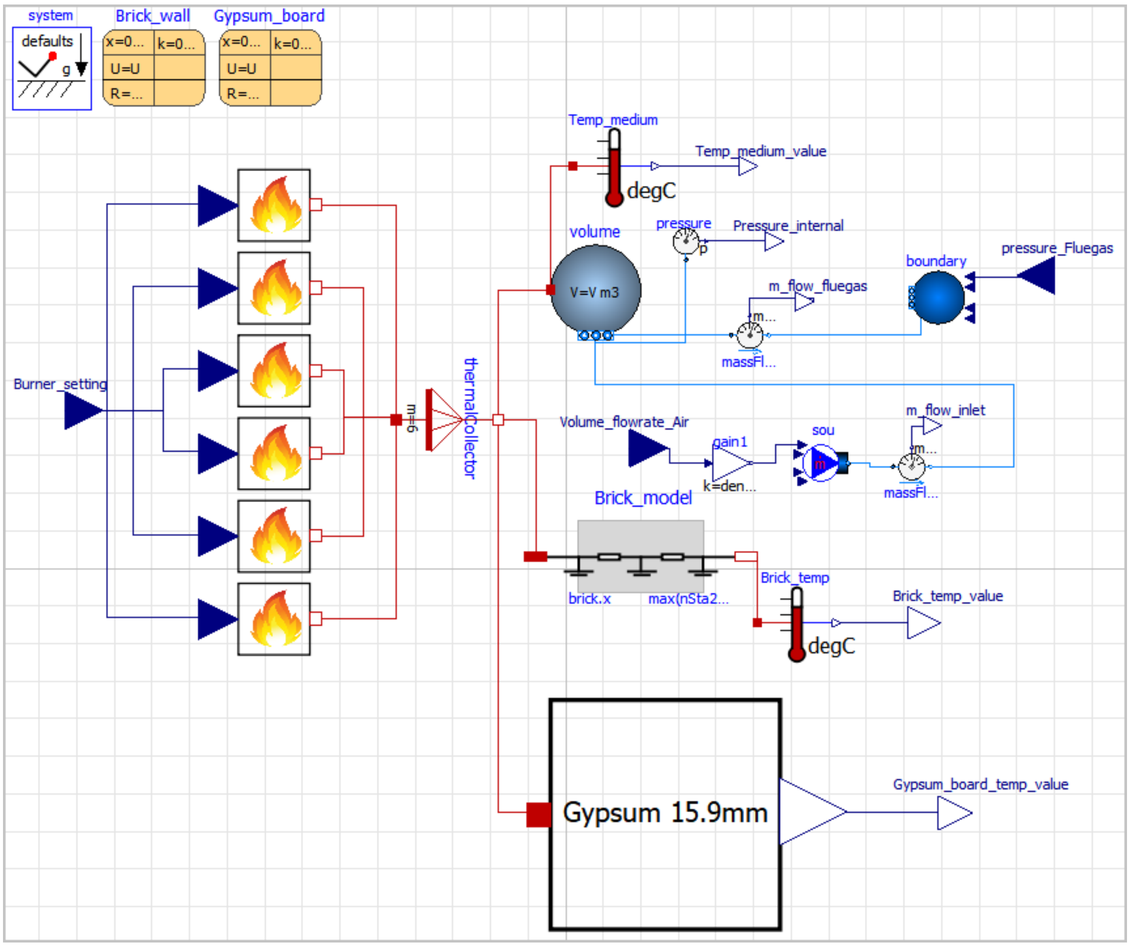
CHALMERS
UNIVERSITY OF TECHNOLOGY



Creating and simulating a digital twin of a fire resistance test furnace

Master's thesis in Systems, control and mechatronics (MPSYS)

FELIX GIMBRINGER

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Creating and simulating a digital twin of a fire resistance test furnace

FELIX GIMBRINGER

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Creating and simulating a digital twin of a fire resistance test furnace
FELIX GIMBRINGER

Supervisor: Anton Albo, Department of Electrical Engineering
Examiner: Petter Falkman, Department of Electrical Engineering
Contact person at AFRY: Oskar Redlund, Division Process Industries
Master's Thesis 2021
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of a modelled system of a test furnace developed for this thesis using OpenModelica

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Creating and simulating a digital twin of a fire resistance test furnace
FELIX GIMBRINGER
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Today, RISE - Sweden's research institute and innovation partner, owns two combustion furnaces where they test different products' fire resistance. Since there are only two of its kind in Sweden, a company would have to wait a long time before a time slot is available. Also, if a small change of the product is made it would have to go through testing once again, leading to a protracted process. A more time-efficient way of dealing with the testing is to build a digital twin of the real furnace.

In this thesis, a model of the physical furnace is created and evaluated in two different setups. The first one is a noninteractive setup in OpenModelica being controlled by a PID-controller. The second one is an interactive, real-time, simulation of an FMU in SIMIT being controlled using an HMI which is communicating with the model and control logic on a PLC through an OPC UA server.

The first setup is put through several scenarios testing the implemented control system along with the fire resistance of a simplified test object. The test object is a 15.9 mm thick gypsum board reinforced with glass fibers. The model is compared against the physical furnace in terms of how the internal temperature develops over time according to the ISO-834 standard. The second setup could not be simulated due to issues with the PLC not completing its initialization process. However, the system setup is designed and evaluated to be fully prepared and compatible for the real PLC-program if this issue would be resolved.

The results from the simulations in OpenModelica show that the internal temperature can be regulated according to the ISO-834 standard. The test object shows interaction with the provided heat flow, but more complex models will have to be developed to with certainty state their ability to withstand being put through a fire scenario. The developed model shows similar characteristics as the physical system when a simulation is compared to real data provided from a physical fire resistance test conducted at RISE's facility. To solve the issues within the PLC, two suggestions are presented. Either to scale down the code to a state where systems not being simulated by the model are present or to manually go through the code and set static values on the required signals.

Keywords: Digital twin, FMI, FMU, HMI, ISO, OpenModelica, OPC UA, PLC

Acknowledgements

First of all, I would like to thank the company of AFRY and their employees for accepting my request of writing my master's thesis with them. I am particularly grateful to my two supervisors at AFRY, Magnus Lundkvist and Oskar Redlund, for providing expertise within the areas studied within the thesis. I would also like to thank the company RISE for wanting to participate in the process. Without their shown interest, this thesis would have never been brought to life.

Along with the companies, a special thanks is directed to my supervisor at Chalmers, Anton Albo. Anton has guided me in tough times and helped me plan future work when problems have arisen. Anton has shown his proficiency when it comes to guiding a thesis from start to finish, showing great knowledge within the areas of virtual commissioning, PLC-programming and server communication to name a few.

Finally, I would like to send thanks to my examiner Petter Falkman, at the Department of Electrical Engineering.

Felix Gimbringer, Gothenburg, May 2021

Glossary

Digital twin a virtual copy of a physical system.

Industry 4.0 Current ongoing industrial revolution.

RISE Sweden's research institute and innovation partner.

Acronyms

CXO Chief eXperience Officer.

DE Differential Equation.
DEM Discrete Element Method.

FMI Functional Mock-up Interface.
FMU Functional Mock-up Unit.

I/O Input/Output.
ISO International Organization for Standardization.

LPG Liquefied Petroleum Gas.

OPC UA Open Platform Communications Unified Architecture.

PLC Programmable Logic Controller.

VC Virtual Commissioning.

Contents

1

Introduction

During the course of the master program **Systems, control and mechatronics** various topics regarding technical systems, simulations and programming have been covered. The aim is to give future engineers a broad systems engineering base, suited to the engineering of complex, computer-controlled (embedded) products and systems[1]. This master thesis will be carried out in order to create a virtual system replicating a physical testing rig located at a testing facility in Sweden. This will meet the aim of the master program in the sense that it will address areas in complex systems, simulation and system control.

1.1 Background

Today, RISE - Sweden's research institute and innovation partner, owns two combustion furnaces where they test different products' fire resistance. Since there are only two of its kind in Sweden, a company would have to wait a long time before a time slot is available. Also, if a small change of the product is made it would have to go through testing once again, leading to a protracted process. A more time-efficient way of dealing with the testing is to build a digital twin of the real furnace. This twin would then simulate the fire and thus reduce both cost and wait time for the company. Additionally, if RISE wants to make any changes to the physical system, the digital twin could be used to carry out testing for the integrated control systems before creating new physical models.

1.1.1 Industry 4.0

One area where ongoing revolution and strive for improvements are always happening is in the industry. Since the first industrial revolution took place almost 300 years ago, technology has been pushed to its limits and thus excelled in what is possible to achieve. From the first revolution where steam-driven engines and utilization of mass production played a major role, today's revolution is driven by the internet and the digitization of industries as a whole [2]. Utilizing *Internet of Things* in production, converting factories into *Smart Factories* or making use of *Artificial Intelligence* are all buzzwords describing this ongoing revolution. These can collectively be put together under the name Industry 4.0. This concept describes the current chapter of the industrial revolution under one broad brand name. The possibilities brought by it can be seen as endless. Vachálek et al. say that possibilities never seen before are now emerging through the use of these new technologies, such as artificial intelligence, robotics, internet of things [3]. Furthermore, Vachálek et al. state that another technological concept brought to life by Industry 4.0 is the concept of the digital twin.

1.1.1.1 Digital twin

According to Armstrong, a digital twin can be seen as *"the virtual representation of a physical object or system across its life-cycle. It uses real-time data and other sources to enable learning, reasoning, and dynamically recalibrating for improved decision making"* [4]. This means that the digital twin should act as a plain copy of the physical system and be able to carry out the same tasks as its physical twin. Furthermore, Armstrong views the digital twin as an essential tool in an engineer's toolbox. This tool should not only be used to look into how a system is performing right now but also how it will perform in the future.

1.1.2 Current industrial situation, problem and need

Despite the vast possibilities brought to life by the implementation of Industry 4.0, it also brings almost as many uncertainties as opportunities [5]. Bourgois et al. mention that when transitioning into new technology, it is often seen that the company is affected on other planes than the technological. Furthermore, Bourgois et al. state that it is crucial to understand the connections between business and social needs; between financial outcomes and innovative strategies and between integrating existing technologies and creating completely new solutions. Deduced from the survey presented in [5], given to Chief eXperience Officer (CXO) across the whole world, it was found that only 14% of the CXO's was prepared for a transition into Industry 4.0. This means that the transition can not only be seen as a technological one but also a social and managerial one. Today, companies are in the early stages of making use of the possibilities brought by Industry 4.0 and when CXO's across the world are ready, the technology is waiting for them [5].

1.1.3 Relevant or similar research

For the project covered in this thesis, it is essential to study different areas to build up a knowledge base before constructing the virtual system. One vital area to explore is the the area of virtual preparation and commissioning. This means studying the use of simulations to test or design various physical systems before making changes to the physical system. This thesis concern the creation of a digital twin using a virtual preparation approach based on the Virtual Commissioning (VC) methodology, thus not focusing on the practical procedure of implementation of a physical system. However, if future changes should be made to the physical system, then the digital twin could be used as a platform to evaluate the planned modifications.

One aspect taking into account when creating a virtual system is never to disregard from the physical systems' safety limitations regulated by ISO-standards. Another is how the verification procedure concludes that the virtual system is a copy of the real physical system. Albo and Falkman presents in [6] how to tackle a VC project in a structured manner. Albo and Falkman presents a framework for categorizing the different levels of detail. An overview of these levels can be seen in Figure 1.1. This framework is created to be able to develop the final product using strategic decisions in a well-planned order will be of greatest interest in this project.

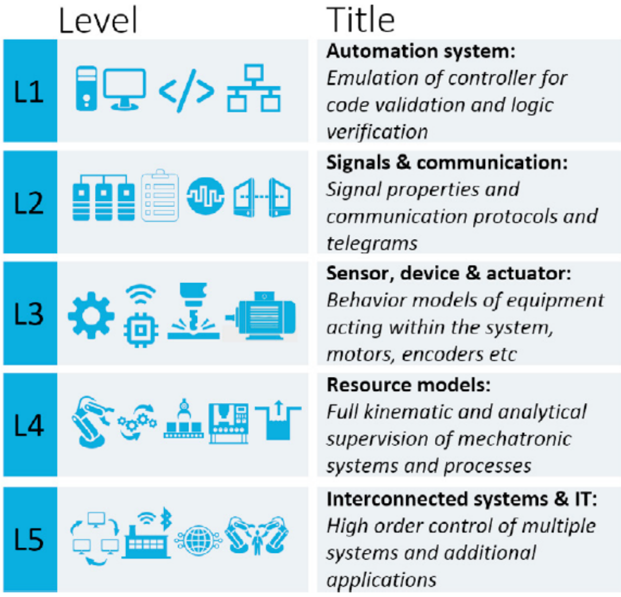


Figure 1.1: "A framework hierarchy for categorizing different levels of details for Virtual Commissioning implementation". Adapted from [6]

Another source of information regarding a standardized approach when setting up a virtual replica of a physical system can be seen in [7]. Here Westbrink and Schwung discuss an approach for VC with respect to a standardized method called the Discrete Element Method (DEM). This approach is contrary to Albo and Falkman's more focused on obtaining a detailed and accurate simulation instead of taking a step back and viewing the whole project as an optimization problem. Although Westbrink and Schwung's take on the simulation can be of interest when looking into the creation of the simplified object, which to be put to the test in the simulation furnace.

Materials put to test in the furnace are e.g. beams, doors or various board materials. The materials are studied until failing to withstand fire, either by breaking or falling off their mountings. According to Just et. al, gypsum boards with a reinforced glass fiber structure can fall off the wall when the entire board reaches a temperature of 700 $°C$. When this temperature has been reached, the glass fiber structure melts and the board itself will fail. Their study conducts that when at least 1% of the material on the heating side of the board has fallen off, the board can no longer withstand fire [8]. In addition to Just et. al, another study conducted by Yoshitani et. al. shows non reinforced gypsum boards put to test [9]. From their experiments Yoshitani et. al. concludes that regular gypsum boards of a thickness of 15.9 mm fails at around $500°C$. Thermal properties of gypsum boards of various types is measured in a study made by Manzello et. al. [10]. The values of these properties are used when creating the model of the test object, presented in section 3.2.2.

1.1.4 Tools for digitalization of hardware components

To be able to create a digital twin, various hardware components have to be converted into software representations. For instance, the physical Programmable Logic Controller (PLC) must be simulated in some software environment. Two of the major companies being used in the industry are Siemens and Schneider Electrics, both of which provide software used for development and simulation of PLC-code. *TIA-portal* from Siemens and *Control Expert* from Schneider. Since the physical PLC runs on a hardware component, Modicon m580 provided by Schneider Electric, is used as a simulation platform for the PLC.

Modelling of the furnace can be done using OpenModelica which is a simulation platform based on the Modelica language. OpenModelica enables modelling of the furnace including realistic dynamic with physical properties and elements. The developed model can be constructed as a Functional Mock-up Unit (FMU) and exported as a black box. The FMU is used as a component when setting up the plant digitally. The communication between the soft-PLC and the software dealing with the plant layout is established using an Open Platform Communications Unified Architecture (OPC UA) server.

1.2 Method

The development of a digital twin from a physical system includes many tasks. Data collection from RISE regarding the physical plant is essential to form a deeper understanding of the system itself. The data contains information about the physical system and information about the different sensors being present during physical testing. Along with the functionality of the sensors, information about how the programmable logic controllers (PLC) operates is also studied.

The digital twin is created utilizing software supplied by AFRY and RISE, such as:

- SIMIT - handles signals and simulates functional mock-up units (FMU) based on the physical system.
- SCHNEIDER - handles PLC logic using a soft-PLC application.
- OPC UA server - signal distribution between system and software.
- OpenModelica - modelling and evaluation of simulated case.

The main workload is focused on understanding the collected data and creating functional mock-up units (FMU) in OpenModelica for the various components integrated into the physical system. The concept of utilizing modularity between the components present in the model is key in this thesis. All of its components are built in a manner that makes them easily connected, and if more functionality is demanded from the model, it is rather uncomplicated to extend its current state.

The development of a digital twin is not done by developing the different components after one another, starting with soft-PLC, followed by the simulation model and finally connecting them together. It is better described as an iterative process where, at first, a setup with a small amount of functionality is built and tested. More and more functionality is introduced into the system until the complete system is created.

Two simulation environments is set up. One noninteractive in Openmodelica designed to test the model validity, and to study the first research question determining if it is possible to evaluate the fire resistance of a simplified object of a physical product. To be able to test the model in OpenModelica, a PID-regulator is developed. The second environment is an interactive one in SIMIT showing the final concept aimed towards industry usage.

The digital twin is compared to the physical system in terms regarding similarities and differences when performing a fire resistance test following the ISO-834 guidelines. The results from the comparison is studied according to the second research question, if and how the digital twin could be beneficial for a company or the industry in general.

1.3 Aim

The scientific take on the master thesis is focused on investigating if the concept of a digital twin working side by side with a physical system can be beneficial for the industry. Important areas of study are time savings in terms of testing control code since simulation time will be significantly decreased. Along with that, it is also of interest to investigate if a simplified model of a physical product can be evaluated using a digital twin approach. The creation of the virtual system will be handed to AFRY and RISE for future development. In other words, the aim is to create a good foundation to make sure it is possible to continue in-house research and development at AFRY and RISE.

1.4 Objectives

To carry out the project with a strategic approach in mind, subtasks, research questions and demarcations are stated and acts as guidelines during the course of the project.

1.4.1 Research questions

During the project, information is be gathered, tests are created and conducted to answer these research questions:

- Can the fire resistance of a simplified model of a physical product be evaluated using a digital twin approach?
- Can the developed simulation environment in the thesis be beneficial for the industry to either gain time savings in terms of reduced simulation time or improve the company's actual system?

1.4.2 Divide the problem into subtasks

Three main areas have been identified as subtasks.

- Creation of soft-PLC
- Creation of FMU-blocks representing the physical model of the furnace
- Communication between PLC and FMU using OPC UA

During the course of the project, these subtasks are broken down into even smaller tasks that are evaluated and documented as part of the final report.

1.5 Demarcations

An overview of the physical system can be seen in Figure 1.2. The red border represents which part of the system that will be excluded in the virtual model.

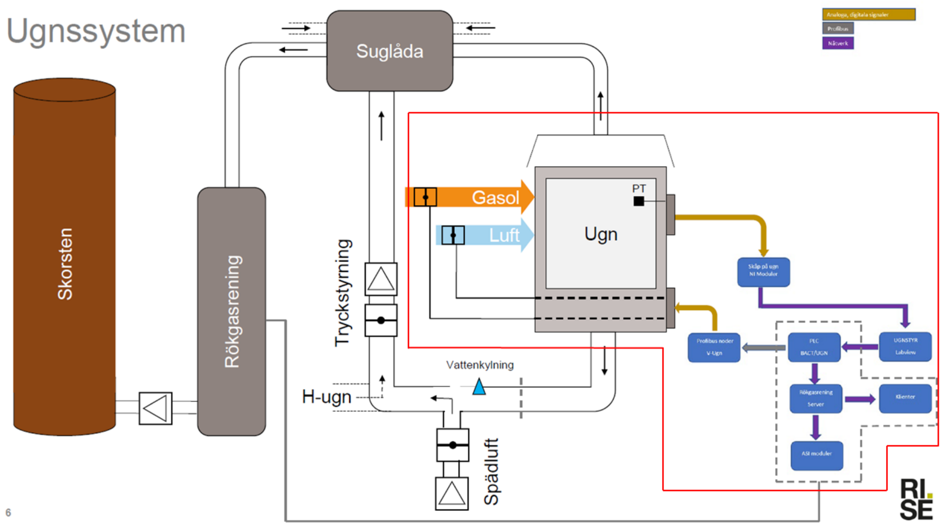


Figure 1.2: System border of the furnace model.

This means that the pressure inside of the furnace will be kept constant during the simulations and a constant volume flow of air is modeled into and out from the furnace.

1.6 Outline

The remaining part of the thesis is divided into five chapters. **Chapter 2** contains the theory behind the concepts being described in the thesis. The developed model and its components are presented in **Chapter 3**. The conducted simulations and their setups are shown in **Chapter 4** followed by their respective results in **Chapter 5**. Finally, a concluding chapter, **Chapter 6** ties the results from the simulations to the presented research questions.

2
Theory

The theory behind the different components in this project, both mathematically and purely informational, is presented in this chapter. The simulation tools such as OpenModelica and SIMIT are using advanced mathematical algebraic equations and are only briefly described. A deeper focus is placed on describing the various subsystems used in the project.

2.1 Modelling

To create complex models of physical products, various approaches can be taken. One approach is to model the behavior in state space and compute new state derivatives at each time step. The time step can be of varying size depending on the dynamics in the model. If the equations describe large dynamics in the system, then the time step will be decreased and if low dynamic is present, a larger step size can be taken.

2.1.1 Modelica language

According to Fritzon and Bunus, *"Modelica is a general equation-based object-oriented language for continuous and discrete-event modelling of physical systems for the purpose of efficient simulation"*. Modelica also supports multi-domain capabilities, letting the user create models of complex systems containing different physical domains, such as electrical, thermodynamic and mechanical in one model altogether. Since Modelica is an acausal modelling language, the reusability of classes is made possible considering that equations do not specify a certain data flow direction. These features make Modelica a very powerful modelling language [11].

2.1.2 OpenModelica

OpenModelica is open source and is developed to make use of the strengths of the Modelica language. The tool is described as *"a unique large-scale integrated open-source Modelica- and FMI-based modelling, simulation, optimization, model-based analysis and development environment"* [12]. In OpenModelica, users can make use of various built-in libraries that contain standard and complex components.

2.1.2.1 Solvers

Two types of solvers can be utilized when simulating in OpenModelica. Single-step solvers and multi-step solvers. The differences between the two are that single step solvers approximate the model behavior by information only given between time, t, and the time step, dt, while multi-step solvers approximate the model behavior at the end of the time step by taking into account information that occurred many steps before the new approximation. In a way, one can say that multi-step solvers depend on the prior approximations while single-step solvers reset and give a completely new approximation at each time step. This resetting yields a larger computational cost when compared to a multi-step solver [13]. However, multi-step solvers need a good initial approximation to stay away from initialization divergence.

To solve this, OpenModelica uses a multi-step ODE solver called dassl. It can handle a variable step size depending on if the size of the state derivatives [14].

2.2 FMI-standard

The Functional Mock-up Interface (FMI) is a free standard that defines an industry-standard way of packaging dynamic models with input, outputs and internal system equations [15]. When exporting a model using the FMI-standard a Functional Mock-up Unit (FMU) is created. This unit can be used in two different modes, Model-Exchange or Co-Simulation. When building models containing physical quantities, such as temperatures or heat flows, these quantities are generally solved using differential equations. Those differential equations are solved using a solver, either inside the FMU or inside the simulation tool into which the FMU is imported. When using the FMU as Model-Exchange, the FMU contains inputs, states and functions needed to compute the state derivatives. The simulation tool sets up the time steps and how to compute the state derivatives. However, when using the FMU as Co-Simulation, the solver is included in the FMU itself and thus the simulation tool only provides the FMU with inputs and a given time step and then reads the computed state derivatives. Figure 2.1 shows this graphically.

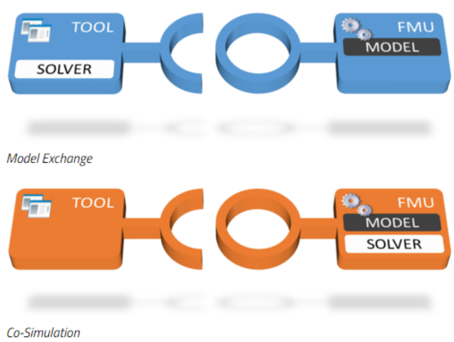


Figure 2.1: Model-Exchange and Co-Simulation in FMI 2.0. Adapted from [16]

In the second major release of the FMI-standard, FMI 2.0, the FMU may be both of type Model-Exchange and Co-Simulation. This means that all of the Model-

Exchange functions and a DE-solver are provided in the FMU and the user can later choose if the FMU should be used as Model-Exchange or Co-Simulation.

2.3 Fire resistance standard

The International Organization for Standardization (ISO) work towards making sure that most aspects of technology and manufacturing processes are met by independent and international standards [17]. The standard that handles tests and evaluation of fire resistance is ISO-834 [18]. This handles the way of testing fire resistance of various building materials such as wooden beams, brick walls or gypsum boards. The temperature inside the furnace should follow a behaviour according to,

$$T = 20 + 345 * \log(8 * t + 1) \tag{2.1}$$

where, t is time [min], T is furnace temperature [° C] at time t as plotted in Figure 2.2.

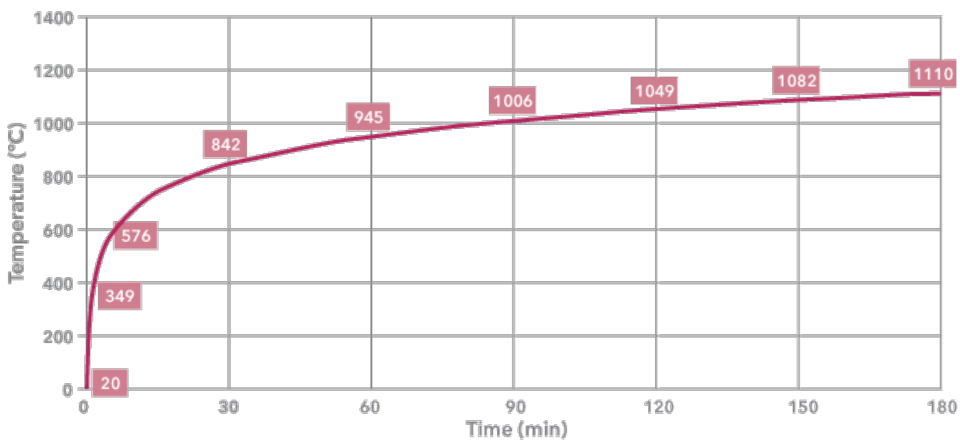


Figure 2.2: Cellulosic curve from iso-834, given by equation (2.1). Adapted from [19]

2.4 Process

The test facility is located at RISE in Borås and holds two different types of furnaces, one horizontal and one vertical. The vertical furnace is the one studied during the project. The quadratic shaped furnace has dimensions of $depth = 1.7$, $width = 3$, $height = 3$. Inside the furnace, three Liquefied Petroleum Gas (LPG) burners are mounted on each side facing the furnace opening. Each burner has a power capacity of 0.7 MW so when all six are being run on full throttle the vertical furnace produces a total of 4.2 MW of power.

2.4.1 Medium

For OpenModelica to be able to compute state derivatives of the various states in the model a medium must be declared. The medium has base properties such as

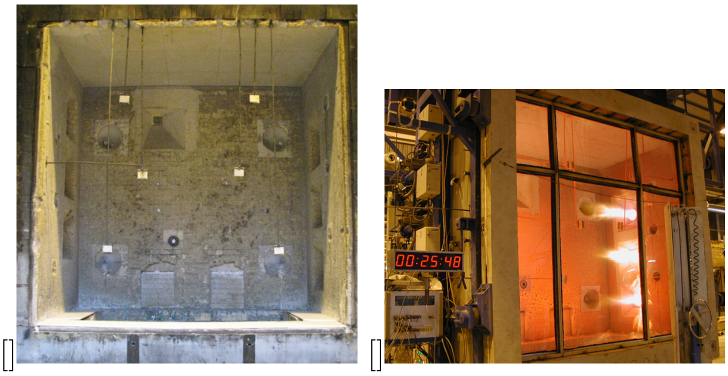


Figure 2.3: (a) Open furnace viewed from front (b) Mounted glass as test object

specific enthalpy and heat capacity declared to itself. The medium used in the model is *FlueGasSixComponents* [20] which is an extension of the medium *MixtureGasNasa*. This model uses data from a NASA report [21] to compute the thermodynamic state properties. The model is valid in the temperature interval of $-73°C < T_{valid} < 5726.85°C$. The medium itself can be viewed as a lumped homogeneous mass and thus the complete mass will be affected by the changes in its thermodynamic states as a whole.

2.4.2 Heat and mass flow rates

Heat transfer is assumed to be ideal with no thermal resistance from the medium. The heat flow rate is injected into the system through a heat port and state derivatives for the thermodynamic states such as temperature, pressure and density are computed. Figure 2.4 depicts the various flows in and out of the system border. The major source of heat flow rate is given from $\dot{Q}_{LPG}$ which describes the provided heat from the combustion of LPG. Also to be able to simulate a flow of air through the furnace, $\dot{m}_{Inlet}$ has a constant temperature of $20°C$. The heat flow rate connected to this mass flow is described by $\dot{Q}_{Inlet}$.

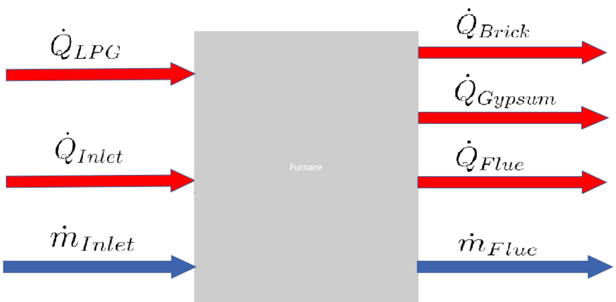


Figure 2.4: Heat and mass flow in and out of the furnace model

Three heat flow rates are present out from the furnace, $\dot{Q}_{Brick}$, $\dot{Q}_{Gypsum}$ and $\dot{Q}_{Flue}$. The first two describes heat dissipating through walls and the test object via conduction and the third one describes the heat flow rate carried by the mass flow of flue gases, $\dot{m}_{Flue}$, passing out of the furnace.

2.5 Control theory

This section describes the basic's in control theory in terms of what type of regulator are available to the industry. Along with the theory, a suggested tuning process is presented.

2.5.1 PID-controller

The most frequently used controller in the industry is a PID-controller. The controller utilizes three different control properties which all play a different part controlling a system.

- **P** - stands for **P**roportional and can be seen as a direct gain to the system. A large P-part will make the system fast and responsive but it will lower the stability margins of the controller.
- **I** - stands for **I**ntegration. Increasing the I-part lowers the stability margins and makes the system more responsive but is essential for eliminating static errors over time.
- **D** - stands for **D**erivative and studies the change of error in the system being controlled. An increase of the D-part will make the controller faster but may introduce a jumpy control process.

The continuous PID-controller can be mathematically described as,

$$u(t) = \underbrace{K_p e(t)}_{P} + \underbrace{\frac{K_p}{T_i}\int^t e(t)\,dt}_{I} + \underbrace{K_d T_d \frac{de(t)}{dt}}_{D} \tag{2.2}$$

where $u(t)$ is the control signal and $e(t)$ is the error between the reference and the measurement, K_p is the proportional gain and T_i and T_d are time constant connected to the integral and derivative time respectively [22].

2.5.2 Tuning process

The PID-regulator used in the model suggests a four-step procedure when tuning for desired behavior [23].

1. Set very large limits and select a P-controller. Manually enlarge the K_p parameter until the closed-loop system behaves satisfactorily.
2. Add integral action and manually adjust T_i until the remaining error goes to zero.
3. If a faster dynamic is desired, add derivative action and start small and sequentially enlarge T_d until the model behaves as desired.
4. Set the limits to the specification.

2.6 OPC UA

OPC UA is a communication protocol developed after studying 15 years of usage of classic OPC communication [24]. Classical OPC is still the most used in the industry since being first implemented but the advantages of OPC UA is increasing. One of the most advantageous parts of OPC UA is that it is platform independent. Classical OPC requires Windows but OPC UA can be implemented using Windows, MacOS or Linux [25]. OPC UA is a communication protocol allowing machines to send and receive data over a secure and reliable channel to one another. This communication takes place in a server/client type of fashion. A server is set up and is given an endpoint address to which clients can connect via TCP/IP. Clients then send and receive data containing information about, e.g variables names, types and current values.

2.7 PLC

A Programmable Logic Controller (PLC) is an industrial computer used typically in manufacturing processes and assembly lines. It is also suitable for harsher environments due to its robustness and reliability [26]. The PLC allows industries to connect sensors and actuators to a computer on-site and carry out logic control sequences to monitor and regulate the process. The hardware setup for a PLC can be described as a rack containing a power supply, a CPU and several Input/Output (I/O) cards mounted on the rack, seen in Figure 2.5

Figure 2.5: An overview of a PLC and its components. Adapted from [27]

The operation takes place in scan cycles and can be described as a three step procedure.

1. Read inputs
2. Execute programs which contain logic control
3. Write outputs

These steps take place during one period and depending on the configuration, the cycle time varies in time from project to project. A PLC will have an I/O-list which is set up in the hardware configuration by the PLC programmer. This list should be set up in a structured manner to let other users of the PLC code easily identify what variables are read as inputs and what variables the PLC writes to as outputs. To be able to run the PLC-code provided by RISE, a software configuration is set up. This configuration includes the required libraries needed to simulate the PLC. A guide on what libraries used and how to install them is described in Appendix A.

2.8 SIMIT

SIMIT is an industrial simulation platform developed by Siemens AG, which is used to set up models of factories or other industrial processes in a simulating environment. SIMIT comes with a number of existing libraries containing standard components such as tanks, valves, conveyor belts and connectors. It is also possible to create components on your own using SIMIT Component Editor (SIMIT CTE). Here a user can set up specific I/O- connections, component behavior and design a custom operating interface that lets an operator intuitively interact with the component. SIMIT CTE also supports FMI-standard and thus makes it possible to create components based on an FMU created in another modelling tool.

Models being simulated in SIMIT can be regulated by controllers communicating with SIMIT through an OPC UA server. With a server up and running SIMIT can make an OPC UA client coupling that lets SIMIT send and receive data through the server which has other clients connected to itself dealing with the control process. A description of this coupling procedure is described in Appendix B.

3

Model

This chapter describes the created model and the two simulation setups. The first one is a noninteractive setup in OpenModelica being controlled by a PID-controller. The second one is an interactive, real-time, simulation in SIMIT being controlled using an HMI which is communicating with the model and control logic on a PLC through an OPC UA server.

3.1 Libraries used

The model was developed using two libraries:

- **Modelica** version 3.2.3
- **Buildings** version 7.0.0

Modelica was used for the standard components of the model, such as PID-regulator, I/O-connectors, volume and heat ports. In the Buildings library, two models were used to model the thermodynamic behaviors of two materials. A brick wall and a gypsum board. The Brick was used to simulate the amount of heat dissipating through the walls of the furnace. The gypsum board is a test object trying to withstand the heat generated from the combustion inside the furnace.

3.2 Components of the furnace

3.2.1 Burner

The combustion of LPG is modeled as a generated heat flow from a given burner setting. In Figure 3.1, the burner model in OpenModelica is presented. The burner setting is an input value in the range of $[0, 1]$ and multiplies this input with a gain of the maximum power generated by the burner. This gain is then fed through a component called prescribedHeatFlow, from the Thermal library in OpenModelica. It takes a given heat flow as input and lets the modeler inject the heat flow amount through a thermal port (port_b) which can be connected to other Modelica components. In this case, the burner is connected to the heat ports of the furnace, test object and furnace brick wall as can be seen in Figure 3.3.

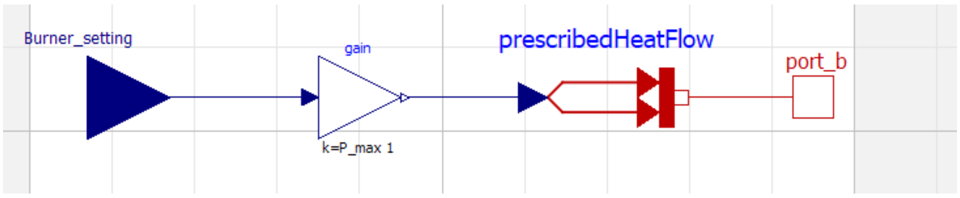


Figure 3.1: An overview of the burner model created in OpenModelica

3.2.2 Test object - Gypsum board, 15.9mm

A test object is created in OpenModelica using the Buildings library. This model describes a 15.9mm gypsum board of type C. Type C gypsum board provides a thermal resistance since being built with reinforced glass fibres in its structure [28]. By reason of being a non homogeneous, multi-layer material, thermodynamic properties of each layers and how they affect one another is hard to determine. The model is therefore simplified to be a single layer homogeneous model with the thermodynamic properties of a type c gypsum board, derived from Manzello et al. [10]. A heat flow is passed through port_a and into the model itself. The Gypsum_model is a component called singleLayer from the library Buildings.HeatTransfer.Conduction and is a model for single layer heat conduction. The model characteristics are declared in a table and are shown in Table 3.1.

Parameter	Unit	Value
Width	$[m]$	1.22
Height	$[m]$	2.44
Thickness	$[m]$	0.0159
Thermal conductivity	$[W/(m.K)]$	0.276
Specific heat capacity	$[J/(kg.K)]$	1017
Mass density	$[kg/m^3]$	800

Table 3.1: Model characteristics of a 15.9mm thick gypsum board

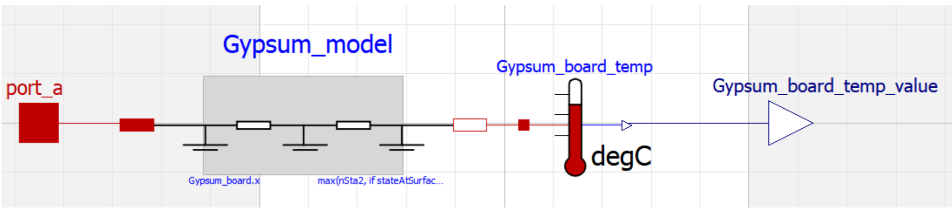


Figure 3.2: An overview of a Gypsum board (15.9mm) test object created in OpenModelica

3.2.3 Brick wall

The temperature element inside the brick wall is placed in the center of its thickness. The wall is a total of 300 mm thick and because of the placement of the temperature sensor only the inner half of the total wall is modeled. The model is build in the same way as the test object described in section 3.2.2, but with different model characteristics. The model characteristics are declared in a table and are shown in Table 3.2.

Parameter	Unit	Value
Width	$[m]$	3
Height	$[m]$	3
Thickness	$[m]$	0.15
Thermal conductivity	$[W/(m.K)]$	0.89
Specific heat capacity	$[J/(kg.K)]$	790
Mass density	$[kg/m^3]$	1920

Table 3.2: Model characteristics of a 150 mm thick brick wall

3.2.4 Furnace

The complete model of the system can be seen in Figure 3.3 and gives an overview of the different components required to make the system complete. The furnace studied in the project uses six gas burners each producing 0.7 MW of power. When operating in full capacity the furnace can deliver 4.2MW of power into the furnace chamber. The heat flow is distributed in equal amounts to the three main components of the furnace. The volume is the furnace's chamber containing the medium. The Brick_model represents the outer wall of the furnace and the test object, which in this setup is the gypsum board.

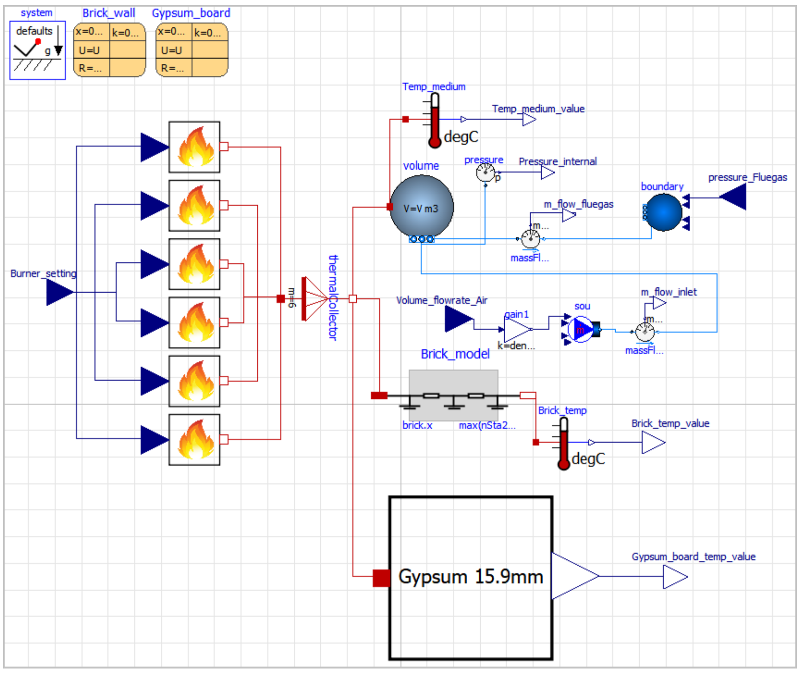


Figure 3.3: An overview of the furnace model created in OpenModelica

The furnace also models the mass flow of air passing through the furnace. The furnace model has several inputs and outputs presented in Table 3.3.

Input	Output
Burner setting	Volume temperature
Flue gas pressure	Pressure internal
Air volume flow inlet	Air mass flow inlet
	Flue gas mass flow
	Brick temperature
	Test object temperature

Table 3.3: Inputs and outputs connected to the furnace model

When running a simulation in OpenModelica, the developer can study all of the model's internal signals along with the ones provided through the I/O connections. However, when the model is exported and used as an FMU in another simulation platform i.e. in SIMIT, these output connections are used to connect signals within SIMIT and then send these through the OPC UA coupling to a client which deals with the control of the model. Depending on how this control procedure works, the most important parameters are set as outputs.

3.3 Control parameters

The tuning of the controller is made by following the steps presented in section 2.5.2. When only using a P-regulator, the model follows the reference well, but a static error is present during the complete simulation. This error is removed by introducing integral action in the controller. Finally, a small derivative action is added to increase the speed in which the controller adjust to errors between the reference and the measured signal. The parameters used in the simulations are presented in Table 3.4.

Control parameter	value
K_p	0.01
T_i	50
T_d	0.08
y_{max}	1
y_{min}	0

Table 3.4: Control parameters derived after tuning

4

Simulation

This chapter includes a description of the two different simulation setups being used in the project. First, an overview of the noninteractive simulation environment being used in OpenModelica is presented. This setup is used to verify that the model behaves as expected when being put through different scenarios.

4.1 Noninteractive simulation in OpenModelica

For simulation with a control sequence in OpenModelica, a model is created which the furnace model is imported into. The created model can be seen in Figure 4.1.

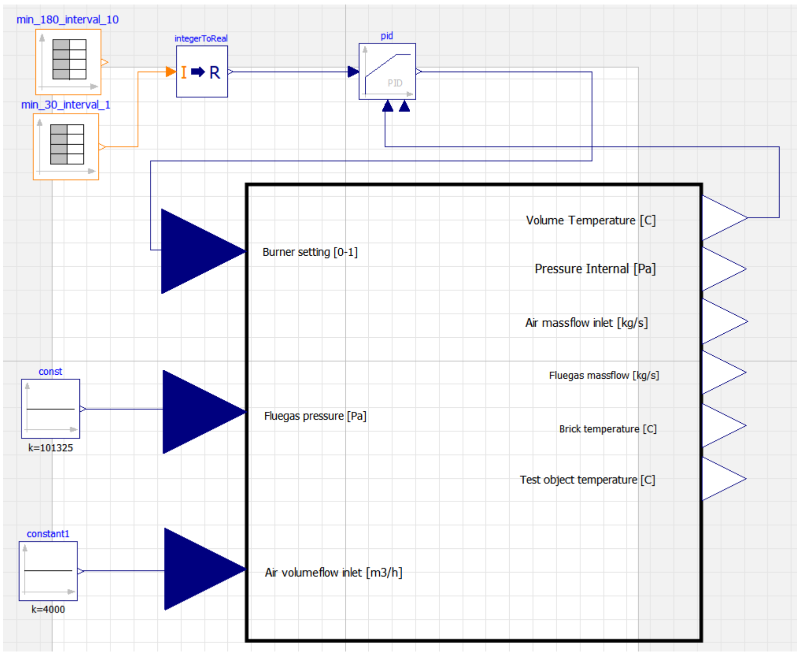


Figure 4.1: An overview of the simulation setup in OpenModelica

The control sequence is managed by a PID-controller that, in this setup, takes input from a table containing values from the ISO-834 temperature curve, described in (2.1). The control loop being used in OpenModelica is different from the one implemented in the PLC and thus the model will behave slightly differently when simulating the FMU in SIMIT.

Two tables are present, min_30_interval_1 handles a 30-minute-long simulation time with reference temperature each second. The other table, min_180_interval_10 is being used when running a three-hour-long simulation with temperature reference every 10-seconds. Since (2.1) rapidly increases and then produces a slow increase, an interval of 10-seconds is considered to be enough.

The scenarios being tested are:

1. A step in reference temperature.
2. Applying maximum heat flow into the furnace for 10 minutes and then switching off LPG-burners and studying the cooling effect from, different and constant values of the volume flow rates of air through the furnace.
3. A 30-minute-long fire resistance test conducted according to ISO-834, followed by a 30-minute-long cooling phase.
4. A 90-minute-long fire resistance test conducted according to ISO-834, compared against the physical system.

The expected behavior from the first function test is that the regulator increases the burner settings to produce a higher heat flow into the furnace. Depending on the controller parameters, the temperature should increase and settle around the new reference value.

The second scenario should show an increase in temperature followed by two different cooling curves depending on the different volume flow rates.

The third scenario is expected to follow the given temperature curve presented in section 2.1. Depending on the tuning parameters in the PID-regulator, the temperature may follow very precise with fast and large control signals as a consequence, or present a slower and more stable dynamic at the cost of the temperature curve not being followed as strictly.

The fourth scenario will test the similarities and differences between the model and the physical system. The comparison will focus on how the temperature develops during the course of time. Along with that, it will also be studied how the temperature is regulated.

4.2 Interactive simulation in SIMIT

This setup describes the interactive simulation setup in SIMIT, that is more complex and closer to a real industrial user case. Here, the model is meant to be simulated in SIMIT and controlled from an HMI communicating with a PLC through an OPC UA-server. The model is exported from OpenModelica as an FMU and imported into SIMIT CTE. The FMU-component is given an operating interface making it possible to control input parameters manually in a structured manner if an operator would like to set input parameters without using the HMI. The FMU-component and its operating interface in SIMIT can be seen in Figure 4.2.

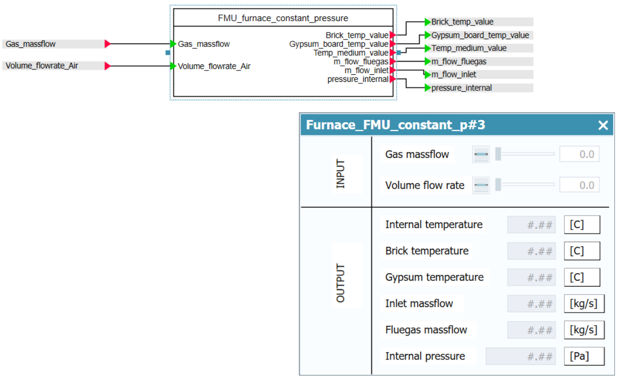


Figure 4.2: Usage of FMU-component in SIMIT alongside the operating interface

4.2.1 SP-Styr

At the physical system, the interaction with the furnace takes place through an HMI called SP-Styr, a tool created by engineers at AFRY using LabVIEW. This tool lets an operator use a graphical interface to interact with the simulation. The HMI communicates with the PLC through an OPC UA-server. This is a continuous bi-directional communication sending e.g sensor values and receiving control parameters during the course of the simulation. This tool is meant to be used when conducting a simulation in SIMIT.

4.2.2 Signal communication

When utilizing the interactive simulation setup, control signals and reference values are being sent to all clients coupled with an OPC UA server. In Figure 4.3, this communication is being shown. Description of the signal blocks inside of SP Styr can be seen in Appendix C.

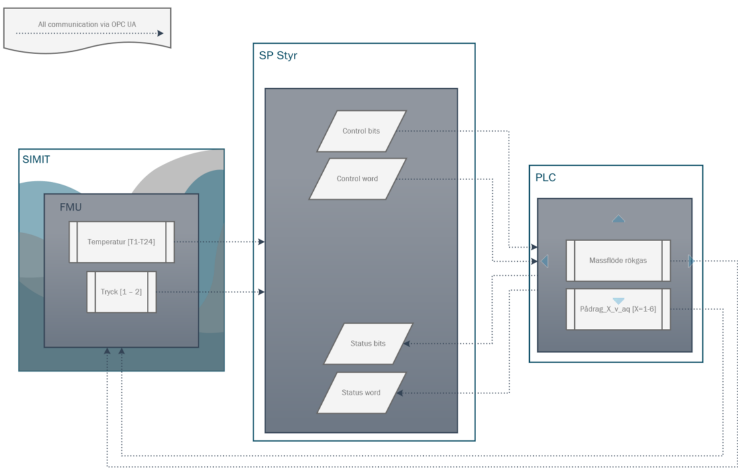


Figure 4.3: Signal communication in an interactive simulation setup in SIMIT.

5
Results

The results of the project are presented both as simulations conducted in Open-Modelica, to test the validity of the model and as a conceptual setup for the final complete project. The modelled system did not meet all requirement for simulation in regards to automatic control using the existing PLC-code.

5.1 Non interactive simulation in OpenModelica

The results are evaluated from the outcome of the simulated test cases described in section 4.1. In addition to the results, brief describing texts are also presented along with the plotted simulation data.

5.1.1 Test case 1

At time $t = 200s$ a step in reference temperature is made and as can be seen in Figure 5.1 the temperature inside of the furnace increases rather slowly, but with a bit of an overshoot followed by a constant decrease until finally reaching the new reference temperature.

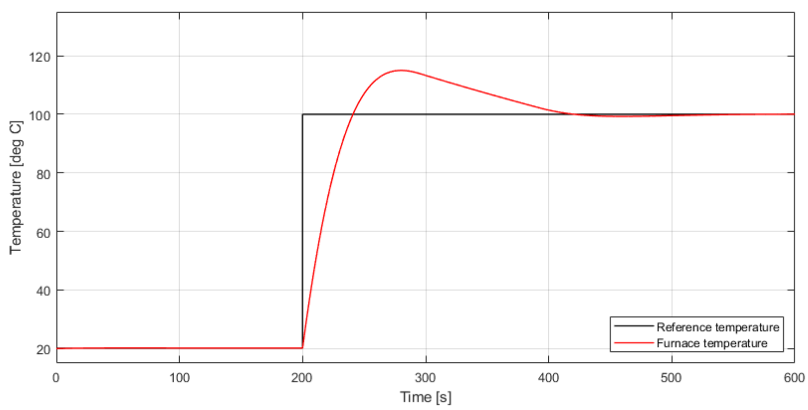


Figure 5.1: Simulation of a step in reference temperature from $20°C$ to $100°C$ at time, $t = 200s$ in OpenModelica.

The control signal from the PID-controller in Figure 5.2 shows a spike at $t = 200s$ followed by a decrease until just before $t = 300s$ letting the furnace cool on its own by outputting a control signal of zero until the new reference temperature have been reached.

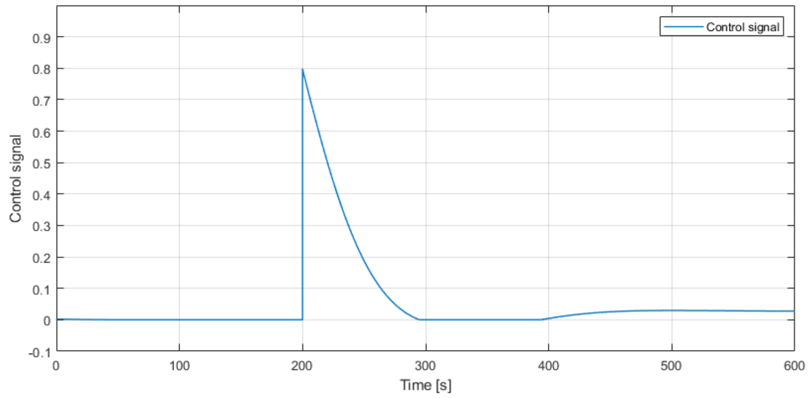


Figure 5.2: Control signal from a step in reference temperature from $20°C$ to $100°C$ at time, $t = 200s$ in OpenModelica.

5.1.2 Test case 2

The two different volume flows of air being studied. The red curve in Figure 5.3 shows a lower volume flow of 4000 $\frac{m^3}{h}$ and the blue curve shows a larger volume flow of 8000 $\frac{m^3}{h}$. The larger volume flow through the furnace brings a larger cooling effect on the system and thus a slower increase but faster decrease in temperature is being shown.

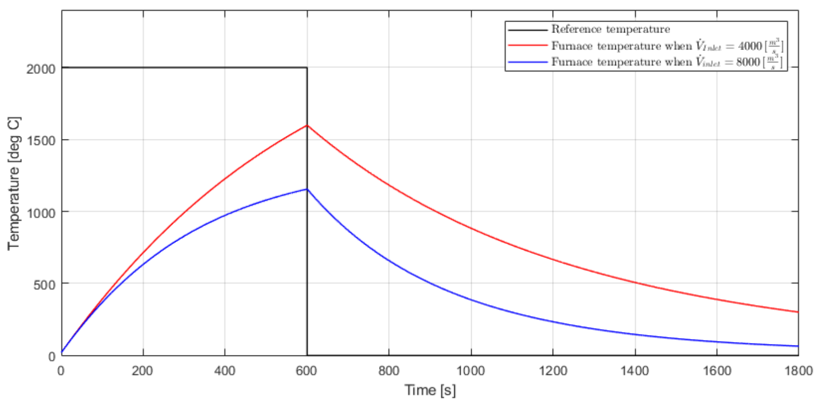


Figure 5.3: Simulation in OpenModelica with an unreachable reference temperature to force maximum heating. Two different volume flows of air through the furnace are studied. At time $t = 600s$ the reference temperature is set to zero letting the furnace cool with no heat being added to the system.

The control input in Figure 5.4 shows a maximum control signal being generated for the first $600s$ since the reference temperature in not reachable for the model in that period of time. Trivially, when the reference temperature is set to $20°C$ at time $t = 600s$, the control signal is set to zero for the remaining simulation time.

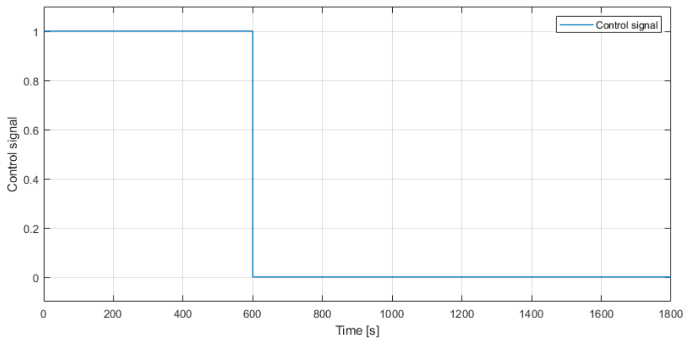


Figure 5.4: Control signal from a step in reference temperature from $2000°C$ to $20°C$ at time, $t = 200s$ in OpenModelica.

5.1.3 Test case 3

A 60 minutes long simulation divided into 30 minutes of heating according to ISO-834 followed by a 30-minute long cooling time is shown in Figure 5.5. The red curve shows the furnace temperature and during the first minute of the simulation, the furnace can not provide sufficient enough heat flow to follow the ISO-834 curve. When the furnace reaches the reference temperature a slight overshoot occurs and after settling from this overshoot the furnace temperature follows accordingly to the reference temperature.

The temperature of the gypsum board shows a rather rapid increase when compared to the temperature of the brick. Since the temperature of the gypsum board reaches a higher value than the furnace inner temperature, a cooling effect is applied to the board at time $t \approx 35\,min$. The temperature of the brick wall is slowly increasing during the whole simulation since it always stays below the temperature curve of the furnace. It is thereby affected by a positive heat flow passing through the brick wall from the inside of the furnace.

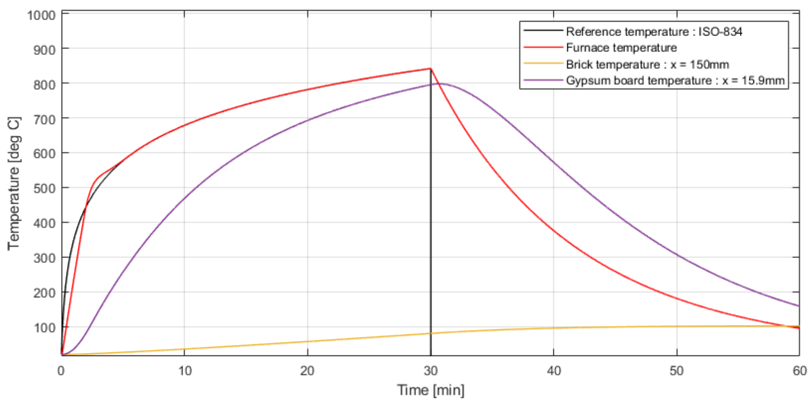


Figure 5.5: Simulation in OpenModelica according to ISO-834 for a duration of 30 minutes followed by a cooling process for a duration of 60 minutes. Reference temperature along with the effects on furnace temperature, outer brick temperature and test object temperature being shown.

The control signal generated during the simulation is seen in Figure 5.6. This control signal shows a dynamic behavior during the complete course of the heating phase. As mentioned above, the amount of needed heat transferred into the system to follow the reference temperature comes up short due to the maximum capacity of the furnace being insufficient. Therefore the control signal is from the start staying constant at its maximum capacity. After the temperature has caught up with the reference, around $t = 3\,min$, the control signal is decreased until settled around 30% of its maximum capacity. At $t = 30$ min the heating phase is done and the control signal turns to zero for the remaining simulation time.

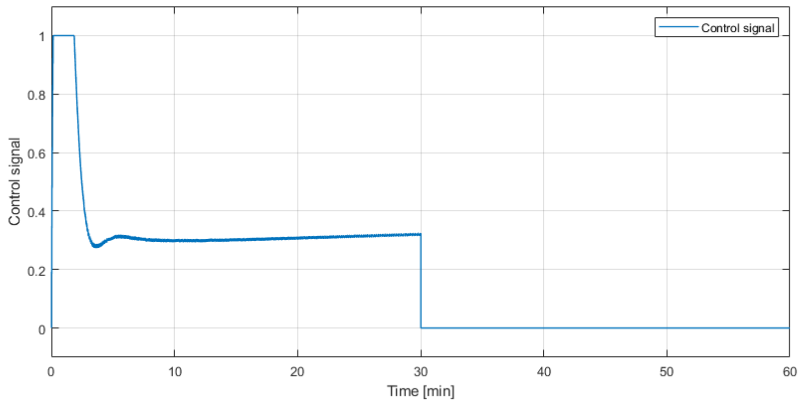


Figure 5.6: Control signal during a 60 minutes long simulation in OpenModelica according to ISO-843.

The provided heat flow during the simulation is shown in Figure 5.7. It follows the same curve as the control signal in Figure 5.6 since the control signal is modelled to be a percentage setting of the capacity of the furnace. From Figure 5.7, it is easy to understand that the furnace operates at full capacity during the first three minutes of the simulation.

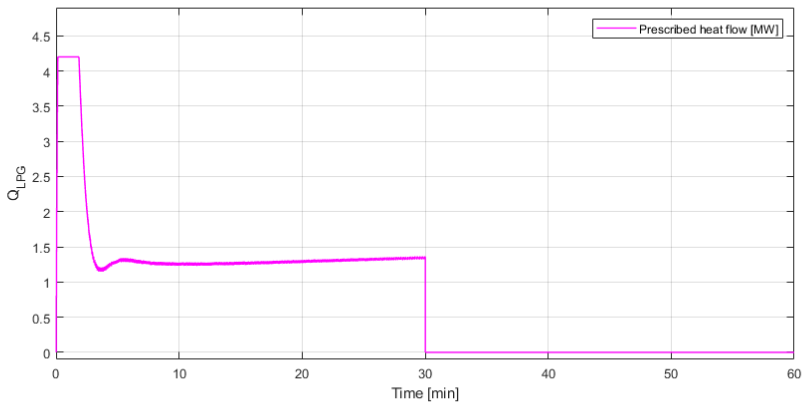


Figure 5.7: Prescribed heat flow in MW during a 60 minutes long simulation in OpenModelica according to ISO-843.

5.1.4 Comparison against physical furnace

The final test case is shown in Figure 5.8 and depicts a comparison between the simulated model and real data from the physical system. It can be seen that the red and blue temperature curve both have difficulties following the black reference temperature for the first five minutes. This may be due to the lack of power available since both furnaces are running on full capacity during this time period. Along with that, both temperature curves also show a slight overshoot when reaching the reference, at time $t \approx 2\,min$. This overshot settles in approximately three minutes after its first appearance. This time-window can be seen in the magnified box in Figure 5.8. During the rest of the run time, the blue curve shows difficulties following the reference. A constant negative error is present for the entirety of the following run time.

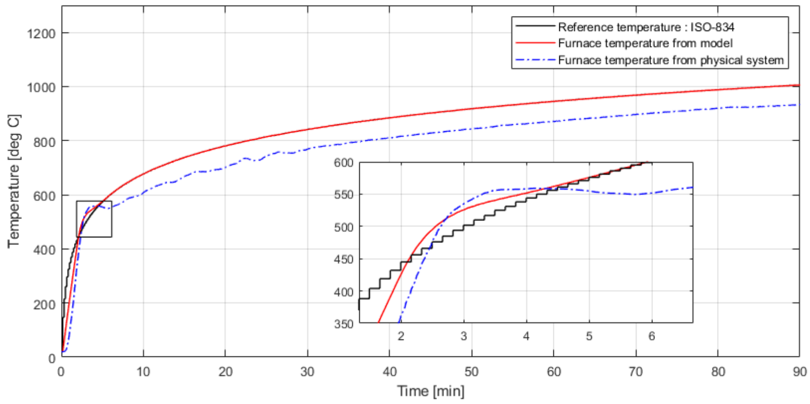


Figure 5.8: Simulation in OpenModelica according to ISO-834 for a duration of 90 minutes. The simulation is compared against real data from the physical system.

5.2 Interactive simulation in SIMIT

The interactive simulation setup in SIMIT depends on many different clients to communicate and work together, as was described in 4.2. To refresh the memory on the setup, the clients communicating with one another were:

- FMU in SIMIT
- SP-Styr for HMI
- Soft PLC in Control Expert
- All communication through an OPC UA server

Results from the complete setup when all of these clients are communicating were not achieved. Separate results from each clients are provided in sections 5.2.1 through 5.2.4.

5.2.1 FMU in SIMIT

Simulating the FMU in SIMIT without any control through an interactive interface is achieved. The FMU is controlled using the operation interface created in SIMIT CTE shown in section 5.2.1. Figure 5.9 shows the user telling the burners to operate at various values. The rate of change in temperature can be seen as having different large values. When the burners are set to be high, the rate of change increases and when the burners are set low the rate of change decreases. Also, when the burners are set to be equal to zero, the temperature decreases very slowly. This is due to the short time span in the simulation.

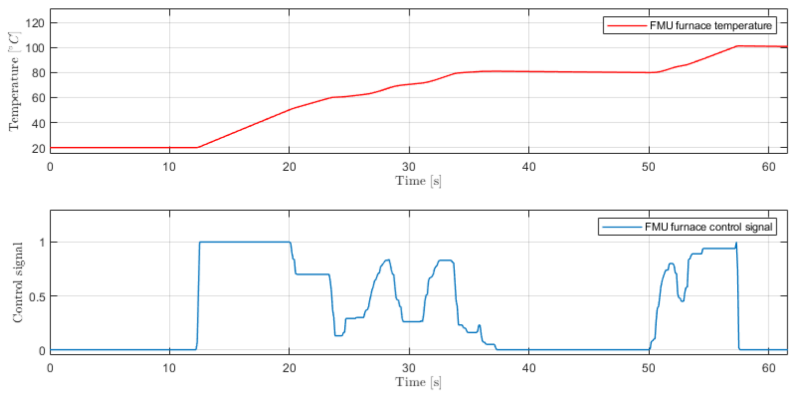


Figure 5.9: Manually controlled burner settings in interactive simulation in SIMIT.

5.2.2 SP Styr for HMI

SP Styr could not be run together with the logic provided by the PLC. This was due to the fact that the HMI was waiting for the PLC to complete the initialization procedure. The PLC got stuck at some of the initialization steps and thus not sending the needed signals to the HMI that the system was go for simulation. One problem identified is that insight about how the PLC-code operates is limited to a few number of people. Along with this, the documentation and overall code structure made it difficult to find out what procedures and where, inside of the PLC, the code got stuck. If the PLC-code would have had better documentation and a more desirable structure, then the problems that was encountered would not have arisen.

5.2.3 Soft PLC in Control Expert

Even though the soft-PLC was able to run in simulation mode, as can be seen in Figure 5.10, the PLC was waiting for certain signals from hardware components that were not present in the built setup. Sometimes the PLC would set itself into HALT-state which means that some procedures have taken too long to execute.

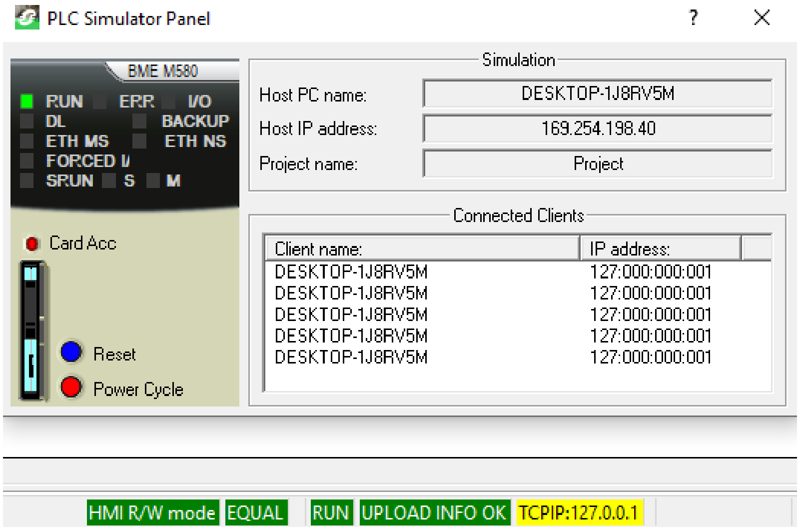


Figure 5.10: PLC Simulator Panel in Control Expert showing the soft-PLC being simulated on a local host in its *RUN*-state.

5.2.4 Communication through OPC UA server

The communication setup through the server is finished in SIMIT. The signals from the PLC have successfully been imported through the OPC UA server but since the PLC awaits certain signal values for its initialization process to complete, no simulation can take place. A part of the configuration is shown in Figure 5.11. The temperature from the FMU is being written to T_1 through T_6 and the burner setting is being read and computed as an average of *Pådrag_brännare_1* through *Pådrag_brännare_6*. The signals being used to compute the burner setting have been manually added to the OPC UA server to prove a concept. These should in a later stage be replaced with the real signals coming from the PLC.

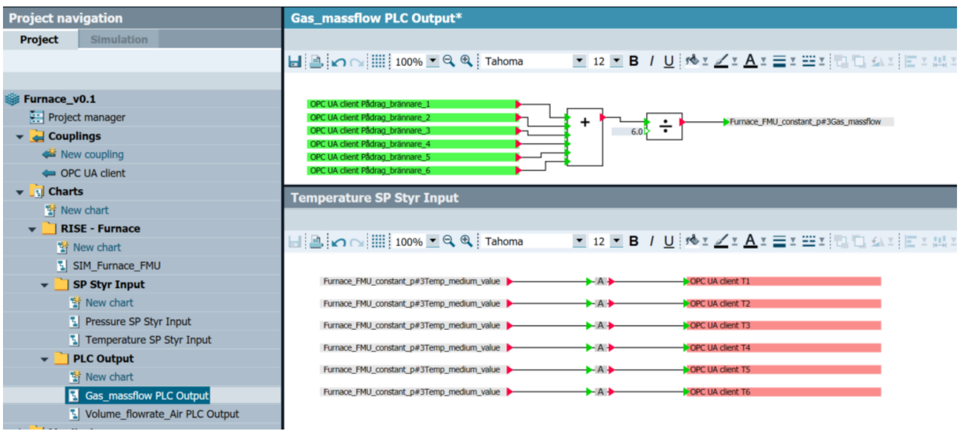


Figure 5.11: Signal configuration of temperature and burner setting in SIMIT.

6
Conclusion

From the studied literature, developed model and conducted simulations, conclusions associated with each area are presented in this chapter. Brief discussions regarding the conclusions are also carried out here.

6.1 Non interactive simulation in OpenModelica

The development of a model in OpenModelica is a time-consuming process. The ease of use of the standard library components speed up the process but learning how the program itself works takes a good amount of time. OpenModelica is nevertheless a good tool to use when modelling since it is free of use and no front-up cost is demanded to use the software. Some disadvantages are that it quite often crashes or freezes but a quick restart resolves these issues. Since the model itself is a simplified model of the real system, one can ask how accurate the model compares against the physical plant. Even so, the test conducted in section 5.1.4, and its result, shows that the simplified model performs close to the physical system.

6.2 Interactive simulation in SIMIT

The results in section 5.2 show that the FMU can be simulated and controlled manually in SIMIT, through the operation interface. The HMI itself works as intended since it is the same program that is used at the physical plant to interact with the system. However, the HMI was not allowed to start a simulation when communicating with the soft-PLC and the model in SIMIT. Since the complete system could not be simulated in SIMIT using control signals from the PLC, two important areas need to be addressed. First, one must describe the problems that occurred and then state what type of work that needs to be done to tackle these problems.

6.2.1 Problems when using control signals from the PLC

The fundamental problem lies within the soft-PLC and its way of handling signals. The PLC-code provided by RISE handles the complete test facility that has more components, which is not dealt with in the current setup. Systems that the initialization process needs to check are e.g. feed water and flue gas cleaning which both are not simulated. When the PLC waits for these signals for too long it sets itself in a HALT-state which means that some procedures have taken too long to execute.

Another problem was when trying to write to internal signals on the PLC through OPC UA, the PLC would not give permission to do so. To work around this issue, an employee at Schneider Electrics suggested to remove the internal addressees from the variable list in Control Expert. This however, brought issues later on since at some places in the PLC-code, variables was told to read from explicit internal addresses instead of from a specific internal variable. To read from explicit internal addresses is not to strive for.

6.2.2 Solutions to presented problems

Two paths have been identified trying to solve the issues occurring with the soft-PLC. The first is to scale down the PLC-code to a state where systems not being simulated are not present in the PLC-code. This could make it possible to simulate the furnace in SIMIT using control from the PLC. It would be beneficial for RISE since they could test out changes on the control system before implementing the changes to the physical plant. However, tests regarding the connecting systems would not be able to be conducted.

The other way is to go through the complete PLC-code structure and manually set the signals needed to complete the initialization sequence. This would make it possible to simulate the model using the real control sequence. Despite being possible to carry out, it is a time-consuming procedure since the PLC-code has got add-on functions built into the original code that have a lack of documentation. Problems may arise after the initialization process has been completed and these will have to be handled at a later stage.

6.2.3 Future work

Along with the suggested solutions, future work that could be of interest for both RISE and AFRY have been identified. One key area is to develop a well defined documentation about the system. It is suggested to set up a functional description over the system that describes the various functions in the PLC and what variables are used in each function. Along with this document, it would be of interest to create a structured I/O-list describing all the signals connected as inputs and outputs.

Another key area is the development of structured interfaces between the software products. In the interactive simulation setup in SIMIT, this interface have been prepared to send and recieve signals via OPC UA. SP Styr have been modified to read temperature and pressure from OPC UA instead of from a physical measurement box. The PLC is the final software in which its interface have not been adapted. To be able to connect all of the systems to one another, the interface of the PLC must be refined to deal with the OPC UA communication. The solutions to the issue with the PLC, and the suggested key areas could be developed by another master's thesis student, consultants at AFRY or RISE themselves.

6.3 Research questions

Two research questions were set up at the beginning of the thesis:

- Can the fire resistance of a simplified model of a physical product be evaluated using a digital twin approach?
- Can the developed simulation environment in the thesis be beneficial for the industry to either gain time savings in terms of reduced simulation time or improve the company's actual system?

The Buildings library brings the possibility to test the fire resistance of various simplified components. In this thesis, a model of a gypsum board was tested and it is concluded that it is possible to make the test object interact in a simulated environment. However, when the test results are being compared to real-made experiments, the results show somewhat similar behavior as data from the experimental tests made by Yoshitani et. al [9]. The experimental data shows a greater ability to withstand fire when compared to the gypsum board model used in the simulation presented in section 5.1.3. The thermodynamic properties of the gypsum board model are not the same as the one in the experimental case but similar nonetheless. Therefore it can be concluded that simplified models can be tested due to their ability to withstand fire. To bring the simulated results closer to reality, more complex models will have to be developed. As an example, the gypsum model used in the simulations was set as a single-layer homogeneous material with static thermodynamic properties. A more exact model would have multiple layers and temperature varying thermodynamic properties. The changes to the model can be made in OpenModelcia due to the fact that the creation of a multi-layer material comes from stacking single-layer materials on one another. This could bring the simulations closer to reality.

When it comes to the benefits that the digital twin brings to the industry, it can be concluded that the developed model of the furnace behaves similarly to the physical system when being simulated in a simplified setup, section 5.1.4. The benefits will be greater when the model can be simulated and controlled by the PLC. Then the industry can use the model to try out new control sequences without needing to interact with the physical plant. This would bring the time of simulation down and no need to book up the physical plant to conduct tests of new PLC-code. The time savings along with the possibility to conduct simulations without taking time from the physical plant would justify continued development of the complete system.

6.4 Ethics

The ethics behind turning industries and factories toward a more automated state has been wildly discussed. As the level of automation goes up, the need for human work labor goes down. As mentioned in 1.1.2, the ongoing revolution is not only affecting change in technology but also social and managerial planes must be seen as areas which will be impacted. When companies and industries implement technological changes in their factories, changes in job opportunities take place. The simulator would be able to take a bit of the workload of the real testing furnace. This would reduce the amount of human labor needed from one place and move it to another. It is reasonable to think that the total amount of human labor actually would increase instead of decrease. However, the operators' educational level would have to be higher for those who handle the simulation instead of the ones who handle the real testing furnace.

One of the United Nations sustainability goals states that ***"We must learn how to use and produce in sustainable ways that will reverse the harm that we have inflicted on the planet"*** [29]. If fire testing could be simulated instead of being performed in a real environment, fewer prototype products could be made. Small changes in existing products would not have to be built and tested once again. These changes could be handled with the change of parameter values in cyber-physical models and then being run through a virtual simulation. This would reduce unnecessary production that, in the long run, negatively impacts the health of the planet.

This leads into the next sustainability goal, ***"Climate change is a real and undeniable threat to our entire civilization"*** [30]. A switch from real testing towards a virtual simulation environment would lead to fewer combustion gasses being released into the atmosphere. In addition to this, the usage of materials when manufacturing prototypes and finished products will be significantly reduced since no real products will be needed for testing in a virtual environment.

Bibliography

[1] *Systems, control and mechatronics, MSc | Chalmers*, 2021. [Online]. Available: https://www.chalmers.se/en/education/programmes/masters-info/Pages/Systems-control-and-mechatronics.aspx.

[2] i-scoop.eu, *Industry 4.0: fourth industrial revolution guide to Industrie 4.0*, 2021. [Online]. Available: https://www.i-scoop.eu/industry-4-0/.

[3] J. Vachalek, L. Bartalsky, O. Rovny, D. Sismisova, M. Morhac, and M. Loksik, "The digital twin of an industrial production line within the industry 4.0 concept", in *Proceedings of the 2017 21st International Conference on Process Control, PC 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 258–262, ISBN: 9781538640111. DOI: 10.1109/PC.2017.7976223.

[4] M. M. Armstrong, "Cheat sheet: What is Digital Twin?", *ibm.com*, pp. 1–12, 2020. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/.

[5] Deloitte Insights, "The Fourth Industrial Revolution is here—are you ready?", *Deloitte Insights*, p. 28, 2018, ISSN: 0148-2963. [Online]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/4364_Industry4-0_Are-you-ready/4364_Industry4-0_Are-you-ready_Report.pdf.

[6] A. Albo and P. Falkman, "A standardization approach to Virtual Commissioning strategies in complex production environments", *Procedia Manufacturing*, vol. 51, pp. 1251–1258, Jan. 2020, ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.175.

[7] F. Westbrink and A. Schwung, "Virtual Commissioning Approach based on the Discrete Element Method", in *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018, pp. 424–429, ISBN: 9781538648292. DOI: 10.1109/INDIN.2018.8471954.

[8] A. Just, J. Schmid, and J. König, "Gypsum plasterboards used as fire protection - Analysis of a database", *SP Technical Research Institute of Sweden*, p. 30, 2010, ISSN: 0284-5172.

[9] K. Yoshitani, T. Shimizu, and M. Yoshida, "Study on prediction method for the insulation of gypsum board using inclusive thermal conductivity in fire resistance tests", *Japan Architectural Review*, vol. 1, no. 4, pp. 548–562, Oct. 2018, ISSN: 2475-8876. DOI: 10.1002/2475-8876.12057. [Online]. Available: https://doi.org/10..

[10] S. L. Manzello, S.-H. Park, T. Mizukami, and D. P. Bentz, "MEASUREMENT OF THERMAL PROPERTIES OF GYPSUM BOARD AT ELEVATED TEMPERATURES", Tech. Rep., 2004.

[11] P. Fritzson and P. Bunus, "Modelica - A general object-oriented language for continuous and discrete-event system modeling and simulation", in *Proceedings - Simulation Symposium*, volume 2002-Janua, 2002, pp. 365–380, ISBN: 0769515525. DOI: 10.1109/SIMSYM.2002.1000174.

[12] P. Fritzson, A. Pop, K. Abdelhak, A. Asghar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, L. Ochel, A. Palanisamy, V. Ruge, W. Schamai, M. Sjolund, B. Thiele, J. Tinnerholm, and P. Ostlund, "The OpenModelica integrated environment for modeling, simulation, and model-based development", *Modeling, Identification and Control*, vol. 41, no. 4, pp. 241–285, 2020, ISSN: 18901328. DOI: 10.4173/MIC.2020.4.1.

[13] *Single-Step ODE Solvers versus Multi-Step ODE Solvers - LabVIEW 2018 Help - National Instruments*, 2021. [Online]. Available: https://zone.ni.com/reference/en-XX/help/371361R-01/lvanlsconcepts/single_vs_multi_step_ode/.

[14] *Solving Modelica Models — OpenModelica User's Guide v1.18.0 documentation*, 2021. [Online]. Available: https://www.openmodelica.org/doc/OpenModelicaUsersGuide/latest/solving.html.

[15] *Functional Mock-up Interface*, 2021. [Online]. Available: https://fmi-standard.org/.

[16] *FMI Standard: Understand the two types of Functional Mock-up Units*, 2021. [Online]. Available: https://www.modelon.com/fmi-functional-mock-up-unit-types/.

[17] *ISO - About us*, 2021. [Online]. Available: https://www.iso.org/about-us.html.

[18] *ISO - ISO 834-12:2012 - Fire resistance tests — Elements of building construction — Part 12: Specific requirements for separating elements evaluated on less than full scale furnaces*, 2021. [Online]. Available: https://www.iso.org/standard/55660.html.

[19] *International fire curves – useful tool for designing fire safety - Promat*, 2021. [Online]. Available: https://www.promat.com/en-gb/construction/projects/expertise/33637/fire-curves/.

[20] *Modelica.Media.IdealGases.MixtureGases.FlueGasSixComponents*, 2021. [Online]. Available: https://doc.modelica.org/Modelica%204.0.0/Resources/helpDymola/Modelica_Media_IdealGases_MixtureGases_FlueGasSixComponents.html.

[21] M. Gordon, "NASA 5P-273 Interim Revision", Tech. Rep., 1976, pp. 73–772.

[22] B. Lennartson, *Reglerteknikens grunder*, Fourth. Lund: Holmbergs, 2002, p. 513, ISBN: 9789144024165.

[23] *Modelica.Blocks.Continuous.LimPID*, 2021. [Online]. Available: https://build.openmodelica.org/Documentation/Modelica.Blocks.Continuous.LimPID.html.

[24] *A Beginner's Guide to OPC – OPC Connect*, 2021. [Online]. Available: https://opcconnect.opcfoundation.org/2017/12/a-beginners-guide-to-opc/.

[25] *OPC and OPC UA explained | Novotek*, 2021. [Online]. Available: https://www.novotek.com/uk/solutions/kepware-communication-platform/opc-and-opc-ua-explained/.

[26] *What is PLC and Why Use It? | Telstar Instruments*, 2021. [Online]. Available: https://www.telstarinc.com/blog/what-is-plc-and-why-use-it/.

[27] *How PLCs Work*, 2021. [Online]. Available: http://www.plcdev.com/book/export/html/309.

[28] *Are You Confusing Type X and Type C Panels? Know the Differences. - DensElement Barrier System*, 2021. [Online]. Available: https://denselement.com/blog/differences-between-type-x-type-c/.

[29] The Global Goals, *Goal 12: Responsible Consumption and Production | The Global Goals*, 2018. [Online]. Available: https://www.globalgoals.org/12-responsible-consumption-and-production.

[30] ——, *Goal 13: Climate Action | The Global Goals*, 2017. [Online]. Available: https://www.globalgoals.org/13-climate-action.

[31] *PROFIBUS Remote Master | Schneider Electric*, 2021. [Online]. Available: https://www.se.com/ww/en/download/document/PROFIBUS+Remote+Master+V1.3/.

[32] *Where can I find the .gsd files for the TSX Momentum 170DNT11000 Profibus DP Communication Adapter?*, 2021. [Online]. Available: https://www.se.com/ww/en/faqs/FA199064/.

[33] *Modbus Communication DTM Library V2.6.16 | Schneider Electric*, 2021. [Online]. Available: https://www.se.com/ww/en/download/document/Modbus+Communication+DTM+Library/.

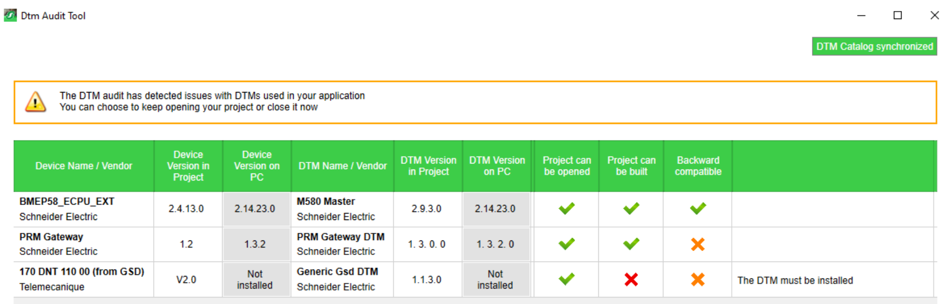
[34] *EcoStruxure™ OPC UA Server Expert - Documents and Downloads | Schneider Electric Global*, 2021. [Online]. Available: https://www.se.com/ww/en/product-range-download/66388-ecostruxure%E2%84%A2-opc-ua-server-expert/#/software-firmware-tab.

[35] *OPC UA Clients - Unified Automation*, 2021. [Online]. Available: https://www.unified-automation.com/downloads/opc-ua-clients.html.

A

Software configuration of Control Expert - DTM libraries

The PLC code is exported as a ZEF-file extension that supports applications containing DTM configuration. Given a ZEF-file, DTM audit tool can be run in Control Expert. This tool searches the file for DTM configuration and presents a list of packages/files that needs to be installed to be able to compile the project. This list will depend on the specific configuration of the PC that runs the code. Given the PC configuration used in this project, the list can be seen in Figure B.3.



Device Name / Vendor	Device Version in Project	Device Version on PC	DTM Name / Vendor	DTM Version in Project	DTM Version on PC	Project can be opened	Project can be built	Backward compatible	
BMEP58_ECPU_EXT Schneider Electric	2.4.13.0	2.14.23.0	**M580 Master** Schneider Electric	2.9.3.0	2.14.23.0	✓	✓	✓	
PRM Gateway Schneider Electric	1.2	1.3.2	**PRM Gateway DTM** Schneider Electric	1. 3. 0. 0	1. 3. 2. 0	✓	✓	✗	
170 DNT 110 00 (from GSD) Telemecanique	V2.0	Not installed	**Generic Gsd DTM** Schneider Electric	1.1.3.0	Not installed	✓	✗	✗	The DTM must be installed

Figure A.1: DTM audit tool in Control Expert with given list of needed packages/files

PRM Gateway can be downloaded from [31] and installed from the file explorer using the provided installation files.

170 DNT 110 00 (from GSD) can be downloaded from [32]. This package, unlike *PRM Gateway* must be downloaded and the extracted, but not installed from the file explorer. This package is installed inside of the Modbus module in Control Expert.

To be able to install **170 DNT 110 00** inside of the Modbus module, one must also download **Modbus Communication Library** [33]. Unzip these and install all packages included using the file explorer.

Navigate to *Tools - DTM Browser* and right click on **<Modbus:10.110.45.11 > PRM_GW**. Navigate to *Device menu - Additional functions - Add GSD In Library*. This will launch an extension that gives the user the means to install a downloaded GSD-file. Figure A.2 shows this graphically.

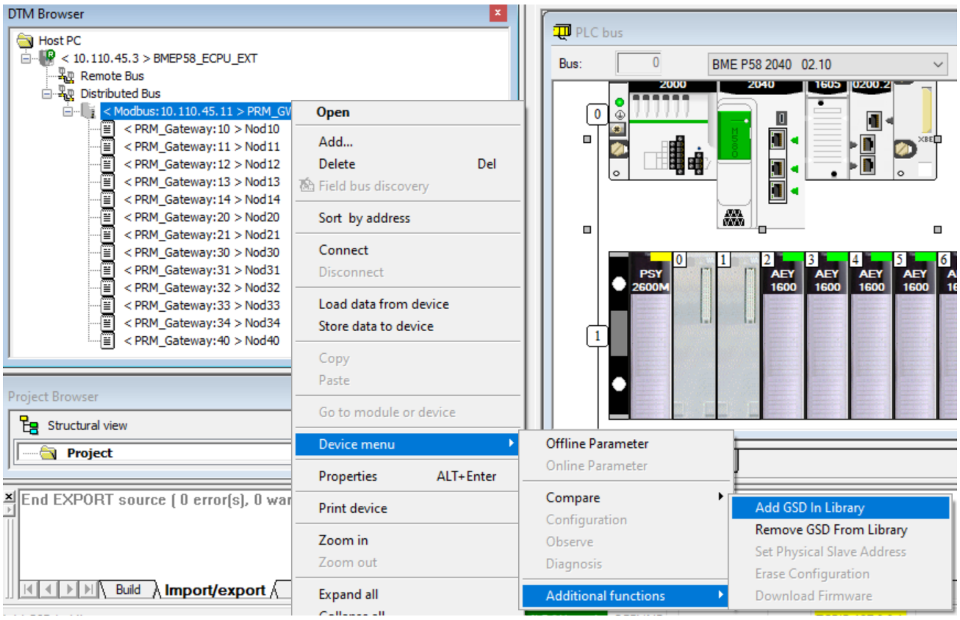
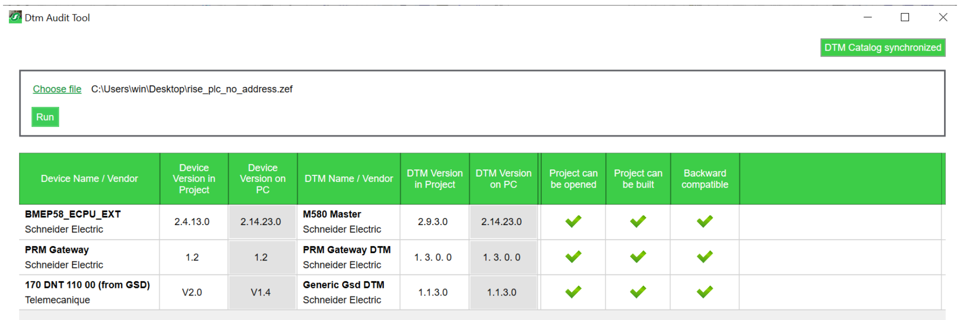


Figure A.2: Path to open GSD Addition tool

Next, browse to the path that contains the downloaded GSD-file. After choosing the correct path, complete the installation by following the instructions given on the GSD Addition tool.

Now run the DTM audit tool again and make sure the list is now updated with green check marks as can be seen in Figure A.3.



Device Name / Vendor	Device Version in Project	Device Version on PC	DTM Name / Vendor	DTM Version in Project	DTM Version on PC	Project can be opened	Project can be built	Backward compatible
BMEP58_ECPU_EXT Schneider Electric	2.4.13.0	2.14.23.0	**M580 Master** Schneider Electric	2.9.3.0	2.14.23.0	✔	✔	✔
PRM Gateway Schneider Electric	1.2	1.2	**PRM Gateway DTM** Schneider Electric	1. 3. 0. 0	1. 3. 0. 0	✔	✔	✔
170 DNT 110 00 (from GSD) Telemecanique	V2.0	V1.4	**Generic Gsd DTM** Schneider Electric	1.1.3.0	1.1.3.0	✔	✔	✔

Figure A.3: Updated list from DTM audit tool in Control Expert

B

Coupling between PLC and SIMIT through OPC UA

B.1 Server configuration

The server was initiated using EcoStructure OPC UA Server Expert Configuration tool that can be downloaded from [34]. In this program a new device is created in **File - New Device Alias**. This device will act as our server. The device is given a name and an address. Since all of the simulation will take place locally, **Device address 1** is set to a local host, 127.0.0.1. Also, to be able to see variables from the PLC, the check mark **Using Data Dictionary** is checked.

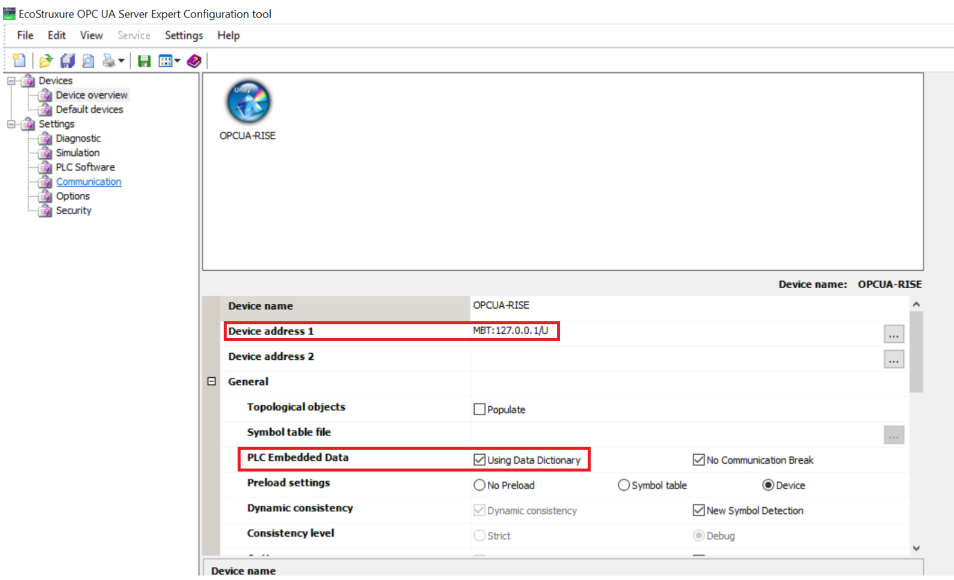


Figure B.1: General settings when setting up OPC UA server configuration

To make it easy to communicate over the server without needing to authenticate new users, **Security - SecurityPolicy** is checked as *Allow "None"*.

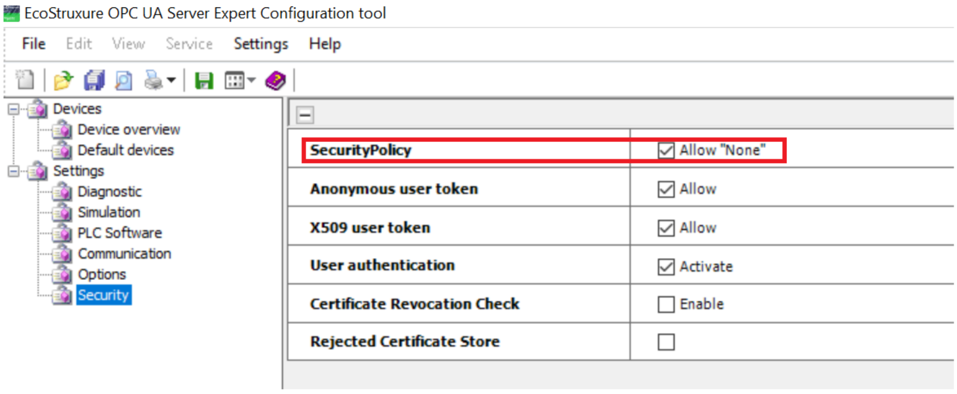


Figure B.2: Security settings when setting up OPC UA server configuration

Save the configuration and open up **EcoStructure OPC UA Server Expert** that is installed together with **EcoStructure OPC UA Server Expert Configuration tool**. This program will generate an endpoint address for which clients can connect to the server.

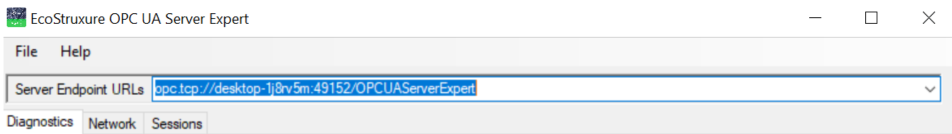


Figure B.3: Auto generated endpoint address for the server

B.2 Connecting a client to a server

In the project, a client called UaExpert from Unified Automation was used. It can be downloaded from [35] and lets users very easily connect to a OPCUA-server.

Navigate to **Server - Add** and specify Configuration Name and Authentication settings. As was mentioned in B.1, the security option was set up to allow new users to connect with no authentication protocol and thus the option of **Authentication Settings - Anonymous** is chosen. Endpoint Url is the generated address provided from **OPC UA Server Expert** also mentioned in B.1. Figure B.4 shows this graphically.

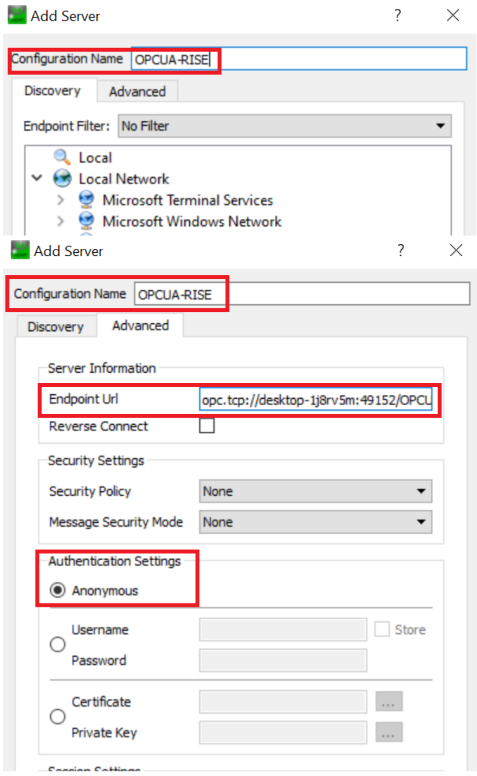


Figure B.4: Configurations settings in OPC UA client

After performing these steps the server/client communication has been established and the client can be connected to the server by right clicking on the server name in the **Project**-window and choosing **Connect**.

B.3 Simulating the PLC

In **EcoStructure Control Expert** navigate to **Tools - Project Settings - PLC Embedded Data** and make sure that **Data dictionary** is checked as in Figure B.5.

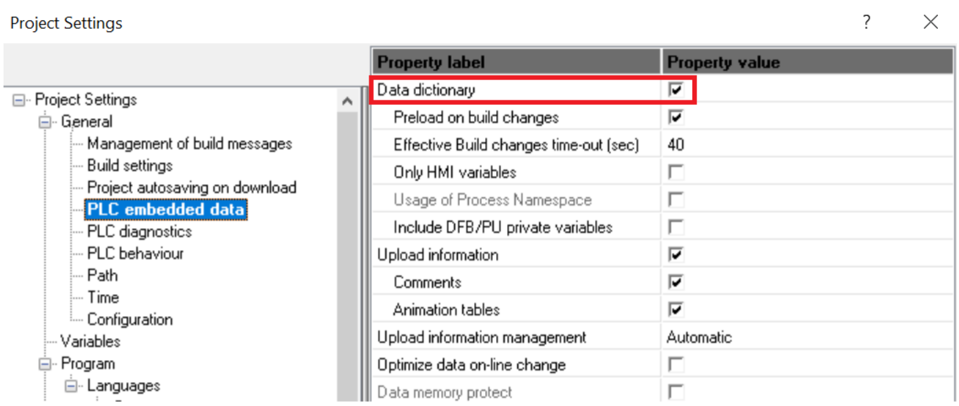


Figure B.5: Project settings in EcoStructure Control Expert

In **EcoStructure Control Expert**, set the PLC in simulation mode by clicking the red square as shown in Figure B.6.

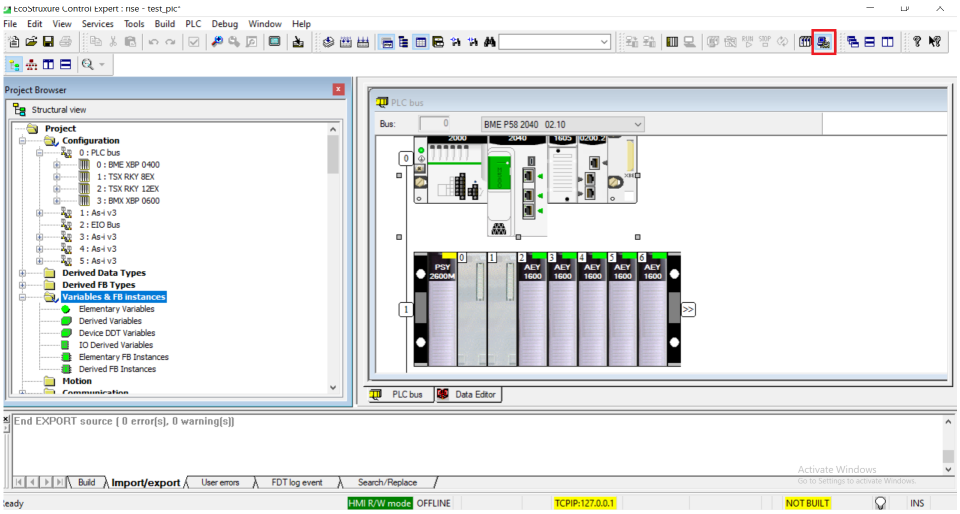


Figure B.6: Simulation mode for PLC

Navigate to **Build - Rebuild All Project** to compile the project. After compiling the project it can be saved. Now navigate to **PLC-Connect** to connect the PLC to the simulation rack. Next, transfer the project to the PLC by navigating to **PLC - Transfer Project to PLC**. Finally the PLC can be started by clicking **PLC - Run**. Now the **PLC Simulator Panel** will show the current state of the PLC simulation, as can be seen in Figure B.7.

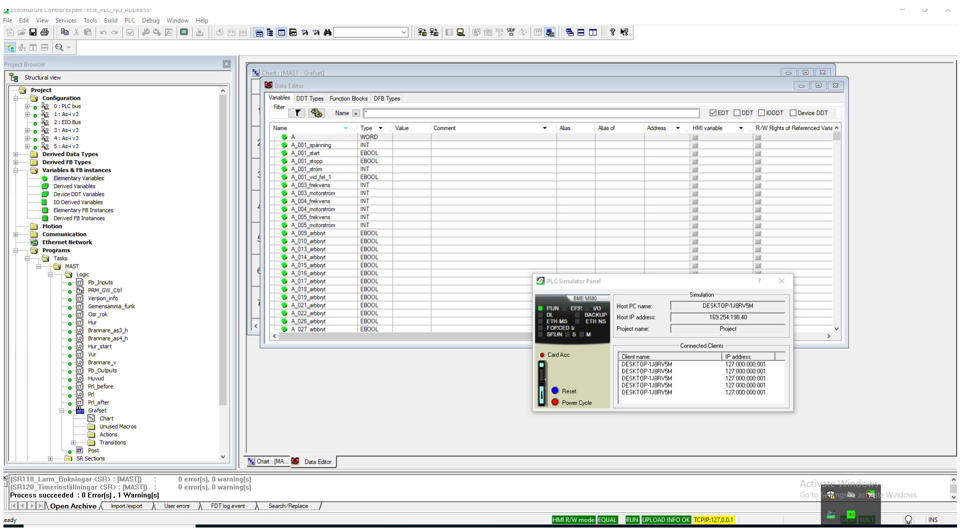


Figure B.7: Online simulation of the PLC

In **UaExpert**, inside the **Address Space**-window one can now navigate to the variable list by browsing to **Root/Objects/DA/OPCUA-RISE** as can be seen in Figure B.8.

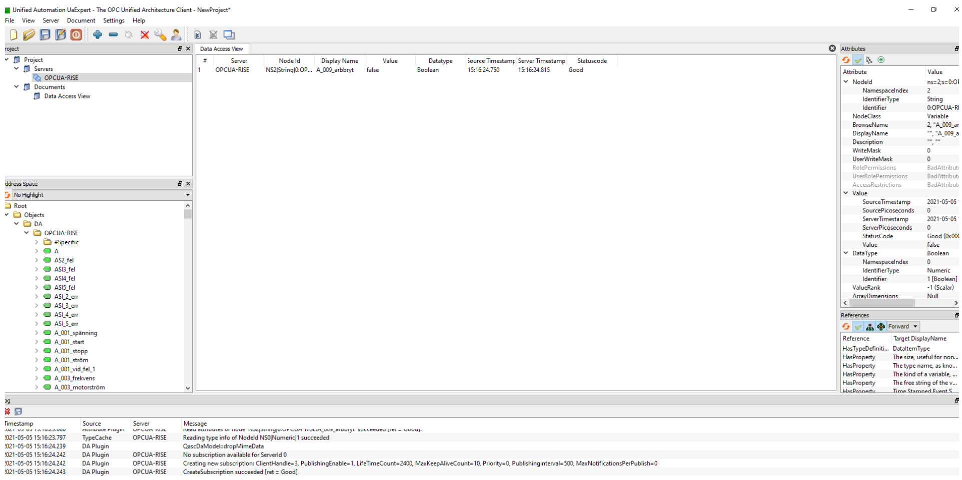


Figure B.8: Online client with variables being read from PLC

B.4 Coupling OPCUA server in Simit

To be able to read and write signals from and to the OPC UA server, the user must be granted full read and write privileges in four directories. This is done by selecting **Full control** for the User as shown in Figure B.11

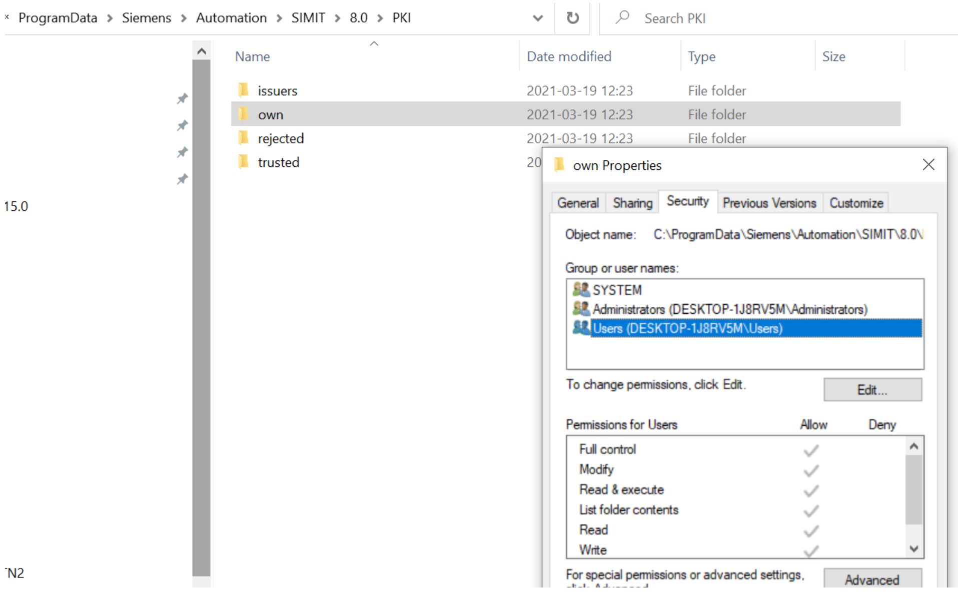


Figure B.9: User security privileges for reading and writing from OPCUA-server

The four directories are located in the following paths:

- C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI \own
- C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI \own \private
- C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI \rejected
- C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI \trusted

In **SIMIT SP**, under **Project navigation**-window select **Couplings - New coupling**. Chose to configure an **OPC UA Client** and paste the endpoint address generated in **EcoStructure OPC UA Server Expert** (described in B.1) under **OPC UA server URL** as shown in Figure B.10.

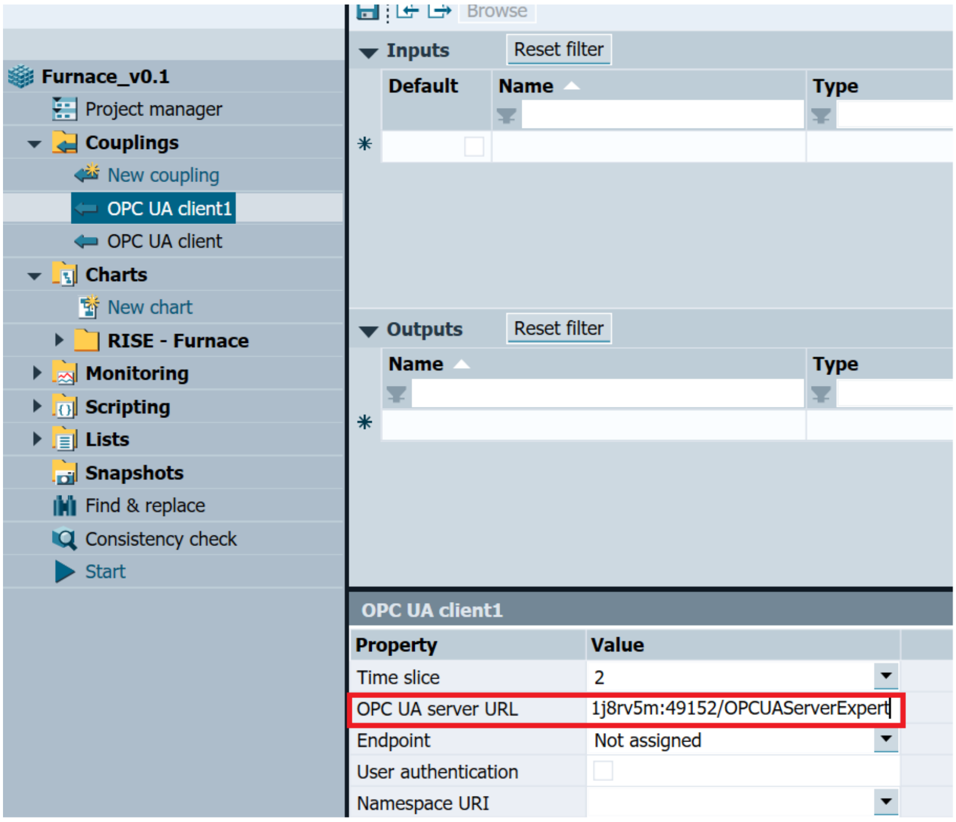


Figure B.10: OPC UA coupling configuration in SIMIT

Under **Endpoint** one may choose between several security protocols which will be used when establishing the connection. Since the server is configured to allow a user to be anonymous, the one in Figure B.11 is chosen.

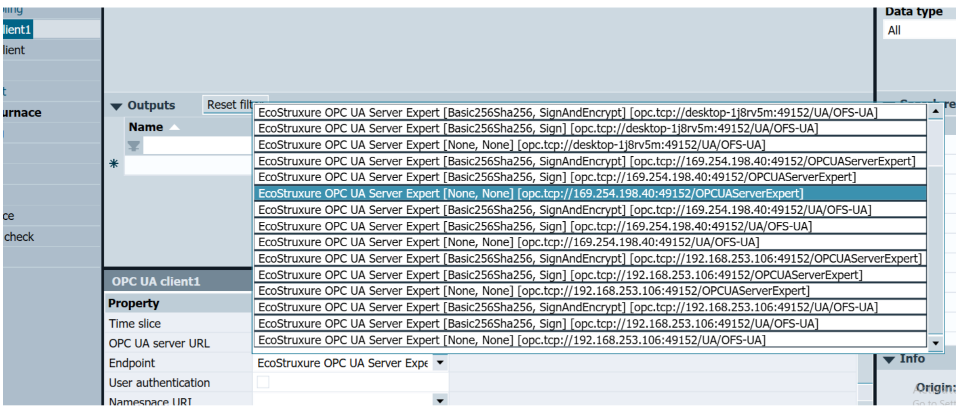


Figure B.11: Chosen Endpoint encryption protocol to be used when connecting to the server

The final setting to configure is the **Namespace URI**. The variables are contained inside of the Namespace shown in Figure B.12.

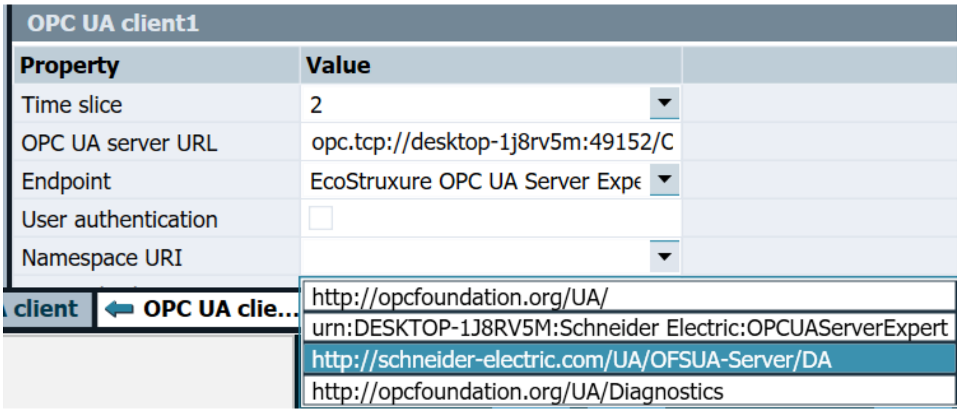


Figure B.12: Chosen Namespace URI to be used when connecting to the server and looking for the variables

Now, after clicking on **Browse** as shown in Figure B.13 the variables posted by the PLC to the OPC UA server will be imported into SIMIT and can now be used to connect SIMIT signals to real PLC-variables.

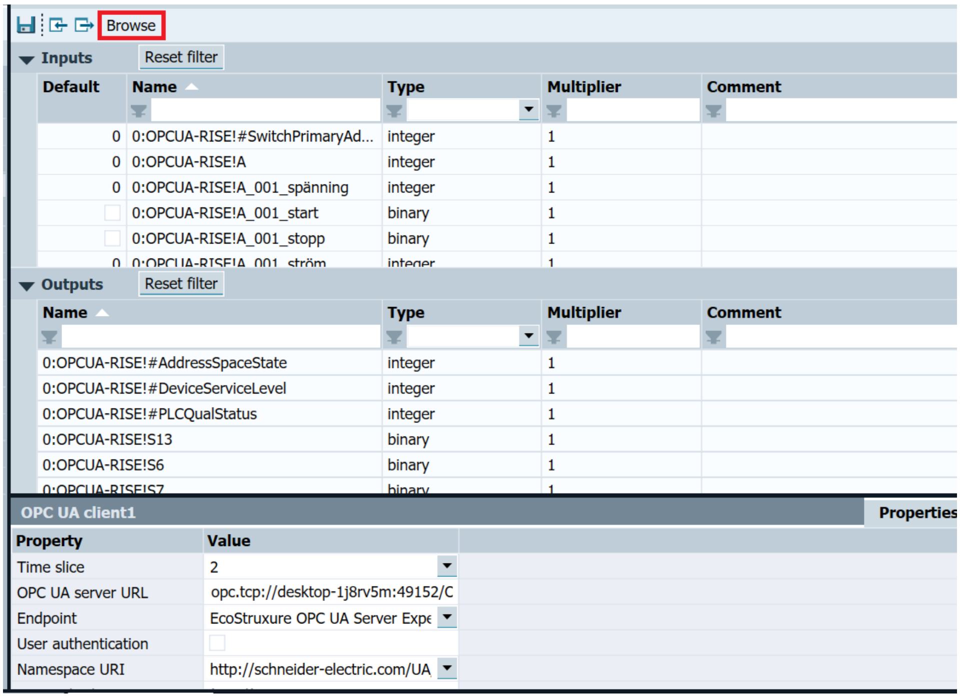


Figure B.13: Published variables from PLC imported in SIMIT via OPC UA server

C

Variable list from HMI

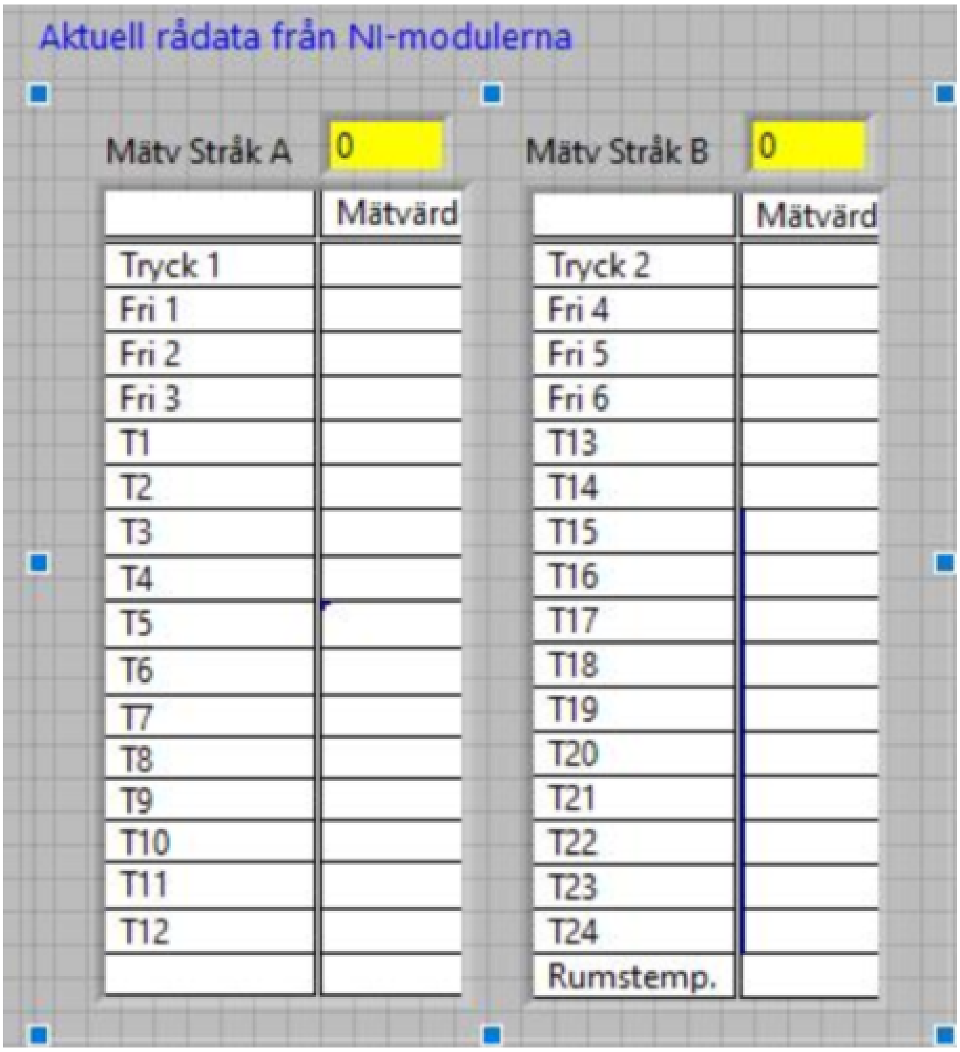


Figure C.1: Variable list 1

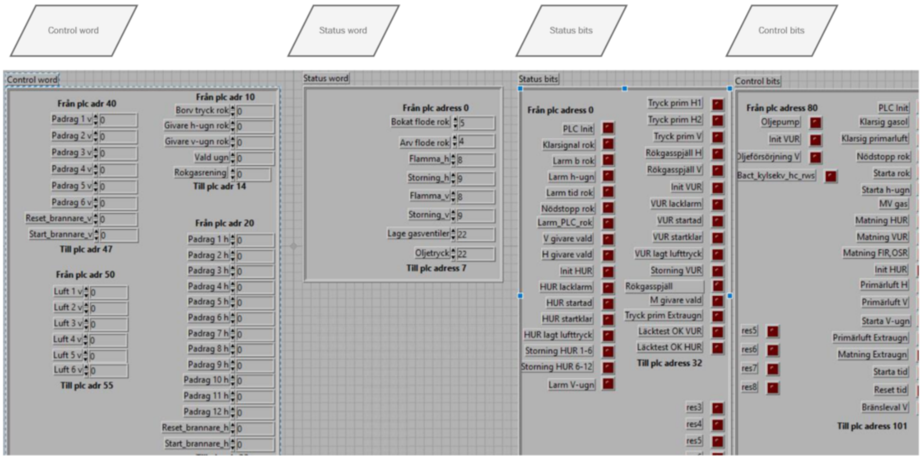


Figure C.2: Variable list 2

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY