

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305880774>

DC Motor Speed Control Using PID Controller In Lab View

Article · February 2015

CITATIONS

11

READS

7,940

3 authors:



Pratap Vikhe

Pravara Rural Engineering College Loni

40 PUBLICATIONS 225 CITATIONS

[SEE PROFILE](#)



Neelam Punjabi

3 PUBLICATIONS 39 CITATIONS

[SEE PROFILE](#)



C B Kadu

Pravara Rural Engineering College Loni

44 PUBLICATIONS 205 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Breast Cancer Detection [View project](#)



DC Motor Speed Control using PID Controller in LabVIEW [View project](#)

DC Motor Speed Control Using PID Controller In Lab View

Pratap S Vikhe, Neelam Punjabi, Chandrakant B Kadu

Abstract- Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. One of the applications used here is to control the speed of the DC motor. Controlling the speed of a DC motors is very important as any small change can lead to instability of the closed loop system. The aim of this paper is to show how DC motor can be controlled by using a PID controller in LabVIEW. DC Motor will be interfaced with LabVIEW using an ATmega 8A Microcontroller. The speed of the DC motor will be set by creating a Graphic User Interface (GUI) for PID Controller in LabVIEW. LabVIEW will send serial command to the DC motor using the PWM pins on the Microcontroller board. DC motor will move with the speed set by the user in LabVIEW. The speed of the dc motor will be sensed by using the IR sensor. From the sensor, the output is sent back to the PID Controller in LabVIEW via ATmega Microcontroller. PID Controller compares the actual speed of the DC motor with the set speed. If its speed is not same, PID Controller will try to minimize the error and bring the motor to the set point value [1].

Keywords: DC Motor, LabVIEW, PID Controller, IR Sensor, Open-Loop, Closed-Loop

I. INTRODUCTION

DC (Direct Current) Motors are two wire (power & ground), continuous rotation motors. When the supply power is provided, a DC motor will start spinning until that power is removed. Most DC motors run at a high RPM (revolutions per minute), examples being computer cooling fans, or radio controlled car wheels.

The speed of DC motors is controlled using pulse width modulation (PWM), a technique of rapidly pulsing the power on and off. The percentage of time spent cycling the on/off ratio determines the speed of the motor, e.g. if the power is cycled at 50% (half on, half off), then the motor will spin at half the speed of 100% (fully on). Each pulse is so rapid that the motor appears to be continuously spinning. A control system is an interconnection of components forming a system configuration that will provide a desired system response. DC Motor will be interfaced with LabVIEW using ATmega 8A Microcontroller. The role of Microcontroller is to pass the set speed to the DC motor using the PWM pins and to get the data (speed) from the motor using the IR sensor through Interrupt.

This paper is organized as follows: Section I includes the introduction to control system, dc motor and the IR sensor.

Section II gives the basics of proportional integral and derivative controller. Section III gives the Hardware Implementation Section IV Implementation of Open-Loop and Closed Loop in Lab VIEW Section V is Conclusion showing the results obtained.

II. PID CONTROLLER

The proportional integral derivative (PID) controller is the most common form of feedback used in the control systems. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner. It can be used for various Industrial applications. As the name suggests, PID algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response [3].

Unlike a simple proportional control algorithm, the PID controller is capable of manipulating the process inputs based on the history and rate of change of the signal. This gives a more accurate and stable control method [5]. The basic idea is that the controller reads the system state by a sensor. Then it subtracts the measurement from a desired reference to generate the error value. The error will be managed in three ways, to:

1. Handle the present, through the proportional term,
2. Recover from the past, using the integral term,
3. Anticipate the future, through the derivative term.

In this paper we will have shown an open loop system and a closed loop system with PID system to control the DC motor.

1. CONTROL SYSTEM

The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output [2].

2. OPEN LOOP SYSTEM

An open-loop controller, also called a non-feedback controller, is a type of controller that computes its input into a system using only the current state and its model of the system. A characteristic of the open-loop controller is that it does not use feedback to determine if its output has achieved the desired goal of the input. This means that the system does not observe the output of the processes that it is controlling. An open-loop controller is often used in simple processes because of its simplicity and low cost, especially in systems where feedback is not critical.

Figure below shows an Open-Loop System

Manuscript Received on February 2015.

Pratap S Vikhe, Associate Professor, Instrumentation & Control Engineering Department, Pravara Rural Engineering College, Loni, Maharashtra, India.

Neelam Punjabi, Lecturer, Biomedical Engineering Department, Vidyalkar Institute of Technology, Wadala, Maharashtra, India.

Chandrakant B Kadu, Associate, Instrumentation Engineering Department, Pravara Rural Engineering College, Loni, Maharashtra, India.

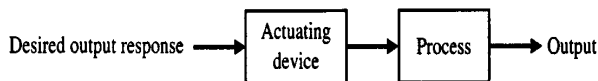


Figure 1: Open Loop System

3. CLOSED LOOP SYSTEM

In a typical control system, the process variable is the system parameter that needs to be controlled, such as temperature ($^{\circ}\text{C}$), pressure (psi), or flow rate (liters/minute). A sensor is used to measure the process variable and provide feedback to the control system. The set point is the desired or command value for the process variable. At any given moment, the difference between the process variable and the set point is used by the control system algorithm (compensator), to determine the desired actuator output to drive the system (plant).

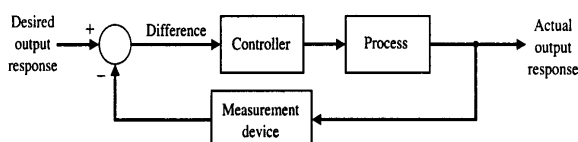


Figure 2: Closed Loop System with PID controller

III. HARDWARE IMPLEMENTATION

This designed controller is implemented in real time for 12volt, 1500rpm, and 1.5A DC motor. The motor is interfaced through ATmega 8A to LabVIEW and was thus controlled. The hardware components used are as follows:

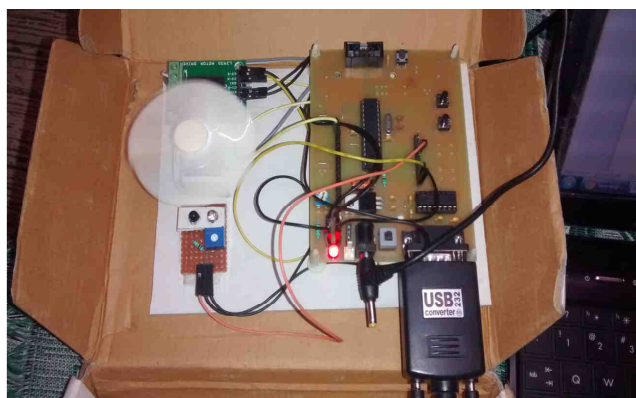


Figure 3: Hardware

ATmega 8A

The high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 8KB ISP flash memory with read-while-write capabilities, 512B EEPROM, 1KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte oriented two-wire serial interface, 6-channel 10-bit A/D converter (8-channel in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, SPI serial port, and five software selectable power saving modes. The device operates between 2.7-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves

throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. The main aim of ATmega microcontroller is to write the speed on the dc motor set by the user using its PWM pins and to read the speed sensed by the IR sensor.

Encoder

Optical encoders are devices that convert a mechanical position into a representative electrical signal by means of a patterned disk or scale, a light source and photosensitive elements. With proper interface electronics, position and speed information can be derived. Encoders can be classified as rotary or linear for measurements of respectively angular and linear displacements. Rotary encoders are available as housed units with shaft and ball-bearings or as "modular" encoders which are usually mounted on a host shaft (e.g. at the end of a motor)[1].

Speed sensor

The sensor is always ON, meaning that the IR led is constantly emitting light. This design of the circuit is suitable for counting objects, or counting revolutions of a rotating object, that may be of the order of 1500 rpm or much more.

Driver circuit

The circuit is built around an L293D motor driver. A driver circuit is an electrical circuit or other electronic component used to control another circuit or other component, such as a high-power transistor. They are usually used to regulate current flowing through a circuit or are used to control the other factors such as other components, some devices in the circuit.

IV. IMPLEMENTATION OF OPEN-LOOP & CLOSED-LOOP IN LABVIEW

A. Open-Loop System

The Block Diagram of Open-Loop is as shown below:

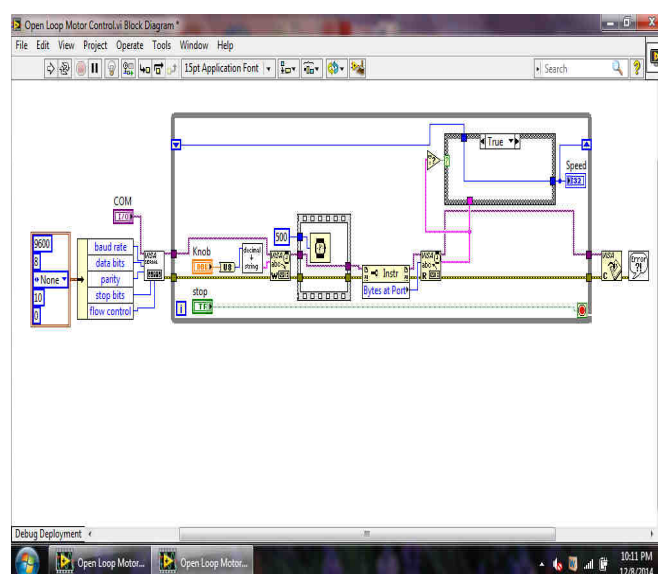


Figure 4: Block diagram of Open-Loop System

The program usually begins with the while loop on the block diagram and initializing the driver connections to LabVIEW. VISA Write will write the command (speed) set by the user on the ATmega microcontroller. VISA Read will read the data (speed) sensed by the sensor. Between every write command and read command there is a delay of

1000ms. At the end of the while loop, VISA Close command is used which will close the session.

Front Panel of Open-Loop system is as shown below:

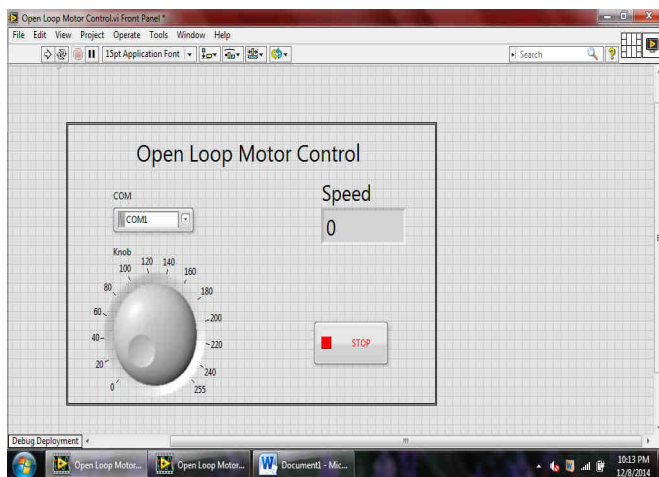


Figure 5: Front Panel of Open-Loop System

The front panel is the actual GUI (Graphic user Interface) for the user to set the speed of the DC motor. For every control on the front panel there is a respective block in the block diagram.

B. Closed-Loop System

The Front Panel Diagram of Closed-Loop System is as shown below:

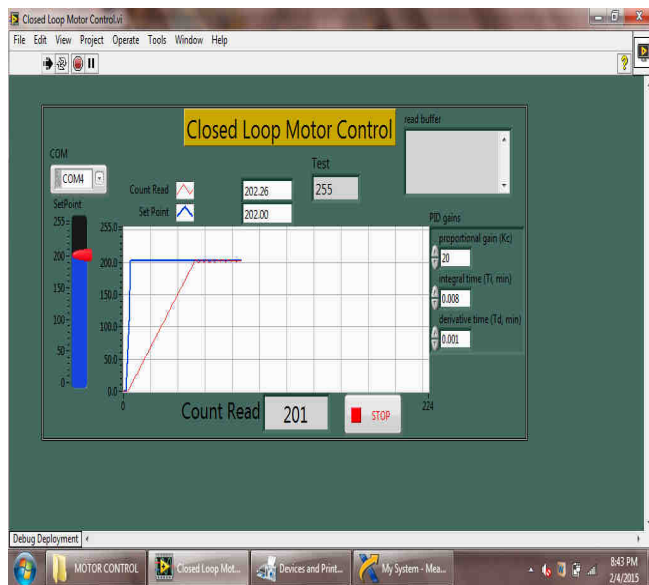


Figure 5: Front Panel of Closed-Loop System

PID controller will compare the setpoint value with the value received from the IR sensor. If the two values are not same, PID controller will try to minimize this error and bring the DC Motor to the desired speed.

The Block diagram of the Closed-Loop System is as shown below:

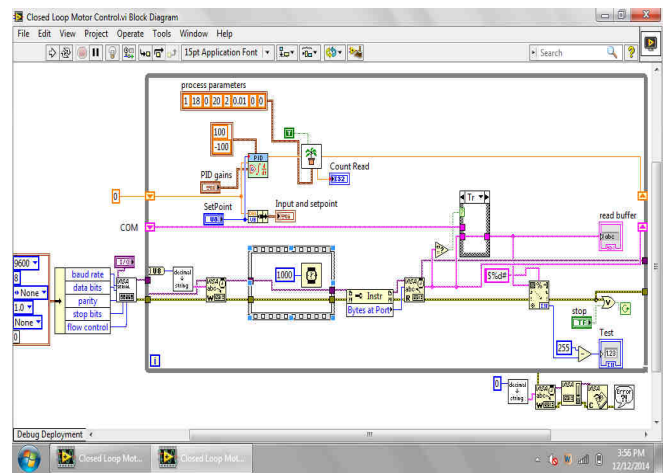


Figure 7: Block diagram of Closed-Loop System

Duty cycle can be varied from 0-255 by varying the user controlled interactive graphical dial on the front panel. The response of the system can be changed by varying the gains of PID controller. These VI's will be burnt in the ATmega microcontroller and interfaced with the dc motor. Setpoint, set by the user will be fed into the pid controller and passed on to the Microcontroller PWM pins. Microcontroller will pass those PWM pulses to the motor along with supply voltage that moves the motor. Shaft of the DC motor will move and number of times it moves will give us the speed at which the motor is moving. IR sensor is used for measuring the rotation of the DC motor.

IV. CONCLUSIONS

The method adopted in this paper is low cost technique of controlling the speed of the DC motor. DC motor is interfaced with PID Controller in LabVIEW via ATmega 8A Microcontroller. Speed of the motor is sensed by using the Infrared Sensor which is sent back to PID Controller as feedback for calculating and compensating the error produced if any. The method implemented can be used for various industrial applications. This technique helps in maintaining the stability of the system.

REFERENCES

1. Priyanka Rajput, "A Study of Speed Control of PMDC Motor using Auto-tuning of PID Controller through LabVIEW ", AKJEC International journal of Technology, Volume 5, No. 1.
2. Salim, Sunil Kumar, Jyoti Ohri "LabVIEW Based DC Motor and Temperature Control Using PID Controller " ,IJARCSSE, Volume 3, Issue-5, May 2013
3. Naveen Kumar R, et. al., "Low Cost Data Acquisition and Control using Arduino Prototyping platform and LabVIEW", IJSR, India Online, Volume 2, Issue 2, February 2013, ISSN: 2319-7064
4. A. 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
5. R. Vilanova, "PID Controller tuning rules for robust step response of first-order-plus-dead-time models", IEEE 2006.
6. M. H. Moradi, "New Techniques for PID Controller Design", IEEE Xplore, 2003
7. Umesh Kumar Bansal, Rakesh Narvey, "Speed Control of DC Motor Using Fuzzy PID Controller", Department of Electrical Engineering, Advance in Electronic and electric Engineering. ISSN 2231-1297, volume 3, Number 9 (2013), pp, 1209-1220.
8. Aditya Pratap Singh, Udit Narayan, et.al., "Speed Control of DC Motor using PID Controller Based on Matlab", Innovative System Design and Engineering, ISSN 2222-1727, Volume 4, No. 6, 2013.

9. Rinku Singhal et.al., "Design of Fractional Order PID Controller for Speed Control of DC Motor ", International journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012, ISSN 2250-3153
10. Philip A. Adewuyi, "DC Motor Speed Control: A Case between PID Controller and Fuzzy Logic Controller", International Journal of Multidisciplinary Science and Engineering, Volume 4, No. 4, May 2013.
11. Megha Jaiswal , Mohna Phadnis "Speed Control of DC Motor Using Genetic Algorithm Based PID Controller", IJARCSSE, Volume 3, Issue 7, July 2013