UNIVERSITI TEKNIKAL MALAYSIA, MELAKA

FAKULTI KEJURUTERAAN PEMBUATAN

BMFG 4924

FINAL YEAR PROJECT II

SIMULATION OF INTELLIGENT PID TEMPERATURE CONTROLLER

Student's Name : ANG LING YUEN
Course : 4BMFA
Supervisor's Name : DR FAIRUL AZNI BIN JAFAR

SIMULATION OF INTELLIGENT PID TEMPERATURE CONTROLLER

ANG LING YUEN

B050910211

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2013

UNIVERSITI TEKNIKAL MALAYSIA, MELAKA

SIMULATION OF INTELLIGENT PID TEMPERATURE CONTROLLER

This report submitted in accordance with requirement of the Universiti Teknikal Malaysia Melaka (UTeM) for the Bachelor Degree of Manufacturing Engineering (Robotics and Automation) (Hons.)

by

ANG LING YUEN
B050910211
890105075533

FACULTY OF MANUFACTURING ENGINEERING
2013

DECLARATION

I hereby, declared this report entitled "Simulation of Intelligent PID Temperature Controller" is the results of my own research except as cited in references.

Signature	:	....
Author's Name	:	....
Date	:	....

APPROVAL

This report is submitted to the Faculty of Manufacturing Engineering of UTeM as a partial fulfillment of the requirements for the degree of Bachelor of Manufacturing Engineering (Robotics and Automation) (Hons.). The member of the supervisory committee is as follow:

……………………………………………

ABSTRAK

Objektif projek ini adalah untuk mensimulasikan pengawal suhu kabur jenis linear Mamdami jenis pengawal kabur dan pengawal suhu kabur jenis bukan linear Takegi-Sugeno dengan menggunakan MATLAB dan Simulink, serta untuk membandingkan prestasi antara kedua-dua pengawal. Dua kajian kes dicipta untuk menguji pengawal. Kajian kes I melibatkan pendandang air, di mana sistem tersebut dimodelkan dengan menggunakan Hukum Joule dan Hukum Termodinamik, dan kajian kes II melibatkan pengering rambut, di mana sistem dimodelkan dengan menggunakan kaedah ARX. Dalam kedua-dua kes, pengawal *Proportional-Integral-Derivative* (PID) telah ditala dan parameter PID kemudiannya digunakan untuk mendapat pengganda pengawal kabur. Keputusan simulasi menunjukkan bahawa pengawal kabur bukan linear terlajak lebih kecil dan mempunyai masa menetap yang lebih cepat berbanding dengan pengawal kabur linear dan pengawal PID, walaupun pengganda terbitan tambahan mungkin diperlukan untuk pengawal kabur bukan linear jika pengganda kamiran adalah cukup besar untuk menjejaskan kestabilan sistem.

ABSTRACT

The objectives of the project are to simulate linear Mamdami type fuzzy temperature controller and non-linear Takegi-Sugeno type fuzzy temperature controllers using MATLAB and Simulink, and to compare the performance between the two controllers. Two case studies were created to test the controllers. Case study I involved a water boiler, where the system is modelled using Joule's Law and Law of Thermodynamics, and case study II involved a hair dryer, where the system is modelled using ARX method. In both cases, a Proportional-Integral-Derivative (PID) controller was tuned and the PID parameters were then used to obtain the gain of the fuzzy controllers. Simulation results confirmed that non-linear fuzzy controller has smaller overshoot and faster settling time compared to the linear fuzzy controller and PID controller, although an extra derivative gain may be needed for the non-linear fuzzy controller if the integral term is huge enough to affect the stability of the system.

DEDICATION

To my beloved parents.

ACKNOWLEDGEMENT

I would like to thanks Dr Fairul Azni for the supervision of my final year project. His help and support plays a crucial role in doing the project, which I am forever grateful for.

I would also like to thanks Dr Omid Motlagh for providing valuable inside to fuzzy logic systems, which make understanding of the principle of fuzzy controller easier to grasp.

I would also like to thanks my fellow classmates for the supports and knowledges they provided me for all these years. I am very happy to be on the same class with you guys.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS, SYMBOLS AND NOMENCLATURES

FL	-	Fuzzy Logic
PID	-	Proportional, Integral and Derivative
Z-N	-	Zeigler-Nichols

CHAPTER 1

INTRODUCTION

In this chapter, the background and the objectives of the final year project would be discussed. Problem statement and scope of the project would also be stated. Lastly, a brief summary of the report structure is done at the end of the chapter.

1.1 Background

This project is inspired by the Type 1a V20 automatic butt welding machine created by AUGUST STRECKER GmbH & Co.KG. (STRECKER, 2012) The controller was created to control the annealing temperature during the welding process to archive the desired tensile strength. The purpose is to prevent the welded wires from becoming too brittle and break easily during processing such as drawing and galvanization. The company where the author interned before, RCI Wire Tech Sdn. Bhd., wished to procure the equipments and lamented of the fact that no local companies are capable of making controllers of the same kind, which inspired the author to attempt a project with the objective of creating such controller.

However, due to lack of experience causes the project to focus on more general application for practical reasons. To understand further on the process of designing such a controller, the project tries to simulate and develop a temperature controller to control the heating and cooling of water. The process is a well-known non-linear process, which often hard to be controlled using conventional linear PID algorithms. Fuzzy component is added to the original PID algorithm to solve this problem. The project is further encouraged by the lack of studies on rather general fuzzy controllers for well defined classes of systems such as heating, cooling and air conditioning (HVAC) system (Precup & Hellendoorn, 2011), which encouraged more studies to propose general fuzzy PID controller for HVAC purposes.

This project is aimed to create a fuzzy temperature controller to overcome the weakness of the PID controller. One of the method is to create a non-linear version of the fuzzy controller, so that the rate of change of the output will vary according to the amount of error and the rate of error itself, which in theory will give faster responses and lower overshoot. Simulation using MATLAB was done to validate the hypothesis.

1.2 Problem Statement

Is it possible to construct a non-linear fuzzy PID controller for temperature control with better performance than the PID controller, which is linear in nature? In theory, a non-linear controller should have smaller overshoot and faster transient response compared to its linear equivalent, and this project was done to prove this hypothesis.

1.3 Objectives

(a) To simulate linear Mamdami type fuzzy controller and non-linear Takegi-Sugeno type fuzzy controller using MATLAB and Simulink.

(b) To compare the performance between the fuzzy controllers and PID controller.

1.4 Scope

The main scope of this project is to test the performance of the control algorithms by comparing the transient response, steady-state error and stability of the controlled process via various analytical tools.

PID controller requires mathematical modelling of the system to simulate and design. For this project, the accuracy of the system modelling is not within the scope of the study; instead, the mathematical modelling was taken and modified from related literatures.

This project also limited its scope to only studying the temperature control. In theory, however, the control algorithms should works fine in other non-linear models, such as chemical plant control.

1.5 Report Structure

Chapter 1 is the introduction for the project. The background and problem statement of the project is described here. The objectives and scopes are also elaborated in this chapter.

Chapter 2 is the literature review, where the basic introduction of PID algorithm and its limitations is discussed. After that, various methods used to tune the PID parameters is elaborated and analysed, which included fuzzy logic.

Chapter 3 is the methodology. In this chapter, the method to simulate the controller is described. The chapter is divided according to case studies, where steps to simulate the model and the controllers in both cases are explained.

Chapter 4 is the result and discussion., where results of the MATLAB simulations are shown and discussed.

Chapter 5 is the conclusion, where the report is concluded based on the results and findings of Chapter 4. Recommendation and possible future projects based on the current project are also discussed.

CHAPTER 2

LITERATURE REVIEW

In this chapter, a review of PID controller is observed, where discussion regarding the limitations of PID controller is carry out. Next, brief introduction of FL and how it is used to solve the limitations of PID was done. A brief summary of the literature review can be found at the end of this chapter.

2.1 PID Controller

PID controller is a controller, which the ideal form of the control algorithm can be defined as:

$$u(t)=K_p e(t)+K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t) \qquad (1)$$

where e(t) is the function of error at time t, while τ is the variable of integration, which takes on value from time 0 to the time t. K_p, K_i and K_d is the tuning parameters for the proportional gain, integral gain and derivative gain respectively. Proportional gain adjusted the output based on the current known error, while integral gain does the same based on the cumulative errors of the process from the beginning to the current time, and derivative gain adjusted output by predicting the future errors based on the derivative of the error function.

However, in industry and in PID tuning studies, the standard form is used instead. The equation is expressed as:

$$u(t)=K_p[e(t)+\frac{1}{T_i}\int_0^t e(\tau)d\tau+T_d\frac{d}{dt}e(t)] \quad (2)$$

where T_i is the integral time, and T_d is the derivative time. In this format, the PID algorithm is more intuitive for the user, as the standard form shows that users can eliminate past and future errors by manipulating the desired times, T_i and T_d, for the respective errors, while the remaining errors can be scaled using K_p, allowing easier transient response design.

It is well known in literature that PID controller has a very stable and robust performance if the controlled system is a linear model. It is also easy to understand the algorithm, as modifying the PID control algorithm only involves changing the three tuning parameters to the appropriate values to gain the desired output. However, it is also well known that PID suffers in performance when it attempts to control a non-linear model, as PID is a linear and symmetrical equation and performance will vary if the system is non-linear (such as in HVAC systems). Manually tuning the parameters is also inefficient and time consuming process, as this method requires operator's experience and prone to human errors and misjudgement. Methods to heuristically tuning the PID parameters had been developed, and one of the popular one is the Ziegler-Nichols (Z-N) method. Ziegler & Nichols (1942) proposed a set of equations to tune various types of PID controller, which is summarised in Table 2.1. To tune the PID using Z-N method, first a user eliminates the integral and derivative part by setting K_i and K_d to zero, and raise the value of K_p until the output starts to oscillate. This ultimate gain, K_u, is noted and the rest of the parameters are tuned according to the desired oscillation period, P_u.

Table 2.1: Zeigler-Nichols method in PID Standard Form

Control type	K_p	T_i	T_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2/P_u$	-
PID	$0.60K_u$	$2.0/P_u$	$P_u/8$

Z-N method is a mathematically proven method, but it's not a fool-proof method. Additional modifications to the parameters are often required to archive the desired performance, which involves some amount of trials and errors.

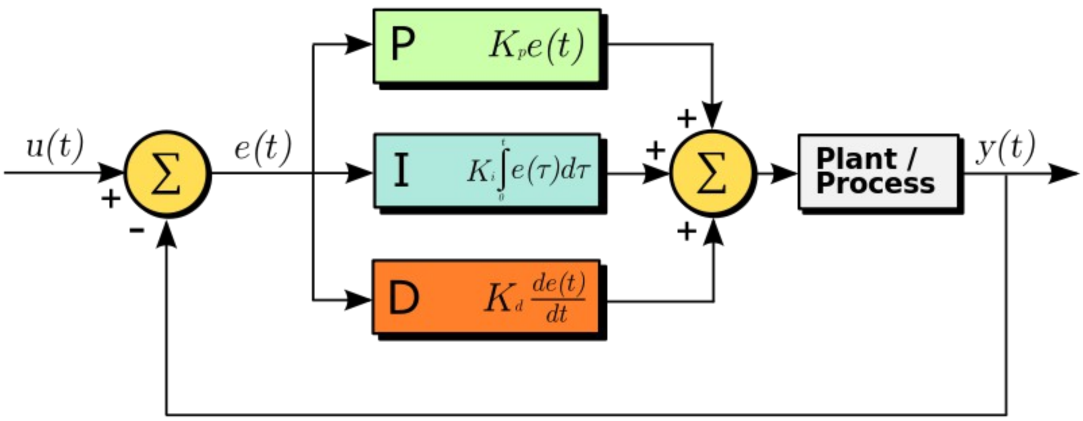

Figure 2.1: PID block diagram (Credit: Arturo Urquizo, CC-BY-SA-3.0)

2.2 Fuzzy Logic

Fuzzy logic (FL) is a type of many-valued logic and probabilistic logic, where the variables is expressed via a set, called fuzzy set, where its elements i.e. the variables have degree of membership and hence have true values between 0 and 1. The concept was first proposed by Zadeh (1965) and Gottwald (2010) simultaneously, which later proved to be useful to control variables with a degree of uncertainty, as shown by Mamdani's (1974) pioneering work, which is sometimes referred by other researchers as Mamdani type fuzzy controller.

FL controller usually contains four parts (Feng, 2006):

(a) Fuzzification interface converts crisp values of the controller input into a fuzzy value.

(b) Knowledge base is where the fuzzy rules and membership functions stored.

(c) Inference engine calculates the fuzzy output based on the information on the knowledge base and the fuzzy input.

(d) Defuzzification interface converts the inferred fuzzy output back to crisp outputs, which is used to control the system.

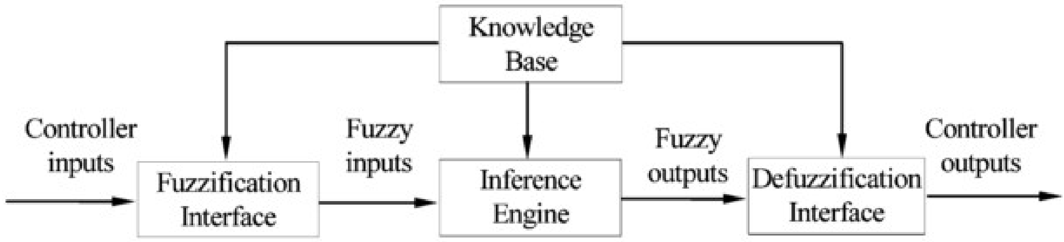

Figure 2.2: Block diagram of a typical FL control system (Feng, 2006).

Fuzzy logic can be classified into three types (Sugeno, 1999). First is the Mamdami type, or Type I as classified by Sugeno, uses the following format for fuzzy rules construction:

$$\begin{aligned}&\textit{If}\quad x_1 \textit{ is } G_1^i, x_2 \textit{ is } G_2^i, \ldots, x_m \textit{ is } G_m^i\\&\textit{Then}\quad y=H^i,\ \ i=0,1,2,3,\ldots\end{aligned}$$

where G^i and H^i are fuzzy sets.

Second type is the simpleton Mamdami type, or Type II as classified by Sugeno, which uses the following format for fuzzy rules construction:

$$\begin{aligned}&\textit{If}\quad x_1 \textit{ is } G_1^i, x_2 \textit{ is } G_2^i, \ldots, x_m \textit{ is } G_m^i\\&\textit{Then}\quad y=h^i,\ \ i=0,1,2,3,\ldots\end{aligned}$$

where h^i is a real number.

Lastly, Takagi-Sugeno (T-S) type, or Type III as classified by Sugeno, uses the following format for fuzzy rules construction:

$$\begin{aligned}&\textit{If}\quad x_1 \textit{ is } G_1^i, x_2 \textit{ is } G_2^i, \ldots, x_m \textit{ is } G_m^i\\&\textit{Then}\quad y=f^i(x_1, x_2, \ldots, x_m),\ \ i=0,1,2,3,\ldots\end{aligned}$$

where f^i is a function, usually expressed as a monomial equation:

$$f^i(x_1, x_2, \ldots, x_m) = a_0^i + a_1^i x_1^i + a_2^i x_2^i + \ldots + a_m^i x_m^i \quad (3)$$

There are a few ways to create PID controller using fuzzy logic system, and one of more popular one is gain scheduling, first created by Tomizuka & Isaka (1993). Gain scheduling uses fuzzy inference to calculate the PID tuning parameters and send the parameters back to the PIC controller. Another popular method is direct action, where instead of tuning parameters, the result of the fuzzy inference is the complete PID control signal, which can be used directly to control the system. More papers focused on direct control type, although Feng (2006) argued that gain scheduling would gain more support and adoption from the industry. Direct action method requires a complete rewrite of the control algorithms, which is costly to implement compared to the implementation method of gain scheduling (by adding an additional FL module to the original control algorithm). However, direct action is easier to design for non-linear models compared to gain scheduling for situations such as "zero control for zero error" and "maximum control for maximum errors". (Hu, Mann, & Gosine, 2001)

The difference between direct action and gain scheduling can be summarised using the following linguistic representation (Hu et al., 2001):

DA: If ("process error" is …)
then ("control action" is …)
GS: If ("process gain" is …)
then ("control gain" is …)

In other words, gain scheduling seeks to control the gain parameters of the PID algorithm, while direct action produces the complete PID algorithm to the plant instead. Figure 2.3 and Figure 2.4 show the difference of these two types of fuzzy PID controller in the form of block diagram.