

**PARAMETER IDENTIFICATION OF A COUPLED TANK LIQUID
LEVEL SYSTEM VIA PARTICLE SWARM OPTIMIZATION**

NUR AZMINA BT OTHMAN

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: PARAMETER IDENTIFICATION OF A COUPLED TANK
LIQUID LEVEL SYSTEM VIA PARTICLE SWARM
OPTIMIZATION

SESI PENGAJIAN: 2009/2010

Saya: NUR AZMINA BINTI OTHMAN

mengaku membenarkan tesis (Projek Sarjana Muda/~~Sarjana/Doktor Falsafah~~)* ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (✓)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:
NO 101-D,

Nama penyelia:

LORONG LEMBAH BUJANG 9/1,

MOHD SYAKIRIN BIN RAMLI

TMN LEMBAH BUJANG FASA 3,

08100, BEDONG, KEDAH.

Tarikh: 24th NOVEMBER 2009

Tarikh: 24th NOVEMBER 2009

CATATAN: * Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.
*** Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara kerja kursus atau penyelidikan atau Laporan Projek Sarjana Muda (PSM).

**PARAMETER IDENTIFICATION OF A COUPLED TANK LIQUID
LEVEL SYSTEM VIA PARTICLE SWARM OPTIMIZATION**

NUR AZMINA BT OTHMAN

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Hons.) (Control & Instrumentation)

Faculty of Electric & Electronics

Universiti Malaysia Pahang

NOVEMBER 2009

SUPERVISOR'S DECLARATION

“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor of Electrical Engineering (Control & Instrumentation)”

Signature : _____

Name : MOHD. SYAKIRIN BIN RAMLI

Date : 24th NOVEMBER 2009

STUDENT'S DECLARATION

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : NUR AZMINA BT OTHMAN

Date : 24th NOVEMBER 2009

DEDICATION

Especially dedicated to my beloved father and mother

Mr. Othman b Zainol and Mrs Hayani bt Abd Hamid

My brothers Muzzammil and Haniff ,Bestfriends and to all faculty staffs

For their infinites care, support, and believe in me.

Sincerely,

Nur Azmina bt Othman

ACKNOWLEDGEMENTS

In the name of Allah S.W.T, the Most Gracious, the Ever Merciful. Praise is to Allah, Lord of the Universe and Peace and Prayers be upon His final Prophet and Messenger Muhammad s.a.w. Thank you for giving me the source of power, guidance in applying knowledge and strength to finish the project and dissertation for completing my Bachelor of Electrical Engineering final year project.

I would like to express my gratitude to my supervisor Mr. Mohd Syakirin bin Ramli for his wisdom, endurance and encouragement during his supervision period. His high motivated spirit and hardworking in every work he do is admirable, and I have always been encouraged to do better in my project and also in living a life.

I also would like to express my appreciation to all lab technicians especially to Mr. Hamka for their help and permission to use lab and equipment needed throughout this project completion.

Special thank to my friends for their kindness in getting and sharing additional information for the project as well as their co-operation and help.

Finally, special thanks extended to my beloved family especially my father, Mr Othman b Zainol and my mother, Mrs Hayani bt Abd Hamid, also brothers for their never ending moral support, financial support and prays for my success.

ABSTRACT

This project focuses on obtaining a mathematical equation for a coupled tank liquid level system. The aim of this project is to identify and obtain the mathematical modeling utilizing Particle Swarm Optimization (PSO). The level of water in each tank is monitored by a capacitive-type probe. Signal conditioning circuits convert the measured capacitance (a function of the water level) to electrical signals in the range 0 to +5 Volt DC. Output from simulated and modeling performance are compared until n set number of iteration for error less output. The fitness function which equal to the sum of errors less square over number of iteration is later used in Visual Basic. Using Visual Basic, a GUI is developed to analyze and display both performances in terms of graph. In addition, both performances are also analyzed in terms of its transient and steady-state responses. The obtained mathematical equation can further be used for controller development such as by simulation in MATLAB/Simulink

ABSTRAK

Projek ini tertumpu pada mendapatkan persamaan matematik untuk tahap cair tangki ditambah sistem. Objektif dari projek ini adalah untuk mengenalpasti dan mendapatkan model matematik memanfaatkan Particle Swarm Optimization (PSO). Paras air di masing-masing tangki dipantau oleh probe taip kapasitif. Litar pengkondisian isyarat menukar Kapasitans terukur (fungsi dari peringkat air) kepada isyarat-isyarat elektrik pada kisaran 0-5 Volt DC. Output daripada simulasi dan pemodelan prestasi berbanding sampai n jumlah iterasi menetapkan untuk kesalahan kurang output. Kebugaran fungsi yang sama dengan jumlah kuadrat kesalahan kurang lebih jumlah iterasi yang kemudian digunakan dalam Visual Basic. Menggunakan Visual Basic, GUI yang dibangunkan untuk menganalisis dan memaparkan persembahan, baik dalam hal grafik. Selain itu, kedua persembahan juga dianalisis dari segi transien dan tanggapan keadaan tunak. Persamaan matematik yang diperolehi selanjutnya dapat digunakan untuk pengembangan controller seperti dengan simulasi di MATLAB / Simulink

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	ABSTRAK	iii
	TABLE OF CONTENTS	iv
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
	1.1 Introduction	1
	1.1.1 Coupled Tank Water Level System	1
	1.1.2 Parameter Identification	2
	1.1.3 Particle Swarm Optimization	2
	1.2 Problem Statement	3
	1.2.1 Coupled Tank Liquid Level System	3
	1.2.2 Programmable Logic Controller	4
	1.2.3 Parameter Identification	4
	1.3 Objectives	4
	1.4 Scope of Project	4

2	LITERATURE REVIEW	6
2.1	Introduction	6
2.2	Coupled Tank Water Level System	6
2.2.1	Coupled Tank Water Level System Application	7
2.3	Parameter Identification	7
2.3.1	Parameter Identification Application	7
2.4	Particle Swarm Optimization	8
2.5	Programmable Logic Controller	9
2.5.1	PLC Application	9
2.6	Matlab Simulink	10
2.7	Graphical User Interface (GUI)	10
3	METHODOLOGY	11
3.1	Introduction	11
3.2	Phase I Project Preview	12
3.2.1	Optimization Algorithm of Coupled Tank Water Level System	13
3.3	Phase II Data Collection	13
3.3.1	Power Consumption	15
3.3.2	PLC CJ1M Setting	16
3.3.3	Analog Input and Analog Output Setting	19
3.3.4	Analog Output Card	19
3.3.5	Analog Input Card	22
3.3.6	Coupled Tank Water Level System	24
3.4	Phase III Modeling Equation of Coupled Tank	25

3.5	Phase IV Data Simulation	28
3.5.1	Parameter Identification	30
3.5.2	Particle Swarm Optimization	30
3.5.3	Particle Swarm Optimization Pseudocode	32
3.5.4	Particle Swarm Optimization with GUI	32
	Visual Basic	
4	RESULTS AND ANALYSIS	35
4.1	Introduction	35
4.2	Project Result	35
4.3	Simulation Result	38
4.3.1	Particle Swarm Optimization in MATLAB	41
	Simulink	
5	CONCLUSION AND RECOMMENDATION	47
5.1	Introduction	47
5.2	Recommendations	48
5.2.1	Recommendation in Technique	48
5.2.2	Recommendation in Hardware improvement	48
5.2.3	Recommendation in Approach	49
5.3	Costing and Commercialization	49

REFERENCES	50
APPENDICES	51
APPENDIX A - PARTICLE SWARM OPTIMIZATION MATLAB CODING	
APPENDIX B - GUI DATA COLLECTION CODING	59

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 3.1	Maximum power consumption for CJ1M	15
Table 3.2	Power Consumption for the PLC	15
Table 3.3	Power Consumption needed by group	16
Table 3.4	Conversion of level transmitter into voltage	25
Table 4.1	Calculated level value based on voltage span	38

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 3.1	Overall Flowchart of the project	12
Figure 3.2	Real Time Data Collection	14
Figure 3.3	Basic Construction of CJ1M PLC	16
Figure 3.4	The IO Table and Unit Setup	17
Figure 3.5	Select Slot Address Type	18
Figure 3.6	Output Table Edit Parameter	18
Figure 3.7	Input Table Edit Parameter	18
Figure 3.8	Analog output card	19
Figure 3.9	Analog input card	19
Figure 3.10	Analog Output Card	20
Figure 3.11	Operation mode switch and unit number switch	20
Figure 3.12	DM and CIO output card setting	21
Figure 3.13	Analog Input Card	22
Figure 3.14	DM and CIO analog input card	23
Figure 3.15	Coupled Tank Water Level	24
Figure 3.16	Block diagram of second order process	26
Figure 3.17	Flowchart to obtain mathematical equation for Coupled Tank System	29

Figure 3.18	Framework of Estimation procedure	30
Figure 3.19	Data logger using GUI visual basic	33
Figure 3.20	Graph water level Vs time for MV	34
Figure 3.21	Graph of water level Vs time for CV	34
Figure 4.1	The Ladder Diagram of CX-Programmer	36
Figure 4.2	GUI visual basic during debug with Data Logger and graphs.	37
Figure 4.3	Data Model	41
Figure 4.4	Periodic signal	42
Figure 4.5	One cycle of periodic signal	42
Figure 4.6	Training data and transfer function outputs	43
Figure 4.7	Train model	44
Figure 4.8	Block Diagram for comparison	44
Figure 4.9	Particles in searching for best positions and velocities	45
Figure 4.10	Scope at train model	45

LIST OF ABBREVIATIONS

PLC	Programmable Logic Control
SISO	Single Input Single Output
MIMO	Multi Input Multi Output
Analog I/O	Analog Input Output
DM	Data Memory
CPU	Computer Processing Unit
PSO	Particle Swarm Optimization
GUI	Graphical User Interface
MSE	Mean Square Error
MV	Manipulated Variable
CV	Controlled Variable
SP	Set Point

CHAPTER 1

INTRODUCTION

1.1 Introduction

1.1.1 Coupled Tank Liquid Level System

The coupled tank apparatus CTS-001 is a computer-controlled coupled-tank system used for liquid level control. The concept of virtual instrumentation is introduced in CTS-001. By using virtual instrumentation techniques, the need for traditional dedicated user interfaces on individual instruments can be eliminated [1]. Instead, the computer can be used to provide the interface normally required on each of the individual instruments, reducing cost and complexity. Virtual instrumentation also allows the user to make major modification to the instrument by simply changing a software program. For example, the computer can be used to carry out the functions of an oscilloscope for displaying input and output response.

1.1.2 Parameter Identification

A Parameter Identification or in the other word as model is a set of mathematical equations that are intended to capture the effect of certain system variables on a certain other system variables. It is usually neither possible nor necessary to model the effect of every variable on every other variable; one therefore limits oneself to certain subsets, for example, the effect of disturbance on output [2]. A model is never perfect and it is therefore always associated with a modeling error. The importance of this mathematical model implemented for the Coupled Tank Water Level systems are it is frequently observed in practice in control plant in today's larger industries and it can be easily use in future when doing MATLAB simulation.

The most traditional system identification techniques are prediction error method (PEM) and the instrument variable method (IVM). Though these traditional identification techniques offer good solution to many real-life systems, they have certain issues like difficulty in determining the model structure and numerical reliability due to need of solving the multidimensional nonlinear optimization problem in PEM case or system of linear equations in IVM case [3]. In another research, it introduced Transfer Function Matrix which is based on the transfer function approach, for example, direct polynomial form of TF and direct frequency domain identification.

1.1.3 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithms are populated-based research algorithm based on the simulation of the social behavior of birds within a flock [4]. They work in the same way, which is, updating the population of individuals by applying some kind of operators according to the fitness information obtain from the environment so that the individuals of the population can be expected to move toward better solution areas [5]. PSO is one from other Intelligent Method of Parameter Identification that is

famously used in today control process because it has stochastic search capability and robustness [6].

Thus, in this paper, it proposes the use of Particle Swarm Optimization in identifying a fitness function for the mathematical equation. The objectives of this project are to identify and obtain the mathematical characteristic of the Coupled Tank Liquid Level system using a suitable parameter identification technique which is the Intelligent Method. Secondly is to develop GUI which will display the actual system performance (in terms of graphs) along with the model performance. In addition to that, both performances will also be analyzed in terms of transient and steady-state responses.

1.2 Problem Statement

There are three major statements that have been emphasized in this project which are Coupled Tank Water Level System, Programmable Logic Controller and Parameter Identification.

1.2.1 Coupled Tank Liquid Level System

Coupled Tank is very special and unique due to the coupled function. There are two major applications in Coupled Tank which is SISO (Single Input Single Output) and MIMO (Multiple Input Multiple Output). The level of water entering tank 1 will be affected by the water level in tank 2. In industrial control application, it is often required to control Coupled Tank Water Level System. It is very popular and widely used in control process industries.

1.2.2 Programmable Logic Controller

Programmable Logic Controller can be thought of as Industrial Computers with specially designed architecture in both their central units (the PLC itself) and their interfacing circuitry to field devices (input /output connections to the real world). PLC is able to received analog external input and produced analog output for its functionality.

1.2.3 Parameter Identification

In Parameter Identification, Particle Swarm Optimization Method is applied because of its free derivation. It required less number of computations to achieve same error goals. PSO varied in real-time, so it is easy to use in software simulation.

1.3 Objective

The overall objectives of this project are;

1. To identify and obtain the Mathematical Model of Coupled Tank Liquid Level System using Particle Swarm Optimization Method in terms of its transfer function.
2. To estimate the parameter coefficient of the transfer function utilizing an intelligent technique namely Particle Swarm Optimization.

1.4 Scope of Project

The scopes of this project are;

1. To determine the modeling mathematical equation of Coupled Tank Water Level System using Particle Swarm Optimization.

2. To compare the modeling output with simulated output to obtain error less between both output.
3. To develop GUI that will display the simulated and model performance in terms of graph
4. To analyze the model performance in terms of transient and steady-state response.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, all the important information related to this project are reviewed. The sources of this information are mainly from journals, conference papers, manuals, and website. Besides that, the literature review can give a brief explanation about Parameter Identification and the importance of using Particle Swarm Optimization in order to obtain the mathematical equation which best represents the Coupled Tank Liquid Level System.

2.2 Coupled Tank Liquid Level System

The reservoir contains two PWM operated motor pumps. These pumps use either 0-5V analog voltage (internal signal conditioning system convert analog to PWM (digital signal)) or external PWM sources for their application. Flow rates of water into tank can be varied by change of these pump voltages. There is a baffle plate separating

the two tanks which can be slide up and down to vary interaction and coupling capacitive probes, one in each tank, provided to measure water level. Output signals from these probes are conditioned to give 0-5V analog output. A water outlet at side near base of each tank connected by a flexible tube returns water to reservoir [1].

2.2.1 Coupled Tank Liquid Level System Application

The industrial application of liquid level control is tremendous especially in chemical process industries. Usually, level control exists in some of the control loops of process control system. Otherwise in many other industrial applications are concerned with level control, may it be a single loop level control or sometimes multi-loop level control. In some cases, level controls that are available in the industries are for interacting tanks. Hence, level control is one of the control system variables which are very important in Process Industries.

2.3 Parameter Identification

In the design of adaptive control system, it is sometime necessary to estimate parameters for systems represented by state-space response [2]. Some generic and user-friendly tools are available for model identification. For example, System Identification Toolbox of MATLAB that contains various techniques for identifying time-series and state space, non parametric and parametric models, is a popular identification tool among researchers and academicians.

2.3.1 Parameter Identification Application

Once the system (plant) generate the output in terms of graph , the usage of Parameter Estimation problem is to find among all possible choices a particular set of parameters which best describes the system. Depending on one's definition of "best"

there are different approaches to such a Parameter Identification problem [2]. In this research, Intelligent Method is used instead of Classical Method. In Intelligent Method, Particle Swarm Optimization is selected to be used in this research based on the popularity of this method in process industry.

2.4 Particle Swarm Optimization

PSO algorithm was developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer), which is a Robust Stochastic Optimization technique based on the movement and intelligence of swarms. It uses a number of particles that constitute a swarm moving around in the search space looking for the best solution [6].

In PSO algorithm, a swarm consists of m -particle, in which each particle is treated as a point in an N -dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles. Each particle uses velocity to determine the direction and value of its “flying”, which follow the current optimum in an N -dimension space [6].

Each particle keeps track of its coordinates in the solution space which are associated with the best solution that has achieved so far by that particle. This value is called personal best $Pbest$. Another best value that is tracked by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called global best, $Gbest$. The basic concept of PSO lies in accelerating each particle toward its personal best and the global best locations.

PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is

important to the algorithm. In PSO, only Gbest gives out the information to others. It is a one way information sharing mechanism. The evolution only looks for the best solution.

2.5 Programmable Logic Controller

A programmable logic controller (PLC) is an industrial computer used to control and automate complex systems. All CJ1 series use the same instruction set and I/O modules, so existing programs and equipment can be easily reused. A common memory area can help integrate processes or coordinate various activities. Besides, CJ1M also provide the analog output with the current rating from 4mA to 20mA which the CQM1H cannot provide both of them [7].

Programmable logic controllers are a relatively recent development in process control technology. It is designed for use in an industrial environment, which uses a programmable memory for integral storage of user-oriented instructions for implementing specific functions such as logic, sequencing, timing, counting, and arithmetic to control through digital or analog inputs and outputs, various types of machines or processes [8].

2.5.1 PLC Application

Programmable logic controllers are widely used in industry and process control. Programmable logic controllers are used in a wide spectrum of applications from factory automation to waste water treatment plant controls and from chemical process plant control to engine management systems. PLCs are used throughout industry to control and monitor a wide range of machines and other movable components and systems. As a part of process control, a PLC is used to monitor input signals from a variety of input points (input sensors) which report events and conditions occurring in a controlled process. For example, a PLC can monitor such input conditions as motor speed, positioning, temperature, pressure and volumetric flow. Its associated peripherals are

designed so that they can be easily integrated into an industrial control system and easily used in all their functions [9].

2.6 Matlab Simulink

Simulink is an environment for multi-domain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing. Simulink provides a graphical user interface for building math models as block diagrams. The graphical interface is popular for developing dynamical models for many fields, such as electronics, hydraulics, chemistry, and especially, control systems.

2.7 Graphical User Interface (GUI)

Microsoft released Visual Basic in 1987 and the latest is in 2008. It was the first visual development tool from Microsoft, and it was to compete with C, C++, Pascal and other well-known programming languages. Programmers have undergone a major change in many years of programming various machines. Overall, it can class Visual Basic as a Graphical User Interface (GUI). A GUI is a graphical (rather than purely textual) user interface to a computer.

Today's major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own graphical user interface elements and ideas. When creating an application, many object-oriented tools exist that facilitate writing a graphical user interface. Each GUI element is defined as a class widget from which you can create object instances for your application.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describe about the flow of this project and the method used to achieve the project objectives. Methodology is the most important and crucial element in a project making as it is needed to guide the development of the project and to keep project finished on time. It is important to achieve the objectives that have been stated or in another word, the project follows the guideline based on the objectives. This chapter has been divided into four phase which is completed with guidance and method for each phase;

Phase I : Project Preview

Phase II : Data Collection

Phase III : Modeling Equation of Coupled Tank System

Phase IV : Data Simulation

3.2 Phase I: Project Preview

This project proposes the use of Particle Swarm Optimization in identifying a fitness function for the mathematical equation. This mathematical equation is best representing the Coupled Tank Liquid Level System. The needs of mathematical equation which can represent the real plants will make the computerized simulation easier. The developed of GUI will display the actual system performance (in terms of graphs) along with the model performance. In addition to that, both performances will also be analyzed in terms of transient and steady-state responses. Figure 3.1 shows the overall flow chart of the project.

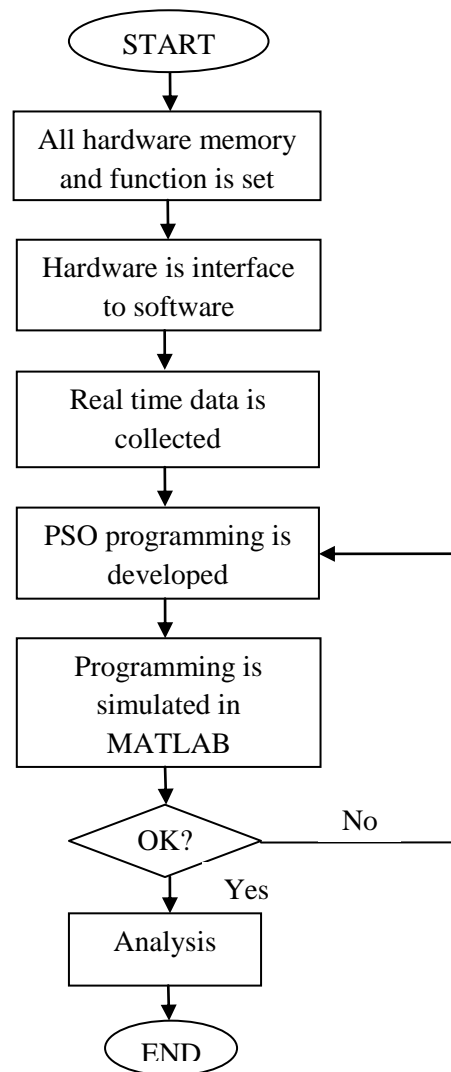


Figure 3.1: Overall flowchart of the project

3.2.1 Optimization Algorithm of Coupled Tank Water Level System

Particle Swarm Optimization Method algorithm find the most optimum position and velocity of its particle in order to get the fitness value. The best fitness function is known for its smallest error compared to set point. For the error comparison, Mean Squared Error is used. The MSE fitness function is based on the absolute error. The mean squared error E_i of an individual program i is evaluated by the following equation (3.1):

$$E_i = \frac{1}{n} \sum_{j=1}^n (P_{ij} - T_j)^2 \quad \text{----- (3.1)}$$

$P_{(ij)}$ is the value predicted by the individual program i for fitness case j (out of n fitness cases or sample cases); and T_j is the target value for fitness case j . For a perfect fit, $P_{(ij)} = T_j$ and $E_i = 0$. So, the **MSE** index ranges from 0 to infinity, with 0 corresponding to the ideal. As it stands, E_i cannot be used directly as fitness since, for fitness proportionate selection, the value of fitness must increase with efficiency.

3.3 Phase II: Data Collection

In this phase of data collection, the periodic signal of water level and time for both MV and CV performance are collected in GUI Visual Basic. The data is then saved in Excel for further used in Matlab Simulink. Figure 3.2 above shows the flow in collecting real time data from PLC and GUI Visual Basic.

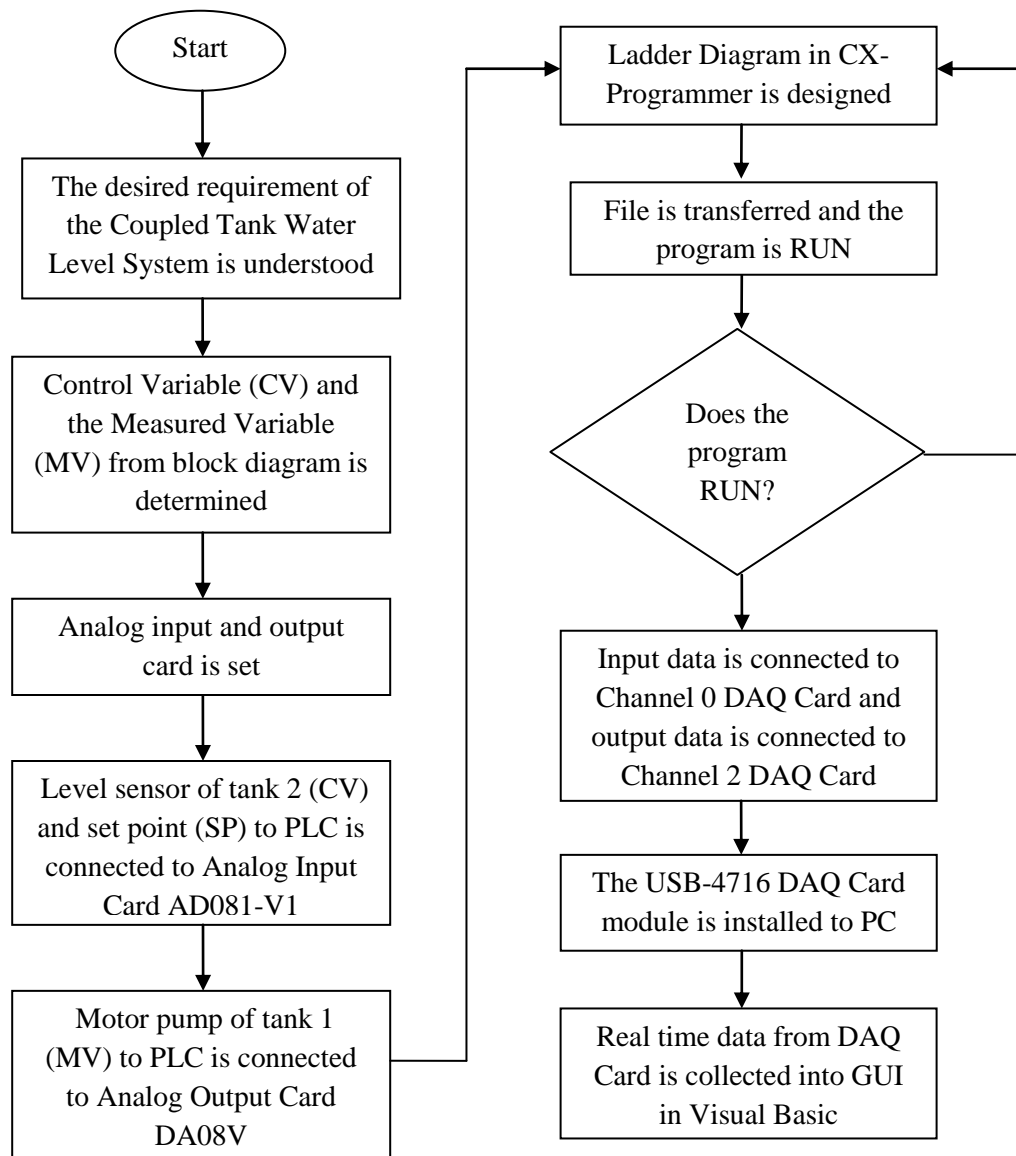


Figure 3.2: Real time data collection from PLC and GUI Visual Basic

3.3.1 Power Consumption

The power circuit is designed for easy installation and wiring in PLC panel. Before the wiring of the PLC started, the power consumption should be calculated. This is to know the value of the current that are needed as well as to make sure that the values do not exceed the maximum capacity. It is also to know the rating of the main circuit breaker (MCB) that is used. Table 3.1, Table 3.2 and Table 3.3 below shows the power consumption by CJ1M, PLC and group.

Table 3.1: Maximum power consumption for CJ1M

Power supply unit	Maximum current consumption			Maximum total power consumption
	5V group (internal logic)	24V group (relay)	24V group (service)	
CJ1W-PA202	2.8A	0.4A	None	14W

Table 3.2: Power Consumption for the PLC

Unit	Model	Quantity	Voltage Group	
			5VDC	24V DC
CPU	OMRON SYSMAC CJ1M – CPU12	1	0.58A	-
Digital Output (DO)	CJ1W – OC211	1	0.11A	0.096A
Digital Input (DI)	CJ1W – ID211	1	0.08A	-
D/A unit (AO)	CJ1W – DA08V	1	0.14A	-
A/D unit (AI)	CJ1W – AD081 - VI	1	0.42A	-

Table 3.3: Power Consumption needed by group

Group	Current Consumption
5V DC	= 0.58 + 0.11 + 0.08 + 0.14 + 0.42 = 1.33A ≤ 2.8A (refer table 3.4)
24V DC	= 0.096A ≤ 0.4A (refer table 3.4)

Therefore, total power consumption CJ1M CPU-12 Unit using equation (2);

$$\begin{aligned}
 P &= (1.33 \times 5) + (0.096 \times 24) && \text{----- (3.2)} \\
 &= 8.954\text{W} \leq 14\text{W (refer table 3.1)}
 \end{aligned}$$

Thus there is enough value of the current that are needed as well as these values do not exceed the maximum power capacity.

3.3.2 PLC CJ1M Setting

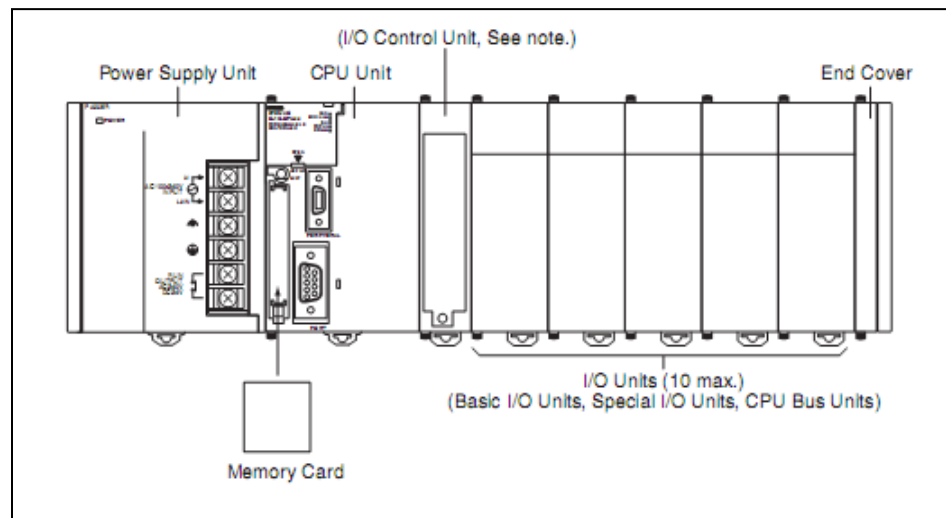


Figure 3.3: Basic construction of CJ1M PLC

Figure 3.3 above give information of the CJ1M consist of one Power Supply Unit, one CPU Unit, one I/O Control Unit and additional I/O Units. Before the setting of

the CJ1M is set, the PLC software should be set first. The procedure to set the CX-Programmer is as follow. First, in CX-Programmer window click on an IO table and unit setup and go to main rack. The formation of PLC card were refer while execute this setting. Figure 3.4 of the IO Table and Unit Setup in CX Programmer were clicked to access this setting. The rack formation in this project was shown below:

First rack : CJ1W – DA08V (Analog input unit 8 point/voltage)

Second rack : CJ1W – AD081-V1 (Analog input unit 8 point)

Third rack : CJ1W – ID211 (DC Input Unit)

Fourth rack : CJ1W – OC211 (DC output Unit)

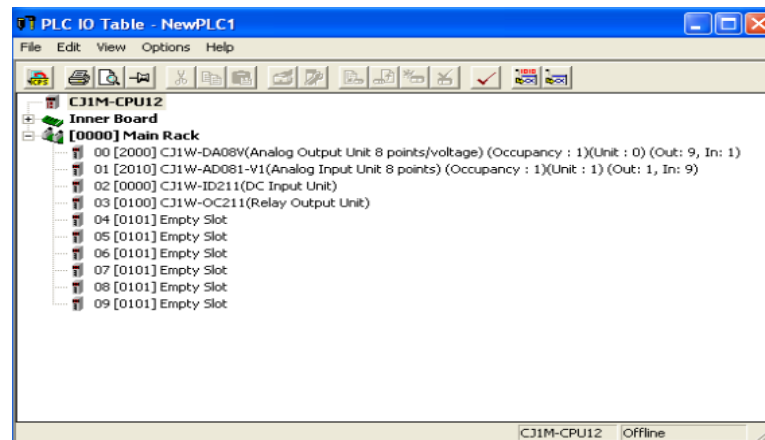


Figure 3.4: The IO Table and Unit Setup

After four analog and digital cards are selected, Options is clicked and Slot/Rack Address is chosen. One more Slot Start Address is added with start address 100. This setting applied for input and output form in ladder diagram because for input, it have written as I: 0.00 and output Q: 100.03. Figure 3.5 below show the selected Slot Address Type.

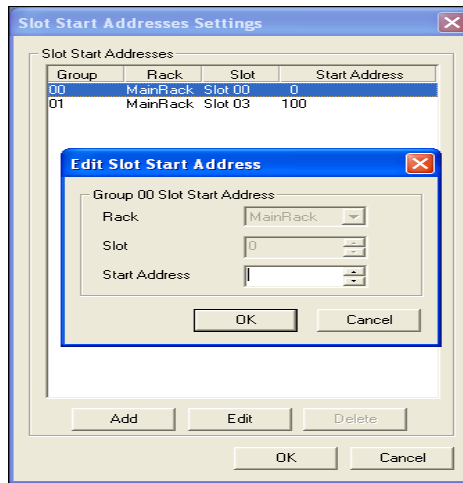


Figure 3.5: Select Slot Address Type

For analog input and output card, the number of input signal use setting and output use setting is enabled based on its number being used. For voltage application, input range setting is set to 0-5V and output signal range setting is set to 0-5V. Figure 3.6 and figure 3.7 of Input Output Edit Parameter show how it has been set.

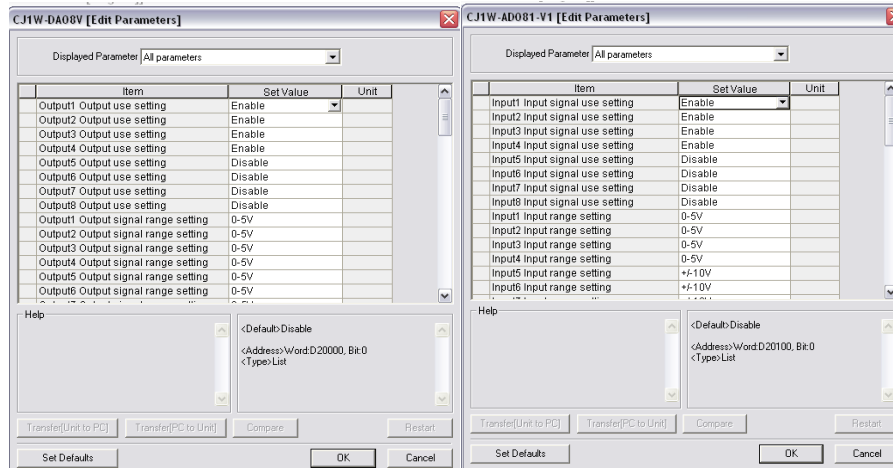


Figure 3.6 & 3.7: Input Output Table Edit Parameter

3.3.3 Analog Input and Analog Output Setting

In order to start using the PLC CJ1M, it is compulsory to set the entire setting requirement for it. The different of CJ1M as compared to the CQM1H is CJ1M necessary to set the entire input output card that are need to use while the CQM1H setting is automatically set. This is due to the availability of the special input card and special output card in CJ1M family. Therefore, this part is the crucial stage and it required to be successfully completed before the PLC is ready to be used. Figure 3.8 show the hardware drawing of analog output card and figure 3.9 show the hardware drawing for analog input card.

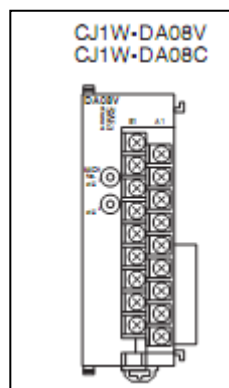


Figure 3.8: Analog output card

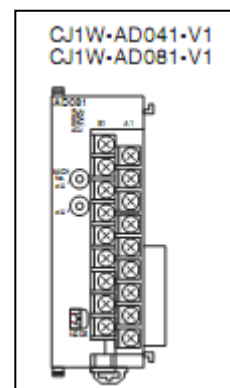


Figure 3.9: Analog input card

3.3.4 Analog Output Card

From the manual, there are several steps that have been followed in order to set up the analog output card. This output card is provided with four output channels with one channel is not use. The configuration of the card is as shown in figure 3.10.

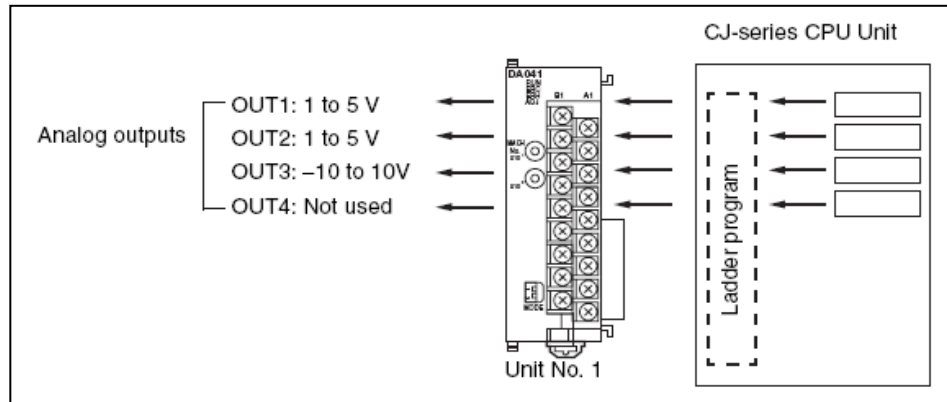


Figure 3.10: Analog Output Card

Motor pump in the Coupled Tank System only receives 0V-5V voltage from analog output card. For 0V-5V output voltage, CJ1W-DA08V output card will give hexadecimal range from #0000 until #0FA0. If the set value is outside the specified range provided below, an output setting error will occur, and the output specified by the output hold function will be output. The operation mode must turn on OFF for normal mode. Figure 3.11 show the operation mode switch and unit number switch.

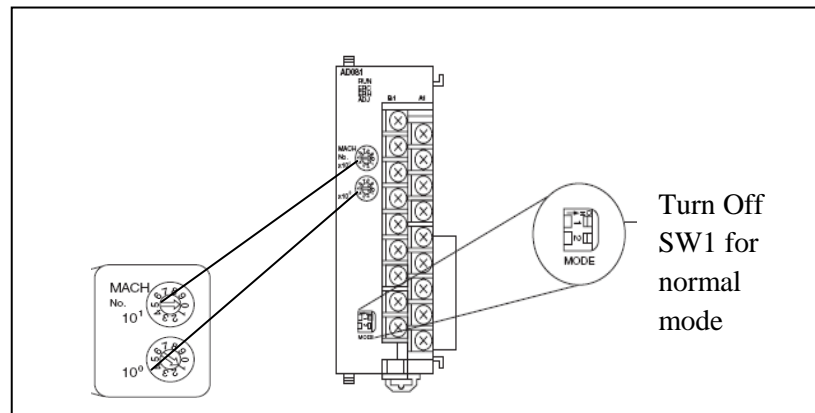


Figure 3.11: Operation mode switch and unit number switch

If two or more Special I/O Units are assigned the same unit number, a “UNIT No.DPL ERR” error (in the Programming Console) will be generated (A40113 will turn ON) and the PLC will not operate.

The CPU Unit and Analog Output Unit exchange data via the Special I/O Unit Area and the Special I/O Unit DM Area. Figure 3.12 show DM and CIO output card setting. The Special I/O Unit Area and Special I/O Unit DM Area word addresses that each Analog Output Unit occupies are set by the unit number switch on the front panel of the Unit.

- The unit number switch is turned to $0x10^1$ and $0x10^0$.
- The words will be allocated to the Analog Output Unit in Special I/O Unit Area CIO 2000 to CIO 2009 and in the Special I/O Unit Area D20000 to D20099.
- The allocation of DM words and bits for normal mode is [D (20000) = #00FF] and the allocation of CIO is [CIO (2000) = #000F]. Appendix is attached for further understanding on how this hexadecimal bit is set.

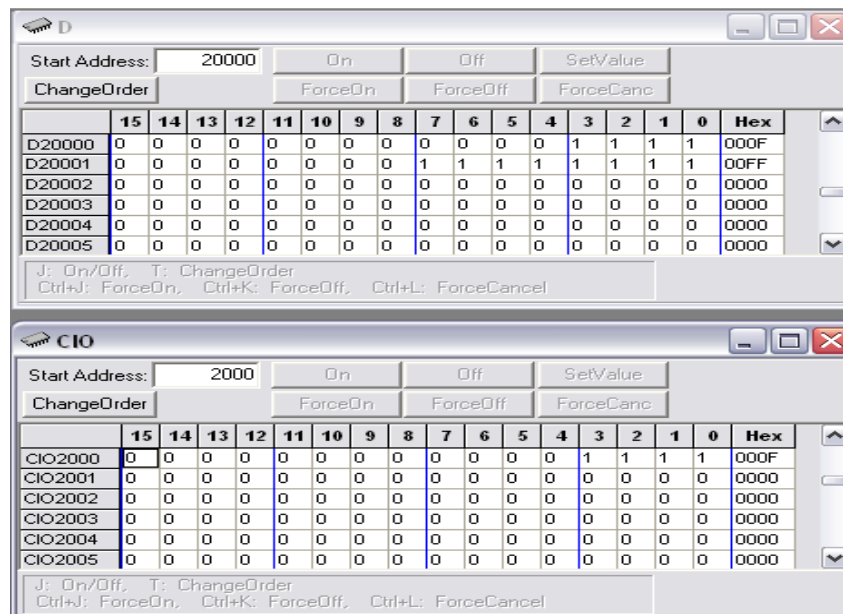


Figure 3.12: DM and CIO output card setting

3.3.5 Analog Input Card

According to the manual that are downloaded through the internet, there are some procedures or steps to set the input card. The input card provides eight channels with one channel are not used as shown in figure 3.13.

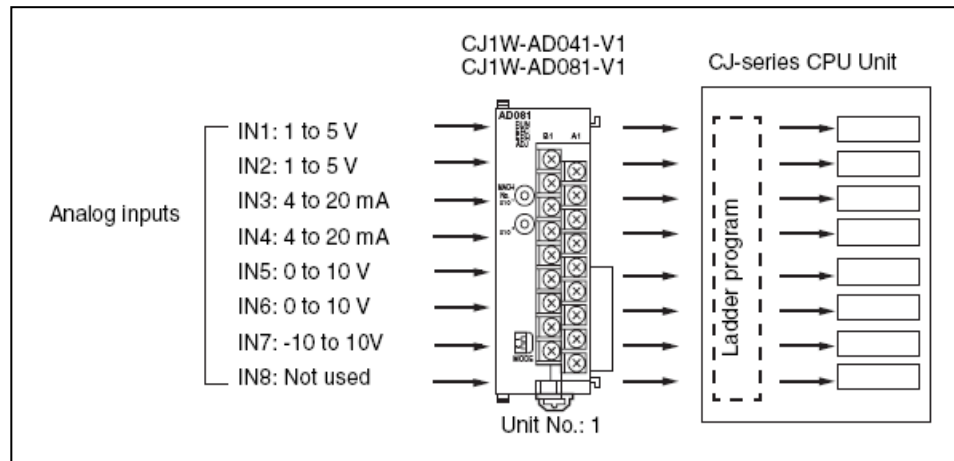


Figure 3.13: Analog Input Card

Input voltage from voltage injector is analog. It must be process into digital bit in order to make PLC understand the data for 0V-5V input voltage. Thus CJ1W-AD081-V1 input card will give hexadecimal range from #0000 until #0FA0. If the set value is outside the specified range provided below, an input setting error will occur, and the input specified by the input hold function will be input. The operation mode must turn on OFF for normal mode

The CPU Unit and Analog Input Unit exchange data via the Special I/O Unit Area and the Special I/O Unit DM Area. The Special I/O Unit Area and Special I/O Unit DM Area word addresses that each Analog input Unit occupies are set by the unit number switch on the front panel of the Unit. Figure 3.14 show the result of DM and CIO analog input card setting;

- The unit number switch is turned to $0x10^1$ and $1x10^0$.
- The words will be allocated to the Analog input Unit in Special I/O Unit Area CIO 2010 to CIO 2019 and in the Special I/O Unit Area D20100 to D20199.
- The allocation of DM words and bits for normal mode is [D (20100) = #000F] and the allocation of CIO is [CIO (2010) = #000F]. Appendix is attached for further understanding on how this hexadecimal bit is set.

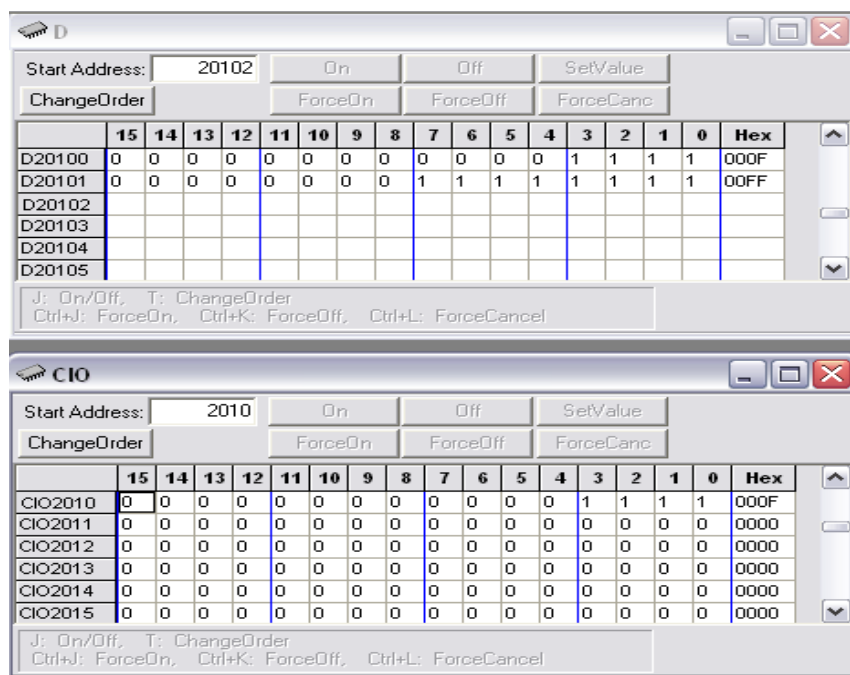


Figure 3.14: DM and CIO analog input card

If two or more Special I/O Units are assigned the same unit number, a “UNIT No.DPL ERR” error (in the Programming Console) will be generated (A40113 will turn ON) and the PLC will not operate.

3.3.6 Coupled Tank Water Level System

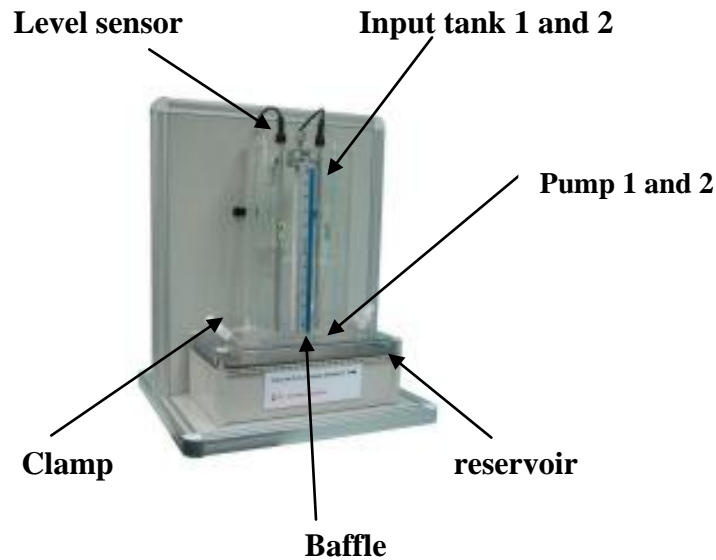


Figure 3.15: Coupled Tank Water Level

Water in reservoir is pumped into the top of each tank by two independent pumps. The head of water in each tank is clearly visible on the attached scale at the front of the tanks. Each tank is fitted with an outlet, at the side near the base. The amount of water which returns to the reservoir is approximately proportional to the head of water in the tank since the return tube at the base of the tank functions as a pseudo-linear hydraulic resistance. The return tube is constructed of flexible tubing so that the resistance may be increased by the use of a screw-type clamp. Figure 3.15 above show the coupled tank labeling.

The level of water in each tank is monitored by a capacitive-type probe. Signal conditioning circuits convert the measured capacitance (a function of the water level) to electrical signals in the range 0 to +5 Volt DC. The zero level has been calibrated to represent the rest point of the water level, that is, when the tank is nearly empty (approximately 20mm on the scale), while the full state (+5V) is calibrated at the level

of the opening to the rear overflow stands-pipe. This occurs at approximately 300mm on the scale.

The level transmitter converted water level range from millimeter (mm) to voltage in range 0V to 5V in order to make PLC understandable. This is because PLC either configured voltage (0V – 5V) or current (4mA – 20mA). Table 3.4 show the conversion of level transmitter into voltage with comparison to calculated voltage.

Table 3.4: Conversion of level transmitter into voltage

WATER LEVEL (mm)	VOLTAGE (Volt)	
	(measured)	(calculated)
300	5.20	5.00
250	4.40	4.17
200	3.60	3.33
150	2.70	2.50
100	1.70	1.67
50	0.90	0.83
0	0.00	0.00

3.4 Phase III: Modeling Equation of Coupled Tank

The Coupled Tank Liquid Level Control System applied second order single input single output system. If the baffle between Tank 1 and Tank 2 is opened slightly, allowing water to flow through it from Tank 1 and Tank 2, the manipulated variable of the control system is now a second order system and the block diagram is given in Figure 3.16. In this configuration the flow into Tank 2 will only be from Tank 1 through

the baffle opening and not from the inlet of Tank 2. Therefore q_2 inflow into Tank 2 is zero for this case.

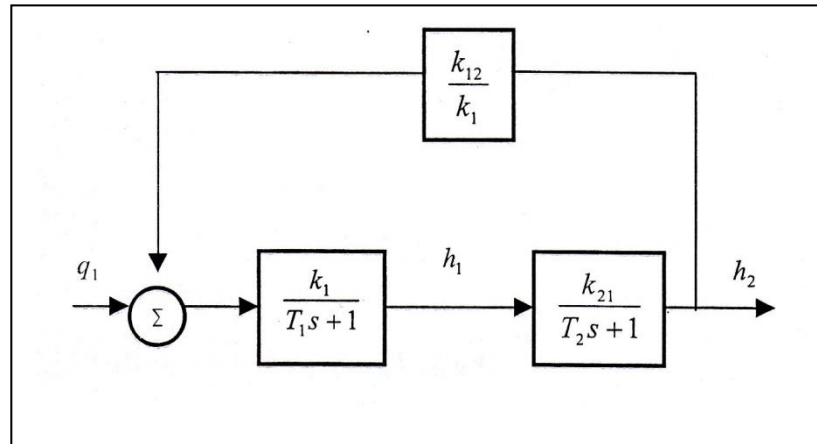


Figure 3.16: Block diagram of second order process

Initialized value;

$$A_1 = 32 \text{ cm}^2 \quad \alpha_1 = 10.78$$

$$A_2 = 32 \text{ cm}^2 \quad \alpha_2 = 11.03$$

$$H_1 = 17 \text{ cm} \quad \alpha_3 = 11.03$$

$$H_2 = 15 \text{ cm}$$

Time constant of the tank dynamics is;

$$T_1 = \frac{2A_1\sqrt{H_1}}{\alpha_1} \quad \text{----- (3.3)}$$

$$T_2 = \frac{2A_2\sqrt{H_2}}{\alpha_2} \quad \text{----- (3.4)}$$

Notice that both the time constant of the tank dynamics are dependent on the operating point H_1 and also the constant of proportionality α_1 . Also the time constant is dependent on the physical design of the tank. For example, the proportionality constant α_1 is dependent on the cross-sectional area of the orifice. It has been shown that the time constant of a dynamic system will influence the transient performance of the system. Therefore, physical design plays an important role in determining the response of a system.

Constant of proportionality are;

$$\alpha_1 = \frac{2A_1\sqrt{H_1(0)}}{T_1} - \frac{2A_1\sqrt{H_1(t)}}{T_1} \quad \text{----- (3.5)}$$

$$\alpha_2 = \alpha_3 = \frac{2A_2\sqrt{H_2(0)}}{T_2} - \frac{2A_2\sqrt{H_2(t)}}{T_2} \quad \text{----- (3.6)}$$

$$K_1 = \frac{1}{\frac{\alpha_1}{2\sqrt{H_1}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (3.7)}$$

$$K_2 = \frac{1}{\frac{\alpha_2}{2\sqrt{H_2}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (3.8)}$$

$$K_{12} = \frac{\frac{\alpha_3}{2\sqrt{H_1 - H_2}}}{\frac{\alpha_1}{2\sqrt{H_1}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (3.9)}$$

$$K_{21} = \frac{\frac{\alpha_3}{2\sqrt{H_1 - H_2}}}{\frac{\alpha_2}{2\sqrt{H_2}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (3.10)}$$

The Transfer Function for the Coupled Tank dynamics is given as;

$$\begin{aligned} \frac{h_2}{q_1} &= \frac{k_1 - k_{21}}{(T_1 s + 1)(T_2 s + 1) - k_{12} k_{21}} \\ &= \frac{k_1 - k_{21}}{T_2 T_1 s^2 + (T_1 + T_2)s + (1 - k_{12} k_{21})} \end{aligned} \quad \text{----- (3.11)}$$

Later, this transfer function will be used in MATLAB Simulink Modelling.

3.5 Phase IV: Data Simulation

The periodic signal which is obtained during second phase of data collection is observed in oscilloscope for right shape and time signaling. Because of periodic signal has infinity repeated almost similar waveforms, we can use only one time cycle wave for training data in Matlab. This training data must be valid and correct. In Matlab, Training Model block diagram and Data Model block diagram is built while Particle Swarm Optimization Algorithm is computed. Figure 3.17 below show a flowchart to obtain mathematical equation for Coupled Tank System.

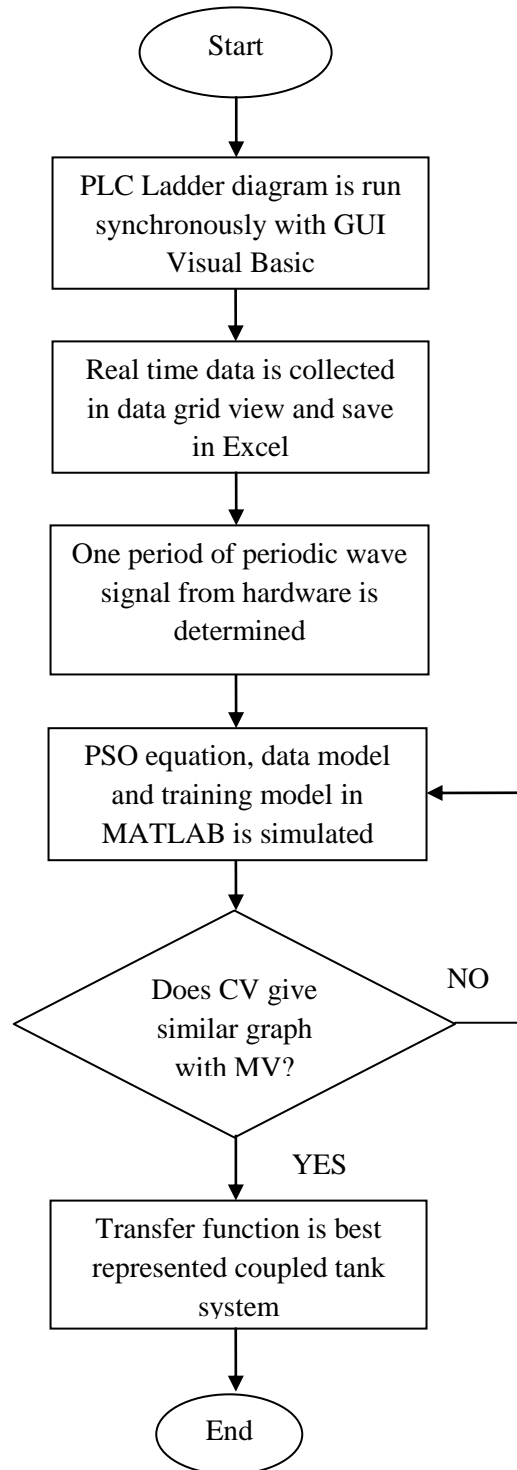


Figure 3.17: Flowchart to obtain mathematical equation for Coupled Tank System

3.5.1 Parameter Identification

Both outputs of actual (simulate) performance and modeling performance is compared in order to obtain error less output. As long as the Minimum MSE is not observed, the parameter dependents need to be changed or updated. The mathematical equation obtained should be best representing system's characteristic. Figure 3.18 below show the framework of Estimation Procedure.

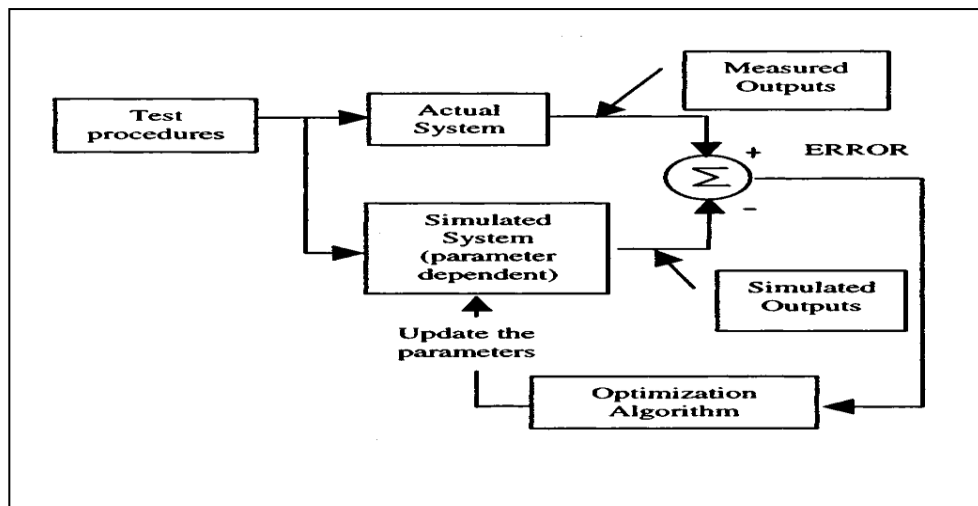


Figure 3.18: Framework of Estimation procedure

3.5.2 Particle Swarm Optimization

As stated before, PSO simulates the behavior of bird flocking. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a bird in the search space. We call it particle. All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. Below equation is used in initialized the particles velocities and positions;

$$x=a+(b-a)*\text{rand}(N,D,1); \quad \% \quad \text{initialization of } x \quad \text{----- (3.10)}$$

$$V=m+(n-m)*\text{rand}(N,D,1); \quad \text{initialization of } v \quad \text{----- (3.11)}$$

In general, there are 3 major algorithms in Particle Swarm Optimization (PSO) which are Individual Best (*pbest*). Each individual compares position to its own best position. Then Global Best (*gbest*) is a social knowledge used to drive the movement of particles included the position of the best particle from the entire swarm. Lastly when a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*. After finding the two best values, the particle updates its velocity and positions with following equation;

$$V(:, :, 2) = W(1) * V(:, :, 1) + c1 * \text{rand} * (pbest(:, :, 1) - x(:, :, 1)) + c2 * \text{rand} * (G(:, :, 1) - x(:, :, 1));$$

----- (3.12)

$$x(:, :, 2) = \text{abs}(x(:, :, 1) + V(:, :, 2));$$

----- (3.13)

V is the velocity, present is the current particle (solution). *pbest* and *gbest* are defined as stated before. *rand* is a random number between (0,1). *c1* and *c2* are learning factors, usually $c1=c2= 1.4$.

Steps to obtain Particle Swarm Optimization algorithms:

1. Initialize swarm
2. Evaluate performance of each particle
3. Compare (*pbest*) to its best performance
4. Compare (*pbest*) to (*gbest*)
5. Change velocity vector of each
6. Move each particle to new position
7. Repeat step 2 until convergence

3.5.3 Particle Swarm Optimization Pseudocode

The pseudo code of the procedure is as follows

For each particle

Initialize particle

End

Do

For each particles

Calculate fitness value

If the fitness value is better than the best fitness value (pbest) in history set current value as the new pbest.

End

Choose the particle with the best fitness value of all the particles as the gbest

For each particle

Calculate particle velocity according equation (i)

Update particle position according equation (ii)

End

3.5.4 Particle Swarm Optimization with GUI Visual Basic

Like the basic programming language, Visual Basic was designed to be easily learned and used by beginner programmers. The language does not only allow programmers to create simple GUI applications, but can also develop complex applications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality.

This GUI in figure 3.19 consisted of various button for various application. It is built independently based on application. Only programmers understand their own program, but with GUI one can develop a form with various applications and interface with another hardware using special module provided together in Visual Basic. For this project, all data collection is exported to excel for further analysis.

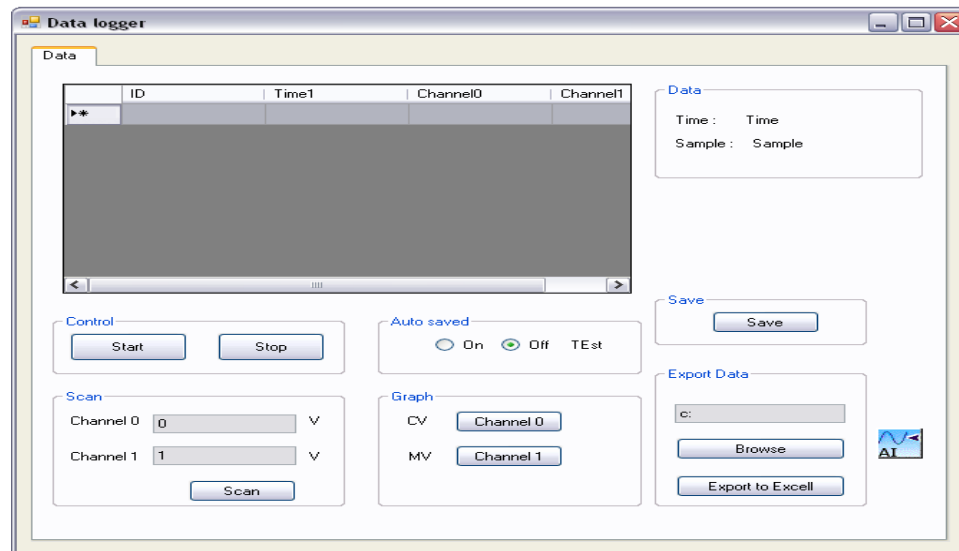


Figure 3.19: Data logger using GUI visual basic

The used of every detail of toolbox in Data logger form;

- **Control** button is clicked to start data collecting from DAQ Card and stop button is clicked to stop the data collecting.
- **Scan** button is clicked to scan voltage value in channel 0 and channel 1 of DAQ Card.
- In **data group** box, it will display the sampling time and total number of sample that was taken.
- In **auto saved** group box, we can select either to be in ON or OFF mode. When we choose to OFF auto save, the collected data will be manually save when we clicked the **Save** button. These data were saved in data logger file created at desktop and exported to Excel form.
- The **data grid view** is divided into 4 column which is ID, Time 1, Channel 0 and Channel 1. All data collecting is uploaded into this data grid view.
- **Channel 0** button consist a graph of Level Vs Time for MV **while Channel 1** button consist a graph of Level Vs Time for CV.
- These two graphs display real-time data collecting while the coupled tank behavior is observed in below figure of figure 3.20 and figure 3.21.

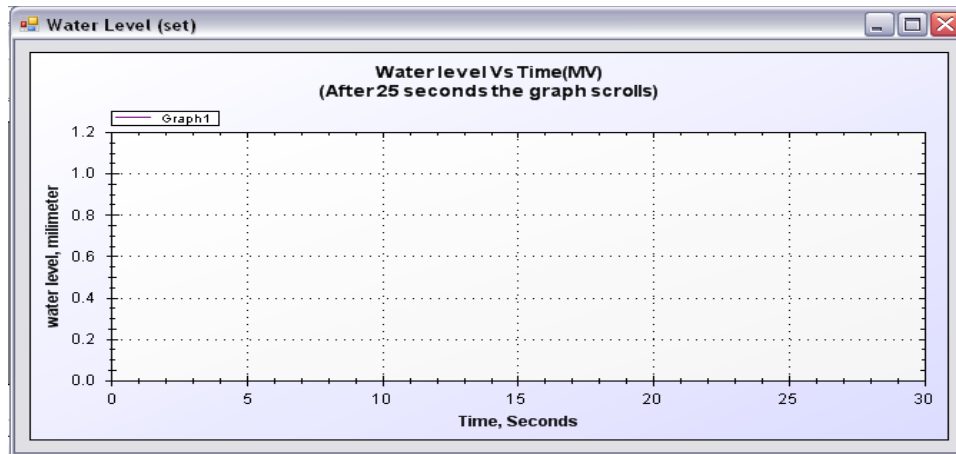


Figure3.20: Graph water level Vs time for MV

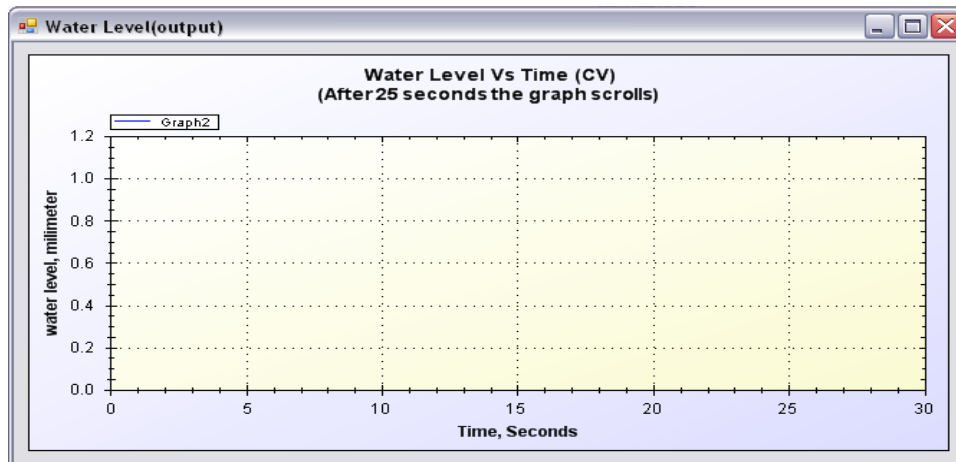


Figure 3.21: Graph of water level Vs time for CV

CHAPTER 4

RESULT AND ANALYSIS

4.1 Introduction

This chapter will discuss about the result and analysis for this project. In this chapter, it is separated into two parts which is focuses on:

1. Project result
2. Simulation Result

4.2 Project Result

A ladder diagram to control the operation of coupled tank system must be designed. The coupled tank is needed to continuously operate in pulse behavior with t second of delay time. Figure 4.1 show the ladder diagram to get the continuous data collection of the system. The flow of this ladder diagram is described as below:

1. In zero and first rung, when Start Push Button is pushed, the system will ON timer 1. When Stop Push Button is pushed, it will stop the entire process of the ladder diagram.
2. The yellow light ON when start button is pushed. The Red light ON when stop button is pushed, yellow light goes OFF.
3. In second rung, there is a primary timer which counts for 50 second for pump 1 been OFF. No water being pumped into tank.
4. In third rung, there is a second timer which counts for 90 second for pump 1 been ON. Water is being pumped into tank in this time period.
5. In fourth rung, the source word of I2011 will move hexadecimal number of voltage range (3V) to temporary address and kept in D100.
6. Lastly, the result that was saved in D100 will be moved to the output in address Q2001.

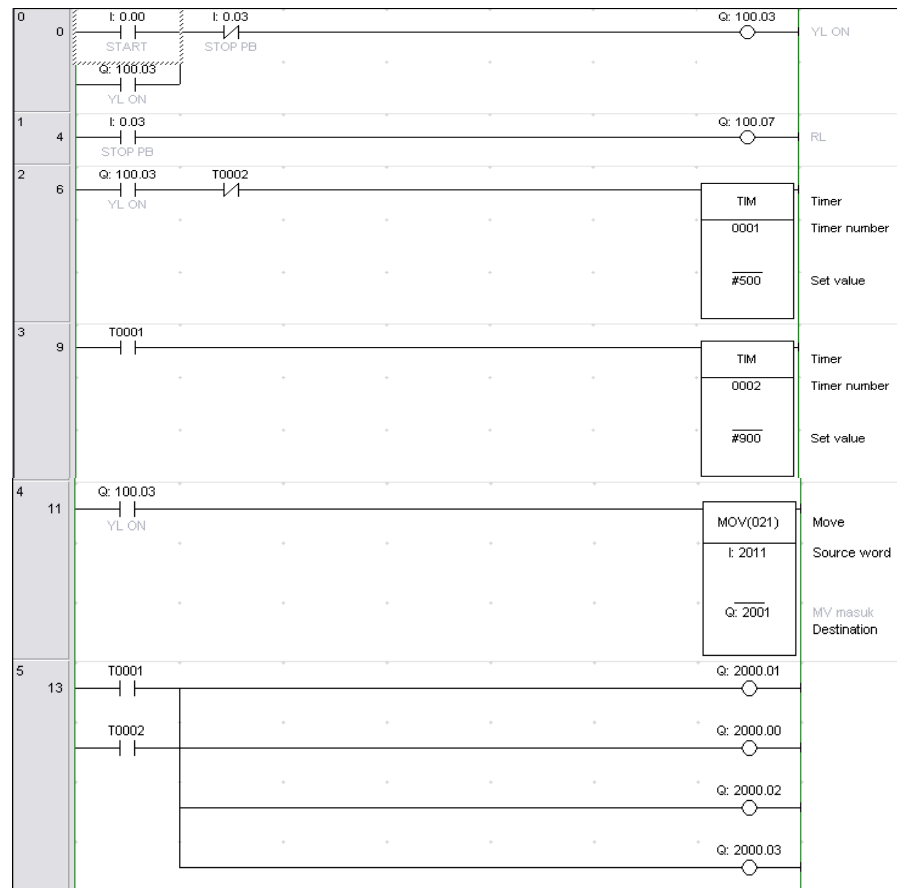


Figure 4.1: The Ladder Diagram of CX-Programmer

After the download is success, the start button is pushed. Analog voltage from analog output card with selected voltage value is connected to Pump 1 of Coupled Tank. For this example, 3 Volt is selected and pump 1 will pumping in water from reservoir entering tank 1. Because of the baffle effect between this twin tanks, water level in tank 2 is affected from water level pumped in tank 1.

This GUI shows how data collection is done. Data is sampled every second and the performance is shown in graph. We can export data collected to excel form. Figure 4.2 show GUI Visual Basic in debug form. The graph beside it is for an injected voltage while the bottom graph shows higher water level can go with injected voltage but in pump rate curve performance.

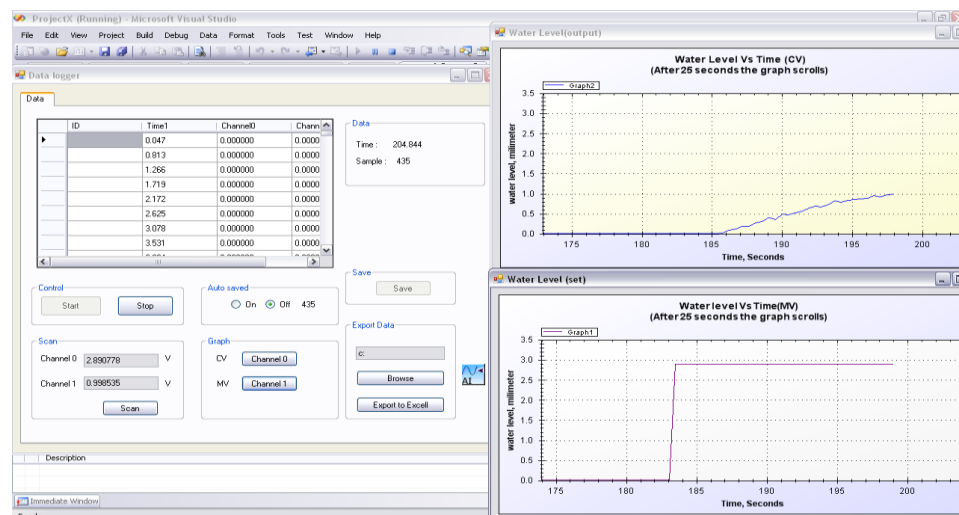


Figure 4.2: GUI visual basic during debug with Data Logger and graphs.

The water level entering tank 1 is controlled by speed of pump 1. The injected voltage, for example 3 V to pump 1 motor will take 30 second to reach approximately 180 mm level. When Pump 1 is off (0 V), the water leaving from tank 2 take 80 second to reach approximate 0 mm level because of the disturbance at clamp 2 is half-opened. The calibration of water level in coupled tank with voltage is been determined. Table 4.1 below show the result of calculated level value based on voltage span.

Table 4.1: Calculated level value based on voltage span

Span %	Voltage (V)	Level (mm)
0	0.00	0.00
20	1.00	60.0
40	2.00	120.0
60	3.00	180.0
80	4.00	240.0
100	5.00	300.0

4.3 Simulation Result

In simulation result, it is shown and proved that the used of Particle Swarm Optimization algorithm has been successfully applied in functions optimization. The optimized function (best fitness function) is needed in Parameter Identification. Particle Swarm Optimization is based on real time situation and always updating particles data on their positions and velocities. The transfer function of Coupled Tank can be obtained by manual calculation based on the equation given in manual. Simulation output with PSO algorithm applied has displayed the values for fitness transfer function.

Initialized value;

$$A_1 = 32 \text{ cm}^2$$

$$A_2 = 32 \text{ cm}^2$$

$$H_1 = 17 \text{ cm}$$

$$H_2 = 15 \text{ cm}$$

$$\alpha_1 = 10.78$$

$$\alpha_2 = 11.03$$

$$\alpha_3 = 11.0$$

Time constant of the tank dynamics is calculated using equation 4.1 and 4.2;

$$\boxed{T_1 = \frac{2A_1\sqrt{H_1}}{\alpha_1} \quad T_2 = \frac{2A_2\sqrt{H_2}}{\alpha_2}} \quad \text{-- (4.1 \& 4.2)}$$

$$\begin{aligned}
 T_1 &= \frac{2A_1\sqrt{H_1}}{\alpha_1} & T_2 &= \frac{2A_2\sqrt{H_2}}{\alpha_2} \\
 &= \frac{2 \times 32 \sqrt{15}}{10.8} & &= \frac{2 \times 32 \sqrt{15}}{11.03} \\
 &= \mathbf{22.95 \text{ sec}} & &= \mathbf{22.47 \text{ sec}}
 \end{aligned}$$

Constant of proportionality is calculated using equation 4.3, 4.4 and 4.5;

$$\boxed{\alpha_1 = \frac{2A_1\sqrt{H_1(0)}}{T_1} - \frac{2A_1\sqrt{H_1(t)}}{T_1}} \quad \text{----- (4.3)}$$

$$\begin{aligned}
 &= \frac{2A_1\sqrt{H_1(0)}}{T_1} - \frac{2A_1\sqrt{H_1(t)}}{T_1} \\
 &= \frac{2 \times 32 \sqrt{15}}{9.7} - \frac{2 \times 32 \sqrt{5}}{9.7} \\
 &= \mathbf{10.78}
 \end{aligned}$$

$$\boxed{\alpha_2 = \alpha_3 = \frac{2A_2\sqrt{H_2(0)}}{T_2} - \frac{2A_2\sqrt{H_2(t)}}{T_2}} \quad \text{----- (4.4)}$$

$$\begin{aligned}
 &= \frac{2A_2\sqrt{H_2(0)}}{T_2} - \frac{2A_2\sqrt{H_2(t)}}{T_2} \\
 &= \frac{2 \times 32 \sqrt{15}}{9.5} - \frac{2 \times 32 \sqrt{15}}{9.5} \\
 &= \mathbf{11.03}
 \end{aligned}$$

$$\boxed{K_1 = \frac{1}{\frac{\alpha_1}{2\sqrt{H_1}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad K_2 = \frac{1}{\frac{\alpha_2}{2\sqrt{H_2}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (4.5 \& 4.6)}$$

$$= \frac{1}{\frac{10.78}{2\sqrt{17}} + \frac{11.03}{2\sqrt{17-15}}} \quad = \frac{1}{\frac{11.03}{2\sqrt{15}} + \frac{11.03}{2\sqrt{17-15}}}$$

$$\mathbf{K_1 = 0.1921}$$

$$\mathbf{K_2 = 0.1878}$$

$$\boxed{K_{12} = \frac{\frac{\alpha_3}{2\sqrt{H_1 - H_2}}}{\frac{\alpha_1}{2\sqrt{H_1}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad K_{21} = \frac{\frac{\alpha_3}{2\sqrt{H_1 - H_2}}}{\frac{\alpha_2}{2\sqrt{H_2}} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}} \quad \text{----- (4.7 \& 4.8)}$$

$$= \frac{\frac{11.03}{2\sqrt{17-15}}}{\frac{10.78}{2\sqrt{17}} + \frac{11.03}{2\sqrt{17-15}}} \quad = \frac{\frac{11.03}{2\sqrt{17-15}}}{\frac{11.03}{2\sqrt{15}} + \frac{11.03}{2\sqrt{17-15}}}$$

$$\mathbf{K_{12} = 0.74589}$$

$$\mathbf{K_{21} = 0.7325}$$

Transfer function for Coupled Tank Liquid Level System

$$\boxed{\frac{h_2}{q_1} = \frac{k_1 k_{21}}{T_2 T_1 s^2 + (T_1 + T_2)s + (1 - k_{12} k_{21})} \quad \text{----- (4.9)}$$

$$= \frac{0.1921 \times 0.7325}{515.6865s^2 + (45.42)s + (0.4514)}$$

$$= \frac{0.1407}{515.6865s^2 + 45.42s + 0.4514}$$

4.1 Particle Swarm Optimization in MATLAB Simulink

The calculated transfer function from above is used in transfer function box in data.mdl in figure 4.3 below. This model must be ran first in order to export data to workspace (tdata.mdl).

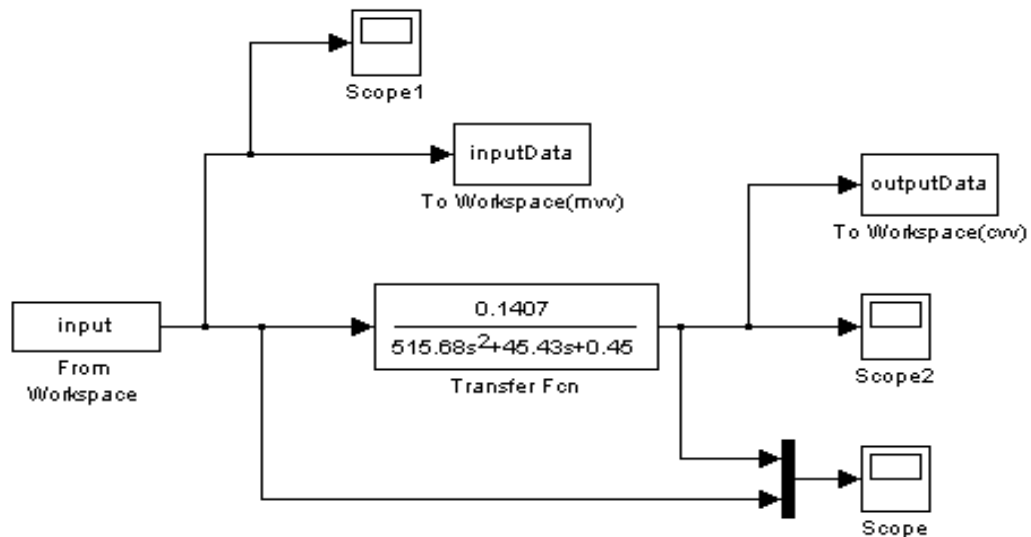


Figure 4.3: Data Model

For the input of mdata.mdl, data was taken from workspace which was from periodic signal collected in GUI. Figure 4.4 below show the periodic signal. Then data which was exported to Excel file was save in Text.txt file. Then the function is called in editor. Only one cycle of periodic signal was selected to train the mdata.mdl. Figure 4.5 below show one cycle of periodic signal. Coding to display periodic signal from GUI data collection is attached in appendix.

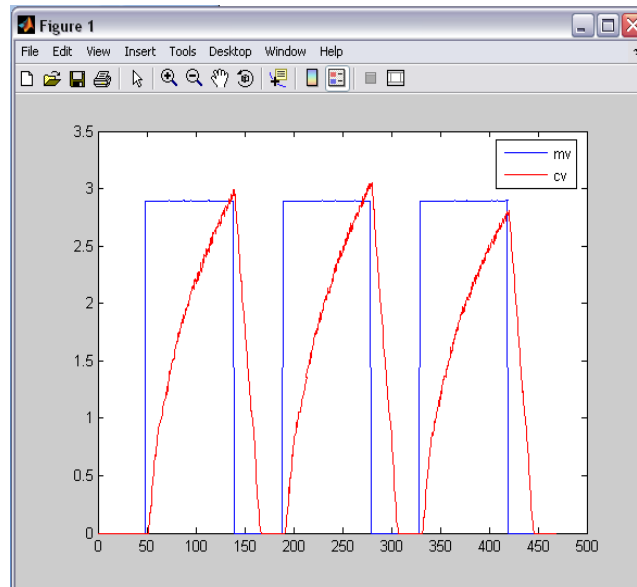


Figure 4.4: Periodic signal

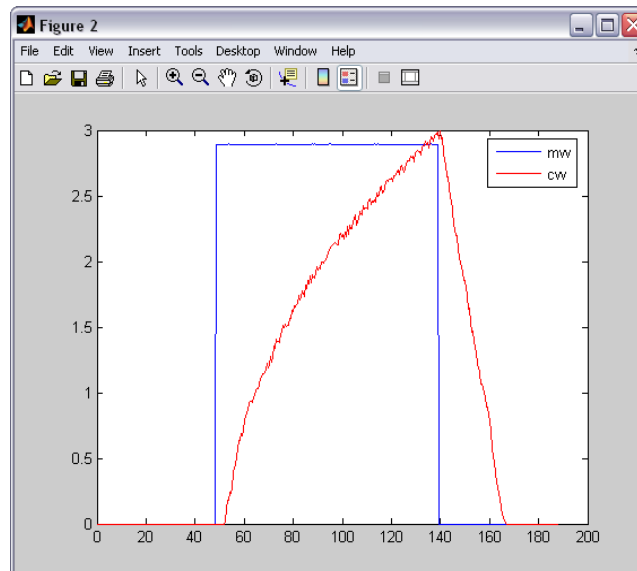


Figure 4.5: One cycle of periodic signal

When datamd1.mdl in debug mode, scope show both graph of input (purple line) and transfer function's output (yellow line). From this scope in figure 4.6 below, we can see there is big error occurred between training data from hardware which is coupled

tank and its own transfer function. Particle Swarm Optimization method is used to do this correction so that less square error between both signals is achieved.

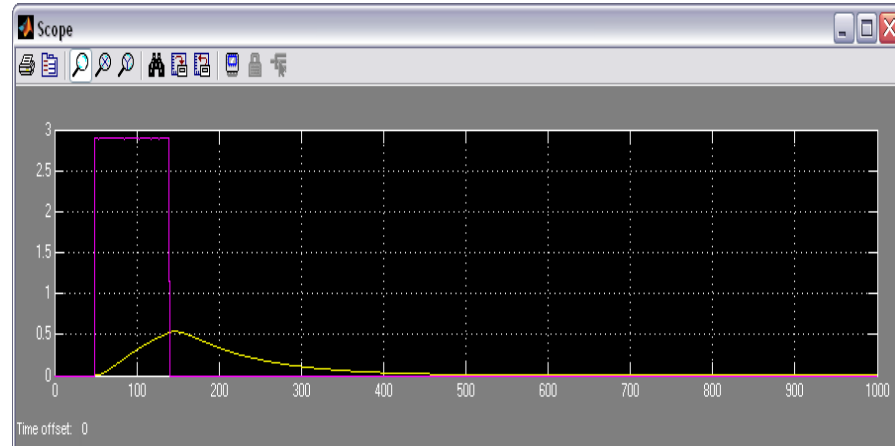


Figure 4.6: Training data and transfer function outputs

Equation 4.1 is the calculated transfer function of Coupled Tank from initialized values given in manual. This mathematical equation represented Coupled Tank. Equation 4.2 is the modeling equation of transfer function for Coupled Tank. The numerator consisted of aa while the denominator consisted of $bb.s^2$, $cc.s$ and dd . After Particle Swarm Optimization method is applied in simulation and successfully obtained less square error, values of aa , bb , cc and dd is observed.

$$G_S = \frac{0.1407}{515.6865s^2 + 45.42s + 0.4514} \quad \text{----- (4.10)}$$

$$G_S = \frac{aa}{bb.s^2 + cc.s + dd} \quad \text{----- (4.11)}$$

Figure 4.7 show the train.mdl, where data of transfer function from model is imported. When Particle Swarm Optimization is applied into this simulation, the output displayed the best fitness function values of aa , bb , cc and dd which were approaching

the original value of calculated transfer function after n number of iteration. The programming for Particle Swarm Optimization is placed at appendix.

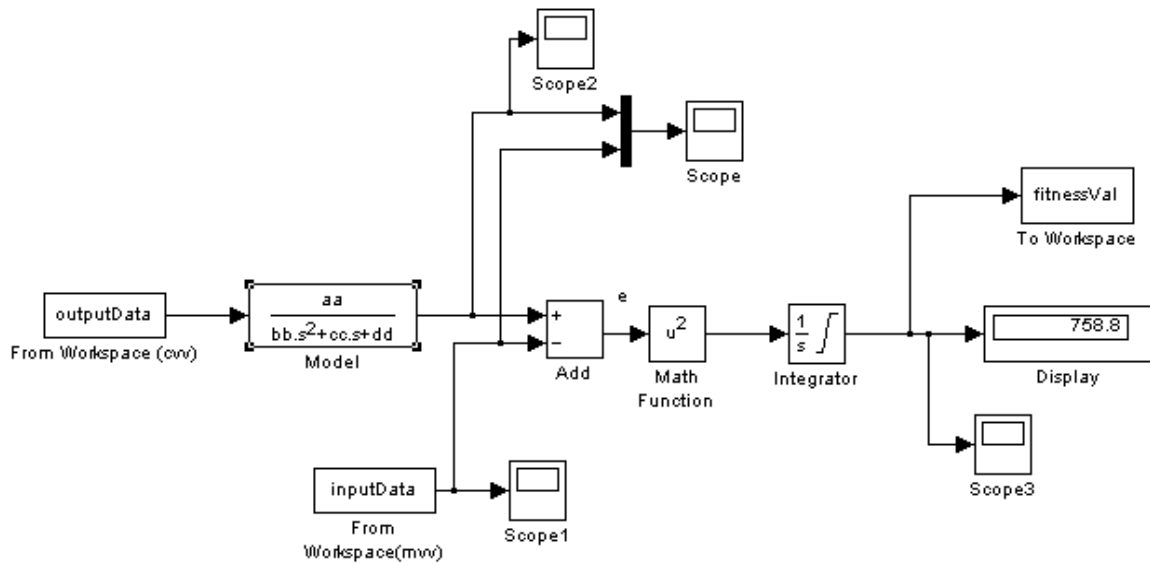


Figure 4.7: Train model

Block diagram in figure 4.8 simply explain step on comparing data from actual plant and data from model. Both data are compared until less square error is obtained. Less square error proved that actual plant transfer function is nearly approaching the value of transfer function for model.

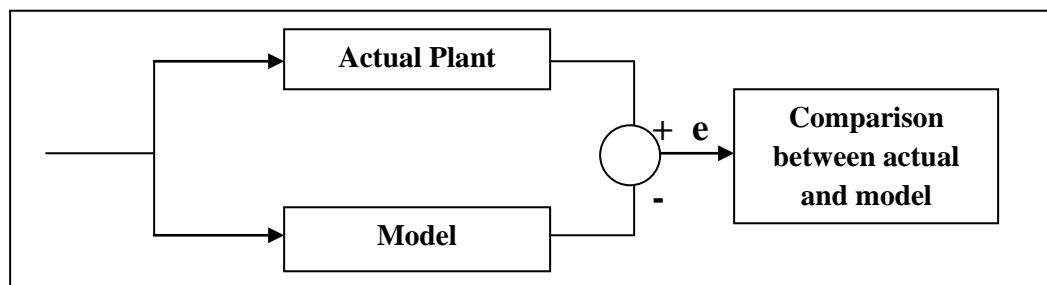


Figure 4.8: Block Diagram for comparison

Figure 4.9 below display the positions and velocities of a clock of particle while searching its pbest and gbest in order to get their fitness value. In figure 4.10, scope show the actual performance curve tried to follow the curve of model performance. The gap between those two lines determined error. The smaller gap between these lines indicates less error. Best Fitness Function indicates in less square error value.

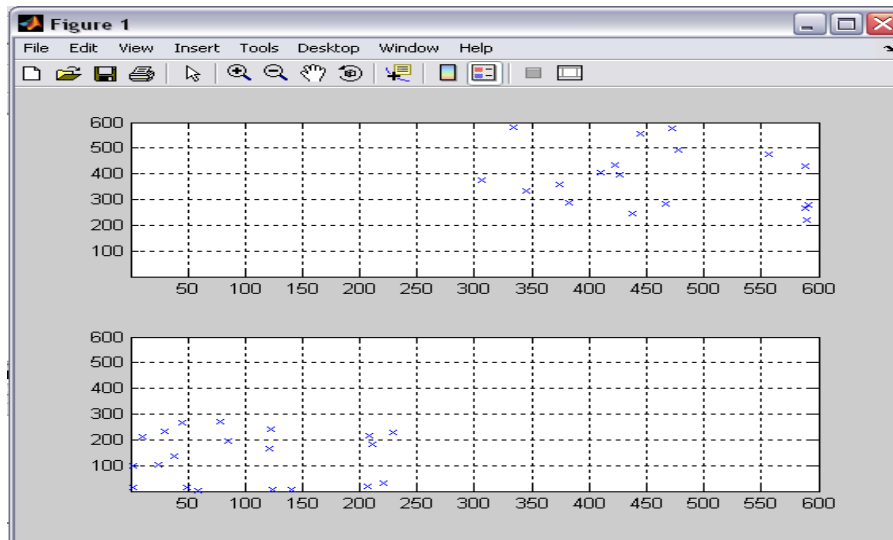


Figure 4.9: Particles in searching for best positions and velocities

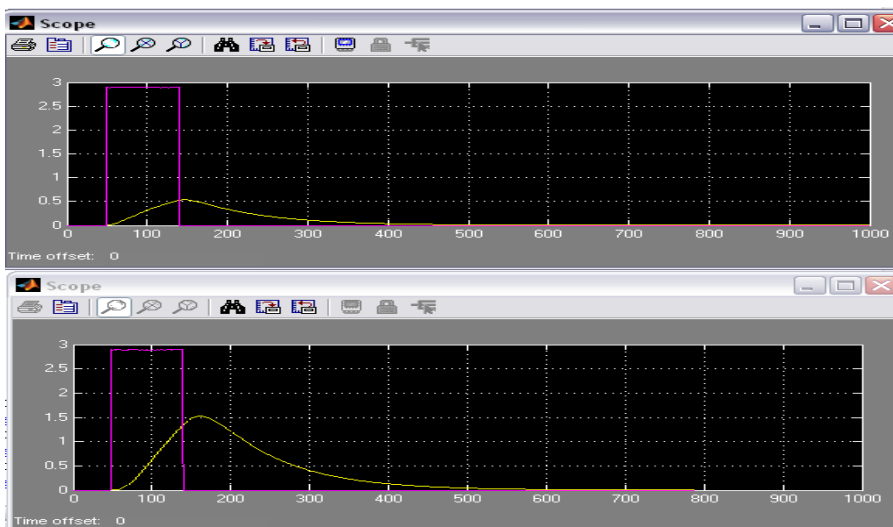


Figure 4.10: Scope at train model

To stop debug in Command Window, ctrl C is pressed. It will give value of actual performance straight after we enter aa, bb, cc, dd. Command Window also can display data_logger_training value, cvv, mvv, inputData and outputData when we enter this command.

Warning: Using a default value of 20 for maximum step size. The simulation step size will be limited to be less than this value.

> In data_final_train at 115

sim('trainMdl.mdl')

aa = 469.1421

bb = 101.8949

cc = 3.3568

dd =181.1167

$$Gs = \frac{aa}{bb.s^2 + cc.s + dd} \quad \text{----- (4.12)}$$

$$= \frac{469.1421}{101.8949s^2 + 3.3568s + 181.1167}$$

Error calculation in Excel

$$\begin{aligned} \text{Average Error} &= \frac{1}{n} \sum Channel1 - Channel0 && \text{----- (4.13)} \\ &= -0.34028 \end{aligned}$$

$$\begin{aligned} \text{Mean Square Error} &= \frac{1}{n} (Channel1 - Channel0)^2 && \text{----- (4.14)} \\ &= 1.572622 \end{aligned}$$

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Introduction

The overall objective of this project are to identify and obtain transfer function of coupled tank liquid level using Particle Swarm Optimization and to estimate the parameter coefficient of the transfer function utilizing an intelligent technique namely Particle Swarm Optimization.

Data collection from hardware is saved as data_logger_training.txt became the input for data model transfer function. The output from data.mdl is exported to train.mdl, where the output for both performances is evaluated for less square error. In this step, we observed two different signal at train.mdl scope after PSO is applied, where it consisted of train data (purple line) and actual data (yellow line).

From the result, it can be concluded that when Particle Swarm Optimization method is applied, the actual values of aa, bb, cc and dd is successfully observed for its fitness transfer function values. But for PSO to search the fitness function needs too long time for debug and resulting bigger error between actual and model performance.

This may be due to the wide range of particle's position from 0.1 to 600 and number of iterations up to 1000. When PSO is applied for a small range of particle's position and velocities, in the range 0 until 5, and dummy data for the transfer function (range in 1 to 10), it successfully gave values of aa, bb, cc and dd which are approaching the transfer function values of its model performance.

5.2 Recommendations

From the conclusion above, there are recommendations to improve this project for future application. The recommendations are divided into three categories which are recommendation in technique, hardware improvement and approach.

5.2.1 Recommendation in Technique

- For Coupled Tank Water Level, PWM signal can be injected for its input instead of analog voltage. Output measurement can be in PWM too instead of level.
- Calibration for Coupled Tank with span 0% - 100% started from 0V until 5V. It is more convenient to start the calibration from 1V to 5V because PLC can also read voltage in the range 1V-5V/4mA-20mA. This is because when we started with 0V, we cannot indicate the reading at 0V either it gave a reading value of zero or zero reading (system not functioning).

5.2.2 Recommendation in Hardware Improvement

For the hardware design, the PLC should be installed with the cooling fan. This is because the CJ1M PLC is very sensitive and easy to heat. So by including the fan, it makes it better not easy to heat hence avoid the CPU from hanging when the program is run.

5.2.3 Recommendation in Approach

Another approach in Intelligent Method that can be used for Parameter Identification is Artificial Neural Network. An artificial neural network (ANN) is an analysis paradigm that is a simple model of the brain and the back-propagation algorithm is the one of the most popular method to train the artificial neural network.

There are several papers reported using PSO to replace the back-propagation learning algorithm in ANN in the past several years. It showed PSO is a promising method to train ANN. It is faster and gets better results in most cases.

In applying PSO in ANN for parameter identification, there are not many parameters in PSO need to be adjusted. We only need to adjust the number of hidden layers and the range of the weights to get better results in different trials. The typical range of particles is 20 - 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

5.3 Costing and Commercialization

Parameter Identification can be based on Classical Method and Intelligent Method. This project focuses on intelligent method which implemented Particle Swarm Optimization. This project is done based on fact on how importance the Parameter Identification to real plant.

- Parameter Identification able to minimize damage to actual plant and saving maintenance cost in future.
- Particle Swarm Optimization and Parameter Identification have potential to be used in other type of operational plant in various industries and fields.

REFERENCES

- Augmented Innovation Sdn Bhd, (2000). *Coupled Tank Liquid Level: Computer-Controlled Laboratory Teaching Package*, Centre for Artificial Intelligent and Robotics (CAIRO): Manual Guide.
- J.Z You, C.D Johnson (1991). *Parameter Identification for MIMO System*. Dept ECE, University of Alabama in Huntsville,US: Ph.D Thesis.
- VSS Rameshkumar Junnuri, Desember (2004). System Identification Using Transfer Matrix, *IEEE Conference on Robotics, Automation and Mechatronics*. 1-3 Desember. Singapore: IEEE, pg 1124-1129
- J. Kennedy, RC. Eberhart, 1995. Particle Swarm Optimization. *Proceedings of the IEEE International Conference on neural networks*. Perth- Australia :IEEE, Vol. 4, pp. 1942 – 1948
- V.S Ghomsheh, M.A. Shoorehdeli, M.Teshnehlab, July (2007). Training ANFIS Structure with Modified PSO Algorithm, *15th Mediterranean Conference on Control & Automation*. Athens- Greece:IEEE, pg T14-003
- W.Tao, Z.Shun-yi, (2008). *Active Queue Management Based on Particle Swarm Optimization PID Algorithm*. Nanjing University of Posts and Telecommunications, Nanjing, China: Ph.D. Thesis.
- OMRON. *CJ Analog IO Units and Operation Manual*: Revised Manual September 2006
- U-tec. *Programmable Logic Controller Training*: Manual August 2004

APPENDIX A
PARTICLE SWARM OPTIMIZATION MATLAB CODING

```
clc, clear, close all

load data_logger_training.txt

tt = data_logger_training(:,1);
mvv = data_logger_training(:,2);
cvv = data_logger_training(:,3);

input=[tt,mvv];
output=[tt,cvv];

figure
plot(tt,mvv,'b-',tt,cvv,'r-');
legend('mvv','cvv')

% *****

sim('datamd1.mdl')

%initialize values of plant's parameters
aa=1; bb=1; cc=1; dd=1;

%Initialization of PSO parameters
wmax=0.9;
wmin=0.4;
itmax=1000; %Maximum iteration number
c1=1.4;
c2=1.4;
for iter=1:itmax
```

```

W(iter)=wmax-((wmax-wmin)/itmax)*iter;
end

% *****

% Initialization of positions of agents
% agents are initialized between -5,+5 randomly
a=0.1; % lower limit
b=600; % upper limit
N=20; % row
D=4; % column

x=a+(b-a)*rand(N,D,1); % initialization of x

% Initialization of velocities of agents
% Between -5 , +5, (which can also be started from zero)

m=0; % lower limit
n=0.1*b; % upper limit
V=m+(n-m)*rand(N,D,1);

% *****

% Function to be minimized. Scheaffer F6
for i=1:N;

    aa = x(i,1,1);
    bb = x(i,2,1);

```

```

cc = x(i,3,1);

dd = x(i,4,1);

sim('trainmdl.mdl')

pause(0.1)

F(i,1,1)=fitnessVal(length(fitnessVal)); % function to be minimized (fitness function)

end

% *****

[C,I]=min(abs(F(:,1,1)));

B(1,1,1)=C; % optimized fitness value

XX(1,1,1)=I; % optimized location of particles

for ii = 1:D; %storing best positions of particles into gbest
    gbest(1,ii,1) = x(I,ii,1);
end

% *****

%Matrix composed of gbest vector

for p=1:N
    for r=1:D
        G(p,r,1)=gbest(1,r,1);
    end
end

aa = G(1,1,1);

bb = G(1,2,1);

cc = G(1,3,1);

dd = G(1,4,1);

```



```

sim('trainMdl.mdl')

Fbest(1,1,1)=fitnessVal(length(fitnessVal));

for i=1:N;
    pbest(i,:,1)=x(i,:,1);
end

V(:,:,2)=W(1)*V(:,:,1)+c1*rand*(pbest(:,:,1)-x(:,:,1))+c2*rand*(G(:,:,1)-x(:,:,1));
x(:,:,2)= abs( x(:,:,1)+V(:,:,2) );

aa = gbest(1,1,1);
bb = gbest(1,2,1);
cc = gbest(1,3,1);
dd = gbest(1,4,1);

sim('trainMdl.mdl')

Fb(1,1,1)=fitnessVal(length(fitnessVal));

% *****

for j=2:itmax-1
    % Calculation of new positions

for i=1:N;
    aa = x(i,1,j);
    bb = x(i,2,j);
    cc = x(i,3,j);

```

```
dd = x(i,4,j);
sim('trainMdl.mdl')
F(i,1,j)=fitnessVal(length(fitnessVal));
end

[C,I]=min(abs(F(:,j)));

B(1,1,j)=C;

for ii=1:D
    gbest(1,ii,j) = x(I,ii,j);
end

aa = gbest(1,1,j)
bb = gbest(1,2,j)
cc = gbest(1,3,j)
dd = gbest(1,4,j)
sim('trainMdl.mdl')
Fb(1,1,j)=fitnessVal(length(fitnessVal));

[C,I]=min(Fb(1,1,:));

if Fb(1,1,j)<=C

    for ii=1:D
        gbest(1,ii,j)=gbest(1,ii,j);
```

```

    end
else

    for ii=1:D
        gbest(1,ii,j)=gbest(1,ii,I);
    end
end

%Matrix composed of gbest vector
for p=1:N
    for r=1:D
        G(p,r,j)=gbest(1,r,j);
    end
end

aa = G(1,1,j)
bb = G(1,2,j)
cc = G(1,3,j)
dd = G(1,4,j)

sim('trainMdl.mdl')
Fbest(1,1,j)=fitnessVal(length(fitnessVal));
for i=1:N;
    [C,I]=min(F(i,1,:));
    if F(i,1,j)<=C
        pbest(i,:,j)=x(i,:,j);
    else
        pbest(i,:,j)=x(i,:,I);
    end
end

```

```
end  
end  
  
V(:,j+1)=W(j)*V(:,j)+c1*rand*(pbest(:,j)-x(:,j))+c2*rand*(G(:,j)-x(:,j));  
x(:,j+1)=abs( x(:,j)+V(:,j+1) );  
  
subplot(211)  
plot(x(:,1,j),x(:,2,j),'x');  
grid on, xlim([0.1,600]), ylim([0.1,600])  
subplot(212)  
plot(x(:,3,j),x(:,4,j),'x');  
grid on, xlim([0.1,600]), ylim([0.1,600])  
pause (0.1)  
end  
  
aa = G(1,1,j-1)  
bb = G(1,2,j-1)  
cc = G(1,3,j-1)  
dd = G(1,4,j-1)  
sim('trainMdl.mdl')
```

APPENDIX B
GUI DATA COLLECTION CODING

```

Imports System.Data
Imports System.Data.OleDb
Public Class Form1
    Dim inc As Integer
    Dim MaxRows As Integer
    Dim con As New OleDb.OleDbConnection
    'Dim ds As New DataSet
    'Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    Dim tickStart As Integer = 0
    Dim volt As Double
    Dim calsample As Integer
    Dim maxSample As Integer
    Dim x As Integer



---



    Private da As OleDbDataAdapter
    Private ds As New DataSet()
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        con.ConnectionString = "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data
Source = Sample2.mdb"

        con.Open()

        sql = " select*from tblSample"
        da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "tblSample")

        con.Close()

        MaxRows = ds.Tables("tblSample").Rows.Count
        inc = -1

        Try
            ds.Reset()
            da.Fill(ds, "tblSample")
            DataGridView1.DataSource = ds.Tables(0)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        timer()

        AxAdvAI1.DataSampleRate = 40000
        AxAdvAI1.SetValueRange(1, CDb1(0), CDb1(5)) 'setting input
range for channel 1
        AxAdvAI1.SetValueRange(0, CDb1(0), CDb1(5)) 'setting input
range for channel 0
        Timer2.Enabled = False



---



    End Sub

    Private Sub timer()
        ' Sample at 50ms intervals
        Timer1.Interval = 1

        'Save the beginning time for reference

```

```

        'tickStart = Environment.TickCount
    End Sub

```

```

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim time As Double = (Environment.TickCount - tickStart) /
1000.0

        Dim voltage As Object
        'Dim strValue As String

        AxAdvAI1.ChannelScanStart = CInt(0)
        AxAdvAI1.ChannelScanCount = CInt(5)

        voltage = AxAdvAI1.AcquireScanChannelData(True)

        'txtChannel1.Text = AxAdvAI1.AcquireScanChannelData()
        'txtChannel2.Text = AxAdvAI1.ChannelScanStart + 2

        txtChannel0.Text = Format(voltage(0), "0.000000")
        txtChannel1.Text = Format(voltage(0), "0.000000")

        'If My.Forms.Form1.rbbtnSimulation.Checked Then
        ' 3 seconds per cycle
        ' Produce dummy data range 0-5V
        'volt = (Math.Sin(2 * Math.PI * time / 3.0)) * 2 + 3
        'volt = (Math.Sin(0.1 * Math.PI * time / 3.0)) * 2 + 3
        'Else
        volt = AxAdvAI1.DataAnalog
        'volt = AxAdvAI1.DataAnalog
        'End If

        ds.Tables("tblSample").Rows.Add()
        ds.Tables("tblSample").Rows(maxSample).Item(1) = time
        ds.Tables("tblSample").Rows(maxSample).Item(2) =
Format(voltage(0) / 0.25, "0.000000")
        ds.Tables("tblSample").Rows(maxSample).Item(3) =
Format(voltage(1), "0.000000")
        calsample = calsample + 1
        maxSample = maxSample + 1
        txtSampleRecord.Text = calsample
        txtTimeRecord.Text = time

        'Dim i As Object

```

```

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnStart.Click
        ds.Tables("tblSample").Rows.Clear()
        maxSample = 0
        Timer1.Enabled = True
        Timer1.Start()
        Timer2.Enabled = True

```

```

    btnStart.Enabled = False
    btnStop.Enabled = True
    btnSave.Enabled = False
    My.Forms.Graph1.Timer1.Enabled = True
    My.Forms.Graph1.Timer1.Start()
    My.Forms.Graph2.Timer1.Enabled = True
    My.Forms.Graph2.Timer1.Start()
    tickStart = Environment.TickCount

```

```
End Sub
```

```

    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click
        Timer1.Enabled = False
        Timer1.Stop()
        Timer2.Stop()
        btnStart.Enabled = True
        btnStop.Enabled = False
        btnSave.Enabled = True
        My.Forms.Graph1.Enabled = False
        My.Forms.Graph1.Timer1.Stop()
        My.Forms.Graph2.Enabled = False
        My.Forms.Graph2.Timer1.Stop()

```

```
End Sub
```

```

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        Dim command_builder As New OleDbCommandBuilder(da)
        da.Update(ds, "tblSample")
        MsgBox("Data Updated")

```

```
End Sub
```

```

    Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnBrowse.Click
        Dim objFolderDialog As New FolderBrowserDialog()
        '==== Pass object as Parameter and get Selected network folder
        txtPath.Text = GetNetworkFolders(objFolderDialog)

```

```
End Sub
```

```

    Public Shared Function GetNetworkFolders(ByVal oFolderBrowserDialog
    _As FolderBrowserDialog) As String

```

```

        If oFolderBrowserDialog.ShowDialog() = DialogResult.OK Then
            Return oFolderBrowserDialog.SelectedPath
        Else
            Return ""
        End If

```

```
End Function
```

```

    Private Sub Load_Excel_Details()
        'Extracting from database
        Dim filename As String

```

```
Try
```

```

        'ds.Reset()
        da.Fill(ds, "proto")
        If ds.Tables.Count < 0 Or ds.Tables(0).Rows.Count <= 0 Then
            Exit Sub

```



```

        End If
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    Dim Excel As Object = CreateObject("Excel.Application")
    If Excel Is Nothing Then
        MsgBox("It appears that Excel is not installed on this
machine. This operation requires MS Excel to be installed on this
machine.", MsgBoxStyle.Critical)
        Return
    End If

'Export to Excel process
Try
    With Excel
        .SheetsInNewWorkbook = 1
        .Workbooks.Add()
        .Worksheets(1).Select()

        Dim i As Integer = 1
        For col = 0 To ds.Tables(0).Columns.Count - 1
            .cells(1, i).value =
ds.Tables(0).Columns(col).ColumnName
            .cells(1, i).EntireRow.Font.Bold = True
            i += 1
        Next
        i = 2
        Dim k As Integer = 1
        For col = 0 To ds.Tables(0).Columns.Count - 1
            i = 2
            For row = 0 To ds.Tables(0).Rows.Count - 1
                .Cells(i, k).Value =
ds.Tables(0).Rows(row).ItemArray(col)
                i += 1
            Next
            k += 1
        Next
        filename = txtPath.Text & "\" & Format(Now(), "dd-MM-
yyyy_hh-mm-ss") & ".xls"
        .ActiveCell.Worksheet.SaveAs(filename)
    End With

System.Runtime.InteropServices.Marshal.ReleaseComObject(Excel)
    Excel = Nothing
    MsgBox("Data's are exported to Excel Succesfully in '" &
filename & "'", MsgBoxStyle.Information)

    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

' The excel is created and opened for insert value. We most
close this excel using this system
    Dim pro() As Process =
System.Diagnostics.Process.GetProcessesByName("EXCEL")
    For Each i As Process In pro
        i.Kill()
    
```

Next

End Sub

```
Private Sub btnExport_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExport.Click
    Load_Excel_Details()
End Sub
```

```
Private Sub btnBULK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    'Sets the device number of AdvAI1 to 0
    AxAdvAI1.DeviceNumber = 0

    AxAdvAI1.ChannelScanStart = 0
    AxAdvAI1.ChannelScanCount = 5

    'Sets DataSampleRate
    AxAdvAI1.DataSampleRate = 40000

    'Starts to acquire bulk data to a file
    AxAdvAI1.AcquireBulkDataToFile(4000, "c:\bulkdata.dat", 0,
True)
    AxAdvAI1.ConvertDigitalFileToAnalog("c:\bulkdata.dat",
"c:\bulkdata.txt", "%g" & Chr(0), Chr(1))
End Sub
```

```
Private Sub txtScan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnScan.Click
    Dim i As Object
    Dim voltage As Object
    Dim strValue As String

    AxAdvAI1.ChannelScanStart = CInt(0)
    AxAdvAI1.ChannelScanCount = CInt(5)

    voltage = AxAdvAI1.AcquireScanChannelData(True)

    'txtChannel1.Text = AxAdvAI1.AcquireScanChannelData()
    'txtChannel2.Text = AxAdvAI1.ChannelScanStart + 2

    txtChannel0.Text = Format(voltage(0), "0.000000")
    txtChannel1.Text = Format(voltage(0), "0.000000")
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Graph1.Visible = True
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Graph2.Visible = True
End Sub
```

```
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    Dim command_builder As New OleDbCommandBuilder(da)
    da.Update(ds, "tblSample")
    'MsgBox("Data Updated")

    x = x + 1
    txtTest.Text = x

End Sub
```

```
Private Sub radSimulate_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radSimulate.CheckedChanged
    'Timer2.Enabled = True
    'Timer2.Interval = 10000
    Timer2.Interval = 10

End Sub
```

```
Private Sub radReal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles radReal.CheckedChanged
    Timer2.Enabled = False

End Sub
End Class
```