Real-Time Implementation of ESP32 based Speed Control of DC Motor

Meruga Kishan Balaji[1], Kakulapati Chandra Praveen[2], Dr. Jatoth Ravi Kumar[3]

[1]Electrical and Electronics Engineering Department, National Institute of Technology, Andhra Pradesh, India.
[2]Electrical and Electronics Engineering Department, National Institute of Technology, Andhra Pradesh, India.
[3]Electronics and Communication Engineering Department, National Institute of Technology, Warangal, India.

***Abstract*:** This project aims to design and implement PID controllers for the speed control of DC shunt motors using the Blynk app on mobile devices and the ESP32 board. The objective is to develop a remote-control system that enables the user to adjust the motor speed conveniently from their mobile phones. The project begins with a comprehensive literature review, exploring exciting knowledge and PID controllers, DC motor speed control, and the use of the Blynk app for remote control applications. The system architecture is designed to facilitate the interaction between the mobile device, Blynk app, ESP32 board, and the DC shunt motor. The PID controllers are designed based on proportional, integral, and derivative terms and are carefully tuned to achieve stability and optimal performance. The obtained results are analyzed, considering performance metrics such as speed response accuracy and stability of the motor control system. The findings demonstrate the successful implementation of PID controllers for speed control validating the feasibility of using the Blynk app and ESP32 board in an IOT paste motor control application.

***Index Terms*:** PID controllers, DC Shunt motor, Blynk app, Internet of Things (IoT), ESP32 board.

1. Introduction

Over the past seven decades, speedy controllers have proven to be highly effective and widely used in a diverse range of industrial applications. The controller particularly of PI/PID types is a fundamental component of the control loop in industrial processes. They play a vital role in maintaining the side's set points, ensuring stability, and regulating critical variables in various systems.

In many industrial processes, PID controllers are favored for simplicity, reliability, and ease of implementation. The offers versatile set-top features and can be integrated into distributed control systems for allowing precise and issuing control over complex processes [1]. The Internet of the IOT is a rapidly evolving technology paradigm that has gained immense popularity in recent years. IoT refers to the interconnection of various devices and objects through the Internet enabling them to collect, exchange, and analyze data without the need for direct human intervention [2].

Industrial processes are often subject to nonlinear dynamics, unmeasured disturbances, measurement delays and lags, noises, components (e.g., valve) non-idealities and nonlinearities, and sensitivity limits and resolution.

This paper was on an IOT-based motor using PID controllers and the Blynk app via an ESP32 microcontroller. The system also allows remote monitoring and control providing precise and stable speed regulation. The ESP32 acts as the interface between the Blynk app and the motor, while PID controllers ensure efficient motor control under varying load conditions. Real-time data visualization on the Blynk app enhances user interaction. The study highlights the practical applications of this system in industrial automation, home automation, and education.

Overall, the research contributes to the advancement of IOT-based motor control and provides a foundation for future enhancements in the field.

The paper is organized as follows. Mathematical model of DC Motor is presented in Section II. In Section III the creation of Blynk platform is presented. In Section IV Implementation of PID Controllers on ESP32 is presented. Using the model obtained PID controllers parameter tuning is performed. In Section V, System Architecture is shown. Section VI summarizes the implementation procedure, and presents the experimental results/comparison using the previously tuned PID controllers. Section VII presents conclusions and future work.

2. Mathematical model of DC Motor

Direct current (DC) motors convert electrical energy into mechanical energy through the interaction of two magnetic fields. One field is produced by a magnet of poles assembly, the other field is produced by an electrical current flowing in the motor windings. These two fields result in a torque that tends to rotate the rotor.

Figure-1

$I_a = Armature\ Current$
$E_b(t) = Back\ emf\ of\ motor$
$R_a = Armature\ Resistance$
$T_L(t) = Load\ Torque$
$T_m(t) = Motor\ Torque$
$\theta_m(t) = Displacement$
$L_a = Armature\ Inductance$
$K_t = Constant\ Torque$
$E_a(t) = Voltage\ applied$
$K_b = Constant\ back\ emf$
$\varphi = Magnetic\ flux$
$\omega_m(t) = Angular\ velocity\ of\ motor$
$J_m = Rotor\ Inertia$
$B_m = Friction\ coefficient\ viscous$

$$T_m(t) = K_m(t)\varphi I_a(t)$$

$$T_m(t) = K_t I_a(t)$$

Where K_i is the constant torque of motor in N-m/A.
Starting with applied input voltage $E_a(t)$ the reason and cause of the equations for the rotating dc shunt motor circuit condition.

$$\frac{dI_a(t)}{dt} = \frac{1}{L_a}E_a(t) - \frac{R_a}{L_a}I_a(t) - \frac{1}{L_a}E_b(t)$$

$$T_m(t) = K_m I_a(t)$$

$$E_b(t) = K_b\frac{d\theta_m(t)}{dt} = K_b\omega_m(t)$$

$$\frac{d^2\theta_m(t)}{dt} = \frac{1}{J_m}T_m(t) - \frac{1}{J_m}T_l(t) - \frac{B_m}{J_m}\frac{d\theta_m(t)}{dt}$$

Where $T_l(t)$ is noticed as a frictional torque for a load applied which is as coulomb friction.

$$\frac{\theta_m(S)}{E_a(S)} = \frac{K_t}{L_a J_m S^3 + (R_a J_m + B_m L_a)S^2 + (K_b K_t + R_a B_m)S}$$

Here, we are taking speed as output. So, we are changing displacement of motor (θ_m) into speed (ω_m).

$$\omega_m(t) = \frac{d\theta_m(t)}{dt}$$

$$\frac{\omega_m(S)}{E_a(S)} = \frac{K_t}{L_a J_m S^2 + (R_a J_m + B_m L_a)S + (K_b K_t + R_a B_m)}$$

As we are giving PWM signal here, the input should be percentage duty cycle of the PWM signal, let it be ‘x’.
If V_t = Load voltage, then

$$E_a = V_t\frac{x}{255}$$

Then the required transfer function is

$$\frac{\omega_m(S)}{x(S)} = \frac{V_t}{255}\left(\frac{K_t}{L_a J_m S^2 + (R_a J_m + B_m L_a)S + (K_b K_t + R_a B_m)}\right)$$

Where,

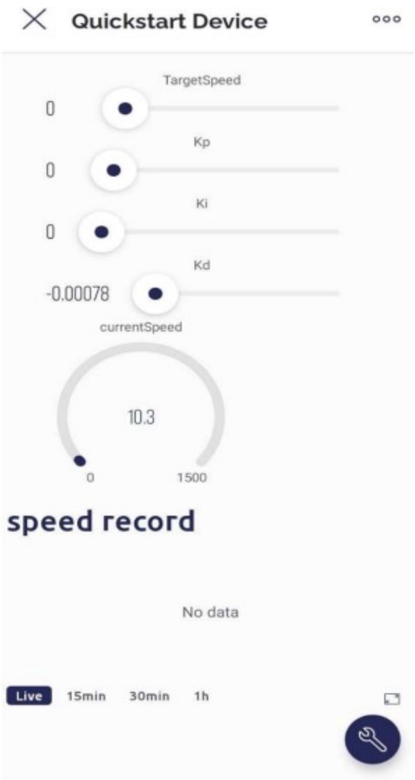
$R_a = 1.86\ \Omega$
$L_a = 1.1295 * 10^{-4}\ H$
$J_m = 8.241 * 10^{-4}\ Kg.m^2$
$B_m = 3.82 * 10^{-3}\ m.\frac{s}{rad}$
$K_t = 0.085711\ N.\frac{m}{A}$
$K_b = 0.085711\ V.s$
$V_t = 230\ V$

The above values of the DC Motor are identified from [5] and [6].

$$\frac{\omega_m(S)}{x(S)} = \left(\frac{773.053}{(9.304 * 10^{-4})S^2 + (15.332)S + 144.512}\right)$$

3. Creating Blynk app

Blynk is an Internet of Things (IoT) platform that provides a mobile app for controlling and monitoring connected the violet remotely. It offers a simple and user-friendly interface, making it easy for developers, and professionals to build IoT applications without extensive programming knowledge Blynk is designed to connect hardware devices, such as ESP32, Arduino, Raspberry Pi, ESP8266 and others, to the Blynk Cloud. The mobile app can act as the control panel, allowing users to interact with their connected devices.



Mobile Dashboard

Figure-2 This is the Blynk app created

4. Implementation of PID controller on ESP32

PID controllers are probably the most widely used industrial controllers. In PID controller proportional(P) control is not to remove steady state error or offset error in step response. This offset can be earned by integral (I) control action. Integral control removes offset but may lead to oscillatory response to slowly decreasing amplitude or even increasing amplitude, both of which are error, initiate any early correction action and tends to increase stability of system.

The time domain is given by,

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de(t)}{dt}$$

By taking Laplace Transform on both sides,

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

Ziegler and Nichols proposed a rule for determining values of Kp, Kd, Ki based on the transient response characteristics of given plant. The closed loop of special based PID is a popular method of tuning the PID controller in this kind of learning critical gain Kc, is induced in the forward path of control system. The high value of gain takes the system to the verge of instability. It creates oscillation and from the oscillations, the value of frequency and time are calculated.

Case-1:

We have determined the motor ' s transfer function theoretically as shown in the above block diagram.
The Motor Transfer function is,

$$\frac{\omega_m(S)}{x(S)} = \left(\frac{773.053}{(9.304 * 10^{-4})S^2 + (15.332)S + 144.512}\right)$$

	K_p	K_i	K_d
Frequency Response Based	0	0.18223	0
Transfer function Based	0.247120474909358	4.68006121376728	-0.00086812

Table-1

Case-2:

We have directly given the motor model to the setup.

	K_p	K_i	K_d
Frequency Response Based	0	0.1642	0
Transfer function Based	0.222860796990413	4.22036935859341	-0.0007834

Table-2

Usually, initial design values of PID controller obtained need to be adjusted repeatedly through computer simulations until the closed loop systems performs are comprises as desired. These adjustments are done in Matlab simulation.[3]

The following is the dc motor block diagram prepared in Matlab Simulink,

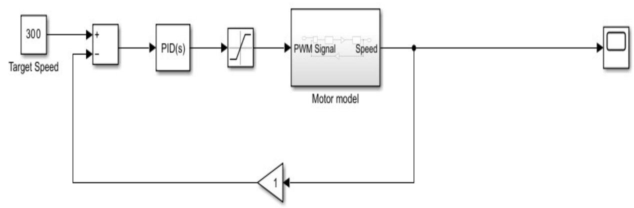


Figure-3

The Motor model subsystem that mentioned below is, given below

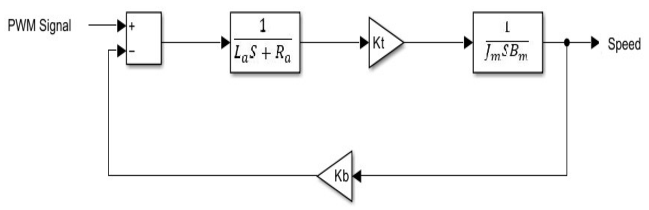


Figure-4

5. Proposed Architecture

Here is the ESP32 access controller. It consists of microcontroller in which the program is written and debug into it to control the speed of DC Shunt motor.
The experimental setup consists of

- DC Shunt motor
- ESP32 WROOM DEVKIT V1
- Speed Encoder (Autonics E51S)
- IGBT based DC Motor Driver
- NPN transistor (BC547)
- Resistor(1KΩ,10KΩ)
- Jumper wires
- 230V variable DC voltage supply

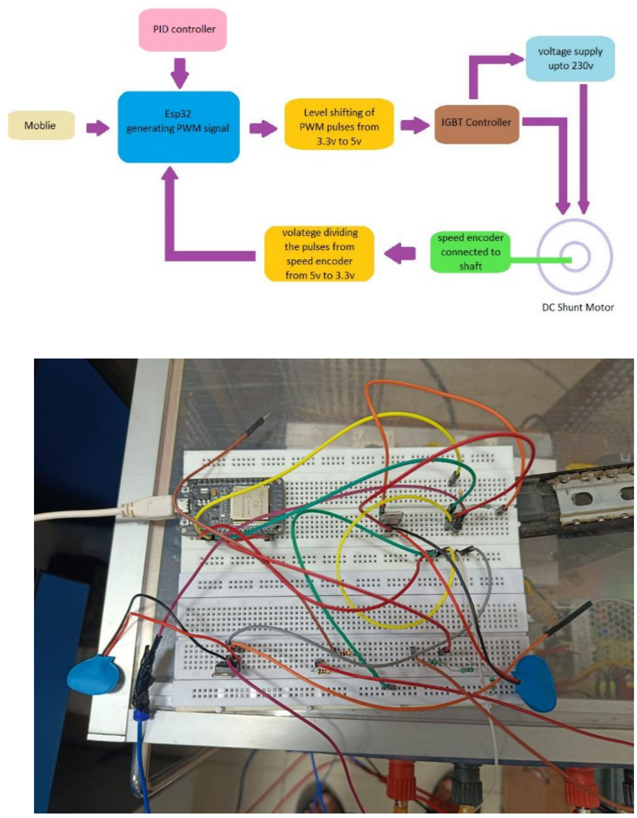


Figure-5 Proposed Architecture

6. Experimental Setup

When the setup is switched ON, and the Target speed is given through the mobile using Blynk app to DC Shunt motor, error is produced. The error is given to the PID controllers then the PID output is generated which is in the range (0- 255). This PID output is used as percentage duty cycle of the PWM signal by ESP32 and generates PWM signal corresponding to PID output. This process will be continued in a loop so that the motor will rotate at the desired speed.

Here, Level Shifting is done on PWM signal generated by ESP32. This is because ESP32 can generate PWM signals up to 3.3V. But IGBT based DC motor driver requires 5V amplitude PWM signals.

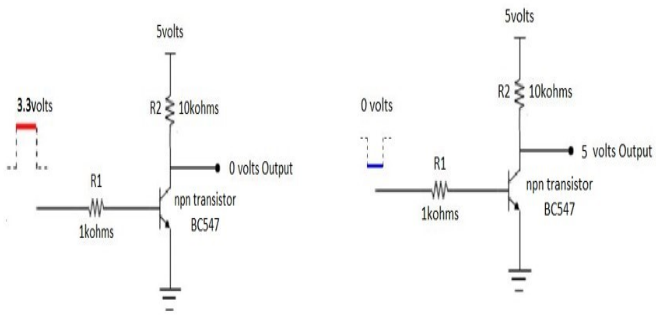


Figure-6 Level Shifting circuit

To interface the speed encoder with ESP32, a voltage divider is employed to convert the 5V pulses from the encoder to a compatible 3.3V level, ensuring safe and accurate signal communication.

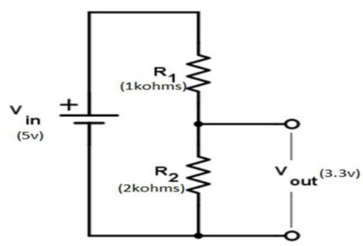


Figure-7 Voltage divider

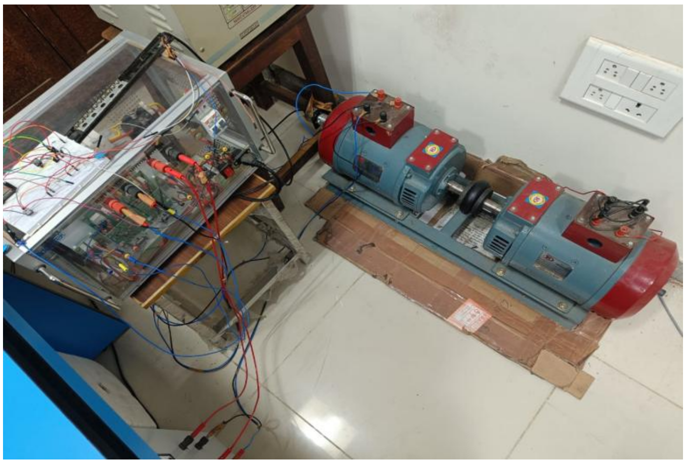


Figure-8

7. Results and Discussion

Case-1:

Graph tuning using Frequency response based.

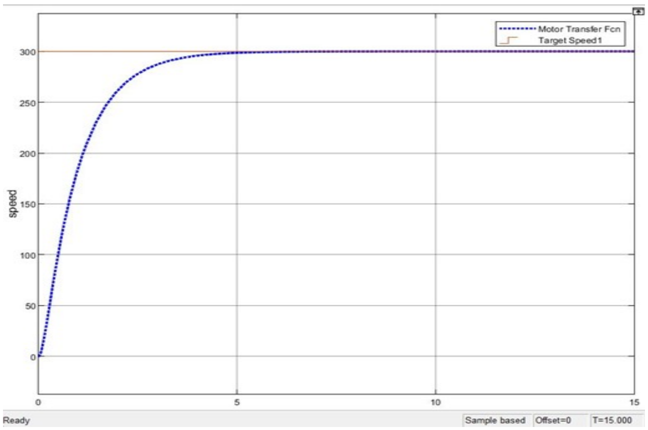


Figure-9

struct with fields:

RiseTime: 2.0263
TransientTime: 3.6858
SettlingTime: 3.6858
SettlingMin: 276.4440
SettlingMax: 300.0000
Overshoot: 0
Undershoot: 0
Peak: 300.0000
PeakTime: 15

Graph tuning using Transfer function based method.

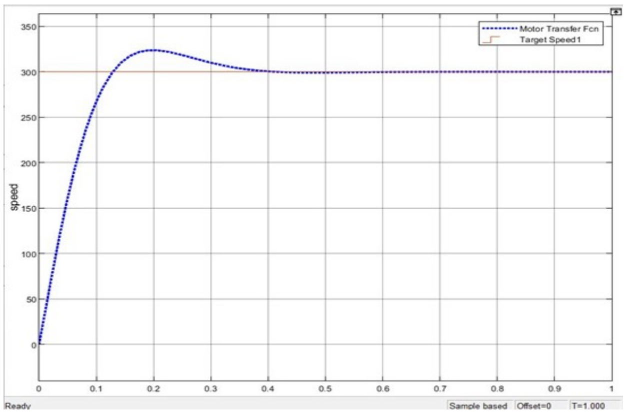


Figure-10

struct with fields:

RiseTime: 0.0931
TransientTime: 0.3298
SettlingTime: 0.3298
SettlingMin: 270.6190
SettlingMax: 323.7059
Overshoot: 7.9020
Undershoot: 0
Peak: 323.7059
PeakTime: 0.2048

Case-2:

Graph tuning using Frequency response based.

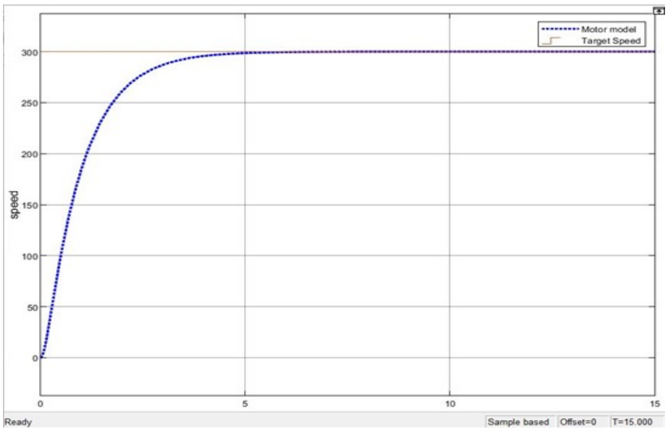


Figure-11

struct with fields:

RiseTime: 2.0286
TransientTime: 3.6900
SettlingTime: 3.6900
SettlingMin: 276.3807
SettlingMax: 300.0000
Overshoot: 0
Undershoot: 0
Peak: 300.0000
PeakTime: 15

Graph tuning using Transfer function based method.

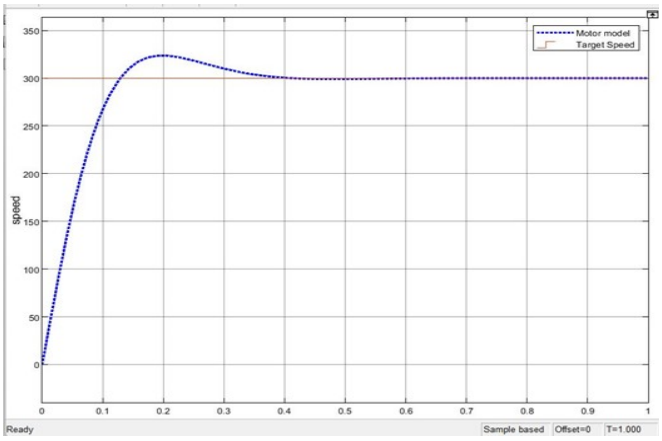


Figure-12

struct with fields:

RiseTime: 0.0931
TransientTime: 0.3299
SettlingTime: 0.3299
SettlingMin: 270.5987
SettlingMax: 323.7042
Overshoot: 7.9014
Undershoot: 0
Peak: 323.7042
PeakTime: 0.2048

8. Conclusion

"Real-Time Implementation of ESP32 based Speed Control of DC Motor" has successfully achieved its objectives of developing an innovative and efficient motor control system.

Throughout the project, we designed and implemented PID controllers to regulate the speed of a DC motor accurately and dynamically. Leveraging the capabilities of the Blynk app, operators were empowered with remote control functionality, allowing them to make real-time adjustments to the motor's speed using their smartphones or tablets. The user-friendly interface provided by Blynk enabled operators to monitor the motor's performance and respond promptly to changes in operational requirements.

In conclusion, the "Real-Time Implementation of ESP32 based Speed Control of DC Motor" project has achieved its objectives of developing a cutting-edge motor control system that exemplifies efficiency, safety, and real-time control capabilities. The integration of IoT technology and PID controllers, combined with the ESP32 board and Blynk app, has brought forth an innovative solution with promising applications in various industries. As technology continues to evolve, this project serves as a stepping stone towards a smarter, connected future, where remote control and automation redefine the boundaries of industrial efficiency and safety.

9. References

[1] C.1.K.H.T. Astorm, Advanced PID Control ISA, Research Triangle Park, 2006.

[2] Ashton, K. (2009). "That 'Internet of Things' Thing." RFID Journal, 22(7), 97-114.

[3] Md Akram Ahmad Speed control of a DC motor using Controllers,2014.

[4] Jailsingh Bhookya, Implementation of PID controller for liquid level system using mGWO and integration of IoT application, 2022.

[5] Bustanual Arifin, Review of Method for System Identification on Motors, 2021.

[6] Darshan Ramasubramanian, Identification and control of DC motor.

[7] Isiyaku Saleh, Speed Control of a BLDC Motor Using Artificial Neural Network with ESP32 Microcontroller Based Implementation, 2022.

[8] Mohamed Zied Chaari, Comparison of Blynk IoT and ESP Rainmaker on ESP32 as beginner-friendly IoT solutions,2023.

[9] Amarapini Divya and Dr.Prasadarao Bobbili, Comparison and Speed Control of DC Motor and DC Servomotor Using IMC Based PID Controller, 2021.

[10] Krismadinata Krismadinata, PI and PID Controller Design and Analysis for DC Shunt Motor Speed Control, 2020.