

Design and implementation of close loop DC motor speed control based on LabView

Ali Nihad Abbas

Dept. of Computer, Studies Planning and Follow up Office, Ministry of Higher Education and Scientific Research, Iraq

Abstract: LabVIEW is a graphical programming language. LabVIEW provides the flexibility of integration of data acquisition software/hardware with the process control application software for automated test and measurement applications. In this paper, an effort is made to control the speed of the DC motor and figure out the design process of a PID controlling system of close loop system using LabVIEW software. By LabVIEW-aided PID controller, the parameters are adjusted to control the motor speed in close loop. Design of controller by applying several methods in analyzing controlled parameters to tune parameter in order to obtain the best process response.

Keywords: LabVIEW, DC Motor, PID controller, Close Loop System and PID tuning method.

I. INTRODUCTION

Virtual instrument is a computer instrument system. The system is based on the computer including a lot of hardware equipment and the consumer design the virtual panel and procedure to achieve the testing and controlling purpose. In recent years, virtual instrument technology has been applied extensively in various fields, such as electronic, industrial production, automation electric power communication and industrial control [1]. DC motors have been popular in the industry control area for a long time, because they have many good characteristics, for example: high response performance, easier to be linear control start torque characteristic etc. There are different control approaches which depend on the different performance of motors. The basic property of DC motor is that speed can be adjusted by varying the terminal voltage. Tuning of PID parameters should be done to get the optimal response. This paper is to design PID controller to supervise and control the speed response of the DC motor with the virtual instrument graphic monitor software LabVIEW and also describes a simulation that involves designing, building, and demonstrating a regulatory control system. There are many algorithms/methods proposed in literature for tuning of PID controller such as Process reaction curve, Ziegler Nichols method, Tyreus and Luyben. LabVIEW is the graphical development environment for measurement, control applications rapidly and at minimal cost and creating flexible and scalable test. The structure of this paper is mainly divided into the

- Mathematic model of DC motor
- PID control theory and Tuning Algorithm
- LabView software simulation results

II. MATHEMATICAL MODEL OF DC MOTOR

The DC motor can be best represented by a transfer function. A transfer function provides a mathematical description for how the inputs and outputs of a system are related. In our case, the input to the system is voltage (V_m) and the output from the system is angular velocity (Ω_m). We can use the equation below to represent the model of our DC Motor where:

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{K_m}{J_{eq}R_m s + K_m} \quad (1)$$

K_m = Motor back-EMF constant (V/(rad/s))

R_m = Motor armature resistance (Ohms)

J_{eq} = Equivalent moment of inertia (kg*m²) (Assume that $J_{eq}=J_m$ (Motor armature moment of inertia)[2].

III. PID CONTROL THEORY AND TUNING ALGORITHM

The development of PID control theories has already 60 years ago, PID control has been one of the control system design method of the longest history. However, this method is still extensively used now [3]. The structure of PID controller is simple; it is the most extensive control method to be used in industry so far. The PID controller is mainly to adjust an appropriate proportional gain (K_p), integral gain (K_i), and differential gain (K_D) to achieve the optimal control performance. The PID controller system block diagram of this paper is shown in Figure 1.

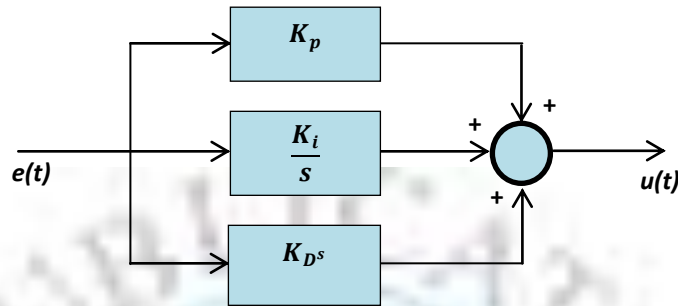


Figure 1: PID Controller System Block Diagram.

The relationship between the input $e(t)$ and output $u(t)$ can be formulated in the following,

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (2)$$

The transfer function is expressed as follows

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{U(s)}{E(s)} \quad (3)$$

The DC motor speed control using Close loop PID controller system block diagram is shown in Figure 2.

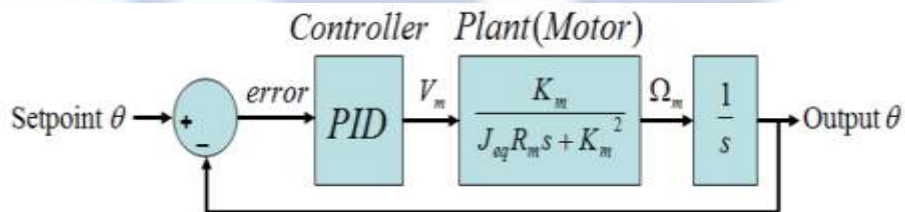


Figure2:Close Loop PID DC Motor Speed Control System Block Diagram

There are many methods proposed for tuning of PID controller. We have used the following four methods for tuning [4].

- Process reaction curve (Open-loop)
- Ziegler Nichols method (Closed-loop)
- Tyreus and Luyben

Process Reaction Curve (Open-loop): This is a manual procedure which is based on measuring the step response of the system .The input variable $r(t)$ is changed from r_0 to r_1 ;the output $c(t)$ is monitored carefully and the step response recorded on a chart recorder in Figure 3. Here $r(t)$ should be as small as possible. On the recorded output, draw a tangent to enable the following measurements:

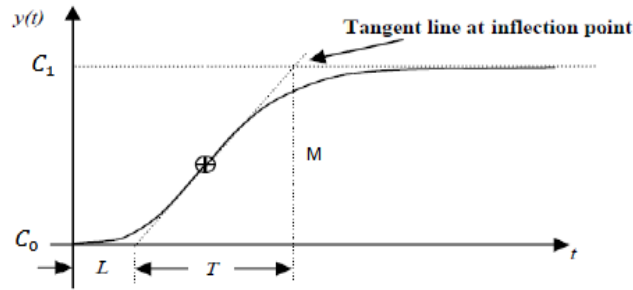


Figure 3: Shows Step Response of the Plant.

- Effective time delay in the system response 'L';
- Recorded output change $M = C_0 - C_1$;
- Measurement of 'T';
- P = percent change in correction unit
- M (measured variable) = percent of full scale range
- R (reaction rate) = maximum gradient of the graph

$$R = \frac{M}{T}$$

Empirical relationships in terms of K_p, K_i, K_D for P, PI, and PID are by process reaction curve is given in Table.1

Table 1: shows process reaction curve parameters

Type of Controller	K_p	K_i	K_D	T_i	T_D
P	$\frac{P}{RL}$	-	-	-	-
PI	$\frac{0.9 P}{RL}$	$\frac{1}{3.33 L}$	-	$3.33 L$	-
PID	$\frac{1.2 P}{RL}$	$\frac{1}{2 L}$	$0.5 L$	$2 L$	$0.5 L$

Ziegler Nichols Method (Closed-loop): Ziegler- Nichols is a type of continuous cycling method for controller tuning. The term continuous cycling refers to a continuous oscillation with constant amplitude and is based on the trial-and-error procedure of changing the proportional gain (K_p). (K_p) is reduced from larger value till the point at which the system goes to unstable state i.e. the point at which the continuous oscillations occurs. Thus the gain at which system starts oscillating is noted as ultimate gain (K_u) and period of oscillations is ultimate time period (P_u). It allows us to use the ultimate gain value, (K_u), and the ultimate period of oscillation (P_u) to calculate (K_c). These two parameters, (K_u) and (P_u) are used to find the loop-tuning constants of the controller (P, PI, or PID) using the formula tabulated in Table 2.

Table 2: Shows Ziegler Nichols parameters

Controller	K_p	T_i	T_D
P	$0.5 K_u$	∞	0
PI	$0.45 K_u$	$\frac{P_u}{1.2}$	0
PID	$0.6 K_u$	$\frac{P_u}{2}$	$\frac{P_u}{8}$

The advantage of this method is that it is a proven online method and includes dynamics of whole process, which gives a more accurate picture of how the system is behaving. The disadvantage is that it upsets the process, uses trial and error method and has a very aggressive tuning. This closed-loop tuning method is limited to tuning processes that cannot run in an open-loop environment.

Tyreus and Luyben: A modification to the above Ziegler-Nichols tuning parameters was made by Tyreus and Luyben shown in Table.3

Table 3: shows Tyreus and Luyben parameters

Type of Controller	K_p	T_I	T_D
PI	$\frac{K_u}{3.2}$	$2.2 P_u$	0
PID	$\frac{K_u}{2.2}$	$2.2 P_u$	$\frac{P_u}{6.3}$

IV. SIMULATION (ZIEGLER NICHOLS METHOD (CLOSED-LOOP))

This model will be used to design a closed-loop controller, which can then be tested with the actual motor. We can represent this transfer function in LabVIEW by using a MathScript Node which is part of the LabVIEW MathScript RT Module. The input parameter values were obtained from the Quanser QNET DC Motor specifications sheet.

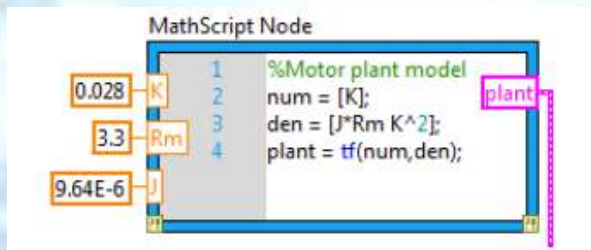


Figure 4: Model of the DC Motor Represented in the LabVIEW Mathscript Node

for this paper we will design a PI controller for our DC motor using the LabVIEW Control Design and Simulation module. The Simulation Loop, which includes a built-in ODE solver for handling integrals and derivative terms, can be found in the Control Design and Simulation palette under Simulation.

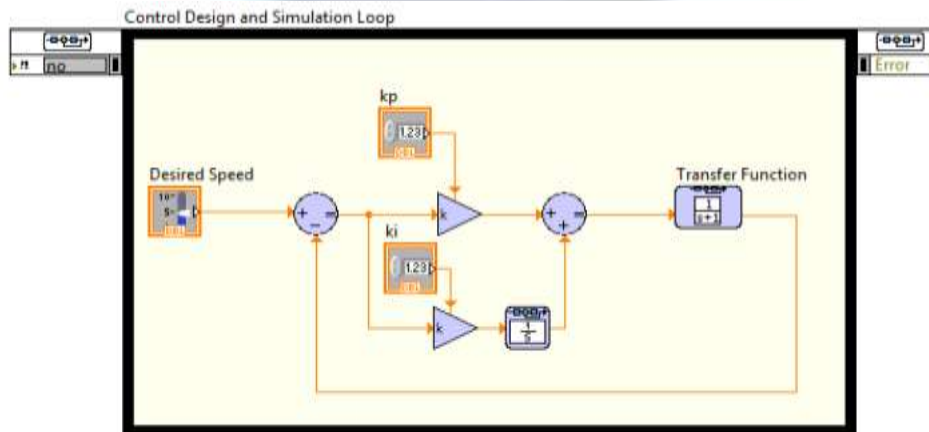


Figure 5: Closed-Loop PI Controller in LabVIEW Control Design and Simulation

The next step is to simulate the response of the DC Motor when modifying the set point or desired speed input. This will allow us to tune the controller parameters or gains to increase the robustness of our system. We will need to combine the Transfer Function or model of the DC motor that we created in Step 1 with our closed-loop controller.

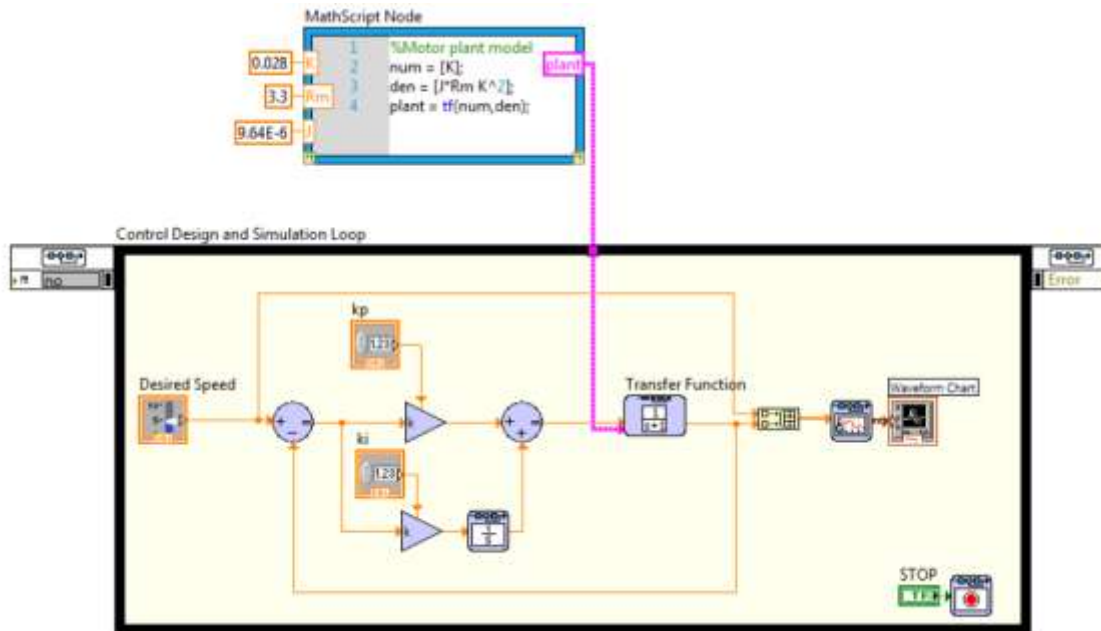


Figure 6: Closed-Loop PI Controller with DC Motor Transfer Function

Now that we can simulate both our controller and the response of the DC motor, we can follow an iterative process to optimize our controller. We will tune the controller parameters from the LabVIEW front panel while verifying the system performance.

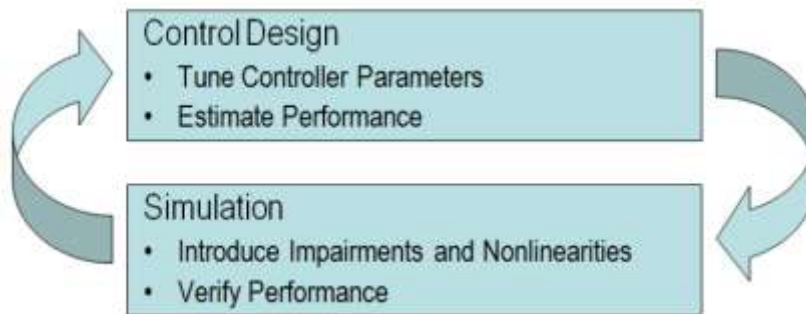


Figure 7: An Iterative Process for Optimizing the Controller

we can use the following steps to tune our controller parameters:

- Begin with gains set at: $K_p = 1$ and $K_i = 0$
- Increase Proportional Gain (K_p) to get desired rise time
- Increase Integral Gain (K_i) to reduce steady-state error if necessary

Once we run the program, we can see the desired motor speed and the estimated motor speed plotted on the Waveform chart. While the rise time looks good with our proportional gain, K_p set to 1, the plot shows a small amount of steady-state error, which is represented by the gap between the desired speed data and the estimated speed data. We can reduce this steady-state error by increasing our integral gain, K_i .

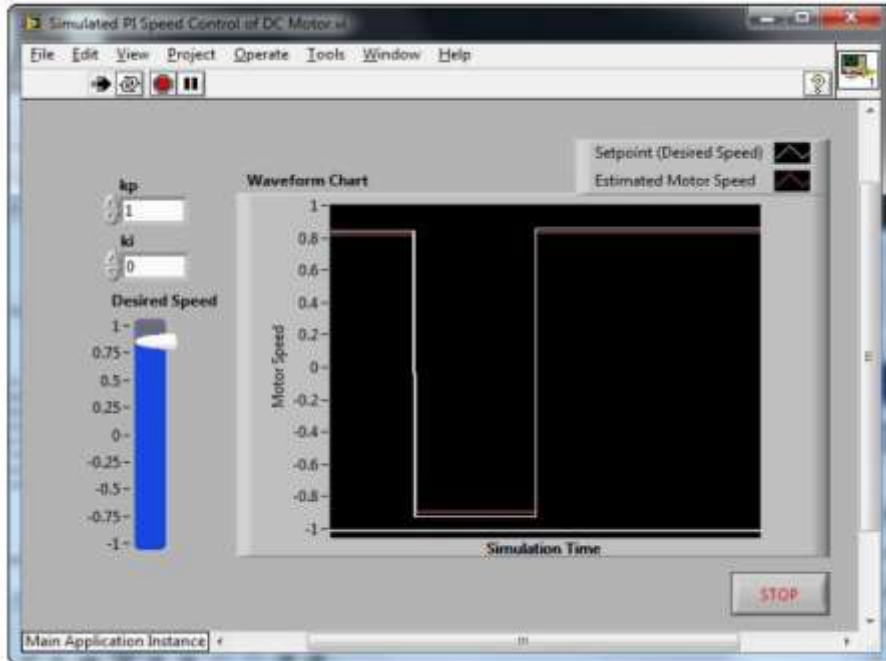


Figure 8: Simulating the DC Motor Response with a Proportional (P) Controller

if we increase the integral gain, K_i , to 10, we have a much better system response.

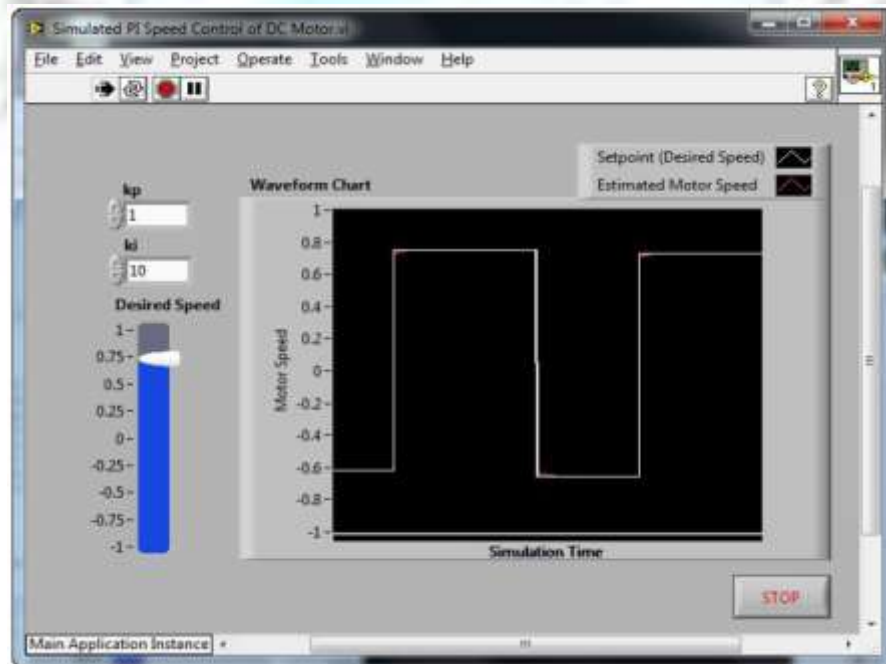


Figure 9: Simulating the DC Motor Response with a Proportional-Integral (PI) Controller

V. IMPLEMENTATION

Now that we've verified that our PI controller works with the simulated DC motor response, we can implement our finalized control system and control the speed of our Quanser DC Motor plug-in board for NI ELVIS. The LabVIEW Control Design and Simulation Module can be used to control real life systems as well as simulated models. To migrate from simulated control to real life control, the plant model can be replaced with hardware input and output functions. In this

case, we will replace the transfer function representing the DC motor with Data Acquisition (DAQ) input and output VIs that controls the actual motor.

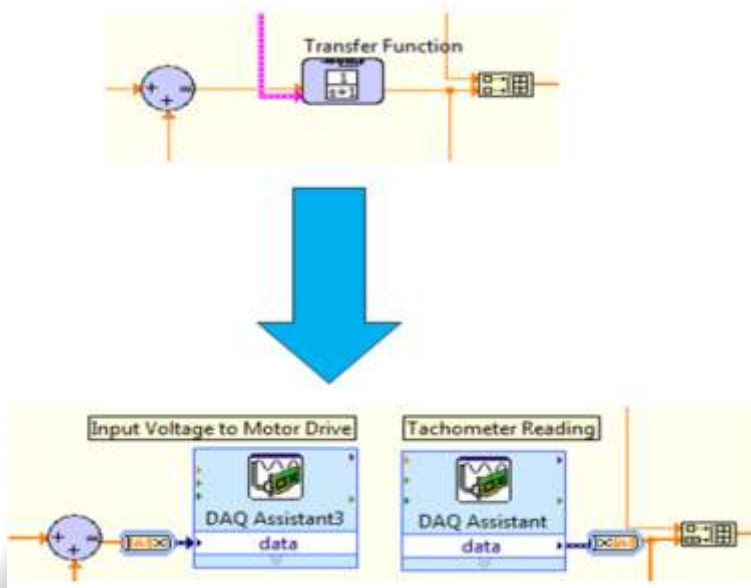


Figure 10: Migrate from Simulation to Real Hardware by Replacing Transfer Function with Hardware Input/output Blocks

Timing is an important consideration when using the LabVIEW Control Design and Simulation Loop with real hardware. Since the Control Design and Simulation Loop uses a built in ODE solver with time steps, it is important to set the Simulation Parameters and the Timing Parameters of the loop to have the same time step. Data acquisition tasks typically use timing parameters, so it is also important to equate the simulation loop timing with the data acquisition timing. We can now run the LabVIEW program and control the speed of the Quanser DC Motor from the LabVIEW front panel.

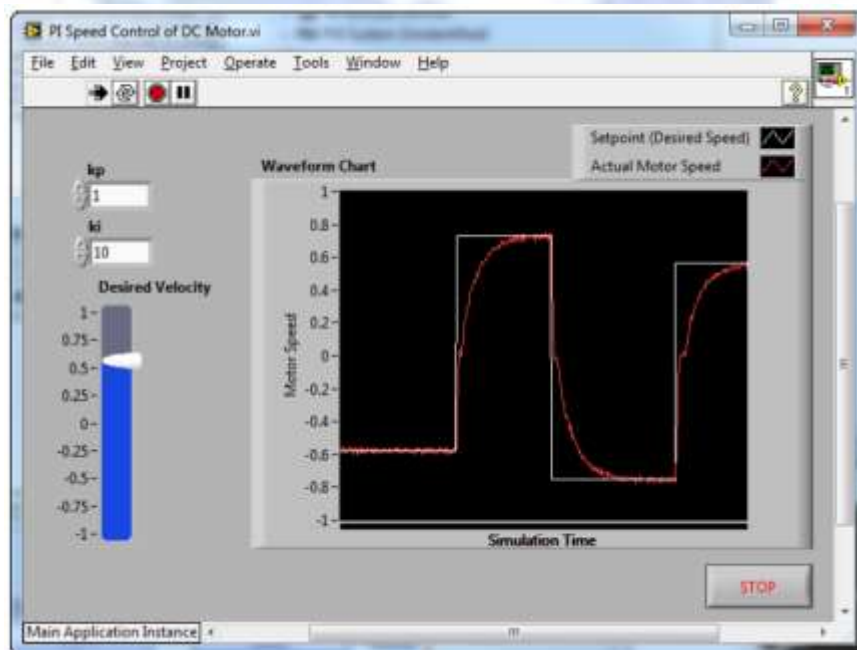


Figure 11: Response of the Actual Quanser DC Motor with our PI Closed-Loop Controller

VI. CONCLUSIONS

In this paper PID controller is designed using LabVIEW. Graphical representation of program gives us clear understanding about the structure and data flow. LabVIEW is a debugging tool. It is better for control application while MATLAB is better for data manipulation. Execution time is better in LabVIEW and survey ability is better in LabVIEW than MATLAB. So we use LabVIEW for DC motor speed control. Tyreus and Luyben method results in less oscillation at the control system response. Less oscillations means better controllability and less sensitivity to change in system condition. This technique can be extended to other types of motors. The parameters of PID controller can also be tuned by using Fuzzy logic or Genetic Algorithm (GA). The system offers successful simulation of a practical example of PID temperature control. With the LabVIEW professional development system or application builder, it can build stand-alone executable by taking standard transfer function. Time of system has been reduced greatly, compared with the system constructed by traditional instrument; it has such advantages as stabilizing the speed in a short time, even with good performance and flexibility. So it is finding that virtual instruments have the good flexibility and powerful functions in the application of the measurement system, which means it is a good choice in the design of industrial control system with high measuring precision, easily manipulation, high accuracy, better stability etc.

REFERENCES

- [1]. Lab VIEW Basics I Course Manual, National Instruments, 1999.
- [2]. PC-based PID Speed Control in DC Motor.
- [3]. PID Control of Continuous Processes by John W. Webb Ronald A. Reis.
- [4]. Control Engineering by K.P Ramachandran. Wiley Precise Textbook, 2011. National Instruments, "LabVIEW Graphical Programming", 2011.
- [5]. National Instruments, "LabVIEW Course LV1 & LV2", 2011.
- [6]. www.ni.com/control toolkit, 2013.
- [7]. Johnson, G.W., Jennings R., "LabVIEW Graphical Programming", McGraw-Hill press, New York, pp. 90-101, 2001.
- [8]. Spoelder, H.J.W., "Virtual instrumentation and virtual environment", IEEE instrumentation and measurement magazine, 2(3), 14-19, 1999
- [9]. Goldberg, H., "What is virtual instrumentation", IEEE instrumentation and measurement magazine, pp. 10-13, 2000.
- [10]. Saco, R., Pires, E., Godfrid, C., "Real Time Controlled Laboratory Plant for Control Education", IEEE Frontiers in Education Conference, pp. T2D12 – T2D16, 2002.
- [11]. The Design of PID **Controllers** using Ziegler Nichols Tuning Brian R Copeland (March 2008)
- [12]. Using the Lab VIEW PID Control Toolkit with the Lab VIEW Control Design and Simulation Module Sep 2009
- [13]. Control Engineering by K.P Ramachandran. Wiley Precise Textbook, 2011.
- [14]. Classical PID Control by Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado (Control System Design, Prentice Hall PTR)
- [15]. Guoshing Huang, Shuocheng Lee, Department of Electronic Engineering, National Chin-Yi University of Technology, 35, Lane 215, Chung-Shan Rd., Sec. 1, Taiping, Taichung, Taiwan, 41111, ROC E-mail:hgs@ncut.edu.tw
- [16]. "Getting in tune with Ziegler-Nichols," Thomas R. Kurfess, PhD, in the Academic Viewpoint column, Control Engineering magazine, Feb 2007 issue, p. 28, <http://www.controleng.com/>.
- [17]. A more recent survey that covers the Ziegler-Nichols and Kappa-Tau tuning rules: "Automatic Tuning of PID Controllers," Karl J. Åström & Tore Hägglund, Chapter 52, The Control Handbook, IEEE/CRC Press, 1995, William S. Levine ed.
- [18]. PID Example: DC Motor Speed Control "ACT Ziegler-Nichols Tuning," <http://ourworld.compuserve.com/homepages/ACTGMBH/zn.htm>
- [19]. Salim, Sunil Kumar, Jyoti Ohri, " LabVIEW Based DC Motor and Temperature Control Using PID Controller", ijarcse, Volume 3, Issue 5, May 2013, pp. 56-63.