



TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Electrical Power Engineering and Mechatronics

SIMULTANEOUS CONTROL OF INTERCONNECTED DIFFERENT SHAPE TANKS SYSTEM USING PROGRAMMABLE LOGIC CONTROLLER

**ERINEVA KUJUGA PAAKIDE SÜSTEEMI SAMAAEGNE
JUHTIMINE PROGRAMMEERITAVA
LOOGIKAKONTROLLERI ABIL**

MASTER THESIS

Student: Nihad Abdullayev

Student code: 194260MAHM

Supervisor: Kristina Vasiljeva, Associate Professor

Co-supervisors: Mart Tamre, Professor
Hossein Alimohammadi, PhD Student

Tallinn 2021

Non-exclusive Licence for Publication and Reproduction of Graduation Thesis¹

I, Nihad Abdullayev (date of birth: 18.09.1997) hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis "Simultaneous Control of Interconnected Different Shape Tanks System Using Programmable Logic Controller",

supervised by Kristina Vasiljeva and Mart Tamre and Hossein Alimohammadi,

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

¹ *Non-exclusive Licence for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

_____ (signature)

_____ (date)

Department of Electrical Power Engineering and Mechatronics

THESIS TASK

Student: Nihad Abdullayev, 194260MAHM
Study programme, MAHM02/18 - Mechatronics
main speciality: Mechatronics
Supervisor(s): Kristina Vasiljeva, Associate Professor, +372 620 2116
Co-supervisors: Mart Tamre, Professor
Hossein Alimohammadi, PhD Candidate

Thesis topic:

(in English) Simultaneous Control of Interconnected Different Shape Tanks System Using Programmable Logic Controller

(in Estonian) Erineva kujuga paakide süsteemi samaaegne juhtimine programmeeritava loogikakontrolleri abil

Thesis main objectives:

1. Research communication and hardware solutions for the Multitank system.
2. Develop a simultaneous control system by using a PLC-based PID controller.
3. Verifying the system by performing experiments, analysing results and providing suggestions for further developments.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Defining problem and project planning	September, 2020
2.	Literature review and analysis	December, 2020
3.	Research and analysis of system	September, 2021
4.	Performing experiments, analysing results	December, 2021
5.	Thesis defence	January, 2022

Language: English **Deadline for submission of thesis:** ".....".....20....a

Student: Nihad Abdullayev ".....".....20....a
/signature/

Supervisor: Kristina Vasiljeva ".....".....20....a
/signature/

Head of study programme: Anton Rassõlkin ".....".....20....a
/signature/

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

CONTENTS

PREFACE	7
List of abbreviations and symbols	8
List of figures	9
List of tables	10
1. INTRODUCTION	11
Overview	11
Problem statement	11
Thesis objective	12
Thesis structure	13
2. LITERATURE REVIEW AND BACKGROUND	14
2.1 Overview	14
2.2 Existing methods and literature review	14
2.3 Conclusion	20
3. HARDWARE AND SOFTWARE	21
3.1 Hardware selection	21
3.1.1 Overview of the Multitank system	21
3.1.2 ABB AC500 PLC	23
3.1.3 ABB CP610 HMI	24
3.1.4 I/O board between PLC and PC	26
3.1.5 I/O board specification	26
3.2 Software selection	29
3.2.1 Automation Builder	29
3.2.2 CODESYS	29
4. DEVELOPMENT	31
4.1 Work schematics diagram	31
4.2 MATLAB diagram	31
4.3 MATLAB network settings	32
4.4 Voltage divider	35
4.5 Automation Builder configuration	36
4.6 I/O module configuration	38
4.7 PLC and HMI communication configuration	39
4.7.1 Ethernet communication	39
4.7.2 Serial communication	39
4.7.3 HMI configuration	40
4.8 PID configuration	41
4.9 Control program development	44

4.9.1 Program structure	44
4.9.2 PID function block	47
4.10 HMI development.....	48
5. EXPERIMENTAL VERIFICATION OF IMPLEMENTATION	50
5.1 Verification of PLC-based PID controller	50
5.2 Comparing MATLAB controller with PLC controller	53
6. FUTURE DEVELOPMENTS	56
SUMMARY.....	57
KOKKUVÕTE	58
LIST OF REFERENCES	59
APPENDICES	62
Appendix 1	62
Appendix 2.....	63
Appendix 3.....	64

PREFACE

This master Thesis topic was proposed by Associate Professor Kristina Vasiljeva at Tallinn University of Technology. The challenge of integrating an ABB controller into an existing Multitank control system was accepted with great enthusiasm. The research of this thesis was carried out in the Alpha Control laboratory, where the Multitank system is located, at Tallinn University of Technology.

The main aim of this thesis project was to integrate an ABB controller into the Multitank system, develop a control algorithm and Human Machine Interface panel and optimise relevant control parameters of the PID controller to ensure a stable and accurate system by measuring liquid level accurately and reaching and maintaining the reference value in the control system.

I would like to express my sincere appreciation to my supervisors – Associate Professor Kristina Vasiljeva and PhD candidate Hossein Alimohammadi for their support, advice, and help to the author throughout this period. Also, I would like to thank Professor Mart Tamre for his guidance during the ‘Master seminar’ course and the thesis writing period.

Keywords: PLC, HMI, MATLAB, PID control, water level control

List of abbreviations and symbols

FPGA - Field Programmable Gate Array

HMI – Human Machine Interface

HVAC - Heating Ventilating and Air Conditioning

I/O – Input/Output

IP – Internet Protocol

ITTHS - Interacting Two Tank Hybrid System

OEM - Original Equipment Manufacturer

PID – Proportional Integral Derivative

PLC – Programmable Logic Controller

SVG - Scalable Vector Graphics

List of figures

Figure 1. General overview of the Multitank system [20]	22
Figure 2. TK 503 adapter [27]	24
Figure 3. Input/output signals cable connection	27
Figure 4. General block diagram of the RT-DAC/USB2 board [22]	28
Figure 5. Workflow scheme	31
Figure 6. MATLAB/Simulink diagram implementing device communication	32
Figure 7. Input block parameters	33
Figure 8. Output block parameters	34
Figure 9. Input/Output blocks board parameters.	35
Figure 10. Creating a new project	36
Figure 11. Selecting processor module	37
Figure 12. Adding I/O module	37
Figure 13. I/O module parameters	38
Figure 14. DA501 I/O module mapping	38
Figure 15. PLC IP configuration	39
Figure 16. Serial communication parameters	40
Figure 17. HMI IP configuration	41
Figure 18. PID circuit block diagram [23]	42
Figure 19. System flow chart	46
Figure 20. PID function block in PLC	47
Figure 21. PID tuning flow chart [24], [25], [30]	48
Figure 22. HMI main page	49
Figure 23. HMI second page	49
Figure 24. Tank 1 and tank 2 characteristics for 5 cm reference value	51
Figure 25. Tank 1 and tank 2 characteristics for 10 cm reference value	51
Figure 26. Tank1 and tank 2 characteristics for 15 cm reference value	52
Figure 27. Tank 1 and tank 2 characteristics for 20 cm reference value	53
Figure 28. MATLAB/Simulink experimental control diagram	53
Figure 29. Comparison of MATLAB and PLC controllers with 5 cm setpoint	54
Figure 30. Comparison of MATLAB and PLC controllers with 10 cm setpoint	55
Figure 31. Comparison of MATLAB and PLC controllers with 20 cm setpoint	55
Figure 32. Complete equipment setup with Multitank	62

List of tables

Table 1. Operating ranges of Multitank tools [20]	23
Table 2. ABB CP610 HMI technical capabilities [28].....	25
Table 3. Analog inputs specification of I/O board [22]	27
Table 4. Analog output specification of I/O board [22].....	28
Table 5. Digital input/output specification of I/O board [22]	28
Table 6. PID parameters respond to change in the gain [24], [25], [30].....	43
Table 7. Ziegler-Nichols PID tuning method [24], [25], [30]	43
Table 8. Both tanks actual level regarding assigned setpoint value.....	63

1. INTRODUCTION

Overview

As an important parameter in process control, the effect of liquid level on production cannot be ignored. To ensure safety in production and quantity and quality of products, timely and effective liquid level control is essential. Fluid level control of the tanks is a critical requirement in the level control system, and it is expected, practical, and representative in industrial processes.

Multitank systems are commonly used in petroleum and chemical industries to control the liquid level in the tanks. Controlling Multitank is a challenging task since both the flow and pressure of the process can be affected, so it is necessary to reach and maintain the level at the setpoint without too much overshoot and delay. Different approaches can be used to control the fluid level in the tank, but as disturbance enters the frame, a critical issue arises, which can impact the output of the whole system.

The Multitank systems relate to problems generally occurring in industrial storage tanks involving liquid level control. Steel manufacturing industries around the world, for example, have repeatedly reported that significant advantages are obtained from accurate control of the mould level in continuous bloom casting. Oscillations in the mould stage tend to stir extraneous particles and flux powder into molten metal, resulting in surface defects in the final product [1].

Developing a control method for the Multitank system aims to research and check linear and nonlinear control methods in practice. The overall goal of the control is to reach and maintain the level of liquid in the setpoint in upper and middle tanks by adjusting pump power and valve settings. There are various liquid level control methods for solving this control problem range from PID to fuzzy logic and adaptive controls [2], [3]. In this thesis work, the PLC-based PID controller method will be used to develop a control algorithm.

Problem statement

This thesis aims to develop a control system that simultaneously controls and tracks the liquid level of 2 interconnected different shape tanks and ensures that a desired liquid level is achieved and maintained during the operation. This system can be used in industries to prevent overfilling and overflow in the liquid tanks. The system is intended to run by PLC autonomously, so there is no need for human involvement during the

operation process, preventing dangerous industrial accidents that can cause human death.

The intended Multitank system case study is in Control Lab in the ICT building. At the moment, the system controlled an external, PC-based digital controller. The level sensors, valves, and pump communicate with a power interface and dedicated I/O board with the control computer. The intention is to change PC-based control to PLC-based control and use different communication protocol between PLC and sensors, valves, and pumps. For industrial purposes, it is applicable to use an industrial controller rather than a PC because of the reliability, durability, easy integration and long-life cycle advantages which makes the system more secure and trusted.

The Multitank system is prototyped in MATLAB/Simulink environment by the manufacturer. Simulation and experimental results have shown that the developed control method is reliable to control the system at desired setpoint value with visible overshoot for some values. To make the system more reliable, accurate and usable both in the industry and in the educational programme and to meet the new technological requirements, it has been decided to implement a PLC-based PID control algorithm and more applicable communication protocol. Applying a PLC-based PID control algorithm by using a higher-level programming language and tuning control parameters are expected to improve the system stability and accuracy.

The complete system consists of 3 interconnected different shape tanks. The initial stage intends to integrate the ABB controller into the control system, create a communication protocol, develop a control algorithm, design and program a Human Machine Interface panel for remote control. The control algorithm and hardware system will be created for the complete system in the following steps if the initial stage experimental results are reliable enough.

Thesis objective

The main objective of this thesis is to integrate the industrial controller into the Multitank system and develop a control algorithm and user interface panel by considering modern solutions and methods.

The main objectives of this thesis research can be concluded as follows:

- Analysing existing methods, previously developed solutions and literature review.
- Developing communication protocol, control algorithm and Human Machine Interface panel.

- Performing experiments, analysing and comparing results.
- Summarizing the completed work and providing suggestions for future development.

Thesis structure

A short summary of the chapters of this study will be given in this section so that the reader can better understand the content.

The introduction parts give brief information about the goal of this thesis work, the research's problem statement and objective, and a short description of the thesis structure.

Chapter 2 contains a literature review and background analysis of previously developed water level control systems, their area of use and advantages and disadvantages of the existing methods. This chapter also gives information about industrial controllers and their programming languages.

In chapter 3, detailed information is provided about the used hardware and software, their working principles, and technical capabilities for this work.

Chapter 4 contains a detailed explanation of the implementation process of the Multitank system with the controller. This includes communication protocol and control algorithm development and parametrisation process. This chapter also describes the configuration, optimization, and tuning process of the used devices and software and their limitations.

Chapter 5 contains information on how and in which settings experiments were conducted, analysis of the collected experimental results and verification procedure of the developed system accuracy. In this chapter also a brief comparison of newly developed systems with the previously developed system is provided.

In chapter 6, the final work is summarised, and a proposal for further development is provided.

2. LITERATURE REVIEW AND BACKGROUND

2.1 Overview

This chapter will further analyse the existing literature that explains the previous methods closely related to this thesis's objective and use the ideas to describe a potential solution for the liquid level control system. PLC control systems have been widely applied in modern industry in many implementations, such as sun-tracking systems, heating ventilating and air conditioning (HVAC) control, water and wastewater treatment control system, and wind energy system manufacturing photo-voltaic applications and so on. One common characteristic of these applications is that it is possible to model them as process control problems. For instance, the valves and pumps are controlled based on real-time process data in the water and wastewater treatment control system. To be more accurate, a liquid level control problem can be determined as a particular process control problem.

2.2 Existing methods and literature review

In [4], the author proposed a system that provides a study of the equipment and simulation required to design and develop a liquid level control system using a Programmable Logic Controller (PLC). The developed model can efficiently supervise liquid level control in multiple tanks simultaneously and maintain a proper liquid level. To provide PLC with level data, three-level sensors were used. Based on the level sensors data, PLC takes decisions turning ON and OFF a pump. A manual switch is used to change the control mechanism between automatic and manual. As the main decision-making module, the Simatic S7-300 universal controller was used. In this system, modifications can be done by using the model of float sensors that would have the precise level, but the cost would rise, and sensor vibration may interrupt the outcome, effectively countering certain flaws in this model. The main advantage of this proposed system is that it may be used to avoid industrial incidents by overfilling any open container or suppressing overpressure conditions by preventing overfilling of any closed container and controlling the leakage is also possible in this system. Having a large number of the input and outputs port of the PLC will allow this system to control multiple numbers of tanks simultaneously single handily, which make the procedure efficient and cost-effective.

In [5], this author mainly focuses on designing and developing a PLC-based liquid level control system. The project is divided into two parts: the first part is to prepare the

overall mechanical design of the desired system, and the second part is focused on designing and implementing a PLC control system. The finite element analysis for the water tank is done in the mechanical design part to check the areas with a high risk of leakage. In addition, to evaluate the influence of the transient pressure on the sensors, a flow simulation is performed in the water tank. On the other hand, in Simulink, the water tank is modelled, and simulation results have shown that the liquid level can be controlled to the desired position by the PID controller. After designing, performing, and analysing simulation results, in the end, the system is implemented into the real PLC hardware, an experimental test has been done, and the accuracy and stability of the development have been verified.

The author designed and implemented a PLC-based liquid level control system remotely using a radar sensor [6]. The primary purpose of this system is to measure the liquid level, temperature, pressure, volume and remotely control these measurements. The system's main components are a discrete level sensor, Radar sensor, temperature sensor, and a programmable logic controller. For the communication among all the components to transfer data, the author chooses the RS-485 bus with an ASCII-based protocol communication method. The authors note that the communication protocol prevents or removes interference during the data transmission process. The authors designed a Human Machine Interface (HMI) to visually monitor and control the system. The experimental and simulation results verified that the system is accurate, stable, and ready to use.

In [7], a test was carried out on a PLC-based single liquid level control system by using a PID controller. For the communication between hardware, Modbus RTU communication protocol is used to connect an HMI which is programmed on NI-LabVIEW with an Allen Bradley Micro830 PLC. This model is proposed for training purposes to provide a complete understanding of the PLC-based process control system architecture. Some essential components and modelling are introduced and compared in this literature, such as the transducer, the water tank model, and the control. Some necessary parameters, such as the current to pressure (I/P) converter and the resistance of the control valve, are evaluated depending on the previous experimental data by using the least-squares method. The authors decided to apply the PID algorithm into the PLC to achieve a better outcome. Ziegler Nicholas method was used to calculate PID parameters. Finally, the authors concluded that the results matched their expectations regarding the experimental results.

In [8], the author did a preliminary test on a liquid mixing and bottle filling system controlled by a PLC. In this literature, the simulation of the system based on

MATLAB/Simulink and PLC programming are included. The system is divided into three subsystems: filling, mixing, and level monitoring and control and the whole system consist of three liquid tanks, several level sensors and a DC motor. Three-level sensors are used to monitor the liquid level to avoid overflow of the mixed liquid stored in the overhead tank. The sensors are used for tracking low, high and emergency levels, respectively. Furthermore, the authors used a PLC-based PID controller as the control method to achieve a better control result. As a result of using PLC combined with PID controller to control the system, they summarised that PLC-based control system could reduce human intervention and operating cost. At the same time, it can increase production efficiency and driver safety.

In [9], the authors designed a level control system using a PID control algorithm on a programmable logic controller to train engineering students on PID control processes. In their literature, a description of PID parameters tuning is introduced. Also, based on the PID parameters, a ladder diagram programming language and development process of the level control system is explained. The feedback sensor used in this system is a level transmitter. Finally, as the experimental result shown in their paper, the PLC-based PID level control system is stable.

This research's primary purpose was to automatically control the liquid level in a production tank consisting of solenoid valves and pumps using a PLC-based PID controller. The control process level is divided into two phases: the washing and the steady flow production [10]. The author programmed and designed a Supervisory Control and Data Acquit (SCADA) user interface to control the system remotely. The control algorithm was created with function and data blocks with the STEP7 software application and connected to the user interface panel. The controller sends control signals to the actuator, which provides an output with a response signal to the component to be controlled. A Proportional Integral Derivative (PID) controller was used to manage the pump's speed and maintain the water level in each tank according to the reference value. Level sensors current and pump voltage signals need to be scaled for the calculations in the control algorithm. From the experimental tests, it has been concluded that both tanks were filled and maintained at the desired water level and controlled from the operator workstation on a centralised PC. The industrial controller control and monitor the process at each stage of the production phase.

Liquid level control systems have been widely used together with the development of industrial automation. In [11], A liquid level control system is designed and developed using a PID control method based on a PLC. With this system, it is possible to adjust the water level automatically. Liquid height obtained by PLC through sensing circuit and

through automatic adjustment mode determines the PID parameters. Through controlling the operating hours of the water pump, PLC monitors liquid volume. Liquid level acquired by PLC from sensors and automatic adjustment mode regulates the PID parameters. As an important parameter in process control, the effect of liquid level on production cannot be ignored. Effective and timely liquid level control is necessary to ensure safety in production and the quality and quantity of products. In the level control system, fluid level control of the water tank is a crucial issue, and it is typical, practical and representative in industrial processes. In this paper, to cover the shortage of previously generated liquid level control methods, a control system with a PID algorithm is developed based on PLC under the guidance of fuzzy control theory. This system brings accuracy and convenience for control, and it has considerable practical value, particularly in such an efficiency-oriented society.

The multiple tank liquid level control systems were designed and implemented to control liquid levels in industrial and manufacturing processes. In this study, the experimental design of the system was conducted and demonstrated using an ATmega microcontroller chip and FPGA technology in the laboratory [12]. The failure rate of the system and some relevant drawbacks are the main challenges. It involves wireless facilities, power consumption, memory capacity and some others. The proposed system experimentation was conducted to show the comparative performance in industrial system manufacturing of FPGA and ATmega. For the FPGA design, the Xilinx 14.1 ISE was used. The experimentation results obtained from the ModelSim simulator demonstrates reliability and stability of the system that total memory usage of 303192 kilobytes, utilise total processing time of 4,99 s and delay time of 6,557 ns. The main drawbacks of this system are, the intelligent microcontroller (ATmega328) used for industrial applications has a shortage lifetime because of its flexibility hazard rate function is high, inherent weakness, mean-time-to-failure (MTTF) and other environmental conditions (involves temperature, heat and others). Since this system demonstrated various advantages over the microcontroller, it is effective for the industrial control system. It involves low power consumption, effective CPU utilisation, memory capacity, expanded device features for different uses and various wireless technology. Because of the high-reliability prediction, the system has a high resistance of failure rate, gradual failure time characteristics and increased failure time distribution. This has been applied and checked to demonstrate that it reacts successfully and rapidly. This research can be enlarged and developed in industries to control and monitor liquids levels such as diesel, crude oil, and kerosene using fuzzy controller or FPGA.

In [13], the author introduced the advancement of the initial control system of the process control laboratory. The implementation of the liquid level, flow time-sharing and

temperature control experiment platform using Profibus-DP Fieldbus technology of Siemens. The recently implemented control system as the master control unit using S7-300 PLC, S7-200 as the slave control unit, touch screens and PC as upper-level management can carry out real-time monitoring on-the-spot operation, centralised management, adjust control parameters, and provide convenience for experimenters. The practical operating results demonstrate that the platform expands equipment function, increases the equipment utilisation rate, and lets students understand and acknowledge advanced control technologies, such as the research of control algorithms, process control engineering, the implementation of PLC, and touch screen technology.

At the field level, as a high-speed data transmission protocol, Profibus-DP solved the communication task in the high-speed serial bus and distributed field devices in an automatic control system (PLC, PC and so on) [14].

In this research, based on Profibus-DP Fieldbus technology, the original system has been improved. An unrelated single variable control system has been optimised into an ordered process control system [10].

In [15], The author uses two components to implement a solution for incorporating environmental processes into teaching platforms: primary control and communication—the communication based on a bus and other specific protocols. The main control algorithms were developed in MATLAB/SIMULINK environment using Siemens S7-1200 PLC as data acquisition server and some regular PC as a client-server, communicating via the Modbus-TCP protocol. The teaching platform is represented by the two interconnected liquid tank system ASTANK2, with actuators and sensors installed. This work focuses on integrating a multi-tank system (ASTANK2) into the MATLAB/SIMULINK environment. ASTANK2 multitank system manufactured by ASTI Automation and consists of 2 different shape tanks: sloped wall and rectangular [9]. The communication protocol between the PLC (of SIEMENS S7-1200 type) and a portable PC integrating MATLAB/SIMULINK application is based on Modbus-TCP protocol, which is tested by using two conventional PI control algorithms. The actual complexity of the proposed solution lies in the implementation of the communication protocol Modbus-TCP in MATLAB and its synchronisation with the SIMULINK environment. It has been approved that the newly implemented communication protocol solution seems more efficient compared with the standard OPC protocol.

In [16], the author compares adaptive PID control and model predictive control in the conical tank process. The level control becomes quite typical due to the nonlinear shape of the tank. Many process industries use conical tanks because their shape contributes to better drainage of solid mixtures, slurries, and viscous liquids. Therefore, the control

of the conical tank poses a complex problem owing to its nonlinearity and continuously changing cross-section. To control the conical tank system, a conventional PID controller is used. A single run of the relay feedback experiments was carried out to characterise the dynamics, including the type of damping behaviour, the ultimate gain, and the ultimate frequency. We can obtain the tuning parameters using the ZN-Closed loop method from knowing maximum gain and frequency. Model predictive control (MPC) is a computer control algorithm that uses an explicit process model to forecast a plant's future behaviour. MPC was designed for conical tank system for multi set point tracking to achieve the desired performance.

In [17], on a laboratory quadruple-tank system, nonlinear receding horizon control (NMPC) is applied, which is a nonlinear multi-variable process with state and input constraints. Nonlinear moving horizon control mainly referred to as nonlinear model predictive control (NMPC), has become an attractive feedback strategy for nonlinear multivariable systems subject to input and state constraints. At the same time, the application can be found in traditional fields and emerging environments. A fast-numerical algorithm called C/GMRES is employed to implement nonlinear receding horizon control of the quadruple-tank system. The continuation method is combined with a fast algorithm for linear equations instead of the Riccati differential equation. The water level of the quadruple-tank system has been controlled by using the nonlinear receding horizon control method. C/GMRES, a real-time algorithm for nonlinear receding horizon control, has been successfully implemented. Although the objective state's asymptotic stability is not guaranteed, nonlinear receding horizon control leads the state sufficiently close to the objective state, demonstrating the control method's practical applicability. Analysis and simulation results show that the computation time is significantly reduced, and nonlinear receding horizon control can be implemented successfully in real-time.

In [18], the author introduced modelling and controlling of highly nonlinear system by using Honeywell PID and manual PID controllers. In this study, the interaction of the two-tank hybrid system was considered a benchmark problem; the main idea was to control the liquid level of the two-tank system. Because of the variation in the cross-section area, controlling the liquid level in ITTHS is a highly challenging task. The modelling process of these hybrid systems is distinct from the convention modelling techniques. The manual method of tuning PID parameters and Honeywell PID controller has been designed for this process application. The Piecewise linearization technique was applied for linearising nonlinear system output. Different Honeywell PID and manual controllers have been designed for all five regions. The transfer function is found for all the regions using the process reaction curve method, and controller parameters are

tuned using Manual PID and Honeywell PID controllers. The ITTHS simulation was carried out in various operational ranges. The simulation experiments were conducted to ensure that the controller excellently adjusted the desired output volume as per the requirement.

The controlling of the Multitank system used in this project was previously done by another student using different control algorithms and control hardware. In [19], the author mainly focused on developing a visual software application and controlling only a single linear tank by PID controller using Raspberry Pi. In this research, the control method was not implemented for different shape interconnected tanks, which is the primary liquid level control problem commonly occurring in industrial storage tanks.

2.3 Conclusion

Different approaches on liquid level control of multi tanks systems are mentioned in previous paragraphs of the first subchapter. From the conducted literature survey, it can be concluded that the main problems on liquid level control of different shape multi tanks systems are tuning optimal PID controller parameters, controlling the tanks simultaneously and continuously without considerable delay and overshoot due to inequality in the area of the cross-section. While doing research, it occurred that implementing the PLC-based PID controller for both interconnected linear and non-linear systems is not widely spread. Also, selecting correct communication protocol, hardware and software, and developing applicable algorithms to control the system according to the requirements is an essential and challenging task.

3. HARDWARE AND SOFTWARE

3.1 Hardware selection

3.1.1 Overview of the Multitank system

The Multitank System [20], includes three separate interconnected different shape tanks fitted with each other with adjustable electric and manual drainage valves, as shown in Figure 1. An extra water reservoir tank is installed at the bottom part from where the pump takes water to fill the controlled tanks. The upper tank has a constant cross-section, the middle tank has a conical, and the bottom one has a spherical cross-section. The PID controller controls the pump located at the bottom part of the system through the PLC; it pumps the water to the upper tank, the output of the first tank is the input of the middle tank and water outflows from the second tank to the third one. The controllable flow valves act as flow resistors, and it is possible to adjust the flow rate by changing operating voltage, however in this solution, due to the limitation in the hardware side, the adjustable valves are controlled as an on/off valve, which means that it has two-position, fully close and open. Manual drainages are used as controllable drain valves but are manually operated. Manual drain can increase the overall drain rate of the system and influence the system time constant at the controller. A level sensor based on hydraulic pressure measurement is installed in each tank. The Multi-tank system is controlled by an external PC-based controller that uses a specific RT-DAC/PCI card to connect with the level sensors, valves, and pump. The real-time windows target driver, which runs in MATLAB/Simulink, controls the I/O board. To read from and send to data from PLC RT-DAC/USB2 data acquisition card used as a bridge. The driver has four PWM inputs, the first of which controls the pump. The valves of the upper, middle and lower tanks are controlled by the inputs valve1, valve2, and valve3. The driver has three analog outputs: level 1, level 2, and level 3 and one digital output, which is used as an emergency button. The liquid level in the tank is indicated in metric units for each analog output. The emergency has a safety purpose and means whether the tank levels in the system are within the safe limits or not. Frequency signals are converted to metric units within the controller block. Multitank hardware consists of the following devices:

- 3 Tanks made of acrylic glass
- 3 Controlled valves
- 3 Manual valves
- Pump: variable flow, driven by 12 V DC motor
- 3 level sensors, piezoresistive
- Power interface

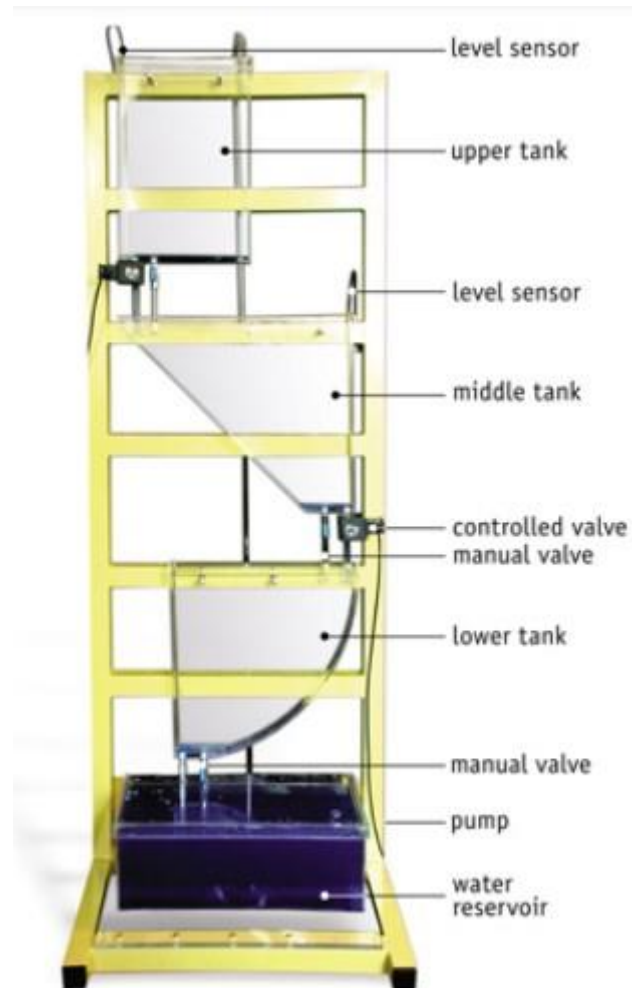


Figure 1. General overview of the Multitank system [20]

The pump value varies from 0 to 1, with 0 representing no duty-cycle and 1 meaning 100% duty-cycle. Due to the height differential between the reservoir tank and the intake of the first tank, water can be pushed with pump values greater than 0.5, which means 50% duty-cycle. The operation value range of the valves is determined by 0.5 and 1, with 0.5 being completely closed and 1 representing completely open. Varying valve settings across operational ranges give variable outflow. Each tank has a 25-centimetre level restriction, and if that limit is exceeded, the alert signal disables the control signal. However, because the value of the level sensors varies significantly, the level in our control model is limited to 20 cm. This makes the system more stable and accurate. As previously stated, if the level exceeds 25 cm, the control block automatically shuts down the system to prevent this action, the limit agreed on 20 cm. Also, by making changes in the control algorithm, a level more than 25 cm is considered an overflow condition. The controller acts differently to prevent overflow rather than shutting down the control system.

Table 1. Operating ranges of Multitank tools [20]

Output devices (%)	Min. value	Max. value
Pump	0.5	1
Valve1	0.5	1
Valve1	0.5	1
Valve3	0.5	1
Input devices (cm)	Min. value	Max. value
Sensor1	0	20
Sensor2	0	20
Sensor3	0	20

3.1.2 ABB AC500 PLC

To control the Multitank system, it has been decided to use ABB PM573 PLC from the ABB AC500 PLC family group because of the technical capabilities and availability in the Alpha laboratory, which creates the advantage of cost and timesaving [26].

AC500 Control Builder provides the following functionalities:

- Five standardised programming languages: Function Block Diagram (FBD), Ladder Diagram (LD), Instruction List (IL), Sequential Function Chart (SFC), Structured Text (ST).
- Debugging functions for the program test: single step, single cycle, breakpoint.
- Free graphical function chart.

Compared with other manufacturers, in ABB AC500 PLC programming environment 61131-3 commands and physical input can be simulated without a real PLC being connected to the hardware, consisting of the applicable malfunctions [26]. After the program is completed and debugged in simulation mode, the program can be downloaded to the actual controller. ABB PM-573 PLC and ABB DA501 input/output module was used to control our system. To program the PLC via personal computer instead of the Ethernet interface, the serial interface via RS-232 cable setup used as the Ethernet port of the PLC was broken, and a COM port is used. Serial communication can be done with RS-232 or RS-485 cable, but rework must be done before connecting to the COM port as ABB PLC does not support standard cables. The order of the pins must be changed as shown in Figure 2. Instead of reworking cables, TK 503 special RS-232/RS-485 to USB adapter can be used, which is designed by ABB for this purpose. Instead of using the TK 503 adapter for this project, it has been decided to rework existing RS232/RS485 cables to avoid wasting time and spending more money.

To fully control the Multitank system, at least 3 analog input and 4 analog output channels were required. However, the DA501 digital/analog I/O module consists of 16 DI, 8 DC, 4 AI, 2 AO. Number of the analog output channels was not enough in the DA501 digital/analog I/O module, therefore, instead of controlling electrical valves as an analog valve, it has been decided to control them as an on/off valve.

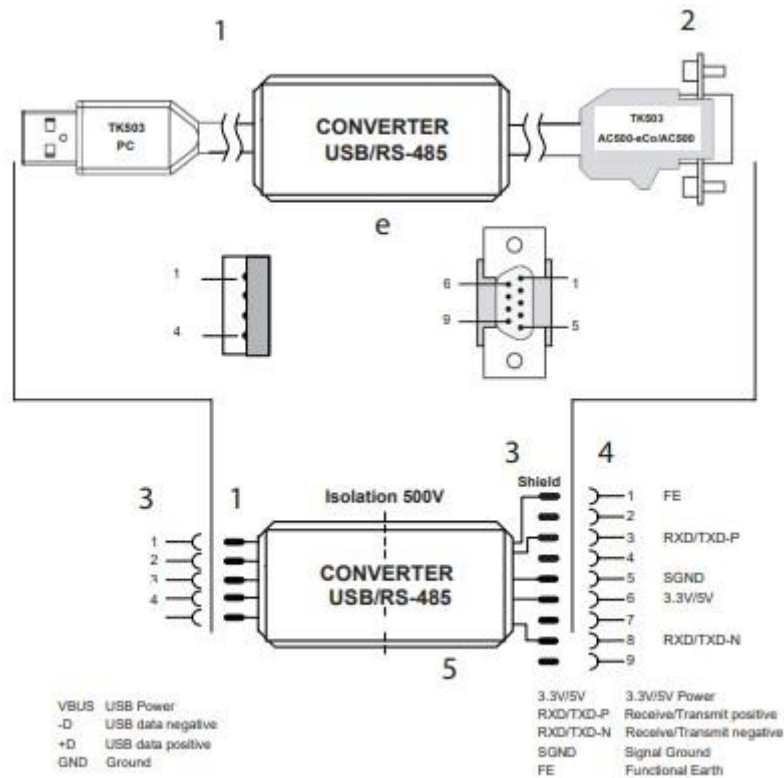


Figure 2. TK 503 adapter [27]

3.1.3 ABB CP610 HMI

The CP600-eCo control panels combine cutting-edge technology and high performance with a stunning design. They were created to provide an exceptional price/performance ratio for demanding applications. The PB610-B software is designed to run on the CP600-eCo control panels. The primary purpose of the HMI in our system is to allow the user to control remotely without interacting with the control system, which is safer and save time. Also, with the help of the HMI, the user can set the tanks setpoint separately, change PID parameters, see the level of each tank in the HMI screen both in visual mode and in values check the system operating time for the previous days, and compare the actual height with the which is measured by the level sensors visually. Because of the technical capabilities, price, and availability in the development laboratory, it is decided to use ABB CP610 HMI to control our control system visually to decrease project compilation cost and time. The technical capabilities of CP610 are listed below [28].

- Works with the PB610 Panel Builder 600.
- Support vector graphics in its entirety. SVG graphic objects, transparency, and alpha blending are all supported natively.
- Control the visibility and transparency of any object on the screen and move, rotate and resize it. Essential and complicated items' characteristics can be changed.
- TrueType fonts are used in multilingual applications. To meet worldwide requirements, easily design, install, and maintain apps in many languages.
- Data can be represented in several ways, including bar graphs, numerical, text, analogue gauges, and visual images.
- Data gathering and logging, trend presentation, alarm handling, scheduler and timed actions (daily and weekly schedules, exception dates), recipes, security and user administration, email, and RSS feeds are just a few of the cutting-edge CP600 capabilities.
- Many communication drivers are available, some of which can communicate with numerous drivers.
- Client-Server capability allows for remote monitoring and control.
- With the PB610-B software, you may simulate both online and offline.
- Scripting language has a lot of power for automating CP600 applications. A good script debugger boosts application development productivity.
- Symbols and objects in vector format.

Table 2. ABB CP610 HMI technical capabilities [28]

Model	CP610
Display / Backlight	TFT Color / LED
Colours	64K
Resolution	1024x600
Diagonal (inches)	10.1" widescreen
User memory	30MB Flash
RAM	512 MB DDR
Serial port	RS-232, RS-485, RS-422 software configuration
Ethernet port	10/100 Mbit
USB port	Host interface V2.0 max 500mA
Real-Time Clock	Yes
Voltage	18-32Vdc
Current rating (at 24VDC)	0.38A

3.1.4 I/O board between PLC and PC

The main idea was to connect all the input and output signals of the Multitank directly to the PLC I/O modules without using a data acquisition card and Simulink for data transmission. Still, because of the higher (200-600 kHz) operating frequency of the level sensors, it has been decided to use RT-DAC/USB2 data acquisition card provided by the Mutitank manufacturer to convert 200-600 kHz to 0-10V for the PLC input. The maximum operating frequency of the ABB AC500 PLC is 50 kHz which is not applicable with the level sensors operating frequency of the target control system. Before deciding to use a data acquisition card between PLC and control system, research has been done, and the leading PLC suppliers are analysed. As a result of the investigation, it became clear that most of the PLC models and their general input/output modules are operating within 0-50 kHz, which is not suitable for this project. For the higher frequencies, the manufacturers offer higher range special input/output modules, which cost more money and delivery time. Therefore, the performance of the RT-DAC/USB2 data acquisition card was tested, and it is decided to be used as a converter and bridge between PLC and the control system.

The RTDAC / USB2 is a multifunctional digital and analog input/output board for real-time data acquisition and control in Windows environments. All boards are built as an OMNI version, which means that the cards can be reconfigured without any hardware changes to introduce new functionalities for all inputs and outputs. The standard configuration of the FPGA chip receives signals from incremental encoders and creates PWM outputs, typical for mechatronic control applications, and is equipped with universal digital inputs/outputs, analog/digital and digital/analog converters, frequency meters, timers, chronometers, and counters.

The RT-DAC/USB2 board is distributed in two versions:

- Analog and digital (RT-DAC/USB2)
- Digital only (RT-DAC/USB2-D)

The Multi-tank system is operated with an external PC-based controller that communicates with the level sensors, valves, and pump by a dedicated RT-DAC/PCI board. The I/O board is controlled by the real-time windows target driver, which operates in MATLAB/SIMULINK. The Simulink interface C-source code S-functions sends a data frame to or read from the RT-DAC/USB2 board.

3.1.5 I/O board specification

For connecting the input/output signal cables, the circuit board is used between PLC and the data acquisition card, which was directly connected to the data acquisition card

through the 40-pin ribbon cable. As seen from Figure 3 below, channels B13 and B15 were used for the analog input signals, channels A1 and A3 were used for analog, A5 used for digital outputs and channel B20 was used as a ground for decreasing the oscillation of the analog signal and ground channels A4 and A6 used for the voltage divider as a ground.

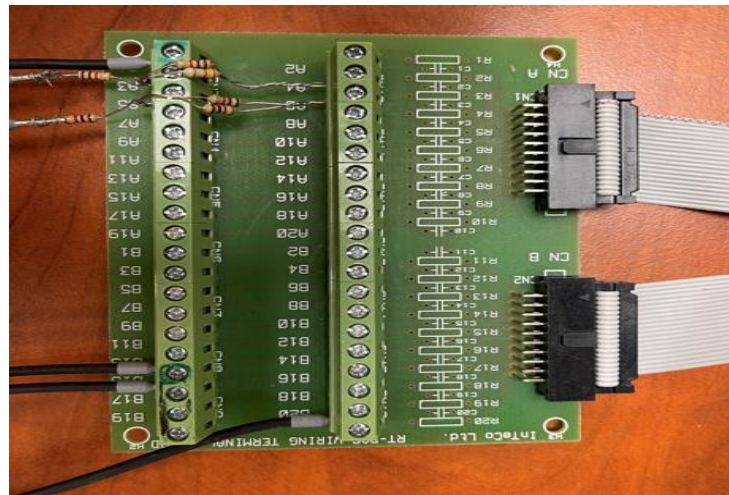


Figure 3. Input/output signals cable connection

Table 3 below provides a general overview of the analog input specification of the I/O board. The table shows that the analog input range is from -10V to +10V. The resolution is 12 bit, and the signal conversion time is 5,4 nanoseconds which allow to send or receive data in higher accuracy with minimum delay.

Table 3. Analog inputs specification of I/O board [22]

Channels	16 single-ended, multiplexed
Resolution	12 bit
Input ranges	$\pm 10V$, programmable gain (x1, x2, x4, x8, x16)
Conversion time	5,4 μs
Trigger	all the A/D channels are scanned automatically when the USB host requires data
Reference voltage	on-board

The I/O board has 4 analog output channels, which supports $\pm 10 V$, $\pm 5 V$ operating ranges with 12/14-bit accuracy [22]. The time delay is 10 nanoseconds which is acceptable for our level sensors. Table 4 below provides a general overview of the analog output specification of the I/O board. However, the I/O board supports both $\pm 10 V$ and $\pm 5 V$ voltage ranges, DA501 I/O module analog input channels support $\pm 10 V$, 4-20 mA and 0-20 mA voltage ranges, therefore, in I/O board $\pm 10 V$ output range has been

decided to use in order to decrease the number of the scaling option to make the signal more accurate.

Table 4. Analog output specification of I/O board [22]

Channels	4
Resolution	12 bit / 14 bit
Output ranges	$\pm 10V, \pm 5V$
Settling time	10 μs (to 0.01%)
Reference voltage	on-board

Table 5 below provides a general overview of the digital input/output specification of the I/O board.

Table 5. Digital input/output specification of I/O board [22]

Channels	26 bi-directional, direction setting; 8 channels shared with PWM outputs; some inputs shared with counter, timer, chronometer, and frequency meter inputs
Direction	bi-directional, individually software programmable
Input voltage	$V_{IH} = 2,0V \div 3,6V, V_{IL} = -0,5V \div 0,8V$
Output voltage	$V_{OH} = 2,4V$ (min), $V_{OL} = 0,4V$ (max)
Output current	2mA per channel
Standard	LVTTL

The block diagram of the RT-DAC/USB2 board is shown in Figure 4 below.

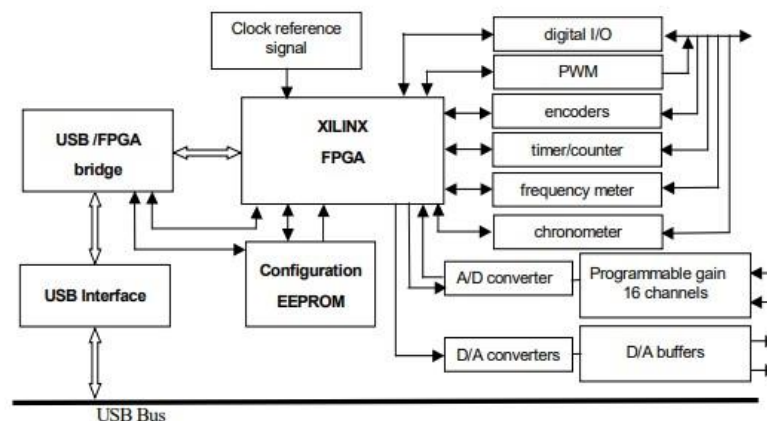


Figure 4. General block diagram of the RT-DAC/USB2 board [22]

The board includes 12-bit successive approximation A/D converters that provide 5 mV resolution across a 10V input range. Gain definition utilizing a digitally controlled analog

amplifier that can provide a more satisfactory resolution. The RT DAC/USB2 board's A/D conversion time is 5,4 nanoseconds. Four 12-bit D/A converters are connected to four analog output channels on the board (14-bit D/A converters are optional). The channels' output voltage is around 10V. Each analog output channel has a maximum current sink of 10 mA [22].

3.2 Software selection

3.2.1 Automation Builder

Automation Builder is the integrated software package that combines the tools necessary to program, configure, maintain, and debug automation projects through a familiar, intuitive interface. Automation Builder connects the engineering tools for PLC, safety, control panels, SCADA, Drives and motion [29].

ABB Automation Builder is a comprehensive software designed for machine builders and system integrators who want to automate their machines and systems cost-effectively and contains the tools required for configuring, programming, debugging and maintaining automation projects from a familiar, intuitive interface. Automation Builder addresses the most significant single cost element of most of today's industrial automation projects: software. ABB Automation Builder covers the engineering of ABB PLCs, Safety PLCs, control panels, drives, motion and robots.

- Three editions of Automation builder are available: Basic (free), Standard and Premium, which are suitable for both small projects and managing the large projects for system integrators and OEMs.
- With the incorporated 30-day free test license, Automation Builder can be used free.

In this thesis work, it has been decided to use the Basic version of the software as all of the capabilities meet the work requirements for successfully developing the control algorithm and HMI interface panel.

3.2.2 CODESYS

CODESYS (Controller Development System) is an IEC61131-3 Engineering Software used by over 400 PLC manufacturers for their controllers [29]. CODESYS is an open-source PLC programming software developed by a German software company by 3S – Smart Software Solutions. ABB Automation Builder is based on this established Software. It offers users practice-oriented, integrated solutions for the convenient

configuration of automation applications. Open interfaces, comprehensive security features, and a convenient connection to a cloud-based administration platform make CODESYS a natural Industry 4.0 platform. End-users employ CODESYS to successfully create sophisticated and straightforward automation applications for industrial controllers used in factories, mobile machines, power and building automation systems, and many other areas. For the controlling of my control system, CODESYS was also used because of the mentioned reasons. The free license and user-friendly interface also allow the user to create control algorithms and human-machine interfaces in a single software.

CODESYS support five different PLC programming languages according to the international industrial standard IEC61131-3. These languages are:

- Instruction List (IL)
- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Sequential Function Chart (SFC)

In this thesis, it has been decided to write a control program in Structured Text (ST) languages as it syntactically looks like high-level programming languages, for example, C and C++. Compared with the other programming languages in the IEC61131-1 standard, the flow and logic of the program written in structured text is easier to read and understand and capture less space.

4. DEVELOPMENT

4.1 Work schematics diagram

The control algorithm is developed on ABB PM573 PLC, which is connected to the laboratory computer by RT-DAC/USB2 data acquisition card. Because of the hardware limitation and sensors higher frequency range, it is decided to use a laboratory computer with RT-DAC/USB2 data acquisition card as a bridge between the control system and PLC. The generated control signals send to the laboratory computer by PLC, where the signals are processed in the MATLAB/Simulink environment. As a transport layer protocol, both PLC and lab computer use UDP data packets for transferring signals. The primary mission of the laboratory computer is to receive and send control signals from the PLC and receive and send data through a data acquisition card to the control system. The workflow schematic diagram can be seen in Figure 5.

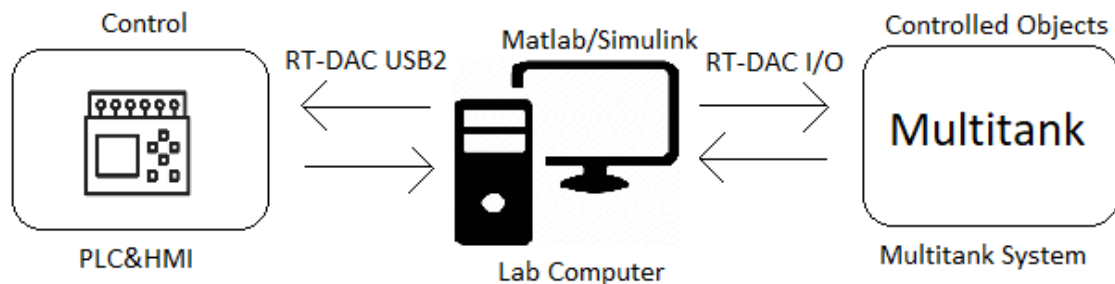


Figure 5. Workflow scheme

4.2 MATLAB diagram

As mentioned in the previous paragraphs, the connection between the control system and PLC was made over MATLAB/Simulink environment to transfer the data. Designing a MATLAB model and using it as a bridge was one of the most challenging tasks faced during the research. Extra knowledge and time were needed to create a stable and accurate data transformation environment. The primary task of the MATLAB model is to develop a real-time environment data transformation environment without data interruption during the real-time controlling. Sensors raw value calibration and scaling are also done over MATLAB model to transfer more accurate data to PLC. Before calibration, the actual value of the level 1 and level 2 sensors while the tanks were empty corresponded to 1 and 1,5 cm in each tank, respectively. By doing the mathematical calibration, it has been succeeded to decrease the error to 0,1 – 0,2 cm

in each tank. Scaling the sensors output voltage was one of the main tasks for accurately controlling the control system over PLC. Before doing a scaling for the sensors output voltage, the values range from -10V to +10V, which cannot be used as an analog input in PLC I/O module as analog input channel receives input voltage between 0V and 10V where 0V corresponds to 0 bit (0 cm in the target control system) and 10V corresponds to 27648 bit (20 cm in the target control system). At the same time, using a scaling option -10V to +10V output voltage of the sensors scaled to 0V to +10V input voltage for the analog input module of the PLC. As seen from the figure below, for the electrical devices, saturation block is used to define the operating ranges of both actuators and pump and prevent voltage spark. MATLAB/Simulink diagram of the whole system can be seen in Figure 6 below.

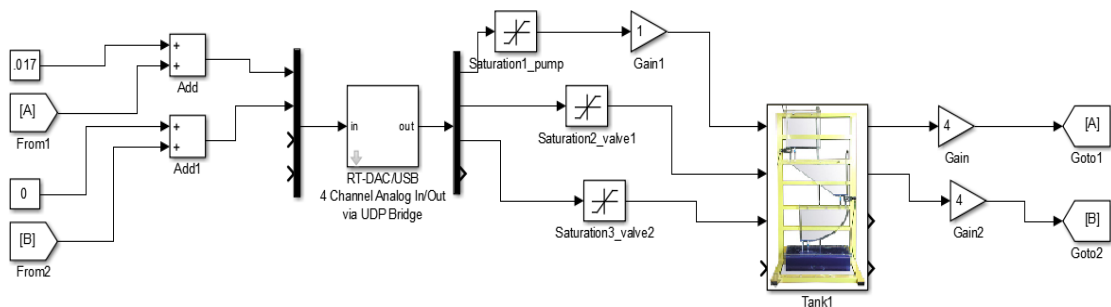


Figure 6. MATLAB/Simulink diagram implementing device communication

4.3 MATLAB network settings

The most common data type is double. The double-precision floating-point format is an 8-byte or 64-bit computer number format. The sender block and receiver block are the essential details in the diagram. The receiver must accept 32 bytes of data, with every 8 bytes being in charge of its own level. In this thesis, because of not using all three level sensors, only level 1 and level 2 values are needed; therefore, instead of 32 bytes, the receiver will accept 24 bytes. Inside the input/output packet, the sender and receiver block is separated, parametrisation of both blocks needs to be completed before activating real-time run time mode for data transmission. Figures 7 and 8 below the main parameters of both input and output block can be seen, respectively.

For double-precision floating-point variables, the following serialisation is used:

- First 8 bytes are responsible for motor value.
- Second 8 bytes are responsible for valve 1 value.
- Third 8 bytes are responsible for valve 2 value.

The specified output block has the same parameters as the input block, but instead of 32 bytes, our system requires 12 bytes.

- First, 8 bytes are responsible for level1 output value.
- Second 8 bytes are responsible for level2 output value.

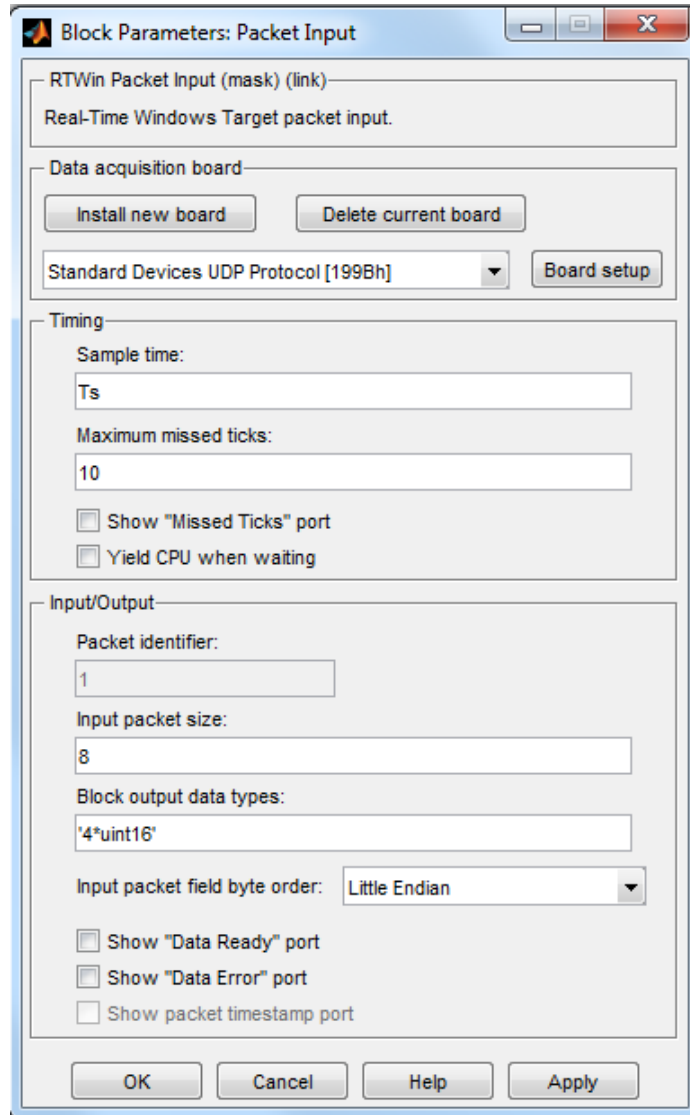


Figure 7. Input block parameters

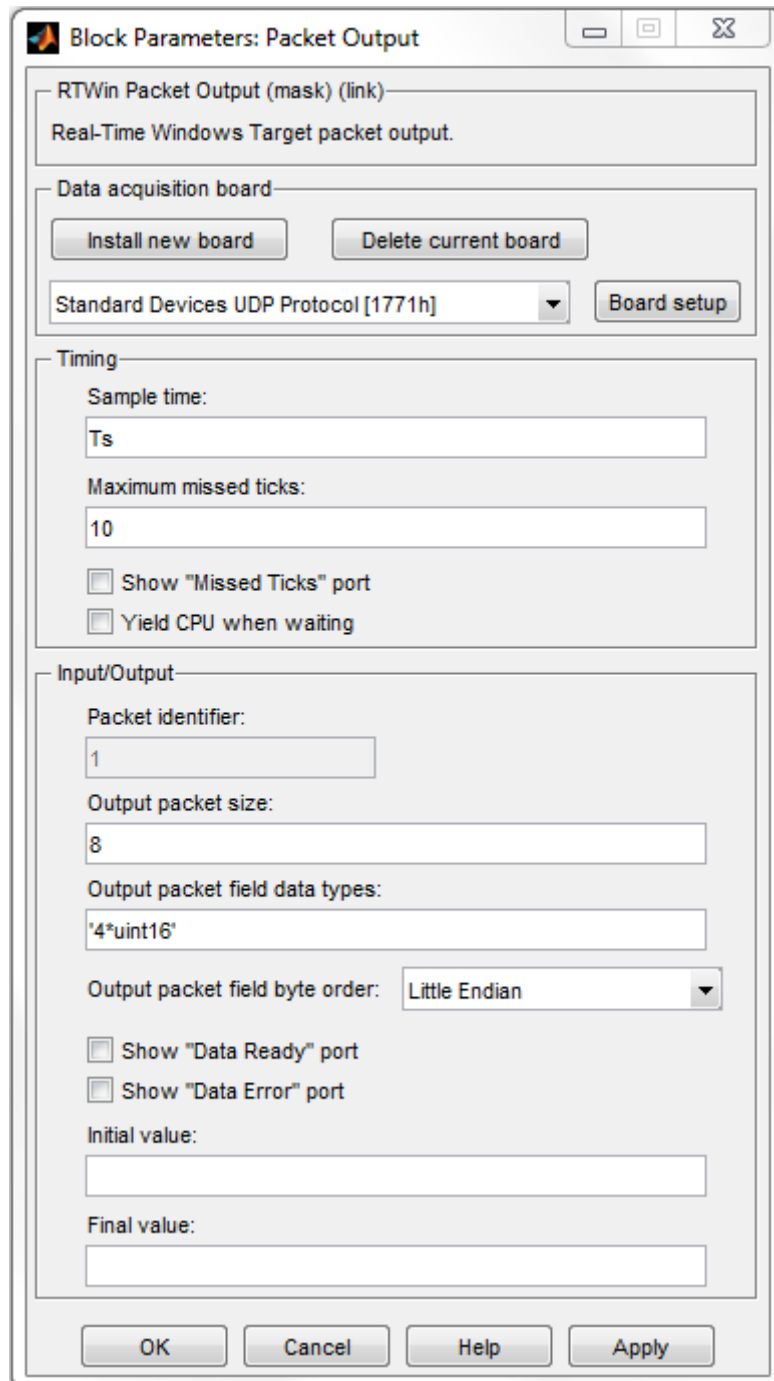


Figure 8. Output block parameters

To create continuous and stable communication for data transmission, it is also required to set board IP addresses and local and remote virtual port parameters. The parameters for the input packet used in our system can be seen in Figure 9 below. Local UDP port number and IP address are the same for the output packet, the only difference is that in the output packet for the remote UDP 5666 port is used.

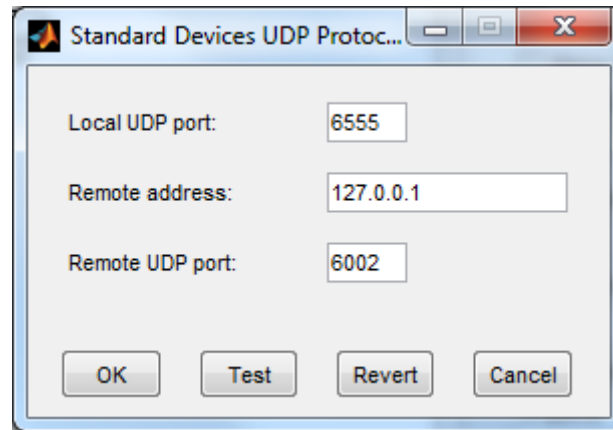


Figure 9. Input/Output blocks board parameters.

4.4 Voltage divider

As mentioned in the previous paragraph, electrical valves are controlled by digital output signals because of the shortage of analog output channels in the DA501 module. The output voltage of the DO channel is 24V which is 2 times higher from the operating range of the RT-DAC/USB2 input channel. The maximum input voltage of the data acquisition card is 10V. Therefore, a voltage divider was used to drop 24V to around 10V for the card. After implementing the voltage divider in the input channel of the data acquisition card, the 24V digital output voltage of the PLC corresponds to the 10V digital input voltage in the data acquisition card. According to the specification of the I/O card, the voltage over 2,4V corresponds to digital input high. While designing the voltage divider resistors, tolerance and load resistance were considered to prevent the I/O card from over-voltage damage from PLC output. It was decided to limit the input voltage to 8V for the I/O card input. Mathematical calculation of the voltage divider describes in 1 as:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2} \quad (1)$$

Where:

V_{out} - Output voltage. This is the scaled-down voltage.

V_{in} - Input voltage.

R_1 - Resistor 1.

R_2 - Resistor 2.

The voltage divider is designed and tested in an online electronic circuit simulator tool before implementing hardware.

4.5 Automation Builder configuration

In Automation Builder configuration, start by creating a new project. For this purpose, from a toolbar in the upper-left corner, File -> New Project -> AC500 project must be selected, see Figure 12.

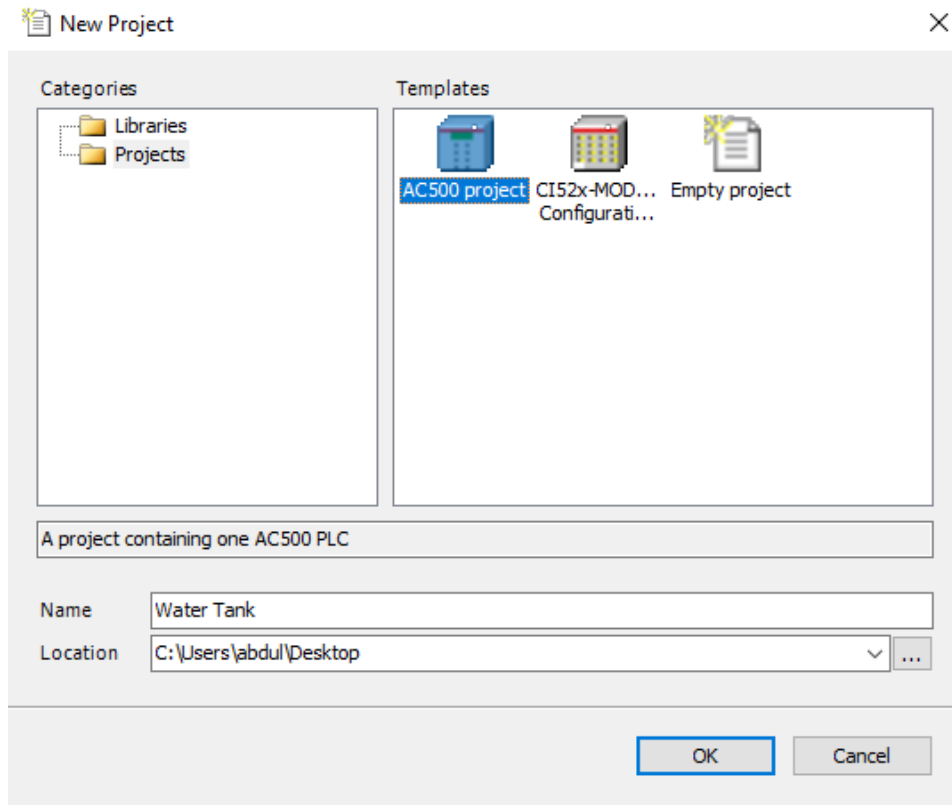


Figure 10. Creating a new project

Next, the intended PLC processor module must be selected from the opening pop-up windows, see Figure 13.

After selecting the processor module, the next step is to add the I/O module to the rack. This step can be done from the project tree by right-clicking on I/O Bus and selecting the add object, see Figure 14.

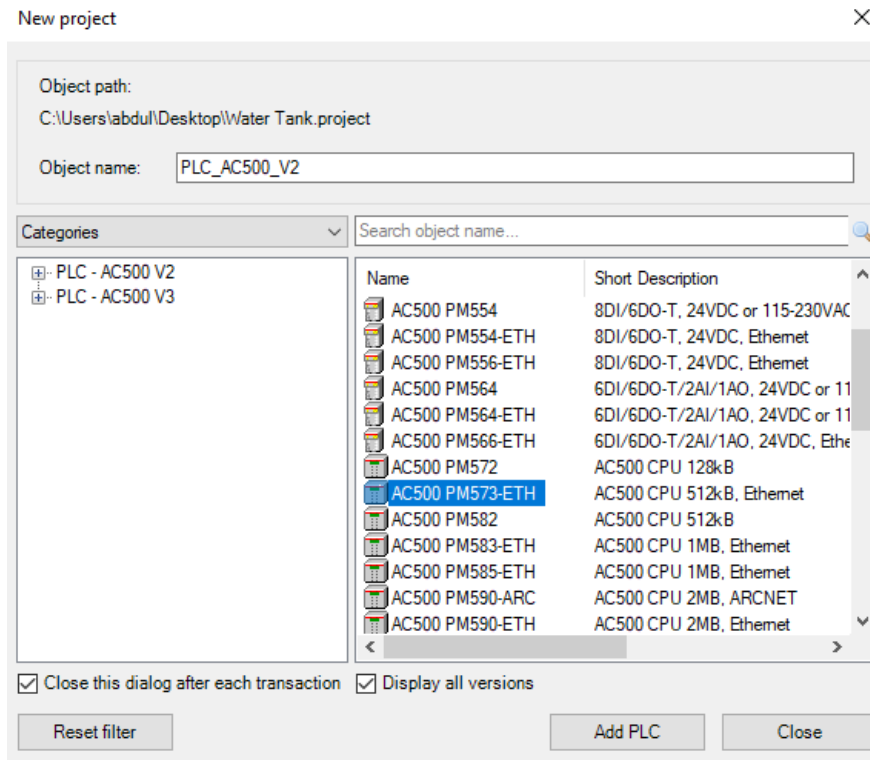


Figure 11. Selecting processor module

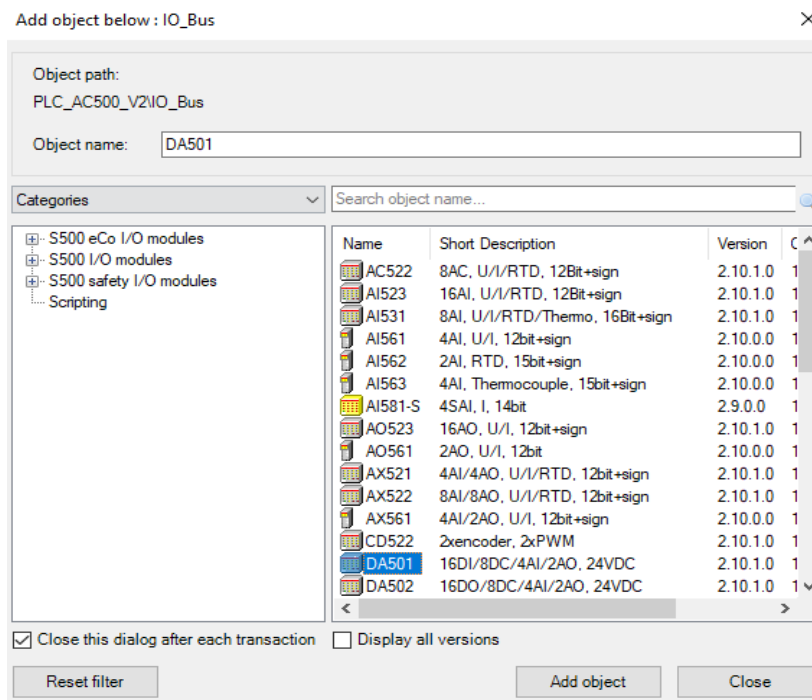


Figure 12. Adding I/O module

4.6 I/O module configuration

After completing the I/O module adding step, the next step is to configure input/ output channels parameters and mapping. To configure the system for specific tasks, setting up I/O module parameters is required, see Figure 15.

ameter	Type	Value
Behaviour outputs at comm. error	Enumeration of BYTE	Off
Substitute value	BYTE(0..255)	0
Input 0, channel configuration	Enumeration of BYTE	0...10 V
Input 0, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Input 1, channel configuration	Enumeration of BYTE	0...10 V
Input 1, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Input 2, channel configuration	Enumeration of BYTE	Not used
Input 2, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Input 3, channel configuration	Enumeration of BYTE	Not used
Input 3, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Output 0, channel configuration	Enumeration of BYTE	-10...+10 V
Output 0, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Output 0, substitute value	WORD(0..65535)	0
Output 1, channel configuration	Enumeration of BYTE	Not used
Output 1, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit
Output 1, substitute value	WORD(0..65535)	0

Figure 13. I/O module parameters

The I/O channels that need to be accessible as global variables must be mapped in the DA501 I/O mapping section, as shown in figure 16. I/O mapping can be done by double-clicking on the I/O mapping list and declaring the variable names for the required variables. In Figure 16, the list of declared variables, their types and addresses are shown.

DA501 Parameters	Find	Filter	Show all		
DA501 I/O Mapping	Variable	Mapping	Channel	Address	Type
I/O mapping list			Digital inputs DI0 - DI15	%IW0	WORD
Information	Level_1		Analog input AI0+	%IW1	INT
	Level_2		Analog input AI1+	%IW2	INT
			Analog input AI2+	%IW3	INT
			Analog input AI3+	%IW4	INT
	Pump		Analog output AO0+	%QW0	INT
			Analog output AO1+	%QW1	INT
			Digital inputs DC16 - DC23	%IW5	WORD
			Digital outputs DC16 - DC23	%QW3	WORD
	Valve_1		Digital output DC16	%QX6.0	BOOL
	Valve_2		Digital output DC17	%QX6.1	BOOL
			Digital output DC18	%QX6.2	BOOL
			Digital output DC19	%QX6.3	BOOL
			Digital output DC20	%QX6.4	BOOL

Figure 14. DA501 I/O module mapping

4.7 PLC and HMI communication configuration

ABB PM573 PLC module supports both serial and Ethernet communication interfaces. In this thesis, both serial and Ethernet communication protocols are used for connection between PLC and PC.

4.7.1 Ethernet communication

An IP address must be configured to connect the PLC and the PC using the Ethernet protocol. Select Tools -> IP-Configuration from the upper toolbar and click Scan to explore the IP address of a connected device. After that, specify the IP address by right-clicking on the programmable device in the project tree and selecting Communication Settings, see Figure 17. Because of security purposes, it is necessary to assign the same subnet mask for both PLC and PC. If the subnet mask and IP address assigning process are completed, the program can be downloaded to PLC, and the program can be run by pressing the Run button on PLC hardware or by clicking the Online -> Login -> Run button on the software.

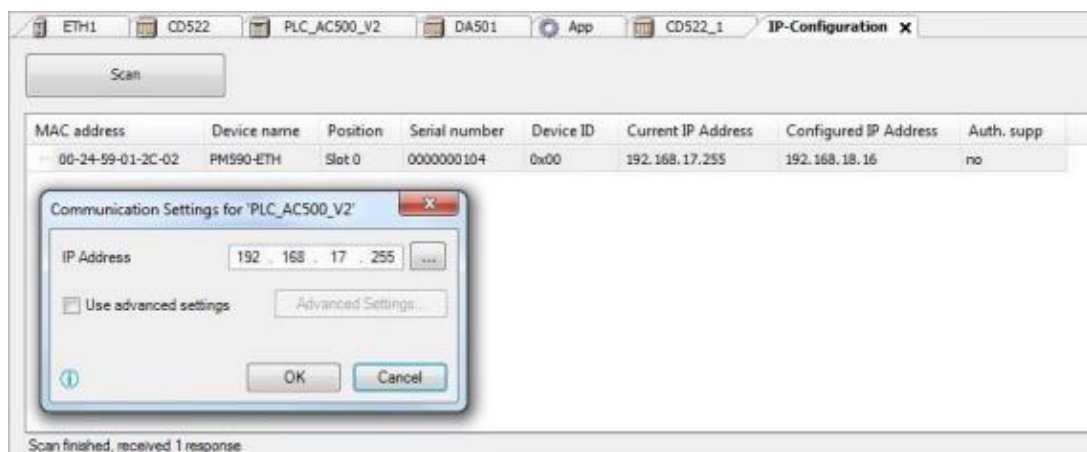


Figure 15. PLC IP configuration

4.7.2 Serial communication

While working on this thesis, serial communication was also used for connecting PC with PLC. To create communication between PLC and PC via serial communication, a PLC COM2 port must be used, and we need to know the Com port number used in the PC. To find which port is used for USB-to-Serial in PC from Device Manager, check the Ports (COM<P). After completing that step PLC communication parameters also need to configure, therefore, from the menu toolbar, click on the Online -> Communication Parameters button, then click on Add new connection and set the following parameters:

- Choose Serial (RS232).
- Name it: ComN_PC, where N is the number of COM port.
- Set the correct parameters for communication via COM2, see Figure 18.

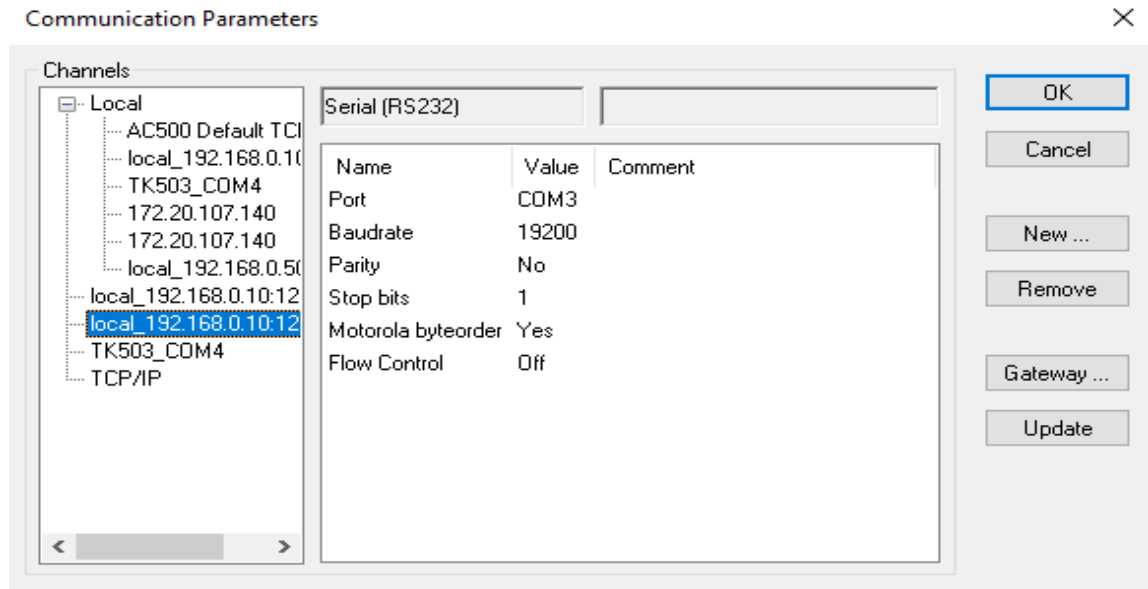


Figure 16. Serial communication parameters

4.7.3 HMI configuration

To connect ABB AC500 PLC with ABB CP610 HMI, the integrated Ethernet port-based TCP/IP interface is used on both devices, and the protocol ABB CODESYS Ethernet is used to communicate between the panel and the PLC. It is essential to bring PLC and panel IP addresses to the same IP group, including the subnet masks. For the PLC, the IP address can be configured as mentioned in the previous paragraph. Still, for the CP610 panel, the appropriate IP address must be configured directly from panel system settings in hardware or web page, as shown in Figure 19.

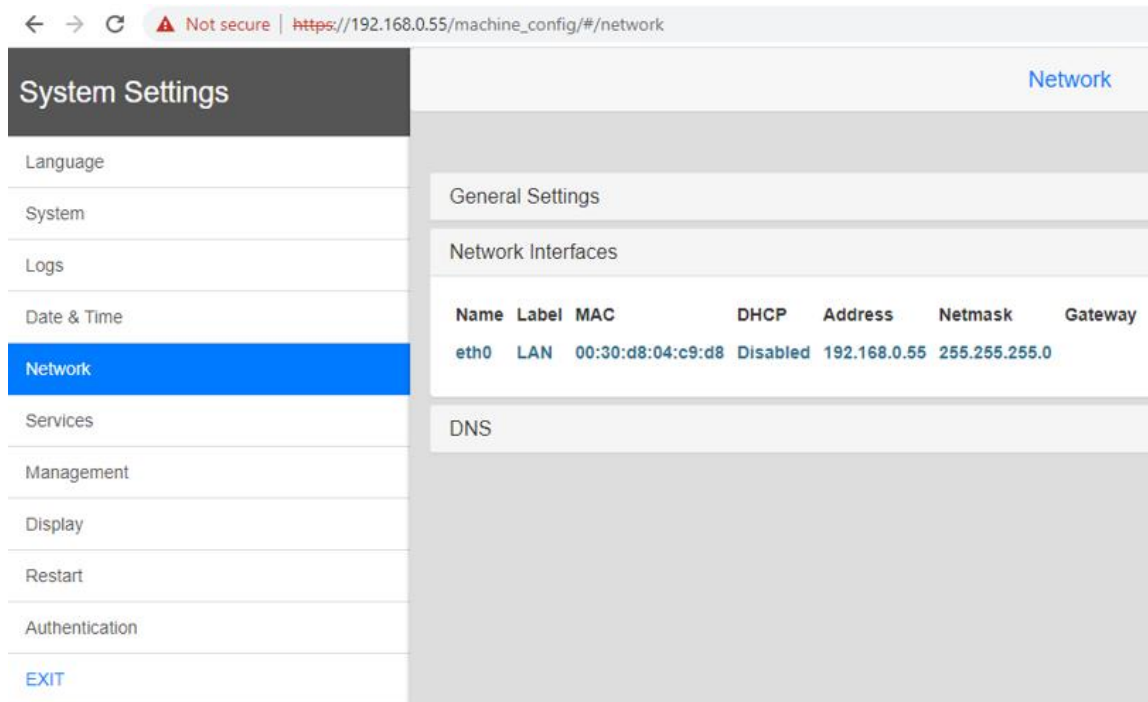


Figure 17. HMI IP configuration

When the IP address is configured for the panel, it is possible to access the HMI settings through the web browser via the link, https://192.168.0.55/machine_config, where 192.168.0.55 is the IP address of the HMI used in this project.

4.8 PID configuration

PID circuits are commonly used as closed-loop feedback controllers and are found in various servo circuits. Proportional (P), Integral (I), and Derivative (D) are the three control settings of the PID circuit, and their letters form the acronym PID. The purpose of all servo circuits is to keep the system at a reference value (setpoint) for an extended period of time. The PID circuit actively manages the system by generating an error signal, which is the difference between the reference value and the current value, to maintain the setpoint value. The three controls are related to the time-dependent error signal, which can be summarised as follows: proportional is based on current error, integral is based on historical error accumulation, and derivative is based on future error prediction. After that, the outputs of each of the controllers are input into a weighted total, which changes the circuit's output, $u(t)$. A control device receives this output, which then feeds its value back into the circuit. It allows the process to actively stabilise the circuit's output to reach and maintain the setpoint value. The activity of a PID circuit is illustrated in the block diagram in Figure 20 below. Depending on the system demand and necessity, one or more controls can be used in any servo circuit (for example, P, I,

PI, PD, or PID). A relatively speedy response with slight overshoot (passing the setpoint value) and ringing (oscillation around the setpoint value) can be obtained by adequately adjusting the controls in a PID circuit [23], [30].

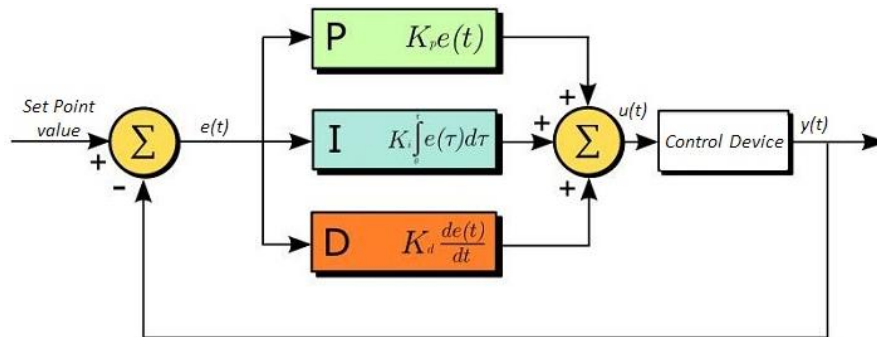


Figure 18. PID circuit block diagram [23]

The output of the PID control circuit, $u(t)$, is given as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2)$$

$$e(t) = SP - PV(t) \quad (3)$$

Where:

K_p - Proportional gain

K_i - Integral gain

K_d - Derivative gain

de - change in the error value

dt - change in time

$e(t)$ - error value

SP - Set Point value

PV - Process value

Larger proportional gains result in more significant changes in reaction to the error, which reduces the controller's ability to respond to system changes. While a high proportional gain might make a circuit respond quickly, a too high number can generate overshoot and oscillations around the SP value. If the proportional gain is too low, the controller will be unable to respond to changes in the system effectively [23], [30].

Integral control effectively increases a circuit's response time while eliminating the steady-state error that strictly proportional gain causes. High gain values can generate severe overshoot of the SP value, resulting in oscillation and instability due to the quick reaction of integral control. With the low integral gain, the circuit will be slower to respond to changes in the system [23], [30].

In contrast to proportional and integral control, derivative control slows the circuit's reaction. It can partially compensate for overshoot and dampen any oscillations induced by integral and proportional control by doing so. The circuit responds slowly when the gain is set too high. If the derivative gain is too low, the circuit will overshoot the SP value [23], [30]. Table 6 below illustrates the consequences of separately raising the gain of each parameter.

Table 6. PID parameters respond to change in the gain [24], [25], [30]

Parameter Increased	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small Change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Decrease Significantly	Degrade
K_d	Minor Decrease	Minor Decrease	Minor Decrease	No effect	Improve (for small K_d)

For tuning the PID controller in this research, Ziegler-Nichols tuning method used and these steps are followed: Integral and derivative gains are settled to zero before manually adjusting the PID controller and increased the proportional gain until output oscillations are visible. The proportionate gain should then be set to around half of this value. Integrated gain must be raised after adjusting the proportional gain until the offset is adjusted on a timeline that works for the system. If this gain is increased too much, the SP value will overshoot, and the circuit will become unstable. The derivative gain can be increased after the integral gain has been established. The differential gain reduces overshoot and brings the system back to SP values rapidly. Significant overshoot would be observed in the system if derivative gain increased too much (because the circuit reacts too slowly). The performance of the PID controller can be improved by adjusting the gain settings, resulting in a circuit that responds fast to changes in the system and efficiently reduces vibrations around the SP value. The Ziegler-Nichols method for PID tuning offers a bit more structured guide to setting PID values in Table 7 below.

Table 7. Ziegler-Nichols PID tuning method [24], [25], [30]

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$0.54K_u/T_u$	-
PID	$0.60K_u$	$1.2K_u/T_u$	$0.075K_uT_u$

Where:

K_p – Proportional gain

K_i – Integral gain

K_d – Derivative gain

K_u – Ultimate gain

T_u – Oscillation period

4.9 Control program development

Control program is developed on the CODESYS programming environment by using ABB PM 573 PLC and DA501 I/O module. The program development can be started after completing hardware and configuration on Automation Builder software by double-clicking the *Application* button from the project tree. Before writing the control algorithm, the program structure flowchart was created to make system control logic understandable and future development easy. While writing a control program, simplicity, readability and accuracy criteria were considered as a main requirement.

4.9.1 Program structure

The process starts by switching the PLC to run state. Fieldbuses need to be checked for the data communication as the bridge is used between the hardware and the PLC, if the data acquisition card is switched off or stopped transferring data, then program execution can be prevented. Because of safety reasons, emergency (E-stop-B), stop (Stop-B) and start (Start-B) need to be checked by PLC before sending or receiving any command or data. First Emergency button needs to be checked; the button needs to be in the released position; if the emergency button is pressed, then PLC automatically ends program execution. The stop push button also needs to be released. Otherwise, the PID controller will not start execution till the stop button is released. To start program execution, the start button needs to be pressed. Otherwise, program execution will not start. As a next step, PLC receives the water level height of the tanks through the level 1 and level 2 sensors and checks the overflow condition. Both middle and upper tanks can be filled max. 25 cm and in our logic 25 cm is overflow limit, height more than 25 cm in both tanks or only in one tank considered as overflow condition and following steps need to be executed to prevent flooding. The program first checks the water level in the upper tank; if the water level in the upper tank is more than OF_L (overflow limit), then the program will check the water height of the middle tank and if the height of the middle tank is also more than maximum limit then program must calculate the volume of the excess water volume of both tanks and the exceeded amount

of water of both tanks must be drained from middle tank by opening valve 2 (V2). After the level on the middle tank reached the calculated height, valve 1 (V1) will be opened to empty the upper tank and maintain the level in limit. Suppose the water level of the upper tank is in range and the program does not detect overflow conditions. In that case, the algorithm will only check the status of the middle tank, and in case of an overflow condition, valve 2 will be opened, and height will be maintained in limit. If no overflow is detected in any tank, then the PID controller (PID_C_ON) starts operation and checks the actual value of the middle tank (SP2). To reach the setpoint in the middle tank, valve 2 must be closed and, valve 1 must be opened as the output of the upper tank is the middle tank's input. In case of any condition, if the level (L) of the middle is more than the setpoint, then valve 1 must close immediately, and valve 2 must be opened by PLC to decrease the level to the setpoint until the level in the second tank reaches the target. Both valve 1 and valve 2 must be closed if the setpoint value maintained in tank 2. In the following step, the PID controller checks the water level in the upper tank, and if the height is not in the setpoint, the pump operates until reaching the setpoint. When the desired level is obtained in the upper tank, the program end execution; if it is found out that the level of the upper tank is over setpoint 1 (SP1), then valve 1 must be open. The excess amount of water must be poured into tank 2 to maintain the level in reference value. If the excess amount of water runs to tank 2, the level of tank 2 will be more than the setpoint. In this condition, valve 2 must be opened automatically to reach the setpoint. When the desired setpoint is obtained on both tanks, then the program stops execution. System complete operation logic can be seen from the flow chart below in Figure 21.

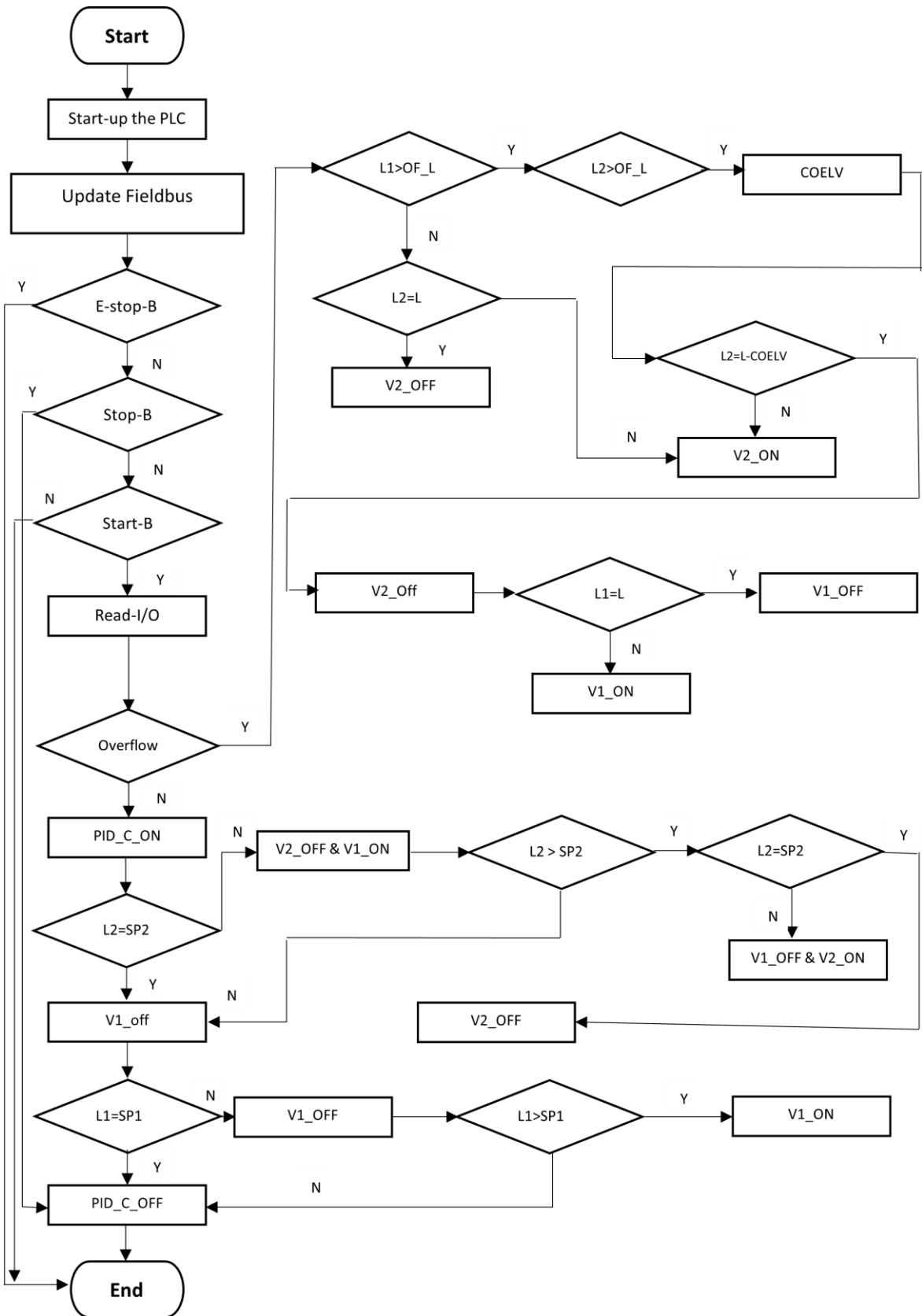


Figure 19. System flow chart

4.9.2 PID function block

The Multitank system is controlled in fully automatic mode, which enables to control the system more accurately and reach a setpoint in less time period with the help of PID controller function block, see Figure 22. PID function block allows to improve system accuracy, maintain higher operating life cycle for the controlled devices, decrease the level oscillation and overshooting.

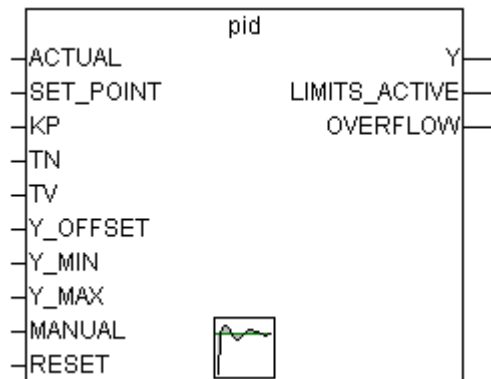


Figure 20. PID function block in PLC

ACTUAL is the real-time value of the tank level. *SET_POINT* is the targeted value of the water level in the tank. *KP* is a proportionality constant (P component), *TN* is a reset time in second (I component), *TV* is a rate time, derivative time, D in second. If *TV* is set to 0, then the PID controller act as PI controller. *Y_MIN* and *Y_MAX* is the minimum and maximum limit of the output value. *Y* is the value of output which is calculated by the function block according to the input value. For tuning the PID controller, the Ziegler-Nichols method was used (Table 7). Several experiments were conducted to increase the PID controller's performance, and average PID parameters were found for all the setpoints. The tuning starts by setting the integral and derivative gains to zero and increasing the proportional gain until the system is unstable. Then the proportional gain should be set to around half of this value according to the used method. After adjusting the proportional gain, the integral gain should be changed, if the integral gain increased too much, the reference value would overshoot, and the control system will become unstable. To reduce overshoot derivative gain increased and SP value comes back rapidly. It has been observed that when increasing the derivative value too much, the overshoot is too much because the system reacts too slowly. After adjusting the PID gain settings according to the Ziegler-Nichols method, to improve system performance for fast responding and less oscillation around the SP values, the gains changed a little by performing experimental tests on different set point levels. PID tuning flow chart diagram according to Ziegler-Nichols method can be seen in Figure 23 below.

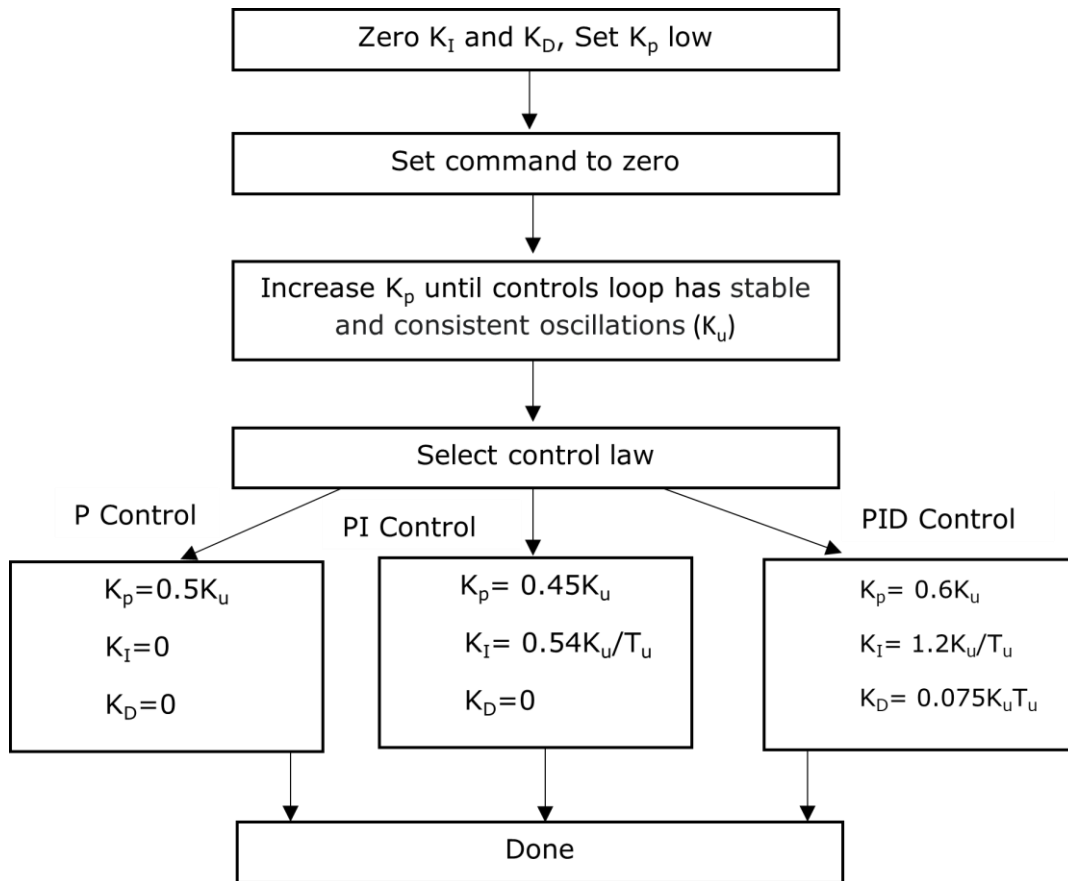


Figure 21. PID tuning flow chart [24], [25], [30]

Where:

K_u - Ultimate gain

K_p - Proportional gain

K_I - Integral gain

K_D - Derivative gain

T_u - Oscillation period

4.10 HMI development

For developing the HMI panel, Panel Builder 600 software and ABB CP610 hardware are used. After the HMI interface is developed and implemented through PC, then the PLC can be connected with HMI directly and assigning tanks setpoints, monitoring the level of both tanks, pump and valves status, overflow and emergency conditions can be done remotely through the HMI interface by the user. Three HMI screens were developed; the primary duty of the home screen is to allow the user to assign setpoints and check actual levels in both tanks, start and stop operation and press the emergency stop button in case of need. The valves and pump status can also be observed on the home

screen, shown in Figure 24. In the second and third screens, the user can check the real-time and historical trends of the upper and middle tanks, Figure 25. The data storing function of the HMI is activated, and by using an external hard disk, the data can be stored for later use, in case of any failure or energy loss, the graphs data will not be lost. The internal storage space is not large enough; the history data can be stored maximum up to 1 day without external hard drive. In case want to store the data for a long time then external hard drive need to be connected to the HMI from the back side, which is already activated, and it will start to save data immediately USB drive plugged in. In this work, the data storage option is activated to compare MATLAB and PLC PID controllers.

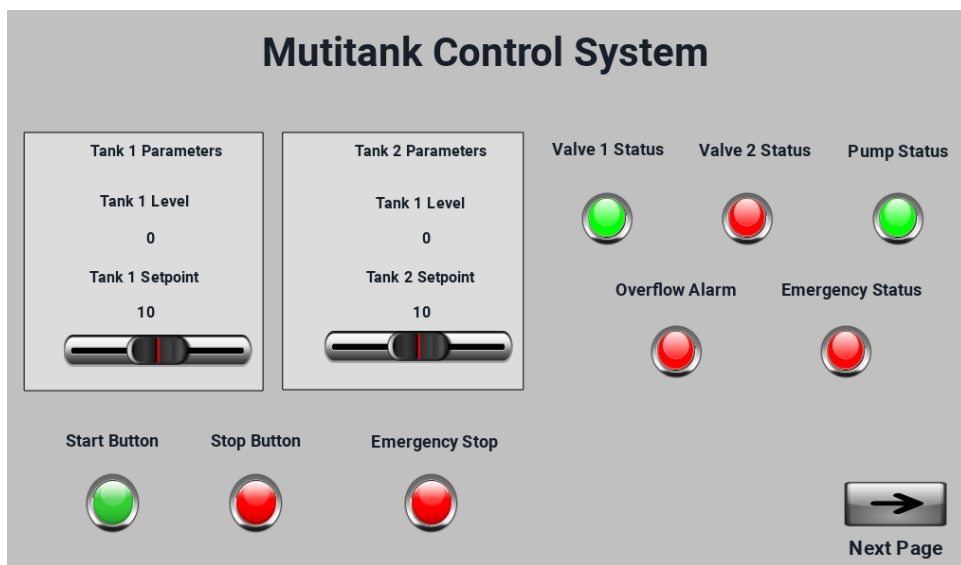


Figure 22. HMI main page

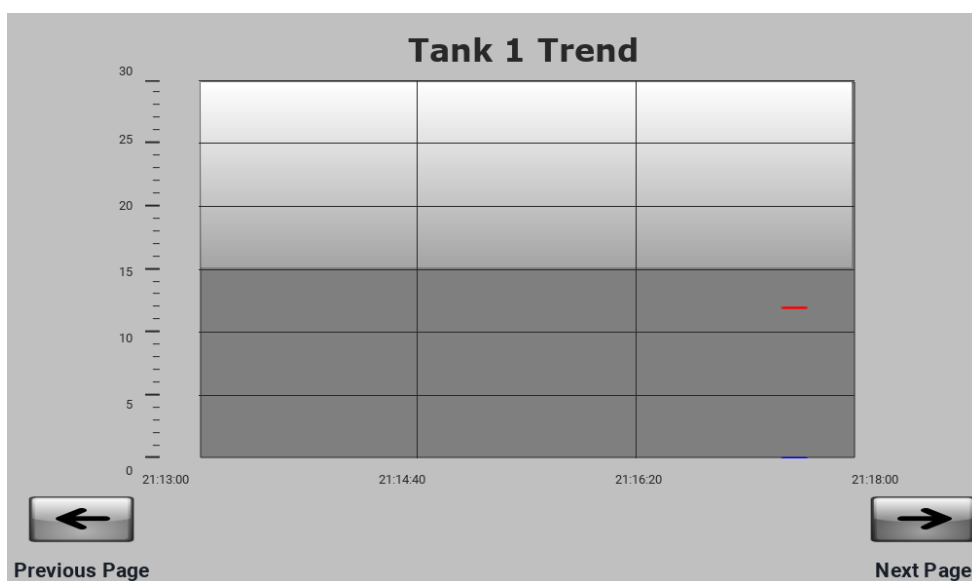


Figure 23. HMI second page

5. EXPERIMENTAL VERIFICATION OF IMPLEMENTATION

5.1 Verification of PLC-based PID controller

After implementing hardware and software, the next step was to check and validate the performance of the developed system and PLC-based PID controller. The main goal of this thesis work was to replace PC based controller with PLC-based PID controller and create a more stable and accurate control system compared with previous implementations by controlling two different shape interconnected tanks simultaneously. In the beginning, it was known that the Multitank system is nonlinear. That is why disturbance is acceptable, and overshoot in liquid level cannot be avoidable for some levels. Also, to increase the robustness, stability, and accuracy of the control system both at the industrial and educational level, it has been decided to control the system with single PID parameters rather than assigning different PID parameters for different control levels each time. PID parameters were obtained by making several experimental measurements and calculations by using the Ziegler-Nichols method. The experiment was performed in 4 different setpoints (5, 10, 15, 20 cm) for both tanks, each time the same setpoint was assigned for both tanks and performance checked by observing the level of water inside the tank visually, or by checking the real-time trend in HMI screen or by analysing the real time retrieved data from HMI trends while the system is running. Different setpoints can also be assigned regarding to the requirement, in the HMI panel, it is allowed to insert setpoint values between 0 – 20 cm. The values bigger than 20 cm are blocked in HMI panel. The collected data throughout the experiment was also used for the comparison of the performance of PLC-based controller with MATLAB based controller to make sure that the task was completed successfully.

The following PID parameters were used for all the setpoints while performing the experiment for both tanks PID controllers: $K_p = 8$, $K_i = 0,5$, $K_d = 0,1$.

The experiment started with a level 5 cm reference value, and the goal was to reach the setpoint value without overshoot during the transient phase. It has been approved that these PID parameters were suitable for controlling both tanks simultaneously without overshoot for assigned reference values. The result of the system behaviour during operation in both tanks can be seen in Figure 26. In figures related to the characteristics of both tanks for experiment must be taken into account that in graphs the green line represents setpoint or reference value for both tanks, the black line represents the actual level on the upper (tank1), and the blue line represents the actual

value on the middle tank (tank 2). From Figure 26, it can be seen that overshoot is almost zero and settling time is nearly 15 seconds for tank 2 and less than 30 seconds for tank 1.

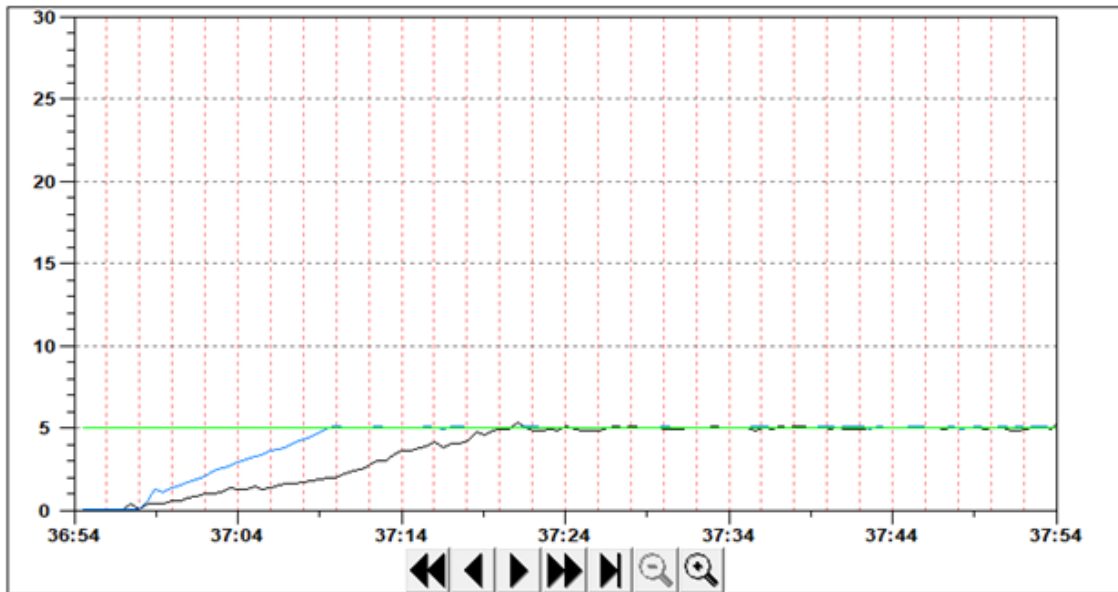


Figure 24. Tank 1 and tank 2 characteristics for 5 cm reference value

After PID parameters were verified for the 5 cm setpoint, the reference value increased to 10 cm to check the dynamics of both tanks for the same PID parameters. From Figure 27, it can be seen that characteristics are almost the same for 10 cm setpoint, overshoot practically equal to 0 cm and settling time for tank 2 is around 25 seconds and 30 seconds for tank 1. The accuracy of the graph is 500 msec, which means 2 actual values for each second for monitoring.

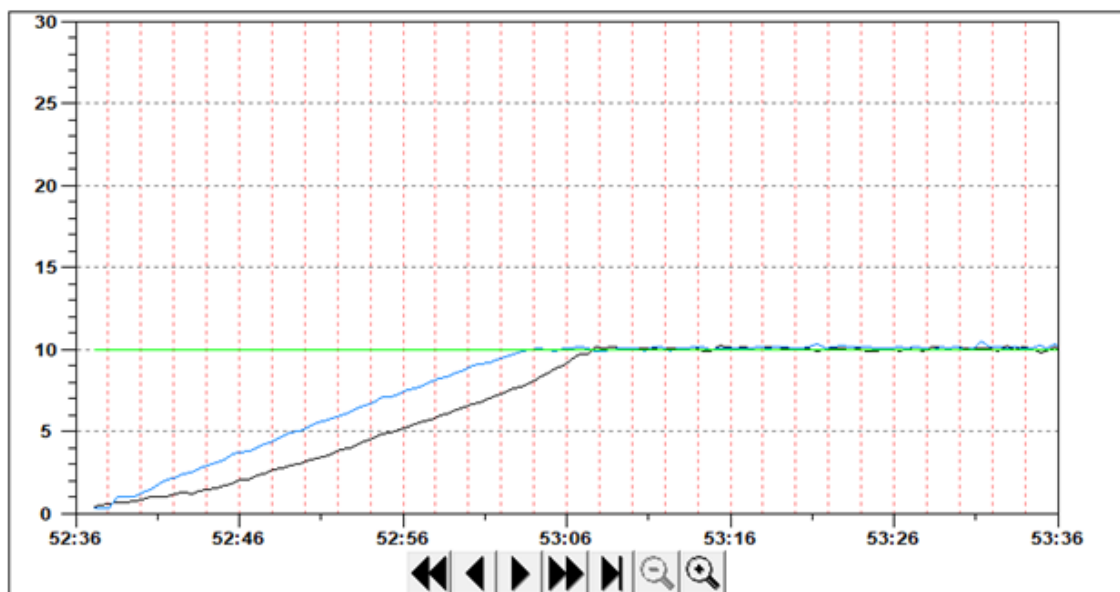


Figure 25. Tank 1 and tank 2 characteristics for 10 cm reference value

It has been approved that the same PID parameters are also applicable for the 10 cm setpoint for controlling both tanks simultaneously without overshoot and too much oscillation during the settling time. The next step is to check the system behaviour with a 15 cm reference value and verify the output results. From Figure 28, it is visible that the results also seem satisfactory for the 15 cm reference value. In tank 2, settling time is approximately 35 seconds and around 0,5 cm overshoot is observed. It is evident that oscillation is also more compared with the 5 cm and 10 cm reference values. For tank 1, the settling time is 5 seconds more from tank 2 and almost 0 cm overshoot and less oscillation. These PID parameters also seem appropriate for the 15 cm reference value.

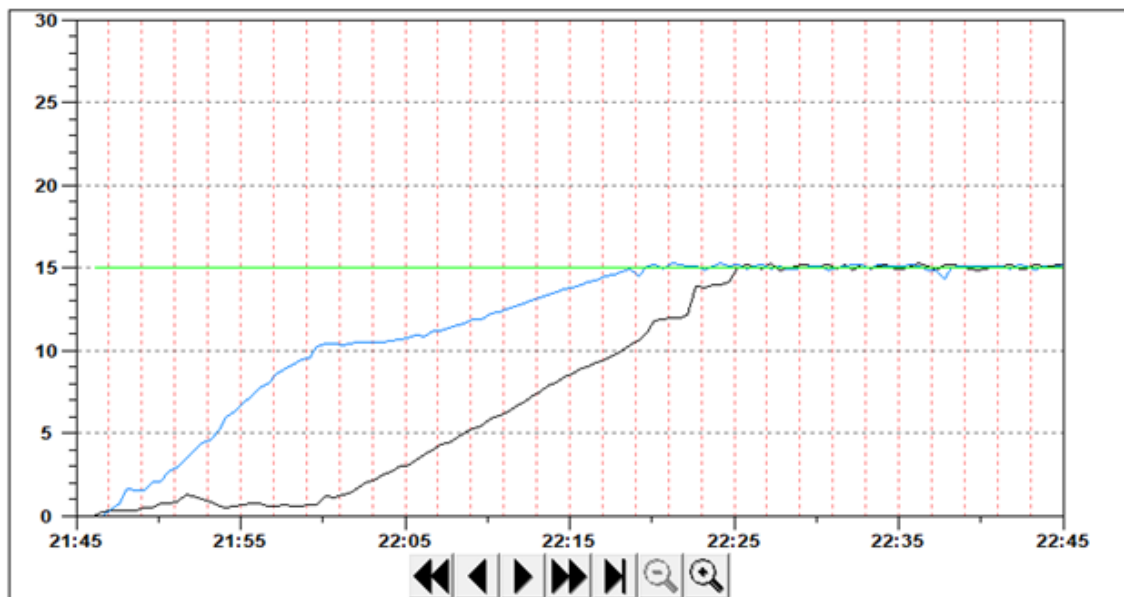


Figure 26. Tank1 and tank 2 characteristics for 15 cm reference value

The same PID parameters were also investigated for the 20 cm reference value, and in Figure 29, the output characteristics of both tanks can be observed. From the Figure 29, it seems that both tanks reached the setpoint with approximately a 1 cm overshoot from the reference value. Controller response to overshoot and settling down a level to reference value took about 8 seconds in both tanks. Output level settled down in reference value in 50 seconds in both tanks, and no more overshoot was observed for the following seconds. The same PID parameters also seem satisfactory for the 20 cm reference value. However, 1 cm overshoot was observed in both tanks for 8 seconds.

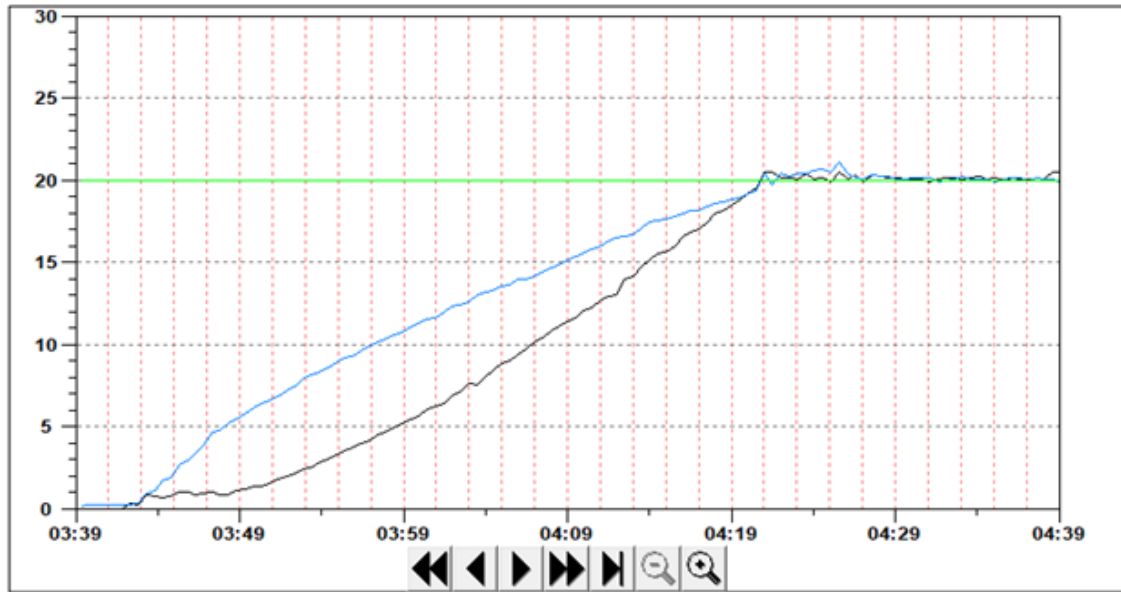


Figure 27. Tank 1 and tank 2 characteristics for 20 cm reference value

5.2 Comparing MATLAB controller with PLC controller

To make a comparison between controllers, similar control loops was developed only for an upper tank in MATLAB/Simulink environment, and experimental testing was conducted on 4 different reference values by using the same PID parameters as PLC-based PID controller use. The sampling interval is 10 msec and monitoring interval is 500 msec, and the experiment lasts 60 seconds for each experiment. The monitoring interval for the PID controller implemented on PLC also was 500 msec for each experiment, experiment time was not limited, and sampling interval is the as MATLAB controller, the experiment time can be finished by pressing stop button on HMI screen. The control diagram developed to control the Multitank system's upper tank in MATLAB/Simulink environment is shown in Figure 30 below.

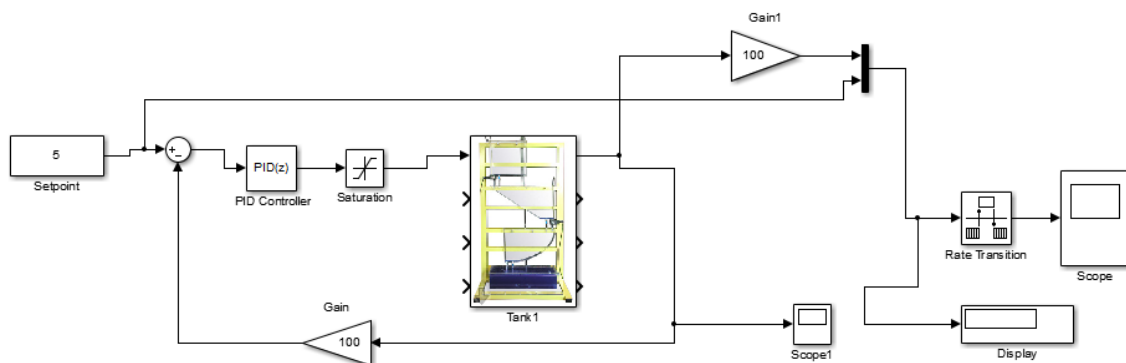


Figure 28. MATLAB/Simulink experimental control diagram

For making a comparison, the data for the PLC-based PID controller retrieved from the HMI historical trends while making verification of PLC-based PID controller for each mentioned setpoint level and data for the MATLAB based PID controller collected from the scopes while experiment is performed for each mentioned reference values. Developing a control algorithm for controlling two tanks simultaneously is a challenging and time-consuming task. Therefore, it has been decided to make a comparison only for the upper tank for each controller. In MATLAB, the control dynamics of the Simulink Discrete PID controller block were tested on 4 different setpoints: 5 cm, 10 cm, 15 cm and 20 cm. The following PID parameters were used for all the setpoints while performing the experiment: $K_p = 8$, $K_i = 0,5$, $K_d = 0,1$, which is identical for the PLC-based PID controller also. The control dynamics of both MATLAB and PLC-based controllers are almost similar for 5 cm and 10 cm reference values, oscillation after reaching the setpoint observed in both cases, as seen in Figure 31. From Figure 31, it can be seen that the steady-state error of the MATLAB based PID controller is more than PLC-based PID controller because the anti-windup mechanism of the PID controller is different in MATLAB, which is a more common issue in water level control system when the outflow is not equal or less from the inflow or vice versa. Transient time for the MATLAB controller is less than the PLC controller due to the fact that the data used in this comparison is related to the dynamics of PLC-based PID controller when controlling two tanks simultaneously. According to that logic, the upper tank started filling after the middle tank reached the setpoint. Overshoot in MATLAB controller can be decreased by enabling an anti-windup mechanism; in this solution, anti-windup has not been enabled to make both control loops as much as similar for both controllers. Settling time for the MATLAB controller is around 45 seconds which is about 30 seconds in the PLC controller. However, damped oscillation is observed more in MATLAB controller rather than PLC controller

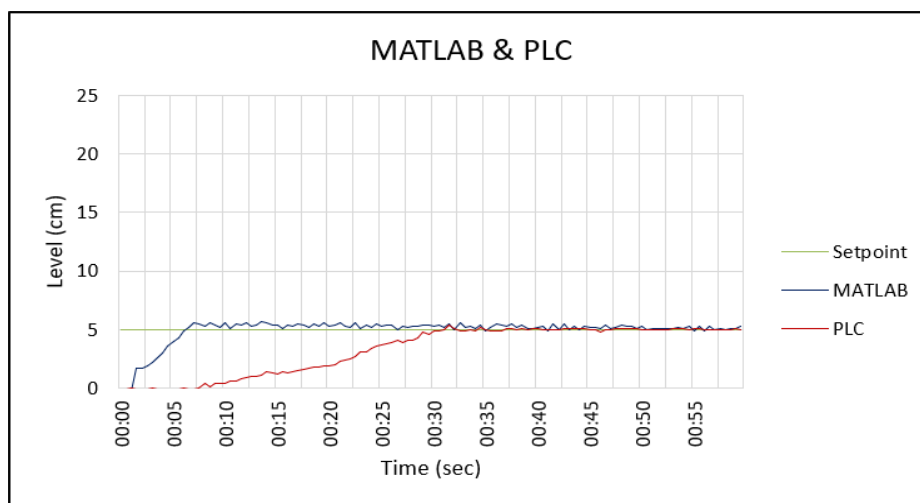


Figure 29. Comparison of MATLAB and PLC controllers with 5 cm setpoint

The experiment was also performed with a 10 cm setpoint in which the results of each controller were almost equal, see Figure 32.

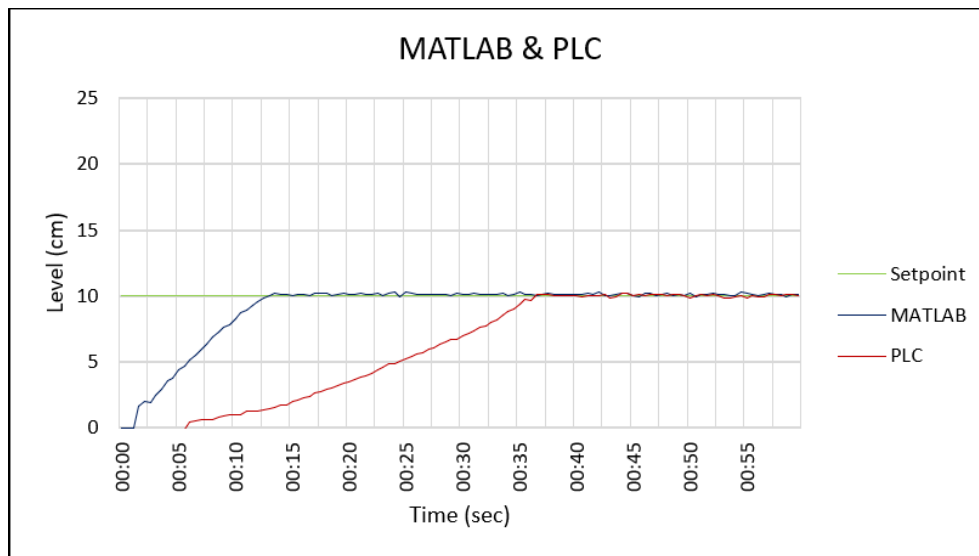


Figure 30. Comparison of MATLAB and PLC controllers with 10 cm setpoint

With the same PID settings the final experiments performed with 20 cm setpoint, and system output dynamics for tank 1 is shown in Figure 33. From the graph it is visible that for the MATLAB controller the difference between desired level and actual level is around 21,5 cm, steady state error is about 1,5 cm. From the final experiment it appears that the same PID parameters do not work for the 20 cm setpoint, proportional gain is much and need to be decreased in order to be precise enough for the requirements.

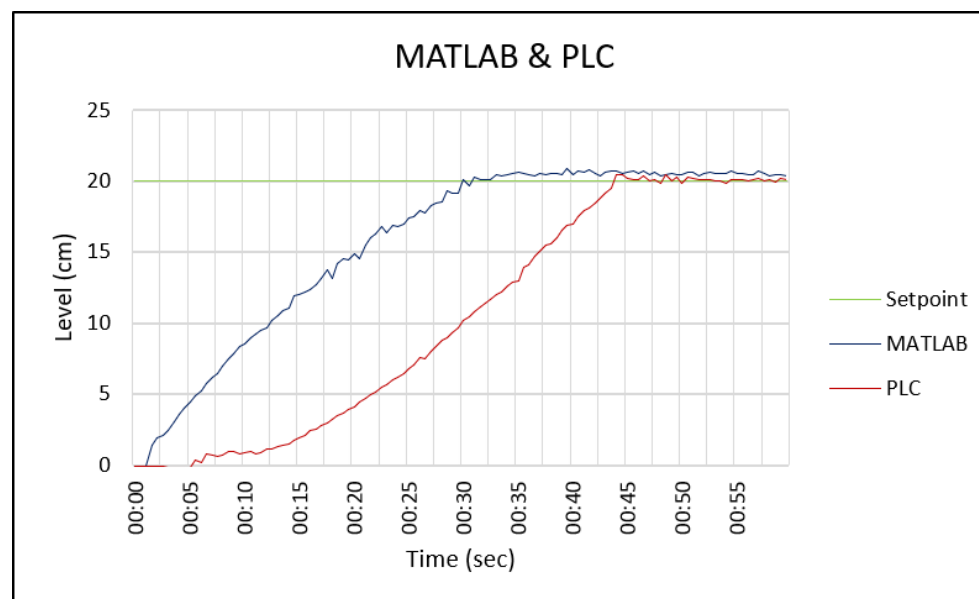


Figure 31. Comparison of MATLAB and PLC controllers with 20 cm setpoint

6. FUTURE DEVELOPMENTS

The complete control system consists of 3 different shape interconnected tanks. In this thesis work, the main task was to integrate control system to the PLC and develop control algorithm for upper and middle tank because of the hardware limitation and project difficulties. Future improvements have a much broader and more diverse scope.

- Adding PLC rack high frequency operating range PLC I/O module with enough analog I/O channels for connecting all the components directly without using MATLAB/Simulink data transmission environment as an bridge which can improve system performance and decrease wiring and data transmission complexity and delay.
- Developing control algorithm for whole system and tuning PID parameters for that configuration.
- Developing a Web-based HMI user interface tool to allow access over a network to the control system remotely.

SUMMARY

The work focused on this thesis is about developing simultaneous control of Multitank system using programmable logic controller located at Alpha Control Laboratory in TalTech. The developed system can be used both for industrial and educational purposes to teach students control methods and PLC programming by using real hardware.

The thesis contains a comprehensive literature review and methodology analysis on controller-based liquid level control, modern controllers' capabilities, configuration and optimisation methodologies. The study indicates that these quickly emerging technologies have made significant progress toward achieving error-free, rapid and cost-effective systems, mainly in the automation sector. Liquid level control systems need comprehensive approaches and designs to ensure robustness, measurement accuracy, and continuity of data transmission.

Development of the system started by analysing required devices that can be used in this work and determining their technical capabilities and availabilities in the laboratory. As a next step, the main task was to create a communication protocol between the controller and control system. Several challenges were experienced while creating a communication protocol, but most were completed successfully, as mentioned in the thesis. The following step was to develop a control algorithm and optimise system capabilities. Therefore, numerous experiments were performed, and applicable modifications were performed both in the control algorithm and parameters according to the experiment results. The development phase was completed by designing and developing the Human Machine Interface panel for the control system to control remotely. The results of the experiments, system limitations and capabilities are described in the thesis.

As a result, the main goal is reached, and the system is now ready to be used. However, the development and performance of the system can be improved by developing a control algorithm for all the 3 interconnected water tanks and instead of using extra bridge application to connect the sensors, valves and pump to the PLC, PLC input/output modules that support high frequency operating range and have enough analog input/output channels for directly connecting all the controlled devices can be used. But it was essential to understand how the system operates and how the devices could be easily connected to the system by using state of the art technologies.

KOKKUVÕTE

Käesolev lõputöö käsitleb Multitank süsteemi samaaegse juhtimise arendamist TalTechi Alpha Control laboris asuva programmeeritava loogikakontrolleri abil. Väljatöötatud süsteemi saab kasutada nii tööstuses kui ka õppeeesmärgil, et õpetada õpilastele kontrollimeetodeid ja PLC programmeerimist kasutades päris tarkvara.

Lõputöö sisaldab põhjalikku ülevaadet kirjandusest ning analüüsi kontrolleripõhise vedelikutaseme juhtimise meetoditest, kaasaegsete kontrollerite võimete, konfigureerimise ja optimeerimise meetodite kohta. Uuring näitab, et need kiiresti arenevad tehnoloogiad on teinud märkimisväärsed edusamme veavabade, kiirete ja kulutõhusate süsteemide saavutamisel, peamiselt automatiseerimise sektoris. Vedelikutaseme juhtimissüsteemid vajavad laiaulatuslikke lahendusviise ja disaini, et tagada andmete edastamises töökindlust, mõõtmistäpsust ja järjepidevust.

Süsteemi arendamine algas selles töös kasutatavate vajalike seadmete analüüsimisest ning nende tehniliste võimaluste määramisest laboris. Järgmise sammuna oli peamine ülesanne luua kontrolleri ja juhtimissüsteemi vaheline sideprotokoll. Sideprotokolli loomisel tuli ette mitmeid väljakutseid, kuid enamik neist viidi edukalt lõpule, nagu on mainitud lõputöös. Järgmine samm oli juhtimisalgoritmi väljatöötamine ja süsteemi võimaluste optimeerimine. Selleks viidi läbi arvukalt katseid ja rakendati muudatusi nii kontrollalgoritmide, kui ka parameetrites, vastavalt katse tulemustele. Arendusetapp viidi lõpule kujundades ja arendades juhtimissüsteemi kaugjuhtimiseks mõeldud juhtimissüsteemi kasutajaliidese paneeli. Lõputöös kirjeldatakse katsete tulemusi, süsteemi piiranguid ja võimeid.

Selle tulemusena on saavutatud peamine eesmärk ja süsteem on nüüd kasutamiseks valmis. Süsteemi arengut ja jõudlust saab siiski parandada, töötades välja kõigi kolme omavahel ühendatud veemahuti juhtimisalgoritmi ja selle asemel, et kasutada täiendavat sillarakendust andurite, ventiilide ja pumba ühendamiseks PLC-ga, saab kasutada PLC sisend / väljundmooduleid, mis toetavad kõrgsageduslikku töövahemikku ja millel on piisavalt analoog sisend-/väljundkanaleid, et ühendada otse kõik kontrollitud seadmed. Oli oluline mõista, kuidas süsteem töötab ja kuidas saaks seadmeid kaasaegse tehnoloogia abil hõlpsasti süsteemiga ühendada.

LIST OF REFERENCES

- [1] Graebe S.F., Goodwin G.C., Control Design and Implementation in Continuous Steel Casting, IEEE Control Systems, August 1995, pp. 64-71
- [2] Cheung Tak-Fal, Luyben W.L., Liquid Level Control in Single Tanks and Cascade of Tanks with Proportional-Only and Proportional-Integral Feedback Controllers, Ind. Eng. Chem. Fundamentals, vol. 18, No. 1, 1979, pp. 15-21.
- [3] Heckenthaler T., Engell S., Approximately Time-Optimal Fuzzy Control of a Two-Tank System, IEEE Control Systems, pp.24-30, 1994.
- [4] R. Das, S. Dutta, A. Sarkar and K. Samanta, "Automation of Tank Level Using PLC and Establishment of HMI by SCADA," IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), vol. 7, pp. 61-67, 2013.
- [5] M. F. Rahmat and S. M. Rozali, "Modelling and Controller Design for a Coupled Tank Liquid Level System: Analysis & Comparison," *Jurnal Teknologi*, pp.113 – 141, 2008.
- [6] Y. Kondratenko, O. Korobko, O. Kozlov, O. Gerasin, and A. Topalov, "PLC based system for remote liquids level control with radar sensor," in 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 1. IEEE, 2015, pp. 47 - 52.
- [7] P. Panchal, A. Patel, and J. Barve, "PI control of level control system using PLC and LabVIEW based SCADA," in 2015 International Conference on Industrial Instrumentation and Control (ICIC). IEEE, 2015, pp. 1196 - 1201.
- [8] M. Sreejeth and S. Chouhan, "PLC based automated liquid mixing and bottle filling system," in IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES). IEEE, 2016, pp. 1 - 5.
- [9] D. Engin and M. Engin, "Auto-tuning of PID parameters with programmable logic controller," in 2013 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2013, pp. 1469 - 1474.
- [10] A.Kossi, "Automated Washing and Level Controlling of a Tank System," M.S. thesis, Dep. Mech. and Produc. Eng., Vaasa Univ., Vaasa, Finland, 2012.

- [11] L. Shengduo and X. Xiaolong, "Design of Liquid Level Control System Based on PLC," in *Software Engineering and Knowledge Engineering: Theory and Practice*, Wei Zhang, Wuhan, China, 2012, pp. 415 – 424.
- [12] A. Lukman, O. Abisoye, J. Agajo, A. Ajao, A. F. Salami and M. B. Muazu, "Automated Multiple Water Tanks Control System Using ATMEGA and FPGA Technology," presented at the IEEE 1st International Conference on Mechatronics, Automation and Cyber-Physical Computer System, Owerri, Nigeria, Mar. 2019.
- [13] X. Ding, Y. Yang, S. Wang and Y. Ma, "A liquid level, temperature and flow control system based on Profibus-DP fieldbus," presented at the 5th International Conference on Advanced Design and Manufacturing Engineering, Yinchuan Ningxia, China, 2015.
- [14] Y. Liang, E. Hu, F. Niu, X. Zhou and D. Xia, "The Control System of Antibiotic Fermentation Based on PROFIBUS-DP Fieldbus" in *Microcomputer Information*, 2009, pp. 61 - 63.
- [15] A. Dumitrascu, J. Culita and D. Stefanoiu "Environment Communication and Control Systems Integrated on Teaching Platforms: Case Study: Double Water Tank System" presented at the 20th International Conference on Control Systems and Computer Science CSCS'20, Bucharest, Romania, 2015.
- [16] T.Pushpaveni, S.Srinivasulu Raju, N.Archana and M.Chandana, "Modeling and Controlling of Conical Tank System Using Adaptive Controllers and Performance Comparison With Conventional PID," *International Journal of Scientific & Engineering Research*, vol. 4, pp. 629 – 635, 2013.
- [17] M. Ma, H. Chen, R. Findeisen and F. Allgöwer, "Nonlinear Receding Horizon Control of Quadruple – Tank System and Real – Time Implementation," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 7083 – 7093, 2012.
- [18] Y. Christy, D. D. Kumar and P. G. Scholar, "Modelling and Design of Controller for Interacting Two Tank Hybrid System (ITTHS)," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 3, pp. 88 – 91, 2014
- [19] A. Smolov, "Controlling of multitank system using Raspberry Pi", M.S. thesis, Dep. Comp. Sys., Tallin University of Technology, 2017
- [20] INTECO, Multitank user manual, 2012.

- [21] Multitank overview. Last accessed 2021-10-20. [Online]. Available: <http://www.inteco.com.pl/products/multi-tank/>
- [22] RT-DAC/USB2 I/O Board User's Manual. Last accessed 2021-11-08. [Online]. Available: <https://www.inteco.com.pl/products/process-io-boards/>
- [23] PID configuration. Last accessed 2021-11-17. [Online]. Available: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=9013
- [24] PID Control, Verification, and Scheduling. Last accessed 2021-12-13. [Online]. Available: <http://robotsforroboticists.com/pid-control/>
- [25] Tuning for PID Controllers. Last accessed 2021-12-13. [Online]. Available: http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf
- [26] PLC Automation, System assembly and device specification. Last accessed 2021-12-09. [Online]. Available: <https://search.abb.com/library/Download.aspx?DocumentID=3ADR010278&LanguageCode=en&DocumentPartId=&Action=Launch>
- [27] AC500 product family. Last accessed 2021-12-09. [Online]. Available: <https://search.abb.com/library/Download.aspx?DocumentID=3ADR024117M02xx&LanguageCode=en&DocumentPartId=&Action=Launch>
- [28] Control Panel CP600-eCo. Last accessed 2021-12-09. [Online]. Available: https://library.e.abb.com/public/9c4bfc80ae894c48af6ebc7a4947dc4d/CP610_Data_Sheet_3ADR010241,%201,%20en_US.pdf
- [29] Installation and configuration manual. Last accessed 2021-12-09. [Online]. Available: <https://search.abb.com/library/Download.aspx?DocumentID=3ADR010649&LanguageCode=en&DocumentPartId=&Action=Launch>
- [30] Y. R. Kuphaldt. "Lessons in industrial instrumentation". Last accessed 2021-12-13. [Online]. Available: <https://www.ibiblio.org/kuphaldt/socratic/sinst/book/liii.pdf>

APPENDICES

Appendix 1

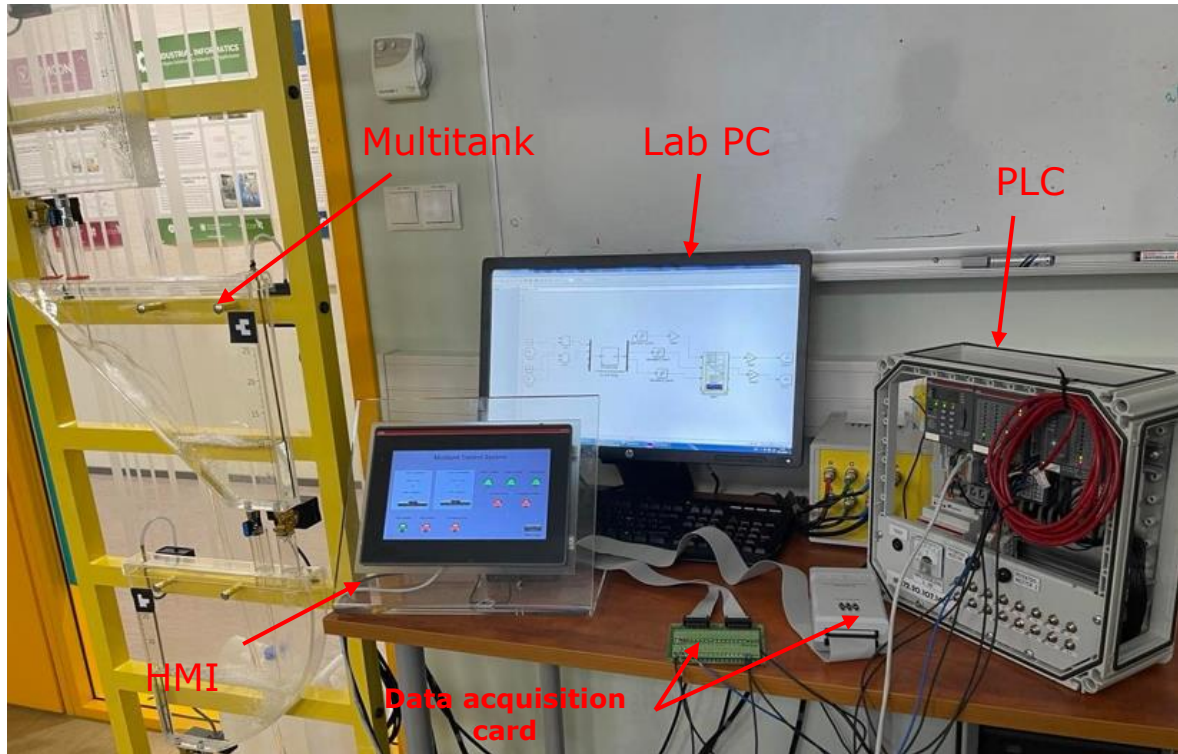


Figure 32. Complete equipment setup with Multitank

Appendix 2

Table 8. Both tanks actual level regarding assigned setpoint value

5 cm		10 cm		15 cm		20 cm	
Tank1	Tank2	Tank1	Tank2	Tank1	Tank2	Tank1	Tank2
4.951	5.096	10.051	10.147	15.075	15.001	20.022	20.045
4.915	5.051	9.874	10.032	15.148	15.084	19.873	20.053
4.959	4.956	9.980	10.163	14.876	15.075	20.093	20.153
5.034	5.057	10.082	10.202	14.947	14.848	20.072	20.148
5.029	5.000	10.030	10.108	15.154	15.033	20.023	20.134
5.079	5.025	10.097	10.088	15.061	15.103	20.058	19.884
5.036	5.003	10.023	10.097	15.047	15.150	19.891	19.948
5.030	5.051	9.860	10.032	15.132	15.139	20.079	20.082
4.986	5.118	9.853	10.070	15.189	14.907	20.094	20.222
4.993	5.037	9.919	10.064	14.897	15.158	20.129	20.075
4.801	5.029	10.015	10.105	14.782	15.080	20.058	20.042
4.994	4.963	9.868	10.076	14.844	15.129	20.15	20.026
4.951	5.077	10.036	10.100	14.911	15.073	20.179	19.881
5.071	4.941	9.9184	10.030	15.058	15.198	20.008	20.014
5.021	4.963	9.895	9.979	14.894	15.054	19.923	20.129
5.050	5.059	10.132	10.094	15.012	14.812	20.073	20.137
5.022	5.019	10.092	10.069	15.050	14.853	20.058	19.959
5.050	5.027	10.065	10.123	14.976	14.991	20.037	20.019
4.944	5.091	10.120	9.953	15.073	15.019	20.516	20.040
4.972	5.032	10.086	10.068	15.098	15.053	20.486	19.910
4.944	5.081	10.057	10.445	15.027	15.050	20.252	20.047
4.951	5.029	10.030	10.103	15.165	15.069	20.109	20.084
4.944	5.074	10.021	10.101	14.842	15.047	20.088	20.050
4.972	5.081	9.885	10.045	15.078	14.982	19.887	20.043
5.029	5.010	9.955	10.123	15.113	15.025	20.045	19.944
5.036	5.036	10.073	10.048	15.028	15.073	20.067	20.108
5.033	5.010	9.952	10.074	14.980	15.057	20.017	20.098
4.981	5.054	10.007	10.033	15.010	15.100	20.017	20.049

Appendix 3

PROGRAM PLC_PRG

VAR

(*Level 1 tags*)

diInput_level1 :REAL ;
diInputMin_level1 : REAL;
diInputMax_level1 : REAL;
diOutputMin_level1 : REAL;
diOutputMax_level1 : REAL;
diScaledOutput_level1 : REAL;

(*Level 2 tags*)

diInput_level2 :REAL ;
diInputMin_level2 : REAL;
diInputMax_level2 : REAL;
diOutputMin_level2 : REAL;
diOutputMax_level2 : REAL;
diScaledOutput_level2 : REAL;

Controller_tank2: PID;
PB_tank2: REAL:=7;
TI_tank2: REAL:=0.5;
TD_tank2: REAL:=0.1;
SP_tank1:REAL;
SP_tank2:REAL;
pump_real: REAL;
pump_int: INT;
SRSTART:SR;
START: BOOL;
Emergency:SR;
STOP: BOOL;
EmergencyStop: BOOL;
Acknowledge: BOOL;
EmergencyBLINK:BLINK;

END_VAR

(* Level1 0-27648 scaling to 0-20cm*)

```
diInputMin_level1:=40;  
diInputMax_level1:=22500;  
diOutputMin_level1:=0;  
diOutputMax_level1:= 20;  
diInput_level1:=level1;
```

```
diScaledOutput_level1 := (diInput_level1-diInputMin_level1)*(diOutputMax_level1-  
diOutputMin_level1)/(diInputMax_level1-diInputMin_level1)+ diOutputMin_level1;
```

(* Level2 0-27648 scaling to 0-20cm*)

```
diInputMin_level2:=-150;  
diInputMax_level2:=21600;  
diOutputMin_level2:=0;  
diOutputMax_level2:=20;  
diInput_level2:=level2;
```

```
diScaledOutput_level2 := (diInput_level2-diInputMin_level2)*(diOutputMax_level2-  
diOutputMin_level2)/(diInputMax_level2-diInputMin_level2)+ diOutputMin_level2;
```

```
SRSTART(SET1 := , RESET := STOP);
```

```
IF Emergency.Q1=TRUE THEN
```

```
STOP:=TRUE;
```

```
pump_real:=0;
```

```
actuator2:=FALSE;
```

```
actuator1:=FALSE;
```

```
END_IF
```

```
IF Emergency.Q1=FALSE THEN
```

```
SRSTART.SET1:=START;
```

```
END_IF
```

```
Emergency(SET1 := ((diScaledOutput_level1 > 25 OR diScaledOutput_level2 > 25)
```

```
OR EmergencyStop), RESET := (NOT(EmergencyStop) AND (Acknowledge)));
```

```
EmergencyBLINK(ENABLE:=Emergency.Q1 , TIMELOW:=t#500ms ,
```

```
TIMEHIGH:=t#500ms , OUT=> );
```

```
(*PID controller for pump*)
pump_int:= REAL_TO_INT(pump_real);
pump:=pump_int;
```

```
Controller_tank2(
ACTUAL:=diScaledOutput_level2 ,
SET_POINT:=SP_tank2,
KP:= PB_tank2,
TN:= TI_tank2,
TV:= TD_tank2,
Y_MANUAL:=0 ,
Y_OFFSET:=0,
Y_MIN:= 16900 ,
Y_MAX:= 27648 ,
MANUAL:= ,
RESET:= ,
Y=>pump_real ,
LIMITS_ACTIVE=> ,
OVERFLOW=> );
```

```
IF diScaledOutput_level2 > SP_tank2 THEN
actuator2:=TRUE;
END_IF
IF diScaledOutput_level2 < SP_tank2 THEN
actuator2:=FALSE;
END_IF
IF diScaledOutput_level2 < SP_tank2 THEN
actuator1:=TRUE;
ELSE
actuator1:=FALSE;
END_IF
IF diScaledOutput_level1 > SP_tank1 THEN
actuator1:=TRUE;
END_IF
IF diScaledOutput_level1 > SP_tank1 THEN
```

```
actuator1:=TRUE;
END_IF
IF SP_tank1=0 AND SP_tank2=0 THEN
actuator2:=FALSE;
actuator1:=FALSE;
pump_real:=0;
END_IF
```

```
IF diScaledOutput_level2>25 THEN
actuator2:=TRUE;
STOP:=TRUE;
ELSIF Acknowledge = TRUE THEN
actuator2:=FALSE;
STOP:=FALSE;
END_IF
```

```
IF diScaledOutput_level1>25 THEN
actuator1:=TRUE;
STOP:=TRUE;
ELSIF Acknowledge = TRUE THEN
actuator1:=FALSE;
STOP:=FALSE;
END_IF
```