Design of PID controller for DC Motor Speed Control Using Arduino Microcontroller

Thirupathi Allam[1], Matla Raju[2], S.Sundeep Kumar[3]

[1](EEE, tiru.allam225@gmail.com)
[2](EEE, matla.raju33@gmail.com)
[3](EEE, shagasandeep@gmail.com)

Abstract: *PID controllers are most popular and most often used controllers in Industry. Popularity of the PID controllers are due to their wide range of operating conditions and functional simplicity. Different types of tuning rules have proposed which can fine tune the system to get desired response. The aim of the paper is to control robotic vehicle speed by controlling Motor speed using PID with reference to obstacles that will faced by vehicle. In this we are presenting method of automatic tuning of PID controller to control of speed of DC motor using Arduino microcontroller. DC Motor will be interfaced with Simulink using an Arduino Uno board. Arduino Uno board plays the role of low cost data acquisition board. Using object distance data measured by sensor PID controller will control the speed of the DC motor within set point limits.*

General Terms
Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords: *PID Controllers, DC Motor, Arduino microcontroller.*

1. INTRODUCTION

In general to control the speed of AC motors we will use some methods like frequency control methods. But in certain applications we will use DC motors so the speed Controlling of DC motor is very important in any application. In this paper the Speed of the DC motor is controlled by using Arduino microcontroller. Simulink will in turn pass this speed to the DC motor using a PWM pins on the Arduino Uno board. Using ultrasonic sensors obstacle distance is measured and PID controller values are calculated with reference to obstacle distance and speed to be controlled. PID controller will generate corresponding PWM pluses to control speed of the DC Motors.

In this paper, Section I gives Introduction, Section II gives the basics of proportional integral and derivative controller. Section III gives the overview of an Arduino micro controller and Arduino interface with Simulink. Section IV Design of PID controller Section V is the results obtained and Conclusion Section VI future scope.

2. PID CONTROLLER

PID controller is the combination of proportional, Derivative and integral controllers. PID controller works in a closed-loop system using the schematic shown below. The variable (e) represents the tracking error, the difference between the desired input value (r) and the actual output (y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The control signal (u) to the plant is equal to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.

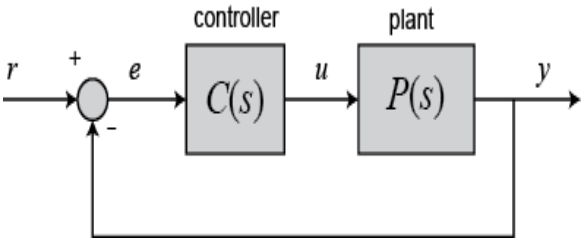

Fig1. PID Controller

This control signal (u) is sent to the plant, and the new output (y) is obtained. The new output (y) is then fed back and compared to the reference to find the new error signal (e). The controller takes this new error signal and computes its derivative and its integral again, ad infinitum.

The transfer function of a PID controller is evaluated as

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (1)$$

Here

K_p = Proportional gain

K_i = Integral gain

K_d = Derivative gain

3. ARDUINO MICROCONTROLLER

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The specifications of Arduino Uno are as follows:

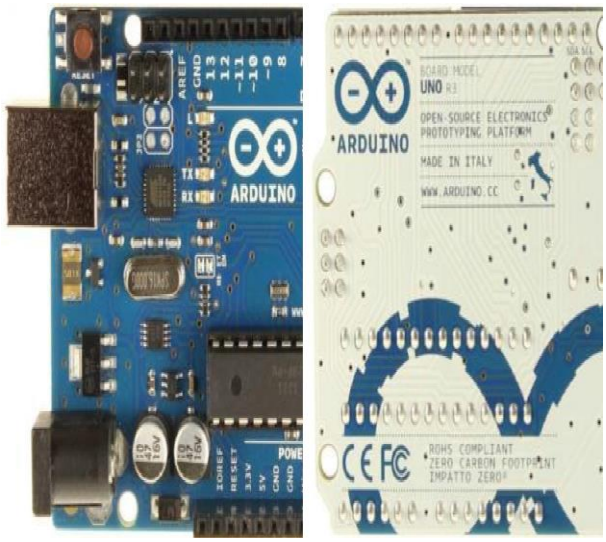

Fig2. Arduino UNO

Arduino Uno has 14 digital input/output (I/O) pins. Conventional, i.e., not PWM, operation of the digital I/O pins is controlled with the pinMode, digitalRead and digitalWrite functions. The pinMode function is used to configure a pin as an input or output. When a digital I/O pin is configured as an input, digitalRead reads the state of the pin, which will be either HIGH or LOW. In an Arduino sketch, HIGH is a predefined constant that is evaluated as “true" in a conditional expression, and is equivalent to a numerical value of 1. Electrically, a value of HIGH means the pin voltage is close to 5 V. Similarly, the constant LOW is interpreted as “false" in conditional expressions, it is numerically equivalent to 0, and electrically the pin voltage is 0. When a digital I/O pin is configured for output, digital Write is used to set the pin voltage to HIGH or LOW. On an Arduino Uno, PWM output is possible on digital I/O pins 3, 5, 6, 9, 10 and 11.

The specifications of Arduino Uno are as follows:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

4. DESIGN OF PID CONTROLLER

PID control is a basic control loop feedback mechanism. The controller minimizes the difference between the measured and the desired value of a chosen system variable by adjusting the system control inputs. PID controller is designed to with reference obstacle distance and it will generate corresponding PWM pulses to control speed of DC motor using Arduino. Motor speed is controlled using PWM techniques. Arduino uno is used generate PWM pulses to control motor speed. Using

sensor data obstacle distance is measured at Arduino pins . By analysing received data and using the condition

i) If obstacle is not found the motor will be operated at Constant speed.
ii) If measured obstacle distance found less than 100m than speed of the motor will be reduced as the obstacle approaches near. If the obstacle passes away then motor speed will be automatically increased and operated at constant speed.

4.1 PWM Pulses

PWM signal consists of a train of voltages pulses such that the width of individual pulses controls the effective voltage level.

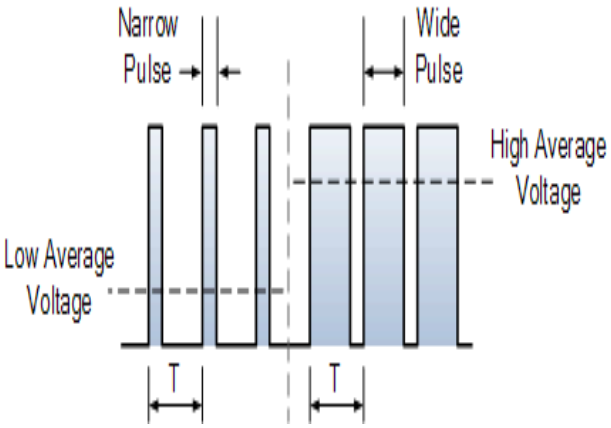

Fig3. PWM Pulses

The output of a PWM channel is either Vs volt during the pulse or zero volts otherwise. If this signal is supplied as input to a device that has a response time much larger than Ton , the device will experience the signal as an approximately DC input with an effective voltage of

$$Vout = Vs(Ton/T) \quad (2)$$

5. RESULTS

Entire Sysems is designed in Simulink tool and loaded on Arduino uno using USB interface. It consist of two blocks PID controller and Motors .

PID controller collects data from sensor to calculate the error and control the speed with in operated range.Motor block consist outpins and PWM Pins of Arduino.

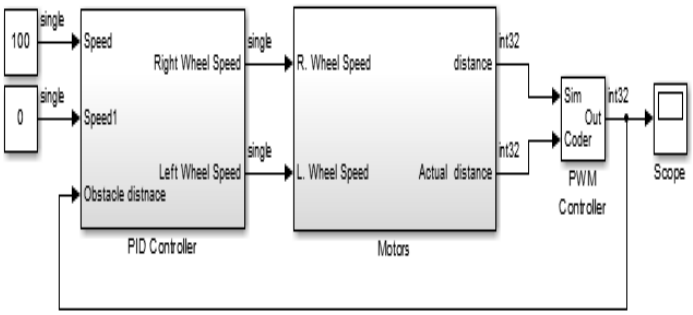

Fig 4. Simulink Block Diagram

Arduino PWM block in Simulink generate square waveform on the specified analog output pin. The block input controls the duty cycle of the square waveform. An input value of 0 produces a 0 percent duty cycle, and an input value of 255 produces a 100 percent duty cycle. The frequency of the waveform is constant at approximately 490 Hz.

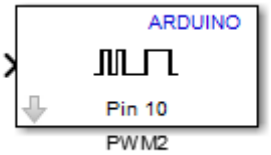

Fig 5. Simulink Arduino PWM Block

Ultra sonic sensors are used to detect the object distance. This object distance is given as analog input suing Arduino analog input pins. Using this data PID controller will generate corresponding singal to the PWM block.

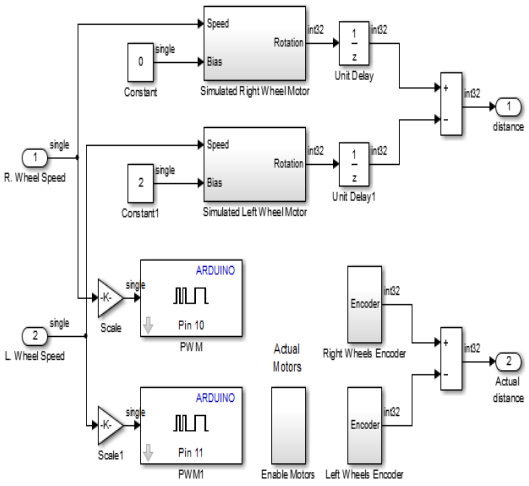

Fig 6. PID Controller

Output of PID controllers is given to the PWM pulses. PWM pulses are given to PIN 10 and PIN 11 in Arduino. An Extra Driven circuit is used to control the DC motors . L293D IC is used to control the current for two motors. Output of PWM pulses control the current drawn for the IC and speed of the DC motors are controlled.

To Check whether PID Controller is working properly or not we can use the encode data that is given at Arduino pins. By observing encoders data versus speed of the motor give the error report. By observing encoder the data error value varies in between 0 to 0.9 .As PID will tune automatically form the measured data we can observe the change very less time.

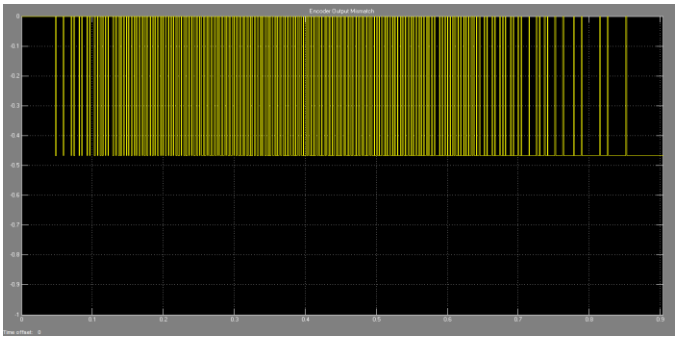

Fig 7. Encoder output data

6. QFUTURE SCOPE

This Design can extended to control four dc motors vehicle. In this Obstacle distance data is taken as refrence to control speed and we can extend this to control servo motors also.Error control further reduced by reducing the operating range difference of the speed.

References :

[1] Arduino in Action Martin Evans, Joshua Noble, and Jordan Hochenbaum

[2] Make electronics By Charles Plat 2009

[3] 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.

[4] Detchrat, et. al. , "IMC-Based PID Controllers Design for Two-Mass System", IMECS 2012 Volume - II, Hong Kong.

[5] Jason T. Isaacs, Daniel J. Klein, Joao P. Hespanha in *'A Guided Internship For High School Students Using iROBOT CREATE'.* Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA(e-mail:fjtisaacs,djklein,hespanhag@ece.ucsb.edu).

[6] V. Subburaman and S. Marcel. Fast Bounding Box estimation based face detection in '*Workshop on Face Detection of the European Conference on Computer Vision (ECCV)*', 2010.