HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ENGINEERING GRADUATION THESIS

Design controller for MIMO system using Arduino board and MATLAB Simulink software

NGUYEN THANH DO

do.nt150952@sis.hust.edu.vn

Automation and electrical system – K60

Instructor: **MSc. Cao Thanh Trung**

Instructure's sign:___________

Department: **Automation**

School: **School of electrical engineering**

HÀ NỘI, 8/2022

MINISTRY OF EDUCATION AND TRAINING

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SOCIALIST REPUBLIC OF VIETNAM

Independence – Liberty - Happiness

GRADUATION THESIS
CONTRIBUTION

Full name: Nguyễn Thành Đô Student ID: 20150952

School year: K60

School: Electrical & Electronic Engineering

Major: Control and automation

1. Topic: **Design controller for MIMO system using Arduino board and MATLAB Simulink software**
2. Thesis contents:
    - Object recognition
    - Object modelling
    - Controller design
    - Control circuit design
3. Supervisor: MSc. Cao Thanh Trung
4. Thesis starting date: 10^{th} April 2022
5. Thesis completion date: 5^{th} August 2022

Month……Date……2022

SUPERVISOR

ACKNOWLEDGEMENTS

In way to become a scientist, the first step of research is really important. Starting with a technology lover, then became a student of Ha Noi University of Science and Technology. I feel myself so lucky to learn, study in an environment that has a depth of academic curriculum; a width of the science subjects, and carries lots of inspiration for future researchers. I thank the school for enabling me to meet many scientists, excellent and dedicated teachers; also gave me lots of valuable experiences at school.

With the most respectful words and my deep gratitude, I would like to send my gratefulness to MSc. Cao Thanh Trung, who directly taught and instructed me, for his enthusiastic help and his valuable knowledge so that I can complete this thesis. A great thanks for your support and your encouragement to motivate me in the journey to be a genuine scientist.

Finally, I’d like to thank my family; friends for encouraging and inspiring me to finish this thesis.

THESIS SUMMARY

Process control plays an important role in control and automation field. In reality, automation processes are complicate, which requires controller for Multi Input Multi Output (MIMO) objects.

Research object in this thesis is a MIMO system which is built in water tank form. Input is hot and cold water flow rate, temperature are measured by sensors and water flow are controlled by pneumatic valves. Output are temperature and level of water in the tank.

This thesis focuses on examine, identify the object, study and design iterative learning controller to control parameters of the system stays closely to set value.

In this thesis, i use Matlab Simulink for object recognition, simulation; Simulink is connected with Arduino I/O to collect data from water mixing tank system in C9-311 laboratory.

This thesis contains 5 chapters:

Chapter 1: System introduction

Chapter 2: Object model

Chapter 3: Design controller for the system

Chapter 4: Design control circuit

Chapter 5: Conclusion

Thesis Author
(Sign, write full name)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. SYSTEM INTRODUCTION

This chapter focuses on the systematic description of the tank system used in the thesis, which provides the goal, the task of the problem of controlling water levels and temperature.

1.1 Multi Input Multi Output (MIMO) watertank system

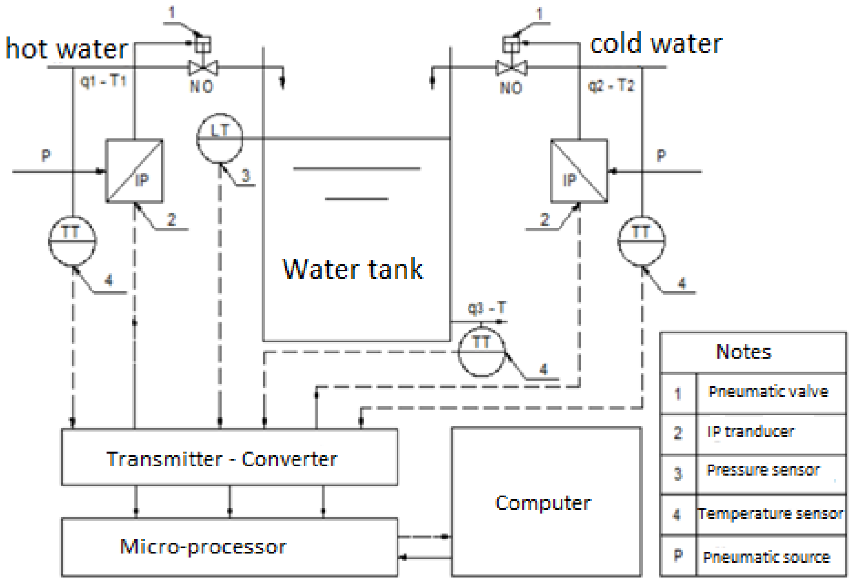

Figure 1.1 Water tank system

<u>Water tank system:</u>

The MIMO system is a multiple input-output system. The control object of the water tank is a system with two inputs and two outputs. The problem is to control the temperature and water level in the tank. Hot and cold water flows are drawn from two overhead tanks. Manipulated variables include hot water flow and cold water flow. Controlled variables are temperature and water level in the tank. The control system consists of the control object, the microprocessor, the transducers, the measuring sensors and the control valve.

1.1.1 Necessities of the stability of level and temperature

At present, the cause of industrialization and modernization is growing strongly, the progress of science and technology, in which automatic control technology also greatly contributes to creating conditions to improve efficiency in the process. production process.

In industries producing liquids such as chemicals, bottled water, milk, fish sauce, cooking oil, etc., the problem of level and temperature control needs to be met with high accuracy to serve the production process. achieve better efficiency, ensure uninterrupted fluid production, increase equipment life. The operator does not need to directly check in the tanks or open and close the pump continuously, the problem of emptying or overflowing in the liquid tank is completely overcome and the temperature in the tank is stable even if the output changes. change. That's why we need to "stabilize water level and temperature".

1.1.2 Application

The system of stabilizing water level and temperature is widely applied in the fields of Industry, Agriculture, in many companies, factories and factories such as: petrochemical industry, chemical industry, processing industry. water treatment, paper production, power generation, sewage tanks... The application of water level and temperature control adjusts the water level in the tank to reduce excess water in the tank as well as ensure the quality of the liquid inside.

There are many methods to monitor and manage fluids such as manual methods, automation methods. And today most of them use the automation method.

Automation solutions to reduce human labor, increase production efficiency in terms of cost as well as accuracy.

Some special application of level and temperature control in industry: Petrochemical refining technology, beverage factory, ...

1.1.3 Control methods

a) Linear control system

In this thesis, we only consider linear control system.

When investigating the kinematics of a control object or a system, it is common for the investigated objects to be considered linear, which leads to the description of the system by a system of linear differential equations. Using the superposition principle of linear systems, we can easily separate the components specific to each working mode for research with rigorous, precise but simple and effective mathematical tools. Using a linear model to describe a system has many advantages such as:

- Simple model, linear model parameters are easily determined by experimental methods without having to go from complex physicochemical equations describing the system.
- Methods of synthesis of linear controllers are abundant and do not take much time to implement.
- The simple structure of the model allows easy monitoring of control results and the model can be reconfigured to suit actual requirements.
- Because of these advantages of linear models that linear control theory and linear models have acquired a wide range of applications.

b) Control methods

Up to now in practice, many methods and systems of flow control have been designed and used. In which the main control methods are:

- Using the traditional PID, from simple to complex, means to design an additional oscillator compensation controller besides the position controller. The PID controller will correct the difference between the desired set signal and the output response of the system, then output a control signal to adjust accordingly.
- Using advanced theoretical tools such as fuzzy logic and artificial neural networks
- Combine fuzzy logic with classical PID
- Combine artificial neural network with classical PID

Each method has its own advantages and disadvantages. All research methods are aimed at improving the quality of flow control. The classical method is applied to control objects with mathematical models. But in practice the control systems are nonlinear, high complexity.

The method of using PID controller is widely used to control objects according to the feedback principle. The reason PID is widely used is because of its simplicity in both structure and working principle. The PID unit is responsible for bringing the static error e of the system to zero so that the transition process satisfies the basic requirements for quality.

In this thesis, we use direct channel decoupling controller to control our system.

1.2 Description of the system

Preliminary description: the system consists of two hot water sources and cold water melted into a tank through two control valves. The tank contains a hand-operated valve at the bottom. Besides, the system has sensors that serve temperature and water levels. Specifically, there is a sensor that measures the water level of the tank and three PT100 sensors measuring hot and coldwater temperature and draining them from the tank. Two information about the system that we need to control is the temperature and height of water in the tank.

1.2.1 The tank

Rectangular, 18x18cm, 35cm.

Metal material.

At a height of 12 feet there is an overflow valve.

There's an exhaust valve at the bottom of the tank, maximum water flows around $4,7.10^{-5}$ m^3/s

1.2.2 Cold and hot water sources

Hot water is allocated from a cold thermos, with temperatures that can reach $75°C$, to the maximum of $4,28.10^{-5}$ m^3/s

The cold water has temperatures depending on the environmental temperature, in summer, the cold-water temperature is about $32°C - 35°C$, the maximum volume of about 2,28. 10^{-5} m^3/s

1.2.3 Valve

Figure 1.2 Pneumatic valve

Valve usually opens (NO), which is controlled by compression from I/P Samson 6102 adapter, traffic valve flow rate proportional to the valve opening from 0 - 100%

Besides, the system has two valves to adjust the amount of water in the tank.

1.2.4 I/P transducer

Figure 1.3 I/P Transducer

It' s working to convert electrical signals into compressed air signals.

Input is the current signal from 4 - 20mA.

Output is a compressed air signal that can be adjusted continuously in 0. 05 - 6.0

The air supply has a 0. 4 bar or 6 psi on the range above.

Linear characteristic (Output is a linear function with input)

1.2.5 Level sensor

Figure 1.4 KL76 sensor

The system uses kl76 sensors, which operates on the principle of pressure difference between the liquid at the bottom and the atmosphere

Voltage source 24VDC±10%

Output is the signal 4-20mA

Two - wire type

Relation between height and electric current is linear relations:

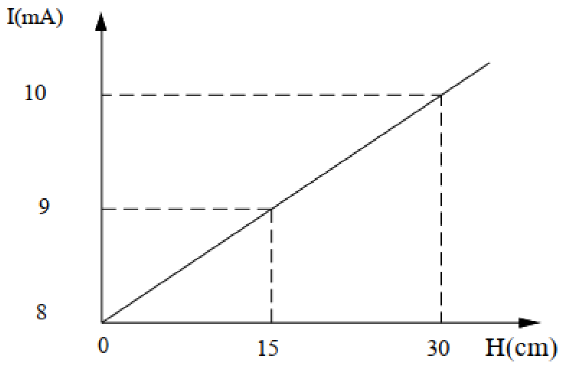

Figure 1.5 Relation between height and electric current of KL76

1.2.6 Temperature sensor

Figure 1.6 PT100 sensor

The material is platinum

The operation based on the principle of resistance of metals varies according to the change of temperature

Measuring ranges from -200 °C to 850 °C

With 0 - 100°C measuring line of Pt100 is linear which determined as follows:

$R_t = R_0(1 + 3{,}9083.10^{-3}t)$

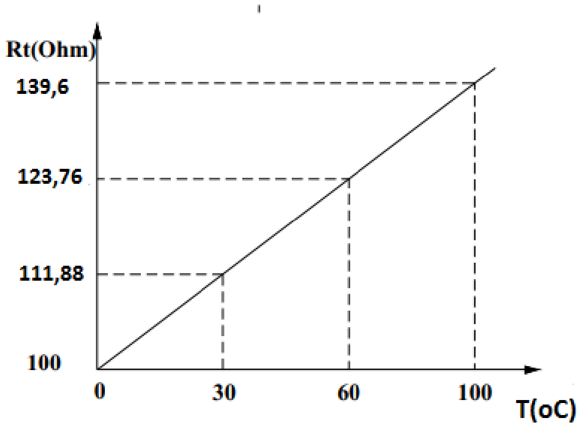

Figure 1.7 Relation between resistor and temperature of PT100

1.3 Control mission

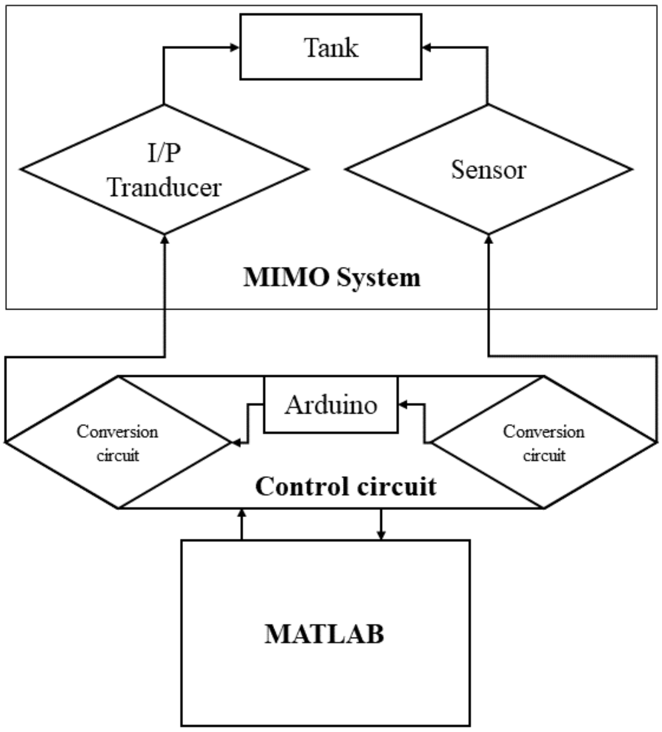

Figure 1.8 Overview of the system

For a system to control a good operation, it must satisfy the following requirements:

- Works stable, sustainable with system noise
- Devices that work smoothly all the time.
- Systems, safe with users

From there, we give the criteria for this system as follows:

- Signals that stably follow the set value.
- Fast response time
- Signals stable control, within the work structure
- Active hardware, reliable, safe
- Convenient software, easy to encapsulate, manage and expand when necessary

1.4 Requirements

The target of working in this scheme is to build a model of MIMO system that controls temperature and water levels.

Now, there's only the system, there's no control circuit and the control software. So, the job is set to two main parts:

- Hardware: The tank system is designed to include 3 RTD sensors that measure hot water temperature, cold water, water in tank and a pressure sensor that measures water levels. At the same time, two traffic valves are compressed air valves that can be controlled so the hardware circuit must have a function of reading sensor signals and signaling. The hardware circuit consists of three R/U conversion circuit for temperature sensors, one I/U conversion circuit for level sensor and two U/I circuit to change PWM signal generated by Arduino to current to control compressed air valves and Arduino. Each signal is collected by conversion circuit and connect to Arduino. This value is transferred to MATLAB to calculate the control signal. The control signal output in the form of voltage, through the conversion to the line.
- Software: the reading of the sensor and the control signal perform on Arduino. And the calculation of the controller is carried out on Matlab. MATLAB received from Arduino I/O a value that corresponds to the sensor value. Then from this value, MATLAB calculates the controller and converts PWM signal through conversion circuit to each valve. So, the software will take the calculation, display and communicate between the control and MATLAB.

1.5 Proceed steps

From the analysis of the system and the task of the control problem, we give the steps that need to be taken to build a complete level control system as follows:

Modelling: Build mathematical equations to describe objects. Using the kinematic equations of the level object described as nonlinear equations in the state space. From this model can apply control methods to the object. In addition to using kinematic equations, for modeling there is also an identification method for variable input to measure the output response of the system. From that information, Matlab's Identification tool can be used to identify objects.

Linearization: Bring the kinematic equation of the level object from the form of nonlinear equation to the form of a linear equation to facilitate the design of the controller later. There are two commonly used linearization methods: linearization around the working point and exact linearization. With the object of the problem under consideration is a level average with a relatively stable working point in the actual process, the method of linearization around the working point proved to be the most suitable and widely used in practice. This is also the method that we consider in this thesis.

Controller design: The design controller must ensure that the water level value adheres to the set value for a certain period of time, the deviation is small enough. The controller here includes both the nonlinear controller for the original system and the linear controller for the linearized system.

Hardware and software design:

To install the control algorithm for the system, it is necessary to have a control circuit including the following functions: read signal from the sensor, send it to the microcontroller to calculate and output valve control signal.

Besides, it is necessary to design a control software installed on Arduino.

CHAPTER 2. OBJECT MODEL

In previous chapter, we present details of the tank system used in the thesis. In this chapter, we will model the object of the vessel from the basic mathematical equations and then simulate and verify the resulting model.

2.1 Problem analysis

2.1.1 Choice of methods for building mathematical models

Due to the parameters of the system such as the (A) area, the thermal source temperature of the (TH) water source, the cold-water source temperature is easily defined by measurement, so we choose the method of modelling theory for analysis and building models.

This is a method of building models based on the basic laws of physics, and basic chemistry. This method is very appropriate to analyze and verify models and helps us understand the inner relations of the process directly related to physical, chemical, or biological phenomena. Besides, it also allows the relative determination of the structure of the model.

2.1.2 Process analysis

As described, the multi-system system here is a tank that has two currents flowing into the flow of traffic. These two drinks of water have different temperatures. At the bottom of the tank, the controller can't intervene in the operation. The task here is to control water levels and temperature in the container with the reservation.

<u>Process variables:</u>

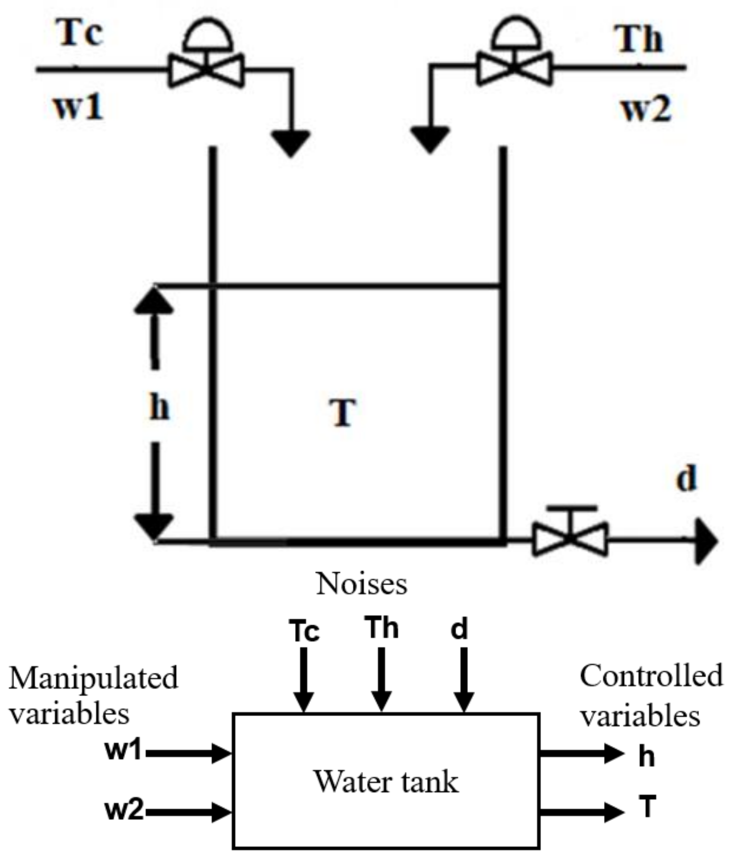

Figure 2.1 Process variables

- Controlled variables:

    Temperature T (°C)

    Water level in the tank h (m)

- Manipulated variables:

    Cold water flow: $\omega_l(m^3/s)$

    Hot water flow: $\omega_n(m^3/s)$

- Noise:

    The flow of water that flows to the: ω (m^3/s) is the passive value depending on the aperture of the valve and water levels. This value is determined according to Bernoulli' s formula as follows:

$$\omega = C.\sqrt{2g}A_0\sqrt{h} = k\sqrt{h} \quad \text{(Eq 2.1)}$$

With:

C is discharge coefficient,

A_0 is valve area,

g is gravity acceleration,

Hot water source temperature:

Cold water source temperature:

Flowing out flow: $\omega(m^3/s)$

2.1.3 Building the model equation

- Mass balance equation:

In a time unit, the volume of water in the tank is equal to the volume of water supplied minus the volume of water flowing out:

$$m = m_n + m_l - m_{out} \qquad \textit{(Eq 2.2)}$$

Therefore, $\rho A.\frac{dh}{dt} = \rho.\omega_n + \rho.\omega_l - \rho.\omega$

With:

ρ is water density (kg/m^3)

A is bottom area of water tank (m^3)

m is water mass inside the tank (kg)

m_n is supplied hot water mass(kg)

m_l is supplied hot water mass (kg)

m_{out} is flowing out water mass (kg)

From the equation above, we have:

$$A.\frac{dh}{dt} = \omega_n + \omega_l - \omega \qquad \textit{(Eq 2.3)}$$

- Energy balance equation

The tank is an object, the sum of energy into the equals equal to total energy output out of the tank and the energy change in the tank. In this scheme, we can skip the energy of the water, heat exchange with environment, ... So, we can say it again: the total amount of energy from the water flowing into the tank equals the total amount of water that flows out of the tank and the heat rises in tank

$$U = U_v - U_r \qquad \text{Eq 2.4}$$

$$\rightarrow \frac{d(C.\rho.AhT)}{dt} = C.\rho.\omega_n.T_n + C.\rho.\omega_l.T_l + C.\rho.\omega.T$$

$$\rightarrow \frac{d(AhT)}{dt} = \omega_n.T_n + \omega_l.T_l + \omega.T$$

With: C is heat capacity (J/kg.K)

$$\rightarrow AT.\frac{dh}{dt} + Ah.\frac{dT}{dt} = \omega_n.T_n + \omega_l.T_l + \omega.T$$

From Equation 2.3, we have:

$$T.(\omega_n + \omega_l - \omega) + Ah.\frac{dT}{dt} = \omega_n.T_n + \omega_l.T_l + \omega.T$$

$$\rightarrow Ah.\frac{dT}{dt} = \omega_n.(T_n - T) + \omega_l.(T_l - T)$$

Therefore, we receive 2 model equations:

$$\begin{cases} \mathrm{A}.\dfrac{\mathrm{dh}}{\mathrm{dt}} = \omega_{\mathrm{n}} + \omega_{\mathrm{l}} - \omega \\ \mathrm{Ah}.\dfrac{\mathrm{dT}}{\mathrm{dt}} = \omega_{\mathrm{n}}.(\mathrm{T_n} - \mathrm{T}) + \omega_{\mathrm{l}}.(\mathrm{T_l} - \mathrm{T}) \end{cases} \qquad \text{Eq 2.5}$$

Let $\underline{x} = (h\ T)^T$, $\underline{u} = (\omega_n\ \omega_l)^T$ and $\underline{y} = \underline{x}$, we have equation system describe water tank system below:

$$\begin{cases} \dfrac{d\underline{x}}{dt} = \begin{pmatrix} \frac{-\omega}{A} \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{A} & \frac{1}{A} \\ \frac{T_l - x_2}{Ax_1} & \frac{T_n - x_1}{Ax_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ \underline{y} = \underline{x} \end{cases} \qquad \text{Eq 2.6}$$

2.2 Model verification

2.2.1 Simulation

We conduct object simulation with flow of cold valve is $2.10^{-5}\ m^3/s$, hot valve is $2.10^{-5}\ m^3/s$ and outlet valve is $3{,}2.10^{-5}\ m^3/s$.

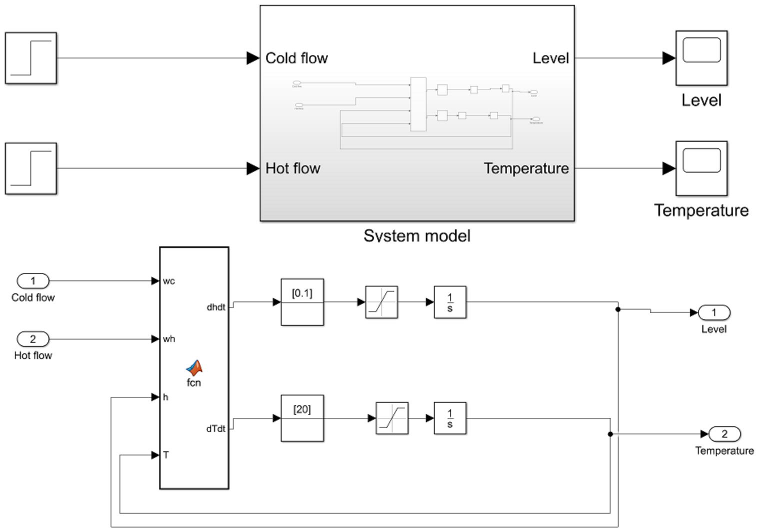

Figure 2.2 Modelling on MATLAB

Here, we assume:

The initial temperature of the tank is 20℃

The initial level of the tank is 0,1 m

Hot water temperature is 70℃

Cold water temperature is 20℃

Outlet flow is $3{,}2.10^{-5}\ m^3/s$

Both hot and cold water flow are equals $2.10^{-5}\ m^3/s$

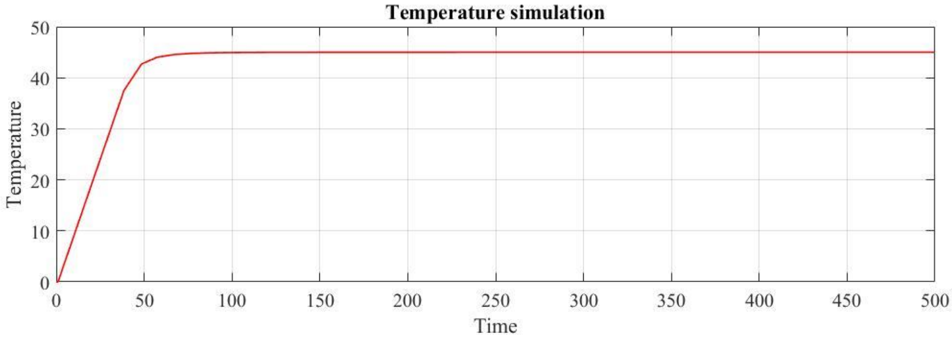

Figure 2.3 Temperature simulation

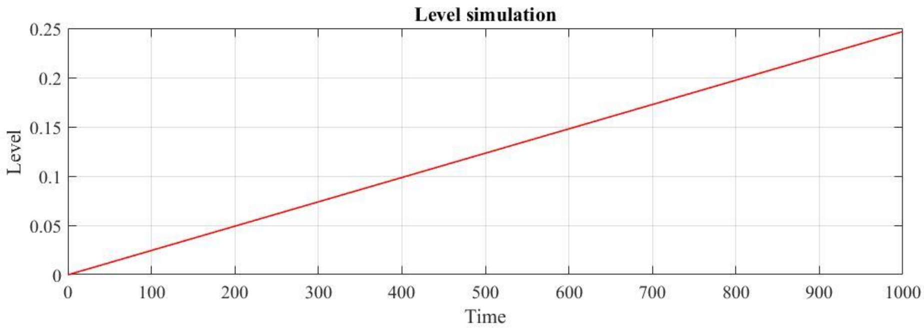

Figure 2.4 Level simulation

CHAPTER 3. DESIGN CONTROLLER FOR THE SYSTEM

3.1 Control analysis

3.1.1 Linearization around the working point

It can be said that linear control theory has developed close to a perfect level and provides us with quite enough tools to analyze and build a system. The linear control system achieves all required qualities. Desire to use that knowledge on nonlinear systems, people has approximated a model nonlinear to linear model in the neighborhood of working state points that we are interested and we transform the nonlinear control problem into a control problem linear. The linearization around the working point is carried out according to the [5] as follows:

Select the working points when the system is stable as follows:

The temperature inside the tank is $\overline{T}$

The initial level of the tank is $\overline{h}$

Cold water flow is $\overline{w_1}$

Hot water flow is $\overline{w_2}$

Flowing out water is $\omega = C.\sqrt{2g}A_0\sqrt{h} = k\sqrt{h}$ (From Eq 2.1)

$$\begin{cases} 0 = \dfrac{\overline{w_1} + \overline{w_2} - k\sqrt{\overline{h}}}{A} \\ 0 = \dfrac{\overline{w_1}\left(Tc - \overline{T}\right) + \overline{w_2}\left(Th - \overline{T}\right)}{A\overline{h}} \end{cases}$$

$$\Leftrightarrow \begin{cases} \overline{w_1} + \overline{w_2} = k\sqrt{h} \\ \overline{w_1} \times Tc + \overline{w_2} \times Th = d \times \overline{T} \end{cases} \qquad \text{(Eq 3.1)}$$

Consider a point located adjacent to the working point:

$$\begin{cases} h = \overline{h} + \Delta h \\ T = \overline{T} + \Delta T \\ w_1 = \overline{w_1} + \Delta w_1 \\ w_2 = \overline{w_2} + \Delta w_2 \end{cases} \qquad \text{(Eq 3.2)}$$

Substituting (3.2) into the system of equations (2.9) we get:

Where Δh very small:

$$\begin{cases} \sqrt{h + \Delta\overline{h}} = \sqrt{h}\sqrt{1 + \frac{\Delta h}{\overline{h}}} \approx \sqrt{\overline{h}}\left(1 + \frac{\Delta h}{2\overline{h}}\right) \\ \left(\overline{h} + \Delta h\right)^{-1} = \overline{h}^{-1}\left(1 + \frac{\Delta h}{\overline{h}}\right)^{-1} \approx \overline{h}\left(1 - \frac{\Delta h}{\overline{h}}\right) = \overline{h}^{-1} \end{cases} \qquad \text{(Eq 3.3)}$$

And ignore the term $(\Delta w_1 + \Delta w_2)\Delta T$.

Thus system (3.3) will become:

$$\begin{cases} \dfrac{d\Delta h}{dt} = \dfrac{\Delta w_1 + \Delta w_2}{A} - \dfrac{k\Delta h}{2A\sqrt{h}} \\ \dfrac{d\Delta T}{dt} = \dfrac{\Delta w_1\left(Tc - \overline{T}\right) + \Delta w_2\left(Th - \overline{T}\right) - (\overline{w_1} + \overline{w_2})\Delta T}{A\overline{h}} \end{cases}$$

$$\Leftrightarrow \begin{pmatrix} \frac{d\Delta h}{dt} \\ \frac{d\Delta T}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{k}{2A\sqrt{h}} & 0 \\ 0 & -\frac{k}{A\sqrt{h}} \end{pmatrix} \begin{pmatrix} \Delta h \\ \Delta T \end{pmatrix} + \begin{pmatrix} \frac{1}{A} & \frac{1}{A} \\ \frac{Tc - \overline{T}}{A\overline{h}} & \frac{Th - \overline{T}}{A\overline{h}} \end{pmatrix} \begin{pmatrix} \Delta w_1 \\ \Delta w_2 \end{pmatrix} \qquad \text{(Eq 3.4)}$$

3.1.2 Controllability

Controllability Concept: A continuous, linear system is said to be controllable if there exists at least one control signal that can bring it from an initial state point x_0 (arbitrary) to the origin 0 in a finite amount of time.

From [1], According to Hautus necessary and sufficient conditions for a controllable linear system without redundant states (3.4) is:

$$\text{Rank (sI – A, B) =n, } \forall s \in \mathbb{C}$$

Also, from [1], according to Kalman, the necessary and sufficient conditions for a controllable linear system without redundant states (3.4) are:

$$Rank(B, AB, A^2B, \dots, A^{n-1}B) = n$$

With balance point $\bar{h} = 0{,}25$ and $\bar{T} = 45$, we have:

$$A = \begin{pmatrix} -0.0005 & 0 \\ 0 & -0.001 \end{pmatrix}; B = \begin{pmatrix} 31 & 31 \\ 3086 & -3086 \end{pmatrix}$$

Applying Kalman criterion for (3.4) with the above equilibrium point, we get:

Rank (B, AB) = 2

Therefore, (3.4) is controllable system.

3.2 Direct channel decoupling controller

3.2.1 Theory

This part will introduce decoupling method for MIMO system, which means design controller D(s) for object has m input and m output signal: $\underline{u}$= (u_1, u_2, …, u_m)^T, $\underline{y}$= (y_1, y_2, …, y_m)^T, where with D(s), system is split to m separated channel (m smaller SISO system), like in Figure 3.1 for 2 input , 2 output object. Obviously that when split into SISO systems, we can apply other controller design methods for each channel so that the system can reach output quality.

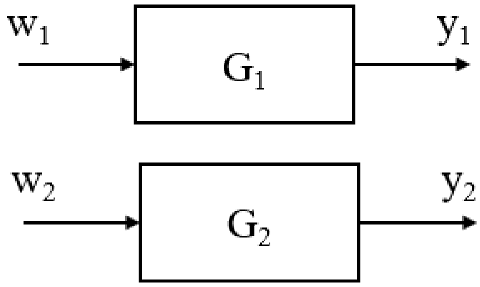

Figure 3.1 Direct channel decoupling control

In 3.1.1, we have calculated system model after linearization around the working point:

$$\begin{pmatrix} \frac{d\Delta h}{dt} \\ \frac{d\Delta T}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{k}{2A\sqrt{h}} & 0 \\ 0 & -\frac{k}{A\sqrt{h}} \end{pmatrix} \begin{pmatrix} \Delta h \\ \Delta T \end{pmatrix} + \begin{pmatrix} \frac{1}{A} & \frac{1}{A} \\ \frac{Tc - \bar{T}}{A\bar{h}} & \frac{Th - \bar{T}}{A\bar{h}} \end{pmatrix} \begin{pmatrix} \Delta w_1 \\ \Delta w_2 \end{pmatrix}$$

The form of this model is $\begin{cases} \underline{\dot{x}} = A\underline{x} + B\underline{u} \\ \underline{y} = C\underline{x} + D\underline{u} \end{cases}$

With

$$A=\begin{pmatrix} -\frac{k}{2A\sqrt{h}} & 0 \\ 0 & -\frac{k}{A\sqrt{h}} \end{pmatrix}; \qquad B=\begin{pmatrix} \frac{1}{A} & \frac{1}{A} \\ \frac{Tc-\overline{T}}{A\overline{h}} & \frac{Th-\overline{T}}{A\overline{h}} \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \qquad D= 0$$

The above model is the representation of the system in the state space domain. To design the demultiplexer controller, we need to model the system in the frequency domain, with the transfer function $G_F(s)$ satisfied:

$Y(s)= G_F(s)U(s)$

Among them are the following, $G_F(s)$ calculated based on the model in the state space domain:

$$G = C.\,(sI-A)^{-1}.B + D \qquad \textit{Eq 3.1}$$

3.2.2 Controller design

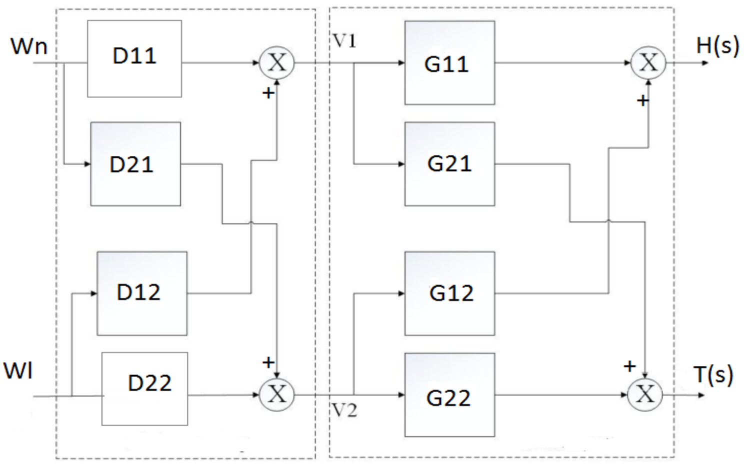

Figure 3.2 Direct channel decoupling controller diagram

Select channel separation compensation $D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}$

We have :

$$\begin{pmatrix} H \\ T \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} . \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

which $\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} . \begin{pmatrix} Wn \\ Wl \end{pmatrix}$

Therefore:

$$\begin{pmatrix} H \\ T \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} . \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} . \begin{pmatrix} \mathrm{Wn} \\ Wl \end{pmatrix}$$

$$= \begin{pmatrix} G_{11}.D_{11} + G_{12}.D_{21} & G_{11}.D_{12} + G_{12}.D_{22} \\ G_{21}.D_{11} + G_{22}.D_{21} & G_{21}.D_{12} + G_{22}.D_{22} \end{pmatrix} \begin{pmatrix} \mathrm{Wn} \\ Wl \end{pmatrix}$$

Set $G_D = \begin{pmatrix} G_{11}.D_{11} + G_{12}.D_{21} & G_{11}.D_{12} + G_{12}.D_{22} \\ G_{21}.D_{11} + G_{22}.D_{21} & G_{21}.D_{12} + G_{22}.D_{22} \end{pmatrix}$

Now we find matrix D where $G_D$is diagonal matrix.

We have :

$$G = C.\ (sI\text{-}A)^{-1}.B + D$$

Where:

$A = \begin{pmatrix} -0.0005 & 0 \\ 0 & -0.001 \end{pmatrix}; B = \begin{pmatrix} 31 & 31 \\ 3086 & -3086 \end{pmatrix}; C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$$sI - A = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 5 \times 10^{-4} & 0 \\ 0 & 10^{-3} \end{pmatrix} = \begin{pmatrix} s + 5 \times 10^{-4} & 0 \\ 0 & s + 10^{-3} \end{pmatrix}$$

$$(sI - A)^{=1} = \frac{1}{(s + 5 \times 10^{-4})(s + 10^{-3})} \begin{pmatrix} s + 10^{-3} & 0 \\ 0 & s + 5 \times 10^{-4} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{s + 5 \times 10^{-4}} & 0 \\ 0 & \frac{1}{s + 10^{-3}} \end{pmatrix}$$

$$C \times (sI - A)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{s + 5 \times 10^{-4}} & 0 \\ 0 & \frac{1}{s + 10^{-3}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{s + 5 \times 10^{-4}} & 0 \\ 0 & \frac{1}{s + 10^{-3}} \end{pmatrix}$$

$$G = C \times (sI - A)^{-1} \times B = \begin{pmatrix} \frac{1}{s + 5 \times 10^{-4}} & 0 \\ 0 & \frac{1}{s + 10^{-3}} \end{pmatrix} \begin{pmatrix} 31 & 31 \\ 3086 & -3086 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{31}{s + 5 \times 10^{-4}} & \frac{31}{s + 5 \times 10^{-4}} \\ \frac{3086}{s + 10^{-3}} & \frac{-3086}{s + 10^{-3}} \end{pmatrix}$$

Therefore, $D = N^{-1} = \begin{pmatrix} 0.0161 & 1.62 \times 10^{-4} \\ 0.0161 & -1.62 \times 10^{-4} \end{pmatrix}$. This is matrix D which we need to find.

$$\Rightarrow G_D = \begin{pmatrix} \frac{1}{s + 5 \times 10^{-4}} & 0 \\ 0 & \frac{1}{s + 10^{-3}} \end{pmatrix}$$

From matrix G_D above, we could see: Hot water flow controls level and cold water flow controls temperature.

- Level control channel : $G_{D11} = \frac{1}{s+5\times10^{-4}}$

This is first order inertia. We use PI controller.

Choose $k_p = 1; k_I = 0,5$, we receive water level response :

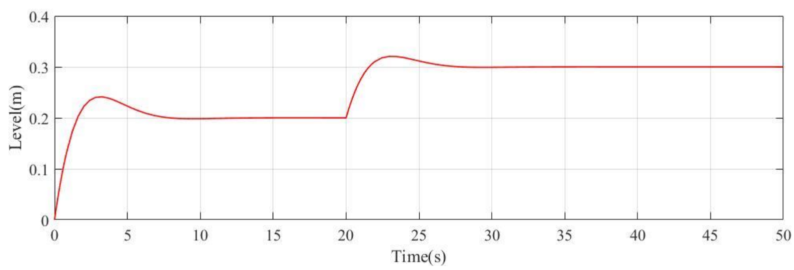

Figure 3.3 Level response

Perform controller parameter tuning with $k_p = 1; k_I = 0,1$, we got response :

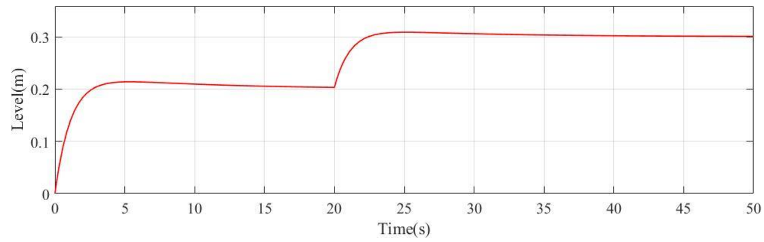

Figure 3.4 Tuned Level response

➤ Temperature control channel : $G_{D22} = \frac{1}{s+10^{-3}}$

This is first order inertia. We use PI controller.

Choose $k_p = 1; k_I = 0{,}5$, we receive water level response :

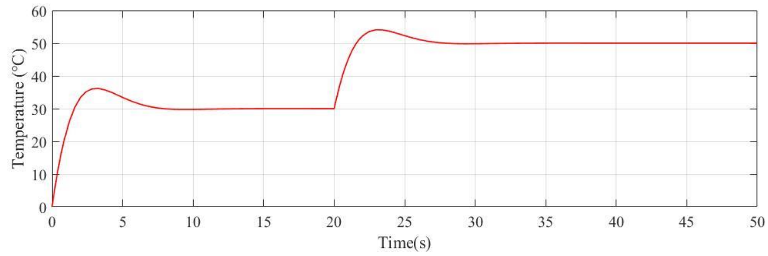

Figure 3.5 Temperature response

Perform controller parameter tuning with $k_p = 1; k_I = 0{,}03$, we got response :

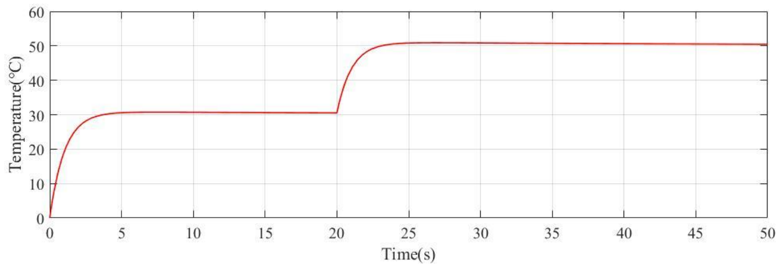

Figure 3.6 Tuned temperature response

We have control diagram:

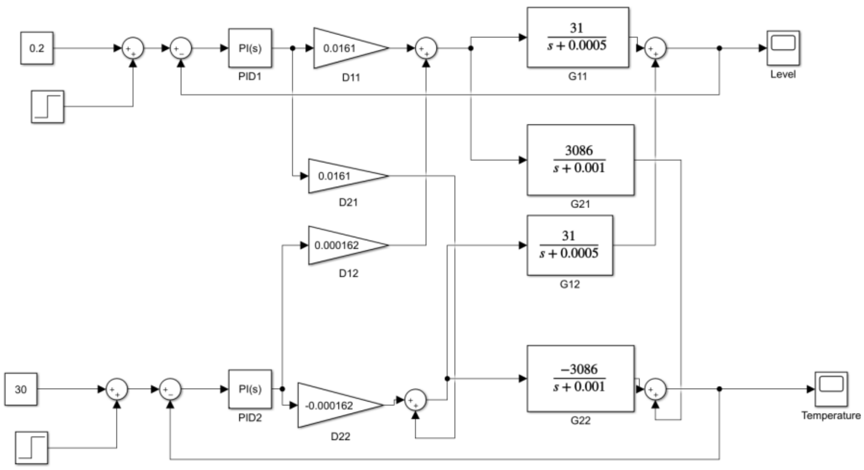

Figure 3.7 Control diagram

3.3 Iterative learning controller design

3.3.1 Introduction

The idea of the iterative cybernetics (ILC) algorithm was first introduced in the mid-80s of the last century, it uses the previous looping condition signal and the error signal and aims to generate a good control signal. than for the next iteration and it keeps the next iteration's bias as low as possible for the whole working time. It can be shown that if certain conditions are met, after several iterations, the tracking bias in all times will tend to zero. This is a fundamental difference between this algorithm and conventional controllers such as PID controllers. However, in ILC, if we have an exact model of the object, we can design the algorithm so that the error will be zero in the first place. The points of the ILC system such as: Iterative learning control is a method by using the previous data, the error signal to generate a better control signal later, increasing the quality and strength of the algorithm. control calculation, control error is reduced.

Iterative Learning Control (ILC) is an intelligent control method for cyclic working systems.

- Principles of learning and control:
  - Study to get $\underline{u}_k(t) \rightarrow \underline{u}^*(t)$, k = 0,1, ...
  - Apply for open loop controller $\underline{u}_N(t)$

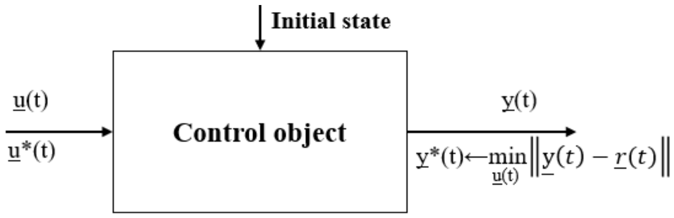

Figure 3.8 Principle of learning and control

- Study rule: Determine the iterative learning formula:

$\underline{u}_{k+1}(t) = \underline{f}_l(\underline{u}_k(\tau), \underline{y}_k(\tau), t)$ with $\tau \epsilon [0, T]$, k=0,1,...,N to have

$$\left\| \underline{y}_N(t) - \underline{r}(t) \right\| \leq \varepsilon$$

- Steps to build an iterative learning controller：

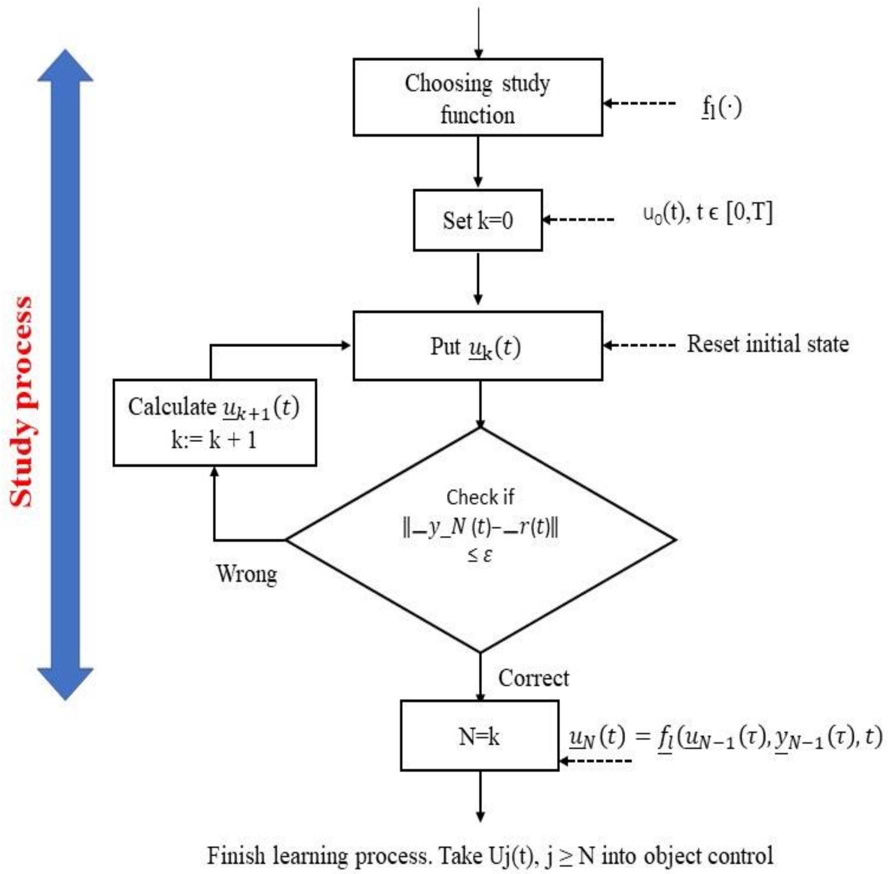

Figure 3.9 Steps to build an itertive learning contronller

3.3.2 Features of iterative learning controller

Some features of ILC:

1. As an intelligent controller, because it does not use the mathematical model of the object.
2. Convergence of the learning process depends on the selected learning function $\underline{u}_{k+1}(t) = \underline{f_l}(\underline{u}_k(\tau), \underline{y}_k(\tau),)$ and of course depends on the kinematics of the control object itself.
3. Every time the set signal $\underline{r}(t)$ or initial state is changed, the learning process must be repeated
4. Apply to objects whose work process repeats over time in a period of fixed time kT $\leq t \geq$ (k+1) T
5. "Learning" because input signal $\underline{u}_k(t)$ is calibrated through the past data $\underline{u}_{k-1}(\tau)$, $\underline{y}_{k-1}(\tau)$, $0 \leq \tau \leq$ T
6. "Iterative" because learning is repetitive and only final result of learning process $\underline{u}_j(t)$, j$\geq N$ is usable (different from recursive).
7. ILCler applied for stable object. With unstable object, people usually stabilize it by traditional controller then use ILCler under feedforward form.
8. When considering the convergence of the learning process, it is common to see $\underline{u}_k(t)$=$\underline{u}_k(k,t)$ is a function in 2-dimension space then apply 2-dimension system theory.

3.3.3 Application for water tank control

Based on the object model built in chapter 3, combined with the ILC designed for the object, we can give a complete simulation diagram of for the object to control the temperature and water level of the water heater system.

- Following steps in 3.3, we must choose a study function:

$\underline{u}_{k+1}(i) = \underline{u}_k(i) + K\underline{c}_k(i)$ with $\underline{c}_k(i) = \underline{r}(i) - \underline{y}_k(i)$, $\tau = i \times T_S$, $0 < T_S << 1$.

- Then we have a MATLAB function that represents the study process.
- Control diagram is represented below:

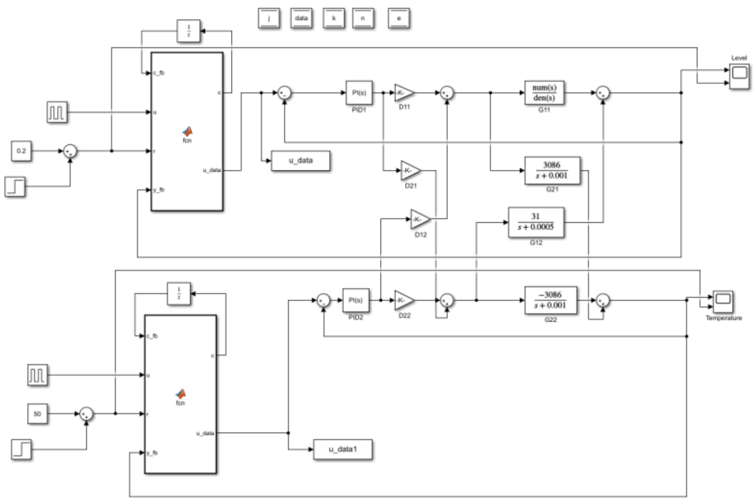

Figure 3.10 Control diagram

PID1 controller parameter: $PI(s) = 3 + 0{,}1\frac{1}{s}$

PID2 controller parameter: $PI(s) = 1 + 0.03\frac{1}{s}$

3.3.4 Simulation result

With 100 learnings, temperature setting is 45 and water level setting is 0.20 and change to 0.25 at T_1=30s, time T=50s

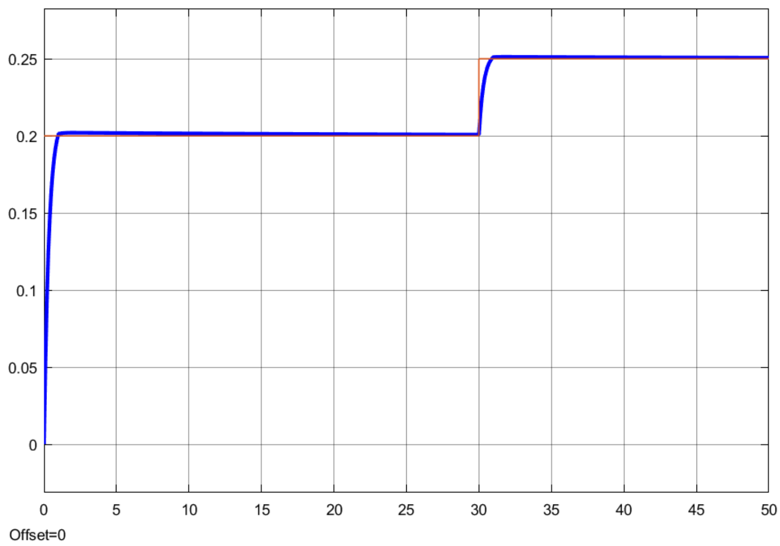

Figure 3.11 Level simulation result

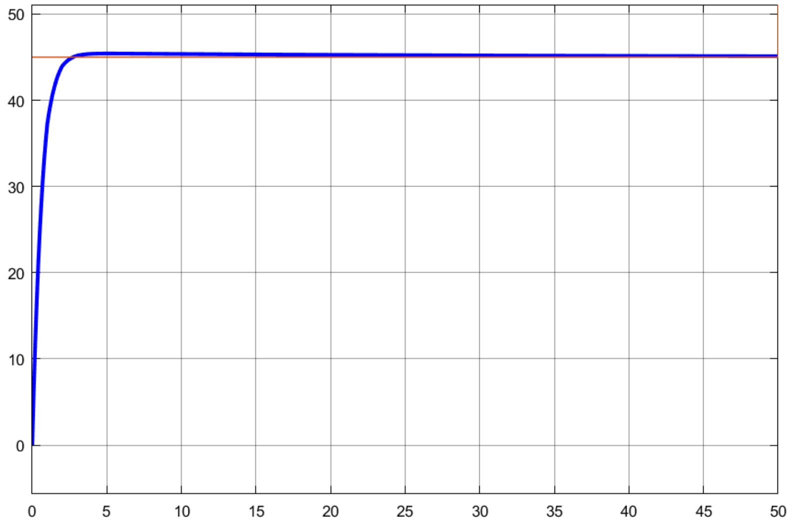

Figure 3.12 Temperature simulation result

With 100 learnings, temperature setting is 60 and water level setting is 0.3, time T=50s

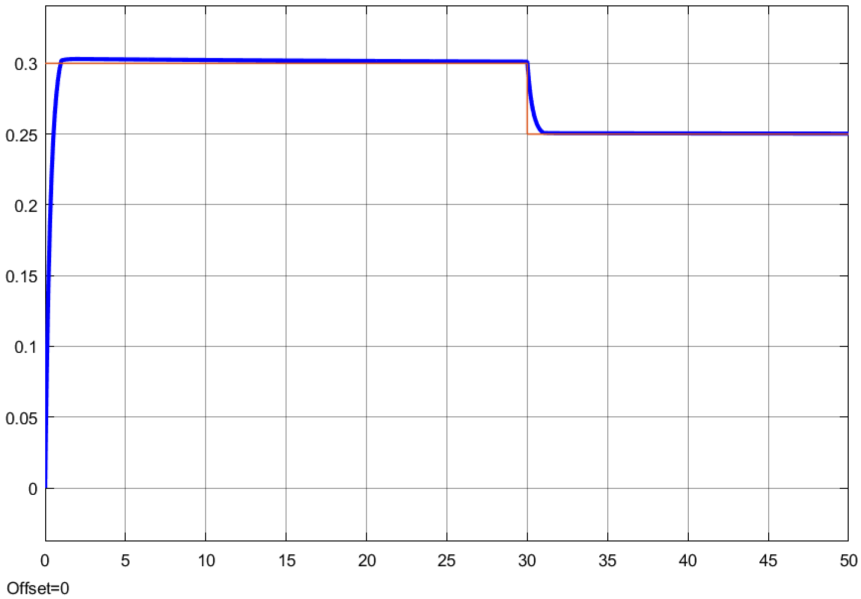

Figure 3.13 Level simulation result

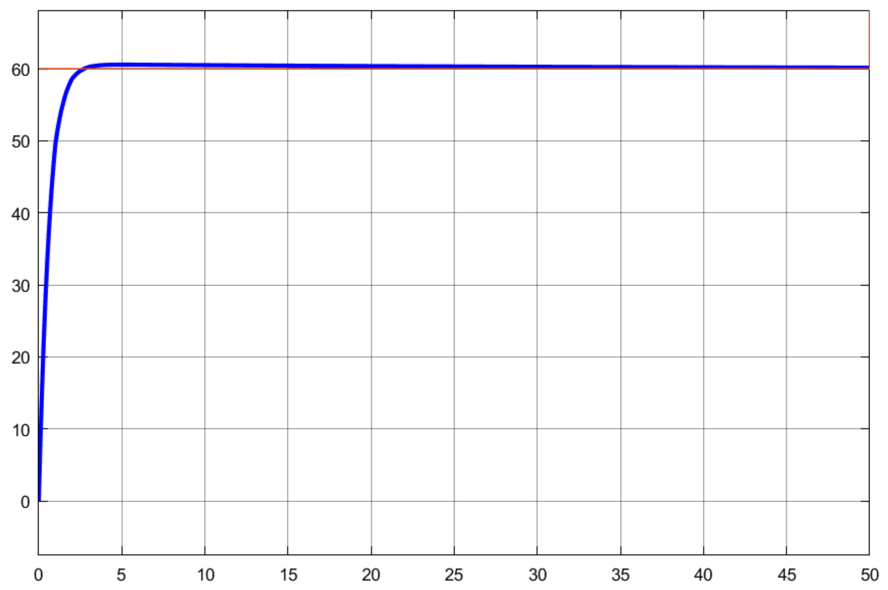

Figure 3.14 Temperature simulation result

Comment:

- The results after 120 times of learning stick to the set value, satisfying the set requirements.
- The deviation when the set value changes is relatively small, the response time is short.

CHAPTER 4. DESIGN CONTROL CIRCUIT

4.1 Hardware design

4.1.1 Overall diagram and principle

a) Overall diagram

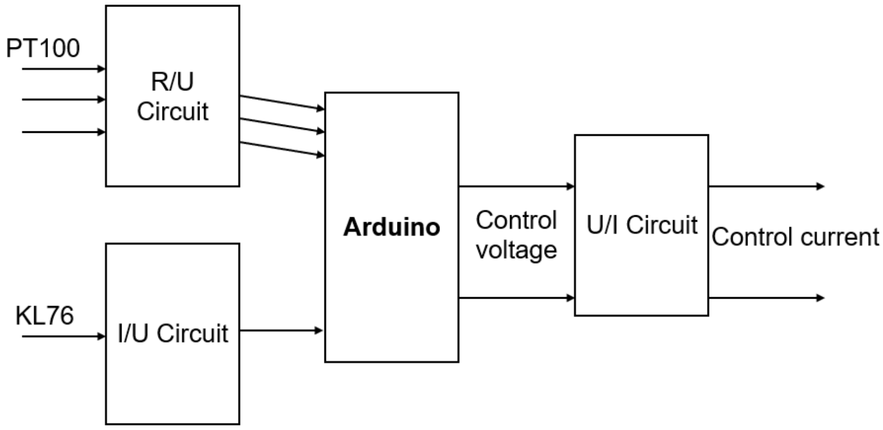

Figure 4.1 Overall diagram

b) Principle

Basic principle of hardware circuit:

- Temperature and level signals of the system will be collected by sensors.
- These signals is calculated and change into 0 – 5V voltage which Arduino can read by analog input.
- Controller will calculate and generate control signals by PWM signal base on set value and data from sensors.
- PWM signal is converted to control current 4 – 20 mA in order to control pneumatic valves.

4.1.2 Hardware components and their tasks

a) Source circuit

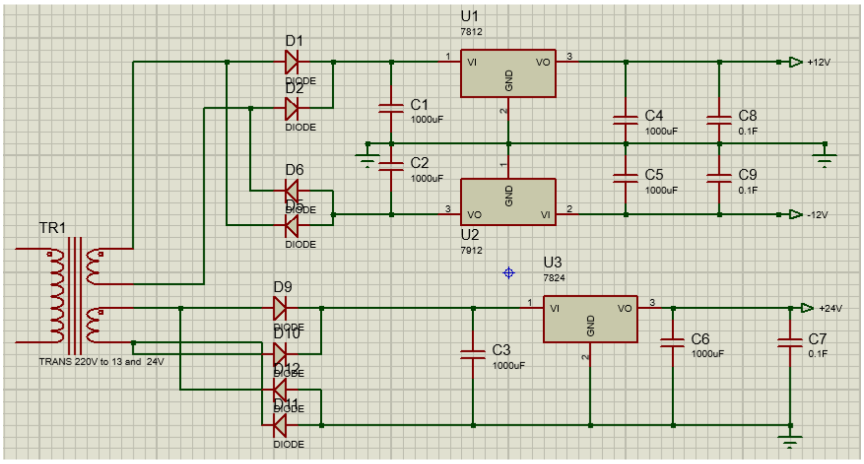

Figure 4.2 Source circuit

A transformer with 2 seperated secondary coil change 220V source to 13VAC and 24VAC. Using 4 diode 4007 for creating positive sine wave in the upper branch and negative sine wave in the lower branch.

To create $\pm 12VDC$ for IC supply , we use Symmetrical source circuit using 7812 and 7912 voltage regulator. Other capacitors to flatten voltage.

24VDC for KL76 sensor, similarly, we use 7824 votage regulator circuit.

b) R/U Converter for temperature measurement

Temperature sensor PT-100 measure temperature of the tank, from that temperature signal change into resistance signal, goes through R/U converter, convert resistance signal to voltage signal, through Arduino (analog input) to Simulink in computer.

PT100 has properties: $\begin{cases} R_T = 100\Omega \; at \; 0°C \\ R_T = 139{,}7\Omega \; at \; 100°C \end{cases}$

This circuit will change Rt value to voltage with requirements:

$$\begin{cases} R_T = 100\Omega \; corresponding \; with \; U = 0V \\ R_T = 139{,}7\Omega \; corresponding \; with \; U = 5V \end{cases}$$

Principle diagram:

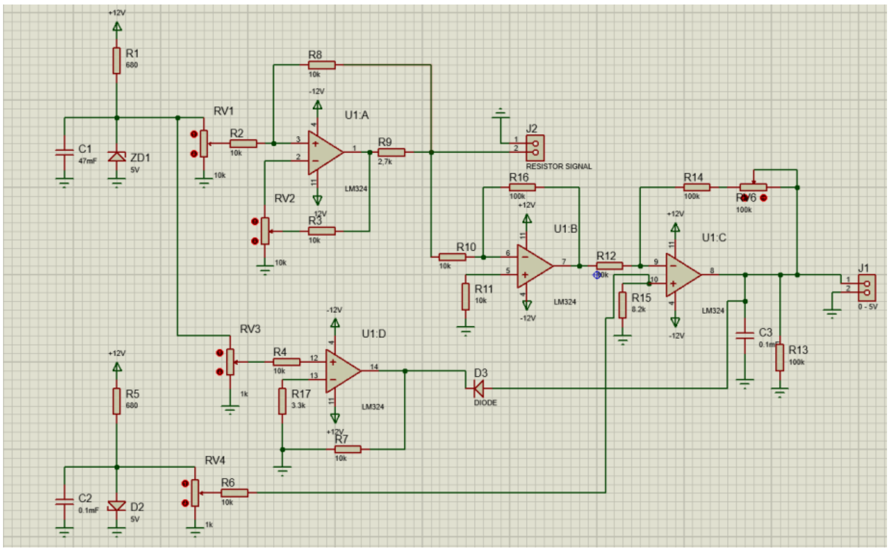

Figure 4.3 R/U Converter

In this circuit we use an OPAMP to stablize the current, and 2 more to amplify signals to 0 – 5V.

We choose high value resistors in the circuit so that when we add R_t the current will stable.

There is a protection circuit for amplification that always below 5V because if add R_t which may create larger voltage than 5V, it will burn out analog input of the Arduino.

This requires 4 OPAMPs so we use TL084 IC for the circuit.

c) I/U Converter for level measurement

Pressure sensor KL76 is supplied from DC voltage 24V, connected in series with a sample resistor R_0 = 68Ω-1%-10W has a constant resistor, does not depend on current through resistor.

KL76 connected with R_0 diagram:

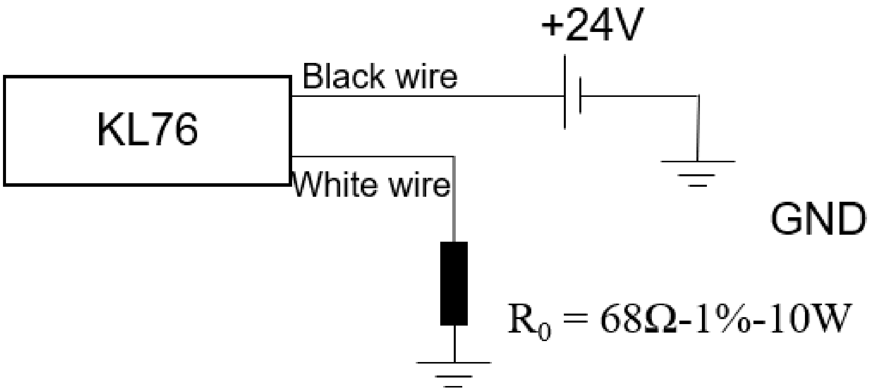

Figure 4.4 KL76 Connection diagram

This sensor is place in the bottom of the tank. Water level will be propotion with the current I through the sensor. Higher the level, higher the pressure on the sensor, the current will be larger. Voltage diffrence of R0 by I current flow through, will be amplify to go to Arduino.

Requirement:

$$\begin{cases} At\ h = 0\ cm \rightarrow\ U = 0V \\ At\ h = 30\ cm \rightarrow\ U = 5V \end{cases}$$

Principle diagram:

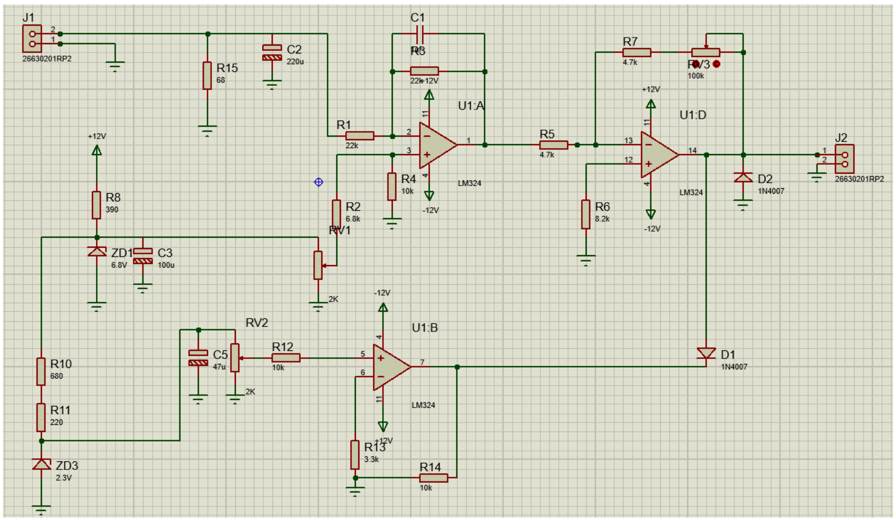

Figure 4.5 I/U Converter

By measurement method, data has been collected below (We have measured ten times and this is the average value) :

Voltage (V)	Level (m)
4.2	0.3
3.45	0.25
2.75	0.2
2	0.15
1.3	0.1

Table 1 I/U circuit voltage measurement

Therefore, we have level-voltage relation:

$$\begin{cases} At\ h = 0\ cm \rightarrow\ U = 0V \\ At\ h = 30\ cm \rightarrow\ U = 4.2V \end{cases}$$

Although it is not as required, but signals is still in the range that Arduino can read.

d) U/I Converter to control I/P tranducer

Arduino only can generate PWM signal, and I/P tranducer receive 4 – 20 mA current to control pneumatic valves, so a circuit to convert PWM signal to current.

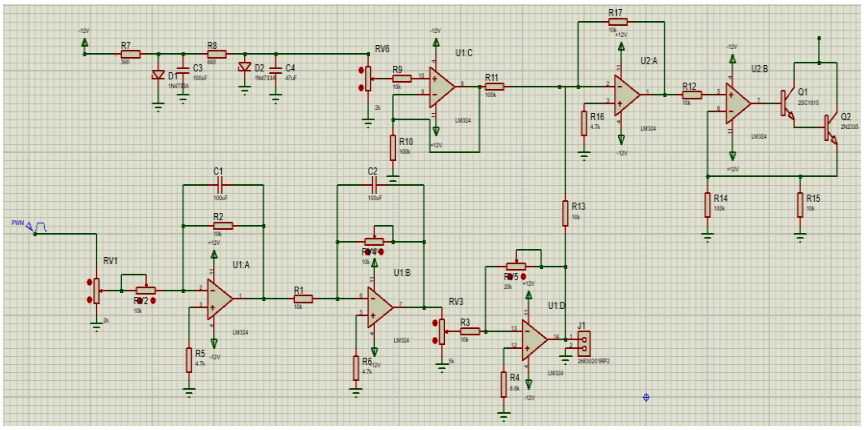

Figure 4.6 U/I Converter

PWM signals from microprocessor to rheostat will be levelled to DC voltage in the ouput pin 14 of second IC, where value is vary from 0 to -5V (depend on pulse width is 0 or 100%). This voltage will be reversed sign to positive in pin 1 of second IC, then goes to voltage repeater circuit on the Emitter of C2335 transitor create stable current follow the voltage on R0=220Ω - 5W in Emitter. Emitter current on R0 is the Collector current of C2335 through pneumatic valve.

Collector current will decide the pressure of valve.

There is a negative voltage -0.9V circuit in pin 8 of second IC and positive voltage 0.9V in pin 1 of second IC to create 4mA current through valve when PWM signal equals 0.

This circuit requires OPAMP in 2 IC, therefore, we use TL084, using bipolar source $\pm 12V$.

4.1.3 Hardware figures

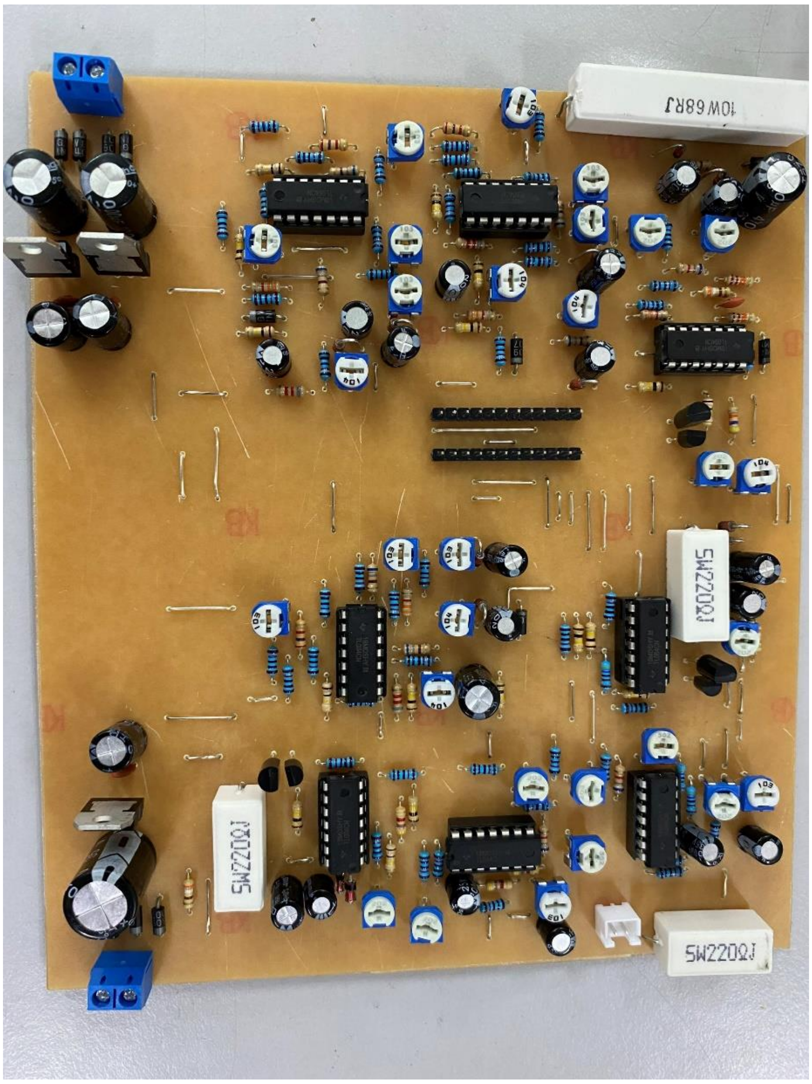

Figure 4.7 Reality combination circuit

4.2 Arduino MEGA ADK and ArduinoIO library

4.2.1 Arduino MEGA ADK

We just use Arduino for data collection and transmit data to computer. All the calculations are performed in MATLAB Simulink. Therefore, we have a requirement for choosing Arduino:

- Compact design
- Easy to buy in the market
- Wild range of open source library that supports for control
- At least 4 analog input and 2 PWM output
- Support library for MATLAB Simulink

From the requirements above, we choose **Arduino MEGA ADK 2650**.

Arduino MEGA ADK 2650 is a popular circuit, with high accuracy and durability.Using Atmega2560 8 bit microcontroller, this arduino can easily control single led, engine control, signal process, data collection and many other application. Arduino MEGA ADK 2650 board is described in figure below:

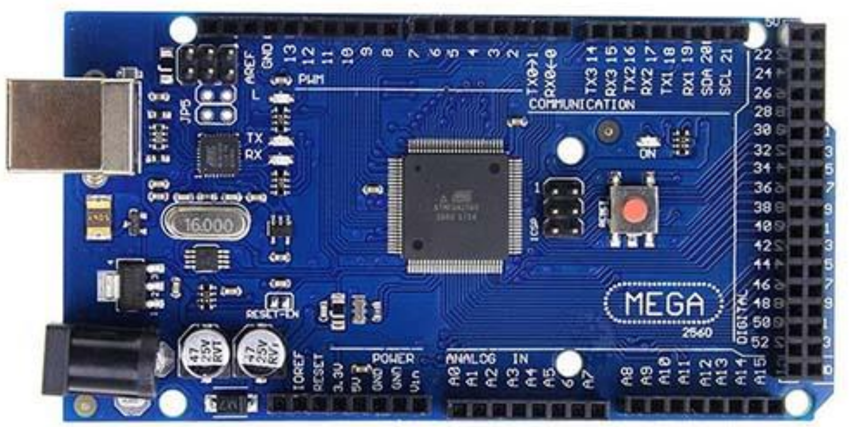

Figure 4.8 Arduino MEGA ADK 2560

a) Voltage source:

The Arduino can be powered by 5V through the USB port or externally powered with a recommended voltage of 7-12V DC and a limit of 6-20V. Usually, powering with a 9V square battery is the most reasonable if there is no power from the USB port available. If the power supply exceeds the upper limit threshold, it will damage the Arduino .

GND (Ground): negative pole of the power supply for Arduino. When using devices that use separate power sources, these pins must be connected together.

5V: output voltage 5V. The maximum allowable current at this pin is 50mA.

3.3V: output voltage 3.3V. The maximum allowable current at this pin is 50mA.

Vin (Voltage Input): to supply external power to the Arduino, connect the positive pole of the source to this pin and the negative pole of the source to the GND pin.

IOREF: the microcontroller operating voltage on the Arduino can be measured at this pin. And of course it's always 5V.

RESET: pressing the Reset button on the board to reset the microcontroller is equivalent to the RESET pin connected to GND through a 10KΩ resistor.

Notes:

The Arduino has no input reverse protection. Therefore, be very careful, check the negative - positive poles of the source before supplying it to the Arduino. Short-circuiting the Arduino's input power will cause the board to burn.

The 3.3V and 5V pins on the Arduino are the pins used to power other devices, not the input power pins. Power supply in the wrong place can damage the board. This is not recommended by the manufacturer.

Supplying external power without the USB port to the Arduino with a voltage above 20V may damage the board.

Supplying a voltage above 13V to the RESET pin on the board can damage the ATmega2560 microcontroller.

The input/output amperage at all Digital and Analog pins of the Arduino if exceeded 40mA will damage the microcontroller. Therefore, if not used for data transmission, we must connect a current limiting resistor.

Supplying voltages above 5.5V to the Digital or Analog pins of the Arduino will damage the microcontroller.

b) Memory

256KB of Flash memory: the programming instructions will be stored in the microcontroller's Flash memory. Usually a few KB of this will be used for the bootloader.

4KB for RAM (Random Access Memory): the values of variables declared when programming will be stored here. The more variables you declare, the more RAM you need. When the power is lost, the data on the RAM will be lost.

4KB for EEPROM (Electrically Erasable Programmable Read Only Memory): this is like a mini hard drive – can read and write data here without having to worry about losing it during a power outage like data on RAM.

c) IO ports

Arduino MEGA ADK 2560 has 54 digital pins to read or generate signals. They only has 2 voltage level are 0 and 5V with maximum current of each pin is 40 mA. Each pin has a pull-up resistor set up in Atmega 2560.

Some special digital pins:

- 2 Serial pins: 0 (RX) and 1 (TX): Used to send (transmit – TX) and receive (receive – RX) TTL Serial data. Arduino MEGA ADK can communicate with other devices through these 2 pins.
- LED 13: on Arduino there is 1 orange led (signed L). When pressing the Reset button, we will see this light blink to signal. It is connected to digital pin 13. When this pin is used by the user, the LED will light up.

- PWM pins: PWM 2 to 13: allow to output PWM pulses with 8-bit resolution (values from 0 → 255 corresponds to 0V → 5V). We can adjust the output voltage at this pin from 0V to 5V instead of just fixed at 0V and 5V like other pins.
- SPI communication pins: SS, MOSI, MISO, SCK. In addition to the usual functions, these 4 pins are also used to transmit data using the SPI protocol with other devices.
- Arduino has 15 analog pins (A0 → A15) that provide 10bit signal resolution (0 → 1023) to read voltage values in the range 0V → 5V. With the AREF pin on the board, it is possible to input the reference voltage when using the analog pins. That is, if we apply a voltage of 2.5V to this pin, you can use analog pins to measure voltages in the range from 0V → 2.5V with a resolution of 10bit.
- Arduino has 2 pins SDA and SCL that support I2C/TWI communication with other devices.

4.2.2 ArduinoIO library

We need ArduinoIO library to connect Arduino directly to conversion circuit.

a) Setting for ArduinoIO library:

- Download and extract the Arduino IDE and the ArduinoIO library from the Arduino and MathWorks website:
  - Arduino IDE: http://arduino.cc/en/main/software
  - MATLAB Support for Arduino (aka ArduinoIO Package)
- Using Arduino IDE Upload file ADIOES.PDE to Arduino ADK 2560 board
- Add ArduinoIO library to Matlab/Simulink:
  - Run MATLAB as Administrator
  - Run install_arduino.m

b) ArduinoIO library blocks

ArduinoIO library blocks in Matlab Simulink is describe in figure below:

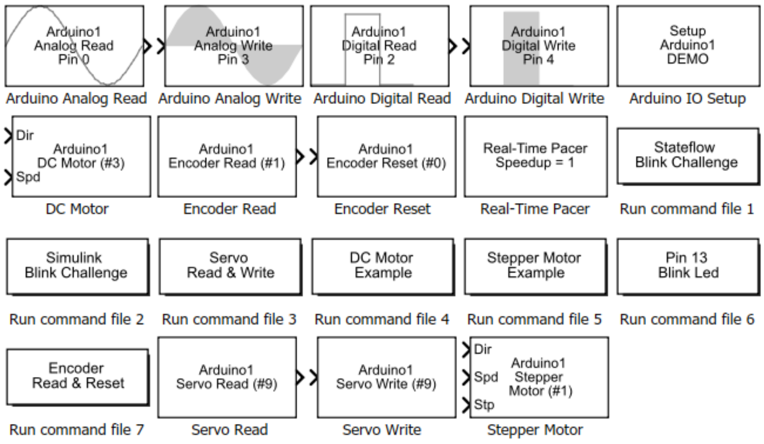

Figure 4.9 Arduino IO library

- **Arduino IO setup** function block: set up communication settings with Arduino.

Connecting the Arduino to the computer will create a serial communication port (for example COM3, COM4, ...). The user must declaire for MATLAB to which port the Arduino is connected.

- **Real-Time Pacer** function block: Set Simulink to run with time

real time.

- **Arduino Analog Read** function block: reads the ADC value on the inputs

Arduino analog. Since the Arduino Mega ADK board can vary the voltage

analog 0÷5V from analog input A0÷A15 to a 10-bit digital value so this block

will get results from 0÷1024 corresponding to the voltage value at the terminals

the same is declared.

- **Arduino Digital Read** function block: reads the value of digital inputs of

Arduino. This block result can be 0 or 1 according to the declared numeric input.

Arduino Digital Write function block: writes 0 or 1 to the outputs

declared number.

- **Arduino Analog Write** function block: output analog value on terminals analog output of the Arduino. Arduino considers the pins to have a control function PWM as analog output pins. Used by Arduino8bit register for PWM control so the value of the Arduino Analog block Write received from 0÷255 corresponds to a PWM pulse of pulse width from 0÷100%. Arduino Mega ADK PWM frequency is 980Hz.

And some other function blocks.

4.2.3 Software design

a) Requirements

A completed system requires a software to calculate controller for the system. This watertank system is designed with following function:

- Receive and process signal from sensor
- Calculate controller and generate corresponded voltage
- Communication between Arduino and Matlab.

Therefore, we create function block in Simulink to connect input and ouput using Arduino.

b) Input:

- Temperature: using Analog input block

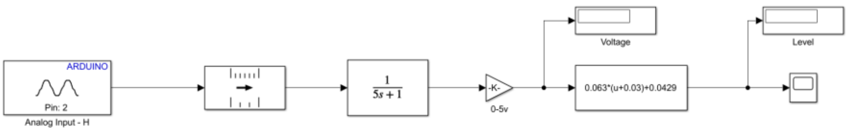

Figure 4.10 Analog input block of temperature

Signal through R/U conversion circuit, change to 0-5V signal go to Analog In pin of Arduino.

Signal through analog read change to 0 – 1023 bit, multiplies with $k = \frac{5}{1023}$ to 0-5 signal range.There is a transfer function block for noise filter. Then signal will through a relation function of R and U to go to return value in the controller.

- Similarly to level signal:

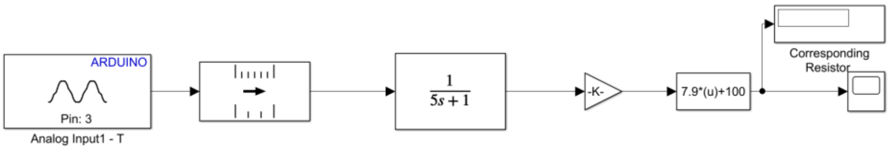

Figure 4.11 Analog input block of level

c) Output

- PWM generation block:

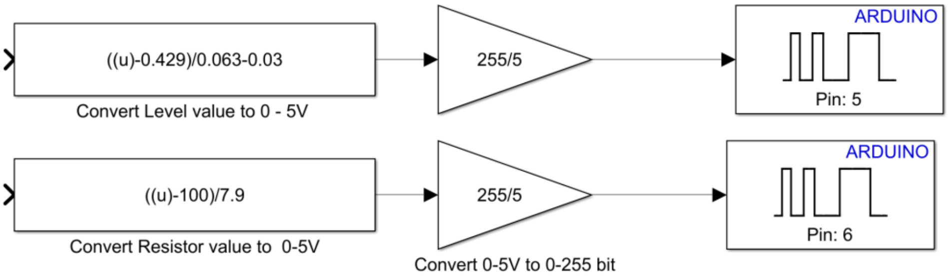

Figure 4.12 PWM output block

Signal is calculated through controller need to change to 0 – 255 bit to PWM pin of the Arduino. Then through a U/I circuit to control pneumatic valve.

For short, we let all above as a subsystem, analog input subsystem and PWM subsystem.

d) Complete design of software on Simulink

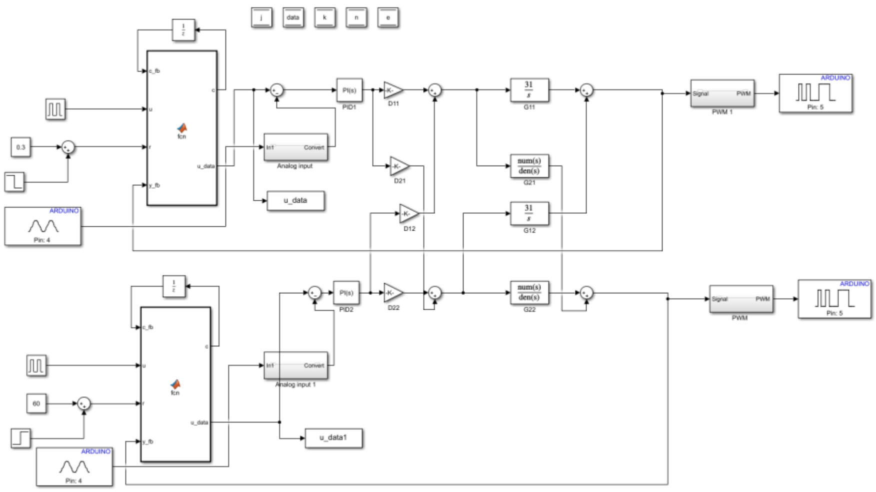

Figure 4.13 Complete design of software

CHAPTER 5. CONCLUSIONS

5.1 Conclusions

- Thesis contents:

  Control circuit is designed but cannot applied to real object due to broken air compressor in the laboratory (cannot check if the output is correct).

  Directly design a popular model in industry.

- For Watertank system:

  Complete object identification as multiple input and output systems, namely the hot and cold system in the laboratory

  Understand and know how to create a simple object iterative controller

  Building an iterative controller model with relatively small error, short response time, real value closely following the set value

- For ILC:

  Examples of systems that operate in a repetitive manner include robot arm manipulators, chemical batch processes and reliability testing rigs. In each of these tasks the system is required to perform the same action repeatedly with high precision. This action is represented by the objective of accurately tracking a chosen reference signal r(t) on a finite time interval.

5.2 Limitations and development directions

- Limitations
  - Due to the broken air compressor, we cannot test our Simulink design.
  - Many parameters of the model have been idealized leading to deviations from the real model
  - It is not the optimal model, it still needs to be changed to make it more reliable
- Development directions:
- This system is very practical topic. Fix the air compressor so that can test our simulation and design

- Intelligent control is the trend of the future industry, and iterative learning control will be an area that promises to grow strongly in the future and apply to other object than watertank system.

REFERENCES

[1] N. D. Phước, Slide Điều khiển học lặp (Iterative learning control), 2021.

[2] H. M. Sơn, Cơ sở điều khiển quá trình, NXB Bách khoa Hà Nội, 2009.

[3] N. P. Quang, Matlab & Simulink - Dành cho kỹ sư điều khiển tự động, NXB Khoa học và kỹ thuật, 2004.

[4] N. D. Phước, Phân tích và điều khiển hệ phi tuyến, NXB Bách Khoa Hà Nội, 2016.

[5] N. D. Phước, Cơ sở lý thuyết điều khiển tuyến tính, NXB Bách Khoa Hà Nội, 2016.