Speed Control of Direct Current Motor By Using Laboratory Virtual Instrument Engineering Workbench (LabVIEW)

S. E. Vivek[1], V. Amit Kumar Singh[2]
[1, 2] Dept of Electrical and Electronics Engineering
[1, 2] Francis Xavier Engineering College, Tamil Nadu, India

Abstract- *This paper is going to be about controlling the speed of the DC motor by using the software called LabVIEW, Arduino and PID controller. LabVIEW is abbreviated as Laboratory Virtual Instrument Engineering Workbench which is a system-design platform and development environment for a visual programming language from National Instruments (NI). Arduino uses a variety of microprocessors and controllers, thus programmed and can be made it to function as we like. Arduino features serial communications interfaces, including Universal Serial Bus (USB) which are used for loading programs from personal computers. The PID (Proportional Integral Derivative) Controller is used where the continuously modulated controlling of speed is required. The speed of the motor is being sensed by using tachometer. The Tachometer is a sensor which measures the revolutions of the motor. From the tachometer, the output is sent back to the PID Controller via Analog and digital pins of the Arduino board which is processed by LabVIEW and Sends the commands to the PID controller. In this paper, we use basic open loop and closed loop system along with PID for control the DC motor*

Keywords- Speed Controlling, Arduino UNO, LabVIEW, LIFA_Base, PID, Tachometer.

I. INTRODUCTION

Laboratory Virtual Instrument Engineering Workbench, which is so called LabVIEW offers the user / programmer / Engineer/ Scientist to do graphical programming which helps the user to visualize and experiment virtually what he/she is expecting to do. The Direct Current Motor (DC Motor) plays a very important role everywhere in industries and laboratories because of their low cost and simplicity. There are three methods to control the speed of the motor. They are voltage control method, armature control method and flux control method. Here, Voltage control method is used to control the speed of this motor. Also, the arduino software and hardware is used as an intermediate between LabVIEW and the machine (Motor). At first, the arduino changes the analog signals into digital signals, so that the LabVIEW software can easily execute it. The user interface was developed in Arduino environment. The important goal of this work is to interface the arduino with LabVIEW by using LIFA_Base thus to control the speed of the dc motor using PID controller.

Modified Harvard architecture of ATmega328 is used here which has the operating voltage of 5V. It is the microcontroller which has 8-bit RISC processor core.

II. ARDUINO AND PID CONTROLLER

Arduino: Based on the ATmega328, the Arduino Uno is a microcontroller board which has 14 digital input/output pins out of which 6 can be used as PWM outputs and 6 analog inputs which are a 16 MHz ceramic resonator, a power jack, a USB connection, an ICSP header and a reset button. It contains everything needed to support the microcontroller and it is very simple to connect it to a computer just with a USB cable and power it with an AC-to-DC adapter or battery for it to get started. This Arduino features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter instead of using the FTDI USB-to-serial driver chip

The Microcontroller used here is *ATmega328*, the Operating Voltage is *5V,* and the recommended Input Voltage is 7-12V where the limits range from 6V to 20V. The Clock Speed 16 MHz There are 14 Digital I/O Pins out of which 6 provide PWM output and also there are 6 Analog Input Pins. The DC Current for each I/O Pin is 40 mA and the DC Current for 3.3V Pin is 50 mA. The Flash Memory of ATmega328 is 32 KB in which 0.5 KB is used by boot loader. The SRAM of ATmega328 is2 KB and the Electronically Erasable and Programmable Read Only Memory (EEPROM) of ATmega328 is 1 KB are the specifications of Arduino used in this work.

PID controller: The proportional–integral–derivative controller (PID) is capable of manipulating the process inputs based on rate of change of the signal and its history. The PID is most commonly used as a feedback controller. It is widely used in Industrial control systems as a generic control loop for feedback mechanism. It calculates error values as the difference between a desired set point and a measured process variable. The controller attempts to minimize the error by adjusting the process control inputs [4]. It consists of an Integral element, a Proportional element and a Derivative element. These three elements are connected in parallel connection. All the elements take the errors as the input. Here, K_d, K_p, K_i are the gains of D, P, and I. The schematic diagram of a system with PID controller is shown in the below Figure. The controller compares the measured process value with a reference set point value. The input from the difference of error value will try to adjust the measured process value into to the desired set point as in the history. The PID controller is an open loop controller as well as closed loop controller. Open loop control which has no feedback is in many cases which are not satisfactory, and it is often impossible because of the system properties. The performance can be improved by adding feedback from the system output, thus it will be a closed loop system.

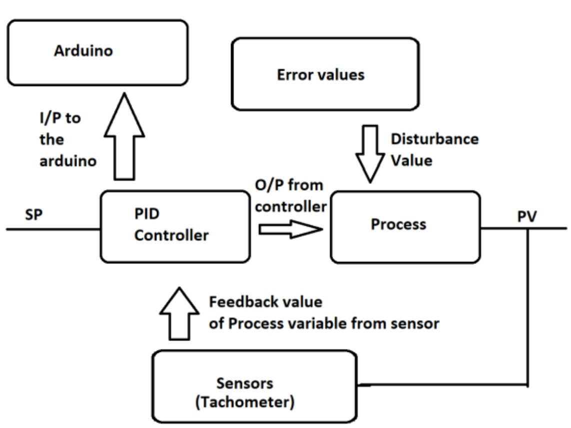

Figure 1: Schematic Diagram of the system of Arduino and PID controller

$$\textit{Error Value} = \textit{Desired Reference Value} - \textit{Measured value} \qquad \text{(i)}$$

Thus there are 3 major methods for managing the errors occurring in the system. Those are by handling the present by Kp, by recovering the past by Ki and by anticipating the future by Kd.

III. A. INTERFACING OF ARDUINO WITH LabVIEW

The Interfacing of arduino with LabVIEW is done by using LIFA toolkit (LabVIEW Interface for Arduino). The Arduino with microcontroller acts as an I/O device that interfaces the LabVIEW in serial connection. This helps controller to move the information from the Arduino to LabVIEW without any adjustment in the communication, synchronization or even in a single line of code. Thus by using the Open, Read/Write, Close properties in LabVIEW, we can access the analog, digital and PWM signals of the Arduino. The Arduino must be connected with the computer by using USB cable, Serial, or Bluetooth devices by LabVIEW LIFA extension toolkit.

III. B. USING PWM ON INTERFACED ARDUINO

Pulse Width Modulation (PWM) is a method of reducing the average power by effectively chopping it up into discrete parts which is delivered by an electrical signal. It is a technique for supplying electrical power as signals in continuous voltage pulses to a load that has a relatively slow response. The effective voltage level is controlled by the width of individual pulses given to the load. Both DC and AC signals can be simulated with Pulse Width Modulation. It acts like a DC signal when devices receive the signal which have an electromechanical response and time is slower than the frequency of the pulses. Here

τ_T - The time duration of voltage signals comprised of electric pulses
τ_s – The time duration for every second
V_o – Output Voltage of PWM
V_{in} – Input Voltage
V_{eff} - Effective DC Voltage

Thus the ratio would be

$$(\tau_T / \tau_s) = (V_{eff} / V_o) \qquad \textbf{(ii)}$$

The ratio of the duty cycle $\tau_T = \tau_s$ has to be equal to equalize the speed of the motor with the earlier measured speed. Here, it is varied to control the speed of the motor. The PID controller is connected in any one of the 14 digital Input /

Output pins. Thus after interfacing of arduino with LabVIEW with the LIFA extension and the controller is fixed in the arduino, the motor is driven and the speed can be controlled by switching between ON and OFF. The arduino has 4 different functions. To get the input, Analog Read and Digital Read functions are used and to generate the output, Analog Write and digital Write functions are used. Since LabVIEW is a digitally functioning programming language, Digital read, digital write and PWM signals are used. Digital Inputs are given in HIGH and LOW format. Electrically, the value for HIGH is 5V and for LOW is 0V. Those are the digital signals 0 and 1 if LOW and HIGH.

Thus the effective voltage would be

$$V_{eff} = V_o \tau_T / \tau_s \qquad \text{(iii)}$$

The applied frequency (f_s) = 500 Hz. Thus

$\tau_s = 1 / \; f_s = 1/500$

This is approximately 2ms. The output level of PWM with the AnalogWrite is an 8-bit value which corresponds to an effective voltage range of 0 to 5 V. Thus the PWM output level would be the product of 51 and effective dc voltage.

IV. FRONT PANEL AND BLOCK DIAGRAM

In LabVIEW, the Block diagram is the area where the function blocks are dragged and dropped. The diagram is a plain window which lets us to program certain functions. The Front panel is the area where the inputs can be given and output can be seen. The block diagram and the front panel of Speed control of DC motor using PWM is shown below.

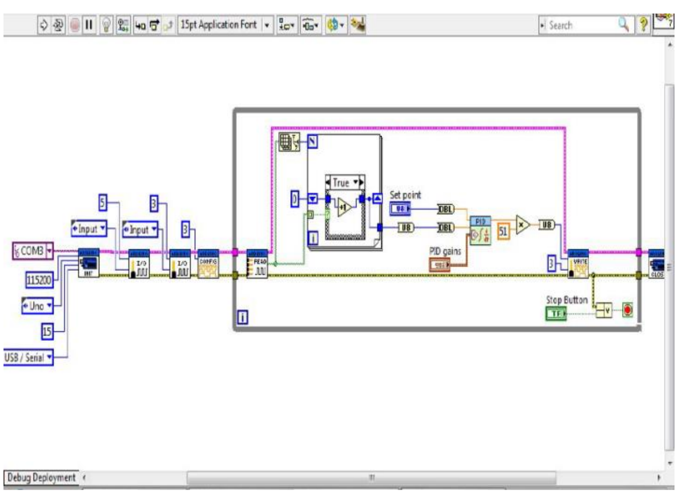

Figure 2 : BLOCK DIAGRAM OF THE CLOSED LOOP SYSTEM

The arduino functions are placed in the block diagram after installing LIFA extension. The program starts with the while loop with a for loop build inside it. The while loop is equipped with stop if true condition where the error is calculated and drawn inside the program. A case structure is also build inside the for loop with cases True and False. The input values are given digitally to the arduino functions and it is being compared with the set point values (5, 3, and 3). All the inputs from the arduino functions are collected as one and supplying it to the For and while loop. The values are given in and got out by various tunnelling modes such as Indexing, last value etc.

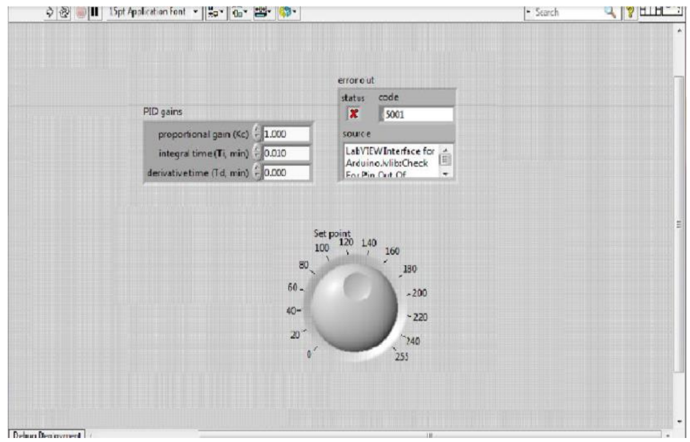

FIGURE 2 : FRONT PANEL OF THE CLOSED LOOP SYSTEM

The front panel of the closed loop system is shown in the above figure. Here, the output can be controlled by varying the input values. The PWM pulses (HIGH, LOW) are used to drive the motor. Tachometer sensor that gives external supply voltage as 5V senses the speed of the motor and passes the value to the LabVIEW. It compares the measured value with the set point value and manages the speed according to that.

V. CONCLUSIONS

Thus the Direct Current motor (DC motor) can be easily controlled just by using PWM signals and Arduino through LabVIEW programming. Here, LabVIEW and Arduino plays vital role to control the speed and manage the PWM signals for sensing and compensating the speed values from tachometer. They are all connected by Wires, USB cables and Blue tooth devices. This majorly helps in maintaining the stability of the system and this method can be used in various industrial applications and in Laboratories.

REFERENCES

[1] S.E.Vivek. K. Reishma, V. Amit Kumar Singh(2020, january 21). International Journal of Advanced Research in Electrical,. Retrieved from "http://www.ijareeie.com/upload/2020/january/4_IMPLEMENTATION_FC.PDF" http://www.ijareeie.com/upload/2020/january/4_IMPLEMENTATION_FC.PDF CITATION SEV20 \l 1033 (S.E.Vivek, 2020)

[2] Olden. P, "Open Loop motor speed control with LabVIEW", SoutheastCon, Proceedings IEEE, pp. 259-262, 2001, (Conference Publications)

[3] Ibrahim kaya, "IMC based automatic tuning method for PID controllers in a smith predictor configuration" , Sciencedirect, 2004

[4] Purna Chandra Rao et. al., "Robust Internal Model Control Strategy based PID Controller for BLDCM" ,International Journal of Engineering Science and Technology, Volume 2(11), 2010, 6801-6811.