People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering
Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of
the Requirements for the Degree of

MASTER

In **Control Engineering**

Title:

Design and Implementation of Coupled Tank Control Using PID Controller

Presented by:

HAMADOUCHE Mohamed

Supervisor:

Dr. R.KACED

Registration number ……../2024

ABSTRACT

Liquid level control is crucial in the industrial field, where the liquid level is required to be maintained to prevent overflows. The coupled tanks system is common in industrial control processes.

The system consists of two tanks, with the liquid flowing through them. Each tank contains an inlet and an outlet. The main principle of controlling this system is to maintain a constant level of liquid in each tank when there are inflows and outflows of liquid.

To control the liquid level in the coupled tanks system, the mathematical model of the system has been derived and evaluated in the form of a linear model. The mathematical model of the system was developed to apply for the PID control system, considering the dynamic behavior of the system.

Once the system had been designed, the corresponding model was implemented using the MATLAB and Simulink software. According to simulation and experimental results, the PID control worked well to stabilize the system at specific set point values.

Keywords:

Coupled tanks system, Mathematical model, PID control and MATLAB/Simulink software.

D E D I C A T I O N

I would like to dedicate this modest work to my entire family, especially my dearest father, Rabah, and mother, Djamila, my sisters, and their children.

I also extend my gratitude to my parents for teaching me life's most important lessons and inspiring my love for learning.

I would like to thank all my friends and close ones, those who answer the call for help with no expectation of personal gain.

ACKNOWLEDGEMENT

First of all, I thank God Almighty Allah for giving me the courage and patience to develop this modest work.

I would like to express my deepest gratitude to my supervisor, Dr. R. KACED, for her support, motivation, and friendship throughout the completion of this project. Her wide knowledge, logical thinking, continuous encouragement, and wise guidance made it an absolute privilege to work with her.

I would like to express my special thanks to Dr. A. OUADI for his patience and sincere efforts, and for all that he did for me, especially his valuable time and useful comments throughout the year.

I also wish to express my sincere thanks to all the members of the institution for their help and support.

Lastly, I would like to express my profound gratitude to my beloved parents and family for providing me with unfailing support and continuous encouragement throughout the years of study and during the process of researching and writing this report.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SISO: Single Input Single Output.

MATLAB: Matrix laboratory.

PID: Proportional-plus-Integral-plus-Derivative.

EOM: Equation of Motion.

CTS: Coupled Tanks System.

$\boldsymbol{f}$**:** denotes a function.

u(t): The control signal at time t.

u[k]: The control output at the discrete time step k.

t: Time or instantaneous time.

Δt: The sampling time.

$\boldsymbol{\tau}$ **:** The time constants.

$\mathbf{K_{dc}}$**:** The DC gains.

$\mathbf{F}_{\boldsymbol{i}}$**:** The flow rate of liquid.

$\mathbf{A_{ti}}$**:** The cross section area of tanks.

$\mathbf{D_{ti}}$**:** The tanks inside diameter.

$\mathbf{A_{oi}}$**:** The outflow orifice area of tanks.

$\mathbf{D_{oi}}$**:** The outflow orifice diameter.

$\mathbf{C_d}$**:** The coefficient of discharge.

$\mathbf{K_P}$**:** The pump flow constant0

g: The gravitational constant (981 cm/s^2).

$\mathbf{v_O}$**:** The outflow velocity.

$\mathbf{K_{pi}}$**:** The pump constant.

SP: Set Point.

PV: Process Variable.

GENERAL INTRODUCTION

This project delves into the principles and practical applications of liquid level control systems, specifically focusing on a coupled tanks system (CTS). This system is commonly used in various industries such as pulp and paper mills, petrochemical plants, and water treatment facilities to ensure operational efficiency and safety.

The coupled tanks system consists of a single pump and two interconnected tanks, each equipped with pressure sensors to measure water levels. The setup allows for different configurations, demonstrating challenges related to fluid dynamics and time delays.

The report also delves into control theory, discussing the use of PID controllers, which are fundamental in regulating system behavior through proportional, integral, and derivative actions. Advanced modeling and simulation techniques using MATLAB and Simulink are employed to design and analyze control systems, highlighting the importance of accurate calibration and real time monitoring.

This comprehensive study serves as a valuable resource for understanding the intricacies of liquid level control and the implementation of effective control strategy in industrial applications.

The project is structured into the following chapters:

1. Introduction: An overview of the liquid level control system is given, in which external factors can pose a major challenge.

2. Materials & Models: Divided into hardware and software sections including model calculation, this chapter will include a detailed architecture of the system and its model. In the model calculation section, a set of differential equations will be used to simulate the behavior of the liquid level control system.

3. Simulated PID Controller for Coupled Tank Process: The simulated PID controller will regulate liquid level for each tanks. It will also adjust the flow rate of the pump as needed. The calculations, including the design and tuning of the PID controller will be done using MATLAB and Simulink software. The results of the simulations will be used for the physical PID controller parameters.

4. Physical PID Controller for Coupled Tank Process: Implementing a physical PID controller can be challenging due to the need for fast and accurate calculations to maintain control of the liquid flow for the system. The values of the PID controller parameters obtained from simulation will be used in the experiment implementation. In addition, careful calibration and testing have been conducted to ensure that the system performs satisfactorily and reliably in the coupled tanks process.

- CHAPTER 1 -
INTRODUCTION

1.1 Background:

Liquid level control is common in many industries, such as pulp and paper mills, petrochemical plants, and water treatment facilities. These facilities need to be precise in order to function well; such precision results in efficiency, safety and preservation of the environment.

Petrochemical plants rely on accurate level control to manage complex chemical reactions and avoid hazardous spills. Water treatment facilities need stringent level monitoring to guarantee the supply of clean water and the proper treatment of wastewater. [1]

1.2 Overview of the work:

The coupled tanks system control is an important process for a tough control techniques and focus on the development of practical, robust and flexible system for a further experimental issue, parameters and an unlimited model changes.

The user is allowed to develop a mathematical model and simulate it using both MATLAB and Simulink software for MATHWORKS by comparing the simulated and the experimental results and analyze the errors occurring by a PID tuning system.

This system enables various single input single output (SISO) configurations. It demonstrates unique challenges related to fluid dynamics, pressure and time delays.

The coupled tanks can be arranged into many experiments of varying difficulty, offering a range of modeling and control challenges in laboratory settings.

1.3 Problem statement:

The Proportional-plus-Integral-plus-Derivative (PID) controllers have found wide acceptance and applications in the industries for the past few decades. It has a simple control structure which was understood by plant operators and which they found relatively easy to tune.

In spite of the simple structures, PID controllers are proven to be sufficient for many practical control problems and hence are particularly appealing to practicing engineers. An abundant amount of research work has been reported in the past on the tuning of PID controllers.

Since many control systems using PID control have proved satisfactory, it still has a wide range of applications in industrial control. In this project, a useful PID Controller design technique will be presented, and implementation issues will also be discussed.

The PID controller will be designed to independently control the liquid levels in tank 1 and tank 2. The simulation of proportional, integral and derivative actions are explained in detail, and variations of the basic PID structure are also introduced. Additionally, the implementation of a PID controller tuning procedure will be presented.

1.4 **Objectives:**

There are several objectives that must be achieved in order to make this project successful:

1. To model the coupled tanks where a mathematical model that describes the relationship between the pump flow rate, the interconnection between the tanks, and the liquid levels in tanks.
2. To develop a PID Controller for controlling the liquid level of tank one and tank two of coupled tanks system.
3. To validate the result from simulation (using MATLAB) through experimental setup (physical implementation).

1.5 **Main definitions:**

In this section, some concepts repeated many times are explained to make the reading of this report more comprehensive.

Dynamic process: System that depends on both the input applied to the process and the current state of the process.

State variables: The smallest amount of variables that can represent a whole dynamic system at any time. The state variables have to be linearly independent, and the minimal number of them is the order of the differential equation that represents the system.

Continuous time model: Model of a process that describes it in the continuous time, with infinitesimal time variations to consider the evolution of the system.

Discrete time model: Model of a process that just takes into account the variable values in samples of time multiple of a base period.

Non-linear model: Mathematical model of a system which equations cannot be represented as a particular case of:

$$f(x) = \sum_{i=1}^{n} m_i x_i \qquad \textbf{1.1}$$

Linearization: Techniques applied to approximate a non-linear model into a linear one. There are two techniques used mainly: linearization around an equilibrium point and feedback linearization. It is important given the easy manipulation of a linear model to control a process.

Equilibrium point: Particular values of the process state used to linearize a model. An equilibrium point x_0 is characterized by the following expression:

$$\left.\frac{df(x)}{dx}\right|_{x=x_0} = 0 \qquad \textbf{1.2}$$

Linearization around equilibrium point: Technique used to linearize a model, and valid with small variations in the process state variables around that equilibrium point. This technique is based on the Taylor series expansion:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \qquad \textbf{1.3}$$

Where x_0 is the chosen equilibrium point. As the equilibrium point equation is an infinite expression, to approximate the model to a linear one the first two terms are used.

State space representation: Matrix representation of a linear model that defines the evolution of all the state variables function of the own state variables and the input. With a model:

$$\begin{cases} \frac{d}{dt}x(t) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases} \qquad \textbf{1.4}$$

Where

- x(t) is the state vector.
- u(t) is the control input.
- y(t) is the output.
- A, B, C, and D are matrices that define the system dynamics.

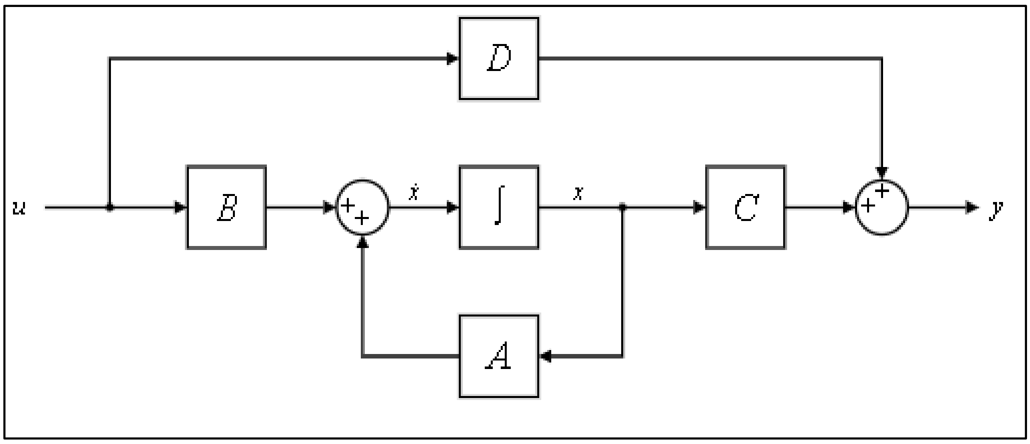


Figure 1.1: Block diagram representation of the linear state-space equations. [2]

Transfer function matrix representation: Technique to represent a process linear model based just in the relation between the inputs and the outputs. To apply it, in continuous time the Laplace transform is used, which discrete equivalent is the Z transform obtaining the following representations. In continuous time: [3]

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{U(s)}} = \boldsymbol{G(s)} \qquad \textbf{1.5}$$

1.6 PID controller overview:

The PID controller is the most common form of feedback. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s. In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used. [4]

It uses just three parameters to influence the control signal it generates, making it both effective and easy to implement. This straightforward approach has made the PID controller a favorite in a wide range of applications.

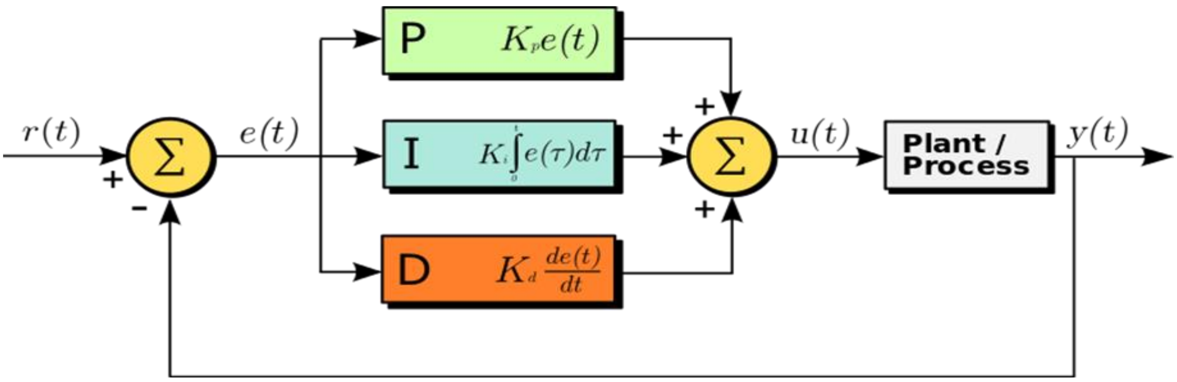


Figure 1.2: Block diagram of continuous-time PID controller in a feedback loop.

Continuous-time PID formula:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \qquad 1.6$$

Where:

- u(t) is the control signal at time t.
- Kp is the proportional gain.
- Ki is the integral gain.
- Kd is the derivative gain.
- e(t) is the error at time t, defined as the difference between the reference input r(t) and the process variable y(t).

The discrete PID equation is a direct consequence of the discretization of continuous terms (1.6). The integral term is transformed in a sum, and the derivative in a slope between two separated points.

There are two ways to the design a discrete PID controller:

- Directly discretizing a designed continuous PID controller.
- Design adjusting parameters directly in a discrete environment.

Discrete-time PID formula:

$$u[k] = k_p e[k] + K_i \sum_{i=0}^{k} e[i] \Delta t + K_d \frac{e[k] - e[k-1]}{\Delta t} \quad \textbf{1.7}$$

Where:

- u[k] is the control output at the discrete time step k.
- Kp is the proportional gain.
- Ki is the integral gain.
- Kd is the derivative gain.
- e[k]is the error at the discrete time step k.
- Δt is the sampling (time).

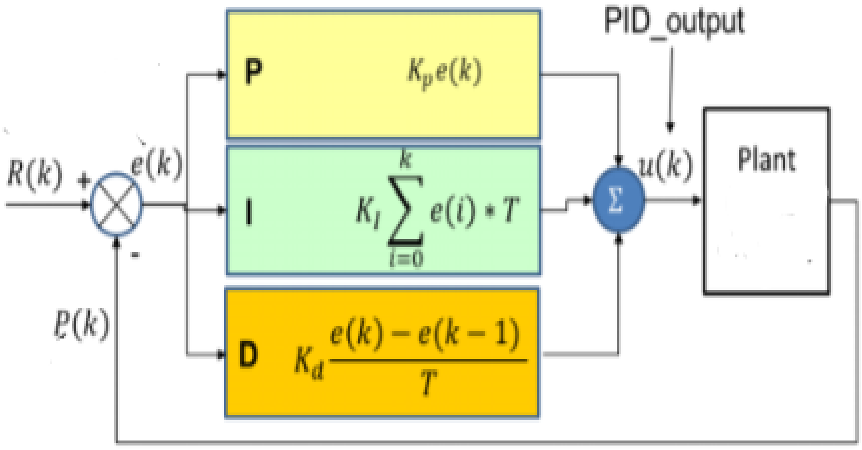


Figure 1.3: Block diagram of discrete time-PID controller in a feedback loop.

1.7 Control effects of proportional, integral and derivative action:

The "three-term" functionalities are highlighted by the following:

1.7.1 Proportional term:

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant. The proportional term is given by:

Continuous-time P formula : $P(t) = K_p e(t)$ **1.8**

Discrete-time P formula : $P[k] = K_p e[k]$ **1.9**

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is very high, the system may become unstable. In contrast, a small gain results in a small output response to a large input error.

If the proportional gain is very low, the control action may be too small when responding to system disturbances. Consequently, a proportional controller will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. [5]

1.7.2 Integral term:

The integral influence is proportional to the variation of the error on time. The integral term is given by:

Continuous-time I formula : $$I(t) = K_i \int_0^t e(\tau)d\tau \quad \textbf{1.10}$$

Discrete-time I formula : $$I[k] = K_i \sum_{i=0}^{k} e[i]\Delta t \quad \textbf{1.11}$$

The most important benefit is that this term eliminates the steady-state error, but it has a disadvantage, which is the fact that the stability of the system is affected.

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller. [6]

1.7.3 Derivative term:

The derivative term is proportional to the rate of change of the error, as we can see on the equations below. The derivative term is given by:

Continuous-time D formula : $$D(t) = K_d \frac{de(t)}{dt} \quad \textbf{1.12}$$

Discrete-time D formula : $$D[k] = K_d \frac{e[k]-e[k-1]}{\Delta t} \quad \textbf{1.13}$$

This term makes an estimation of the future error and by that it can increase or decrease the speed of correction, because it can work in an early way when there are detected any changes on the error. This term is very sensitive to high frequency noises. [7]

Table 1.1: The effect of gain coefficient on the performance of PID controller.

Parameter	**Rise time**	**Overshoot**	**Settling time**	**Steady state error**
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Minor decrease	Decrease	Decrease	No effect

1.8 Conclusion:

In this chapter, an overview of the work is given, including the objective of the project, main definitions concerning control systems, and the problem statement. At the end of this chapter, an overview of PID controller is introduced.

- CHAPTER 2 -

Materials & Models

2.1 Introduction:

This chapter serves as an introduction to the hardware and software used in this project and the way those components are correlated to implement the system.

The modeling of the system where a set of differential equations is used with a state space representation and since our system is nonlinear, a linearization is required. For an exact analysis, a system identification is required to determine the model parameters of the physical system.

2.2 Coupled tanks system description:

The coupled tanks plant is a bench-top apparatus commonly used in educational and research settings to study liquid level control.

This system consists of a water basin, a pump, and two tanks of uniform cross-section as shown in Figure 2.1. The tanks are mounted in such a way that water from the first tank flows into the second tank, and subsequently returns to the main water reservoir. This setup creates a closed, recirculating system that is ideal for examining various control strategies and fluid dynamics principles. [8]

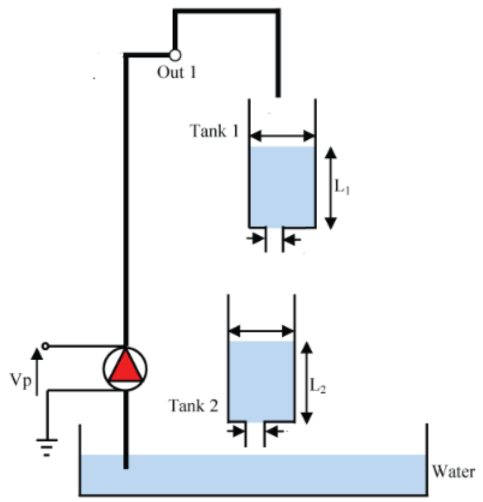


Figure 2.1: Coupled tanks system. [9]

The water basin acts as the main reservoir, holding the water that circulates through the system. The pump moves the water from the basin into the first tank, initiating the process. The first tank, positioned higher, allows water to flow into the second tank due to gravity. The second tank then channels the water back to the main reservoir, maintaining the closed-loop nature of the system.

The uniform cross-section of the tanks ensures consistent flow characteristics, which is crucial for accurate modeling and analysis. This consistency aids in the examination of fluid dynamics principles, such as flow rates and pressure changes. Additionally, the ability to adjust the pump speed and monitor water levels in tanks to study system responses and control techniques.

2.3 Main components of the coupled tank system:

The primary components of the coupled tank system include a pump, two tanks, and various connectors and sensors. The pump draws water from the main reservoir and delivers it to the upper tank through quick connect orifice labeled "Out1".

Each tank has an outflow orifice at its bottom, through which water exits under atmospheric pressure. The system includes pressure-sensitive sensors at the top of each tank to measure the water level.

- **Overall frame:** A Plexiglas bench top holds the whole system components.
- **Water Tanks:** Two identical, transparent plastic tanks, each with a capacity of up to one liter, provide clear visibility of the water levels during operation.
- **Water reservoir:** It is the lower part of the system, which is the input source of the pumps. This main water reservoir ensures a continuous supply for circulation within the coupled tanks.
- **Water tubing:** Tubes connecting the reservoir with the pump to fill the upper tank.
- **Water Pump:** The RS-385 DC mini water Pump is a compact and efficient pump designed for educational purpose. With its low-voltage dc operation and versatile performance, it ensures optimal water circulation while consuming minimal energy. [10]

Figure 2.2: RS-385 DC water pump.

- **Pressure sensor:** This HX710B air pressure sensor module uses a high-precision AD sampling chip, adopts a 0-40 KPa air pressure sensor, can connect a 2.5mm hose, can detect water level, and other air pressure Features. [11]

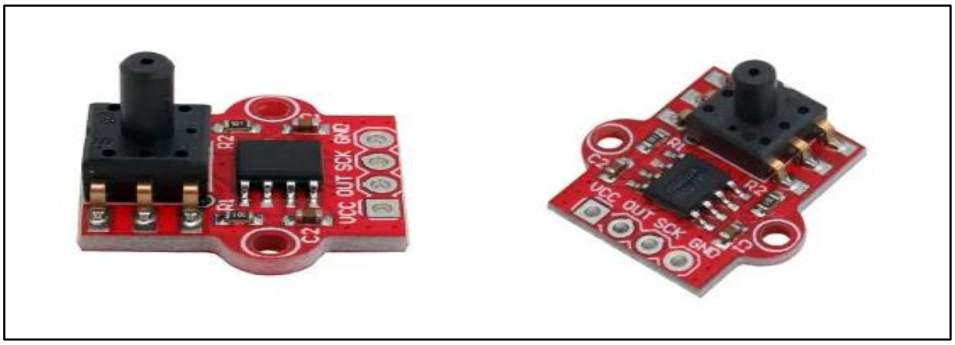


Figure 2.3: HX710B air pressure sensor.

- **Arduino Mega:** It is a microcontroller board based on the ATmega 2560. It features 54 digital input/output pins, 14 of which can be used as PWM outputs, 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The board includes all necessary components to support the microcontroller; it can be connected to a computer via a USB cable or powered with an AC-to-DC adapter or battery for immediate use. [12]

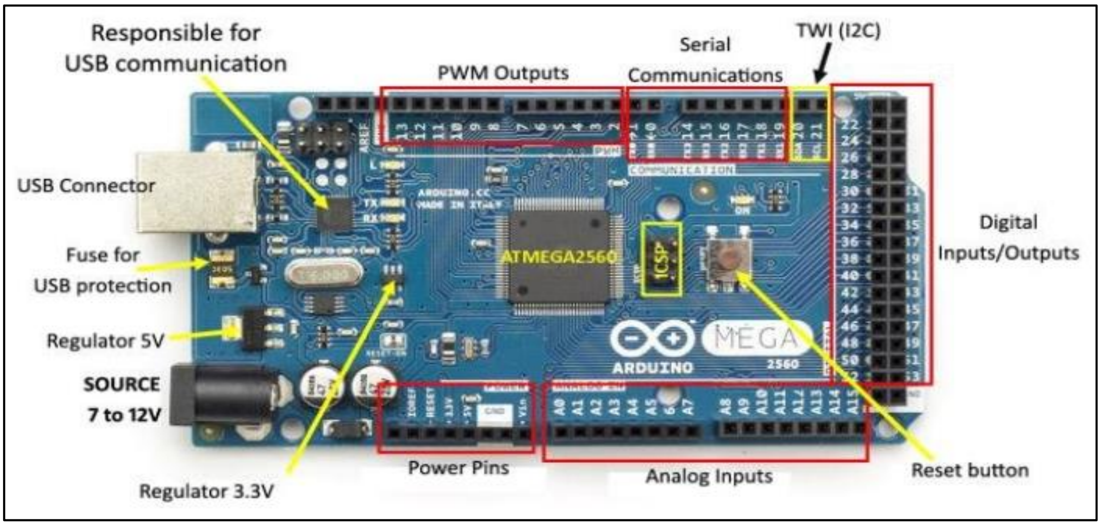


Figure 2.4: Arduino ATmega 2560 borad.

- **Arduino motor shield L293d:** L293D shield is a driver board based on L293 IC, which can drive 4 DC motors and 2 stepper or Servomotors at the same time. Each channel of this module has a maximum current of 1.2A and the voltage is less than 25v or less more 4.5v. this compatible with arduino UNO and ATMega. [13]

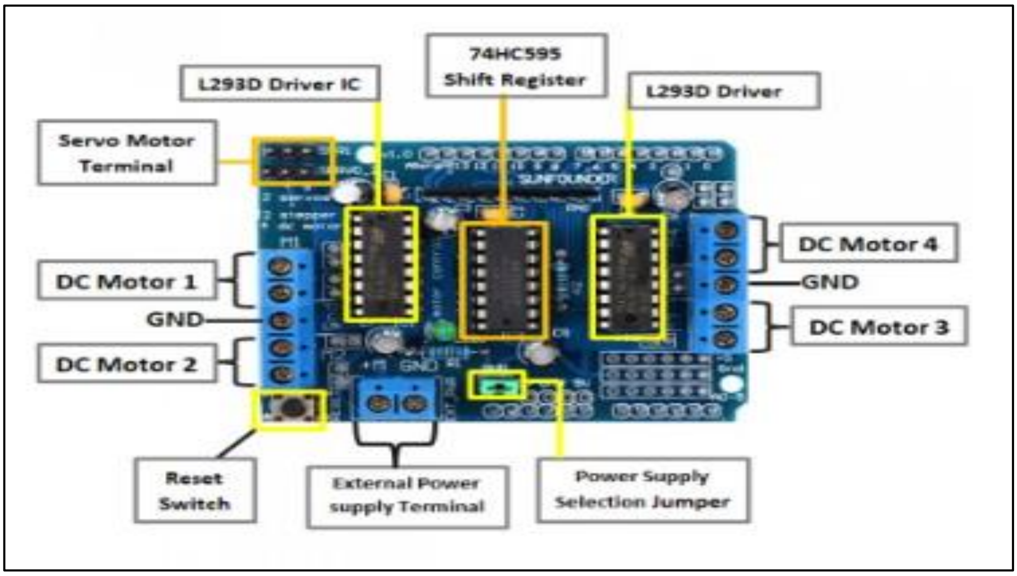


Figure 2.5 : Arduino motor shield L293d board.

- **Power supply:** due to the different components that has been used in the circuit implementation a need of power sources was required where we used a simple 12V DC source.

2.4 Wiring of the system:

The coupled tanks system can indeed be divided into three main parts: the computer-based human interface, the hardware controller, and the real system. A general block description is given in the following figure 2.6.

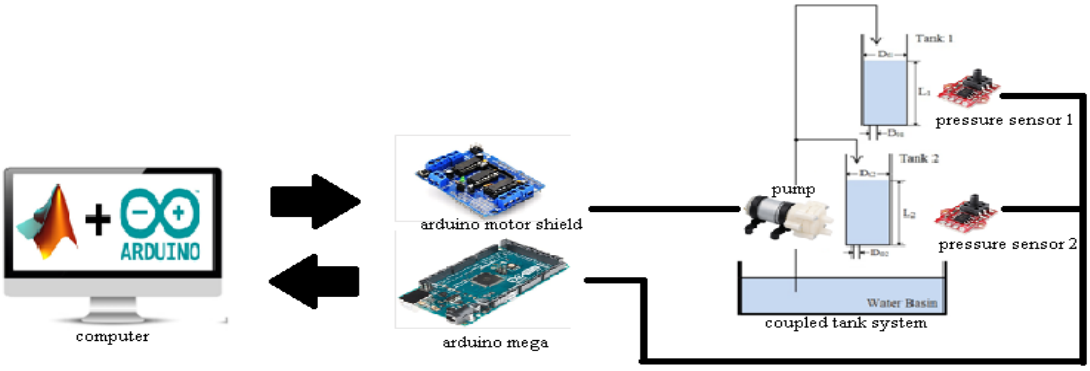


Figure 2.6: General block description of the system.

The computer-based human interface involves a computer running MATLAB software interfaced with an arduino microcontroller. The computer serves as the platform for programming, monitoring, and controlling the system. A human operator can interact with the system via MATLAB, allowing for setting desired water levels, starting or stopping the pump, and monitoring physical data from the sensors. MATLAB is used to develop control algorithms, including a PID controller, that determine how the pump should operate based on sensor readings.

The hardware controller consists of an arduino Mega and an arduino Motor Shield L293D. The arduino Mega acts as the central controller, executing the control algorithms developed in MATLAB. It collects data from the sensors and sends control signals to the pump. The arduino Motor Shield extends the capabilities of the arduino Mega by providing the necessary hardware to control the pump, managing its power requirements and control signals.

The real system includes a physical coupled tank setup, which comprises two tanks where water levels need to be controlled. Water flows from Tank 1 to Tank 2 and eventually into a water basin. Pressure sensors in each tank measure the water levels and provide physical data to the microcontroller. The pump, controlled by the arduino Motor Shield based on the control signals from the arduino Mega, moves water into Tank 1. The water basin collects water from Tank 2 and recycles it back into the system through the pump.

In summary, the overall system features a computer running MATLAB to design and monitor control algorithms, an arduino setup to execute these algorithms and control the pump, and a physical coupled tanks system to be controlled. This setup allows for practical implementation and testing of control algorithms, linking theoretical concepts with experimental applications.

2.5 Analysis and modelling:

A theoretical study of the system is necessary before implementing the controller. This provides us with information on the system behavior and allows us to control it and achieve good performance without risking overflows.

In this part, we will address the modeling of the coupled tanks system to develop an adequate controller. Following this, we will proceed with the simulation using MATLAB and Simulink software. Finally, we will implement it in a physical system.

Modeling is the process of creating algebraic or differential equations to characterize a system response to a given input.

Therefore, the input to the process is the voltage to the pump V_p and its output is the water level in tank 1, L_1.However, the water level Equation of Motion (EOM) in tank 2 still needs to be derived. The input to the tank 2 process is the water level in tank1,L_1 and its output variable is the water level in tank 2, L_2.

The purpose of the present modelling is to provide the system open-loop transfer function, $G_1(s)$ and $G_2(s)$, which in turn will be used to design an appropriate level controller. [14]

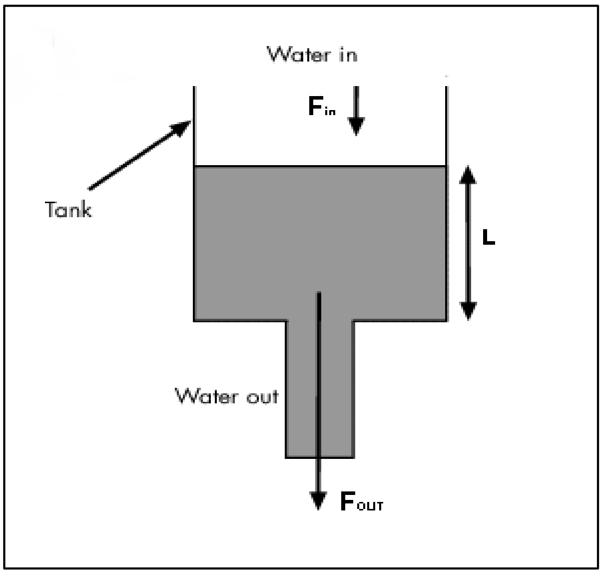


Figure 2.7: Schematic diagram of tank fluid flow system.

Therefore, it should express the resulting EOM under the following format:

$$\frac{\partial L_1}{\partial t} = f(L_1, V_p) \quad (2.1)$$

$$\frac{\partial L_2}{\partial t} = f(L_2, L_1) \quad (2.2)$$

Where: f denotes a function.

In deriving the Tanks EOM the mass balance principle can be applied to the water level in tank1 and tank 2 as follows:

$$A_{t1}\frac{\partial L_1}{\partial t} = F_{i1} - F_{o1} \qquad 2.3$$

$$A_{t2}\frac{\partial L_2}{\partial t} = F_{i2} - F_{o2} \qquad 2.4$$

Where:

A_{t1} is the cross section area of Tank 1. F_{i1} and F_{o1} are the inflow rate and outflow rate, respectively.

A_{t2} is the cross section area of Tank 2. F_{i2} and F_{o2} are the inflow rate and outflow rate, respectively.

The volumetric inflow rate F_{i1} to tank 1 is assumed directly proportional to the applied pump voltage, such that:

$$F_{i1} = K_p V_p \qquad 2.5$$

The volumetric inflow rate to tank 2 is equal to the volumetric outflow rate from tank 1, that is to say:

$$F_{i2} = F_{o1} \qquad 2.6$$

Applying Bernoulli equation for small orifices, the outflow velocity from tank 1 and tank 2, can be expressed by the following relationship:

$$\begin{cases} v_{o1} = \sqrt{2gL_1} & 2.7 \\ v_{o2} = \sqrt{2gL_2} & 2.8 \end{cases}$$

Where, **g** is the gravitational acceleration in (cm/s^2)

The outflow rate from tank 1 and Tank 2 can be expressed by:

$$\begin{cases} F_{o1} = A_{o1} v_{o1} & 2.9 \\ F_{o2} = A_{o2} v_{o2} & 2.10 \end{cases}$$

Where the tanks Outflow orifice Area is defined as follows.

$$A_{o1} = \frac{\pi * D_{o1}^2}{4} \qquad 2.11$$

$$A_{o2} = \frac{\pi * D_{o2}^2}{4} \qquad 2.12$$

The outflow rate from tank 1 and tank 2 can be written as:

$$\begin{cases} Fo1 = A_{o1}\sqrt{2gL1} & 2.13 \\ Fo2 = A_{o2}\sqrt{2gL2} & 2.14 \end{cases}$$

Rearranging results in the following equation of motion for the tank 1 and tank 2 system:

$$\begin{cases} \dfrac{\partial L_1}{\partial t} = \dfrac{K_p V_p - A_{o1}\sqrt{2}\sqrt{gL_1}}{A_{t1}} & \mathbf{2.15} \\ \dfrac{\partial L_2}{\partial t} = \dfrac{-A_{o2}\sqrt{2}\sqrt{gL_2} + A_{o1}\sqrt{2gL_1}}{A_{t2}} & \mathbf{2.16} \end{cases}$$

The above two equations (2.15) and (2.16) are the nonlinear mathematical model of the coupled tanks plant.

In order to design and implement a linear level controller for the tank 1 and tank 2 systems, the open-loop Laplace transfer functions should be derived. However by definition, such transfer functions can only represent the system dynamics from linear differential equations. Therefore, the nonlinear EOM of tank 1 and tank 2 should be linearized around a quiescent point of operation.

The nonlinear mathematical model can be linearized about a specific operating pump voltage and water levels for both tanks. The linearization process is considered for each tank separately. [15]

The specific operating pump voltage and water levels for both tanks be denoted as V_{p1}, L_{11} and L_{21} then the pump voltage and water levels are changed as follows:

$$\begin{cases} V_p = V_{po} + V_{p1} \\ L_1 = L_{10} + L_{11} \\ L_2 = L_{20} + L_{21} \end{cases} \quad \mathbf{2.17}$$

At the equilibrium point, all time derivative terms equate zero.

$$K_p V_{po} - A_{o1}\sqrt{2}\sqrt{gL_{10}} = 0 \quad \mathbf{2.18}$$

$$A_{o1}\sqrt{2}\sqrt{gL_{10}} - A_{o2}\sqrt{2}\sqrt{gL_{20}} = 0 \quad \mathbf{2.19}$$

Solving the equation (2.18) for $\boldsymbol{V_{po}}$ gives the pump voltage at equilibrium. $\boldsymbol{V_{po}}$ results to be a function of L_{10} and K_p , as expressed below:

$$V_{po} = \frac{A_{o1}\sqrt{2}\sqrt{gL_{10}}}{K_p} \quad \mathbf{2.20}$$

Solving the equation (2.19) for L_{10} gives the tank 1 water level at equilibrium. L_{10} results to be a function of $\boldsymbol{L_{20}}$, as expressed below:

$$L_{10} = \frac{{A_{o2}}^2 L_{20}}{{A_{o1}}^2} \quad \mathbf{2.21}$$

Using Taylor's series expansion to linearize the two equations that describe the nonlinear mathematical model of the coupled tanks plant, the linearized model is obtained as:

$$\begin{cases} \dfrac{\partial L_{11}}{\partial t} = \dfrac{K_p V_{p1}}{A_{t1}} - \dfrac{1}{2}\dfrac{A_{o1}\sqrt{2}gL_{11}}{A_{t1}\sqrt{gL_{10}}} \\ \dfrac{\partial L_{21}}{\partial t} = \dfrac{1}{2}\dfrac{A_{o1}\sqrt{2}gL_{11}}{\sqrt{gL_{10}}A_{t1}} - \dfrac{1}{2}\dfrac{A_{o2}\sqrt{2}gL_{21}}{A_{t2}\sqrt{gL_{20}}} \end{cases} \quad 2.22$$

The linearized model of the coupled tanks liquid level system can be described by the following state-space model:

$$\begin{bmatrix} \dfrac{\partial L_{11}}{\partial t} \\ \dfrac{\partial L_{21}}{\partial t} \end{bmatrix} = \begin{bmatrix} -\dfrac{A_{o1}g\sqrt{2}}{2A_{t1}\sqrt{gL_{10}}} & 0 \\ \dfrac{A_{o1}g\sqrt{2}}{2A_{t1}\sqrt{gL_{10}}} & -\dfrac{A_{o2}g\sqrt{2}}{2A_{t2}\sqrt{gL_{20}}} \end{bmatrix} \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} + \begin{bmatrix} \dfrac{K_p}{A_{t1}} \\ 0 \end{bmatrix} [V_{p1}] \quad 2.23$$

$$y = [0 \quad 1] \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix}$$

From the linear equations of motion, the system open-loop transfer function in the laplace domain can be defined by the following relationship:

$$\begin{cases} G_1(s) = \dfrac{L_{11}(s)}{V_{p1}(s)} = \dfrac{K_{dc1}}{\tau_1 s + 1} \\ G_2(s) = \dfrac{L_{21}(s)}{L_{11}(s)} = \dfrac{K_{dc2}}{\tau_2 s + 1} \end{cases} \quad 2.24$$

Where $K_{dc1}$and K_{dc2} are the open-loop transfer function dc gains. τ_1 and τ_2 are the time constants.

$$\begin{cases} K_{dc1} = \dfrac{K_p\sqrt{2}\sqrt{gL_{10}}}{A_{o1}g} \quad , \tau_1 = \dfrac{A_{t1}\sqrt{2}\sqrt{gL_{10}}}{A_{o1}g} \\ K_{dc2} = \dfrac{A_{o1}\sqrt{L_{20}}}{A_{o2}\sqrt{L_{10}}} \quad , \tau_2 = \dfrac{A_{t2}\sqrt{2}\sqrt{gL_{20}}}{A_{o2}g} \end{cases} \quad 2.25$$

Figure 2.8 shows a block diagram for a coupled tanks system in terms of the open loop transfer function of each tank. [16]

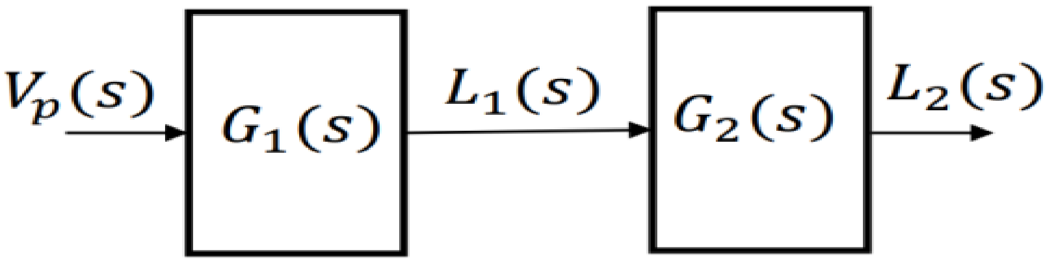


Figure 2.8: Tanks connection block diagram. [14]

2.6 Parameter determination:

2.6.1 Pump constant:

The relationship between the pump voltage and flow rate is determined by measuring the volumetric flow rate in a graduated cylinder, where:

$$\mathbf{F}_i\ (\mathbf{cm}^3/\mathbf{s}) = \mathbf{Kp}_i(\mathbf{cm}^3/\mathbf{s.v}) * \mathbf{V}_i(\mathbf{v}) \qquad \mathbf{2.26}$$

Where:

F_i **:** The flow rate (cm^3/s) .
Kp_i **:** The pump constant $(cm^3/s.v)$.
V_i **:** The pump voltage (v) .

For different pump voltages, the time for the liquid to move from 0 ml to 100 ml was recorded in table 2.1 below:

Table 2.1: Experimental data relating pump voltage and flow rate.

Voltage(v)	3	4	5	6	7	8	9	10	11	12
Time(s)	16.745	11.255	8.475	6.783	5.97	4.868	4.412	4.064	3.673	3.324
Flow Rate $(\mathbf{cm^3/s})$	5.972	8.885	11.799	14.742	17.271	20.576	22.575	24.63	27.248	30.120

By using the data in table 2.1, we obtain a graph shown in figure 2.9 of the flow rate versus pump voltage.

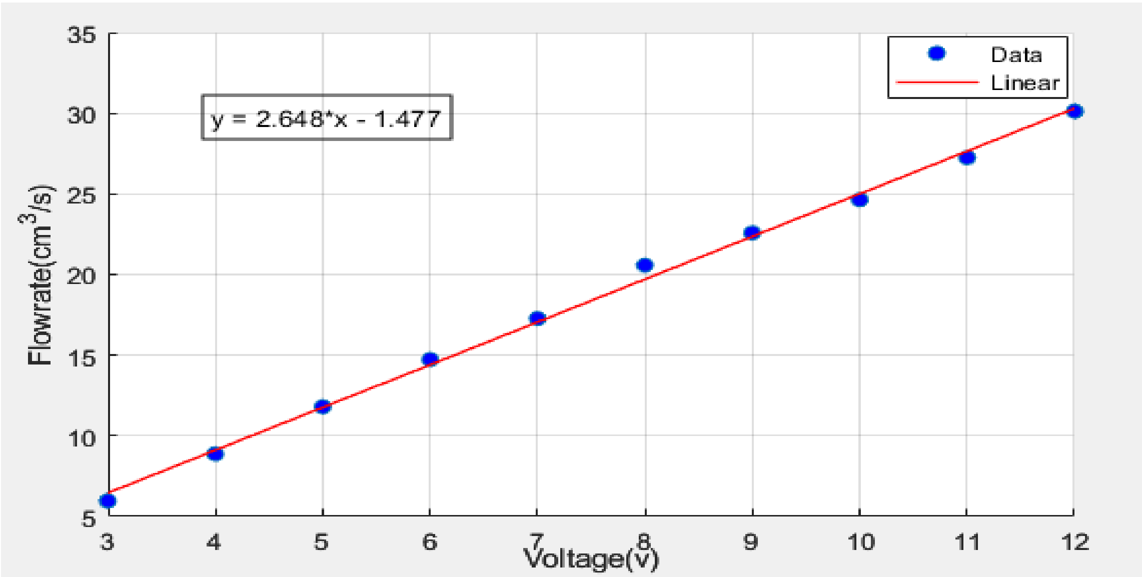


Figure 2.9: Pump flow constants for recorded data.

Using linear regression with the least squares method, the derived linear equation for the given data is:

$$\mathbf{F}_i\ (\mathbf{cm}^3/\mathbf{s}) = \mathbf{2.65} * \mathbf{V}_i(\mathbf{v}) \qquad \mathbf{2.27}$$

2.6.2 Orifice coefficient:

The orifice coefficient, also known as the discharge coefficient is a dimensionless number that characterizes the flow of fluid through an orifice. The coefficient is crucial in tank procedures to accurately determine the flow rate through an orifice.

The theoretical flow rate $\boldsymbol{Q_t}$ through an orifice is given by Torricelli's Law can be described by the following equation:

$$\boldsymbol{Q_t = A_{oi}\sqrt{2gL_i}} \quad \textbf{2.28}$$

Where:

$\boldsymbol{Q_t}$ is theoretical flow rate.

$\boldsymbol{L_i}$ is the water level.

$\boldsymbol{A_{oi}}$ is the cross-sectional area of the orifice.

g is the acceleration due to gravity.

The actual flow rate $\boldsymbol{Q_a}$ can be measured experimentally by collecting and measuring the volume of liquid discharged over a known period of time. To determine the actual flow rate, we follow a specific procedure:

- Fill the tank into a different levels with a closed orifice.
- Open the orifice and measure the time it takes for the water level to decrease from a specific starting level to a lower level.

The coefficient of discharge $\boldsymbol{C_d}$ is the ratio of the actual flow rate to the theoretical flow rate:

$$\boldsymbol{C_d = \frac{Q_a}{Q_t}} \quad \textbf{2.29}$$

Using the given equation 2.29, the orifice coefficient values are tabulated in table 2.2 below.

Table 2.2: Orifice coefficient values for each tank.

$\mathbf{C_{d1}}$	$\boldsymbol{C_{d2}}$
0.2609	0.4074

2.7 Coupled tank model parameters:

The table 2.3. Below, lists and characterizes the main parameters (mechanical and electrical specifications, convertion factors, constants) associated with the two tank specialty plant.

Some of these parameters can be used for mathematical modelling of the coupled tank system as well as to obtain the numerical equation of motion (EOM) for both tanks.

Since the open-loop transfer function of the coupled tanks system is given above by (2.24) and the unknown open-loop system parameters are obtained. Using the parameters given in table below 2.3.

Table 2.3: Physical coupled tanks system model parameters.

Symbol	**Description**	**Value**	**Unit**
K_P	Pump flow constant	2.65	cm3 /s.v
V_{Pmax}	Pump peak voltage	12	v
L_{1max}	Tank 1 height (i.e. maximum water level)	18.00	Cm
L_{2max}	Tank 2 height (i.e. maximum water level)	18.00	Cm
D_{t1}	Tank 1 inside diameter	6.000	Cm
D_{t2}	Tank 2 inside diameter	6.000	Cm
D_{o1}	Tank 1 outflow orifice diameter	0.685	Cm
D_{o2}	Tank 2 outflow orifice diameter	0.5	Cm
A_{t1}	Tank 1 inside cross-section area	28.27	Cm^2
A_{t2}	Tank 2 inside cross-section area	28.27	Cm^2
A_{o1}	Tank 1 outlet area	0.368	Cm^2
A_{o2}	Tank 2 outlet area	0.196	Cm^2
P_{range}	Pressure sensor range	0 – 40	kPa
g	Gravitational constant	981	Cm/s^2

2.8 Liquid level using a pressure sensor:

2.8.1 Modelling liquid level in a tank:

To model the level of the liquid in a tank is to measure the pressure and level relationship between the generated pressure from the sensor and the real level readings from the tank.

It is possible for a pressure sensor to measure the current level of any liquid. Here is an example setup, shown in figure 2.10 where the pressure sensor is placed at the top of tanks.

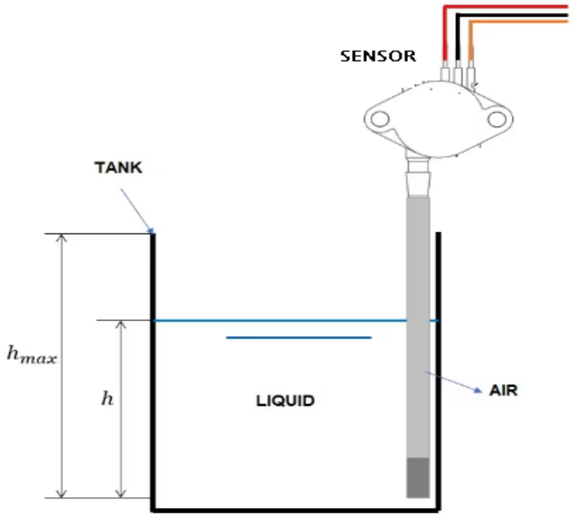


Figure 2.10: Pressure sensor placement. [17]

As the liquid level rises, the pressure inside the tube on the sensor increases. The height of the liquid is proportional to pressure with the formula:

$$\boldsymbol{P = \rho h g} \qquad 2.30$$

Where (ρ) is the density of the liquid, (h) is the liquid height and (g) is the acceleration due to gravity.

By measuring the pressure (P) at the top of the tank and knowing the liquid density (ρ), we can rearrange the formula to solve for the liquid height (h):

$$\boldsymbol{h = \frac{P}{\rho g}} \qquad 2.31$$

2.8.2 Calibration procedure for the pressure sensors:

The purpose of this procedure is to accurately calibrate the pressure sensors in tanks to measure liquid levels. Calibration ensures that the pressure readings correspond to the actual liquid levels, which is crucial for precise monitoring and control in various applications.

1. Ensure that each tank to be calibrated is completely empty. If any liquid is present, empty the tank entirely before proceeding.
2. Observe the pressure reading on the serial monitor with the empty tank.
3. Record this initial pressure value $\boldsymbol{P_0}$, which represents the baseline pressure of the empty tank.
4. Pump water into tank 1 until it is full, ensuring a consistent and even filling process.
5. Use a plugging device (such as a finger or a stopper) to seal the orifice of Tank 1.
6. Observe the pressure reading on the serial monitor with the tank full of water, record this final pressure value $\boldsymbol{P_1}$.
7. Establish a linear relationship between the pressure differential and the water level height to calculate the height of the water column in centimeters (cm),

$$\boldsymbol{h = \frac{1}{\rho g}(P_1 - P_0) + h_{offset}} \quad \textbf{2.32}$$

2.9 Configurations of coupled tanks system:

A single coupled tanks system can be used to set up different types of experiments, as described below in figure 2.11. Each of these configurations results in a unique control problem.

2.9.1 Configuration 1 Single Input Single Output (SISO):

In this system, the pump feeds into tank1 and tank2 is not used at all. A controller is designed to regulate or track the level in tank1. Different level can be tried for tank1.

2.9.2 Configuration 2 State-coupled SISO system:

In this system, the pump feeds into tank1 which in turn feeds tank2. A controller is designed to regulate or track the level in tank2. Different level can be tried for tank2.

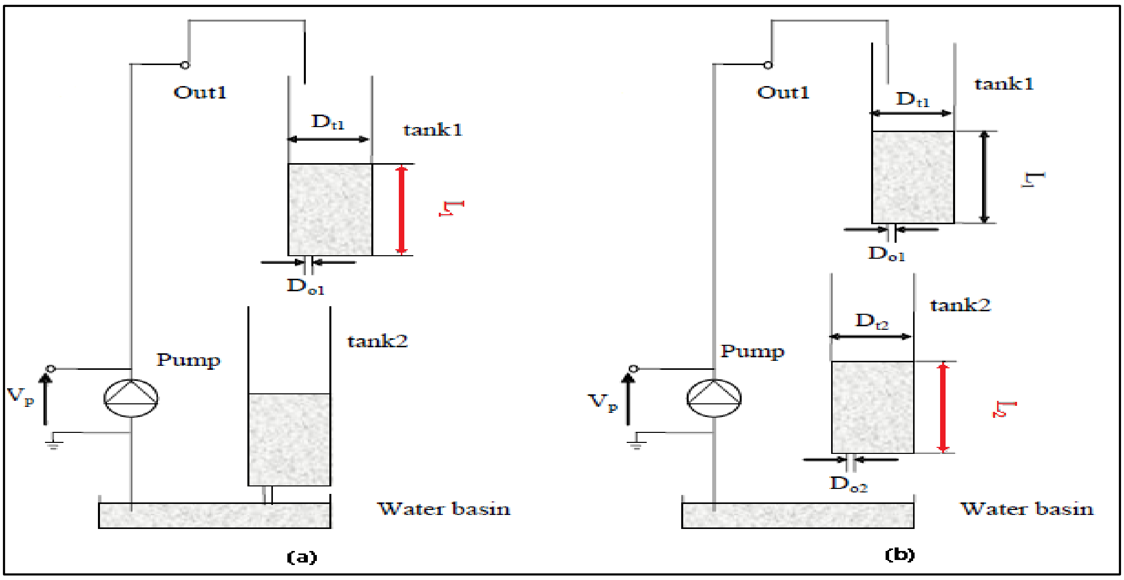


Figure 2.11: (a) SISO Configuration, (b) State-Coupled SISO Configuration.

2.10 Flowchart of project:

Figure 2.12 shows the overall progress for both software and hardware development, which will discuss later. This project will be divided to two parts to make sure this project run smoothly. The first part is a software part, which will cover modeling the controller. The controller must be designed and simulated using MATLAB and Simulink.

The second part will cover for the hardware part. In this part, the coupled tank liquid level system will be assembled to make sure it will run properly.

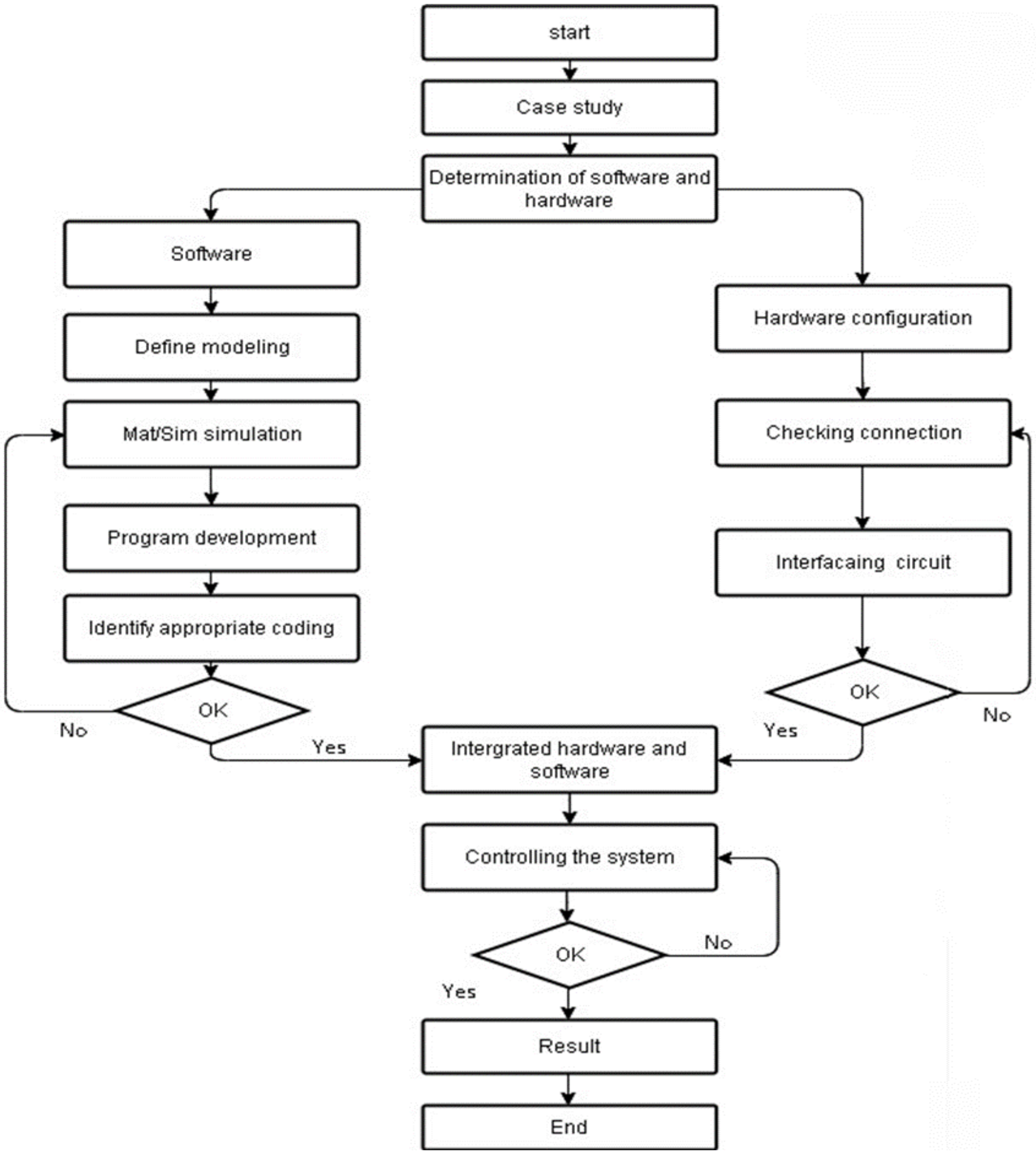


Figure 2.12: Flow chart for software and hardware development.

2.11 **Conclusion:**

In this chapter, the non-linear mathematical model of the coupled tank process is developed and then linearized using the Taylor series. Additionally, the open-loop transfer function of the system is provided. The modeling of the system involves creating differential equations to characterize the system response.

At the end of this chapter, the parameters of the system, such as the pump flow constant, tank dimensions, and orifice coefficients, are determined to develop the tanks model. The model parameters are essential for both the mathematical modeling and the numerical equations of motion for the tanks.

- CHAPTER 3 -

Simulated PID Controller
For
Coupled Tank Process

3.1 Introduction:

This chapter demonstrates the use of MATLAB and Simulink for modeling, analysis, and control design, with a particular emphasis on the functions provided by the control system toolbox.

3.2 MATLAB Simulink description:

MATLAB and Simulink form a powerful duo that helps engineers and scientists solve complex problems. MATLAB is a programming platform that shines in numerical computing, data analysis, algorithm development, and visualization.

Simulink complements MATLAB as a graphical tool for building models of dynamic systems. With its intuitive drag-and-drop blocks representing real-world components, Simulink allows users to test and analyze their models simulated before moving to physical construction.

This close integration between MATLAB computational capabilities and Simulink visual modeling makes them invaluable in various engineering and scientific fields. Whether it is control systems, robotics, aerospace, or automotive design, these tools are essential for tackling the challenges in these industries. [18]

3.3 Simulink blocks description:

This part presents a general block of a Simulink system, detailing the various parameters required for system control and analysis. These parameters include the set point value, scope, PID controller, data logging, and coupled tanks model.

The designing of PID controller to control coupled tanks system using MATLAB software. This software is used to create the Simulink model for PID Controller. The performances of PID Controller are evaluated in terms of overshoot, rise time and steady state error.

Then, the gain for each parameter will also be tuned in this software and the validity for each parameter will be compared using the reference value (set point). Figure 3.1 shows the MATLAB, Simulink block for PID Controller combines with coupled tanks model.

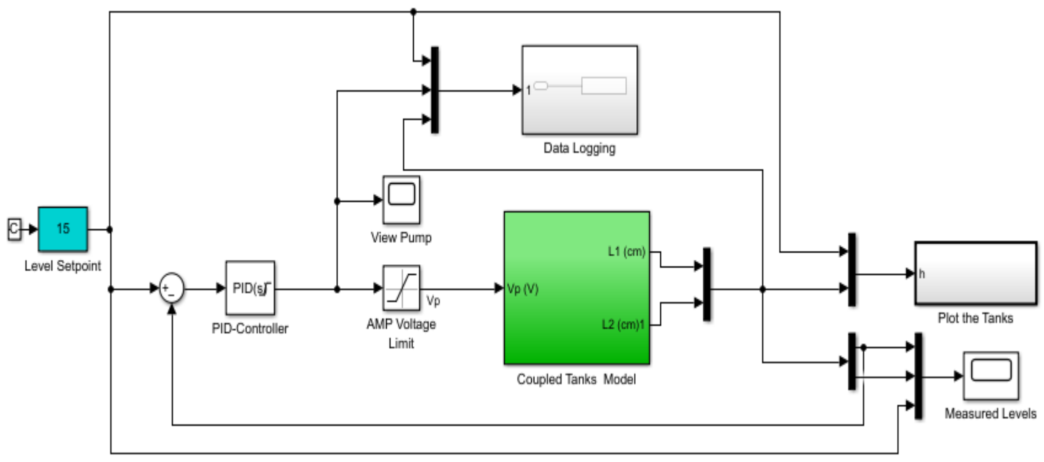


Figure 3.1: Simulink schema to simulate closed loop with PID controller.

Using the coupled tanks model in Simulink requires a script file, in this case, two have been used. The 1st one is to initialize the tanks parameters used in the program. The 2nd is for plotting the tanks.

It is possible to observe the response through the animation created, which shows the level in both tanks in blue varying over time, as well as the desired reference in red (figure3.2).

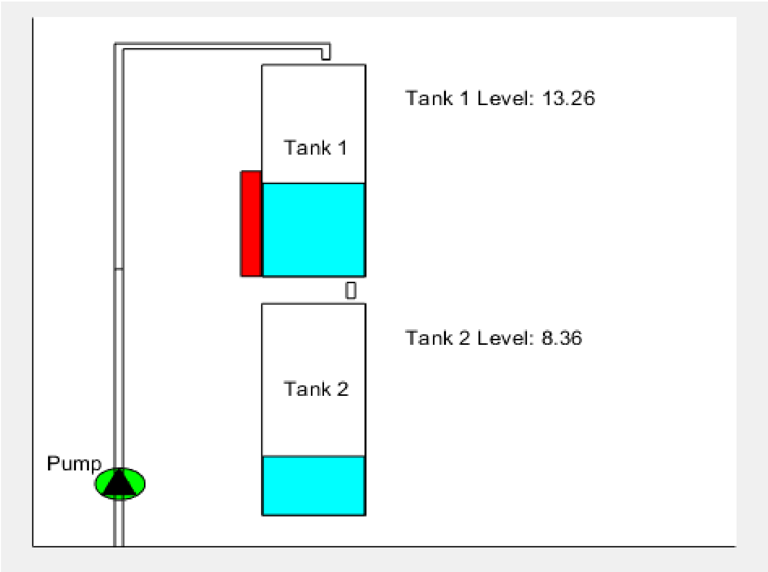


Figure 3.2: Coupled tanks system animation in Simulink.

3.4 MATLAB PID tuner procedures:

The MATLAB PID tuner uses a linear model to compute the PID gains. The coupled tanks is a nonlinear system. Typically, we would linearize the nonlinear model about a certain operation point, we can instead use the Simulink linearization tools to find the linear model.

To linearize and tune a PID controller, we first define the coupled tank model block. We load the model linearizer application as shown on figure 3.3 and ensure the analysis. We change the operating point to control a specific parameter at a desired value (10cm and 5cm in our case).

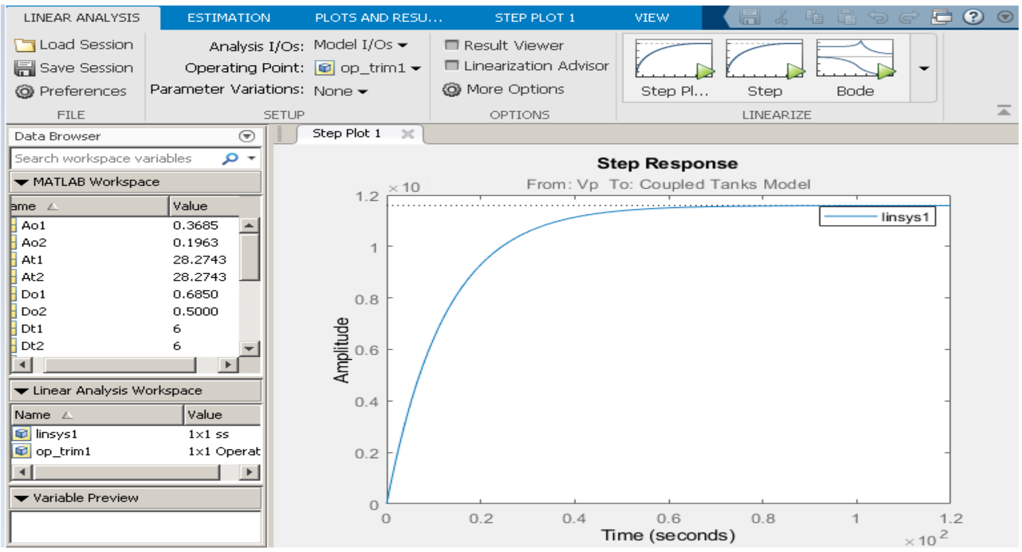


Figure 3.3: MATLAB linearizer application.

In the PID Tuner, we import the linear model variable and view the Step Plot to see the block response and tuned response (Figure 3.4). We set design goals such as overshoot, settling time, and steady-state error. We adjust the PID tuner settings, including response time and transient behavior sliders, to achieve the desired response. We update the PID Controller block gains in the Simulink model to the tuned values. [19]

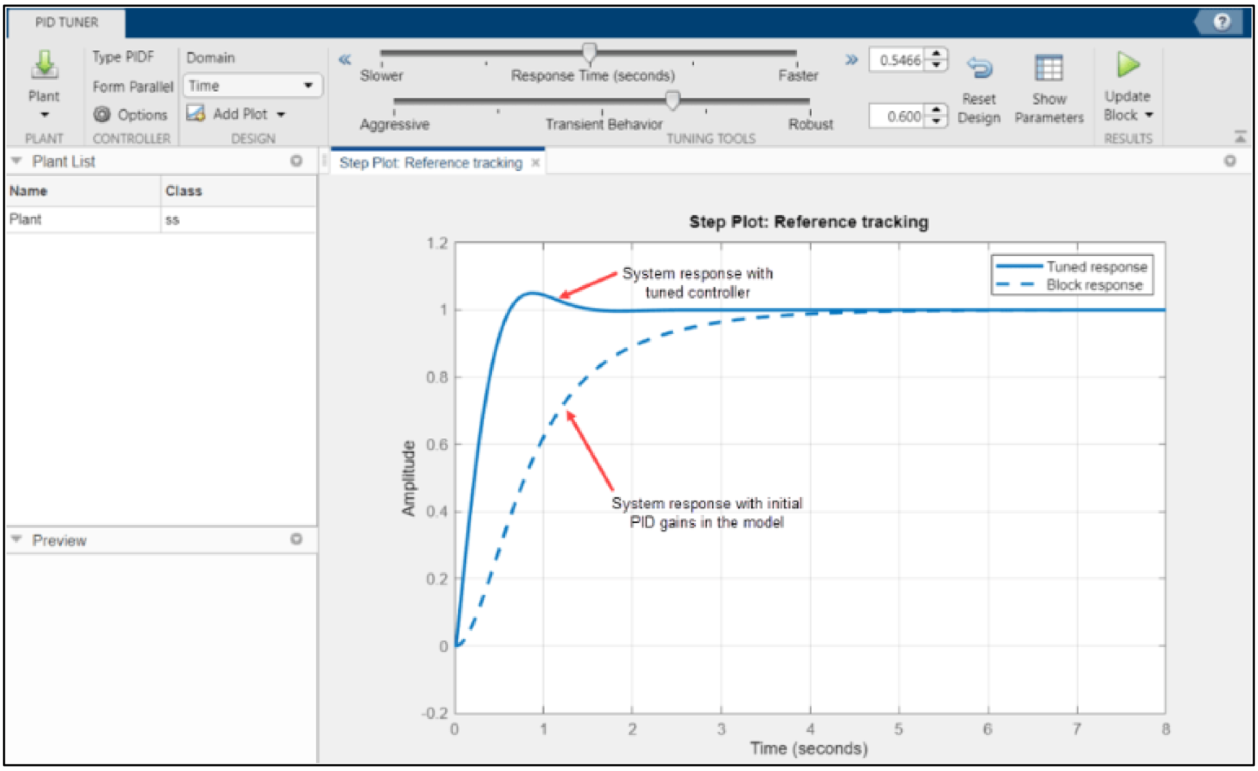


Figure 3.4: MATLAB PID tuner application.

If the response of the initial controller design does not meet the requirements, we can interactively adjust the design. PID Tuner gives two options for refining the controller design:

1. The Response Time slider to make the closed-loop response of the control system faster or slower.
2. The Transient Behavior slider to make the controller more aggressive at disturbance rejection or more robust against plant uncertainty.

3.5 Analysis and results:

By running the previous programs and simulating it with different set points as follow:

Table 3.1 cases for experiments

	Cases	**Operating point**
Configuration1	Case 1	10 cm
	Case 2	5 cm
Configuration2	Case 1	10 cm

3.6 **Simulation results:**

The linear model was tested on environmental of Simulink, MATLAB whereas PID have been simulated under some circumstances such as one or different step input signals. The following results shown on figures bellow which illustrates the closed loop response of system. Note that the graph with red line represents the set point, while blue line represents the closed loop system response and the green line the controller output.

3.6.1 Configuration 1 (Tank 1):

At this point, we present the proposed control strategy. The main control objective is to maintain the liquid level in tank 1 at desired level by adjusting the pump flow rate.

3.6.1.1 Case 01:

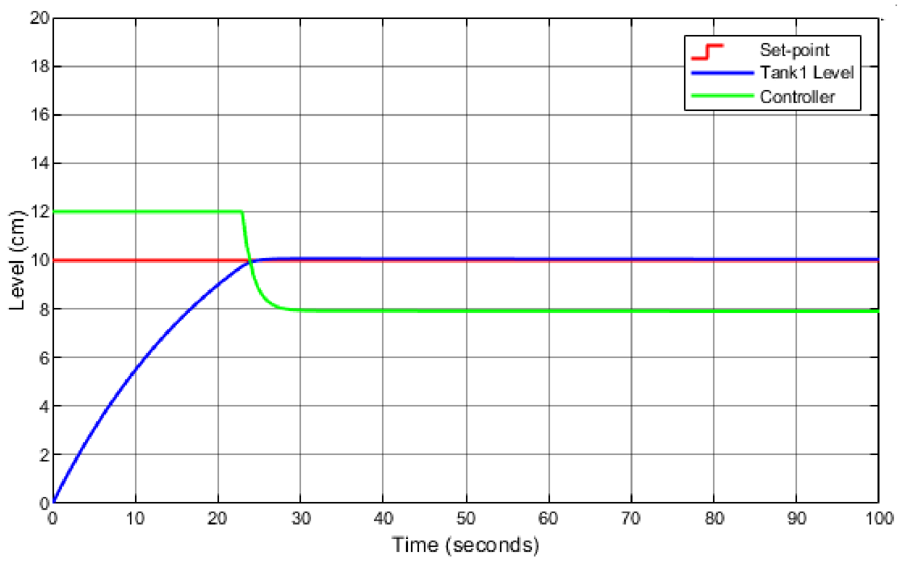


Figure 3.5: PI controller result with single step (case 1).

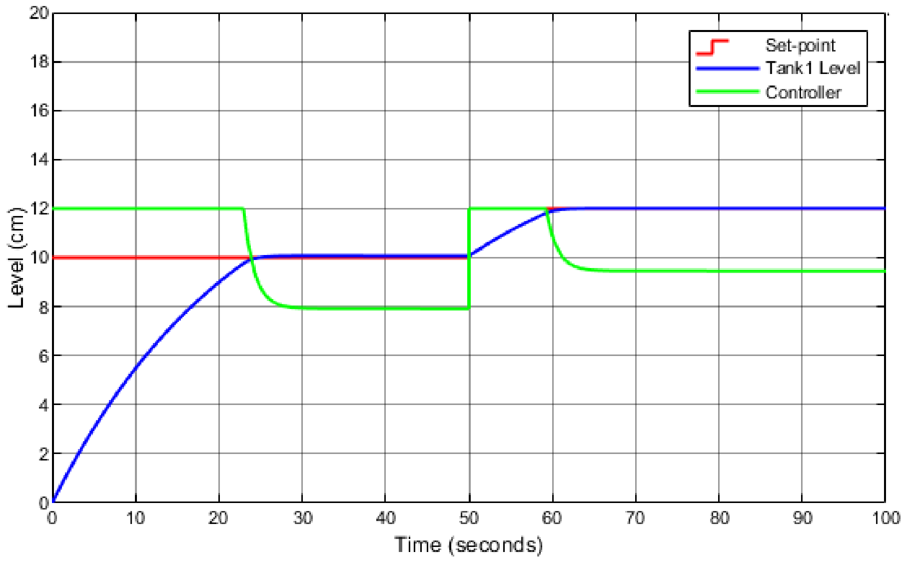


Figure 3.6: PI controller result with different step (case 1).

Figure 3.5: It represents the closed loop response of tank 1. From t =0 to t = 100 s, tank 1 desired level is set to 10 cm, the rise time of the system is 18.53s. The output achieves the set point value at time equals to 25.47 s.

Figure 3.6: Same as figure 3.5 from t = 0 to t = 50 s but from t = 50 to t = 100 s, the set point is changed from 10 cm to 12 cm. Clearly the output signal of the controller increase in order to increase the speed of the pump and that result in increasing the water level to the desired point then it settles. The proportional gain is set equal to 12.8326 and integral gain is set equal to 0.089968.

3.6.1.2 Case 02:

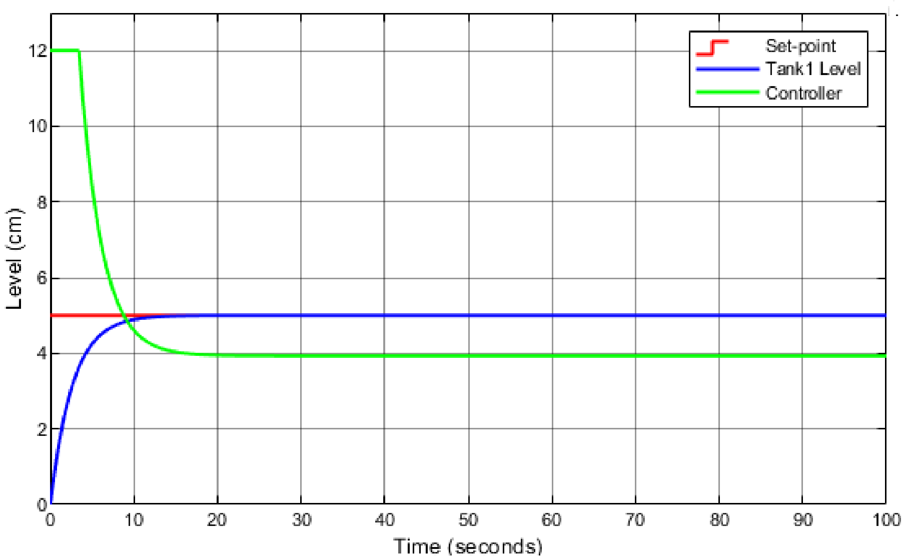


Figure 3.7: PI controller result with single step (case 2).

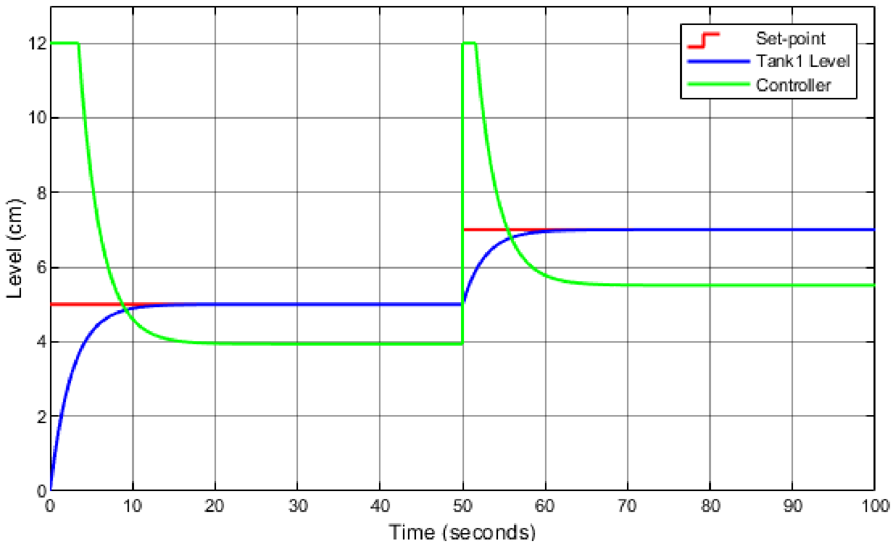


Figure 3.8: PI controller result with different step (case 2).

Figure 3.7: It represents the closed loop response of tank 1. From t = 0 to t = 100 s, tank 1 desired level is set to 5 cm, the rise time of the system is approximately 5.83 s. The output achieves the set point value at time equals to 10.296 s.

Figure 3.8: same as figure 3.7 from t = 0 to t = 50 s but from t = 50 to t = 100 s, the set point is changed from 5cm to 7 cm. Clearly the output signal of the controller increases in order to increase the speed of the pump and that result in increasing the water level to the desired point then it settles. The proportional gain is set equal to 6.712 and integral gain is set equal to 0.2991.

3.6.2 Configuration 2 (Tank 2):

At this point, we present the proposed control strategy. The main control objective is to maintain the liquid level in tank 2 at the desired level by adjusting the pump flow rate. We should also consider water level in tank 1 to prevent overflow.

3.6.2.1 Case 01:

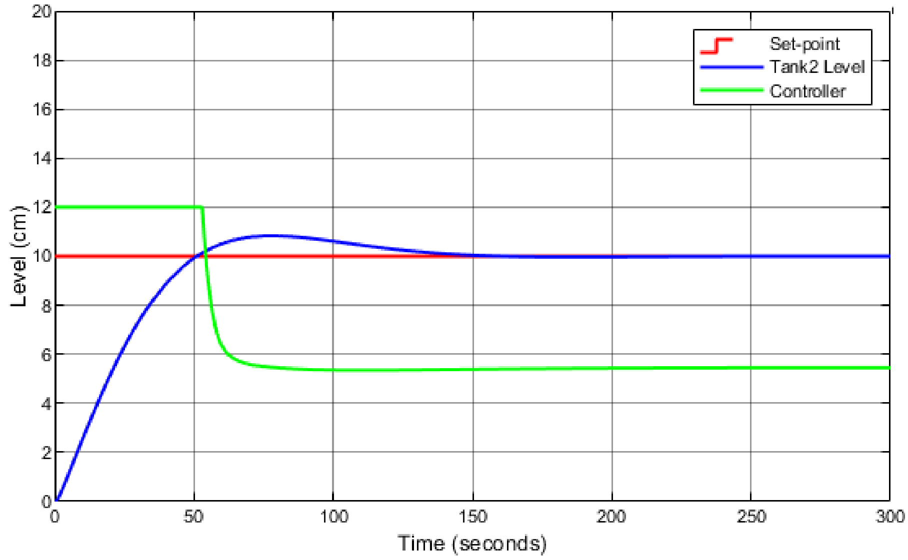


Figure 3.9: PID controller result with single step (case 2).

Figure 3.9: It represents the closed loop response of tank 2 From t =0 to t = 300 s, tank 2 desired level is set to 10 cm, the rise time of the system is approximately 32.2 s with overshoot of 7.20%. The output achieves the set point value at time equals to 132.56s. Clearly the output signal of the controller at maximum in order to increase the speed of the pump and that result in increasing the water level until reach the set point for the first time then decrease to settles at the desired point. The responses have overshoot because of the influence of upper tanks, but it is not really high, even knowing that the step applied is wide.

The proportional gain is set equal to 1.562, integral gain is set equal to 0.03302 and derivative gain is set equal to 13.91.

3.7 **Conclusion:**

The result was obtained using PID tuner of MATLAB and this result was very acceptable for our application. Thus complying with our design goals. Given that we have a set of gains matching our requirements in simulation, therefore we can apply this for physical implementation.

In this chapter, we presented the dynamic modeling of coupled tanks systems, this modeling is studied with linear model, followed by a simulation test to illustrate the dynamic behavior of the system. Simulation tests were conducted to determine the best tuning parameters for the PID controller that are tabulated in table 3.2.

Table 3.2: PID values for the simulation model

	Cases	**P**	**I**	**D**
Configuration1	Case 1	12.8326	0.089968	/
	Case 2	6.712	0.2991	/
Configuration2	Case 1	1.562	0.03302	13.91

-CHAPTER 4-
Physical PID Controller
For
Coupled Tank Process

4.1 Introduction:

In the following section, we introduce the three cases with the different PID controller parameters which we used with the physical process and their given results.

The process involves a closed loop system setup with a coupled tanks system, where the level must be controlled using a feedback control loop.

The measured inputs to the designed PID will be provided by a level sensor (pressure sensor). The designed PID controller will generate the necessary control signal, which will then be acquired by the arduino board and motor shield to control pump in other word the level of the tanks.

4.2 Simulink blocks description:

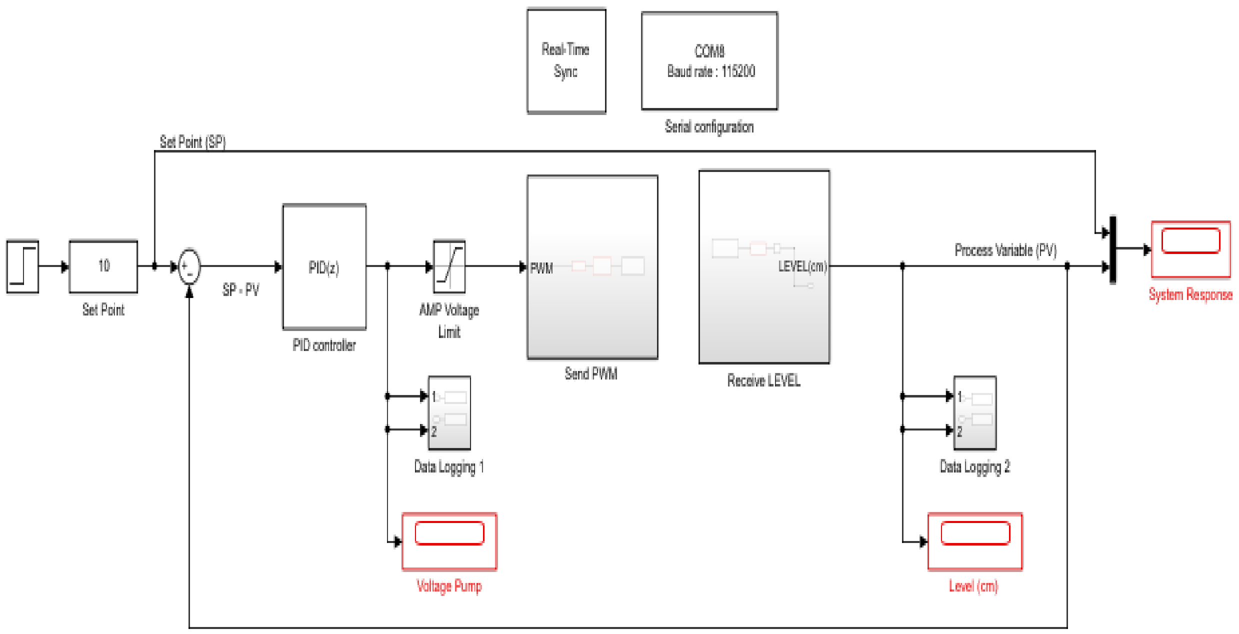


Figure 4.1: Simulink real time model.

The block diagram depicts a feedback control system, possibly for a liquid level control application. The system employs a proportional-integral-derivative (PID) controller to regulate the level of a liquid in a tank.

The PID controller compares a desired set point (SP) with a measured process variable (PV), which is the actual liquid level. The difference between the (SP) and (PV), termed the error signal, is processed by the PID controller. The PID controllers output manipulates a control signal that regulates the voltage supplied to a pump.

The pump speed is adjusted based on the control signal, by influencing the flow rate of the liquid into the tank. This feedback loop continues until the measured level (PV) aligns with the desired set point (SP).

Data logging components are incorporated within the system to record the set point, the measured level, and the system's response. This data can be vital for monitoring and analyzing the performance of the control system.

4.3 Analysis and results:

After simulation tests, the following step is to test the PID control laws in the real process. It is important for the right identification and modeling of system, steps are introduced around the equilibrium point, in order to test the performance of the system.

4.3.1 Configuration 01 results:

At this point we present the proposed control strategy. The main control objective is to maintain the liquid level in tank 1 at desired level by adjusting the pump flow rate.

4.3.1.1 Case 01:

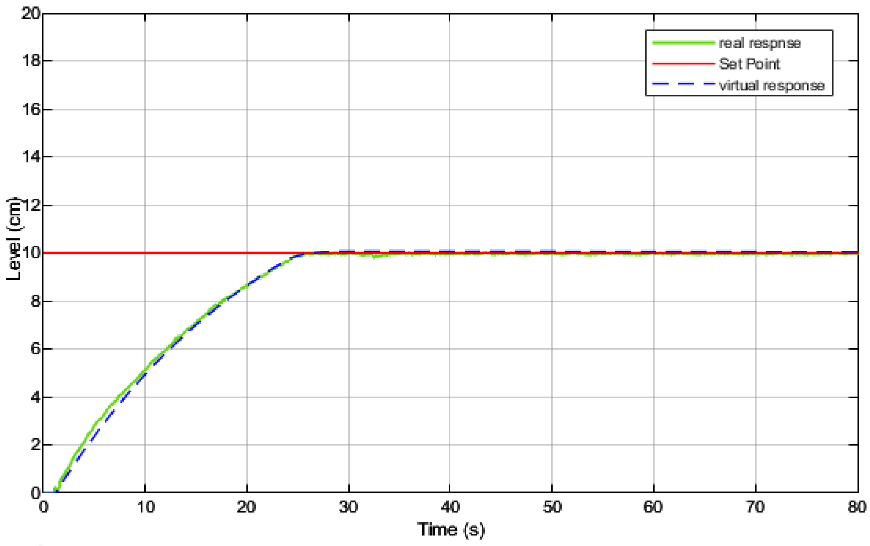


Figure 4.2: Comparison of tank1 Level response between experimental data and Simulink simulation (case 1).

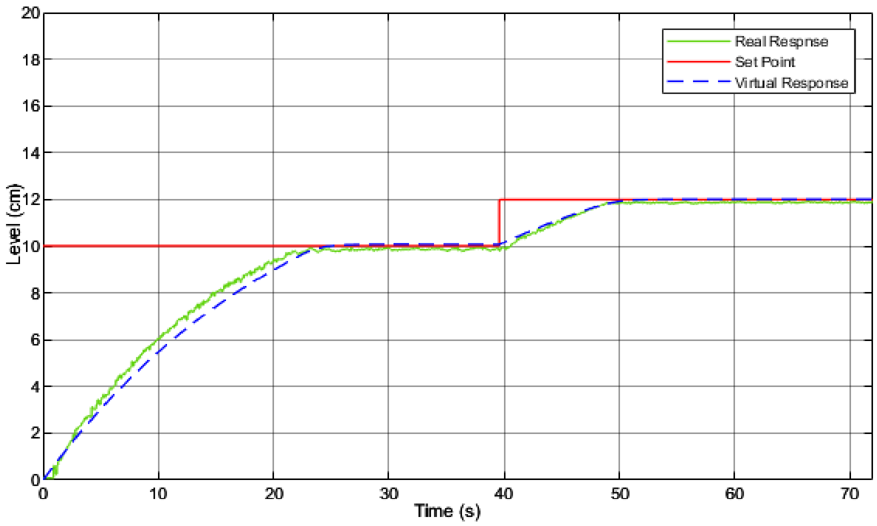


Figure 4.3: Comparison of tank1 level response between experimental data and Simulink simulation (different steps).

As it can be observed in figure 4.2 and figure 4.3 the experimental response of the system is approximately similar to the simulated one. Analyzing the system behavior:
- Steady state error of approximately 0 is reached in both.
- The overshoot is almost what it was expected from the simulation. Percentages of 0% are obtained for both.
- Settling times of 25.47 seconds in simulation output, and 24.23 seconds for real system.
- Rise times of 18.53 seconds in simulation output, and 17.9 seconds for real system.

4.3.1.2 Disturbance case 1:

If we applied amount of disturbance, we would obtain result as shown in figure.4.4 that the system response under PI controller returned to the reference point after a short period where the disturbance happened to the system. This confirms that our control is robust.

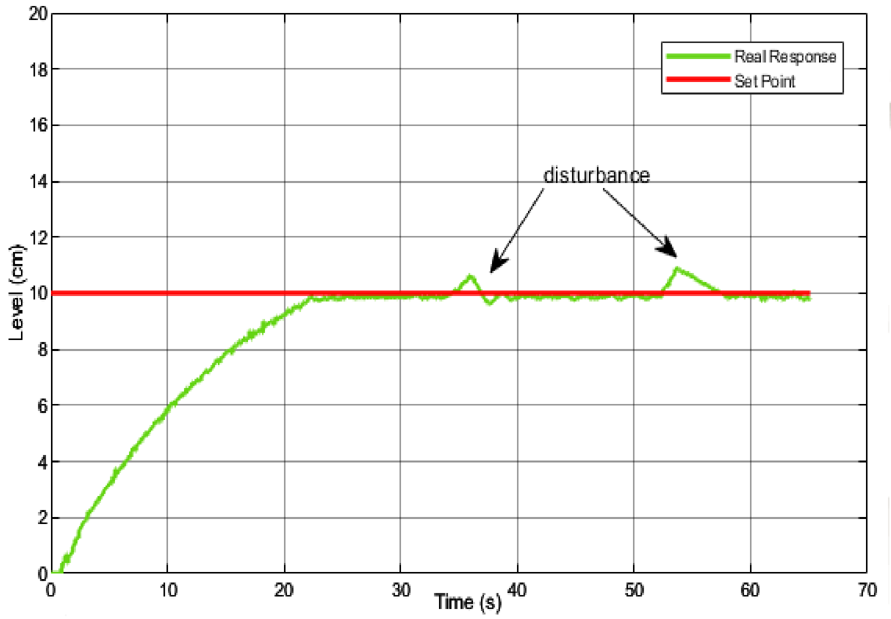


Figure 4.4: Experimental PI controller result with disturbance (case 1).

Here is a summary of the results shown in table 4.1 for the coupled tanks simulation using linearized model and the hardware coupled tanks system.

Table 4.1: Comparison of PI controller performance (case 1).

DESIGN GOALS	SIMULATION	HARDWARE
PID Gains	P = 12.8326 I = 0.089968 D = 0	P = 12.8326 I = 0.089968 D = 0
Percent overshoot (%)	0%	0%
5% Settling time (S)	25.47s	24.23 s
Steady-State Error (CM)	0.0cm	0.05 cm
Rise time (S)	18.53s	17.9s

4.3.1.3 Case 02:

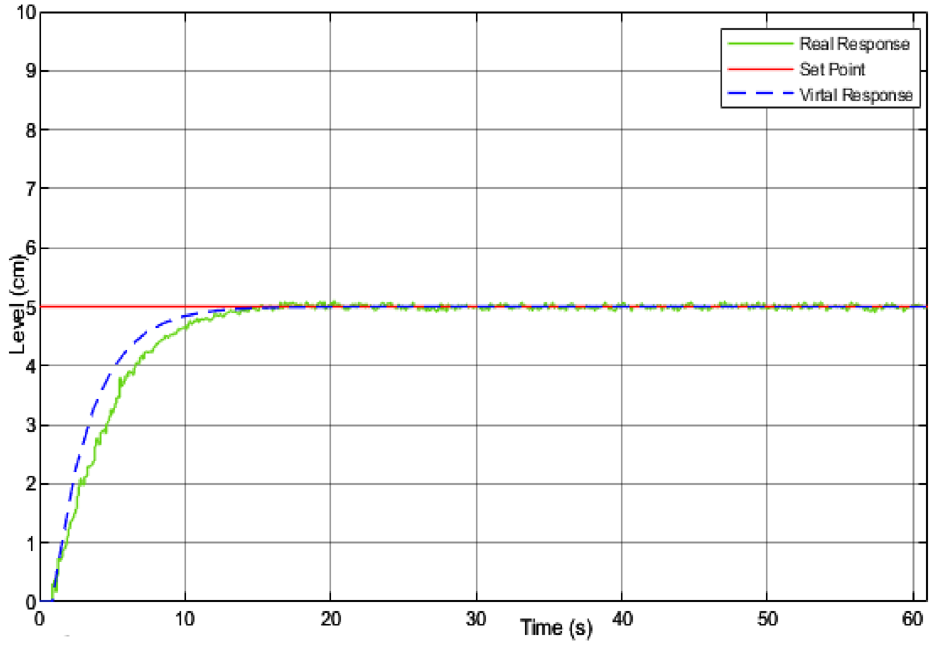


Figure 4.5: Comparison of tank1 level response between experimental data and Simulink simulation (case 2).

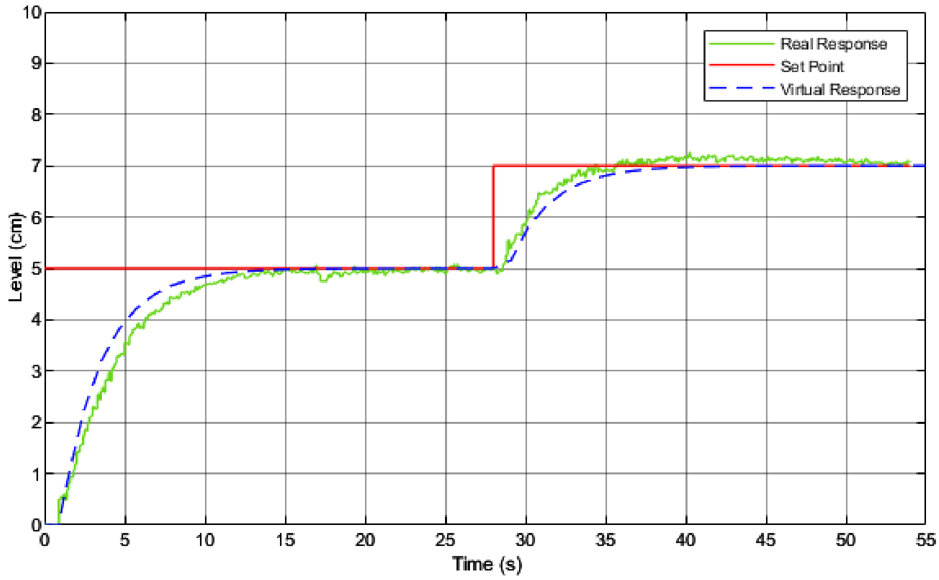


Figure 4.6: Comparison of tank1 level response between experimental data and Simulink simulation (different steps).

As it can be observed in figure 4.5 and figure 4.6, the experimental response of the system is approximately similar to the simulated one. Analyzing the system behavior:

- Steady state error of 0 is reached in simulation, and 0.104 cm for real system.
- The overshoot is almost what it was expected. Percentages of 0% are obtained for both implementation.

- Settling times of 10.296 seconds in simulation output, and 12.1 seconds for real system.
- Rise times of 5.83 seconds in simulation output, and 6.4 seconds for real system.
There is slight difference, but this also happens because of experimental error.

4.3.1.4 Disturbance case 2:

If we applied a disturbance, we would obtain result as shown in figure 4.7 that the system response under the PI controller, which returns to the reference point shortly after the disturbance. This indicates that our control is robust, effectively mitigating the disturbance and maintaining stability and performance.

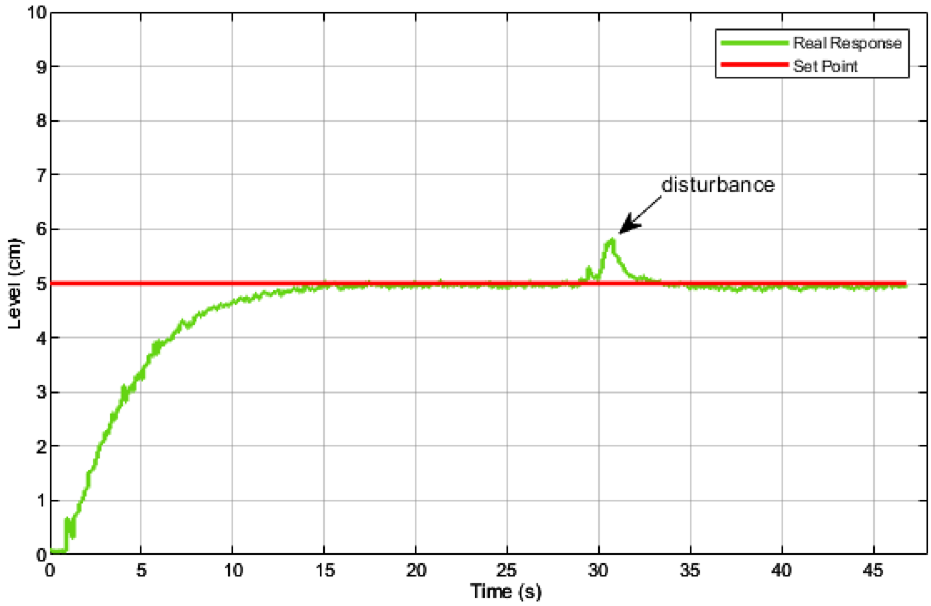


Figure 4.7: Experimental PI Controller result with disturbance (case 2).

Here is a summary of the results shown in table 4.2 for the coupled tanks simulation using linearized model and the hardware coupled tanks system.

Table 4.2: Comparison of PI controller performance (case 2).

DESIGN GOALS	SIMULATION	HARDWARE
PID Gains	P = 6.712 I = 0.2991 D = 0	P = 6.712 I = 0.2991 D = 0
Percent overshoot (%)	0%	0%
5% Settling time (S)	10.296s	12.1 s
Steady-State Error (CM)	0.0 cm	0.104 cm
Rise time (S)	5.83s	6.4s

4.3.2 Configuration 02 results:

At this point, we present the proposed control strategy. The main control objective is to maintain the liquid level in Tank 2 at the desired level by adjusting the pump flow rate. Additionally, we need to consider Tank 1 to prevent it from overflowing.

4.3.2.1 Case 01:

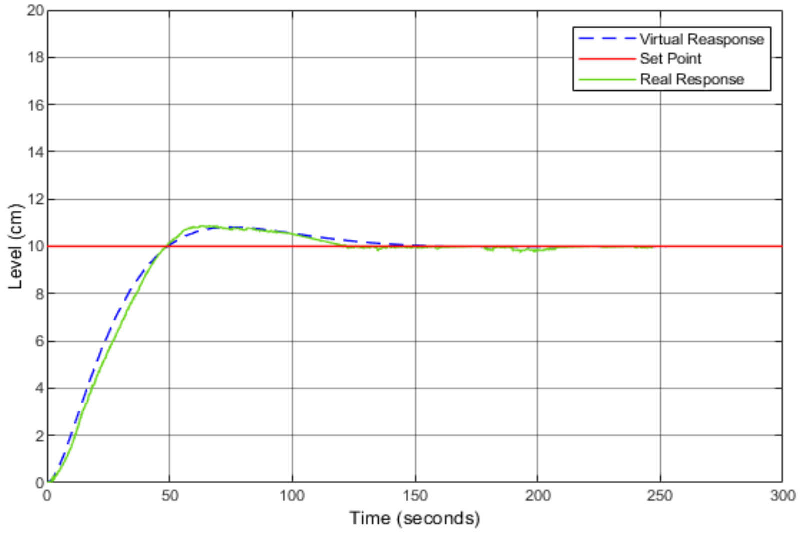


FIGURE 4.8: Comparison of tank 2 level response between experimental data and Simulink simulation (case 1).

As it can be observed, in figure 4.8, the real response of the system is approximately similar to the simulated one. Analyzing the system behavior:

- A steady-state error of 0.09 is reached in the simulation, while the real system has a steady-state error of 0.138 cm.

- The overshoot in the real system is similar to the expected overshoot from the simulation, with percentages of 7.20% for the simulated output and 8.14% for the real output. These are higher than the designed values, likely due to the influence of the upper tanks.

- The settling times are 132.56 seconds for the simulation and 129.89 seconds for the real system.

- The rise times are 32.2 seconds for the simulation and 33.7 seconds for the real system.

- The responses exhibit overshoot due to the influence of the upper tanks, but it is not significant, especially considering the large step input applied.

- The slight differences between the simulated and real systems can be attributed to experimental error.

4.3.2.2 Disturbance case 1:

If we applied amount of disturbance, we would obtain results as shown in figure.4.9 that the system response under PID controller returned to the reference point after some period after the disturbance happened to the system since it is not big. Due to the derivative sensitivity to disturbances, filtering is necessary.

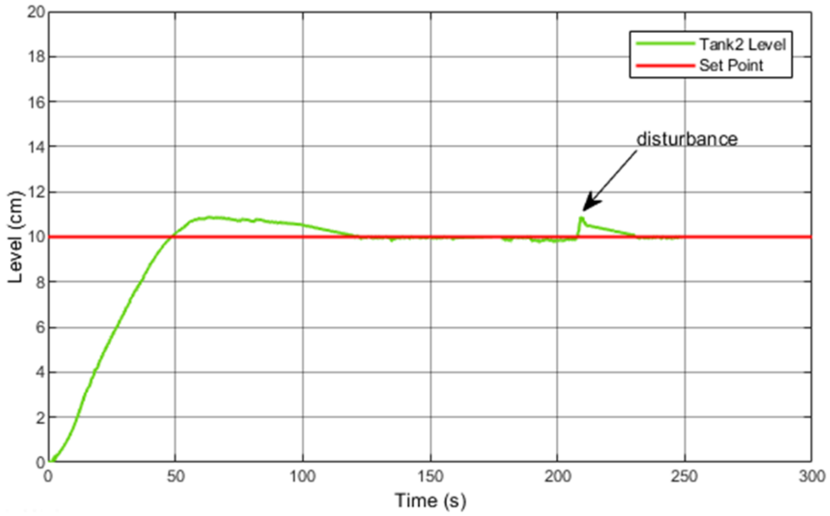


Figure 4.9: Experimental PID Controller result with disturbance (case 1).

Here is a summary of the results shown in table 4.3 for the coupled tanks simulation using linearized model and the hardware coupled tanks system.

Table 4.3: Comparison of PID controller performance (case 1).

DESIGN GOALS	SIMULATION	HARDWARE
PID Gains	P = 1.562 I = 0.0330 D = 13.91	P = 1.562 I = 0.0330 D = 13.91
Percent overshoot (%)	7.20%	8.14%
5% Settling time (S)	132.56 s	129.89 s
Steady-State Error (CM)	0.09cm	0.138 cm
Rise time (S)	32.2s	33.7 s

4.4 Discussion of results:

After comparing the simulated and experimental results, the closeness in performance measures such as settling time, rise time, and steady-state error further supports the accuracy of our model.

The small differences we see might be due to practical issues in experimental tests. For example, sensor hysteresis can cause slight delays and inaccuracies in measurements, which do not occur in the idealized simulation. Non-linearity in the system components can also create differences from the expected behavior. Other factors, like slight variations in the fluid, can affect the system response.

Despite these factors, the control strategy shows robustness, maintaining good performance under both normal conditions and disturbances. This confirms that our model effectively captures the main dynamics of the system, providing a reliable basis for control design and testing.

4.5 Conclusion:

This chapter dealt with the PID control of the coupled tank process real implementation. The obtained results are discussed and compared to the simulated ones. At the end of this chapter, a comparison between the PID control in simulation and real implementation results is presented.

The laboratory two tanks process was studied and simulated in the "control design and simulation" toolkit of MATLAB. The process under PID controller was analyzed. The physical implementation of the controller was carried out using arduino board.

For step changes the PID controller gave good result for both simulation and real implementation. Thus, better performance and robustness can be guaranteed.

GENERAL CONCLUSION

This project demonstrates the control of a coupled tanks system using MATLAB, Simulink software, and an arduino board, providing an accessible and cost-effective solution.

The MATLAB Simulink environment is utilized for modeling the coupled tanks system, offering robust tools for the design, testing, and development of control strategy in a simulated environment prior to real implementation. The arduino board functions as the physical controller, receiving data from tank level sensors, executing control algorithm such as PID control, and sending control signals to pumps to maintain desired liquid levels.

This project serves as a valuable reference for control engineers seeking an accessible and adaptable control system. It enhances both educational experiences and practical applications across various industrial contexts.

Future work will focus on enhancing control capabilities by implementing advanced algorithms, such as fuzzy logic control techniques for configurations 1 and 2, and comparing the results with those obtained using the PID controller. Additionally, we can explore more challenging configurations, such as the simultaneous control of both tanks, ensuring synchronized operation and maintaining consistent liquid levels. These enhancements will further improve the system applicability and performance in diverse industrial settings.

REFERENCES

[1] Prof. Karl Åström and Prof. Karl Henrik Johansson, «www.made-for-science.com,» [En ligne]. Available: "https://www.made-for-science.com/de/quanser/?df=made-for-science-quanser-coupled tanks-usermanual.pdf".

[2] «wikipedia,» [En ligne]. Available: https://en.wikipedia.org/wiki/State-space_representation.

[3] A. F. NAVARRO, «Security Analysis of a Wireless Quadruple,» May 2011.

[4] «wikipedia,» [En ligne]. Available: https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller.

[5] «www.ctrlaltftc.com,» [En ligne]. Available: https://www.ctrlaltftc.com/the-pid-controller/the-proportional-term.

[6] «/www.ctrlaltftc.com,» ctrlaltftc, [En ligne]. Available: https://www.ctrlaltftc.com/the-pid-controller/the-integral-term.

[7] «controlstation.com,» [En ligne]. Available: https://controlstation.com/blog/derivative-affect-pid-controller-performance/#:~:text=Derivative%20is%20the%20third%20term,and%20Set%20Point%20(SP)..

[8] Prof. Karl Åström and Prof. Karl Henrik Johansson, «www.made-for-science.com,» [En ligne]. Available: https://www.made-for-science.com/de/quanser/?df=made-for-science-quanser-coupled tanks-coursewarestud-labview.pdf.

[9] N. W. I. A. Shehu, «Applications of MPC and PI controls for liquid level control in coupled tank systems,» chez *IEEE International Conference on Automatic Control and Intelligent Systems*, 1 October 2016.

[10] «rajguruelectronics.com,» [En ligne]. Available: https://rajguruelectronics.com/ProductView?tokDatRef=MTc2MTQ=&tokenId=NzI=&product=DC%20water%20Pump%20R385%20New.

[11] «www.electroschematics.com,» [En ligne]. Available: https://www.electroschematics.com/pressure-sensor-guide/ .

[12] «rs-online,» [En ligne]. Available: https://docs.rs-online.com/ee06/0900766b80db99b0.pdf.

[13] «Rajguru Electronics,» [En ligne]. Available: https://5.imimg.com/data5/PX/UK/MY-1833510/l293d-based-arduino-motor-shield.pdf.

[14] O. Y. Ismael, «Introduction to process control (part1),» 2020.

[15] D. T. SEKBAN, «Physical Application of Sliding Mode Controller for Coupled Tank,» *International Journal Applied Mathematics,* 2016.

[16] D. Qamar, «Sliding Mode Control for Coupled tank Liquid Level Control System,» chez *10th International Conference on Frontiers of Information Technology*, 2012.

[17] «/community.element14.com,» 2019. [En ligne]. Available: https://community.element14.com/challenges-projects/design-challenges/liquid-level-monitoring-using-a-pressure-sensor.

[18] «mathworks,» 2019. [En ligne]. Available: https://www.mathworks.com/matlab-simulink-overview-61216.html.

[19] M. L. S. A. ENGINEER, «www.quanser.com/,» 27 APRIL 2021. [En ligne]. Available: https://www.quanser.com/blog/mathworks-blog/leveraging-the-simulink-control-design-toolbox-for-the-quanser-coupled tanks-system/.

APPENDIX A: Pressure sensor calibration.

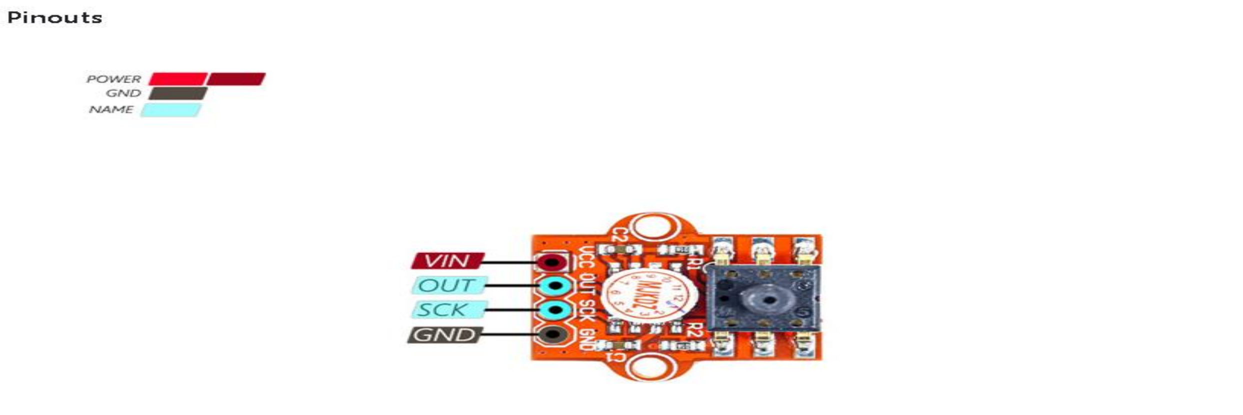
Technical Specifications:

VIN: Module power supply (3.3v or 5 v).

GND: Ground.

SLC: I2C Clock.

OUT: Digital output data.



The code sets up two HX710 sensor objects to read data from pressure sensors. It then continuously reads sensor values, converts them to pressure and liquid level, and transmits those readings over serial communication for monitoring.

➢ Initializing the input and output pins of the Arduino board.

calibration_pressure_sensors §

```
#include "HX710.h"

const int DOUT1 = 52;
const int PD_SCK1 = 50;
const int DOUT2 = 48;
const int PD_SCK2 = 46;

double level1, pressure1;
double level2, pressure2;

HX710 sensor1;
HX710 sensor2;

#define SERIAL_PLOTTER

void setup() {
    Serial.begin(115200);
    sensor1.initialize(PD_SCK1, DOUT1);
    sensor2.initialize(PD_SCK2, DOUT2);
}
```

➢ Converting the sensor output from ADC bits to pressure in Pascals, then to level in centimeters.

```
void loop() {
    int32_t value1, value2;

    // Reading from sensor 1
    while (!sensor1.isReady());
    sensor1.readAndSelectNextData(HX710_DIFFERENTIAL_INPUT_40HZ);
    value1 = sensor1.getLastDifferentialInput();
    pressure1 = (5.960 * 10E-06 * value1 - 50);
    level1 = (pressure1 / (9.81 * 997) * 1000);
    level1 = (3.44 * level1 - 3.00) ;

    // Reading from sensor 2
    while (!sensor2.isReady());
    sensor2.readAndSelectNextData(HX710_DIFFERENTIAL_INPUT_40HZ);
    value2 = sensor2.getLastDifferentialInput();
    pressure2 = (5.960 * 10E-06 * value2 - 50);
    level2 = (pressure2 / (9.81 * 997) * 1000);
    level2 = (3.125 * level2 - 2.67 ) ;
```

➢ Displaying the pressure value for each sensor along with their corresponding levels.

```
    Serial.print("level1 is: ");
    Serial.println(level1);
    Serial.print("pressure1 is: ");
    Serial.println(pressure1);

    Serial.print("level2 is: ");
    Serial.println(level2);
    Serial.print("pressure2 is: ");
    Serial.println(pressure2);

    delay(1000);
}
```

APPENDIX B: Script files for MATLAB Simulation.

The following program is the MATLAB file script for initialize the Coupled Tanks Model.

Initialize.m

```
% Tank 1 Height (i.e. Maximum Water Level) (cm)
L1_MAX = 20;
% Tank 2 Height (i.e. Maximum Water Level) (cm)
L2_MAX = 20;
% Tank 1 Inside Diameter (cm)
Dt1 = 6;
% Tank 2 Inside Diameter (cm)
Dt2 = 6;
% Medium Outflow Orifice Diameter (cm)
Do1 = 0.685;
% Medium Outflow Orifice Diameter (cm)
Do2 = 0.5;
% Tank 1 Inside Cross-Section Area (cm^2)
At1 = pi * Dt1^2 / 4;
% Tank 2 Inside Cross-Section Area (cm^2)
At2 = pi * Dt2^2 / 4;
% Tank 1 Outlet Area (cm^2)
Ao1 = pi * Do1^2 / 4;
% Tank 2 Outlet Area (cm^2)
Ao2 = pi * Do2^2 / 4;
% Pump Flow Constant (cm^3/s/V)
Kp = 2.65;
% Gravitational Constant (cm/s^2)
g = 981;

```

The following program is the MATLAB file script for obtain open loop transfer function for tank1 and tank2.

```
function [G1] = tank1_model(L10,Ao1,At1,Kp,g)
    % Laplace Representation of the Tank #1 system: G1 = L11/Vp1
    % DC gain [V/cm]
    Kdc1 = Kp * sqrt( 2 * L10 / g ) / Ao1;
    % time constant [s]
    tau1 = At1 / Ao1 * sqrt( 2 * L10 / g );
    % open-loop TF: G1 = L11/Vp1
    G1 = tf( Kdc1, [tau1 1]);
end
function [G2] = tank2_model(L10,L20,Ao2,At2,Ao1,g)
    % Laplace Representation of the Tank #2 system: G2 = L21/L11
    % DC gain [V/cm]
    Kdc2 =  Ao1 * sqrt( L10 / L20 ) / Ao2;
    % time constant [s]
    tau2 = At2 / Ao2 * sqrt( 2 * L20 / g );
    % open-loop TF: G2 = L21/L11
    G2 = tf( Kdc2, [tau2 1]);
end

```