Development of Auto Tuning PID Controller Using Graphical User Interface (GUI)

T. L. T. Mohamed, Nor Fazli Adull Manan, Zulkifli Mohamed, and Mohd Afzan Mohd Anuar

***Abstract*—Auto tuning PID controller is designed using MATLAB, Simulink and Graphical User Interface (GUI) as well as the optimization of PID controller without excessive mathematic calculations. Ziegler-Nichols second method is used for designing purpose of the auto tuning PID controller. Some drawbacks have been discovered using this technique, where in some cases analysis and design problems can be tedious, since the solution involves trial and error. There are other methods utilized to solve this problem by given access to computers or programmable calculators configured with appropriate MATLAB and GUI software. The true power and advantage of the program developed lies in its ability to represent both compensated and uncompensated relationship and also ability to learn these relationships directly from the data being modeled. The results will then being validated by using manual calculation, MATLAB and Simulink. The value of *Kp, Ti* and *Td* in Ziegler-Nichols formula will be calculated using manual calculation while step response graph for each cases will be solved solve using Simulink.**

***Index Terms*—Auto tuning, PID, graphical user interface (GUI), ziegler-nichols second method, simulink, proportional controller (Kp), Integral controller (Ti), derivative controller (Td).**

I. Introduction

A control system consists of subsystems and processes (or plants) assembled for the purpose of obtaining a desired output with desired performance, given a specified input. Two major measures of performance are the transient response and the steady-state error [3]. Furthermore, a system can be tuned using specific type of controllers in order to increase the stability of the system with desired requirements. It is known as PID control technique that provides controllers such as proportional, integral and derivative types of controllers [9].

PID control technique has been one of the most utilized control techniques in system like pneumatic and electronic systems and has been proved its wide range of applications since it is first introduced [5]. The two most popular PID controller techniques were the step reaction curve experiment, and a closed-loop “cycling” experiment under proportional control around the nominal operating point. Some of simple PID setting formulae are such as the Chien Hrones–Reswick formula, Cohen–Coon formula, Ziegler–Nichols tuning, Wang–Juang–Chan formula and Zhuang–Atherton optimum PID [4]. One of them is called as Ziegler–Nichols tuning formula which is approaches to identify the equivalent first-order plus dead time model, which is essential in some of the PID controller design algorithms

G. Ziegler and N. B. Nichols (1948) recognized that the step responses of a large number of process control systems exhibit a process reaction curve [4]. This curve can be generated from either experimental data or dynamic simulation of a plant [1]. The S- shape of the curve is characteristic of many high-order systems [10], and such that plant transfer functions may be approximated by which is simply a first order system plus a time delay of *td* seconds. The constants can be determined from unit step response of the process. Ziegler and Nichols gave two methods for determine the constants for a controller. Quarter decay ratio method, the choice of controller parameters is based on a decay ratio of approximately 0.25. This means that a dominant transient decays to quarter of its value after one period of oscillation. A quarter decay corresponds to $\zeta = 0.21$ and is a good compromise between quick response and adequate stability margins [8].

II. Ziegler-Nichols Second Method

Ziegler-Nichols second method is an ultimate sensitivity method; the criteria for adjusting the parameters are based on evaluating the system at the limit of stability rather than on taking a step response. The proportional gain K is increased until continuous oscillation is achieved (marginally stable). The period of oscillation P_{cr} should be measured when the amplitude of oscillation is quite small.

If frequency response experiment can be performed, the crossover frequency ω_{cr} and the critical gain K_{cr} can be obtained from the Nyquist plot and root locus plot. Let $P_{cr} = 2\pi/\omega_{cr}$. Ziegler and Nichols suggested that it can be found by putting the controller in the proportional mode and increasing the gain until an oscillation takes place. The point is then obtained from measurement of the gain and the oscillation frequency [6]. This result, however, is based on linear theory, and although the technique has been used in practice, it does have major problems [4].

In the second method, the value K_p is set as the variable, T_i is infinity and the value of Td is equal to zero. Using the proportional control action only, the value of K_p is increased from zero to a critical value. This analysis will depend to the step response graph; the value of K_{cr} is determined when step response were produced marginally stable graph. And the oscillations of the graph will gain the corresponding period P_{cr}. The frequency of the sustained oscillation ω_{cr}, where $2\pi/\varpi_{cr} = P_{cr}$ Ziegler and Nichols sugested[6] that we set the values of the parameters K_p, T_i and Td according to the

Manuscript received November 14, 2012; revised December 23, 2012.

The authors are with the Faculty of Mechanical Engineering, Universiti Teknologi MARA, Selangor, Malaysia (e-mail: Luqmantm@proton.com, norfazli@salam.uitm.edu.my, zulkifli127@salam.uitm.edu.my,: m_afzan82@yahoo.co.uk).

formula shown in Table I.

TABLE I: ZIEGLER-NICHOLS PARAMETER TUNING [1]

Type of Controller	K_p	K_i	K_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	(1/1.2) P_{cr}	0
PID	$0.6K_{cr}$	0.5Pcr	$0.125/P_{cr}$

Notice that the PID controller tuned by the second method of Ziegler-Nichols gives:

For general equation:

$$Gc(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (1)$$

For P – Controller:

$$Gc(s) = K_p \quad (2)$$

For PI – Controller:

$$Gc(s) = \frac{K_p.T_i s + K_p}{T_i s} \quad (3)$$

For PID – Controller:

$$Gc(s) = \frac{K_p.T_i.T_d s^2 + K_p.T_i s + K_p}{T_i s} \quad (4)$$

where;

$Gc(s)$ is controller equation.

K_p is proportional controller.

T_i is integral controller.

T_d is derivative controller.

III. GUI DESIGN

GUI icons selected for both windows are the same, where they only differ in terms of the decoration and amount of GUI icons required. Types of GUI icons selected are the command radio button, check button, check box, popup menu, list box, push button and toggle button for discrete input and the other two command axes and static text for display output [7].

In designing of this auto tuning programming, it involves one graphical user interface (GUI) but divided into two sections, root-locus analyzes section and Ziegler-Nichols tuning section. This idea was created base on the limitation in application of Ziegler-Nichols in tuning the uncompensated system. In designing the controller, the branches of the root-locus plot should cross the $j\omega$ axis. The operation is started when user enters the transfer function. The transfer function is separated into two edit text of numerator and denominator. Then user need to click the apply push button to generate the root-locus plot. If the user clicks cancel push button, the numerator text, denominator text and root-locus plot will reset. The purpose of root-locus analyzes section is to illustrate the root-locus plot and from the illustrated plot, it can define whether the branches of the root-locus intercepted the $j\omega$ axis. There are analyzes push button in the root-locus section which purpose is to recognize whether the branches crossing or not.

Ziegler-Nichols tuning section responsible in tuning the controller based on the origin of transfer function. In this section, it will display the parameters required in the controller designed. Value of *Kcr, ωcr, Kp, Ti* and *Td* will be displayed at PID parameter. There were several push buttons in this section to perform several tasks. To operate this section, users need to choose the type of controller at controller popup menu then click on design controller button to display the parameters. For illustrate the step response of the response, users need to click uncompensated button. In this section, there are two different step responses one is open-loop response and the other is closed-loop response and can be decided at the response type popup menu. In order to display the compensated system, user need to click display compensated push button. The displayed step response was according to type had being chose. In order to display other type controller, it can be achieved by choosing other type of controller and click design controller then click the display compensated to display the step response plot. Other push buttons that consist in Ziegler-Nichols tuning were resume button and hold button. Resume push button responsible in to reset the step response plot and the parameter display. The function of hold push button is to hold the step response graph. The legend show the type of controller based on the color of the response plot. Time scale is created to limit the step response plot.

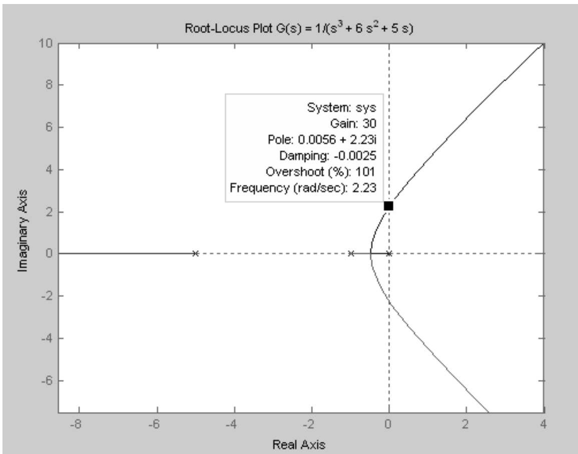

Fig. 1. Root locus plot using MATLAB command

IV. VALIDATION

The validation process is performed using MATLAB command, manual calculation and Simulink. Using MATLAB command, root loci was being plot, and from the plot picks values of the crossing points of the root-locus branches with the $j\omega$ axis. At this point the value of critical gain *Kcr* and critical frequency were achieved. Next step is manual calculations based on Ziegler-Nichols second method to interpret the parameters *Kp, Ti,* and *Td* and obtain the transfer function of controller using MATLAB command. Simulink will be the final step of validation where at this stage Simulink responsible in processing the data to get the output of the system after introduced controller in the system.

The output is based on step responses and these results were compared with the Auto Tuning programming to ensure the programming developed was valid.

As for the first step in validate, considered 0 – zero with 3 – poles. Consider a transfer function as below:

$$Gc(s) = \frac{1}{s^3 + 6s^2 + 5s} \tag{5}$$

Plot root-locus based on transfer function case 1 as shown in Fig. 1:

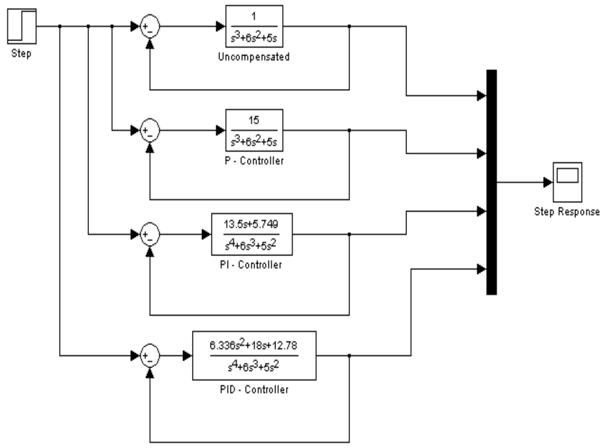

Fig. 2. Simulink diagram

From the root-locus plot, pick value crossing points of the root-locus branches with the *jω* axis. At this point the value of critical gain Kcr = 30 and critical frequency *ωcr* = 2.23 rad/s.

Manual calculation is done based on Ziegler-Nichols second method to interpreted the parameters *Kp, Ti,* and *Td* and obtain the transfer function of controller system.

$$K_{cr} = 30 \tag{6}$$

$$P_{cr} = \frac{2\pi}{\omega} = \frac{2\pi}{2.23} = 2.817 \tag{7}$$

P – Controller

$$K_p = 0.5K_{cr} = 15 \tag{8}$$

$$T_i = \infty \tag{9}$$

$$T_d = 0 \tag{10}$$

$$Gc(s) = 15 \tag{11}$$

PI – Controller

$$K_p = 0.45K_{cr} = 13.5 \tag{12}$$

$$T_i = \frac{1}{1.2}P_{cr} = 2.348 \tag{13}$$

$$T_d = 0 \tag{14}$$

$$Gc(s) = (13.5 + \frac{5.749}{s}) \tag{15}$$

PID – Controller

$$K_p = 0.6K_{cr} = 18 \tag{16}$$

$$T_i = 0.5P_{cr} = 1.409 \tag{17}$$

$$T_d = 0.125P_{cr} = 0.352 \tag{18}$$

$$Gc(s) = (18 + \frac{12.775}{s} + 6.336s) \tag{19}$$

The controller and overall transfer functions are obtained using MATLAB command and it is applied in the Simulink diagram with closed-loop feedback [11]. The Simulink consists of uncompensated system, P – Controller system, PI – Controller system and PID – Controller system. The diagram is as shown in Figure2

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). This applies to papers in data storage. For example, write "15 Gb/cm2 (100 Gb/in2)." An exception is when English units are used as identifiers in trade, such as "3½ in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength *H* is A/m. However, if you wish to use units of T, either refer to magnetic flux density *B* or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

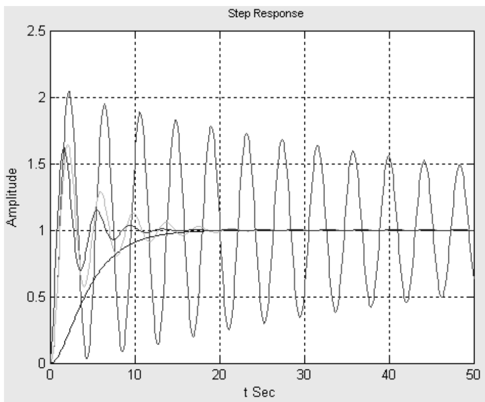

Fig. 3. Step response obtain by auto tuning programming

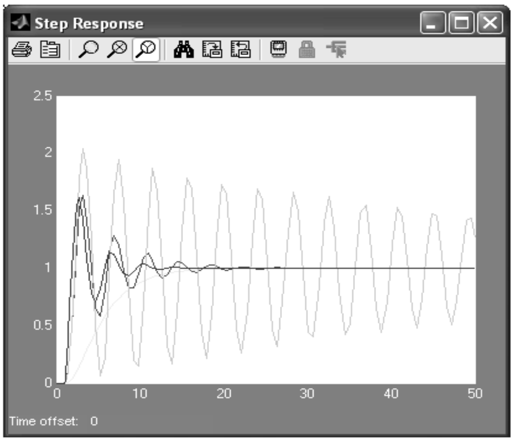

Fig. 4. Step response obtain by simulink

V. Validation Result

Fig. 3 represent the step response of uncompensated system, P – controller system, PI – controller system and PID – controller system obtained by Auto Tuning PID controller programming and Figure 4 represent the step response of all the three type of controllers and uncompensated system that obtained by Simulink. Both results show the same trend and values that proved the auto tuning program developed is valid and functional.

VI. Conclusion

Ziegler-Nichols method is utilized to convert from manual calculation into programmable calculation. The main objectives of the research is to perform the optimization of auto tuning PID controller (Ziegler –Nichols) using MATLAB, Simulink and Graphical User Interface (GUI) as well as acquiring the design of PID controller without excessive use of mathematical calculation. This research has achieved its objectives in a way that a fully functional auto tuning program is designed to obtain desired result without excessive use of mathematics. Furthermore if the mathematical model or transfer function of the system is known the program solves the problem in order to obtain the most suitable controller.

References

[1] K. Ogata, *Modern Control Systems*, Prentice Hall, 2005.

[2] N. S. Nise, Control Systems Engineering, Wiley, 2008.

[3] K. Ogata, System Dynamics, 3rd Edition, Prentice Hall, 1996.

[4] X. Yang, Q. Chen, and D. P. Atherton, "Linear Feedback Control: Analysis and Design with MATLAB," SIAM, 2007.

[5] M. Araki,"Control Systems, Robotics, and Automation – Vol. II - PID Control," Encyclopedia of Life Support Systems (EOLSS).

[6] A. A. Kumar, *Control Systems*, Prentice Hall, 2007

[7] D. Hanselman and B. Littlefield, *Mastering MATLAB 6 - A comprehensive Tutorial and Reference*, Pearson Education, 2001.

[8] D. Valerio and J. S. D. Costa, "Ziegler-Nichols Type Tuning Rules for Fractional PID Controllers," Technical University of Lisbon, Lisboa, Portugal, 2005.

[9] M. J. Willis, "Proportional-Integral-Derivative Control," Dept. of Chemical and Process Engineering University of Newcastle, 1999.

[10] Y. G. Wang, W. J. Cai, and Z. G. Shi, "PID autotuning for integrating processes with specifications on gain and phase margins," School of Electrical and Electronic Engineering Nanyang Technological University, Singapore 639798, 2001.

[11] C. Moler, "Numerical Computing with MATLAB," SIAM, 2004.