COMPUTER AIDED EDUCATION FOR CONTROL SYSTEMS: PID CONTROLLER DESIGN OF CLOSED-LOOP SYSTEMS

AWANG MU'AWALUDIN BIN AHAT

May 2006

"I hereby declared that I had read through this report
and found that it has complete
the partial fulfillment for awarding the Degree of Bachelor in
Electrical Engineering (Power Industry)."

Signature : ....

Supervisor : Mdm Azrita Bte Alias

Date : 4 May 2006

COMPUTER AIDED EDUCATION FOR CONTROL SYSTEMS: PID CONTROLLER DESIGN OF CLOSED-LOOP SYSTEMS

AWANG MU'AWALUDIN BIN AHAT

This Report Is Submitted In Partial Fulfillment Of Requirements For The Degree of Bachelor In Electrical Engineering (Power Industry)

Fakulti Kejuruteraan Elektrik
Kolej Universiti Teknikal Kebangsaan Malaysia

May 2006

"Hereby, I declare that this report is a result of my own research idea except for works that have been cited clearly in the references."

Singnature : ....

Name : Awang Mu'awaludin Bin Ahat

Date : 4 May 2006

To my beloved parents

For their strength, motivation and

will power are incredible inspiration

to me each and everyday.

ACKNOWLEDGEMENTS

Alhamdulillah and a very thankful to the Faculty of Electrical Engineering, KUTKM for this great opportunity and co-operative during my completion of this final project. Then, I am very glad to express my deepest appreciation to my supervisor, Madam Azrita bte Alias where she had guided me, contribute ideas, theories and give best support for my project. Also for her sincere evaluation as well as attentive guidance to help me understand the methods and procedures involved in electrical engineering field. This appreciation also dedicated to those who involved in my research of the project such as in giving ideas, explaining the purposes and the importance of this project. Last but not least, I would like to extend my acknowledgement to all my family and friends for the supports, giving me full-hearted co-operation and informative references. I hope that this short, convenient and easy-to-use report would help students of electrical engineering or those who involved in control system engineering to get better knowledge as their understanding in electrical engineering field.

ABSTRACT

More than half of the industrial controllers in use today utilize PID or PID modified schemes. It is possible to apply various design techniques for determining parameters of the controller if a mathematical model of the plan can be derived. In terms of designing the closed-loop of PID controller, the controller tuning rules would be discussed as the process of selecting the parameters to meet its given of performance specifications. The Ziegler-Nichols tuning rules would be applied in designing the PID controllers. The understanding of the purposes of Ziegler-Nichols tuning rules would be discussed as generally to produce a better transient and steady-state response of the plant system. The method that can be applied due to these tuning rules can be seen in the simulation using appropriate software such as MATLAB. The contents of this report covers lots of information for understanding in the PID controller design based on Ziegler-Nichols tuning rules and the results of simulation of MATLAB shows the characteristics of the PID controller closed-loop design. At the end of this project, the graphical user interface-GUI would provide the calculation and simulation for determining parameters of the PID controller which will meet the transient and steady-state specifications of the closed-loop systems using Ziegler-Nichols tuning rules.

ABSTRAK

Sebahagian besar daripada industri pengawal pada hari ini menggunakan pengawal PID yang telah diubahsuai mengikut skim. Adalah menjadi mudah sekiranya pelbagai kaedah mereka bagi menentukan parameter pengawal sekiranya sesuatu model matematik dapat diperolehi. Dalam usaha mereka gelung tertutup pengawal PID, hukum aturan pengawal dibincangkan sebagai proses memilih parameter bagi menemukan spesifikasi persembahannya. Hukum aturan Ziegler-Nichols diaplikasikan bagi mereka pengawal PID. Pemahaman kepada tujuan hukum aturan Ziegler-Nichols dibincangkan secara umum bagi menghasilkan tindakbalas sementara dan stabil yang lebih baik kepada sistem pelan. Kaedah yang boleh diaplikasikan berkaitan dengan hukum aturan ini dapat dilihat melalui simulasi yang sesuai menggunakan perisian MATLAB. Kandungan laporan meliputi banyak maklumat sebagai pemahaman dalam mereka pengawal PID berdasarkan hukum aturan Ziegler-Nichols dan keputusan simulasi MATLAB menunjukkan ciri-ciri mereka gelung tertutup pengawal PID. Di akhir projek ini, grafik pengguna antara muka-GUI akan menyediakan pengiraan dan simulasi bagi menentukan parameter pengawal PID yang menemukan ciri-ciri tindakbalas sementara dan stabil bagi sistem gelung tertutup menggunakan hukum aturan Ziegler-Nichols.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PID Controller

A Proportional-Integral-Derivative controller is a standard component in industrial control applications. It measures an "output" of a process and controls an "input", with a goal of maintaining the output at a target value, which is called the "set point". *[6]*

In this project, the designing of the PID controller is based on the Ziegler-Nichols tuning rules to get the gain of parameter in control system. The gains are considered as derivative gain Kd, proportional gain Kp and integral gain Ki. For this method, the mathematical model of the control system is unknown. So, the Ziegler-Nichols tuning rules are proposed to get the mathematical model of the control system to meet the transient and steady-state specifications of the closed-loop system.

There are several methods for tuning a PID loop. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response speed of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters. *[1]*

If the system must remain online, one tuning method is to first set the I and D values to zero. Increase the P until the output of the loop oscillates. Then increase I until oscillation stops. Finally, increase D until the loop is acceptably quick to reach its reference. The best PID loop tuning usually overshoots slightly to reach the set-point more quickly, however some systems cannot accept overshoot. *[4]*

Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral/reset, derivative/rate) to the optimum values for the desired control response. The optimum behavior on a process change or set point change varies depending on the application. Some processes must not allow an overshoot of the process variable from the set point. Other processes must minimize the energy expended in reaching a new set point.

Generally stability of response is required and the process must not oscillate for any combination of process conditions and set points. Tuning of loops is made more complicated by the response time of the process; it may take minutes or hours for a set point change to produce a stable effect.

The outcome of this project is to implement a user-friendly software using GUI as the understanding for PID closed-loop system. The students of control system engineering will get a lot of benefit from this project as they can learn the basic application of Ziegler-Nichols tuning rules to design the PID closed-loop system. They also can design the high stability control system using this method.

1.2 Ziegler-Nichols Tuning Rules

Ziegler and Nichols have developed PID tuning methods back in the early fourties based on open loop tests (less known than for example the Cohen-Coon formulas) and also based on a closed loop test, which is maybe their most widely known achievement.

The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Ziegler-Nichols suggested rules for tuning PID controllers (meaning to set values of Kp, Ti and Td) based on experimental step responses or based on the value of Kp that results in marginal stability when only proportional control action is used. Ziegler-Nichols rules, which are briefly presented in the following, are useful when mathematical models of the plants are not known.

These rules can, of course be applied to the design of systems with known mathematical models. Such rules suggest a set of values of Kp, Ti and Td that will give a stable operation of the system. *[7]*

However, the resulting system may exhibit a large maximum overshoot in the step response which is unacceptable. In such case we need series of fine tunings until an acceptable result is obtained. In fact, the Ziegler-Nichols tuning rules give an educated guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for Kp, Ti and Td in a single shoot.

There are two methods called Ziegler-Nichols tuning rules. In the first method, we obtain experimentally the response of the plant to a unit-step input. If the plant involves neither integrator (s) nor dominant complex-conjugate poles, then such a unit – step response curve may look S-shaped curve. Such step-response curves may be generated experimentally or from a dynamic simulation of the plant.

The S-shaped curve may be characterized by two constants, delay time, d and time constant, T. The delay time, d and time constant, T are determined by drawing a tangent line at the inflection point of the S-shaped curve and determining the intersections of the tangent line with the time axis and line K. The example of open-loop response due to the Ziegler-Nichols reaction curve is shown in the figure 1.2(a). The second method will be applied if the output does not exhibit S-shaped curve. *[5]*

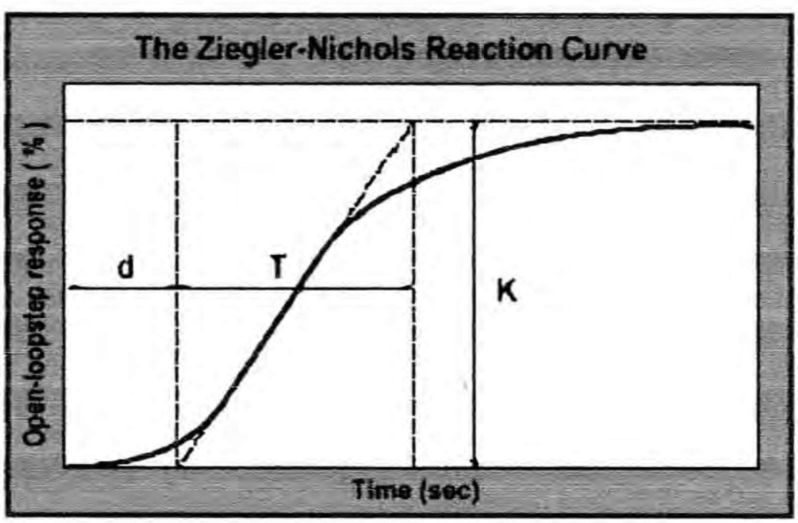

Figure 1.2(a): Open-loop response due to the Ziegler-Nichols reaction curve.

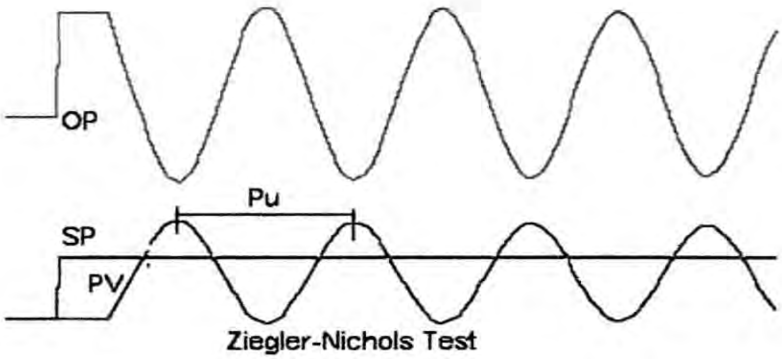

Figure 1.2(b): Shows the Ziegler-Nichols test output. Note: Pu=Tu

Table 1.2: Shows the modified Ziegler-Nichols PID tuning.

	Kp	Ti	Td
P	0.33Ku		
PI	0.33Ku	2Tu	
PID	0.20Ku	0.80Tu	0.20Tu

1.3 SIMULINK In MATLAB Software

Simulink is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.

For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this, it can be drawing the models just as with pencil and paper (or as most textbooks depict them). This is a far cry from previous simulation packages that require formulate differential equations and difference equations in a language or program. *[6]*

Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. It also can customize and create the blocks. For information on creating the blocks, see the separate Writing S-Functions guide. Models are hierarchical, so models can be built using both top-down and bottom-up approaches.

It can view the system at a high level, and then double-click blocks to go down through the levels to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact.

After define a model, simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations.

Using scopes and other display blocks, the simulation results can be seen while the simulation is running. In addition, the parameters can be changed immediately to see what happens, for "what if" exploration.

The simulation results can be put in the MATLAB workspace for post processing and visualization. Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes. And because MATLAB and Simulink are integrated, it can also simulate, analyze, and revise the models in either environment at any point. *[2]*

1.4 Graphical User Interface

A graphical user interface (GUI) is a user interface built with graphical objects -- the components of the GUI -- such as buttons, text fields, sliders, and menus. If the GUI is designed well-designed, it should be intuitively obvious to the user how its components function. For example, when you move a slider, value changes; when an OK button is clicked, the settings are applied and the dialog box is closed.

Fortunately, most computer users are already familiar with GUIs and know how to use standard GUI components. By providing an interface between the user and the application's underlying code, GUIs enable the user to operate the application without knowing the commands would be required by a command line interface.

For this reason, applications that provide GUIs are easier to learn and use than those that are run from the command line. The sections that follow describe how to create GUIs with GUIDE. This includes laying out the components, programming them to do specific things in response to user actions, and saving and opening the GUI.

1.5 Project Objectives

The objectives of this project are:

a) To implement a user-friendly software using GUI as the understanding for PID closed-loop system.

b) To produce the steady-state response and transient response of the plant system using Ziegler-Nichols tuning rules.

c) To determine the parameters of the PID controller of a plant that will meet the transient and steady-state specifications of the closed-loop system.

1.6 Project Scope

The scope that will be focused on this project is the implementation of a closed-loop system of PID controller based on Ziegler-Nichols tuning rules. Through this method, the specification of the gain parameters such as Ki, Kp and Kd in the PID controller which could be obtained in order to produce responses to the transfer function of the plant system. Besides that, the project would prove the effectiveness of the Ziegler-Nichols tuning rules to improve the steady state error and transient response.

1.7 Problem Statements

This project would act as an alternative in facilitating PID closed-loop system design. The usefulness of PID controls lies in their general applicability to most control systems. In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used.

It is well-known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control. The Ziegler-Nichols proposed rules for determining values based on the transient response characteristics of a given plant.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction of MATLAB

In this introduction, the information about MATLAB software will be discussed as it covers the start-up with MATLAB, example of MATLAB programming, calculation of parameters of controllers and many others which focused with the consideration of understanding the designing of the PID closed-loop system using this software.

2.1.1 MATLAB Software

The basics of MATLAB are the most common classical control design techniques (PID, root locus, and frequency response), as well as some modern (state-space) control design. In MATLAB, an m-file can be used as the development of software. An m-file, or script file, is a simple text file where it can place MATLAB commands. When the file is run, MATLAB reads the commands and executes them exactly as each command sequentially at the MATLAB prompt. All m-file names must end with the extension '.m' (e.g. plot.m).

If a new m-file creates with the same name as an existing m-file, MATLAB will choose the one which appears first in the path order (help path for more information). To make life easier, a name for m-file which doesn't already exist will be choose. To see if a filename.m exists, types help filename at the MATLAB prompt. *[2]*

For simple problems, entering the requests at the MATLAB prompt is fast and efficient. However, as the number of commands increases or trial and error is done by changing certain variables or values, typing the commands over and over at the MATLAB prompt becomes tedious. M-files will be helpful and almost necessary in these cases.