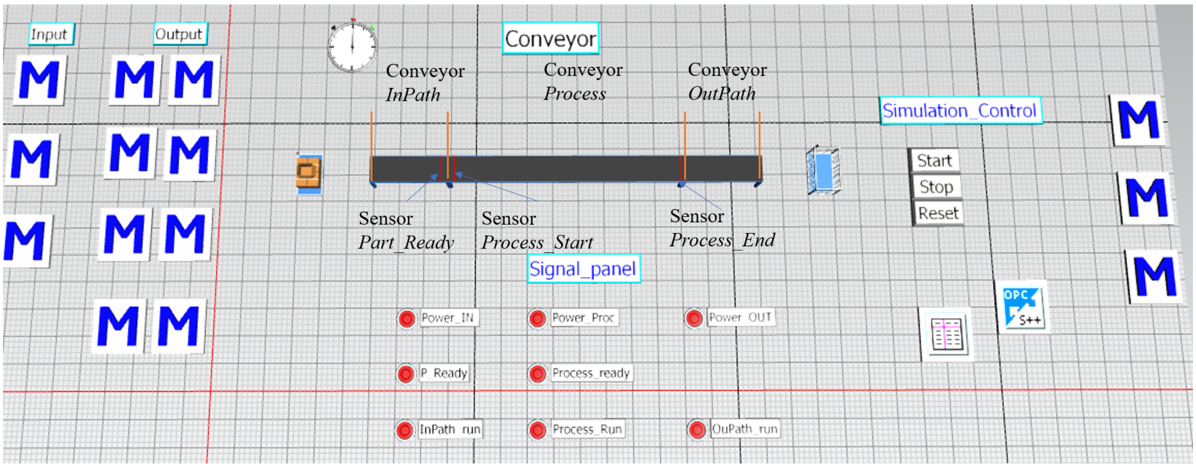
CHALMERS
UNIVERSITY OF TECHNOLOGY



PLC Integrated Discrete Event Simulation for Production Systems

PLC signal integrated simulation establishment & Comparison towards discrete event simulation

Master's thesis in Production Engineering

Viktor Engström
Zhizhong Liao

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017

PLC Integrated Discrete Event Simulation for Production Systems

PLC signal integrated simulation establishment & Comparison towards discrete event simulation

Viktor Engström
Zhizhong Liao

Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
Gothenburg, Sweden 2017

PLC Integrated Discrete Event Simulation for Production Systems
PLC signal integrated simulation establishment & Comparison towards discrete event simulation
Viktor Engström
Zhizhong Liao

Supervisor:
Ilker Erdem, Department of Industrial and Materials Science
Tobias Börjesson, Swedish Match

Examiner:
Anders Skoogh, Department of Industrial and Materials Science
Tomas Jönsson & Mattias Wermé , Swedish Match

Master's Thesis 2017
Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Picture of model from RQ1 created in this Master thesis

Typeset in LaTeX
Gothenburg, Sweden 2017

PLC Integrated Discrete Event Simulation for Production Systems
PLC signal integrated simulation establishment & Comparison towards discrete event simulation
Viktor Engström
Zhizhong Liao
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Swedish Match decide to explore the possibility of integrating PLC logic into their DES model for their production line. This thesis investigates the possibilities for building communications between DES simulation model and PLC logic within the environment of Siemens Plant Simulation and PLCSIM. The communication frame and procedures regarding configuration of platform establishment is under investigation. Another aim is to make a comparison between two different simulation models regarding performance and complexity. An investigation is made via a case study, where a PLC integrated simulation model is developed based on the frame proposed in the aim, checking key performance indicators of two simulation models against data acquired from the real production, in order to reveal accuracy of the two simulation models. Besides the complexity of the two simulation models w investigated in order to compare the workload in the model development. Since the complexity of simulation heavily influences the cost and time in the phase of simulation model development. Therefore, a conclusion whether the PLC integrated simulation worth efforts can be drawn or not.

Keywords: Production simulation, PLC signal integrated simulation

Acknowledgements

We want to thank people from both Chalmers Technical University and Swedish Match who have contributed with help in this thesis. Swedish Match provided the topic of the thesis and the opportunity to conduct the project at their factory. Especially we want to thank following people:

Tobias Börjesson - The supervisor at Swedish Match, provided us with help of DES related problems and was our contact person during the thesis at the company.

Tomas Jönsson & Mattias Wermé - They are the examiners at Swedish Match and they made it possible for us to conduct this thesis at the company.

Per-Vilhelm Andersson - One of the employees at Swedish Match within the department of automation, helped us with accessing software and knowledge of the production line.

Joakim Kronvall - One of the employees at Swedish Match within the department of automation, helped us with Siemens PLC software regarding this thesis.

Ilker Erdem - PhD Student, from div. Production Systems, IMS at Chalmers Technical University. He is the supervisor in this thesis, who made it possible for us to move forward the project with his knowledge and recommendations within this topic.

Anders Skoogh - Associate Professor and Director of Master's Programme in Production Engineering at Chalmers Technical University. He is the examiner of this thesis.

Viktor Engström & Zhizhong Liao, Gothenburg, August 2017

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

DES (Discrete Event Simulation) which is one area of general simulation, is able to simulate the flow of material through a production, network or logistics system (Hloska & Kubin 2014). DES enables to carry out experiment to test different scenarios for optimization.
Today's manufacturing systems contain conveyor system, robots, safety system and other equipment (Virtual Commissioning 2017). Control software takes up over 50% of the whole function of highly automated production equipment (Glas 1993). PLC (Programmable Logic Controller) is widely used in the production. It would be interesting to integrate PLC signal into the DES model for the assumption of a more accurate result. The project introduces a practice to integrate PLC signal into the simulation model within the environment of Siemens software PLCSIM and Plant Simulation. A case study, in which modification is implemented on a simulation model, is carried out to investigate the difference in accuracy between PLC signal integrated simulation model and the discrete event simulation model. Also, an investigation upon complexity is carried out to compare, the different workload in model development phase of two simulation models.

1.2 Research questions

The following research questions are stated and addressed throughout the course of this thesis:
RQ1: ***How can engineers use PLC signal as input to a simulation model?***
RQ1 aims to provide the frame and steps in integrating PLC signal into the simulation model, which introduces the channel configuration through DES tool, OPC software and PLC simulator. It needs to identify necessary objects and codes during the development. The validation of the integrated model is necessary.
RQ2: ***What are differences between such a model compare with an ordinary simulation model?***
The difference of the two simulation models is compared in two perspectives, which are accuracy and complexity. The result of simulation model is the most interesting part in the simulation project. Result comparison between the two simulation models against data acquired from real production, reveals the accuracy of the simulation model. Complexity measures reveal the difference in model construction, coding and

computational time, to reflect the workload difference in the two simulation models.

1.3 Purpose

The DES model is developed to predict production, helping the company to make right investment. With a developed simulation model and verified PLC code, a question that "is it possible to integrate the PLC code into the simulation model" raised up. Part of the thesis was to figure out a method to integrate PLC code into a discrete event simulation model. Another part of the thesis tries to compare the difference of PLC signal integrated simulation model with original simulation from the perspective of accuracy and complexity. In practice, the accuracy of simulation model influences the performance of simulation model while complexity influences the workload of simulation project. At the end, a conclusion is generated based on the result of investigation.

1.4 Aim

Based on the purpose of the thesis, there are two aims that shall be achieved at the end of the thesis work. The first aim is to enable integration of PLC signal into DES model. The method shall include framework, necessary steps and key procedures of implementation, which are of importance to integrate PLC signal successfully into DES model based on provided tool and software. The second purpose is achieved by carrying out a case study. The case is a developed DES model and a set of verified PLC code of a conveyor system. The PLC code is currently implemented on the conveyor system. By using the result of first aim, a PLC signal integrated simulation model is developed. The evaluation starts from two aspects which are accuracy of simulation result and complexity of simulation model development. The evaluation matrix follows previous research work. The result shall tell whether the PLC integrated simulation is able to gain better accuracy and worth doing.

1.5 Scope & Delimitations

The platform is built on within the environment of the company. The software are TIA Portal V13, PLCSIM, Plant simulation V13 and Kepware. The PLC signal should be sent and received from a virtual PLC controller. Physical PLC controller is not used in the thesis.
The case study is limited to the manufacturing process at Swedish Match Gothenburg factory. The selected case is the buffer conveyor system. Other production systems are not considered in the thesis. For the selected case, following files are provided, which are the simulation model and PLC code. Therefore, there will be neither base simulation modelling nor PLC coding. Necessary modifications are allowed on the simulation model and PLC code, in order to enable the signals travelling between DES software and PLC simulator. Otherwise there will be no modification of the structure of the simulation models nor PLC code in this thesis.

The evaluation is carried out through a case study. Investigation of improvement possibility is not in the scope of the thesis.
The complexity evaluation matrix used for RQ2 does not evaluate the workload of PLC programming, OPC and IT settings. Cost is not the parameter in consideration.

1.6 Outline

The outline of the thesis is shown below:

Name	Command
Ch 2: Scientific Framework	Necessary scientific part to cover the thesis
Ch 3: Methodology	The procedure to solve RQ1 & RQ2
Ch 4: Result	Results of RQ1 and RQ2
Ch 5: Discussion	Result analysis and discussion
Ch 6: Conclusion	Conclusion of the thesis

2

Scientific Framework

2.1 Discrete Event Simulation

2.1.1 Fundamentals of Discrete Event Simulation

Simulation is an engineering tool that is used to create models of a real system. The purpose is to inherited the real system behavior to a virtual environment, to display it as a simulation model (Shannon 1998). The model is able to conduct experiments in order for the researcher to understanding the behavior of the real system. By using the simulation model, the researcher can detect current errors in the production or future improvement possibilities by analyzing its behavior. One of the greatest advantages of using simulation models is the time saving of the real commissioning in the project. If problems are detected in the simulation, it can be adjusted before the implementation in the real system.
Discrete event simulation is a simulation procedure that consists of a modelling concepts of a system's different features that is put together into a coherent set of procedures. The features are linked with a mathematical relationship that represents the real-life function for example of a machine. The computer converts the mathematical relationships into data. The data is estimated as values of the system performance by the procedures in the software (Fishman 2001).

2.1.2 Advantages and disadvantages of DES

Some of the advantages, mentioned above an early detection of errors and investigation of possible improvements in the current production would be beneficial. Even if the simulation model contains a simplified representation of the real system, it still provides valued information. The simulation model brings more accurate data by feeding reliability parameters (Kampa et al. 2017).
DES provides the opportunities for researchers to organize their theoretical beliefs and empirical observation of the system and to implement the logical implication. It is performed without jeopardizing the real system. DES leads to a improved understanding of the system, and brings into perspective of the need for detail asl well as relevance of the system. It also improves the understanding of how the system operates rather than how individuals think the system operates. Specific hypotheses about how or why certain phenomena occurs can be tested. With DES the analysis of the model can be accomplished faster in a shorter amount of time in comparison to a real system. It's easier to manipulate the DES model than real system and test new implementation. It is general less costly than performing investigation on the

real system directly (Fishman 2001; Banks et al. 2005).
Skoogh and Johansson (2008) states that DES projects relies strongly on the quality of the input data. Because of that, the input data management is very important. Data collection is time consuming. Another disadvantage regarding DES is the amount of time it needs to perform a simulation study. Especially in early conceptual phases, in order to reduce project time, a quick response on the analysis are preferable (Skoogh & Johansson, 2008).
Discrete event simulation provides possibility to use analytical methods, however these methods on the other hand can not provide any solution to a problem (Fishman, 2001). When a simulation is made, engineers/ researchers should perform additional work in order to find a solution to the problem they discover.
The individual skill of the model builder is a factor influencing simulation model reliability (Banks et al. 2005). Results from a simulation can be difficult to interpret, because of the outputs are in fact random variables (which is based on random input). Therefore, it can be hard to separate whether an observation is a result of the systems interrelations or of the simulations randomness. Other disadvantages from Banks et al. (2005), are time consumption and expense of the analysis tool. If they are insufficient in consideration when the project resource is low (Banks et al. 2005). The result of that simulation can be invalid.

2.1.3 Steps in a Simulation Study

According to Banks et al. (2005) model building requires special training and it is learned over time and experience. These steps are needed to perform a simulation study in order to create a model are following (Banks et al. 2005):
1. Preparation
a. Problem formulation: Clearly defined problem in order to reach a common understanding.
b. Setting of objectives and overall project plan: Create measurable project goals, project time frame, delimitation and level of detail.
c. Model conceptualization: Create a simple model with logical relation between model entities in terms of creating a basis of discussion for common understanding.
d. Data collection: Suitable parameters of the project are detected and data of these parameter is collected.
2. Model building
a. Model translation: Creation of the simulation model; coding the conceptual model; input data according to the preparation.
b. Verified: Verifying each element of the model for the sake of correct behavior.
c. Validated: The general behavior of the model is well represented of the real system in order to get measurements relevant to the study.
3. Analysis
a. Experimental design: Perform different types of analytically methods in order to get relevant result.
b. Production runs and analysis: Perform a certain number of runs in order to get the right amount of result to get it valid, find the problems of the current production and find improvement possibilities.

4. Result & Implementation
a. Documentation and reporting: Save the data of the simulation runs.
b. Implementation

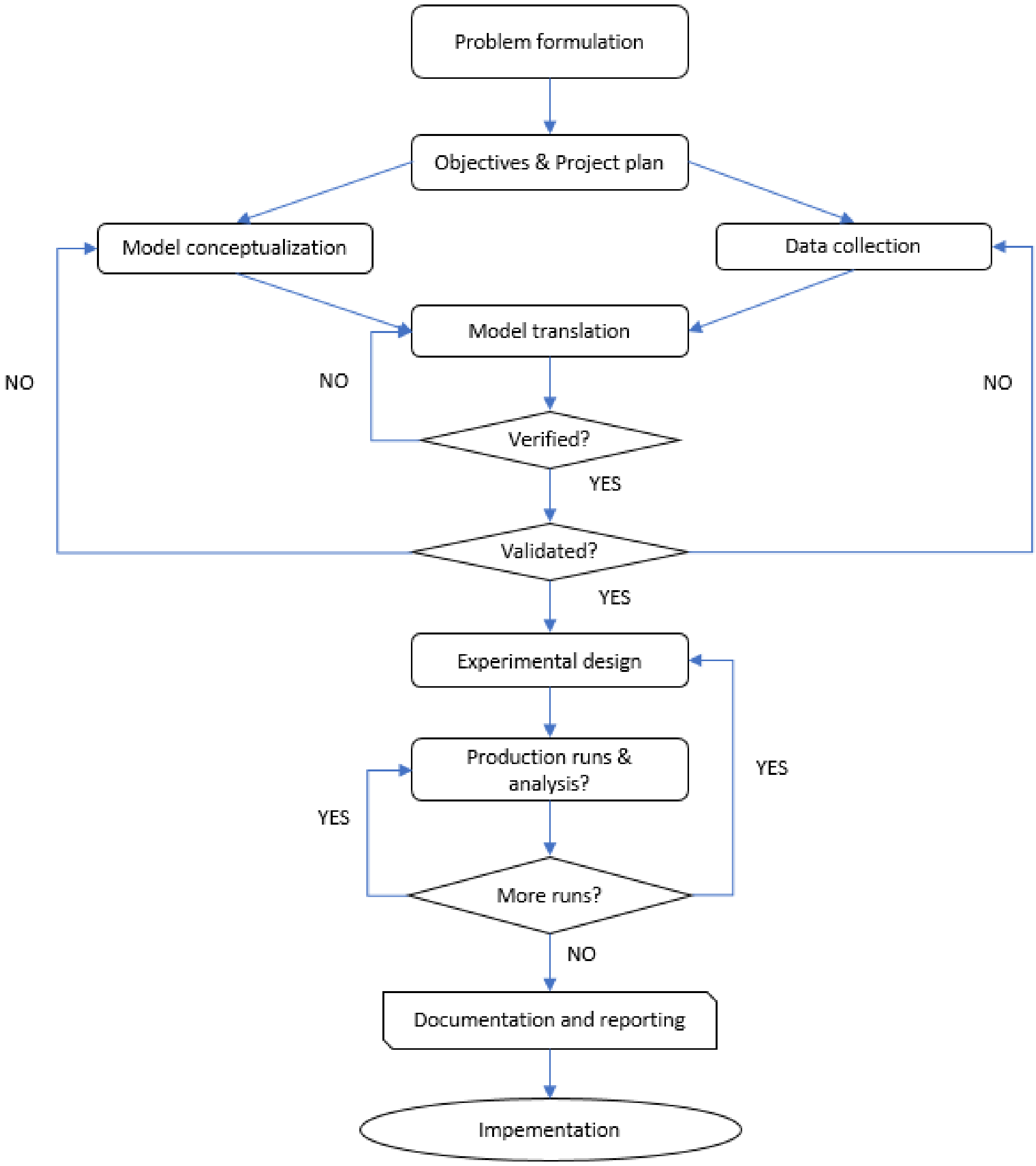


Figure 2.1: Simulation model development flow chart (Adapted from Banks et al. 2005)

2.1.4 Data collection

Data management is the entire process of preparing quality assured, simulation adjusted, representations of the relevant input data parameters for a simulation model. It contains the process from identifying the relevant input parameters, collecting all relevant data that is required to use the parameter as suitable simulation inputs, converting the collected raw data to a quality assured representation and document-

ing data for future research and reusage (Skoogh & Johansson 2008).
The data is classified into three categories by Robinson and Bhatia (1997) based on the availability and collectability. Based on the data category, the method of getting proper data from data sources are different. For data category A, data is possibly extracted from suitable data sources. For data category B, data is recorded by time study, observation and documentation. For data category C, it is necessary to make good estimations.

Table 2.1: Data classification (Adapted from Robinson & Bhatia 1997)

Category A	Available
Category B	Not Available but collectable
Category C	Not Available and not collectable

Table 2.2: Method for getting data in different category (Adapted from Input Data Management 2015)

Category A	• Production engineers' own documented measurements • Automated Collection Systems (PLC logging) • Computerized manual collection systems • ERP, MES, MRP or maintenance systems • Process design documents • Documents from previous gathering efforts
Category B	• Time studies using stop-watch • Time-studies using video analysis • MTM-studies using video analysis • Frequency studies • Follow-ups during several days • Ledgers for operators to fill in • Interviews
Category C	• Interviews • Focus (expert) groups • Historical data from similar processes • Machine vendor information • Process design documents (MTM calculations etc.) • Use more simple distributions (e.g. triangular) • Combine sources!

2.1.5 Validation & Verification

Validation and verification are important aspects of a created model. Validation is confirming that the model is an accurate representation of the real system. This is achieved by performing calibration of the model and compare it to the actual system and its behaviour. By using the discrepancies between the model and actual system, the insight will gain and use it to improve the model even further. Verification phase

in the model building is ensuring that the model design has been transformed into a computer model with a sufficient accuracy (Robinson, 1997). The model is compared to the conceptual model in order to check whether the model is implemented correctly, as well as the input parameters and logical structure of the model are represented correctly (Banks et al. 2005).
The validation models could be used as the accuracy measurement in order to detect the model's accuracy. Typical measurement of performance for simulation models is primary throughput, system cycle or response time, and work in process. Secondary or explanatory measures can be such as resource utilization, size of local buffers, and throughput for subsystems or particular part types (Carson, 2002). In order to provide an accurate model (a model towards the real system) black-box validation can be used in order to check the overall correctness of the simulation model (Sarget 2013). Black-box validation refers to the determination of the overall model, and its output behaviour, reflects the real world with sufficient accuracy (Robinson 1997; Sargent 2005). The output can be represented as a mean value together with standard deviation in comparison to the real systems values (Banks et al. 2005; Majid et al. 2010).
Despite the fact Banks and Majid provide what method they used, the information how they used it could be lacking from thesis to thesis. In the thesis and articles mention above, throughput was used as accuracy measure to see how far from the reality the model was. Reason of the lack of quantitative measure from the validation phase, could be be explained through the use of experts' opinion in order to get the model accepted.
According to Banks et al. (2005), the accuracy of the simulation (in validation phase) is important in order to determine if the model is good enough. Brooks & Tobias (1996) states that confidence in the model structure, either on a theoretical basis or on the basis of successful previous experience, is important. Carson (2002) also states that different outputs is not enough but also the behaviour of the processes which represent the real system processes provides more knowledge about the accuracy of the model. Experts of the real system, validate the DES model by their opinion and knowledge in order to ensure the model with enough accuracy. Visual accuracy or animation refers to the model's operational behavior, which is displayed graphically as the model responses through time. The movement of parts in the factory during the simulation runs is shown visually through the simulation software. This can be observed by the authors and as well experts of the real system. Turing test (Sargent 2013) is another validation technique that can be executed by experts. By providing test result from simulation runs to the expert without providing information on which result represents the models or real system. If the experts cannot distinguish the model result towards the real system it can be stated as accurate.
Another essential validation part is to see the behaviour in the simulation model against the real system process (Banks et al. 2005; Brooks & Tobias 1996). White-box validation refers to the determination of that subsets of the computer model correspond to the real world. Various aspects such as timing, control of flows, control of elements and control logic, are checked (Balci et al. 1996). Animation and operational graphics (Sargent 2013) are two validation techniques. Values of various performance measurements are shown graphically as the model runs though time

in operational graphics. It ensures the performance measures and the model is behaving correctly toward the reality. Another technique by Sargent (2013) is trace, where the behaviour of a type of entry in a model is traced though the model in order to determine if the model's logic is correct and if the accuracy is obtained. By ensuring the simulation model's behaviour is correct on process level, the total performance measure can be evaluated in a more correct way.

2.1.6 Relationship between simulation and emulation

Emulation can be defined as imitation of a system by another system (Hloska & Kubin 2014). McGregor (2002) raised three similarities of simulation and emulation.

1. Tend to build and represent model in three dimensions.
2. Result of should be accurate enough to be useful.
3. Models behave as similar as real systems or machines.

It is easier to understand and interpret result with 3D models than 2D models. Simulation and emulation builds models to help people understand logic's behind the theory, make decision and modify current models.
Besides similarities, there are differences as well. McGregor (2002) identifies it from aim, time, and requirement.
Engineers use simulation model to test different scenarios, such as design of experiment, and compare the result based on metrics. Emulation is more used to verify performance of a system, such as a PLC control system. The simulation model runs scenarios in quite short time while the emulation model runs in real time. For simulation, repeatability of the result is more important. Emulation focuses more on robustness. The material flow is monitored through statistic results, which is required as "results remain statistically meaningful" (McGregor 2002). In the emulation model, the control logic can be validated by ensuring that all jobs are routed to their intended destinations (Schiess 2001). The uncertainty in emulation is communication networks since it is non deterministic. Thus, a robust emulation model is important to ensure control system can run under real conditions.

2.1.7 Simulation model complexity

In a simulation project, the simulation model shall accomplish the necessary aspects within investigation. The most common aim is to predict the behavior of a real system. The best simulation model shall provide sufficient result, which fulfils the objectives of the study, but also consumes reasonable resources. A too complex model consumes considerable resources which may exceed the cost and time budget. The result of a too simple model is useless and misleading.
Simulation model is the simplification of the real system but should equip enough level of detail in order to draw a conclusion. It is found that simulation model complexity has critical effects on the development phase of a simulation project and it is widely recognized in the simulation community that a simple model is preferable to a complex model (Chwif et al. 2000). "Model Simple-Think Complicated" is

one of "Five Principles of Simulation Modelling" (Pidd 1996). "Simplification is the essence of simulation" (Salt 1993) and "complicated models have no divine right of acceptance" (Pidd 1996) stressed this points of view. Even though a simple model is preferred, complex and large models grow at a significant rate (Arthur et al. 1999). It is recognized that the model complexity has important effect on model performance (Brooks & Tobias 1996). The relationship is summarized that the complex model is expected to have better validity, accurate result as well as more detail than a simple model. However, a complex model has disadvantages, such as more resource consumption and potential errors, harder to understand and less portable. The evaluation of simulation modelling shall include the benefit together with cost, which enables to present overall conclusion of a simulation project. Brooks & Tobias (1996) listed 11 elements in evaluating performance of simulation projects:

1. The extent to which the model output describes the behavior of interest.
2. The accuracy of the model's results.
3. The ease with which the model and its results can be understood.
4. The portability of the model and the ease with which it can be combined with other models.
5. The probability of the model containing errors.
6. The accuracy with which the model fits the known historical data.
7. The strength of the theoretical basis of the model including the quality of input data.
8. The time and cost to build the model.
9. The time and cost to run the model.
10. The time and cost to analyze the results of the model.
11. The hardware requirements of running the model.

It seems intuitive to understand “complexity”, but there is no general definition of complexity when applied to a model (Brook & Tobias 1996). Different definitions are used by different points of view, but none of them cover all the aspects of complexity. Based on the work flow of simulation model construction, the amount of representation of real production is heavily depended on the aim of the simulation project. The aim determines the level of detail and scope of a simulation model. Scope and level of details are two aspects in the concept of complexity (Chwif et al. 2000). Scope refers to the scale of the model. Scope of modelling machine process and a plant logistic is different. Level of details, or granularity in another word, refers to the volume of information provided by the conceptual model.

Since complexity heavily influences the workload of simulation model establishment, it is important to carry out complexity estimation of the simulation model in the planning phase. Complexity measures try to objectively quantify the complexity of a simulation model (Chwif et al. 2000). In the study by Popovicsa & Monostoria (2016), the authors state that most previous approaches for measuring complexity of manufacturing model lie in using axiomatic design theory, information theory, nonlinear dynamics, or the combination of them. However, none of these measures cover all the perspectives of complexity and the measurement shall be done in a given technique which means the result of measurement is constrained by tools used

for simulation development (Chwif et al. 2000).
Since DES models are representations of real systems, containing the similar components and logic, of the real system are applicable in measuring the complexity of DES model (Popovicsa & Monostori 2016). Complexity of real production systems can be defined by physical and functional domains. Complexity in physical refers to time independent complexity and time dependent complexity. Time independent complexity includes physical configurations and interconnections of components. Time dependent complexity refers to the uncertainty of the system's behavior (Efthymiou et al. 2012). The function complicity refers to uncertainty in achieving the functional requirements (Popovicsa & Monostori 2016).
Popovicsa & Monostori (2016) describe an approach to determine the complexity of DES models, which indicates the effort of model development. The measures of the approach inherit the classification of manufacturing complexity in the physical domain. The approach measures DES model complexity from two aspects, which are structural complexity measure and software complexity measure. In the software complexity, it includes algorithmic complexity and computational complexity. The case study by the research is done in Tecnomatix Plant Simulation 12 which is the same software that used in this thesis. The tests revealed that four elements heavily influence complexity measures in simulation models, which are the size of the model, the number of modelled events, the granularity and the complexity of the control logic.
The structure complexity measures layout and connections of the elements in the simulation models, which are measured by M1 and M2. Besides, complexity of objects in simulation model is measured by M3, M4 and M5.
Algorithm complexity is presented by program codes in the simulation model. The code expresses the control logic of a production system. McCabe's cyclomatic complexity measure is used which is suitable for structured programs. The complexity of a program block is equal to the number of predicates in the code plus 1. The overall algorithmic complexity of is the sum of every program block cyclomatic complexity, which is M6. M7 measures the total number of lines in all the program blocks.

Table 2.3: Complexity measures adapted from Popovicsa & Monostori (2016)

Structure complexity	M1: Number of objects M2: Number of connections M3: Number of attributes M4: Number of changed attributes M5: Number of not inherited attributes
Algorithmic complexity	M6: Total cyclomatic complexity M7: Total length of program codes
Computational complexity	M8: Computational complexity (Simulation time)

2.2 Virtual Commissioning

2.2.1 Background

The conventional development of PLC logic is the last phase, followed by control applications and mechanical design (Pellicciari et al. 2010; Bathelt & Meile 2007). Verification and optimization of the code are often lagged until mechanical design is finished. Since PLC code is tested and verified online, engineers must debug the PLC programs against live equipment, integrating conveyor system, robots, safety system and other equipment (Virtual Commissioning 2017). An investigation for the German Association of Machine Tool Builders (VDW) shows that the commissioning control logic takes up 60% of total commissioning time or 15% of total project duration (VDW-Bericht 1997). In order to keep competitiveness and long-term profit, the shorter downtime is highly interested (Lee & Park 2014).
Digital Industry 4.0 enables a digitized value chain (PwC 2015). The Virtual Factory Framework (VFF) is one of the EU funded project. VFF aims to support an integrated virtual environment enabling factory process along all the phase of its lifecycle (Ghielmini et al. 2013).
Simulation is recognized as a useful tool in bottleneck detection, utilization calculation and flow optimization. However, the conventional simulation languages are not suitable to use in detail design (Rullán 1997). In conventional simulation models, control logic is described as independent entity flow between processes. In the detail design phase of production line, the model should acquire the capability to predict both production capability and physical validity of co-working machines and control programs (Park, et al. 2007).
Virtual commissioning is an attractive way to solve the problem in conventional PLC verification (Carlsson et al. 2012). Within such an environment, engineers are able to test and debug control, logistic and transport systems (Versteegt & Verbraeck, 2002) or complex production systems (Hoffmann & Maskoud 2010) before testing on real equipment. It allows mechanical design and control programming in parallel. And the real production is not influenced while the virtual prototypes are under commissioning (Reinhart & Wünsch 2007).

2.2.2 Approach for virtual commissioning

Auinger et al. (1999) and Lee & Park (2014) describe four approaches (Figure 2.2) for PLC logic commissioning. The content of each concept is the same but different names are used in the different studies. They are traditional commissioning, hardware in the loop (soft-commissioning), reality in the loop and software in the loop (offline commissioning). Traditional commissioning involves real production system with physical control system to verify control program.

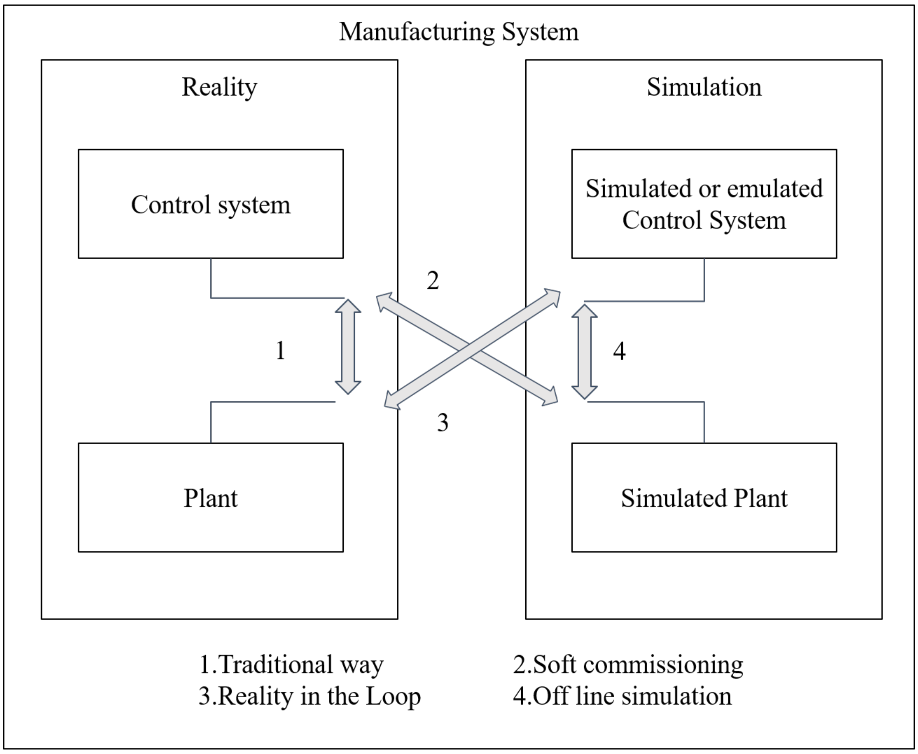


Figure 2.2: Approaches of commissioning (Adapted from Lee & Park 2014 & Auinger et al. 1999)

HIL (Hardware in the loop) deploys real PLC system and simulated automation system. RIL (Reality in the loop) uses simulated PLC system with real process system. SIL (Software in the loop) uses both simulated control system and process system. Both of HIL and SIL are able to identify and correct errors in PLC in a virtual process at the same time the real production is not occupied. Reinhart & Wünsch (2007) compares HIL and SIL. HIL enables complex control program commissioning with different plant levels under laboratory condition with real PLC. In SIL, the plant model and PLC program run on a normal PC, which doesn't require PLC hardware. The problem is that the outdated version of software, resulting in unavailability of some control systems, and abstract model prevents it becoming an exact reproduction of the control behavior. Simulation models in both HIL and SIL are able to get connection with control program by using communication protocol (Dzinic & Yao 2013). In HIL, the problem of running a real PLC with simulation model is so called free-wheeling (Carlsson et al. 2012). Since PLC and simulation run asynchronously, problems occur and affect the quality of virtual commissioning.

2.2.3 Offline simulation

This chapter describes the framework of offline simulation as well as work flow of virtual commissioning proposed by previous study. Park, et al. (2008) propose an architecture of a PLC programming environment which is shown in Figure 2.3. The environment is able to synchronize a PLC program with virtual plant model, in order to carry out visual verification of a PLC program. The environment contains two layers, which are an application layer and a model layer. The application layer, contains plant model visualizer and PLC program simulator. The model layer, it consists of plant model, PLC program and I/O mapping model. The virtual plant model includes equipment and devices of the production facility. The PLC program is the corresponding control logic. The I/O mapping model sets up the channel between the plant model and PLC program. The authors stresses benefits of reusability of a virtual model as well as intuitive recognition of the state transition diagram of a virtual device model.

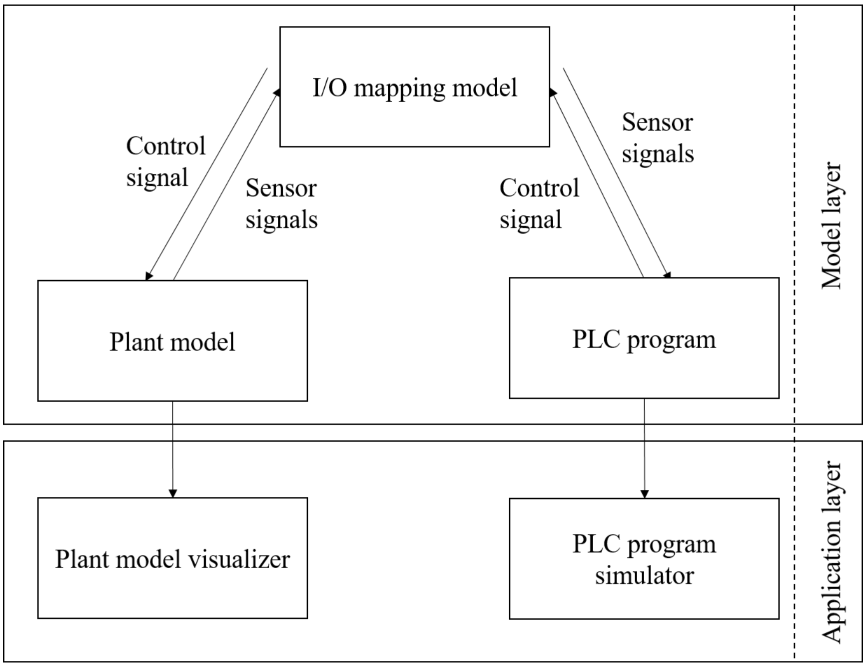


Figure 2.3: Proposed programming environment adapted from Park, et al. (2008)

Guerrero et al. (2014) describe the virtual commissioning process in the environment of Process Simulate. The study follows five steps which are characterizing the system, computer aided design, virtual environments, testing the virtual environments, and virtual environments as a monitoring system.
In the study by Dzinic J. & Yao, C. (2014), steps of setting up simulation based virtual commissioning are described as data collection, 3D component modelling, PLC programming & I/O variables, simulation modelling, and communication establishment between PLC program and simulation model.
Data collection is the first step and plays critical role in the virtual commissioning. Makris, et al. (2012) list required data as following:

1. The 3D model of targeted commissioning object, including data of geometry, kinematic, electronics, and programs.
2. Production layout, position of equipment and facilities.
3. Material flow of associated process and operation sequence.
4. Relevant control systems, either physical PLC or a PLC simulation software.
5. I/O signals of the control system with respective mapping on the resource com-

ponents.
6. Additional components or signals necessary in the virtual commissioning project commissioning process.
7. IT configuration, for example communication protocols, for the networking between the control system and the simulation model.

In the simulation model development, production layout and operation sequence is built. 3D modeling can be classified into component modeling and plant modeling (Hoffmann & Maskoud 2010). Component modeling refers to the low level modeling where components are not available in the library of simulation software. Models of these components have to be built in CAD tools and imported into the simulation software later. Plant modeling refers to high level modelling. Components that are predefined and stored in the library of simulation software have no need to build additionally. The built-in models are equipped with functional interaction of mechanical behaviour with actuators and sensors.
Interfaces between the simulation software Plant Simulation and the Siemens protocol is S5, S7 or PCS7 (Hloska & Kubín 2014).

2.2.4 Cons. & Pros. of VC

Previous study has put much effort in stressing out the advantage of testing control system by virtual commissioning. Several previous studies identify advantages opportunities and possible applicable occasions for virtual commissioning (Drath, Weber & Mauser 2008). The most mentioned benefits are better quality and shorter time in testing control program compared to conventional methods (McGregor 2002; Young & Heider 2002; Mueller 2001; Johansson & Nilsson 2015; Schiess 2001; Siemens 2013). Other advantages are visualization of the control system, which is suitable in purpose of training and education (McGregor 2002; Guerrero et al. 2014). Operational teams and students get more knowledge about "Know How", which improves operational performance and reduces incidents. To summary, virtual commissioning helps verify and correct logic problems off-line, save time and money in field verification and testing. It prevents unnecessary stoppage of production, avoids investment waste, and reduce workload for field employees. Moreover, it allows distant commissioning which would affect the strategy and total cost of commissioning.
Despite of advantages and potentials, drawbacks and difficulties exist when virtual commissioning comes to the reality.
Virtual commissioning is time consuming and costly. The total time spent in activities of modelling, debugging and validation tends to increase as the scale of project becomes larger (Carlsson et al. 2012). Lee & Park (2014) mention the project needs significant amount of time and efforts with regarding to building virtual models. The workload remains the same even though the time is saved in debugging and correction during real commissioning stage since it is done concurrently with other projects.
One difficulty of widely implementing virtual commissioning is the effort of introductory. Drath, Weber & Mauser (2008) point out the high cost of introductory.

The expense includes direct cost and indirect cost. The direct cost includes buying software and training. The indirect cost can be inefficient production due to poor performance at introductory phase. Task of virtual commissioning requires high level of talent. People needs training to get familiar with the tool. It takes time for people to become skillful, which makes hiring specialized people hard and expensive. Reinhart & Wunch (2007) also state cost and skillful people are resistances for virtual commissioning widely used among industry.
Beside perspectives of cost and time, OPC, which plays important role in virtual commissioning, has no mechanisms (Bernhardt & Sabov 2004) to guarantee synchronization, resulting in simulation model and control system running in different pace. This could result in unreliable virtual commissioning, that errors in real world are hidden or fake errors are detected (Carlsson et al. 2012).

2.2.5 OPC

Realistic Robot Simulation (RRS), Fieldbus emulation and OPC, are methods to connect simulation with PLC logic. RRS is a standard interface connecting simulation model and control system of robot. Fieldbus is suitable for the small case where processes run relatively slow (Carlsson et al 2012).
OPC is short for OLE for Process Control, continuously developed by OPC Foundation. It is a universally accepted standard, enabling data exchange between different industrial automation system (Mahnke, Leitner & Damm 2009). OPC aims to solve the problem communication caused by different interfaces from different vendors. OPC is server-client based solution for data exchange of process data (Unified Architecture 2017). Both application of consuming and providing data can be used as a client or as a server (Mahnke, Leitner & Damm 2009). The latest specification is OPC Unified architecture (UA) which overcomes several drawbacks of previous specification.
OPC UA integrates all functionality of the individual OPC classic specifications into one extensible framework (Unified Architecture 2017). The client establishes a connection to the server by creating an OPCServer object. Items with identical settings are grouped together, called OPCGroup object. The updated rate defined by client determines the time gap of cyclic check by server. OPC is also able to monitor the quality of data. It is classified by three categories: accurate (good), not available (bad), or unknown (uncertain).

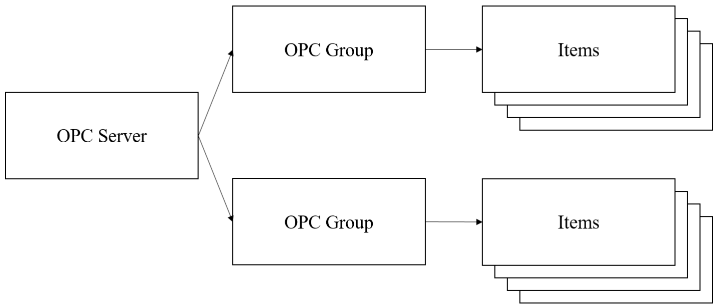


Figure 2.4: Information architecture of OPC UA (Adapted from Mahnke, Leitner & Damm 2009)

2.2.6 Communication through OPC interface

The signal in simulation model is able to connect to the I/O symbols in PLC. By doing this, machines, equipment and conveyor in the simulation model are controlled by real control code. Carlsson et al. (2012) raise a standard OPC communication model containing has 8 components. The framework starts from an application, connecting to the OPC server via OPC client and interface. The OPC server then connects to the gateway via vendor specific interface. The communication ends up at PLC via vendor specific interface.
Johansson & Nilsson (2015) study and test the performance of two software concerning virtual commissioning. One of the software is Plant simulation. The communication used in this software is built in the same way as Henrik Carlsson mentions. In the project, the client application is Plant Simulation. The OPC server is SIMATIC OPC server, which is configured through SIMATIC NET. The PLC controller is Siemens S7-300 CPU 314-2 DP/ PN. Both of OPC server and PLC are from Siemens. IP/ TCP are configured to ensure that PLC is able to talk to OPC server. Additionally, both OPC server and Plant simulation run on the same computer in their project.
Dzinic & Yao (2014) apply SIL in their study where the simulated plant and virtual control system are used. The advantage of this method is that a complete virtual commissioning project can be offered with no hardware required during the design and validation phase. The virtual plant is built in Experior. In the project, Experior HMI is able to communicate with Siemens software/hardware PLC. However, it is not possible to establish a direct connection with Experior and instead has to use an external connection tool NetToPLCSim. NetToPLCSIM connects PLCSIM via S7PROSim COM while it enables connection with Experior HMI via TCP/IP port 102.

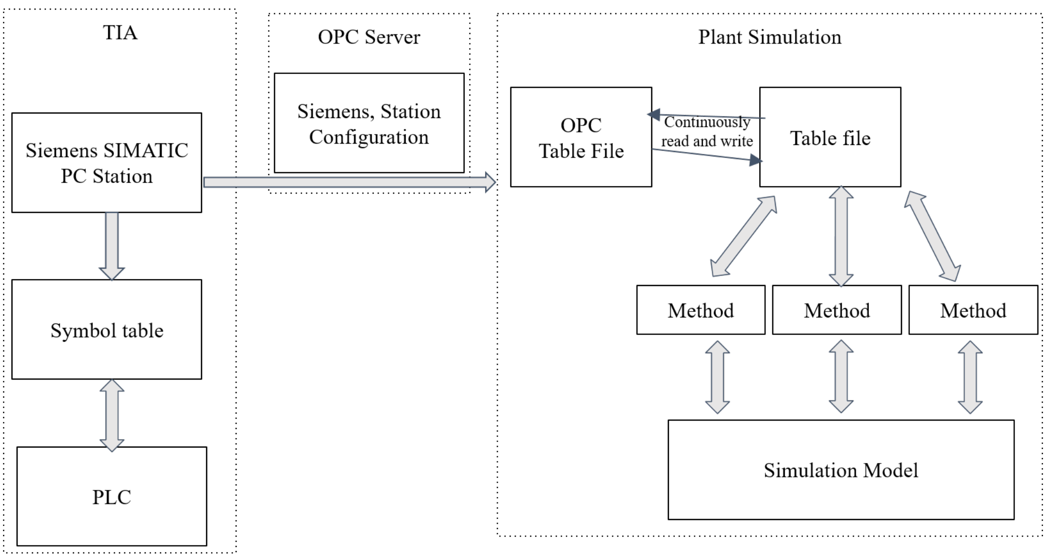


Figure 2.5: Platform structure adapted from Johansson & Nilsson (2015)

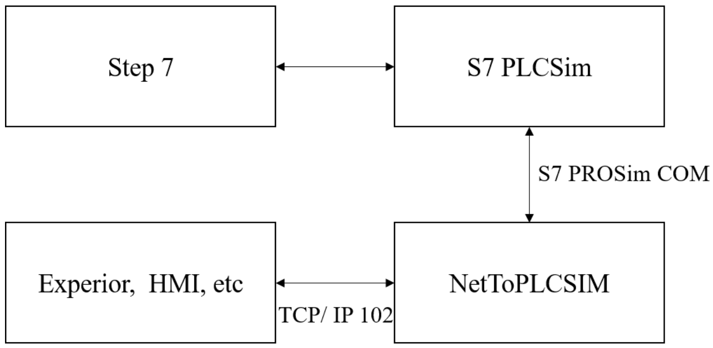


Figure 2.6: Communication frame adapted from Dzinic & Yao (2014)

2.3 Software environment

2.3.1 Tecnomatix-Plant Simulation

The simulation model is built in Siemens Tecnomatix Plant Simulation. Tecnomatix is supported by Teamcenter Manufacturing PLM platform. One of the five basic solutions is manufacturing simulation. Plant Simulation is one of the solutions in the manufacturing simulation. It is a DES tool that aims at creating digital models of logistic systems (Plant Simulation 2017). Engineers are able to create production lines and optimize performance. Several useful analysis tools, are built in and provided in the software.

2.3.2 Objects in Plant Simulation

To simulate behavior of sensors and actuators, *Method* is necessary to create. *Method* is an object that enables "programs control that other objects start and which Plant Simulation then executes during the simulation run" (Reference Help 2016). When events happen, *Method* is used to set or reset the value of input and output value, with the help of *Attributes*. For example, *Method* can embed in the entrance or exit of the conveyor. In Plant simulation, sensors can be triggered by the front or rear side of a coming product, or set in light barrier mode.
All objects have predefined attributes controlling their behavior or representing their state. By clicking the icon *Show Attributes and Methods* in Plant simulation, *Attribute* and *Method* of selected objects are visible.
The programming language used in Plant Simulation is SimTalk. Simtalk is an objective oriented programing language and can be divided in two different parts in Plant Simulation (Bangsow, 2010):
• Control structures and constructs.
• Standard *Method* of the material and information flow objects. They are build-in and they form basic functionality.
Simtalk extends the function of modeling and controlling simulation model. It is very useful in achieving additional function and detailed property.
OPC UA Interface module in Plant Simulation enables access and data exchange between Plant Simulation and automation technology systems. The exchanged data

is *Item* in *OPC UA Interface*. *Read Interval* and *Write Interval* decide the data update interval.

2.3.3 TIA Portal & PLCSIM

TIA is short for the totally integrated automation. It combines functions from digital planning and integrated engineering to transparent operation (Totally Integrated Automation Portal 2017). The main function includes programming, communication, testing, commissioning, documentation and diagnostic.
PLCSIM is a simulation software which provides an environment where the virtual PLC is able to run on a personal computer or programming device. Tags or parameters of PLC program are able to be monitored with the help of built-in user interface. By using PLCSIM, PLC hardware is not a prerequisite to test PLC program.

2.3.4 NetToPLCSIM

NetToPLCSim is a software that enables extension of the PLCSIM by a TCP/IP network interface. It is able to exchange data with PLCSIM, which allows testing HMI/SCADA-Software without real hardware (NetToPLCSim 2015). There are two versions provided currently. One is S7online version. This version uses S7 protocol to transmit data via network layer. It supports multiple connections between clients and PLCSIM. Another version is S7ProSim. Three major drawbacks expose compared to S7online. Data area is limited in this version, as well as speed is slower. The connection between PLCSIM and client is constrained to one. Currently, it is not developed furthermore.
To successfully set up the connection, a 5-step configuration procedure is necessary. A unique user name shall be assigned. The IP address of PLCSIM CPU and LAN-interface shall be reachable. Rack or Slot is determined by the type of CPU.

3
Methodology

In this chapter, the methodology of this master thesis will be presented. Both research questions used some common methodologies, such as literature study and on-site understanding. A platform of integrating PLC signal into simulation model was constructed, of which the result was used in simulation model development. The PLC integrated simulation model of a specific production line was then developed on the platform constructed in the earlier step. The model then was compared with the original simulation model of the same production line with regarding to model performance, specifically model complexity, and model accuracy. A flowchart shows the procedure of workflow is in Figure 3.1. Each area is then described in detail in following chapters.

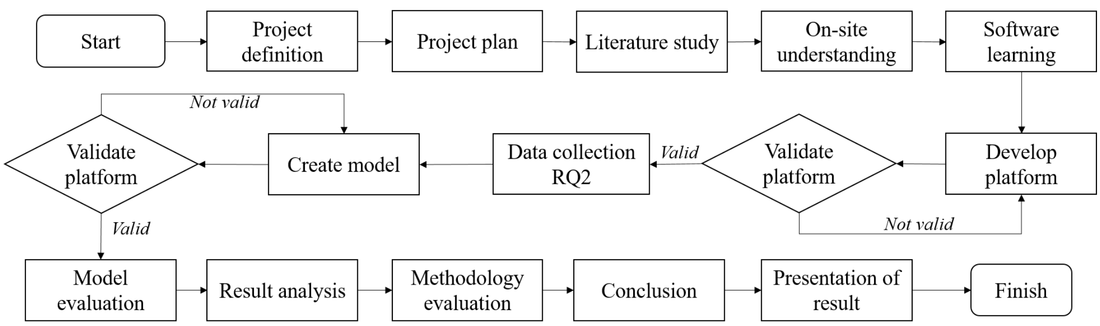


Figure 3.1: A flowchart of work procedure

3.1 Result generation

For result generation, platform construction and simulation model development were analyzed. First, previous study and practice learned from literature study contributed to the current state of PLC signal integration with simulation model. The result of literature gave the positive answer of integrating PLC signal into simulation model. It was also necessary to construct the platform under the circumstance of Swedish Match. Specific literature which had similar software environment were chosen to gain deeper insight of PLC signal integration. Meanwhile, software used in the company, including Plant simulation and TIA Portal, and other necessary software were learned with the help of experienced engineer in the company. The platform was constructed to answer the RQ1 under the current circumstance of Swedish Match. A pilot model was used to validate the the proposed configuration procedure.

As the PLC integrated simulation platform was constructed, the platform was then used in the development of a PLC integrated simulation model of a case study. Swedish Match provided the original DES model and PLC code of the chosen case. The PLC integrated simulation model was developed upon original simulation model, PLC code and platform. The development procedure followed the methodology provided by Banks et al. (2005) with some changes. The change, for example, was no improvement work production system was conducted in this thesis. The simulation model was validated through several methodologies against the real case to confirm its integrity. An analysis was conducted to compare the difference of two simulation models with regarding to simulation performance. Within the scope, time and capability of the thesis, the comparison criteria focused on model complexity and result accuracy. A framework of model complexity from literature study was used to measure the model complexity of the two simulation models. The results of two simulation models were compared against real production data.

3.2 Genchi Genbutsu

To be able to get quick hands-on experience of simulation model and PLC code of the case study, a good understanding of the respective production shall be the first thing to take. Genchi Genbutsu is a method, where the person observes the objects and watch its behavior to gain knowledge of the object. RQ1 and RQ2 has different requirements for on-site tour.

RQ1 requires the understanding of current software and hardware environment used in the company. Therefore, Genchi Genbutsu has focused on learning software used in the thesis as well as hardware configuration. Guide and help from different engineers together with hands-on experience accelerated the process of understanding the environment. The software learning was conducted with the help of engineers from different apartment. A guide presentation was conducted to introduce the IT information of the company which played critical role in developing platform later.

Beside the implementation of the solution from RQ1, the RQ2 requires other steps in order to successfully reach an answer. The procedure of RQ2 requires some changes in the already existing simulation model. Presenting on site to see, analyze and question the production line are essential to gain an in-depth understanding of the production line. Therefore, the on-site study has taken several times to gain the knowledge of production flow, material flow and respective equipment. This procedure will be used to gather knowledge of the execution of the PLC logic in the real system and the behavior of the real system itself.

Two field trips with experienced worker and expert were scheduled. One trip was with experienced worker and another trip was together with the conveyor system developer from a consultant company. A set of questions were raised up regarding with the control system and corresponding physical equipment to attain good understanding of the conveyor system.

3.3 PLC signal integrated simulation development

The creation of PLC signal integrated DES was divided into two interconnected parts.
The first part was to build up the PLC signal integrated simulation. The platform was developed based on the map of signal communication. Within the framework, software includes those are currently used in Swedish Match as well as other software. A procedure of configuration was proposed to enable signal transportation within the framework.
The second part was validation phase. A pilot model was constructed by following the proposed procedure to validate the platform in order to answer RQ1.

3.3.1 Platform construction

The platform construction started from determining the framework. The choice was made based on the software and hardware requirement. Since the platform aimed to integrate simulated PLC signal into DES model, the framework of communication was chosen to achieve communication between DES software with PLC simulation software. Then, effort turned to make research on how to realize communication with currently used software under certain hardware circumstances. The work related with software configuration within each software and setting in company IT environment.

3.3.2 Platform validation

The functionality of the platform was to send and receive signal between Plant simulation and PLCSIM. Both DES and PLC simulator should react the signal received while send out the triggered signal. The function of platform was validated with the help of a pilot model.

The validation process followed by:
a. Interpret the logic, build PLC hard configuration and implement code in the TIA Portal.
b. Construct simulation model in the Plant Simulation.
c. Build communication. Connect Plant Simulation with PLCSIM.
d. Run simulation model controlled by the PLC program to commission the expected result virtually and logically.

The behavior and the result of the simulation run were then compared to the expected result and behavior. If the result lay in the description of system behaviour as well as checklist, the validation was seen as successful. The pilot model was a conveyor system. The conveyor system was controlled by a PLC program. Parts transported through three conveyors, named as "InPath", "Process" and "OutPath". Conveyors were controlled and ran independently. "Process" was one segment, where only one product at a time could be processed. If another product arrived, while a product was on "Process", the "InPath" should be stopped until the product left

"Process". There were three sensors on the conveyor, one at the end of "InPath", one at the beginning and one at the end of the "Process". All conveyors ran at 0.5m/s.

Table 3.1: Components of the pilot model

Name	Attribute	Value
InPath	Conveyor	Length 3 m
Process	Conveyor	Length 9 m
OutPath	Conveyor	Length 3 m
P_Ready	Sensor	Position 2.7 m at conveyor InPath
P_Start	Sensor	Position 3.2 m at conveyor Process
P_End	Sensor	Position 11.7 m at conveyor OutPath
Snus	MU	1.6 m x 0.6 m 0.1 m

3.4 Develop PLC integrated simulation model

With the current simulation model and PLC program as base for the construction of the future model, the development of the future simulation model put effort on two perspectives. The main idea was to integrate the PLC signal from the PLC simulator into the DES model. The first perspective was to construct the absent components in simulation model. The data of those components were collected by field trip. The second step was to integrate the PLC code into the simulation model. Based on Bank's methodology (Banks et al, 2005), the development of the future model was conducted. Some of the early steps were distinguished in the process and others were added or rearranged in the later phase of the development, as result analysis, documentation and reporting. Following steps were used in development of the future model: data collection, verification, validation, experimental design.

3.4.1 Data collection

PLC code worked with the physical components, such as sensors, conveyors and scanners. The knowledge gained from field trip, with the comparison of original simulation model, identified the gap of missing components. The original simulation model provided strong foundation for development for the PLC integrated simulation model. The machine data was provided by Swedish Match. The further development of simulation inherited the data such as working speed, MTTF and MTBF.

Data collection included communication information. Key information such as domain, IP address, firewall, port of communication and user authority, was collected with the help of IT department in Swedish Match. Most of data was collected when carrying out the RQ1. Since the IT environment remained the same during the case study, the IT data was not necessary to collect again.

The input data for the simulation model was the production sequence from the real production. It was collected manually though Genchi Genbutsu from the real system. Time of the arriving units from the real system was collected through Category B collection (Robinson & Bhatia 1995). When the units were arriving to a certain point at the line, that time was noted. In order to distinguish the production variation different joints were used which was addressed for certain products. This data was later on used in the simulation model during the experimental runs. Video recording was performed simultaneously in order to ensure the right number of units was collected and the sequence between the different units were correct. The same video recording was used in order to collect data for the utilization measurement for the real system.

3.4.2 Verification

The verification was performed in three steps. The first two steps verified the PLC logic and simulation separately before the connection was performed to verify the PLC integrated simulation model.
Verification of the PLC logic was executed though a list of activities. The list included all the activities in correct order and what the activities contained. The activities stated what signals were in use and what signals should be set or reset during the execution of the code, as well if the variables in the PLC logic should show a certain value in the activities. If the PLC logic could display the correct behavior based on the different activities in the correct order through PLCSim, the code was verified.
The changes of the current simulation model included parts that enabled the PLC to control the simulation model and it was necessary to verify this before implementation. Code of the future model was gone through to ensure correct I/O signal from the PLC logic was addressed to correct function in Plant simulation. The function stated which functionality the signal had and what part of the simulation model where addressed to (i.e. conveyor or sensor).
When both the PLC logic and the simulation models were verified, the connection was executed. Then the behavior of the PLC signal integrated simulation model was observed to see if the connection was correct and if the model behaved in the right way. This could be done by reusing the list of activities from the PLC logic verification. Instead of using PLCSIM, verification could be done in Plant Simulation graphically through its virtual behaviour.

3.4.3 Validation

The validation was performed through two methods, Black-box and White-box validation. This was to ensure the behaviour of both internal and external of the model. Black-box validation was executed by comparing total performance of model output with historical data (Brooks & Tobias 1996) from the real system. It was based on same input for both simulation models and real system.
White-box validation checked if the behaviour was correct internally by using animation, trace and experts opinion. Animation were executed by comparing the behaviour of different parts of the simulation model towards the real system. It was compared by using statistical result from the simulation runs based on data from the real system. In this case, output values were used (as throughput and utilization) from simulation and reality. Trace was used by watching the graphical

behaviour in the simulation with comparison of that in real system. Certain parts of the simulation were visually compared to same part of the real system in order to guarantee the simulation was good enough to represent that specific part. Expert's opinion was used in order to validate the future model as well.

3.4.4 Experimental design

The PLC signal integrated DES model was compared towards the real system and the original simulation model. Simulation model performance was conducted for comparison.
Accuracy of the result was compared between PLC signal integrated DES model and original simulation model against real system based on output values as throughput and utilization.
The common input was production sequence from the real system. In total 10 hours data of input sequence of the real system was collected in 5 times. Each data collection ran for two hours due to limitation of tryout version OPC server. Each simulation run was based on the data collection of the real system. The simulation run was compared towards the real system based on the same production sequence separately.
Throughput and utilization of the real system was collected through data from video recording from the production line. Result of throughput and utilization from the simulation was compared towards the real system. It was important to compare the models and real system based on same input (Banks et. al, 2005). If the different production sequences is compared between simulation runs and real system the result can be difficult to interpret.

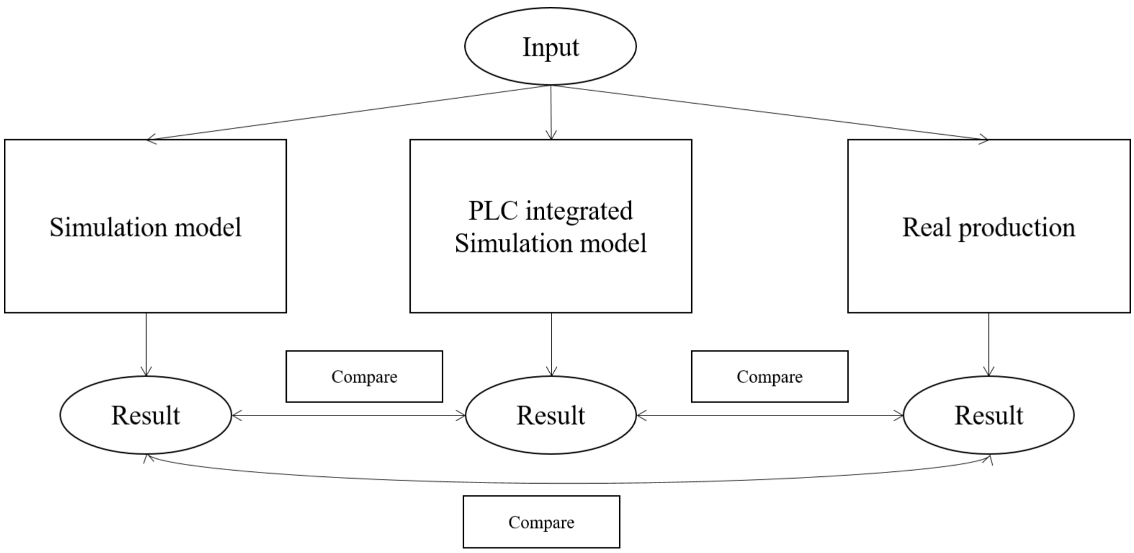


Figure 3.2: Comparison of simulation models and real production methodology

Some delimitation was considered before conducting the experimental runs. In the original model, there was no failure rate, no setup time nor repair. The PLC signal integrated simulation inherited those attributes from the original simulation model. Because of this, the simulation runs were only needed to be conducted once for each

production sequence.
Complexity measurement was compared between PLC signal integrated simulation model and DES model by complexity measurement proposed by Popovicsa & Monostori (2016). The available measures in the project are M1 (number of objects), M2 (number of connections), M3 (number of attributes), M6 (Total cyclomatic complexity), M7 (Total length of codes), M8 (computational time). M4 (number of modified attributes) and M5 (number of non-inherited attributes) are not available because the original case model was not developed by thesis workers. For the M8, the data was acquired from the 5 simulation runs which carried out in the accuracy measurement. The simulation time was documented.

4
Result

Results of the two research questions are introduced. The first subchapter introduces the establishment of communication with simulated PLC and simulated plant model. The second subchapter goes through the model development of the case study. The third subchapter introduces the result of comparison regarding accuracy and complexity.

The IT equipment and software are as following:
Laptop at Swedish Match with Windows 7
Processor: Intel Core i5-2520M @2.5GHz
Memory: 8GB
PLC software: TIA Portal V13, SIMATIC Manager, SIMATIC NET
Simulation Software: Tecnomatix Plant Simulation 13.1
OPC server: KEPServerEX Version 6.1, OPC quick client
Other software: NetToPLCSIM S7online-Version (NetToPLCSim (V0.9.x))

4.1 Platform establishment

In this chapter, the result of RQ1 is introduced. The result gives positive answer for this research question. The answer of RQ1 is described in three sections. The first section describes the framework of the PLC signal integrated simulation platform. The second section describes the procedures for platform establishment. The third part shows the validation result which proves the platform work.

4.1.1 Map of communication & Platform frame

Figure 4.1 shows the map of signal transportation. Signals from PLCSIM, which is the result of the logic, passes through NetToPLCSIM and OPC server, reaching to the OPC UA interface which is embedded in the Plant simulation software. In the OPC UA interface in Plant simulation, if the value of the signal is changed, then the respective control command in Plant simulation is triggered to change the status of relative actuator.
The backward process is when the status of the sensor in the simulation model is changed, relative control command is triggered to change the value of signal in the OPC interface. Then, the signal is sent to Kepware OPC server. The server passes the signal to PLCSIM through NetToPLCSIM. The signal then is the input value of the PLC logic.

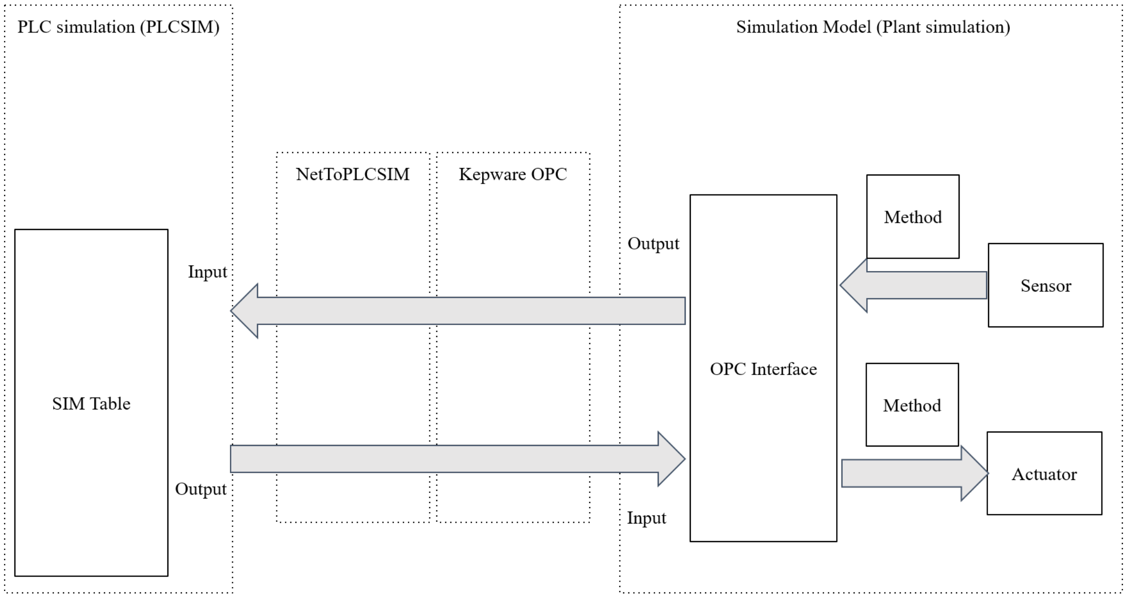


Figure 4.1: Signal transportation map

Based on the signal communication map, the platform is constructed on two computers (see Figure 4.2). On computer 1, it locates the Siemens PLC programming software TIA Portal V13, PLC simulation software PLCSIM, NetToPLCSIM and Kepware server. The simulation software, Plant simulation, is located in the computer 2. In the Plant simulation, the *OPC UA Interface* and *Method* are used. *Method* is the control command which sends signal between simulation model and OPC interface. Between 2 computers, OPC UA is used as communication protocol.

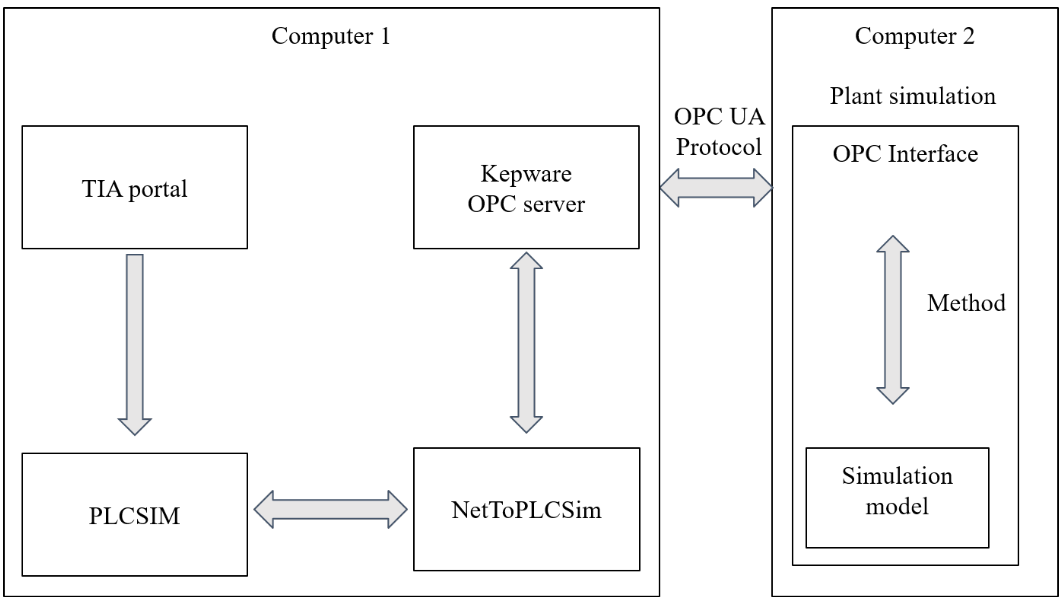


Figure 4.2: Platform frame

4.1.2 Platform construction

To successfully establish the connection, which means that all these software are linked accordingly, there are 4 parts to be configured (See Figure 4.3). Detail procedures are illustrated in Appendix A.

a. Download PLC program to PLCSIM with specified IP address.
b. Configure the NetToPLCSim to establish interface between PLCSIM and OPC server.
c. Configure the Kepware OPC server.
d. Configure the embedded OPC interface in Plant simulation.

Before the targeted PLC program is downloaded from Siemens TIA Portal to PLCSIM, configuration is needed regarding IP address and safety. The IP address is the address of the virtual PLC CPU. Interface of PG/ PC is selected as TCP/ IP. Then tag file is imported. The PLC simulation file is saved and ready for the future use.
Step b is configuration of NetToPLCSIM. NetToPLCSIm plays a role linking PLCSIM with OPC server. Four factors need to be defined, which are user name, IP address of the LAN-Interface where the OPC server runs, IP address of the virtual PLC CPU, and rack & position of the CPU on the machine frame. The IP addresses are visible while clicking “browse” when the simulation PLC CPU is up (The PLCSIM mode is switched to RUN-P) and network is available. Before starting running the NetToPLCSIM, clicking “Get 102” with administrative authority to free the port 102.
Step c solves the communication between OPC server and client. Based on the type of the OPC, there are two different approaches according to whether using OPC DA or OPC UA. OPC UA is chosen to use, this because of two reasons. It simplifies the configuration procedure compared to the OPC DA. OPC DA relies on the Microsoft DCOM technology. The unified architecture of OPC uses a communication stack besides of Microsoft. It reduce of the heavy administration work. Secondly, Plant simulation V13.1 releases OPC UA module, which is a good chance to take the advantage of convenience of OPC UA. In the Kepware OPC, configuration includes communication channels and targeted virtual PLC. PLC tags, which are the times used in the communication, are added. OPC UA needs configure the endpoint which is the address of the OPC UA Interface in the Plant simulation. IT setting is carried out to authorize reaching endpoint through the firewall.
In step d, the connection enables the communication between OPC client and plant simulation. In Plant simulation, the OPC client is embedded in the plant simulation. The module is called "OPCUAInterface". If step c is succeed, the client can able to detect available server by typing down the right URL address. In the module, "group", "name space", "read interval", and "write interval" are defined to successfully talk to OPC server. In the "group", items are defined by "identifier type", "identifier", "data type", "alias", "changed-value control". PLC variables are connected to the respective components. Plant simulation requests data read after "read interval" and writes data after "write interval". After every read from OPC server, changed value item triggers the *Method* notified in "changed-value control". Therefore, the components in Plant simulation can react to PLC signal.

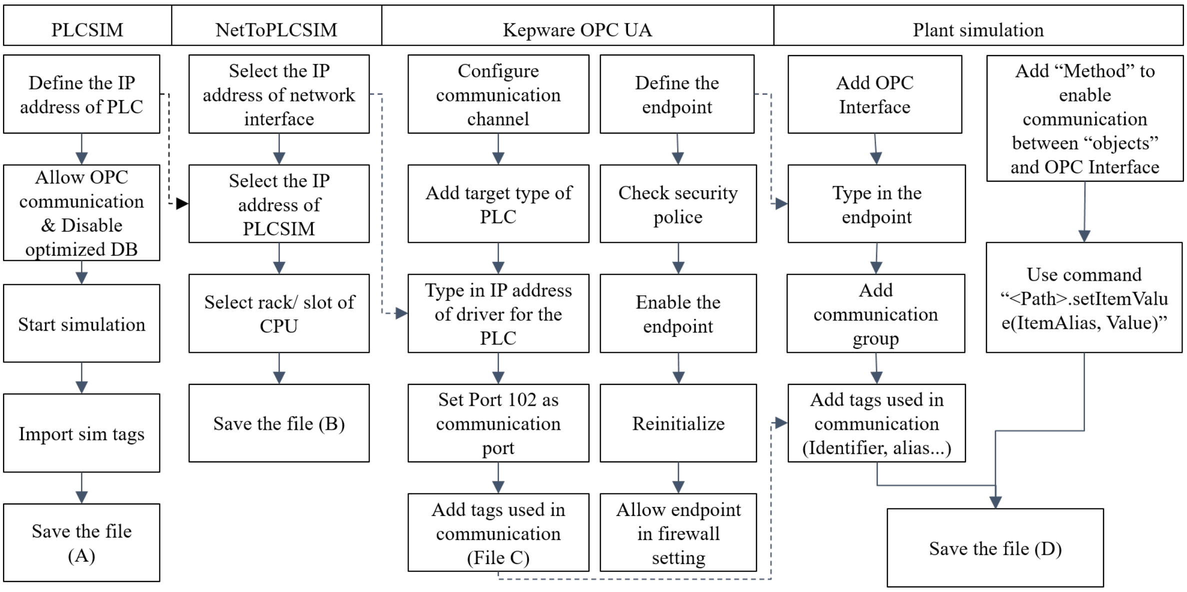


Figure 4.3: Steps of platform construction

4.1.3 Validation

To validate whether successful communication is achieved or not, the test aims at two objectives.
a. The validation model is developed upon the PLC signal integrated simulation platform which is developed based on the procedure described in the previous chapter.
b. Data shall be communicated via OPC server. Input data from Plant Simulation shall be sent to the PLCSIM and output data from PLCSIM shall be sent to Plant Simulation. Both communications success if all the signals update in the Kepware OPC monitor. The function of PLC program shall be commissioned by animation. Based on the steps described in the chapter 3.4.2, the result of validation is shown in following paragraphs. The first step is the implementation of PLC logic. The PLC logic is built in TIA Portal.
The second step is to construct a simulation model in the Plant Simulation. The model is built in Plant simulation based on the description. The model develops production facilities, control logic, OPC UA Interface, signal panels and other accessories.

The third step is to configure Kepware OPC server and NetToPLCSIM settings. The PLC tags are defined in the OPC server. The NetToPLCSIM is configured to connect PLCSIM and OPC server. Then, OPC Interface in Plant simulation is ready to connect with Kepware OPC server.
When all the preparation work is done, the simulation model is ready to run. The validation commissions the model signal communication in OPC server, whether the signals update according to the event continuously. Also, it commissions the expected animation virtually. To validate the function of the system, a set of test scenarios is made. The result of the validation is shown in Table 4.1.

The result shows that the simulation model fulfills all the scenarios both logically and visually. The PLC signals are transferred between PLCSIM and Plant simulation via Kepware OPC server. This result validates the PLC signal integrated platform works, based on the proposed procedure of platform configuration.

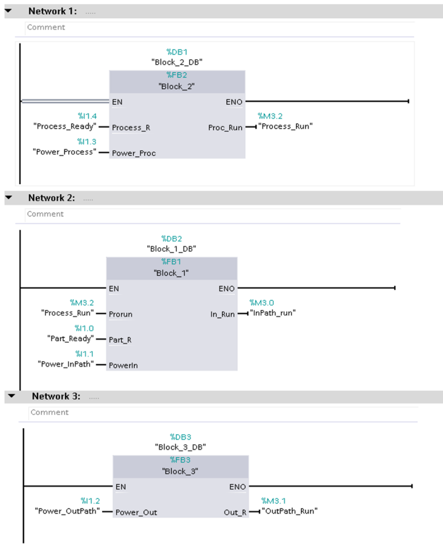


Figure 4.4: PLC programming code block

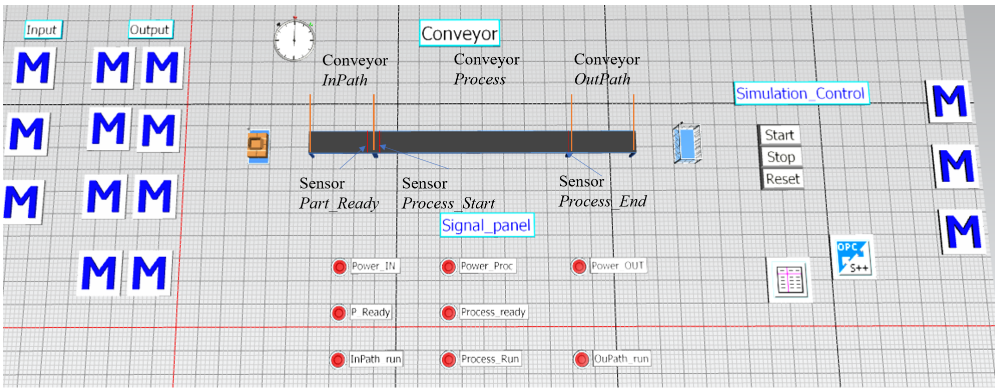


Figure 4.5: Simulation model

Item ID	Data Type	Value	Timestamp	Quality
Channel1.Device1._Rack	Byte	0	14:06:12.451	Good
Channel1.Device1._Slot	Byte	2	14:06:12.451	Good
Channel1.Device1.InPath_Power	Boolean	1	14:11:44.004	Good
Channel1.Device1.InPath_Run	Boolean	0	14:12:30.268	Good
Channel1.Device1.input	Boolean	0	14:06:12.471	Good
Channel1.Device1.OutPath_Power	Boolean	1	14:11:43.545	Good
Channel1.Device1.OutPath_Run	Boolean	1	14:11:43.545	Good
Channel1.Device1.output	Boolean	0	14:06:12.471	Good
Channel1.Device1.Part_Ready	Boolean	1	14:12:30.268	Good
Channel1.Device1.Process_Power	Boolean	1	14:11:43.185	Good
Channel1.Device1.Process_Ready	Boolean	1	14:12:30.211	Good
Channel1.Device1.Process_Run	Boolean	1	14:12:30.219	Good

Figure 4.6: Signal monitoring

Table 4.1: Test scenario

No.	Condition	Input	Result	Output	Pass?
1	All power is on. There is no parts on conveyors.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=0 Process_Ready=0	InPath runs, Process stops, OutPath runs	In_Run=1 Pro_Run=0 Out_Run=1	Pass
2	All power on. There is one part before the first sensor.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=0 Process_Ready=0	InPath runs, Process stops, OutPath runs	In_Run=1 Pro_Run=0 Out_Run=1	Pass
3	All power is on. There is one part at the the first sensor.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=1 Process_Ready=0	InPath runs, Process stops, OutPath runs	In_Run=1 Pro_Run=0 Out_Run=1	Pass
4	All power is on. There is one part arrives at the second sensor.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=0 Process_Ready=1	InPath runs, Process runs, OutPath runs	In_Run=1 Pro_Run=1 Out_Run=1	Pass
5	All power is on. There is one part arrives at the first sensor when Process runs.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=1 Process_Ready=1	InPath stops, Process runs, OutPath runs	In_Run=1 Pro_Run=1 Out_Run=1	Pass
6	All power is on. There is one part leaves the last sensor while another part is at the first sensor.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=1 Process_Ready=0	InPath runs, Process stops, OutPath runs	In_Run=1 Pro_Run=0 Out_Run=1	Pass
7	All power is on. There is one part leaves the last sensor while no part is at the last sensor.	Power_InPath=1 Power_Process=1 Power_OutPath=1 Part_Ready=0 Process_Ready=0	InPath runs, Process stops, OutPath runs	In_Run=1 Pro_Run=0 Out_Run=1	Pass

4.2 Case model development

4.2.1 Description of case study

The conveyor system consists of: an inbound conveyor, 6 short buffer conveyors, 6 long buffer conveyors, an outbound conveyor and two sorting conveyors.
The inbound conveyor line receives carbon boxes from different production lines in the facility. Each production line has its own buffer conveyor. The incoming carbon boxes are sorted to different buffer conveyors. It is handled by a scanner located before the buffer conveyor. The scanner reads the bar code on the carbon box, with which the system refers the carbon box to its correct buffer conveyor. Only one product is in the scan area each time with the help of two sensors. When the product reaches the first sensor, it activates the stopping wheel and scanner. The stopping wheel stops the following products until the first product leaves the scanner area. After the scan area, the product goes into the area where pushers push the box into the buffer conveyor. Along the inbound conveyor, there are 8 sensors to detect the position of the product. The function is to detect the position of the product and push it into the correct buffer conveyor.
There are six short buffer conveyors. The short buffer conveyors receive the carbon boxes from the inbound conveyor. On each buffer conveyor, there is a sensor placed close to the entrance. When a box is sent into the short conveyor, that sensor

gets activated and triggers the short conveyor to move until the product passes the sensor. The sensor also counts the number of products on the conveyor. When there is one batch on the conveyor, the conveyor starts move the products into the long buffer conveyor.
Each conveyor has a sensor placed close to the entrance and has the same function as the sensor of the short conveyor. The batch is sent to the outbound conveyor belt when it has reached its full batch size. FIFO is the rule when products leave. There is another sensor at the end of the conveyor. When that sensor is activated, it blocks other lines move products to the outbound. The inbound conveyor cannot send products to this conveyor neither. When the last product passes through, both long and short buffer conveyors stop.
The outbound conveyor is a single conveyor on which a sensor counts boxes in order to tell when the long buffer conveyors stop. When the first product reaches the sensor, it starts the outbound conveyor and sorting conveyor. The outbound conveyor stops receiving products from other conveyors until the current work is done. When the last product leaves, the outbound conveyor stops.
There are two sorting conveyors. When both conveyors are not occupied, the boxes go into the same conveyor as previous batch. If one of the conveyor is occupied, boxes go into other. If both conveyors are occupied, it locks the outbound as well as long buffer conveyor.

4.2.2 DES model development

Based on the original DES model, changes is made in order to integrate PLC signal. The PLC signal integrated DES model contains sensors on different conveyors. Each sensor is referred to a control *Method* in Plant Simulation. Each *Method* is responsible to send signal to the OPC UA interface. Control *Method* is also created for conveyors which are going to be controlled by the PLC. The result of PLC logic changes value in Plant simulation through these control *Method*. To manage the PLC signal, OPC UA Interface is created with a list of signals and respective control *Method*. Through OPC UA Interface, signals arrive from, or depart to the virtual PLC can be sorted with the right *Method*. The simulation model is showed in Appendix B, figure B1.

4.2.3 PLC program

The new PLC logic reuses the FIFO function from the original code. The FIFO function controls which conveyor is in the line of being emptying to the outbound conveyor. This is controlled by which line has reached to their end position with a full batch. If the line is first in line then that line empties to the outbound conveyor. When it's finished, the second ready batch on another conveyor is set in the first in the list.
The PLC logic handles following parts: short conveyors, long conveyors, and the outbound conveyor. These parts are handled to a different extent by the PLC, dependent on the number of input and output signals involved in that specific part. Signal from the short conveyor are the number of units on the conveyor and the

conveyor status. If a small batch is full and the short conveyor is moving, the long conveyor is set to moving, until the small batch has left the short conveyor. As well, if the FIFO function tells the long conveyor whether the full batch is set as first or not. Five in-signals and one out-signal are used for each line. SG_in, SG_middle, SG_end, Short_conveyor and Outbound_state are in-signals and Long_converyor is the out-signal. The sensor signals SG_in, SG_middle and SG_end control the counting procedure of conveyors. SG_end also sends signals to the FIFO function. For the short conveyor, when signal SG_in is triggered, it increases the first counting variable. When units arrive on the long conveyor, SG_middle decreases the value of the first counting variable and increases the second counting variable. When the first counting variable is reached to 15, the variable buffer1_ready is enabled. Buffer1_ready is disabled when all the units have left the short conveyor to the long conveyor. The same principle applies for buffer2_ready as well. It is enabled when it reaches to 30 and is disabled when it reaches to 0.
Figure 4.7 is an example of the created PLC logic from the first line in the system 1. The P-flank only increases by one when the sensor gets triggered.

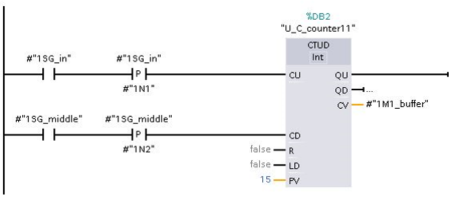


Figure 4.7: PLC code example

Figure 4.8 is the state transition diagram of case study. *P* addresses the current physical position of the batch. *DoT* addresses that one scenario is fulfiled and the batch is sent to its next position *P*. Arrows show which in-signal affect which variable and as well when the output signal is affected.

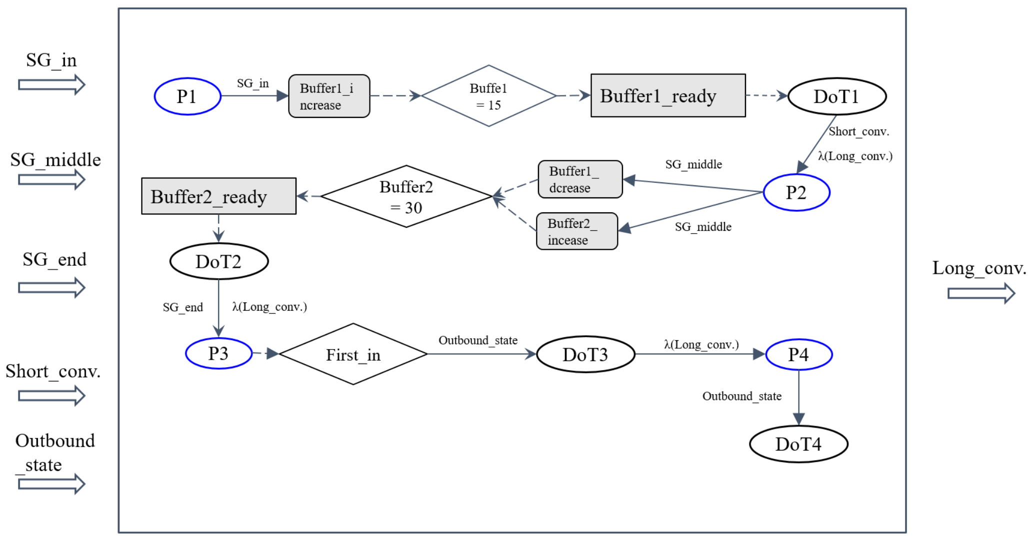


Figure 4.8: State transition diagram

In-signals:
SG_in– Sensor placed at the beginning of short conveyor of the line.
SG_middle– Sensor placed at the middle of the line, between short and long conveyor of line.
SG_end– Sensor placed at the end of long conveyor.
Short_conveyor– Signal if the short conveyor is active or not.
Outbound_state– Signal that determine if the outbound conveyor is occupied or not.
Out-signal:
Long_conveyor– Signal that activate or deactivate the long conveyor of the line.
Variables:
Buffer1 - Keeps track of the number of units on short conveyor.
Buffer2 - Keeps track of the number of units on long conveyor.
Buffer1_ ready - Set when it reaches its maximum capacity of 15, and stay set until it is 0 again.
Buffer2_ready - Set when it reaches it maximum capacity of 30, and stay set until it is 0 again.
First_in - Set if the batch is first in line of all other lines.
Position:
P1 - In the beginning of the short-conveyor.
P2 - In the beginning of the long-conveyor.
P3 - In the end of the long-conveyor.
P4 - In the outbound-conveyor.
Scenarios:
DoT1 - Buffer1_ready is enabled and short-conveyor is set to activated.
DoT2 - Buffer2_ready is enabled and long-conveyor is set to activate.
DoT3 - Batch is placed at P3 and checks if its first in line in the FIFO.
DoT4 - Batch is sent to outbound-conveyor.

4.2.4 Verification

The verification phase was performed. First, the verification of the PLC logic and DES model were separately conducted followed by the mentioned method in Chapter 3, and at last the PLC signal integrated DES model was verified. The PLC logic fulfiled the criteria from the checklist (See Table B.1). The verification of the code in Plant simulation performed by *walked though*. Some signals were addressed to wrong variable and noticed later in the verification phase of the PLC signal integrated DES model. The same checklist was used during verification of the PLC signal integrated DES model. The last verification was essential in that sense it was no simulation runs could be performed in order to determine how functional the simulation model was.

4.2.5 Validation

The two validation methods, black-box and white-box, were used by comparing throughput and utilization from simulation model towards real system. Throughput values are identical for each set of production sequence for both models towards the real system. One of the white box methods *Animation*, shows the utilization of the PLC signal integrated DES model is close enough to represent real system. *Trace* was executed by running the simulation and comparing it with the video recording of different parts of the simulation. The result has showed that all of the parts in the simulation models behaved close enough towards the real system.

4.3 Model comparison

4.3.1 Accuracy comparison

In this subsection, the accuracy measures are presented. The utilization from short and long conveyor is presented together with throughput for each data collection.

Table 4.2: Throughput result

Production sequence	Real System	DES model	PLC model
Run 1	90	90	90
Run 2	150	150	150
Run 3	180	180	180
Run 4	90	90	90
Run 5	300	300	300

Table 4.3 shows the throughput result for each simulation model and from the real system. All the simulation models and the real system show the same throughput result under different production sequence.

4.3.2 Utilization result

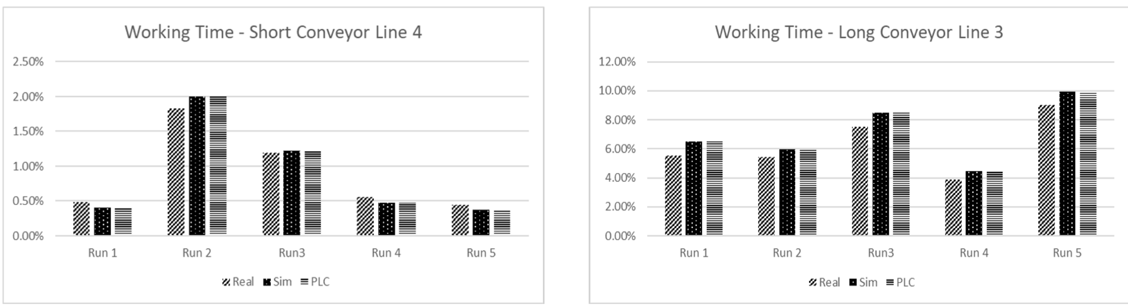


Figure 4.9: Utilization result (example)

The utilization result is presented with the percentage of working time of conveyors. Each chart displays the working time of one conveyor under five production sequences, within the original simulation model, the PLC signal integrated simulation model and the real system. Both simulation models display the same result while the real system in general displays a lower working time than the simulation models. Except for some occasion at line 4 at the short conveyor, the working time was displayed as higher at the real system. The result of all conveyor lines is shown in Appendix B, Figure B.2.

4.3.3 Complexity comparison

The complexity measurement followed the methodology by Popovicsa & Monostori (2016). The comparison measured both simulation models from three perspectives which were structure complexity, algorithmic complexity, and computational complexity.

Structure complexity and algorithmic complexity measures the effort in building up the simulation model. Therefore, the original simulation model and the PLC signal integrated simulation model were measured by parameters from M1 to M7. The result of M1 to M7 is shown in Table 4.4. M4 and M5 were not measured due to lack of data. Since the original simulation model was not developed by the thesis workers, M4 and M5 of the original simulation model was not accessible. Moreover, the PLC signal integrated simulation model was developed based on the original model. It did not make sense to measure the M4 and M5 of the PLC signal integrated simulation against the original simulation model. By this reason, the M4 and M5 were not included in the result.

The result shows that the PLC signal integrated simulation model has higher number in parameter M1, M3, M6 and M7. The result indicates that the PLC signal integrated simulation model builds more objects than that of original simulation model. The coding program, either the complexity of logic or the length of code is much more complex in the PLC signal integrated simulation model. M2 which is the number of connections, remains the same.

Table 4.3: Structure & algorithmic complexity

No.	Name	DES model	PLC int. model	Difference
M1	Number of objects	48	66	18
M2	Number of connections	54	54	0
M3	Number of attribute	4106	4232	126
M6	Total cyclomatic complexity	9	84	75
M7	Total length of program codes	43	556	513

Computational complexity measured the time of simulation run under the given condition. In the comparison, the original simulation model and the PLC signal integrated simulation model run 5 simulations. In every simulation, the production time was set as 1 hour and 55 minutes, which was 6900 seconds. The experiment was repeated five times by using five different production sequences. M8 is shown in Table 4.4.

From the result, the computation time in PLC signal integrated simulation is constantly 6900 seconds. The main reason was that it was not possible to start fast forward simulation in the PLC signal integrated simulation. The time factor was set to "1". The computation time of the original simulation model is around 70 seconds.

The time used running PLC signal integrated simulation model is about 98.5 times than that in the original simulation model.

Table 4.4: Computation complexity (M8)

Runs No.	Original simulation model	PLC signal integrated simulation model
1	70.8 s	6900 s
2	71.0 s	6900 s
3	70.2 s	6900 s
4	67.8 s	6900 s
5	72.1 s	6900 s

5
Discussion

This discussion chapter contains following topics: methodology evaluation, production sequence collection, accuracy measurement, PLC logic, complexity measurement, Plant simulation capability and recommendations for future study.

5.1 Methodology evaluation

The project methodology (Figure 3.1) presents necessary steps of the master thesis project. The research questions have respective methodology. The methodology approach for RQ1 referred to virtual commissioning procedure conducted by Dzinic & Yao (2014), Guerrero et al. (2015) and Johansson & Nilsson (2015). Due to the aim and tools involved in the thesis project, the project drew more effort in communication configuration which is the major difference compared to other studies. The methodology did not include steps of computer aided design and 3D component modelling. There was no additional modelling in this project since objects in Plant simulation library were used. The data collection followed the requirement proposed by Makris, et al. (2012). The project collected data of objects parameters, production layout, material flow, PLC code, I/O mapping model and IT information in both pilot model and case model development.
The methodology of RQ2 was based on Banks et al. (2005) simulation model development. The steps in the project were objectives & project plan, data collection, model translation, verification & validation, experimental design and Documentation & reporting. In the verification step, experts opinion was distinguished in this phase. It was because lack of experts with knowledge of both real system and original simulation model, to evaluate the PLC signal integrated DES model. Instead of experts opinion, trace and animation were used. Validation and accuracy measurement used the same parameters. Both steps were similar but contributed to a different purpose. Future improvement investigation was not considered in the thesis, because no improvement investigation was needed.
The complexity measurement followed the methodology by Popovicsa & Monostori (2016). Measure 4 & 5 were not applicable in the project since the original simulation case model was not developed by these workers. Therefore, the number of modified and non-inherited attributes could be obtained. Besides, the result of computational complexity tended to stress the effect of time dependent signal in the PLC signal integrated simulation.

5.2 Production sequence collection

The video recording procedure ensured that all the units were captured during data collection in different sets of production sequence. A issue occurred in the video recording regarding the first arriving unit from the inbound conveyor. It was harder to capture the correct working time during the first arriving unit of the batch collecting, by watching the video. As the batch grew bigger the easier the collecting was captured. Since the camera was placed at the location to capture the entire production system, parts near the camera were easy to measure; parts in distance were worse captured and harder to document the measures. It could cause an insecure result of the utilization for different lines. For the first arriving unit, a retake was necessary in order to get more reliable result. The movement was easier to interpret as the batch moves to the camera position closer.

5.3 Accuracy measurement

Both simulation models represent the same production system with high similarity. The PLC signal integrated model was created from the basis of the original DES model. Neither changes of the behavior nor speed of conveyors in the model was made. The difference was the PLC program took the role of logic and controlling the model.
Based on the results, there is no significant difference regarding accuracy result between simulation models. The difference occurs in the comparison of the different models utilization towards the reality. Utilization of lines is displayed as lower in the real system, except one line of the five simulation runs. One of the reasons could be different structure procedure in simulation models compared to the real system. One of the examples, happens at the joint of the inbound conveyor and the short buffer conveyor. In both simulation models. The short conveyor starts to move before the unit triggers the sensor (See Figure 5.1). In the real system, the product is pushed into the short conveyor. When it triggers the sensor on the short conveyor, the short conveyor starts to move (See Figure 5.1). The reason of modelling difference is explained in chapter 5.5. The different running time causes a lower utilization result of short conveyors in the real system in comparison to the simulation models.

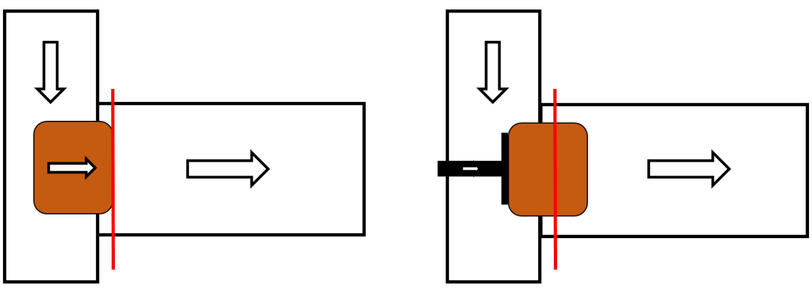


Figure 5.1: Joint schematic in simulation models (left) and real system (right)

Another case explains the different utilization result of long conveyors. In the PLC signal integrated model, the sensor is placed vertically because the diagnostic sensor is not supported in the simulation software (See Figure 5.2). The long conveyor starts to move at the same time as the pre-batch starts to leave the short conveyor. In the real system, the long conveyor starts to move when the pre-batch arrives and triggers the middle sensor which cross references the short and long conveyor (See Figure 5.3). It leads to extra working time of long conveyor and higher utilization in simulation models.

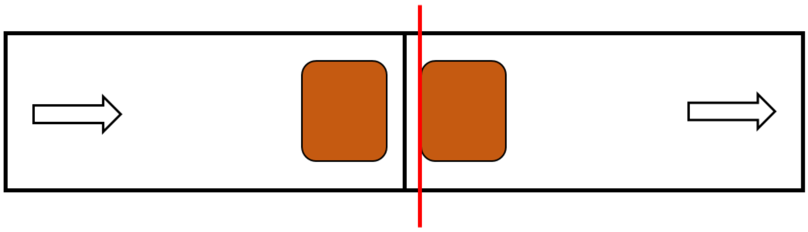


Figure 5.2: Sensor location schematic in simulation models

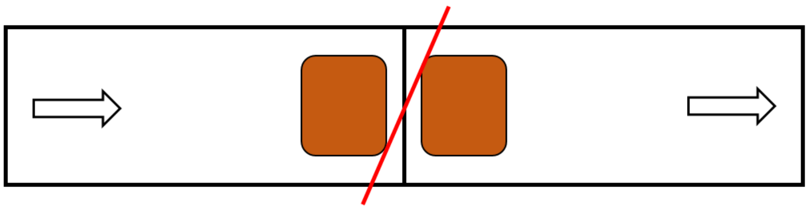


Figure 5.3: Sensor location schematic in real system

Could the cross-referencing sensor be simulated in another way?
In order to close the gap of software incapability and reality, two sensors could be used in instead of one to simulate the procedure closer towards the reality (see Figure 5.4). If one or both sensors are triggered then the long conveyor is set to activate. The proposal illustrates an alternative way to simulate the function of a diagonal sensor, given that diagonal sensor can not be used in Plant Simulation. It is important that the two sensors are placed close in range, that the gap is not too large to deactivate the long conveyor. This alternative solution provides a more accurate utilization result.

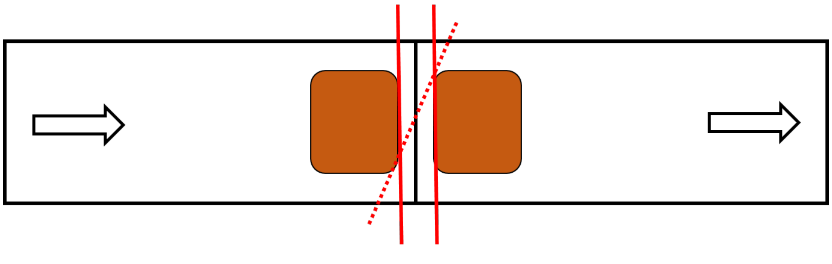


Figure 5.4: Alternative solution, sensor location simulation model

The exact result of the throughput is reasonable. The outbound conveyor sends out the units in batches. If the same number of batches are sent out, the same number of units are sent through. Besides, since the utilization of simulation models and the real system is close to each other, it indicates that the same number of batches and throughput is accurate enough.
In the result of the waiting time at the short conveyor on line 4, some runs displayed a higher value in the real system in comparison towards the simulation models. That kind of result occurs when the number of input units on that specific line is to low. Because of the low number of units going through the conveyor, it could be interpreted as a more inaccurate result. Comparing to a more used line as line 3 where the result is 6% or more at multiple times an assumption could be made that the result is more reliable than from line 4 as an example.

5.4 PLC logic

The PLC logic used in the PLC signal integrated simulation model, reused parts of the PLC code which was used in the real production. The FIFO function was one of the main functions remaining in the PLC logic. It handled the which I/O signal based on the first arriving batch on one of the long conveyors. It enabled the possibility for the project to reuse the function in the PLC logic of PLC signal integrated model. Other parts were not possible to reuse and needed modification due several reasons.
The original PLC code was part of the control program of the whole factory which contained parts that were excluded in the code for the PLC signal integrated DES model. The original code contained more parts of the production than the thesis was delimited to. The original PLC code handled the information flow from the upstream towards downstream, which was excluded from this work.
The PLC logic from the real system was delimited because Plant Simulation could not handle some physical or electrical components. The incapability narrowed the case study to integrate only parts of the functionality of the PLC logic.
The pushing mechanisms at the inbound conveyor were distinguished. The original PLC code contained fuses function for electrical use and acceleration function for the motors of the different conveyors. The new PLC logic can handle counting procedure where signals set and reset by the sensors in Plant Simulation.

Validation of PLC code
There are possibilities to use the PLC signal integrated DES model as a validation tool for validating the PLC code, but this is narrowed to only small scale of what a PLC can accomplished. As mentioned above the functionality could be simulated. If the PLC logic contains a more detailed PLC logic, a more kinematic friendly tool should be used instead of virtual commissioning. PLCSIM could provide a good validation of the logic based on using a checklist (i.e. the one provided in the result chapter). It contained only a display which signals were triggered during use. Some signals were manually triggered by the user, to simulate i.e. arriving and exiting units on the conveyors. Through Plant Simulation the manually triggering is not needed. However, because of the real time use of the PLC, the validation cannot be

as quick in comparison to the PLCSIM.

5.5 Complexity measurement

The result has shown that in the complexity criteria, the PLC signal integrated simulation model is more complex than the original simulation model except M2, M4 and M5. It indicates that the PLC signal integrated simulation model needs to build more objects, program more codes, deal with more complex logic and use much more time than the original simulation model needs.

5.5.1 Structure and algorithmic complexity

In general, the result indicates higher complexity of the control logic and a higher level of granularity needed in the PLC integrated simulation model.
The difference in M1 is the number of sensors on the conveyor belt. There are 18 sensors in the PLC signal integrated model than the original discrete simulation. M2 remains the same because no other objects are inserted other than sensors in the PLC signal integrated simulation model. Sensors are embedded in the conveyor, so there is no need to create connections to link them with conveyors. M3 are the total number of attributes of the simulation models. Each sensor has 7 attributes. Therefore, the difference of M3 is 126.
From the result of M6 and M7, it is found that a more complex control logic and heavier coding workload are necessary in the PLC signal integrated simulation model. The main reason is the control *Method* used for communication with OPC Interface. Every sensor and actuator needs its own *Method*, in total 24. Sensors need respective *Method*s to send signals. The OPC Interface uses a *Method* to trigger each actuators when the value of signals change. In the *Method*, the main function is to react with changed value with the help of predicates, for example with a “if” statement. It leads to majority difference in M6. It also increases the number of lines in the code in the model construction. Other *Method*s, functioning formatting layout, accessories and monitoring conveyors status also increase lines of the code.

5.5.2 Computational measurement

The significant difference between the PLC signal integrated simulation model with original simulation model is the simulation time. The PLC signal integrated simulation model uses real time factor while the original simulation models uses the fast forward time factor. Two reasons lead that the PLC signal integrated simulation model can only run in real time.
To synchronize the time frame in PLCSIM and Plant simulation, it is necessary to set the same time factor in both softwares. However, the PLCSIM used in the thesis doesn’t support the fast forward function. The second reason is that the data communication between the DES model with PLC simulator takes time. Signal takes time travelling between two computers through several software. The PLC

simulator does not receive the data immediately sent from Plant simulation. The same situation happens to the reverse process. The time used in data transportation cannot be decreased with respect to the faster time factor. An example of running simulation model with different time factors is conducted to illustrate the effect of time uncertainty. Four different time factors are set in the experiment, which are 1, 5, 10 and 20. The case is the pilot model used in the RQ1 validation.
The case is that when the first product is running on the *Process*, the second product travels on the *InPath* until it hits the sensor *P_Ready*. When the second product hits the sensor *P_Ready*, the value of *P_Ready* is changed and sent to PLC simulator. The result of logic, which is the *InPath_run*, is changed to 0 and sent to Plant simulation. If Plant simulation receives the value, the *InPath* stops. The second product which running on the *InPath* conveyor also stops. The result documents the stop position of the second product. The same case is running in the DES model as a benchmark. It shall be mentioned that the case of PLC signal integrated simulation is not carried out in ideal Internet environment, that Internet speed is not under control. The case aims to display the influence of time delay intuitively.
In the DES model, the result remains the same even with different time factors (See Figure 5.5). The product stops right at 2.70m. The sensor *P_Ready* is placed at 2.70m. Since the pure DES model is time independent, a faster simulation speed would not influence the result.

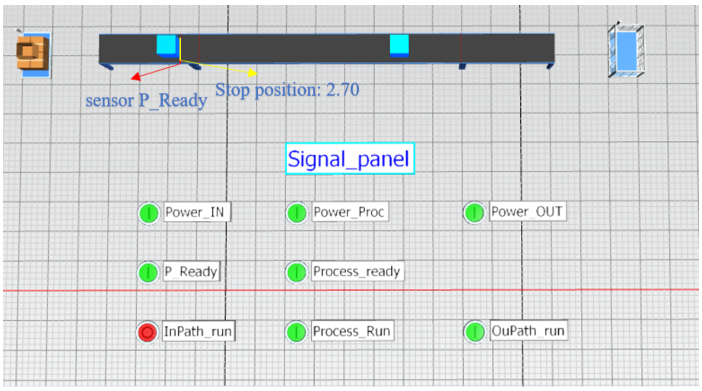


Figure 5.5: Results from DES model (without PLC signal) in different time factors

In the PLC signal integrated simulation model with time factor of 1, the product stops at the distance of 2.75m. The time delay causes 0.05m travel distance. In the time factor of 5, the product stops at 2.84m. In the time factor is 10, the product stops at 3.15m. The time factor of 15, the product stops at 3.30m. (See Figure 5.6) In ideal situation which is the DES model, the product stops almost right at the sensor. However, the effect of time uncertainty is shown up in the PLC signal integrated simulation model. The product keeps travelling until Plant simulation receives the signal. As the time factor increases, the effect of time uncertainty magnifies. It causes strange or even wrong behavior in the simulation model. In the cases of time factor 10 and 15, the product already passes the *InPath*, entering the *Process*. When the time factor increases to 10 for example, the actual speed of conveyor increases 10 times. Since the time of delay remains the same, the product travels 10 times longer distance until the signal reaches Plant simulation.

The result of this example shows the problem of time uncertainty, as well as lack of synchronization mechanism in OPC UA, which lies on the same road as the study by Carlsson et al. (2012).

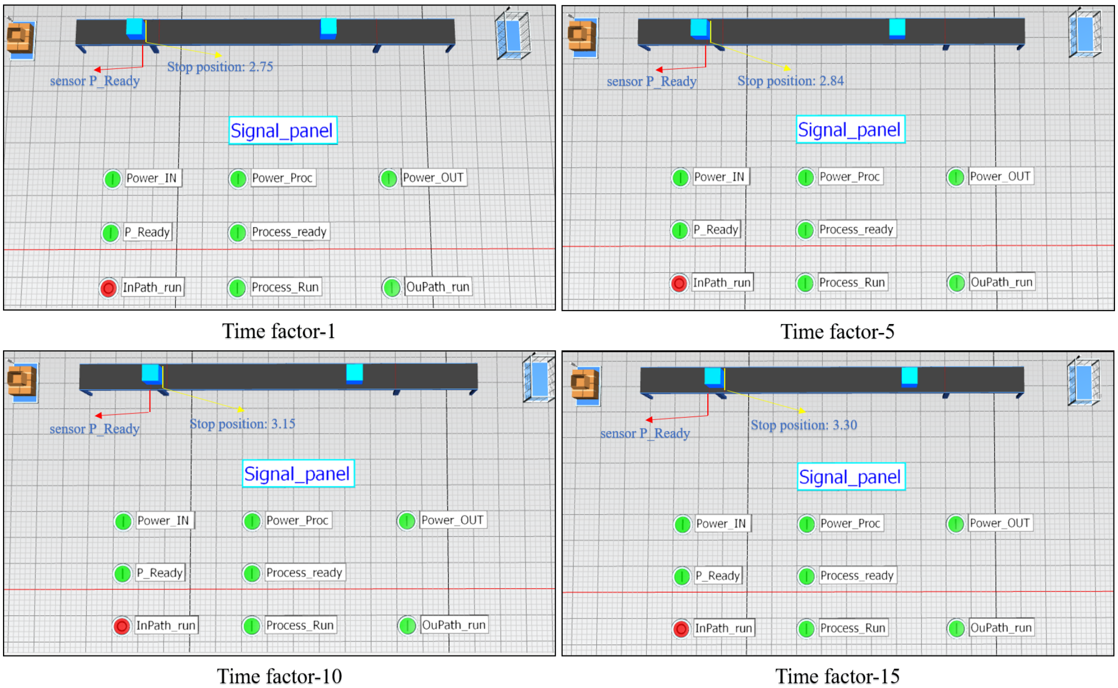


Figure 5.6: PLC signal integrated simulation in different time factors

Since the complexity of models heavily influence the workload of simulation model development, it infers to more time consuming and cost in the project of PLC signal integrated simulation model compared to the discrete event simulation.

5.6 Plant simulation capability

In the case study, it is found that Plant simulation lacks some physical modules which represent the real equipment. It influences the re-usability of the original PLC code in the case study and drives effort in modifying the code. The following equipment cannot be found in the library of Plant simulation in the current version.
The barcode scanner which reads the QR code on the label of the product, is not available in the library of Plant simulation.
The sensor can be only placed vertically with the conveyor belt. The diagonal sensors are not supported in the software.
The stop wheel in real production cannot be emulated in Plant simulation.
The pusher is not available in Plant simulation.
The signal from these components are not possible to generate, and send to PLC simulator. It leads to some PLC code is disabled due to lack of necessary signals. Therefore, the PLC code in the case study is modified to fit in the function provided by Plant simulation.
For example, the pusher is the machine which pushes the carbon box into respective

conveyor belt while several lines of PLC code work in the background to tell when and how does the pusher work. In Plant simulation, routine choice refers to an embedded module called converter. The converter decides and chooses which routine shall the carbon box go along with by judging the attributes of incoming carbon boxes. The original PLC code is not suitable in this case, but should be modified in order to fit in the function of the converter.
The thesis work of Johansson & Nilsson (2015), stated that the build-up of the PLC signals were faster for the Virtual commissioning software (Simuatik3D) in comparison of the DES software (Plant Simulation). Simuatik3D was more user-friendly when it comes to connecting the PLC into the software. Troubleshooting and debugging was more comprehensive for the Plant Simulation platform because all the additional coding for the I/O-signal handled by the software. They mentioned as well there were some delimitation with the modules in Plant Simulation to represent real system as the same level as that in the Simuatik3D model. Because of this, the same PLC logic did not fit both simulation models. It leaded them to re-structure the code for the Plant Simulation model. There were a small differences between the result of two simulation model and could be affected by the different structures in PLC logic and model structures. Johansson & Nilsson (2015) did not compare their model toward a real system. The accuracy of the Plant Simulation model was compared towards Simuatik3D model.
Discrete event simulation provides a great possibility to represent a real system to simulate its behaviour and find future improvement potential (Banks et al, 2005). As a validation tool for PLC logic, there are still some capability problems in the software as Plant Simulation. In the thesis, modification of the PLC logic received from the company was necessary to conduct to collaborate with DES tool.

5.7 Future study

The future study could involve the following:
In the case, the data collection of the utilization measures and also the production sequences could be more precise if the data was collected automatically. The manual data collection could have some affect the end result in the accuracy measurement. In future studies, regarding time dependent measurement, to avoid human error, it is more preferable to use computerized tools. It reduces the human error. Automated collection of throughput could also be preferable, but it is not needed in the same extent as the time dependent data collection.
The data type used in the thesis is limited to Boolean. However, the OPC server supports other types of data than Boolean, such as string, time and integer. For the future study, the investigation to explore the possibility to integrate other types of data can be made.
The PLC signal integrated simulation model is developed based on a built simulation model. The original simulation model was not created by the thesis workers. It leads to unavailable data in the measure of M4 and M5 when comparing complexity of two simulation models. To obtain full version of the complexity measures, the future work could develop two simulation models from the same base. The data of M4 and M5 of two the simulation models then would be available by comparing

against the base condition.
As the software is continuously updating, new functions and updated patches are interesting to test. The future version of Plant simulation will implement direct interface to PLCSIM advanced. In this project, due to the software version, there is no chance to try out the direct Interface. The new interface spends less time in data communication than OPC UA protocol. A faster communication is attractive to test with a higher time factor, reducing simulation time possibly.

6
Conclusion

RQ1: ***How can engineers use PLC signal as input to a simulation model?***
The result gives a positive answer that it is possible to integrate the PLC signal into the simulation model within the provided environment. The PLC signal integrated simulation platform was constructed by following steps described in Appendix A. The Platform is constructed on a software which are build on Plant simulation, Siemens PLCSIM, Kepware OPC, and NetToPLCSIM. This is running on two computers and successfully fulfill the aim. The signal transported between Plant simulation and PLCSIM via Kepware OPC server by using OPC Unified Architecture communication protocol.

RQ2: ***What are differences of PLC signal integrated simulation model compared to an ordinary simulation model?***
Based on the result, there are no differences in accuracy measures between the two simulation models. Both the original simulation model and PLC signal integrated model showed the same values in throughput and working time for each simulation runs. Therefore, it is stated that the integrated PLC signal does not affect the accuracy performance. When it comes to the complexity result, the PLC signal integrated model displayed a higher value on each category except M2, M4 and M5. It indicates that the PLC signal integrated model is more complex in compassion towards the original model. The extra effort in enabling the communication and real time factor increase the complexity. The conclusion is that the difference between two models can not be distinguished in compassion of only performance result of the simulation itself. The difference occurs due to the different structures, coding and time factors in simulation models.

Bibliography

[1] Arthur,J.D., Sargent, R.G., Dabney, J.B., Law, A.M., & Morrison, J.D. 1999. 'Verification and validation: what impact should project size and complexity have on attendant V&V activities and supporting infrastructure?'. In: Proceedings of the 1999 Winter Simulation Conference, ed., P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, pp.148 - 155. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey

[2] Auinger, F., Vorderwinkler, M., & Buchtela, G. 1999. 'Interface driven domain-independent modeling architecture for "soft-commissioning" and "reality in the loop.' In: Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future, Volume 1, pp.798–805.

[3] Bangsow, S. 2010. 'Manufacturing Simulation with Plant Simulation and SimTalk'. Springer-Verlag Berlin Heidelberg.

[4] Banks, J., Carsson, J.S., Nelson, B.L., & Nicol, D.M. 2005. 'Discrete-Event System Simulation'. (Fourth edition). Prentice-Hall, Upper Saddle River, New Jersey.

[5] Bernhardt, R.& Sabov, A. 2004. 'Virtual automation system standards'. In: Proceedings of IFAC Cost Oriented automation, Gatineau/ Ottawa, Canada, 2004, pp.43-48.

[6] Balci, O. et al. 1996. 'Department of defense verification, validation and accreditation (VV&A) recommended practices guide'. Alexandria, VA: Defense Modeling and Simulation Office.

[7] Bathelt J. & Meile J. 2007. 'Computer Aided Methods Supporting Concurrent Engineering when designing Mechatronic Systems Controlled by a PLC'. In: Proceedings of International Conference on Manufacturing Automation, 2007, Singapore.

[8] Brooks, R. J. & Tobias, A. M. 1996. 'Choosing the best model: level of detail, complexity and model performance'. In: Mathematical and Computer Modelling, Volume 24, Issue 4, August 1996, pp. 1-14.

[9] Carlsson, H., Svensson, B., Danielsson, F., & Lennartson, B. 2012. 'Methods for Reliable Simulation-Based PLC Code Verification'. In: Proceedings of IEEE Transactions on Industrial Informatics, Volume 8, Issue: 2, May 2012), pp. 267 - 278. DOI: 10.1109/TII.2011.2182653

[10] Carson, J.S. 2002. 'MODEL VERIFICATION AND VALIDATION'. In: Proceedings of the 2002 Winter Simulation Conference. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

[11] Chwif, L. 2000. 'ON SIMULATION MODEL COMPLEXITY'. In: Proceedings of the 32nd conference on Winter simulation, pp. 449-455. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

[12] Drath, R., Weber, P. & Mauser, N. 2008. 'An evolutionary approach for the industrial introduction of virtual commissioning'. In: Proceedings of 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, 2008, pp. 5-8.

[13] Dzinic J. & Yao, C. 2014. 'Simulation-based verification of PLC programs'. Master's Thesis, Chalmers University of Technology. Available at: Chalmers Library. [13 February 2017].

[14] Efthymiou, K., Pagoropoulos, A., Papakostas., N., Mourtzis.,D& Chryssolouris, G. 2012. 'Manufacturing Systems Complexity Review: Challenges and Outlook'. Procedia CIRP, Volume 3, 2012, pp. 644–649.

[15] Fishman, G.S. 2001. 'Discrete-Event SImulation: Modeling, Programming and Analysis'. New York: Springer-Verlag Inc.

[16] Ghielminia, G., Pedrazzoli, P., Roverea,D., Terkajb, W., Boëra, C.R., Masoc, G.D., Milellad,F., & Saccob, M. 2013. 'Virtual Factory Manager for Semantic Data Handling', In: Proceedings of CIRP Journal of Manufacturing Science and Technology 6(4):281–291.

[17] Glas, J., 1993. Standardisierter Aufbau anwendungsspezifischer Zellenrechnersoftware. MIT 80 Abb. Berlin [u.a.], Springer (1993) (iwb-Forschungsberichte; 61).

[18] Guerrero, L.V., López, V.V. & Mejía, J.E. 2014. 'Virtual Commissioning with Process Simulation (Tecnomatix)'. Computer-Aided Design and Applications, pp.S11–S19.

[19] Hloska J.& Kubín M. 2014. 'Virtual Commissioning of Mechatronic Systems with the Use of Simulation'. In: Březina T., Jabłoński R. (eds) Mechatronics 2013 pp. 33-40. Springer, Cham.

[20] Hoffmann, P.& Maskoud, T. 2010. 'Virtual commissioning of manufacturing systems: a review and new approaches for simplification'. In: Proceedings of the 24th European Conference on Modeling and Simulation, pp. 175–181 (2010)]

[21] Input Data Management 2015. MPR271–Simulation of Production Systems, Department of Product and Production development, Chalmers University of Technology.

[22] Johansson, M.& Nilsson, J. 2015. 'Verification of PLC logic in simulation programs'. Examensarbete inom huvudområdet Automatiseringsteknik Kandidatnivå 30 Högskolepoäng, Vårtermin 2015.

[23] Kampa, A., Gołda,G. & Paprocka, I. 2017. 'Discrete Event Simulation Method as a Tool for Improvement of Manufacturing Systems' Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Silesian University of Technology, ul. Konarskiego 18A, 44-100 Gliwice, Poland

[24] Lee, C.G. & Park, S.C. 2014. 'Survey on the virtual commissioning of manufacturing systems'. In: Journal of Computational Design and Engineering, 1(3), pp.213–222.

[25] Makris, S.,Michalos, G. & Chryssolouris, G. 2012. ’Virtual Commissioning of an Assembly Cell with Cooperating Robots’. In: Proceedings of Advances in Decision Sciences, 01/2012.

[26] Mahnke, W., Leitner, SH., & Damm, M. 2009. ‘OPC Unified Architecture’, Springer-Verlag Berlin Heidelberg.

[27] Majid, M.A., Uwe Aickelin, U.& Siebers, P.O. 2010. ‘Comparing Simulation Output Accuracy of Discrete Event and Agent Based Models: A Quantitative Approach’. School of Computer Science, University of Nottingham, Nottingham, UK.

[28] McGregor, I. 2002. ’The relationship between simulation and emulation’. In: Proceedings of the 34th conference on Winter simulation: exploring new frontiers, pp. 1683-1688.

[29] Mueller, G. 2001. ’Optimizing PLC Controls on a High- Speed Bottling Line’. In: Proceedings of 2001 Symposium. Salt Lake City, UT: Brooks-PRI Automation, Inc. NetToPLCSim 2015. Thomas Wiens.

[30] Park, S.C., Park, C.M., Wang, G.M., Kwak, J. & Yeo, S. 2008. ’PLCStudio: SIMULATION BASED PLC CODE VERIFICATION’. In: Proceedings of the 2008 Winter Simulation Conference.

[31] Park, S.C., Park, C.M. & Wang, GN., 2007. ’A PLC programming environment based on a virtual plant’, In: Proceedings of The International Journal of Advanced Manufacturing Technology, December 2008, Volume 39, Issue 11, pp 1262–1270.

[32] Pellicciari, M., Andrisano, A.O., Leali, F. & Vergnano, A. 2010. ’Engineering method for adaptive manufacturing systems design’. In: International Journal on Interactive Design and Manufacturing 3(2), pp: 81–91.

[33] Pidd, M. 1996. ’Five simple principles of modelling’. In: Proceedings of the 1996, Winter Simulation Conference, ed. J. M. Charnes, D.M. Morrice, D. T. Brunner, and J. J. Swain, pp. 721-728. Institute of Electrical and Electronics Engineers, Piscataway, N.J.

[34] Plant Simulation 2017. Siemens Product Lifecycle Management Software Inc. Available at:<https://www.plm.automation.siemens.com/en_us/products /tecnomatix-/manufacturing-simulation/material-flow/plant-simulation.shtml>. [11th April 2017]

[35] Popovicsa, G. & Monostoria, L. 2016. ’An approach to determine simulation model complexity’. In: Proceedsing of Procedia CIRP 52 (2016), pp. 257–261.

[36] Reference Help of Plant Simulation 2016. Siemens Product Lifecycle Management Software Inc.

[37] Reinhart, G., & Wünsch, G. 2007. ’Economic application of virtual commissioning to mechatronic’. In: Production Engineering December 2007, Volume 1, Issue 4, pp. 371–379.

[38] Robinson, S. 1997. ‘SIMULATION MODEL VERIFICATION AND VALIDATION: INCREASING THE USERS’ CONFIDENCE’. In: Proceedings of the 1997 Winter Simulation Conference ed. S. Andradbttir, K. J. Healy, D. H. Withers, and B. L. Nelson

[39] Robinson, S., & V. Bhatia 1995. 'Secrets of successful simulation projects'. In: Proceedings of the 1995 Winter Simulation Conference. C. Alexopoulos et al. (eds). 61/67, IEEE, Piscataway, New Jersey.

[40] Rullán, A. 1997. 'Programmable logic controllers versus personal computers for process control', In: Proceedings of Computers & Industrial Engineering, Volume 33, Issues 1–2, October 1997, Pages 421-424.

[41] Salt, J. D. 1993. 'Keynote address: simulation should be easy and fun!'. In: Proceedings of the 1993 Winter Simulation Conference, ed. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 1-5. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

[42] Sargent, R.G. 2010. 'VERIFICATION AND VALIDATION OF SIMULATION MODELS'. In: Proceedings of the 2010 Winter Simulation Conference, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

[43] Sargent, R.G. 2005. 'Verification and Validation of Simulation Models'. In: Proceedings of the 2005 Winter Simulation Conference, Arlington, 130-143.

[44] Sargent, R.G. 1984.' Simulation model validation, Chapter 19'. In: Simulation and Model based Methodologies: An Integrative View. Springer-Verlag: Heidelberg, Germany, pp 537–555.

[45] Sargent, R.G.1982. 'Verification and validation of simulation models'. Chapter IX. In: Cellier FE (ed). Progress in Modelling and Simulation. Academic Press: London, pp 159–169.

[46] Schiess, C. 2001. 'Emulation: debug it in the lab — not on the floor'. In: Proceedings of the 33nd conference on Winter simulation, pp. 1463-1465.

[47] Shannon, R.E. 1998. 'Introduction to Art and Science of Simulation'. In: Proceedings of the Winter Simulation Conference, eds. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, Washington, DC, Dec. 13-16, pp. 7-14 Siemens 2013. KUKA Systems do Brasil. Siemens Product Lifecycle Management Software Inc.

[48] Schiess, C. 2001. 'Emulation: debug it in the lab — not on the floor'. In: Proceedings of the 33nd conference on Winter simulation, pp. 1463-1465.

[49] Skoogh, A. & Johansson, B. 2008. 'A METHODOLOGY FOR INPUT DATA MANAGEMENT IN DISCRETE EVENT SIMULATION PROJECTS'. In: Proceedings of the 2008 Winter Simulation Conference, S. J. Mason et. al eds.

[50] Tecnomatix Realize innovation 2017. Siemens Product Lifecycle Management Software Inc. Available at:<https://www.plm.automation.siemens.com/en_us/products/tecnomatix/>. [03 March 2017]

[51] Totally Integrated Automation Portal 2017. Siemens AG, 1996 – 2017. Available at: <https://www.industry.siemens.com/topics/global/en/tia-portal/pages/default.aspx>

[52] Trybul, W.J. 1994. Building simulation models without data. In: Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference.

[53] Unified Architecture 2017. OPC Foundation, The Industrial Interoperability Standard™.

Avaiable at: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. [11th April 2017]

[54] VDW-Bericht, N. 1997. Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen, Verein Deutscher Werkzeugmaschinenhersteller, Aachen (1997).

[55] Versteegt, C. & Verbraeck, A. 2002. 'The Extended Use of Simulation in Evaluating Real-Time Control Systems of AGVs and Automated Material Handling Systems'. In: Proceedings of the 2002 Winter Simulation Conference, pp. 1659-1666, Piscataway, NJ.

[56] Virtual Commissioning, 2017. Siemens Product Lifecycle Management Software Inc.

[57] Young & Heider 2002. 'Emulating an Order Fulfillment System'. In: Proceedings of 2002 Symposium. Salt Lake City, UT: Brooks-PIU Automation, Inc.

A

Platform establishment procedure

This document introduces the procedure of setting up the communication between PLC simulation software PLCSIM and DES software Plant simulation via OPC software Kepware OPC. The third-party software NetToPLCSIM takes part in the communication as well.
This document is divided into two sections. The first section introduces the configuration of connection from the very beginning till successful establishment of communication.
The second section introduces the procedure of communication verification. The aim is to make sure that configuration is right. The result of this phase is that communication is built and ready to use for simulation run.

A.1 Communication configuration

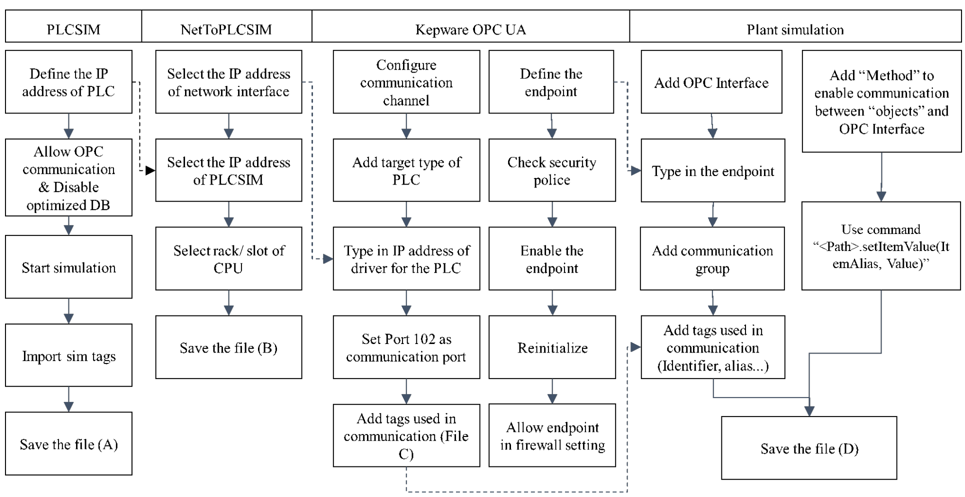


Figure A.1: General procedure

The communication between simulation software and PLC simulation goes through several stages. As the OPC server plays core role transporting data, the configuration divides in two parts based on relationship of server and client. Configuration on server and client computers are introduced separately.
Figure A.1 shows the general procedures. Steps linked by the dash arrows share the

same information.

A.1.1 Server-side configuration

At server side, there locates software including TIA portal, PLCSIM, NetToPLC-SIM and OPC server. This chapter introduces configuration of each software as well as IT configuration.

A.1.1.1 Download PLC program

Add a new subnet. Define the IP address of the PLC CPU.

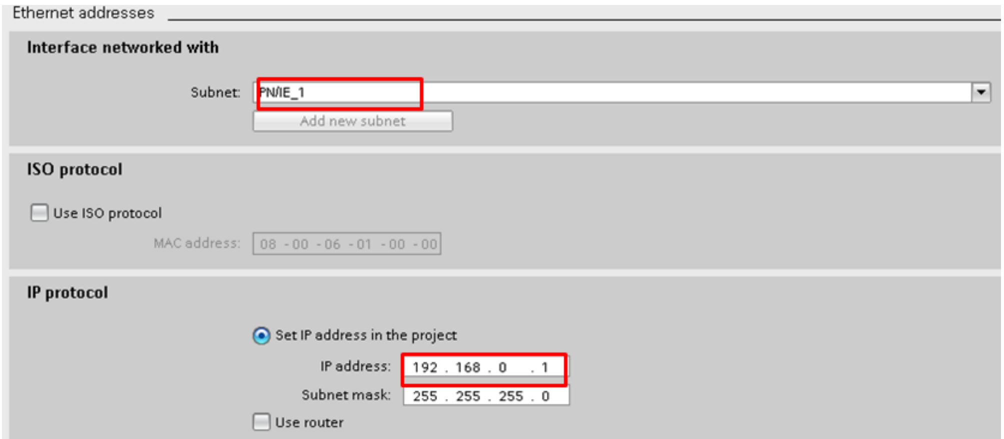


Figure A.2: Assign IP address

Select the PLC that is used in the project tree. Compile the program by clicking "Compile" icon on the task bar. Click "Start simulation" icon to start the PLC simulation. Save the simulation file.

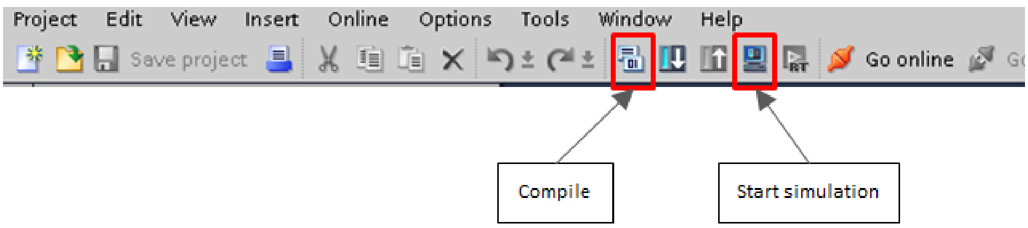


Figure A.3: Compile & Start simulation

When PLC CPU 1500 series is used, it is necessary to tick "Permit access with PUIT/GET communication from remote partner (PLC, HMI, OPC, . . .)" in "Connection mechanisms".

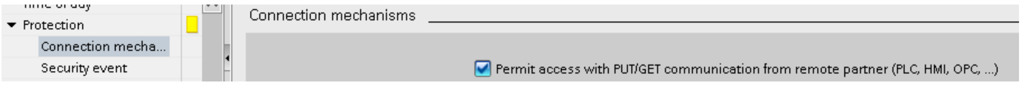


Figure A.4: Tick security option

In the case study, it is found that the "I" is not working for the CPU 1500 series when connecting OPC server. Therefore, "M" is used instead of "I" for input data although in the reality input data uses "I".The CPU 300 series works fine with "I".

A.1.1.2 NetToPLCSIM configuration

Start NetToPLCSIM program by administration authority.
Click "Add" icon to add a new connection.

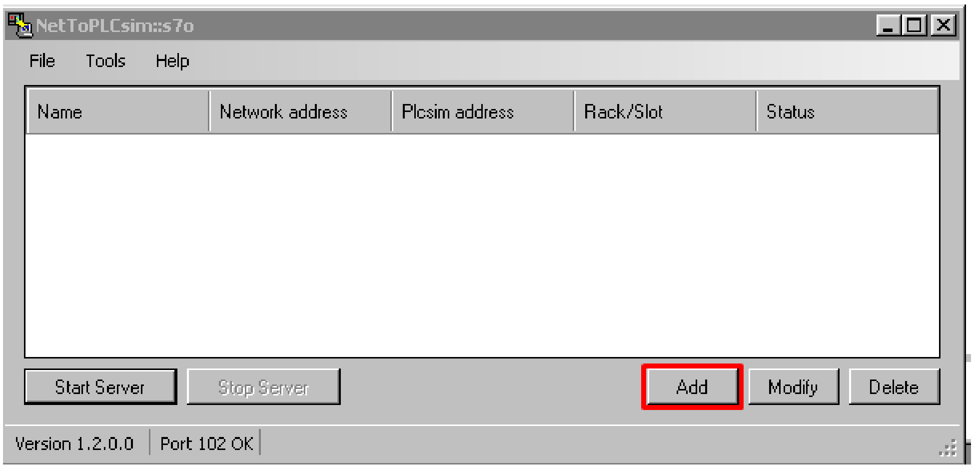


Figure A.5: Add a connection

Insert name in "Name" field. In "Network IP Address" field, insert the IP address of network which the OPC server is running on. Click "..." icon to check the available network. Here, the IP address of local computer is used.

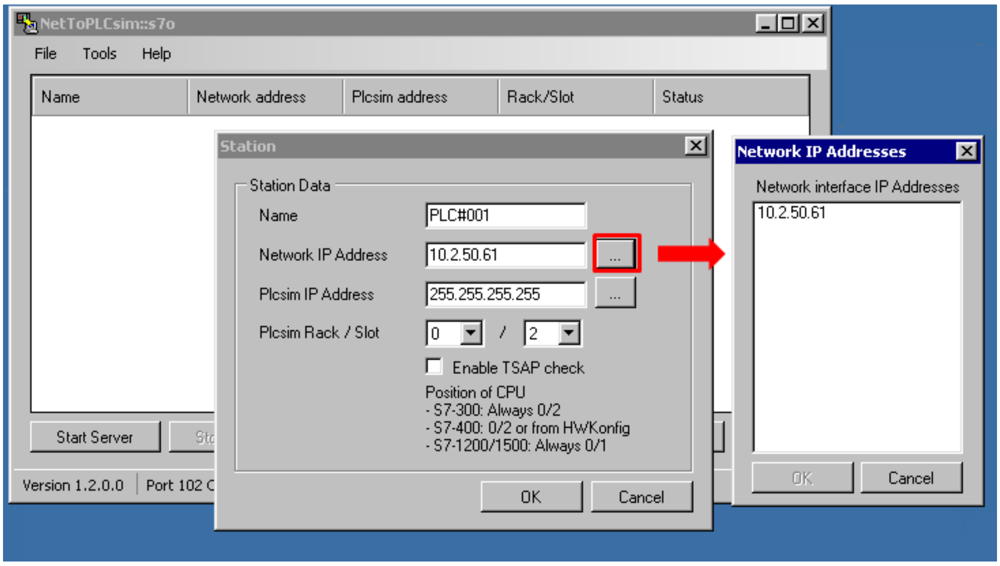


Figure A.6: Assign Network IP address

In "PLCSIM IP Address" field, insert the IP address of PLC which is running in PLCSIM. The IP address can be found in PLCSIM status bar. Click "..." icon to check the available PLC address.

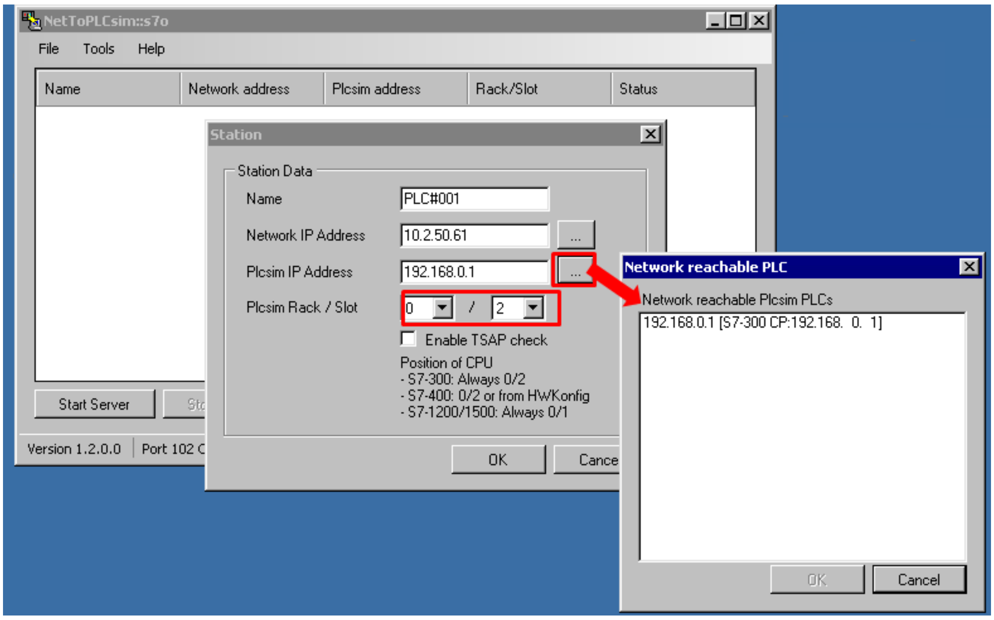


Figure A.7: Assign PLCSIM IP address & Rack / Slot

Select the Rack/Slot position of the CPU. The Rack/ Slot of CPU can be checked by clicking "Devices and configuration" in project tree in TIA Portal.
Click OK to finish the configuration and save the file.

A.1.1.3 Kepware configuration

Open "KepServerEX 6 Configuration" by administration authority. In this step, "OPC UA Configuration", "Reinitialize" and "Configuration" will be used. They can be found by clicking on "KepServerEX 6 Administration" icon in the system tray.

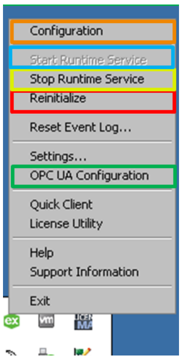


Figure A.8: The place of three icons

Click "OPC UA Configuration" (marked in green).
Define the endpoints. Click "Server Endpoints" tab. Click "Add" tab to insert a new endpoint. The endpoint is the port that Plant Simulation talks to.

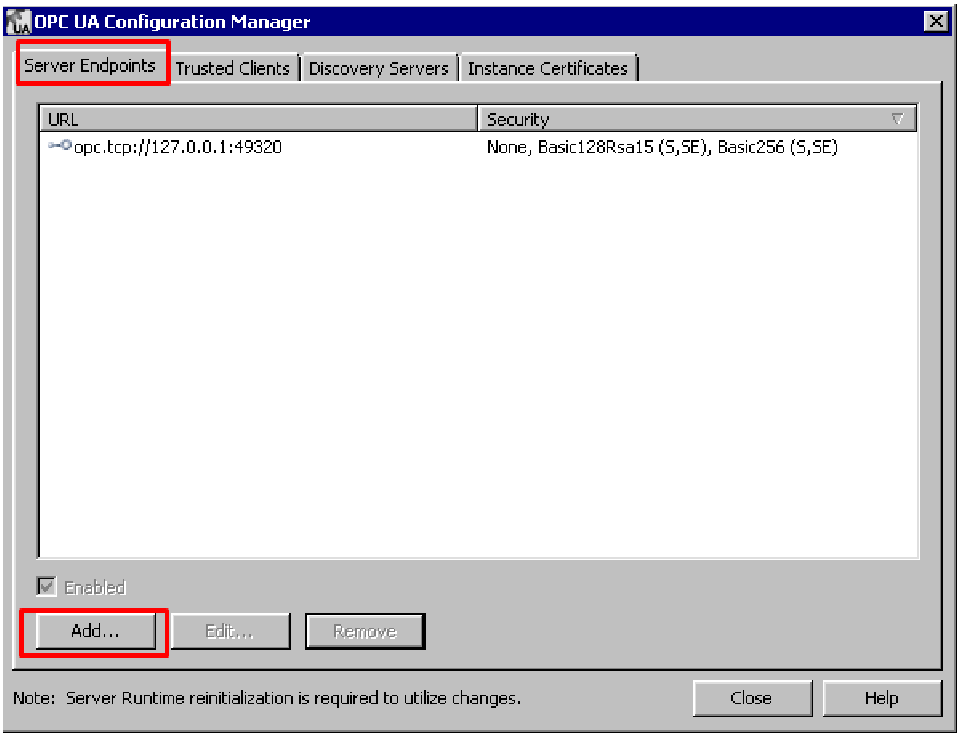


Figure A.9: Add a new server endpoint

Select Network Adapter and port number that will be used. Untick "Basic128Rsa15" and "Basic256" in security policies if no security certificates are going to be used. "None" is selected as default.

Click "OK" to finish the setting. If the endpoint address/ URL is in grey, select the address, and tick "Enabled". Click "Reinitialize" (marked in red in Figure A. 8) to refresh the program and active the changes.

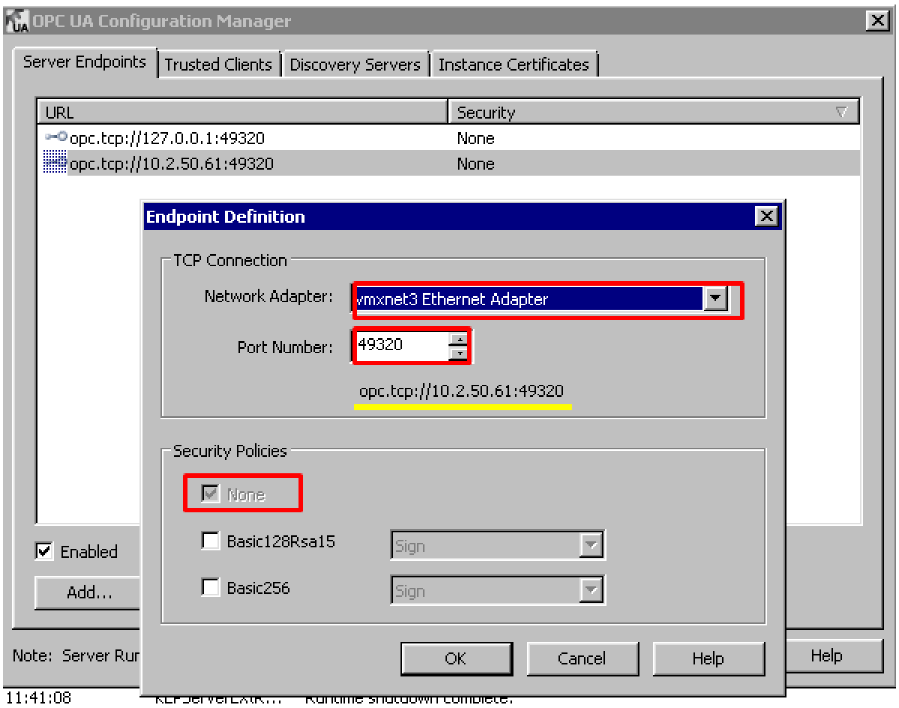


Figure A.10: Configure TCP connection & security policies

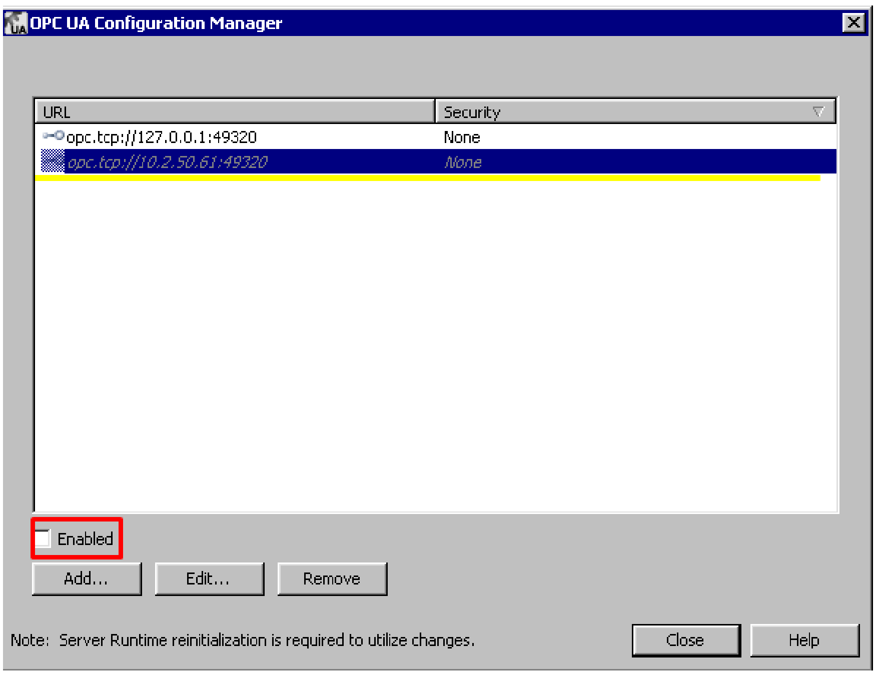


Figure A.11: Enable the endpoints

Click "Configuration" (marked in orange in Figure A.8) to open the configuration window. Click "add a new channel" to start the channel wizard.

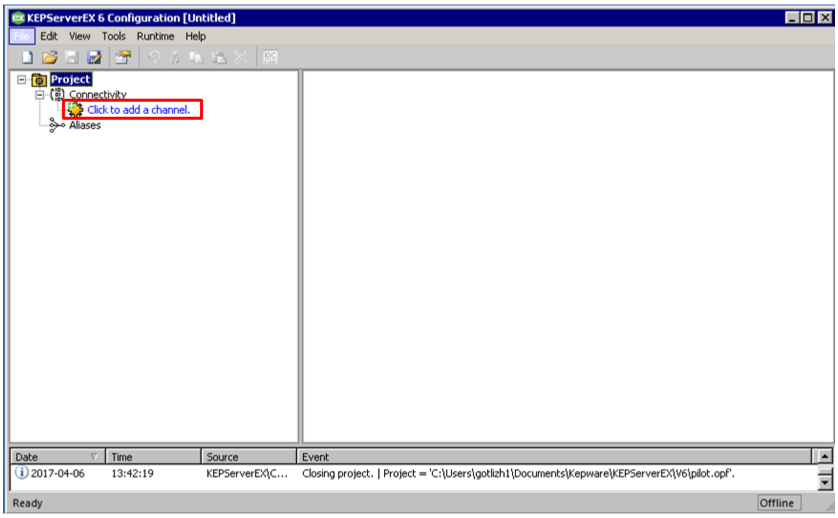


Figure A.12: Add a new channel

The channel wizard edits content including "Type of channel" "Name", "Network adapter" and others. In this case, the channel uses "Siemens TCP/IP Ethernet". All other settings are default. Under the channel tree, click "Add new device" to start the device wizard. In the wizard, the "Type of PLC", "Name", "Device's driver specific station", "Port number" and others. S7-300 is selected in "type of CPU" because it is used in the pilot model. The "Device's driver specific station" is IP address of the computer. "Port number" is 102. Other settings are default. By setting this, the Kepware now is available to talk to NetToPLCSIM.
In the device, tags are added. These tags are used to communicate between PLCSIM and Plant simulation. Click "add new tags". Define the tags name and address. The address tells where the tags are in PLCSIM. The address of PLC tags is classified with "I", "M" and "Q", which in Kepware OPC are "IB", "MB" and "QB" respectively.

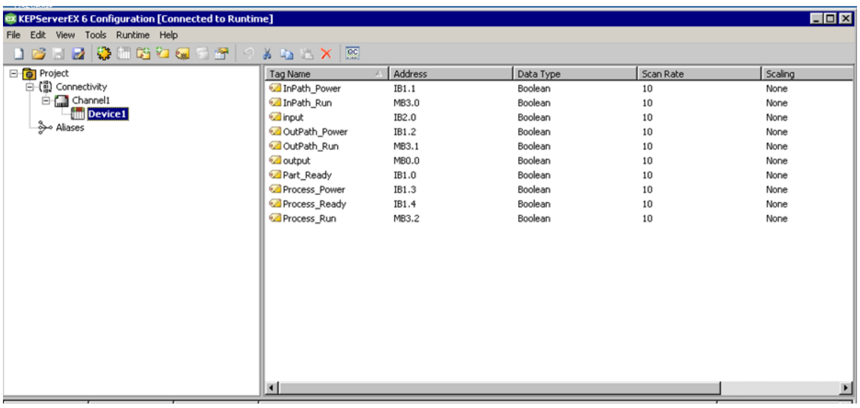


Figure A.13: Add tags

A.1.1.4 Security system

Port used to talk in OPC server with client shall be set in the firewall exception list. From "Start menu", type down command "firewall.cpl". Click "Advanced setting". The port number configured in last step shall be set as exception in inbound and outbound rule.
Port 49320 is used. This port shall be set in the exception list in the exception list. Authorization is required to change the firewall setting.

A.1.2 Client-side configuration

A.1.2.1 Plant Simulation

This step sets up the OPC Interface. OPC Interface is a bridge between Plant simulation and OPC server. On one hand, the OPC Interface contains signals sent/ received to / from PLC. It also builds the bridge with "Alias" used in Plant simulation. "Alias" is the name of variables used in simulation model. When value of "Alias" changes, for example, sensor triggered, value in OPC interface changes. When OPC Interface receives signal changes from OPC server, it can trigger the method that makes actuator reacts. Add a "OPCUAInterface" icon in the simulation project. The icon usually locates in "Information flow" in the "Toolbox". If it is not there, it can be found by clicking "Manage Class Library".

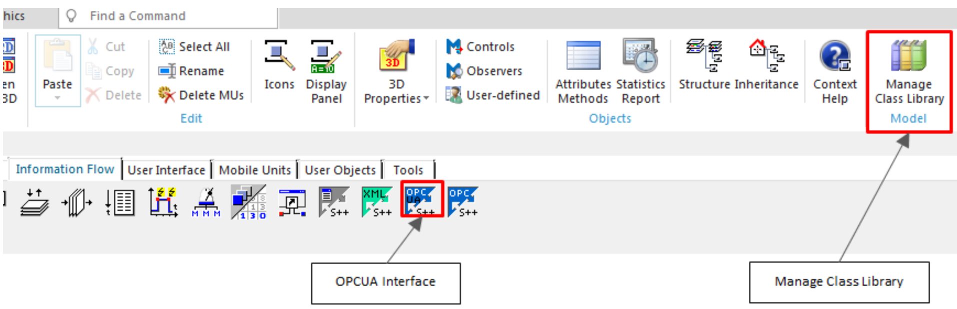


Figure A.14: Find "OPCUAInterface" icon

Double click the "OPCUAInterface" icon. Add the OPC address which is the endpoint defined in "OPC UA Configuration".

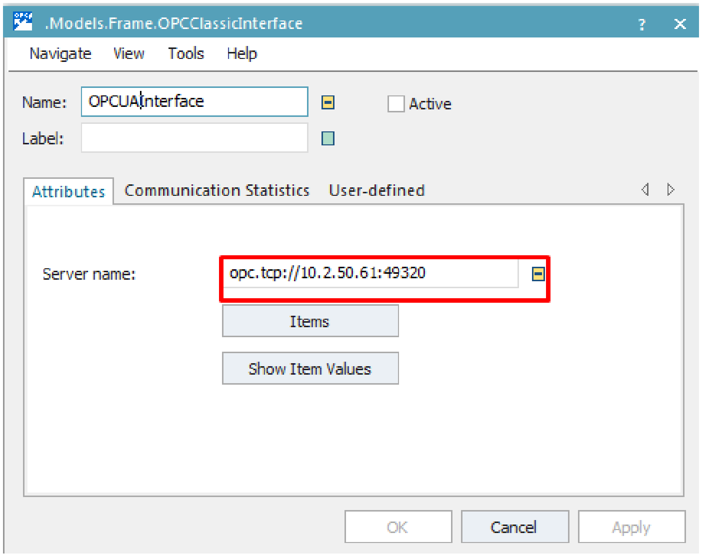


Figure A.15: Assign OPC server name

Click "Item" icon. Fill in the group name, namespace, read & write interval. The group name is the name of the group defining in Kepware. The group name of pilot model is "Channel1. Device1". Namespace is "2". Read interval is set as 5 ms. Write interval is 0 ms.

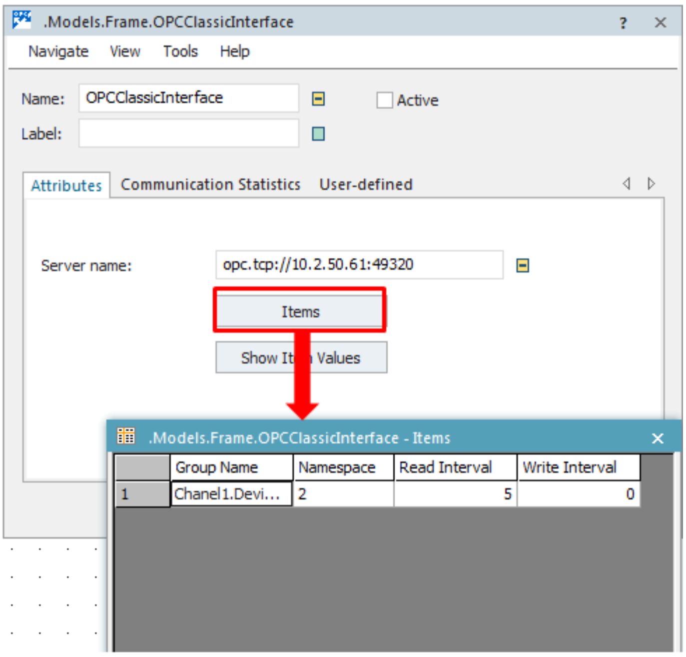


Figure A.16: Define the group item

Double click the target group to open the item list. "Identifier Type", "Identifier" and "data type" is available in OPC server.
Identified type is "string".

Identifier is the name defined in Kepware OPC server. For example, the PLC tag "InPath_Run", is written as "Channel1.Device1. InPath_Run".
"Data type" is Boolean.
"Changed-value control" refers to the Methods triggered by change of signal from virtual PLC. For example, Method "InPathrun" controls when the "InPath_Run" changes, then changes the status of the conveyor.

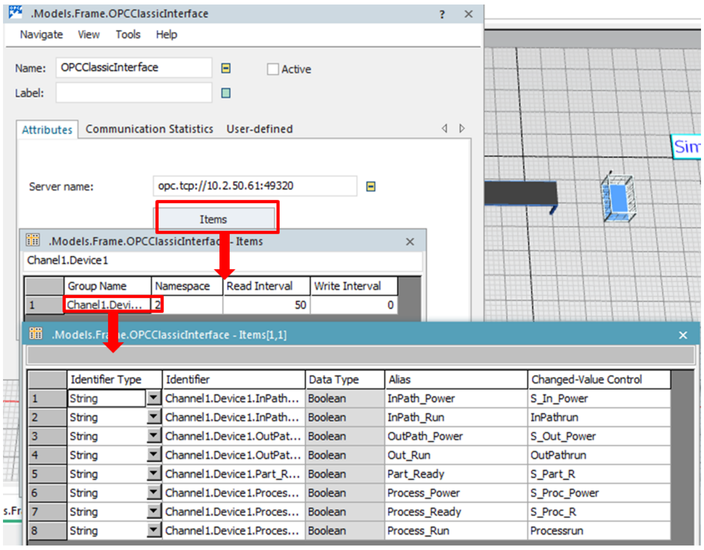


Figure A.17: Define items property

A.1.2.2 Set communication between Plant simulation and OPC Interface

OPC Interface receives sensor value changes from simulation model. Alias "Part_Ready" is taken as an example. The sensor sends the value to the OPC Interface. It is achieved by code "setItemvalue ("Groupname|Alias", Boolean)".

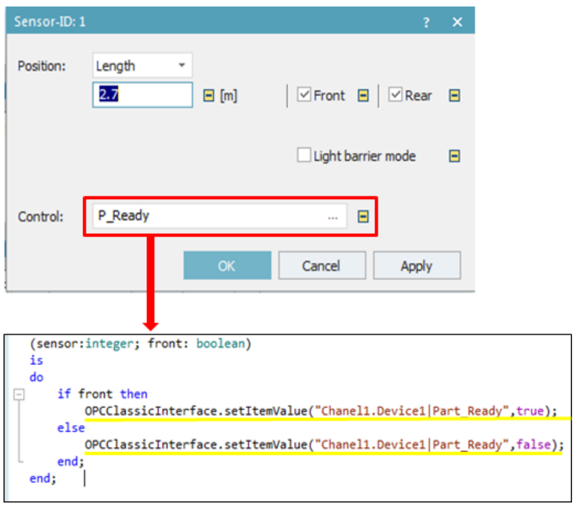


Figure A.18: Code embedded in the sensor "Part_Ready"

OPC Interface also reacts with the value changes. This is achieved by "Changed-value control". When the value of ROL changes, the method in "Changed-value control" is triggered to change the states of actuator. "InPath_Run" is taken as example. When the ROL changes from "false" to "true", method "InPathrun" is triggered.

```
(Alias:string; Value:boolean)
is
do
    if Value then
        InPath.speed:=Table_Conveyor["Speed","InPath"];
        Checkbox3.value:=true;
    else
        InPath.speed:=0;
        Checkbox3.value:=false;
    end;
end;
```

Figure A.19: Code of "Changed-value control"

A.1.3 Security system

Software used to talk in OPC server with client shall be set in the firewall exception list. From "Start menu", type down command "firewall.cpl". Click "Advanced setting", Plant simulation and OPCEnum are set as exceptions in inbound and outbound rule.

A.1.4 Use third party tool to confirm communication

A third-party tool, UaExpert is used. This software can detect and browse available OPC server. Open UAexpert. Click "Add server" icon in the task bar. Double click "Custom discovery" and add a server.

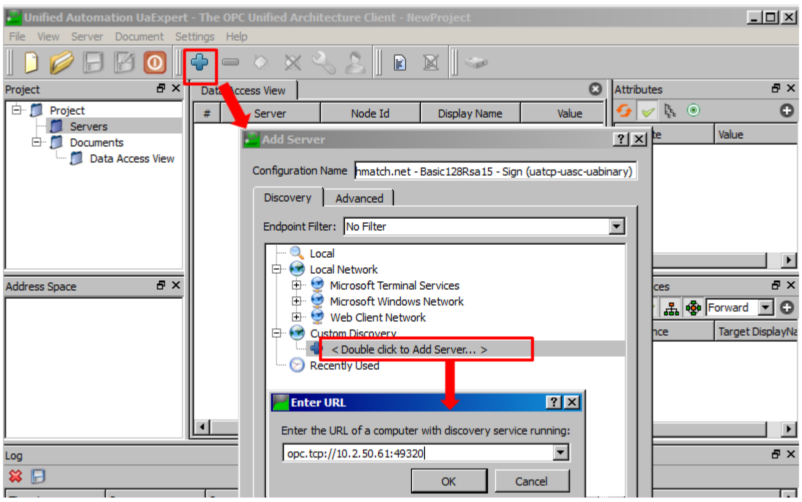


Figure A.20: Add a server in UAExpert

Enter the address of the remote OPC server that will be connected. Click OK. The available OPC servers are visible. Choose the OPC server. Choose the connection security type. Since "None" is ticked in security policies of OPC UA, choose "None".

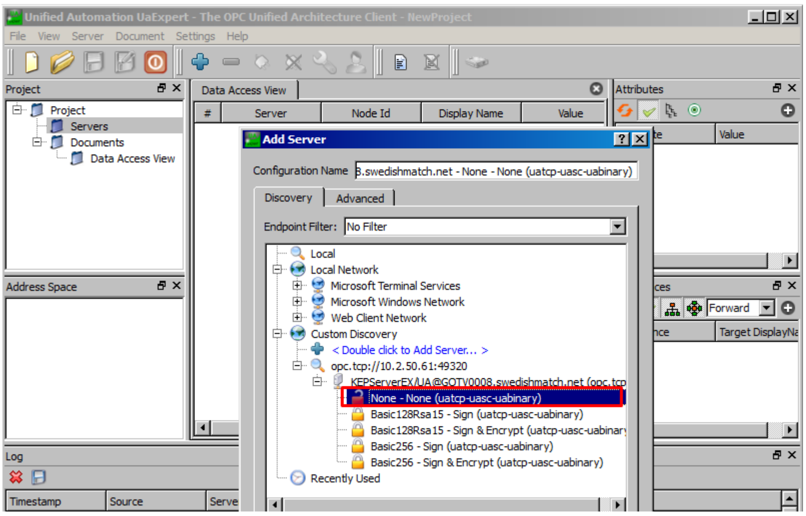


Figure A.21: Select the target server

In "Address space", check whether tag values can be read and, and whether it is "Good" quality or not.

A.2 Communication verification

The general procedures in verification is shown in Figure 22. After finishing the first section configuration, the user gets files from PLCSIM, NetToPLCSIM, and Plant simulation. In this section, the configuration and files are going to be tested. The verification phase also divides into server side and client side.

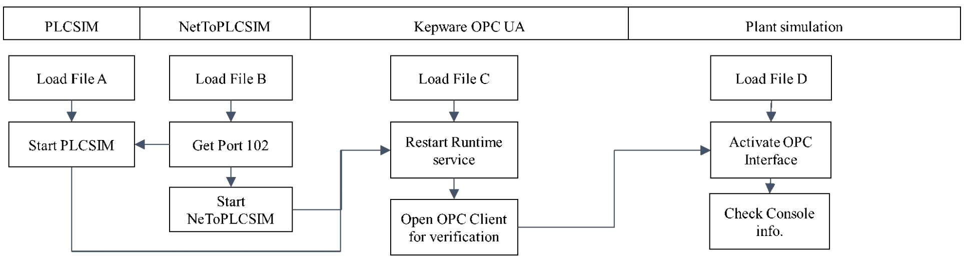


Figure A.22: Verification procedure

A.2.1 Server side

Start NetToPLCSIM by administration authority. Click "Tools" in the toolbar and select "Get Port 102". A successful notification will pump up if the port is available. The bottom shows the Port status. Load the saved configuration file by clicking "File->Open". Click "Start Server".

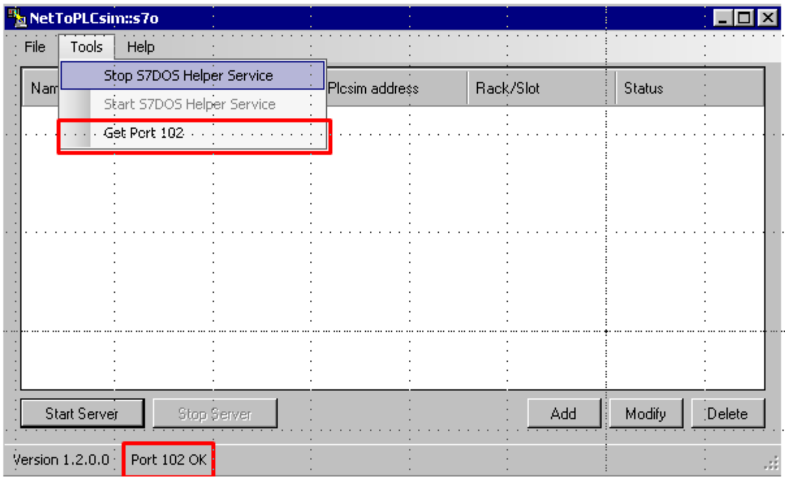


Figure A.23: Enable Port 102

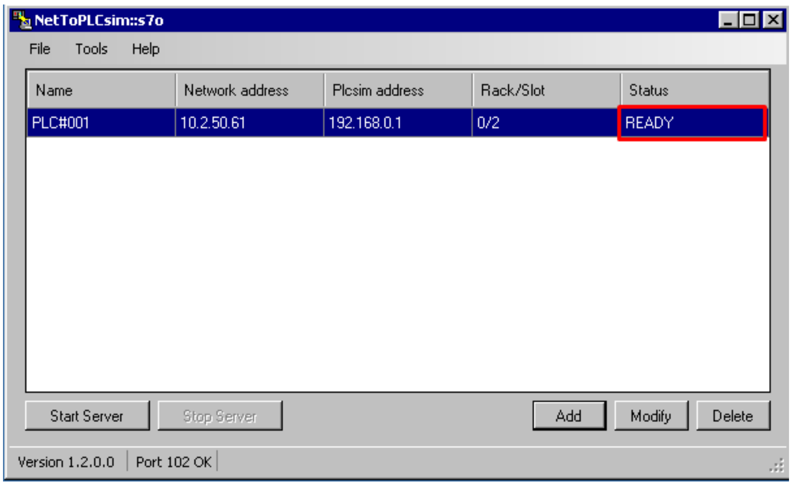


Figure A.24: Load the configuration file

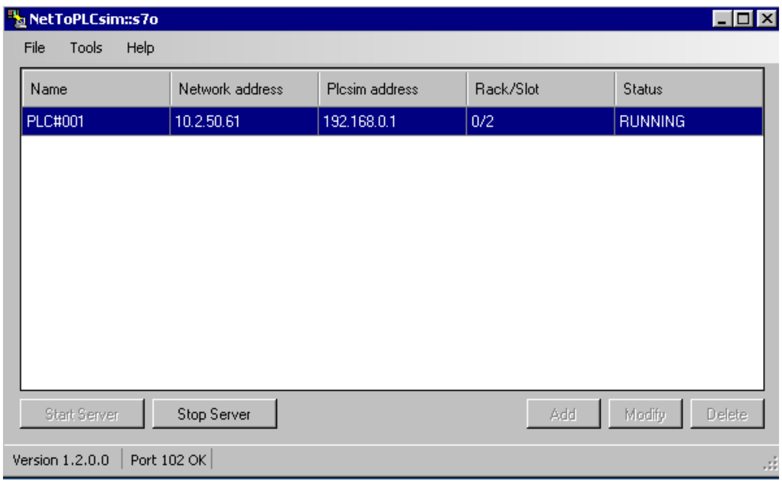


Figure A.25: Start server

Open the PLC simulation file. Click "Start" to start PLCSIM. In the PLC simulation file, select interface "TIP/ IP" (If the software is S7 PLCSIM). If the software is S7 PLCSIM V13, there is no need to do this.

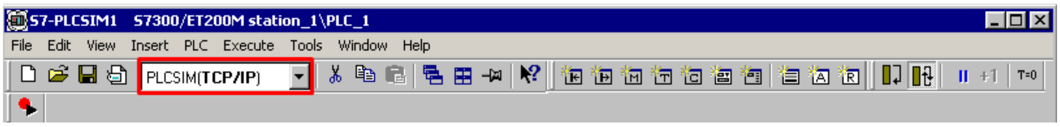


Figure A.26: PC/PG Interface in S7 PLCSIM

Right click on "KepServerEX 6 Administration" icon in the System Tray and select "Configuration". Load the saved project.
Right click on "KepServerEX 6 Administration" icon in the System Tray, select "Stop runtime service" (marker in yellow in Figure A.8) and select "Start runtime service" (marker in blue in Figure A.8). Select the targeting device and click "Quick Client" icon. A window called "OPC quick client" pumps up.

Figure A.27: "Quick Client" icon

Select the device. If the quality of items is good, the connection is successful in the server computer.

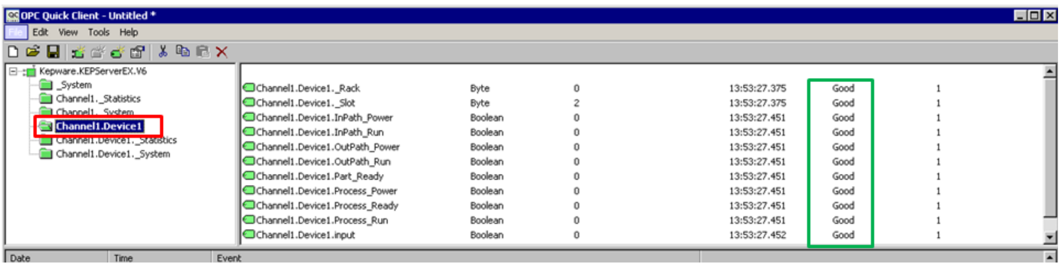


Figure A.28: Use OPC client to verify the connection

A.2.2 Client side

Open the DES model file. Double click the "OPCUAInterface" module. Tick "Active" and click "Apply". If there is a green dot at the left head of module icon, the connection is successful. The connection information is shown in the console screen. If the connection fails, the problem could be wrong server name, wrong namespace, wrong item (for example, identifier type, identifier, data type), unacceptable read/ write interval.

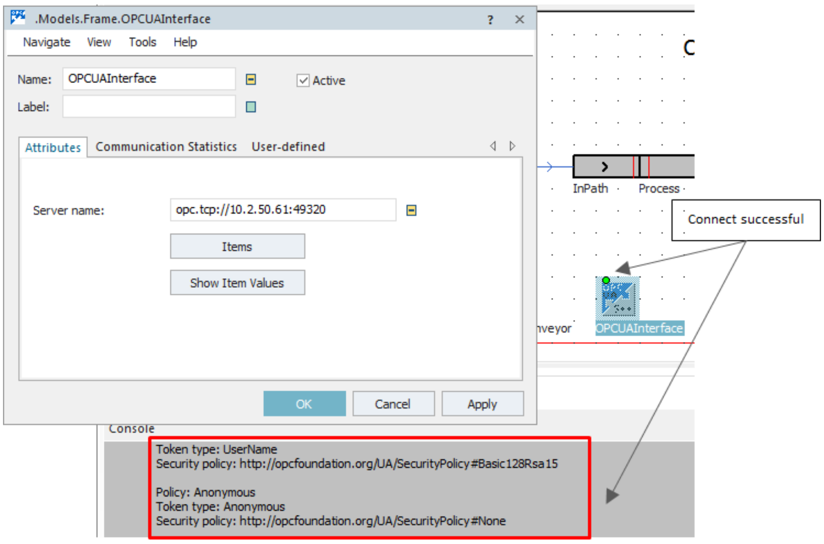


Figure A.29: The connection is successful

After verification of the connection, the establishment of communication between PLCSIM with Plant simulation is finished. Now, it is ready to run the simulation model with real time PLC signal. In Plant simulation, open the "Event Controller", select "real time" and write scale factor "1". This is because PLCSIM cannot scale the speed of as Plant simulation. Real time simulation synchronizes time in Plant simulation and PLCSIM.

B

Case study

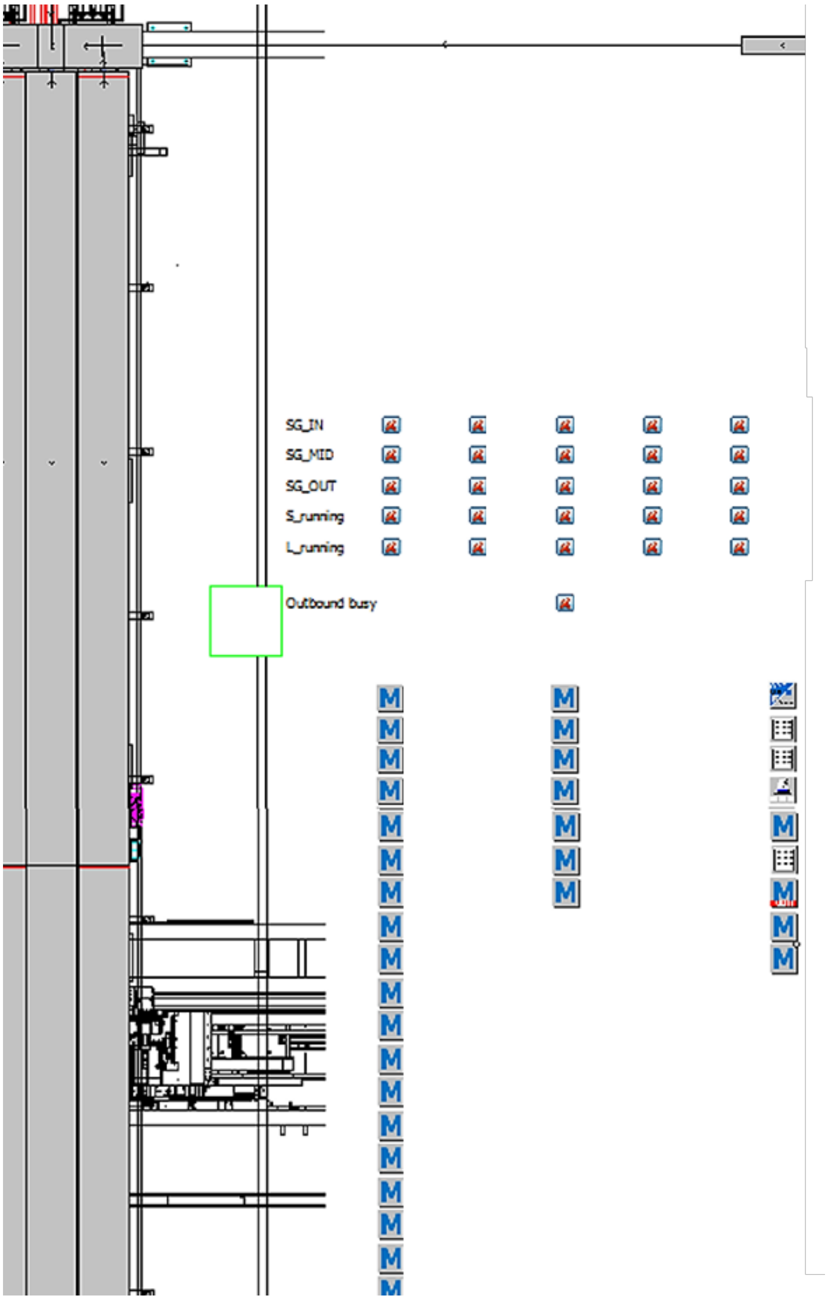


Figure B.1: PLC signal integrated simulation mode (part)

Table B.1: Checklist for separate PLC-logic and PLC integrated DES-model

No.	Condition	Input	Result	Output	Pass?
1	The product moves from the inbound to short buffer conveyor (is not 15),short buffer runs	SG_in=1	The counter increases by 1 and compare to 15, long buffer stops	Counter_short+1 long_buffer_run = 0	Pass
2	When there are 15 products on the short buffer conveyor and long buffer conveyor start to run	buffer1_ready = 1 short_buffer_run = 1	The long buffer conveyor runs	long_buffer_run = 1	Pass
3	When the products tends to move out, it touches the SG_mid	SG_mid = 1	The counter for long buffer increases. The counter for short buffer decreases	Counter_long+1 Counter_short-1	Pass
4	When the first product moves from the short buffer conveyor to the long buffer	SG_mid=1	The counter for long buffer increases. The counter for short buffer decreases	Counter_long+1 Counter_short-1	Pass
5	When 15 products move out	buffer1_ready = 0 short_buffer_run = 0	The long conveyor stops. The counter long is equal to one batch	long_buffer_run = 0 Counter_long = 15	Pass
6	When 30 products on the long buffer	buffer1_ready=0 short_buffer_run =0 buffer2_ready=1	The long buffer conveyor moves to SG_end	long_buffer_run =1	Pass
7	When the outbound is free, the batch touches the SG_end (the first in the queue)	SG_end=1 buffer2_ ready=1 Out_occupied =0 ready_transport_ buffer =1 FIQ = the first in queue	The counter of long buffer conveyor minus 1. Long buffer conveyor is ready to go	Counter_long-1 ok_transport_ buffer=1 long_buffer_run =1	Pass
8	When the outbound is free, the batch touches the SG_end (not the first in the queue)	SG_end=1 buffer2_ready =1 Out_occupied=0 ready_transport_ buffer=1 FIQ=the first in queue	The counter of long buffer conveyor minus 1. Long buffer conveyor is ready to go.	Counter_long-1 ok_transport_ buffer =1 long_buffer_run =1	Pass
9	When the outbound is not free the batch touches the SG_end (the next in the queue)	SG_end=1 buffer2_ready =1 Out_occupied=0 ready_transport_ buffer=1 FIQ/=the first in queue	The counter of long buffer conveyor minus 1. Long buffer conveyor is ready to go	Counter_long-1 ok_transport_ buffer=0 long_buffer_ run=0	Pass
10	When the batch is moving and touching the SG_end	SG_end=1 buffer2_ready=1 Out_occupied =1 ready_transport_ buffer=1	The long buffer waits	Counter_long-1 ok_transport_ buffer=0 long_buffer_ run=0	Pass
11	When the last product of this batch just leaves the long conveyor	SG_end=1 buffer2_ready=1 Out_occupied=1 ready_transport_ buffer=1 FIQ=the first in queue	The long buffer conveyor shall run	Counter_long-1 ok_transport_ buffer=1	Pass
12	When the last product of this batch just leaves the long conveyor	SG_end=0 Counter_long=0 Out_occupied=1 FIQ=the second in queue		buffer2_ready =0 ready_transport_ buffer=0 long_buffer_ run=0	Pass
13	When the first in queue empties	FIQ=the second in queue	The second in queue is ready to move out	buffer2_ready=1 ready_transport_ buffer=1 long_buffer_running =1	Pass
14	When the last product leaves the outbound	Outboundreset=1 Outbound=1	The long buffer stops	buffer_ready=0 long_run=0	Pass
15	All lines are empty	All -> 0	The conveyor stop	All -> 0	Pass

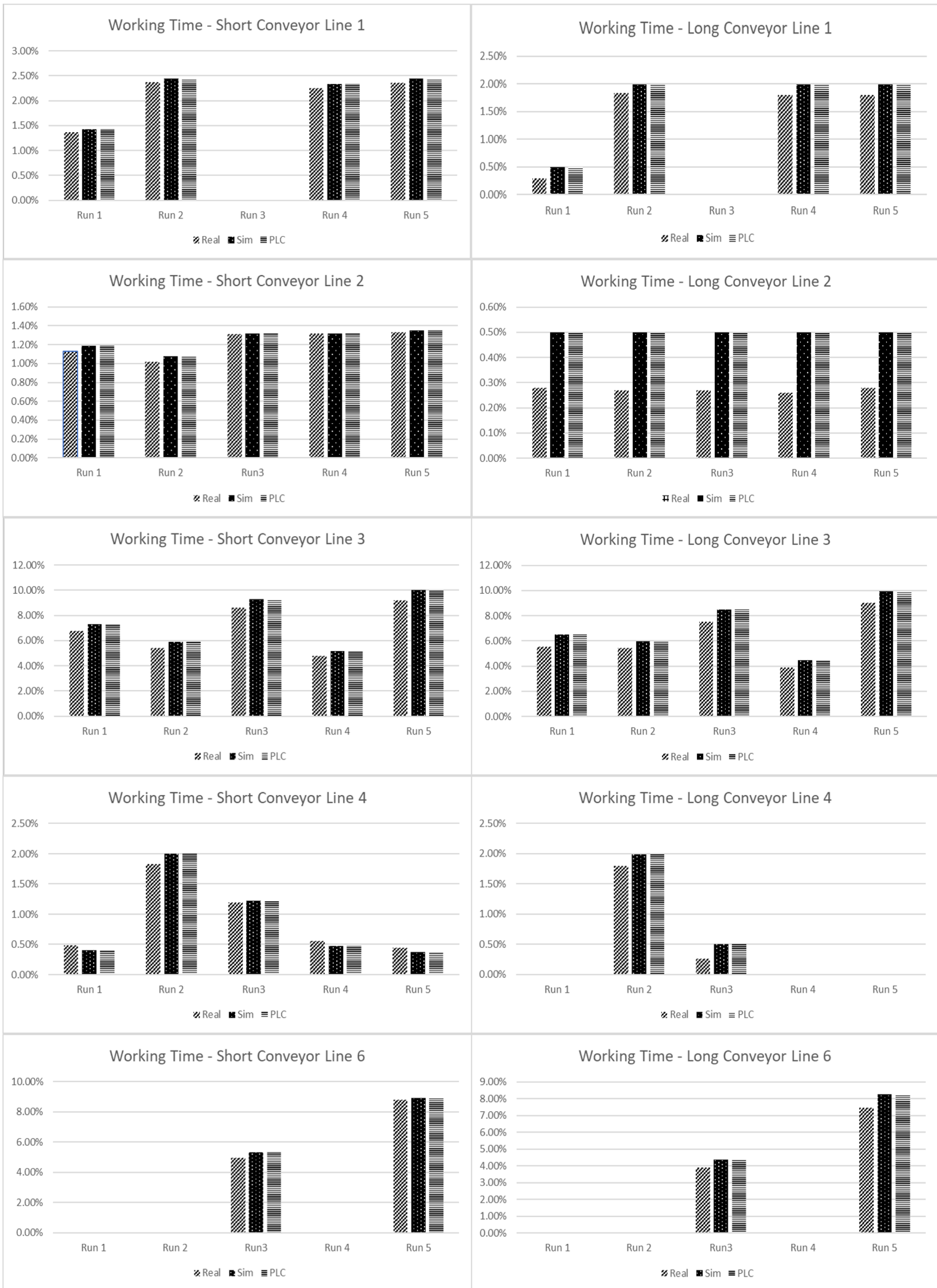


Figure B.2: Utilization result of case study