department of Signals and systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013
Report No. EX016/2013

Abstract

Virtual Commissioning is the way to verify control programs using a simulated model. It can be attainable in two distinctive approaches, Software-in-loop (SIL) and Hardware-in-Loop (HIL) simulation. In SIL simulation, both plant and control systems can be built virtually but in HIL simulation, simulated plant is controlled by real control systems. The second approach is studied in this thesis, where an Allen Bradley (AB) PLC is used to control the virtual model of a filling module in Tetra Rex 28(TR/28) machine from TetraPak and to check whether it is possible to get real time communication between AB PLC and virtual model. Experior Xcelgo software is used for virtualization of the filling model and the development of communication protocols to communicate with the external PLC. The TR/28 filling module is developed in a virtual environment along with dynamical behaviors by importing existing CAD drawings and write internal control logics. Real time communication is established between the AB PLC and the virtual model using EtherNet/IP-CIP communication protocols.

Keywords: Virtual Automation, Hardware in Loop Simulation, Virtual Commissioning, Experior Xcelgo, Rockwell Automation, RSLogix5000, Emulation

Acknowledgements

We would like to give special thank to:

Sathyamyla Kanthabhabhajeya, Doctoral Student at Signals and Systems,
Supervisor at Chalmers University of Technology.

Petter Falkman, Senior Lecturer at Signals and Systems, Examiner, Chalmers
University of Technology.

Bent Aksel Jørgensen, Director, CEO, Xcelgo A/S, Denmark

Steen Hother Jensen, Xcelgo A/S, Denmark.

Tobias Erlandsson & Mohammad Mashiur Rahaman, Göteborg 13/4-2013

Nomenclature

VC Virtual Commissioning

VR Virtual Reality

HiL Hardware in the Loop

HMI Human-Machine Interface

SiL Software in the Loop

HC Hybrid Commissioning

CAD Computer Aided Design

PLC Programmable Logic Controller

AB Allen Bradley

TR/28 Tetra Rex 28

ECU Electronic Control Unit

OSI Open System Interconnection

ODVA Open DeviceNet Vendors Association

Contents

1	Introduction	1
1.1	Overview	1
1.2	Background	2
1.3	Problem Description	2
1.4	Aim	3
1.5	Current Research	3
1.6	Delimitations	3
2	Technical Background	4
2.1	Simulation and Emulation	4
2.2	Virtual Commissioning	5
2.3	Programmable Logic Controllers	7
2.4	Communication with Ethernet/IP-CIP	9
3	Description About Software and Products	11
3.1	Xcelgo Experior	11
3.2	Rockwell Automation	14
4	Methodology	16
4.1	Development of Model	16
4.2	Establishment of Communication	19
5	Results & Discussions	22
6	Conclusion & Future Work	25

List of Figures

2.1	Emulation Concept Overview	5
2.2	Verification of real control programs in SIL simulation	6
2.3	HIL simulation to verify control programs	7
2.4	PLC Architecture	8
2.5	EtherNet/IP-CIP communication structures	10
3.1	Functionalities of Xcelgo Experior	11
3.2	Experior's own catalog for physics simulation	12
3.3	Communication protocols in Experior to communicate with external PLC software	13
3.4	Experior control panel to check the control logics	14
3.5	Display of production output	14
4.1	Building a new Project in Experior	16
4.2	Positioning of CAD files by drag and drop method to get the desired co-ordinates	17
4.3	Lift and volume profile	18
4.4	Communication protocol is connected to communicate with Allen Bradley PLC and Experior	19
4.5	Allocating IP address, slot number and types of communication	20
4.6	Ladder diagram written in RSLogix 5000 to run virtual model	20
4.7	Adjusting PLC tags that are imported from .CSV file	21
5.1	Virtual model of TR/28 machine	22
5.2	Synchronous communication between Xcelgo and Allen Bradley PLC	23

1 Introduction

This chapter serves as a foundation for understanding and interpreting of the contents of this report including basic overview of theory that is used, background of Tetra Pak's current research, the needs for conducting this thesis with specific aim and the results that are sought after.

1.1 Overview

Most of the modern manufacturing industries consist of highly automated workstations. The advantages are often accredited of being automated include higher production rates and increase productivity, efficient use of materials, product quality, improved safety, reduced dependencies on labors and reduced hours of factory[1]. However, modifying and controlling these workstations with respect to the change in products and its characteristics is a great challenge. So it is indispensable for any industry to design these control systems in the early design phase without hampering current production, to ensure the benefits of automatic workstations before commissioning to the final production. Virtual simulation is advantageous in the early design phase of a production system, for not only to measure the production output but also the physical validity and efficiency of co-working machines without damaging the real equipments and control programs. Some important factors that influence the simulation for example, the layout of the systems, the flow of production, and the associated co-working parts are required to be accurate while modeling it to the virtual environment. Simulation of a virtual model often termed as Virtual Reality (VR) can mitigate this risk by adding actual physics associated with the system. Hence the use of control points from where controlling operations are needed to be performed is much more visualized in the simulation environment.

Verification of a control systems is necessary to check whether the control programs are feasible to acquaint the required behaviors of machines or not. Formal verification provides a means for verifying a controller implementation over all possible operating conditions by using discrete mathematics before it is commissioned on the real system[2]. There are other ways to verify and validate the Programmable Logic Controller(PLC) programs. According to Auinger et al (1999), verification can be arranged in four different basic systems of configurations

- Real plant & real control system(traditional)
- Simulated plant & real control system(hardware in loop simulation)
- Real plant & simulated control system(reality in loop)
- Simulated plant & simulated control system(complete virtual commissioning)

Among these four ways of verification, complete virtual commissioning and hardware in loop simulation are of great interest. In [4] these two ways of verifications are categorized from Virtual Commissioning. In both systems the plant models are built in virtual environment, so there is no risk of sabotaging the real equipments, if these control programs are tested in system level of testing. The way of using the control system differs between these two configurations. Obviously, complete virtual commissioning is

much more suitable since both plants and control systems can be build in simulation environment. But hardware in loop simulation is necessary where real control feedback from real PLC verification is important to run the simulated model in closed loop system.

1.2 Background

Tetra Pak, a world leading company to provide food processing and packaging solution is interested to shorten the development time from initial phase with the project needs to the final phase with the delivery of optimized and verified product. The existing workflow process in Tetra Pak mainly involves testing on the physical level. These are time consuming since redesigning and changing the tools and machineries are necessary each time test conditions are changed. There is also some risk of failures when verifying the control programs in this way. In their current research there is no finalised solution yet to verify control programs using virtual commissioning system. So emphasizing on virtual environment for this development project is of great interest to Tetra Pak and Chalmers University of Technology, which can provide the solution for sequential activities by testing these activities parallel and verify the control programs in virtual reality.[5]

1.3 Problem Description

The machine that has been evaluated for this thesis work is Tetra Rex 28 (TR/28). The TR/28 is a filling machine for chilled products with a high capacity and high performance. It has been designed to carry out different sizes of packages ranging from 237 ml to 1.136 ml with a capacity of 14,000 packages per hour[5]. This machine has two identical conveyors parallel to each other that are operated by electric servo motor and each conveyor has two action points where two packages are lifted by mechanical lifter in each step of every cycle. These two conveyors are symmetrical and identical and operations in every single step are same. The mechanical lifters are operated by cam driven servo motor for lifting up and down the lifters with packages. The choice of using servo motor is for getting accurate positioning with precise motion. The lifter follows a motion profile by moving the lift up faster than the lift's down motion and fill the carton while lift is moving down. So there is need to control the servo axis for precise positioning and motions. In general servo axis can be controlled by getting feedback that determines the accurate positions. And the whole system including Servo axis is controlled by Allen Bradley (AB) PLC. The PLC has an Electronic Control Unit (ECU) in its closed loop system with the hardware.

In Virtual Commissioning using AB PLC, controlling of motion from servo axis is great challenge. AB has the capability of capturing and calculating the high speed motion from servo motor and able to correct its current position. It is necessary to conduct Hardware in Loop simulation when comes virtual commissioning with AB, where simulated plant model is controlled by real AB PLC. Modeling a servo motor in virtual environment can not fulfill all the requirements needed in an AB PLC. And also

there is another challenge to simulate virtual model in real time and there might be communication delay between the AB PLC and the virtual model.

1.4 Aim

The aim of this thesis is to build a virtual model for TR/28 and verify the PLC code to run this virtual model in hardware in loop simulation.

1.5 Current Research

According to the current needs of Tetra Pak, a research has been carried out at Chalmers University of Technology to get a solution whether it is possible to work with motion control and hardware in loop simulation in AB PLC. The software that is used for this thesis is Xcelgo Experior that has the capability to work with simulation, emulation and virtual automation projects[6]. The plant model is built in Experior window by importing Pro/Engineering Computer Aided Design (CAD)-files to form virtual reality and control points are identified in virtual model. Digital Input/ Output (I/O) signals are developed for establishing communication between Experior and AB PLC. Control programs are written using RSLogix 5000 in form of Ladder Diagram (LD) to control the TR/28 filling module. Verification of these PLC programs is carried out in Hardware in Loop (HIL) Simulation. Because of complexity in virtual servo drive when it is considered in terms of feedback with real PLC, the motion of servo motor is controlled in C# scripting. Hence control programs are verified and communication delay between PLC and simulation model is observed.

1.6 Delimitations

There are some factors that have not been considered in this thesis. Although TR/28 is completely automated and comprises of several modules, this thesis is restricted only with the filling module. The outputs that can be possible after controlling the motion and total cost needed to implement this new system in existing pattern are exempted from this thesis.

2 Technical Background

This thesis required knowledge on Virtual Manufacturing systems, detail descriptions about PLC, HIL simulation, Communication with Ethernet Industrial Protocol - Common Industrial Protocol (EtherNet/IP-CIP), control programs and verification of these programs.

2.1 Simulation and Emulation

In computer science simulation and emulation are ways to imitate a system by using computer software and replace hardware equipments. Both try to create the same output from the same input that the original system does. The purpose of simulation and emulation models are almost the same although the way of using models are different.

Simulation is used as an experimentation tool to find optimal solutions for system. It does this with processes that are not necessarily similar to those of the system. It is the output that is similar to the real systems output. Simulation can be used to calculate flow and logistics problems in production and distribution of products in order to, for example find bottlenecks. It is possible to get production optimization faster than traditional ways by doing multiple runs with different inputs in a fraction of the time on a computer. In fact, simulation can also be used when the system is not even existing to test how the system should work and to be able to find necessary changes to the design of the system. When making conclusions about what the simulation show, one must always take uncertainties into consideration. Errors may pile up to, and in the end, create a much bigger deviation that the inputs may suggest. Hence the simulation model is not necessarily similar to the real system, the output can deviate and conclusions should take this into consideration.

Emulation tries to exactly match the output of the system and produce the process as similar to those of the system as possible. The purpose of emulating is to substitute the real system, eg. hardware with software doing the same thing. The benefits of doing this is mostly that experimenting can be done without damaging expensive equipment and still get a feedback that is reliable. The benefit of less time consumption is however not as great as in simulation because of the system runs in real-time and surrounding equipment could still be parts of the real system and therefore operates at the same speed as it would in the real system. Changes to the model are however much less time consuming and of course also costs less. Emulation models is often used as a tool for verification of performance of parts of the real system without any extensive risk.[7] Figure 2.1 describes the basic difference between simulation and emulation while working with PLC.

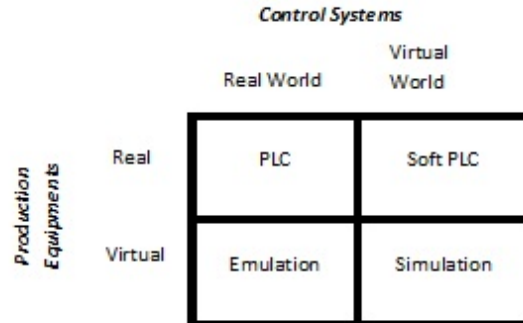


Figure 2.1: Emulation Concept Overview

2.2 Virtual Commissioning

Virtual commissioning (VC) is the way to verify and validate control programs by using computerized simulated plant instead of using real machineries before commissioning to the final production. It can also be described as “Verification in simulation model” where a simulation model of manufacturing system is used to allow commissioning before building the system[8]. All the tasks are modeled virtually including Human Machine Interface (HMI) tools so there is no risk of endangering lives, machine equipments and a real time communication is possible between PLC and virtual model. So it is possible to test and validate the control programs to provide decision making support from a plethora of decisions without implementing the PLC to control the real plant. One of the major requirements to get benefit from virtual commissioning is time needed to build and verify virtual model must be shorter than the traditional way of verification.

In VC system, several important requirements need to be considered as necessary for industrial design. Here the requirements are real control codes for controlling a simulated plant and HMI tools should be remained unchanged compared to its original codes, the plant itself needs to be depicted the real engineering behaviors in simulated model[9]. Mechatronic systems provide best solution for modeling of mechanical and electrical behaviors. It is the integration of complex mathematical algorithm describing the physics of the elements or subsystems being modeled. Virtual simulation is useful to represent the sensors and actuators in an extensive and comprehensive manner. But the model might not behave in the same way as real world system, due to lack of understanding about the real systems, inability to model correctly the problem, inability to translate or code correctly the conceptual model to the computerized model or lack of simulation knowledge, unclear simulation objectives, difficulties to animate the complex machineries etc[10]. So a simulated plant model’s fidelity must match its intended use where most typical parts i.e. sensors and actuators can be modeled and have the capability of giving and receiving signals with the PLC. In virtual commissioning, along with simulated

plant, the machine control, including input and output signals and the controlling of these signals connection are also modeled.

VC has been studied for more than ten years. According to Makris et al. and Bangsow et al. VC can be subdivided into two categories, Software in Loop (SIL) Simulation and HIL simulation. In SIL simulation, virtual plant model is controlled by using soft PLC by downloading the source controllers into virtual controller in PC based controller. And in HIL simulation, is also termed as “Soft Commissioning”[12], is the technique to verify or validate the control programs where embedded systems or simulated plants model can be controlled by real hardwired control system . Both are able to deliver real time simulation for controlling with using appropriate communication protocols. The simulation model provides all the process signals in real time that are next converted into D/A module (Digital to Analog) and supplied to the controller as a voltage, with which control signals are produced by the controller and supplied via A/D converters (Analog to Digital) to run the simulated model[13]. But there should have possibility of being some differences or error between real plant and virtual simulator. This could lead to far deviation from simulated tested result to actual model and could also have the possibilities to damage real machine equipments even though control programs show good result in simulated plant. So hybrid commissioning, a new approach has been described in [14], where three stages of commissioning occurs. Here, verification is completed using HIL or SIL simulation, then mixture is created between virtual and real plant, and finally commissioning with real plant is achieved.

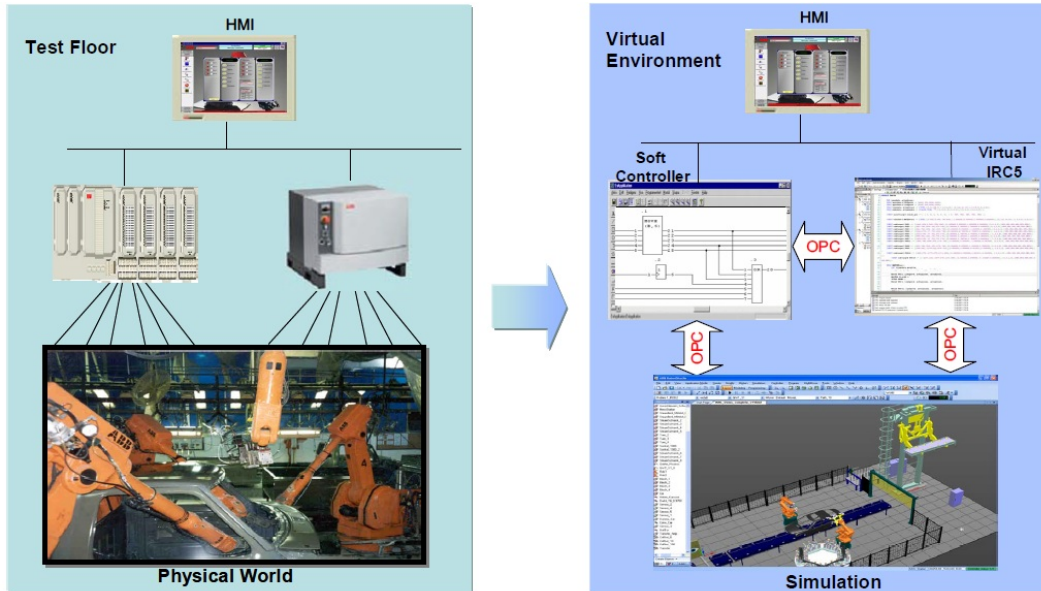


Figure 2.2: Verification of real control programs in SIL simulation[9]

Figure 2.2 represents VC where robots from ABB and their dynamic systems are built with virtual 3D modeling. Here HMI, PLC controller and IRC5 (fifth generation robot controllers) are modeled in integrated computer graphics and are used to verify simulated model that corresponds to the behaviors of real robot. Figure 2.3 shows, ECU(Hard PLC) is interfaced with the host PC in a closed loop system and gives control feedback to virtual environment.

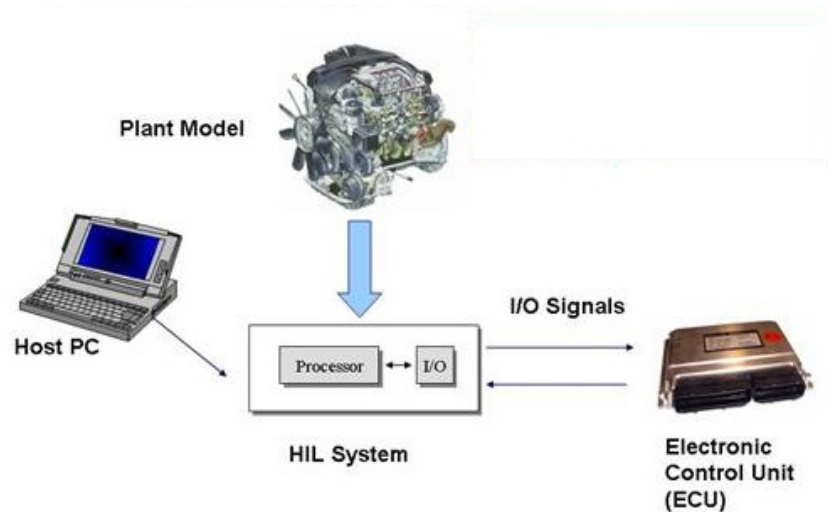


Figure 2.3: HIL simulation to verify control programs[13]

There are several simulation software that provide the solution to work with “virtual commissioning”. Most of the software provides special functionalities for working with the design of actuators and sensors systems in virtual environments, uses built in libraries to communicate with models and real PLC to compete in markets with competitors. Some of them uses physics engine of the computer graphics to run simulated complex and bulky machineries fast enough to comply with the output signals from controller. Typical simulation software that exhibits most these behaviors described above are WinMOD, Process Simulate, Invision, Delmia, 3D Emulator, Xcelgo Experior etc.

2.3 Programmable Logic Controllers

A Programmable Logic Controller (PLC) is a computer based control system to control the devices or actuators as an output after getting signals from the input devices according to the decisions based on control program. It was first introduced in 1960s/1970s in automotive industries. Before that time automatic manufacturing was achieved by using of relays, cam timers drum sequencers and closed loop controllers. These controllers composed of hundreds or even thousands of relays to control manufacturing units with high time consumption and expensive to make changes on. The speed of the operation can be greatly enhanced by replacing the relays and timers in a single PLC program. And the main benefit in using a PLC is the ability to change and replicate the operation

or process while collecting and communicating vital information.

PLC hardware consists of Central Processing Unit (CPU), Random Access Memory (RAM), Read Only Memory (ROM), power supply unit and other devices. The CPU, the core of the PLC has microprocessors to allow arithmetic operations, logical operations, local area network, computer interface etc. Figure 2.4 represents PLC controller describing every elements including I/O terminals, CPU, power supply etc.

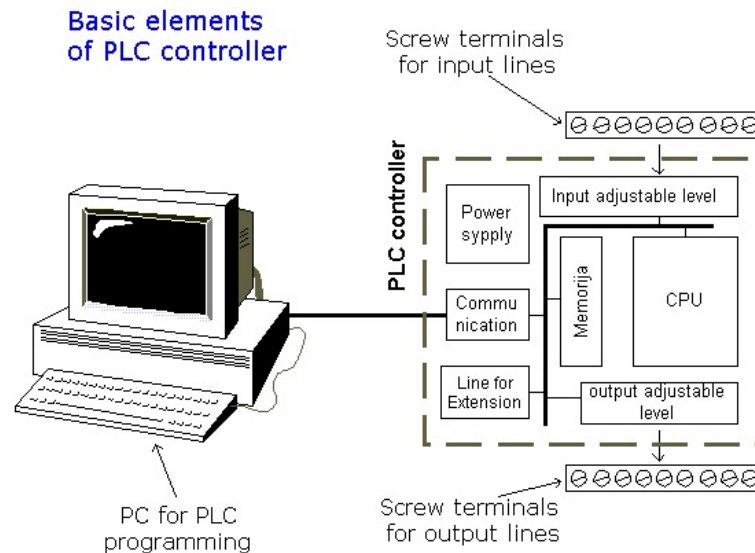


Figure 2.4: PLC Architecture[15]

According to IEC (International Electromechanical Commission IEC 61131-3) programming of PLC includes five different languages like Sequential Function Chart (SFC), Functional Function Block (FFD), Ladder Diagram (LD), Structured Text (ST) and Instruction List (IL). FFD and LD are graphical languages while ST and LT are textual languages and SFC is a graphical method, which represents the functions of a sequential automated system as a sequence of steps and transitions.[16]

Among these Ladder Diagram (LD) program is most widely used because of its simplicity and idea close to relay logic. It represents a programming language by graphical way which can be acted as a transition from hard wired circuits to soft wired circuits. It consists of two vertical lines to indicate power supply from left to right and rung in horizontal axis symbolizing control circuit. Program is executed by checking all the input conditions to the left from top to bottom and executing the logic to the output to the right. An example of this can be seen in Figure 4.6. Each rung might have inputs, outputs where both can be normally open or normally close, timer to give the precision in time, counter to count the number of events occurred etc. Input modules act as electrical connectors between external field devices and internal processor of PLC. External field devices such as start/stop push buttons, limit switch, proximity switch, and internal bits give output as electrical signal to internal processor of PLC and these are used as PLC inputs to get energizing associated output in ladder logic. Output module is used to

communicate between internal processor of PLC and external field devices where PLC provides signals.

2.4 Communication with Ethernet/IP-CIP

To be able to control the virtual model with a PLC a connection is needed between them. There are several ways of PLC communication. Among these the most commonly used is EtherNet/IP connection between the PC with the emulation software running the model or external microprocessor to run the field devices and the PLC. For complete VC, PLC software is installed for programming the PLC and that communicates with the simulation software in same CPU. In that case internal busses are used and able to interchange bits or signals with the virtual PLC to communicate with emulation software. But for HIL simulation, it is necessary to develop an external communication protocol to communicate with the real PLC and the virtual model.[17]

Ethernet is the most popular and widely deployed local area network (LAN) technology. Ethernet protocol operates in the physical and data link layer of the Open System Interconnection (OSI) networking stack. In Ethernet a single communication channel is shared by the Ethernet devices in the network. The data are exchanged in here as a frame. Usually, Broadcast mechanism is used in Ethernet technology. Multicasting is also possible in Ethernet where a packet is sent to a subset of the hosts. Ethernet protocol provides some guidelines to construct frames, these contains the source and the destination address. Destination address is used to identify the recipient of the communication. When an Ethernet device transmits a frame, it transmits to all other nodes. All the nodes check the destination address in the frame. If any node finds the destination address is the same of its own, the node pick the frame and send to upper layer for further processing. All the other nodes discard the message. Ethernet uses the carrier sense multiple access (CSMA) with collision detection as a media access protocol. In career sensing, each host in the network sense the career if any data is being transmitted over the Ethernet cable, the host stops transmitting and try to send the data after an interval.

EtherNet/IP is special typed communication protocol developed by Rockwell automation. It was first introduced in 2001 and is managed by Open DeviceNet Vendors Association (ODVA) to provide best industrial network solution for manufacturing automation[18]. It has basic difference compared to enterprise level of Ethernet communication, where Ethernet uses physical, data link and network layers of the OSI model but EtherNet/IP uses CIP(Common Industrial Protocols) to the upper three layers (Application, Presentation and Session layer), where CIP is object oriented protocol with having attributes, services and behaviors. Using of CIP has great advantage in communication protocol, it has special functionalities of messages and services in manufacturing automation applications, these are control, safety, synchronization, motion, configuration and information and also make it possible to use manufacturing applications over internet and common network shared activities[19]. Figure 2.5, represents OSI model describing EtherNet/IP-CIP communication, where usage of different layers with different functionalities are described.

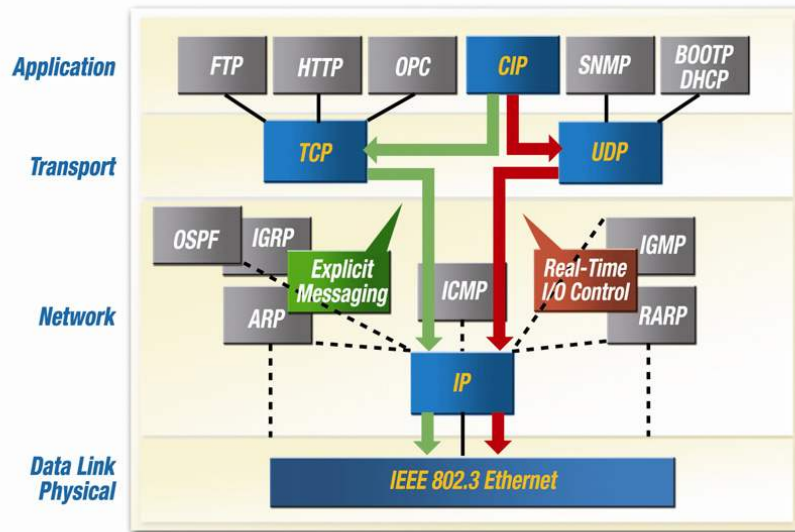


Figure 2.5: EtherNet/IP-CIP communication structures[18]

EtherNet/IP-CIP allows two types of messaging connections, one is implicit messaging and another is explicit messaging. Implicit messaging is often termed as real time messaging which basically transfers CIP I/O data to EtherNet/IP. And for implicit messaging it uses UDP (user datagram protocol) over IP with UDP port number 2222. For explicit messaging it uses Transmission Control Protocol (TCP) over Internet Protocol (IP) with port number 44818, where CIP messages of unicast, multicast and broadcast communication via TCP is established to facilitate request-response transactions. CIP uses producer consumer network model to provide exchanging of application information between sending device (producer) and receiving device (consumer) by transmitting data by once without transmitting it every time from a single server to multiple clients.[18]

3 Description About Software and Products

This chapter represents two major softwares. In this thesis Xcelgo Experior is used for developing a model with communication protocols and the other is AB PLC to write control programs and control the virtual model for this entire project. This section includes introduction to these software and some basic functionalities that they compete for developing model, control programs and communicating with each others.

3.1 Xcelgo Experior

Experior has a strong ambition to provide state of the art software for simulation, emulation and virtual commissioning. Xcelgo is a small Danish company situated in a small town Ry and has a vision to deliver best return to customers in terms of virtual automation and commissioning. Experior virtual automation platform has strong functionality not only with the virtual commissioning but also 3D visualization, physics with discrete simulation and emulation, training and optimization where user specific applications can be developed[6]. Figure 3.1 shows the functionalities of Experior Xcelgo as a virtual commissioning software.

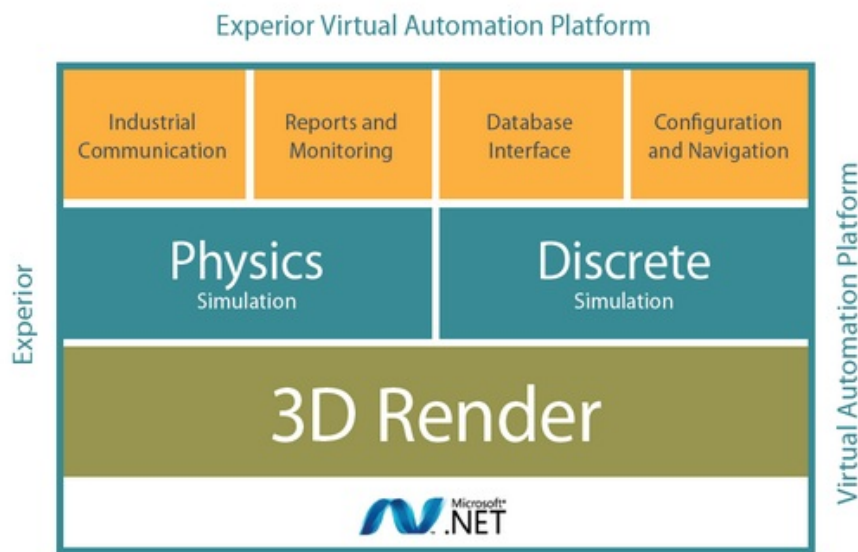


Figure 3.1: Functionalities of Xcelgo Experior

Experior has its own catalogs for 3D modeling and simulation; it consists of basic conveyors, feeders, mechanical pushers, conveyor gates, electric lamps, different types of sensors (photo eye, barcode scanner, x-ray etc), electric motors etc. Each catalog is resourceful to describe the behaviors of real equipments. Their dynamic system is skilful that they can even calculate the dynamic friction between different objects. Experior's

built in catalog has enough resources to show up the flow of processes in a manufacturing line that needs to be controlled, for example sorting of bags in airport with manikins movement and storing facilities, thus can form virtual reality in simulation environment. So with this distinctive Experior catalogs, well-designed assembly can be built for discrete simulation and testing. Figure 3.2 represents Experior built in catalogs of different kinds that can be usable from their library for simulation in virtual model.

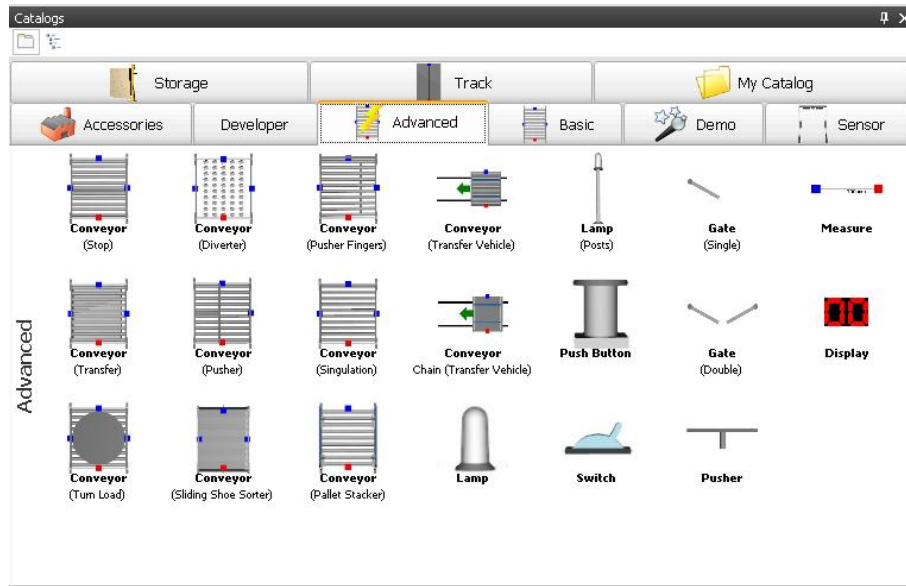


Figure 3.2: Experior’s own catalog for physics simulation

In order to represent machinery equipments in computer display and to get virtual reality users need to build users own catalog. Modern manufacturing industries comprise of different types of machinery equipments, most of them are bulky in nature and make simulation difficulties when it comes to virtual reality. Virtual reality can be developed by importing the CAD models in virtual environment, integrating them and adding dynamics to different rigid parts when necessary. Rendering is the process of imitation in real time simulation, where dynamics in different rigid parts, collision detection, trajectories are appended and required giving faster response in order to withstand with real time environment. 3D rendering in Experior provides that solution by importing existing CAD drawing of machinery equipment and converting to 3D graphical images. Its window consists of 3-directional vector to indicate the direction in virtual floor with proper scaling that depicts the actuality. Using of NVIDIA PhysX in Experior can simulate the mechanics of rigid bodies, soft elements and particles in an intuitive, graphical way. PhysX is a specialized powerful tool provided by NVIDIA that can compute the physical effect and has the strong benefit of using graphics processing unit (GPU) of the NVIDIA card instead of using traditional CPU to get faster response.

Experior uses application program interface (API) in .NET framework(C#) platform for the development, deployment and execution of the distributed systems . C# is

a object oriented programming language which inhabits with spacious features for development of projects and some other external applications. There are many Dynamic Link Library (DLL)-files in this API to describe the functionalities, these contain hidden code and data that can be useful to several program at the same time. Experior API usually consists of widespread using of classes, strong library to support different logical operators, methods and properties for extensive use of different functions to support development of models etc.

Communication can be established between Experior and PLC software by creating I/Os in virtual model. As shown in Figure 3.3, Experior supports several communication protocols to communicate with external PLC software. In some cases communication can be done without using external hardware, to perform pure virtual commissioning. Experior uses several plug-in for real time communications.

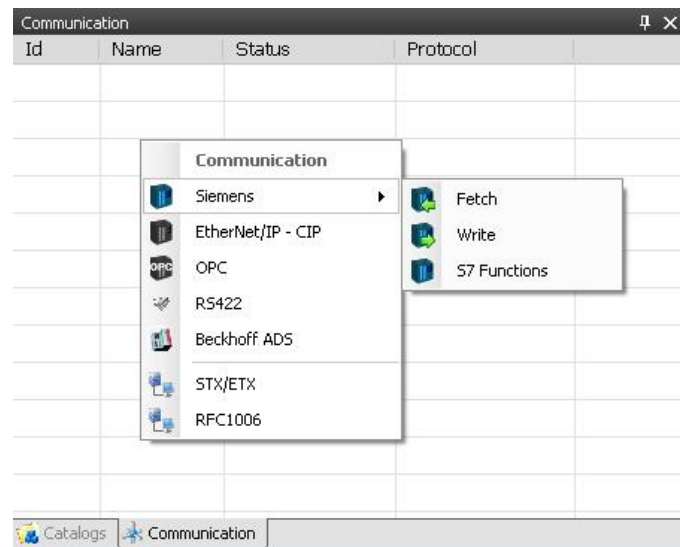


Figure 3.3: Communication protocols in Experior to communicate with external PLC software

Experior has control panel to test the control logics that is developed in scripting without using real PLC program. This is usually achievable by using Siemens Fetch/Write in the communication panel. Appropriate bits and bytes are allocated between control panel's components and virtual PLC inputs/outputs. Push button is used to send bits/bytes to the model where actuators system can be activated. Lamps receive signals from the sensors and blink while sensors are activated. Figure 3.4 describes the virtual control panel.



Figure 3.4: Experi control panel to check the control logics

Experi uses an analysis window to display the output. Here the total output can be displayed as load and verifies output values in number for discrete systems and optimal output can be calculated by analysis and changing the values in different control points. Figure 3.5 shows a blank display of showing production output, can be usable for checking the production output.

Statistics			
	Total	Count	Max.
Loads	0	0	0

Figure 3.5: Display of production output

There is another option for monitoring this whole simulation process. By capturing pictures and recording videos of operations, systems can be improved in offline mode by changing different values in different control positions. It is also possible to add sound effects on animated video for better presentation.

3.2 Rockwell Automation

Rockwell Automation is one of the world's biggest providers of industrial automation products. Their range of products and services reach from PLC hardware to software and information solutions. Rockwell was founded as Compression Rheostat Company in 1903 by Lynde Bradley and Stanton Allen. They changed the name to Allen-Bradley Company in 1910 and during the following decades they grew to an international company. Rockwell International bought Allen-Bradley in 1985 and Rockwell Automation

was formed. Today Rockwell Automation is active in more than 80 countries worldwide and has over 21 000 employees[20].

3.2.1 Rockwell Software - RSLogix 5000

Rockwell Software is the brand Rockwell Automation uses for their software applications. RSLogix 5000 is a software developed by Rockwell Software to be able to control and configure their hardware. RSLogix 5000 offers an IEC61131-3 compliant interface, symbolic programming with structures and arrays and a comprehensive instruction set that serves many types of applications [20]. The aim for RSLogix 5000 is to make it easier to develop complex control solutions and also offers greater access to real-time information. The software includes editors for ladder logic, structured text, function block diagram and sequential function chart which makes it possible to create programs for every level of complexity. RSLogix 5000 has an detailed walkthrough and guide on how to maximize the use of the software and makes it easier even for a new user to reach the desired outcome.

3.2.2 Allen Bradley PLC

The brand Allen Bradley is still used for Rockwell Automations hardware such as PLC, I/O modules, motors and drives, motor and motion controls, power supplies, relays, sensors, safety equipment to name a few. Their hardware can be used in every imaginable equipment and systems and for every level of complexity. They are usually managed and supported by software developed by Rockwell Software.[20]

4 Methodology

This chapter will describe the method of TR/28 filling module and its associated control functions are built in virtual environment using Xcelgo Experior software, the constraints and the assumptions that are made during this development.

4.1 Development of Model

To simulate TR/28 in virtual environment, 3D CAD drawings of TR/28 needs to be imported in Experior's window. User's own catalog has been built instead of using existing catalogs of Experior for this 3D emulation. This work has been done by opening a new project in .NET framework(C#). After installing Experior and .NET framework(C#) on the same CPU it will show the option "*Experior Catalog Template*" if a new project is selected to build. Figure 4.1 shows the way to selection of new project using .NET framework(C#).

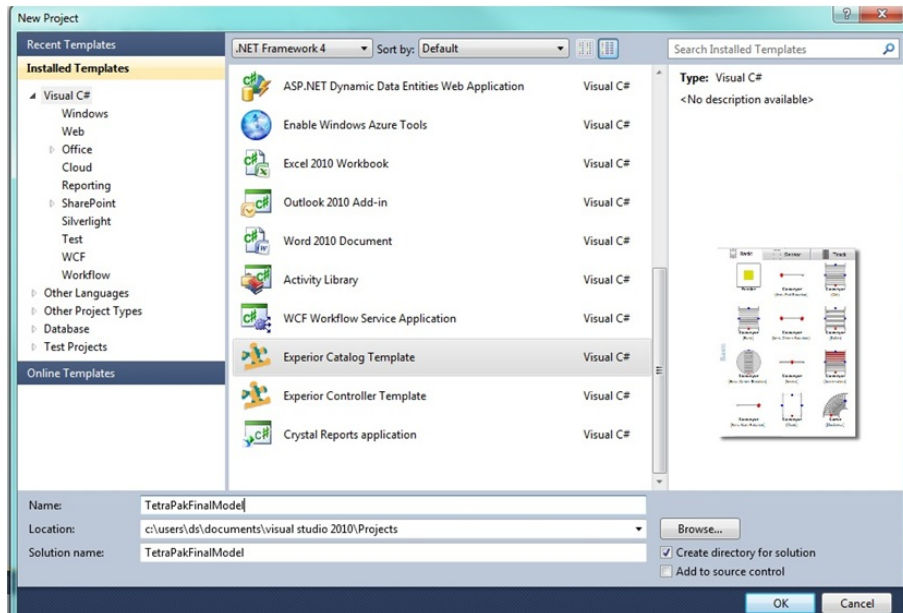


Figure 4.1: Building a new Project in Experior

After building a new project in .NET framework, the directory of the project execution file has been changed. This is accomplished by changing the project properties from C# toolbar where output path is set as the same folder as Experior's installed folder to get access of the necessary references or libraries from API and external Experior executable program is used to debug this project.

ProE CAD files of TR/28 machine are given from TetraPak. Because the thesis is limited to only work with the filling part of this machine, only the necessary CAD files with extension of *.Product* are selected since they are combination of different Part files. 3D rendering can be done in Experior by importing the CAD files in its C# command

window but only the format of *Microsoft DirectX* (extension with .x) is compatible. So *DeepExploration* software is used to convert those selected ProE CAD files into desired format and this work has been done by Xcelgo. While formatting in *DeepExploration* the movable part files were separated from original product file to assign the dynamic properties or motions. Importing of .x format CAD files is completed by using Mesh event in C# API.

The imported CAD files are assembled in Experiior's window using drag and drop method. Edit mode has been used after debugging the project and necessary coordinates have been noted down in text documents after placing these CAD files in appropriate locations. In C# command window, these noted coordinate values have been used in trial and error method until the final assembly has been built. Figure 4.2 describes edit mode used in Experiior to place the CAD models by using drag and drop system.

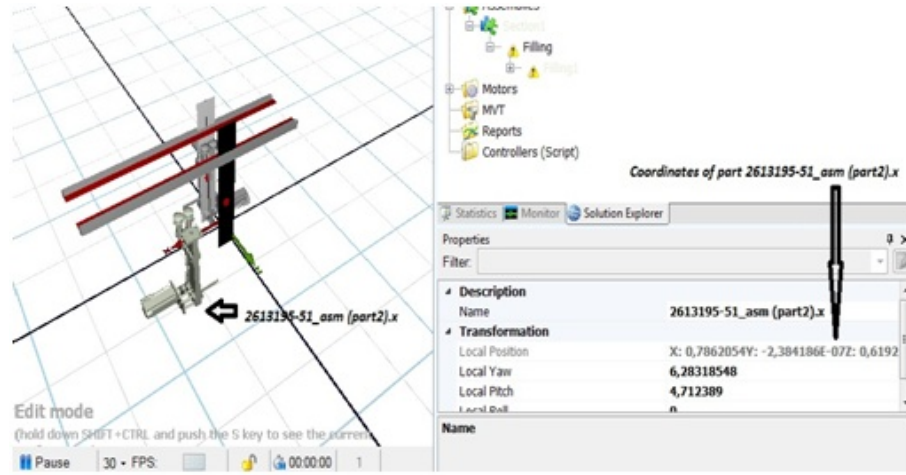


Figure 4.2: Positioning of CAD files by drag and drop method to get the desired coordinates

TetraPak filling module of TR/28 has two different production units, both are asymmetric which consists of the same upper feeder for liquid filling, washing hopper for cleaning filling unit that consists of water, two different servo motor for lifting cartons, two electric servo motors to operate conveyor belts, two conveyor belts, and cartons for filling with liquid. Mechanical lifters are operated by servo motor and used to lift up and down the cartons. Upper part of mechanical lifter is divided into two holding positions, 9 cm apart from each other to carry two cartons at the same time. Conveyor belts are used to hold and move these cartons and operated by electric motor. All the components except cartons are imported from CAD files. The required dimensions for this manufacturing unit are analyzed and used from previous project that was conducted by Chalmers Automation Research Group in Delmia Automation[5].

In order to describe the simulation model for TR/28, it is obvious that description of one unit is enough to describe the others since both are symmetric. The movable parts are mechanical lifter and cartons. In Experiior API, for modelling of a motor in

virtual environment, it is needed to add transport section in the route of moving part. In reality cartons are attached with conveyor belt by hangers. For simplicity in this simulation model, two transport sections have been added at the same side and position of conveyor belt1 where each transport section has been used to carry one kind of load and electric motor1 has been attached for moving these loads in both transport sections. Therefore, cartons have been added as load1 and load2 by 9 cm apart, same distance as between two holding position in mechanical lifter, from each other in transportsection1 and transportsection2. In scripting, *Core.Routes.ActionPoint* command has been used to place imaginary sensor to detect the loads in particular route. Continuous loads have been generated in simulation model by using those action points.

For another moving part mechanical lifter, in one unit which is originally operated by servo axis to get accurate motion and position. In order to work with AB PLC with servo axis control, PLC needs real negative feedback from servo axis to control the motion and position of servo drive. And it is a great challenge for any developing software to model virtual servo axis drive to work with AB PLC control. Due to this complexity in development of servo axis in virtual environment, in this model electric motor without servo axis control was used. The motion has been controlled in scripting by setting the motor speed two times higher when lifter is going upward than going downward and the upward position has also been defined in coding. In actual system the cartons are filled with liquid when lifter is moving down. In this simulation model, lamps are used to indicate the filling of liquid. Figure 4.3 represents lift profile and volume profile used in simulation model.

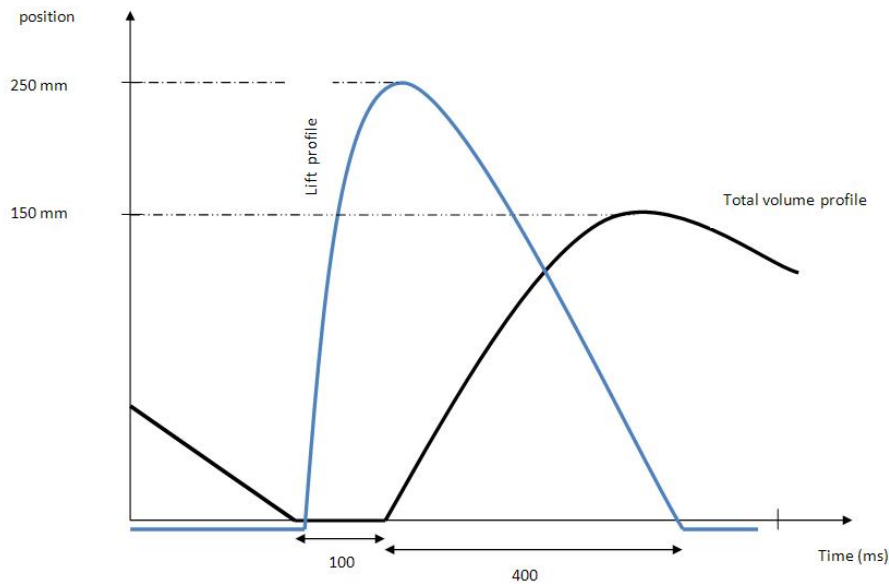


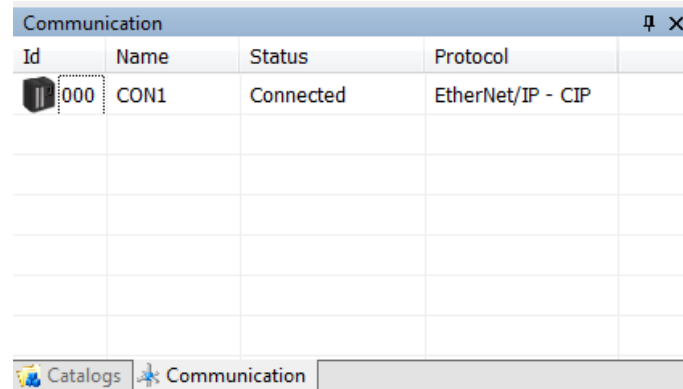
Figure 4.3: Lift and volume profile

To communicate with the external PLC it is necessary to build communication pro-

protocols in the form of I/O in virtual environment. This is usually done in Experior API by using the reference `Experior.Core.Communication.PLC`. In this Tetra Pak model actionpoints that are needed to send signal to the real PLC of the status of the current condition acted as PLC input. Motors along with some activities in different conditions have been added as PLC output to get signal and activate when necessary. In total this model has ten PLC inputs and six outputs established by C# scripting to run this virtual model with AB PLC.

4.2 Establishment of Communication

A communication option in Experior enables a connection with a Rockwell PLC. The PLC that has been used for this thesis work is an AB type L63 LOGIX5563 on a backplane type: 1756-PB72/C. A communication protocol type EtherNet/IP-CIP, see Figure 4.4, has been chosen from Experior software that is compatible with building the real time communication between Experior and AB's ECU. Communication has been developed via an Ethernet cable between the PLC and PC with the virtual model. For this purpose an Ethernet card has been put on the backplane of the PLC, type EN2T/A to connect the Ethernet cable to.



Id	Name	Status	Protocol
000	CON1	Connected	EtherNet/IP - CIP

Figure 4.4: Communication protocol is connected to communicate with Allen Bradley PLC and Experior

In Experior, communication has been set up using Ethernet network connection with definite IP-address and slot position of the ECU of AB PLC and the communication word length that is needed. Here explicit messaging has been automatically developed by using of CIP port number 44818. In this communication mode PLC is used as client and Experior as server. This setup can be seen in Figure 4.5

To control the filling machine model in Experior the Rockwell software, RSLogix 5000 has been used. Ladder diagrams have been established for the sequence that controls the model. Each I/O-tag has a specific name and description that is corresponding to the tagnames set in the C# coding for I/Os in the virtual model. I/O tags are exported from RSLogix 5000 and formed "Alias I/O list.csv". This file can be seen by using Excel sheet where csv export-import file is created with proper descriptions of aliases and tags.

Properties	
Filter:	
Identification	
Name	CON1
Id	000
Communication	
CPU Slot Number	1
Size	DINT
Port	44818
IP Address	192.168.1.2
Mode	Client
Auto Connect	False

Figure 4.5: Allocating IP address, slot number and types of communication

I/O's in LD diagram from RSLogix 5000 are called "Alias" which point to a specific bit on a specific byte on the word that is transferred to the virtual model. Aliases make it easier to recognize their location compared using the full length name of the specific bits. For example tags that are used in Experior are exported from RSLogix 5000 where it is changed to correspond to the correct bit. These tags are used in the Ladder program in the rungs needed to control the virtual filling machine. This program is downloaded to the PLC and the program execution is viewed in the RSLogix 5000 window, as shown in Figure 4.6. The tags are exported from RSLogix 5000 to Experior and are connected to the correct action on the virtual model, shown in Figure 4.7.

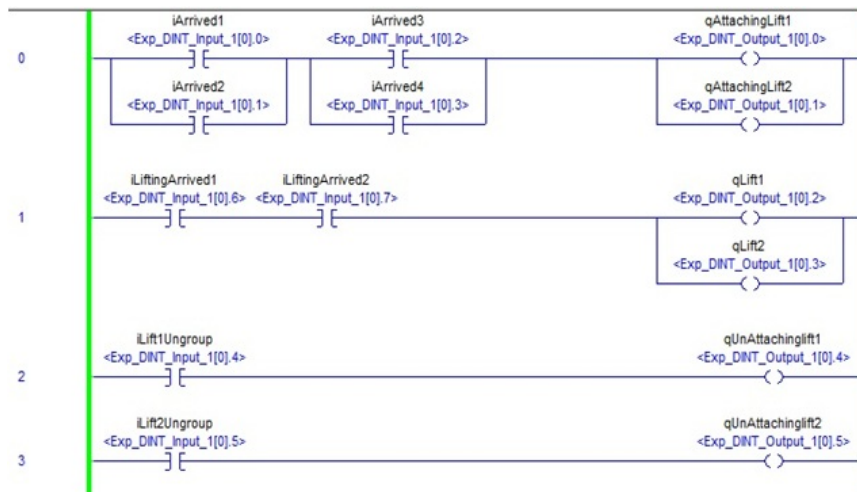


Figure 4.6: Ladder diagram written in RSLogix 5000 to run virtual model

Though main objective of this thesis is to make this model possible to communicate with Allen Bradley PLC, a test communication has been accomplished to test and verify

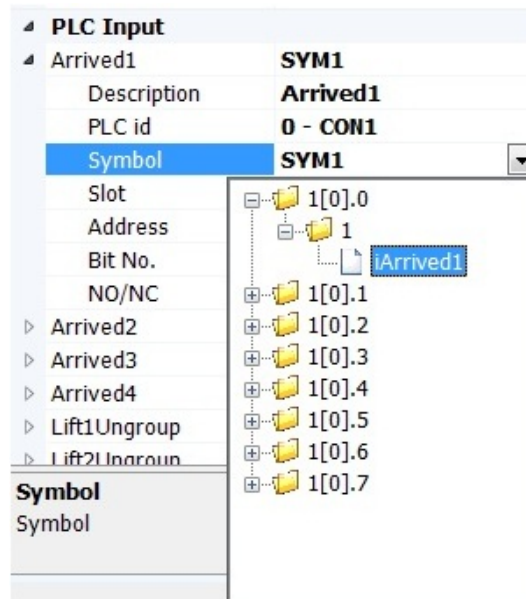


Figure 4.7: Adjusting PLC tags that are imported from .CSV file

the control logics in pure virtual commissioning, without adding any hardware in the loop system. TwinCAT software has been used for this purpose. Communication with TwinCAT is established with a plugin in Experior that makes it possible to create a program to control the model. The communication type “BeckhoffADS” is chosen in the “Communication” tab in Experior. I/O’s that are used in TwinCAT PLC control are available through the command “Read Symbols”. I/O’s that are created with C# coding are then available to connect to a PLC input and output on the dropdown menu for the specific tag. The virtual filling machine model is then controlled by a virtual PLC and the program execution can be viewed in the TwinCAT window.

5 Results & Discussions

A virtual 3D model, shown below in Figure 5.1, is developed by rendering the CAD files in Experior window with visual effect. This model comprises with all the necessary dynamics to depict the real TR/28 machine's filling module except the flow of liquid to fill the cartons. But the model itself is able to show the filling time by switching the lamps ON/OFF during filling process. And since nozzles are not connected to each other, these nozzles are controlled individually. Filling is occurred only when it detects there is a carton in the mechanical lifter. This model is also capable of showing that kind of behaviors by letting the lamp associated to each nozzle ON/OFF, if there is carton or not. Motions from virtual motor are controlled by using commands from API.

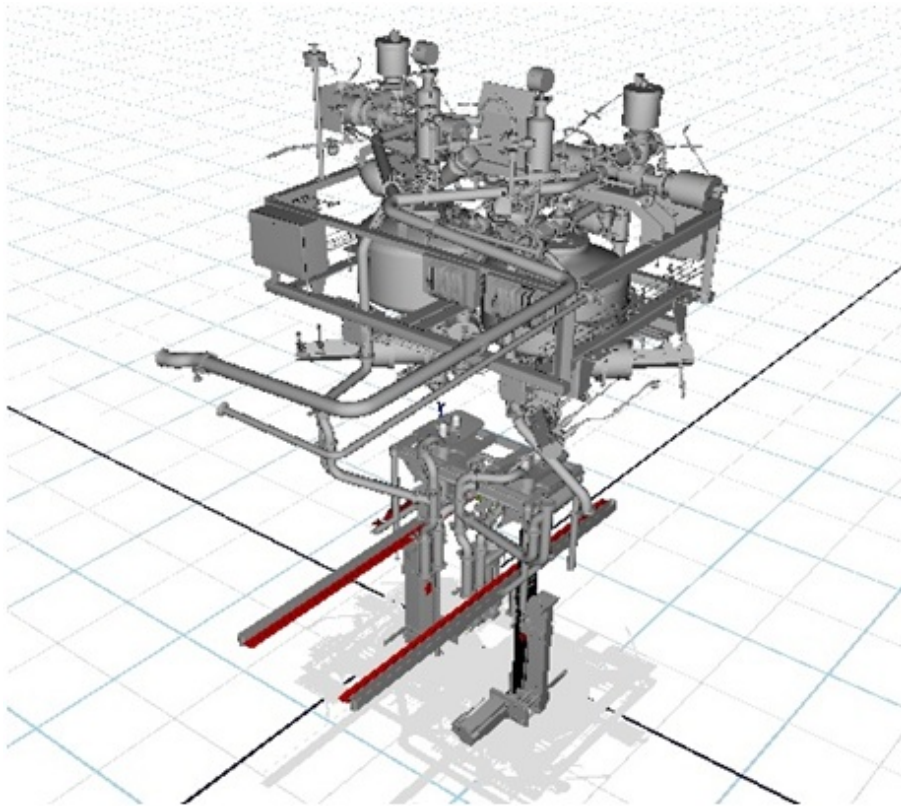


Figure 5.1: Virtual model of TR/28 machine

Control points have been assigned with specific behaviors in scripting using appropriate function developed by Experior API to control this filling unit. Control points are visible in form of PLC input and output in the Experior model's properties.

Also one issue is that, to import large CAD-files into Experior the processor speed needs to be high as well as the graphics card to work with the dynamics in the model. The using of Nvidia physics is useful, but it is still required to use a high speed CPU when it comes to work with large CAD drawings.

Some assumptions were made to compare the concurrent research of Tetra Pak. Due to complexity and non disclosure issues, the real PLC codes are not used in this project, instead, a simple PLC program to control the Lifter and Filling piston is written with RSLogix 5000. This thesis has another limitation of controlling the servo axis using virtual model with Experior and AB PLC. But due to complexity in servo motor when it comes to virtual modeling and lack of robust solution for communication between Experior and AB PLC, normal electric motor has been used and speed ratio of this motor has been defined in C# scripting to give reflection of original servo drive.

Real time communication has been achieved by simultaneously sending and receiving signals between AB PLC and virtual model of TR/28 in HIL simulation. But sometimes there is little delay between exchanging signals which causes the simulation stop after long run. This simulation model has been tested in different speed in Experior by keeping the same actual speed ratio, to test how long it can run without interrupting. Every sensor in the virtual model sends output signal to real PLC for a certain amount of time. At the beginning it was set to 0.2 seconds. The solution is improved by enlarging the duration time of sensors to 0.6 seconds for giving signals to withstand with the communication delay between these two. Figure 5.2 demonstrates the communication between real PLC and Experior.

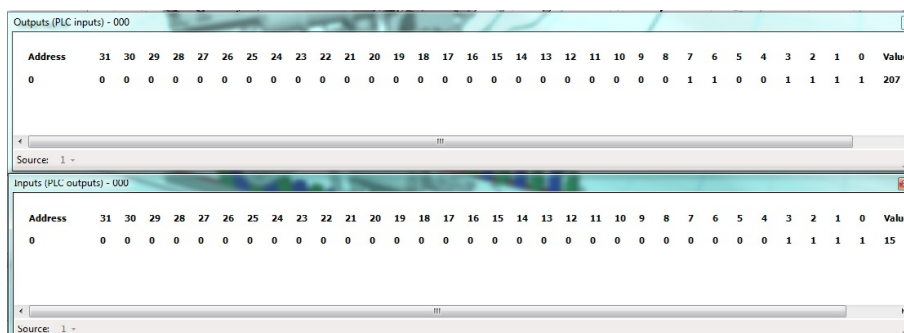


Figure 5.2: Synchronous communication between Xcelgo and Allen Bradley PLC

Outputs from Experior are acted as inputs for PLC and vice versa. From Figure 5.2, it is obvious that a real time communication is achieved, when addresses 0,1,2,3,6,7 are activated it generates bits for PLC inputs. According to LD diagram in RSLogix 5000, these are iArrived1, iArrived2, iArrived3, iArrived4, iLiftingArrived1, iLiftingArrived2 for PLC input. PLC outputs that are connected with these inputs are qAttachingLift1, qAttachingLift2, qLift1 and qLift2, are also activated and send these signals to Experior to activate this tasks in the virtual model.

The communication between the PLC and the virtual model running in Experior is difficult to achieve due to the way that bits are named in RSLogix 5000 compared to how it is done in Experior. These problems starts at how programmers at Xcelgo assume that Rockwell is naming their bits are not actually the way they are doing it. It is difficult to understand the way communication is managed by Rockwell and hence, it

is a non-disclosure pattern, makes it complex to follow. However, when a connection is established it is stable and fast and is working for the simple program that was used to control the model.

The communication with TwinCAT works and is easy to establish. Compared with Rockwell, Beckhoff has made it easier for others to communicate with their PLC software. Hence, the interaction between the software is working as it should.

6 Conclusion & Future Work

Now-a-days verification and validation of PLC code of highly automated manufacturing industries are limited in physical level of testing with real hard drive. Demand for virtualization in this purpose is rapidly increasing to check the fidelity of the control programs in specified tasks. According to current requirement, this thesis is intended to virtualize the filling module of TR/28 using Xcelgo Experior software and to verify the control programs that are written in RSLogix 5000 with this virtual model. Real time communication has been developed between Experior and AB PLC to test and verify the control programs with EtherNet/IP-CIP connection.

The emulation software Xcelgo Experior made it possible to solve the task in an organized way. After taking cursory look on Results, it shows that this thesis gives comparatively good output in a transparent and intuitive manner to achieve the final goal. But in meticulous points of view lots of improvements are still necessary for deploying this verified control programs in real world.

In future work with Xcelgo Experior and the communication with AB PLC and the software RSLogix 5000, there are some factors to consider. When working with new software it is important to have access to good information on how the software works and how to use it in a way that satisfies your needs. The documentation on Xcelgo's own "Wiki" is good and with assistance from a senior developer at Xcelgo it is possible to achieve a HIL simulation. To complete this thesis Xcelgo has been very helpful in giving an introduction to Experior and answering questions on difficulties that has come up. In the future a more extensive documentation available for users would make development of new models more independent from expert help from Xcelgo.

The plugins that are created by Xcelgo to communicate with RSLogix 5000 and AB PLC is not a stable connection because of the complexity. It would be a big benefit if in the future it would be possible to improve the stability and make it easier to establish the connection.

With further development of the model to include the whole TR/28 and not just the filling part would be a natural way to improve the result of the emulation. The servo drive control would also be important to include in order to completely have use of the emulation model and to be able to test new PLC code running on the same type of PLC without using real physical equipment.

There could be another possibility of working with hybrid commissioning in very first time to check the errors between real plant and virtual plant with real control system. Hence virtual model can be modified to minimize the differences between these two plants or it could be possible in next run of virtual commissioning to realize these differences and corrective action could be possible when deploying these control codes in real plant.

References

- [1] I. Automation, Benefits of industrial automation (March 2013).
URL www.industrial.automation-au.com/pages/benefits.htm
- [2] M. Rausch, B. Krogh, Formal Verification of PLC Programs, in: In Proc. American Control Conference, 1998, pp. 234–238.
- [3] F. Auinger, M. Vorderwinkler, G. Buchtela, Interface driven domain-independent modeling architecture for 'soft-commissioning' and 'reality in the loop' (1999).
- [4] S. Bangsow, U. Günther, Creating a model for virtual commissioning of a line head control using discrete event simulation, in: Use Cases of Discrete Event Simulation, Springer, 2012, pp. 117–130.
- [5] A. Hollander, S. Sappei, Virtual Preparation of Tetra Pak Filling Machine (2011).
- [6] Xcelgo, Xcelgo Experior (March 2013).
URL <http://www.xcelgo.com>
- [7] I. McGregor, Equipment interface: the relationship between simulation and emulation, in: Proceedings of the 34th conference on Winter simulation: exploring new frontiers, WSC '02, Winter Simulation Conference, 2002, pp. 1683–1688.
URL <http://dl.acm.org/citation.cfm?id=1030453.1030699>
- [8] P. Hoffmann, R. Schumann, T. M. Maksoud, G. C. Premier, Virtual commissioning of manufacturing systems a review and new approaches for simplification, in: Proceedings 24th European Conference on Modelling and Simulation, 2010.
- [9] R. Drath, P. Weber, N. Mauser, An evolutionary approach for the industrial introduction of virtual commissioning, in: Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, IEEE, 2008, pp. 5–8.
- [10] L. Chwif, M. R. P. Barretto, R. J. Paul, On simulation model complexity, in: Simulation Conference, 2000. Proceedings. Winter, Vol. 1, IEEE, 2000, pp. 449–455.
- [11] S. Makris, G. Michalos, G. Chryssolouris, Virtual commissioning of an assembly cell with cooperating robots, *Advances in Decision Sciences*.
- [12] H. Schludermann, T. Kirchmair, M. Vorderwinkler, Soft-commissioning: hardware-in-the-loop-based verification of controller software, in: Proceedings of the 32nd conference on Winter simulation, Society for Computer Simulation International, 2000, pp. 893–899.
- [13] W. Grega, Hardware-in-the-loop simulation and its application in control education, in: 29th ASEE/IEEE frontiers in education conference, session 12b6-7, 1999.

- [14] S. Dominka, F. Schiller, S. Kain, Hybrid commissioning-from hardware-in-the-loop simulation to real production plants, in: The 18th IASTED International Conference on Modelling and Simulation, ACTA Press, 2007, pp. 544–549.
- [15] PAControl, Programmable logic controller - plc (March 2013).
URL www.pacontrol.com/PLC.html
- [16] PLCopen, IEC 61131-3 (March 2013).
URL http://www.plcopen.org/pages/tc1_standards/iec_61131_3
- [17] O. D. V. Association, Ethernet/IP (March 2013).
URL <http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/tabid/67/lng/en-US/Default.aspx>
- [18] O. D. V. Association, The CIP advantage technology overview series (March 2013).
URL http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00138R3_CIP_Adv_Tech_Series_EtherNetIP.pdf
- [19] O. D. V. Association, Common Industrial Protocol (March 2013).
URL http://www.odva.org/Portals/0/library/publications_numbered/PUB00122R0_CIP_Brochure_ENGLISH.pdf
- [20] R. Automation, Rockwell Automation (March 2013).
URL <http://www.rockwellautomation.com>