MAT
J O U R N A L S

Design and Fabrication of speed control of A. C. Motor using PID controller

Rajesh Pansare[1]
Assistant Professor[1]
Department of Mechanical Engineering
K. J. Somaiya College of Engineering,
Mumbai - 400077, India.
rajeshpansare@somaiya.edu[1]

Vijay Bhosale[2]
Assistant Professor [2]
Department of Mechanical Engineering
K. J. Somaiya College of Engineering,
Mumbai-400077, India.
vijaybhosale@somaiya.edu[2]

Abstract
The objective of this paper is to study the basic concepts of PID controller and applications in industry. The paper also concentrates on how this concept of PID controller can be used speed control of A. C. motor. With the presented work the author has tried to design and fabricate the experimental set up for studying the speed control of A. C. motor. With the said experimental set up the output characteristics of P, I, D and PID combined controller are obtained. Based on this output characteristics of various combination of controllers and different values, the effect on output is studied. Each type of controller (i.e. P, I, D and PID) has its own effect on the output characteristics. The proportional controller has the effect of reducing rise time but no change in steady state error. The integral controller eliminates the steady state error whereas the derivative controller increases the stability of system. The aim of said work is to study and verify all above characteristics.

Keywords: *Proportional, Derivative, Integral, PID, Controller, Closed loop, System, speed control.*

INTRODUCTION
The Feedback control mechanism uses information from measurements. The main characteristic of feedback control system is the output is sensed as shown in figure 1.
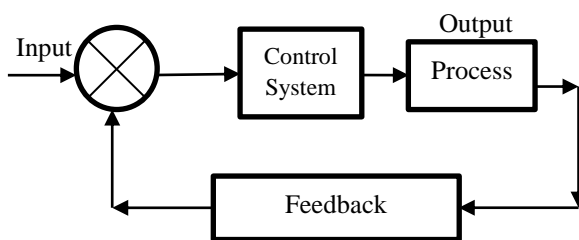
Fig 1: *Feedback control system*

There are two main types of feedback control systems: 1) positive feedback 2) negative feedback. The positive feedback increases the size of the input but the negative feedback is used to decrease the size of the input. Various types of controllers are used in feedback control system such as,

1. Proportional controllers.
2. Integral controllers.
3. Derivative controllers.

Combinations of these three controllers are written below:

4. Proportional and integral controllers.
5. Proportional and derivative controllers.
6. PID controller

A PID is widely used in feedback control of industrial processes and has remained the most widely used controller in process control until today. They provide control signals that are proportional to the error between the reference signal and the actual output (proportional action), to the integral of the error (integral action), and to the derivative of the error (derivative action) [1]. If u (t) and e (t) are control signal and error signals respectively, then,

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(T)dT + T_d \frac{d}{dt} e(t) \right] \quad ----- \quad (1)$$

Where, e (t), T_i and T_d are the parameters of control action to be tuned. Also the corresponding transfer function of PID controller will be as follows,

$$K(s) = K_p \left[1 + \frac{1}{Ts_i} + T_d s \right] \quad --------- \quad (2)$$

The PID controller is most dominating form of feedback used now a day. More than 90% of feedbacks used are PID. [2] The PID controls are used for wide range of control problems such as process control, motor drives, flight control, automotive etc.

This paper mainly deals with the design and fabrication of speed control of A. C. motor using PID controller. The effect of each type of controller on the output is obtained and studied.

LITERATURE REVIEW
In most of the industrial control systems, the PID controllers are very commonly used because the number of parameters required to be tuned are very less. The technique used for design is Ziegler–Nichols method which relies on the parameters obtained from plant step response [1]. Such controllers require three important parameters for design. They are proportional gain, integral time constant and derivative time constant [3].
The transfer function of PID controller will be as follows,

$$G_{c(s)} = K_p + \frac{K_i}{s} + K_d s$$

Where K_p, K_i and K_d are the Gains of proportional, integral and derivative control respectively.
Let, $T_i = K_p / K_i$ = Integral time constant,
$T_d = K_d / K_p$ = Derivative time constant.
Then, the another useful equivalent equation for transfer function of PID controller will be as follows,

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

The equivalent equation for Discrete-time of PID controller will be as follows,

$$u(k) = K_p e(k) + K_i T_s \sum_{i=1}^n e(i) + \frac{K_d}{T_s} \nabla e(k)$$

Where, u(k) is the control signal of system, and e(k) is the error between the reference input and the process output. The various response curves can be varied by varying the values K_p, K_i and K_d or K_p, T_d and T_i [3].
The response of the controller can be described in terms of its responsiveness to an error, the degree to which the system overshoots a set point, and the degree of any system oscillation. The use of the PID algorithm does not guarantee optimal control of the system or even its stability. To make the system stable is the requirement of the any controller. For that it is desirable to know the complete set of PID parameters of the plant before controller design [4].
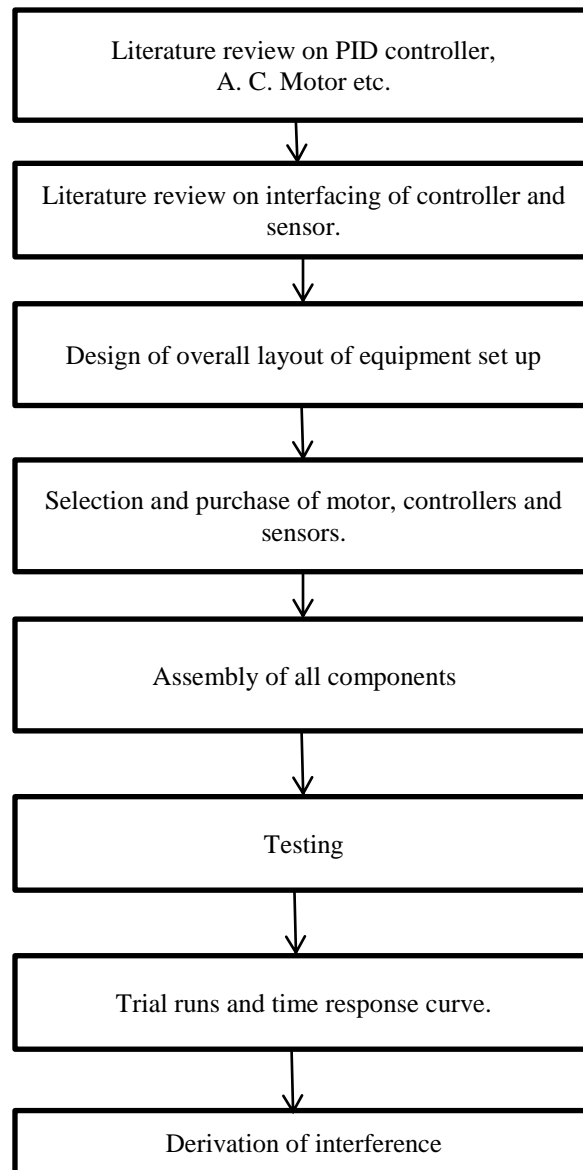
RESEARCH METHODOLOGY
The goal of the said work is to construct a model of speed control of A. C. motor using a PID controller. The setup has the following components:
- PID controller.
- AC motor driver.
- AC motor.
- LCD display.
- Knobs.
- Shaft encoder with proximity sensor

By altering the various parameters in the PID controller, we can observe different system responses to various inputs parameters of PID controllers. The feedback is taken by using microcontroller from infrared proximity sensor.

The feedback is then given to PID controller and error signal is calculated based on this feedback signal. The final output is then indicated using digital indicator. The said target is achieved by following various steps as shown in the flow chart in figure 2.


Fig 2: Research Methodology

ARDUINO BOARD

Arduino board makes the things easier as it uses simplified version of C++ and is already made microcontroller. The image of the same is as shown in figure 3. This Arduino microcontroller can be easily programmed, erased and reprogrammed. It is open source platform for constructing and programming of electronics. The board is designed and manufactured in such a way that it can receive and send the information to most of devices. The

hardware called Arduino uno is used in this kit.
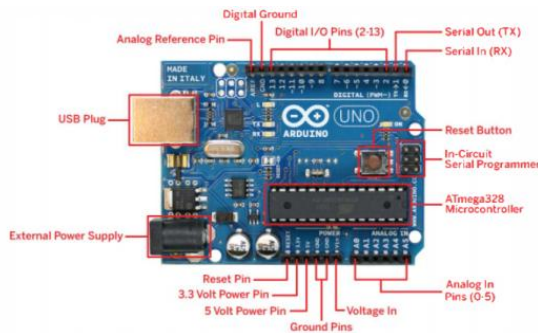
Fig 3: Arduino Uno [5]

It is user friendly hence most commonly used now a days in electronic applications which are microcontroller based[5]. The signals generated or recorded by computer is very expensive. Hence the Arduino uno kit provides low cost equipment for such applications[6].

The specifications of Arduino microcontroller kit are as follows,

ARDUINO MICROCONTROLLER
- Microcontroller ATMEG328
- Architecture AVR
- Operating Voltage 5 V
- Flash memory
- 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB
- Clock Speed 16 MHz
- Analog I/O Pins 6
- EEPROM 1 KB
- DC Current per I/O Pins
- 40 mA on I/O Pins; 50 mA on 3,3 V Pin

GENERAL
- Input Voltage 7-12 V
- Digital I/O Pins
- 20 (of which 6 provide PWM output)
- PWM Output 6
- PCB Size 53.4 x 68.6 mm
- Weight 25 g
- Product Code

LAYOUT OF SET UP
The Set up consists of total six main components. The components are PID controller, AC motor driver, AC motor, LCD display, Knobs and Shaft encoder with proximity sensor. They are connected as shown in figure 4. By varying the supplying voltage, the speed of A. C. motor can be varied. The voltage is varied until the torque required by the load is developed, at the desired speed. The torque developed is proportional to the square of the supply voltage and the current is proportional to the voltage. Hence, to reduce the speed for the same value of the same current, the value of the voltage is reduced and as a result, the torque developed by the motor is reduced. The single phase motors are controlled by a single phase Triac Voltage Controller. The terminal voltage across the stator winding can be varied for obtaining the desired speed control by controlling the fire angle of TRIAC. These controllers are known as Solid State fan regulators.
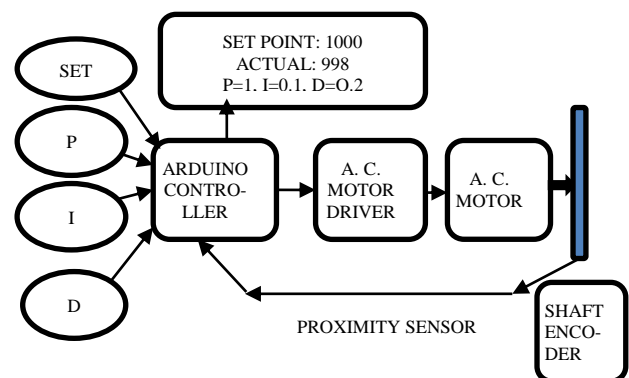

Fig 4: Layout of set up

The initial set speed is given to the motor and the motor shaft rotates as per the given speed. But due to various losses like friction, it might not run as per the set speed. To measure the motor speed, proximity sensor is used. It is a LED and photo transistor pair which generates pulses corresponding to motor speed. A slotted disk having twelve slots is fixed on the motor shaft and it cuts the path

between the LED and photo transistor when the motor rotates.

These pulses which vary from 0 - 5V are then converted to ± 5 V using an OPAMP. The pulses from optical switch are given to a Frequency to voltage converter. The Frequency to voltage converter produces a DC voltage (0-5 V) proportional to motor speed. Various readings are taken by turning the knobs of PID controller and the results are observed by plotting the graphs for various readings.

In this project, we are using an incremental rotary encoder. As shown in figure 5. It shows a typical rotary encoder used in our said work.
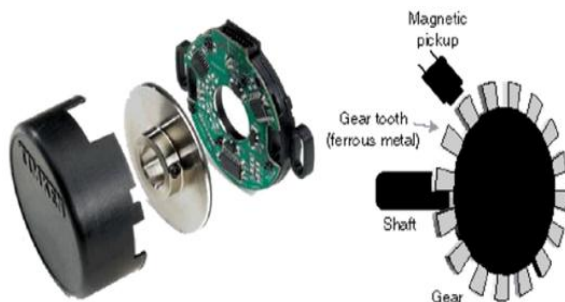
Fig 5: Rotary encoder [8]

An H-Bridge is an electronic circuit that enables a microcontroller, such as an Arduino, to control the movement of a motor. A microcontroller cannot be connected directly to a DC motor because the microcontroller cannot supply the high current required by the motor. The H-bridge contains the high-current switches needed for motor control. These switches are controlled by signals from the Arduino.

An H-bridge allows a motor to be moved forwards or backwards. The direction is determined by the sequence of control signals from the Arduino.

Motor speed is determined by setting the duty cycle of the control signals. This type of speed control is called pulse-width modulation. The motor will run at full speed when the duty cycle is 100%.

When the duty cycle is 0%, however, the motor will stop and will no longer move [8].

CODING IN ARDUINO KIT

Arduino microcontroller board and open hardware that enables the use of low-cost, application specific hardware. The board is programmed using the Arduino language, which is based on C/C++. The sample program used in the said work is shown below,

```
#include <TimerOne.h>
/*
Purpose: to detect zero crossing pulse at
 INT0 digital pin 2, which after delay
 switches on a triac.
 Power output to triac activated by external switch.
 */
#include <PID_v1.h>
#include <LiquidCrystal.h>
#define triacPulse 5
#define SW 4
#define aconLed 13
unsigned long rpm_t=0;
unsigned long alp_t=0;
unsigned zflag=0;
char sys_status=0;
int val,dislen;
volatile unsigned long rpm=0;
volatile unsigned int hits=0;
volatile int alpha;
double Speed,SetP,Output;
float consKp=2, consKi=5, consKd=1;
LiquidCrystal lcd(6,7,8,9,10,11);
PID myPID(&Speed, &Output, &SetP,
consKp,        consKi,        consKd,
REVERSE); //DIRECT
```

CONSTRUCTION OF SET UP

The setup consist of components like PID controller, AC motor driver, AC motor, LCD display, Knobs, Shaft encoder with proximity sensor. All these components are fitted on one metal box. All electronic components are fitted inside the box and control knobs along with digital display is
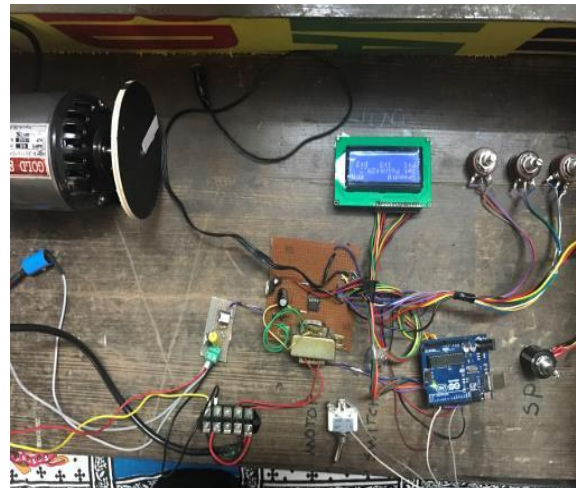
fitted on the top of box as shown in the figure 6,

Fig 6: Photograph of set up

The A. C. motor is also fitted on the top of box. The electronic components fitted inside the box can be accessed for maintenance using pivoted flap as shown in the figure 7,
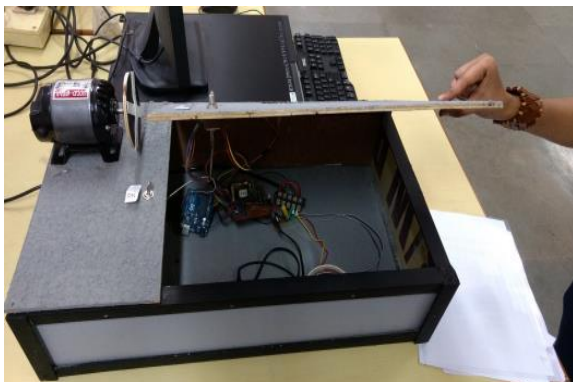
Fig 7: Pivoted flap for electronic components

The connections of various electronic components is as shown in figure 8,

Fig 8: Connections of electronic components

RESULTS
The main purpose of this work is to study the time response characteristics of system. The speed control of A. C. motor is the system selected for this study. This is achieved using PID controller and Arduino board.

If we select the set point as 1504 RPM as the speed of A. C. motor, the time response obtained for various values of PID controller were as shown in table 1.

Table 1: Response at set point 1504 RPM

Set Point	P	I	D	Rise Time	Peak Point Speed	Actual Speed
1504	5	0	0	4	1920	1440
1504	0	5	0	2	2340	1500
1504	0	0	1	1	4650	-
1504	10	5	1	1	2980	1440
1504	10	5	0	0	2540	1520
1504	10	10	0	0	2780	1560
1504	10	15	0	0	2660	1540
1504	12	20	0	0	2920	1580
1504	6	6	0	1	3000	1560
1504	3	2	1	0	3920	-
1504	15	15	1	0	4080	-

Then the selected set point was 3501 RPM as a speed of A. C. motor, then the time response obtained for various values of PID controller were as shown in the table 2,

Table 2: Response at set point 3501 RPM

Set Point	P	I	D	Rise Time	Peak Point Speed	Actual Speed
3501	8	6	0	1	4680	3480
3501	8	4	0	2	4680	3540
3501	8	9	0	1	4650	3480
3501	8	9	2	0	4200	-
3501	11	9	0	1	3900	3420
3501	6	4	0	1	4380	3620
3501	3	6	0	1	4140	3480

CONCLUSION

A closed loop control of A. C. motor is developed using PID controller and Arduino microcontroller kit. PID control was successfully implemented in Arduino microcontroller kit and tested on speed control of A. C. motor. The output values also agreed with the input values given to microcontroller.

The various time response characteristics are obtained from the set up by varying the values of PID controller and studied.

REFERENCES

1. J. C. Basilio and S. R. Matos: *'Design of PI and PID Controllers with Transient Performance Specification'*, Brazil, 2002.
2. K.J. Astrom, T. Hagglund, '*The future of PID control*', Sweden, 2001.
3. Zhen-Yu Zhao, Masayoshi Tomizuka, Satoru Isaka, 'Fuzzy Gain Scheduling of PID Controllers', 1993.
4. Serdar Ethem Hamamci, 'An Algorithm for Stabilization of Fractional-Order Time Delay Systems Using Fractional-Order PID Controllers', VOL. 52, NO. 10, OCTOBER 2007.
5. Yusuf Abdullahi Badamasi, 'The Working Principle Of An Arduino', Nigeria, 2014.
6. Alessandro D'Ausilio, 'Arduino: A low-cost multipurpose lab equipment', The Italian Institute of Technology, October 2011.
7. Rajesh Pansare, Manoj Palsodkar, '*Macros in Catia to Read Values from MS Excel - A Tool for Automation*', International Journal of Innovative and Emerging Research in Engineering Volume 3, Special Issue 1, ICSTSD 2016.
8. Tim young Blood, '*Use an Arduino to control a Motor*', June 3 2015.