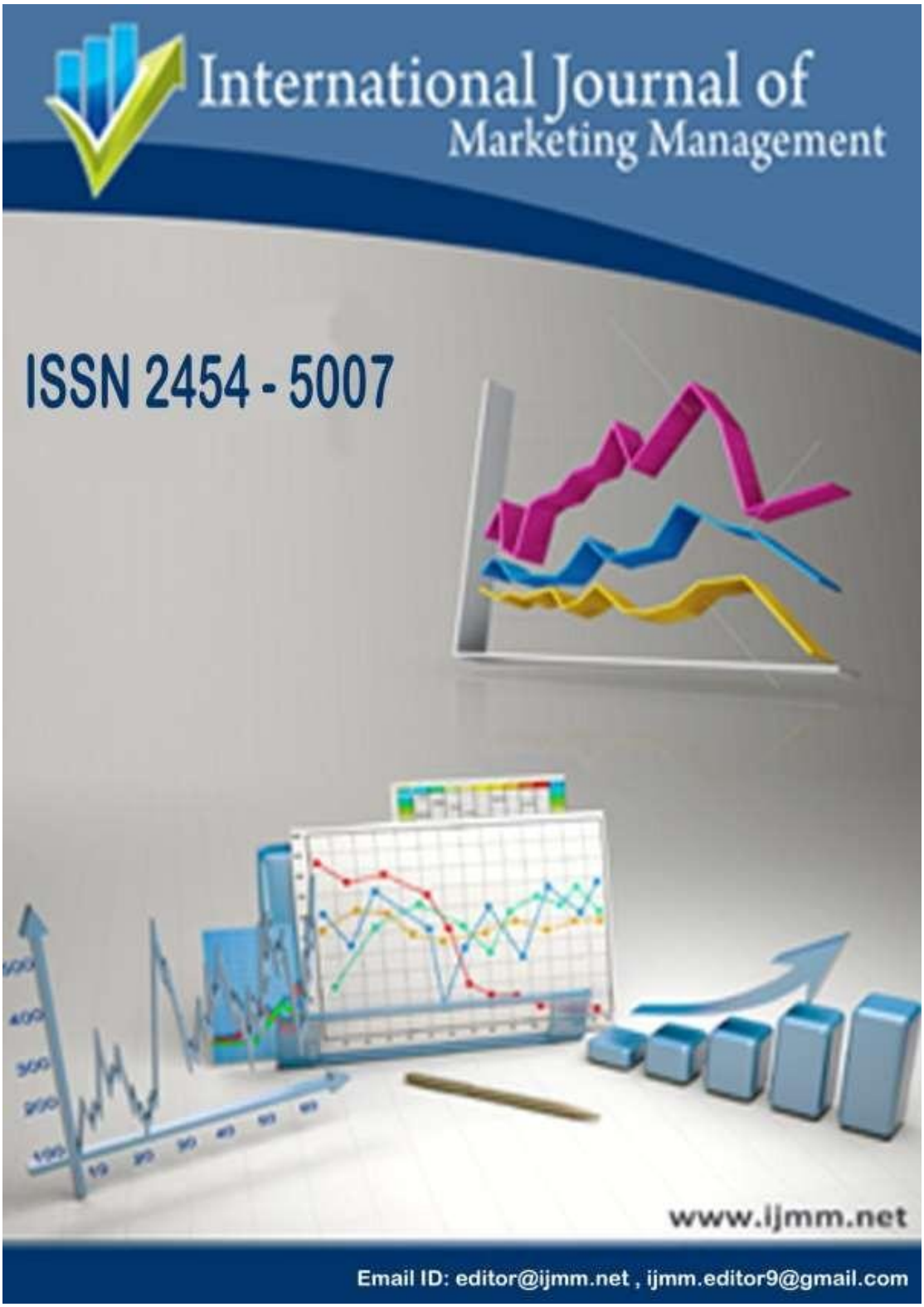

International Journal of
Marketing Management
ISSN 2454 - 5007
www.ijmm.net
Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

Using a Remote Closed-loop Control System to Assess the Performance of an Embedded Speed Control System for a DC Motor

[1] Mr.V.Ramudu,[2] Cherala Saketh,[3] Dommata Madhu,[4] Sujeeth Kumar Yadav,[5] Kancha Raj Kumar,

[1]Assistant Professor, Department of EEE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

[2,3,4,5.] Student, Department of EEE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

***Abstract*—** The purpose of this article is to assess how well an embedded platform for a remote access virtual laboratory functions. In this lab, we'll be focusing on the standard issue of DC motor speed regulation. A single-board computer running a control algorithm based on Python is linked to the embedded platform via USB protocol, and it may be accessed remotely from numerous sources.
The embedded platform is thoroughly analyzed in this article, which also includes experimental findings and analysis. The following terms are used in the index: embedded platform, remote access laboratory, DC motor, speed control, and Python.

I. INTRODUCTION

A student's understanding of engineering theory and practice is well grounded in the engineering laboratory courses. The physical installations that make up an engineering laboratory may show the working principles of any system in various modes or at different operating points. Traditional hands-on labs demand a lot of room, specialized equipment, and trained personnel to run well. The most significant drawback of traditional hands-on labs is that students at a single school have exclusive use to the best facilities and equipment. For many students, the traditional hands-on laboratory isn't conducive to gaining a thorough understanding of the physical system. The majority of standalone systems are now online due to advancements in powerful computing facilities, specialized software, networking protocols, and communication protocols with increased bandwidth. There has been a shift away from traditional laboratories and toward VRLs, or virtual and remote labs. Access to both real and virtual resources, both locally and remotely, is one of the key benefits of VRLs [1]. The design of a virtual laboratory's hardware and software has been covered in [2]. Detailed in [3] are the learning results of both traditional VRLs and traditional hands-on labs. Due to the presence of a network connecting the controller to the physical process, VRLs operate on a network control system. The literature discusses web-based network control laboratories [4] and time-sharing remote access systems for VRLs [5]. Among the most fundamental issues in mechatronics, control systems, and electromechanical systems is the DC motor. Setting a constant speed for the DC motor is the primary objective of the benchmark challenge. A DC motor, load, speed sensors, and an embedded control system make up the experimental setup for controlling the speed of the motor. Embedded platforms for speed control of DC motors have been created by several researchers in the control literature. These platforms include dsPIC30F4012, MCP2551, ARM, and FPGA. In [10], the authors propose a didatic platform that allows for discrete control of the speed and position of a DC motor. There are VRLs for DC motor speed control in the literature in addition to the more traditional, hands-on laboratory methods. Using the ARM, TINA, and STM32 microcontrollers, VRLs are created to regulate the speed of DC motors [11–13]. In addition to microcontrollers, embedded FPGA controls for DC motor speed control have been developed in [14], [15]. A DC motor with a reconfigurable microarchitecture and a dynamic pipeline system was created by the authors of [16]. This study makes use of a control technique that is based on computation at the edge. In [17], the difficulties of integrating a PID controller into a digital environment are detailed. There are other evaluations of sensorless speed control of DC motors in the literature. In this case, the speed estimator is utilized to estimate the motor's speed using armature voltage and armature current instead of a speed sensor [18]. You may find the FPGA code for controlling the DC motor's speed without using any sensors in [19]. Programming

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

languages used to build the embedded control platform range from low-level languages like Assembly or C to high-level ones like MATLAB or LabVIEW. These days, beginners in programming are often advised to start with an open-source language like Python as their software/programming environment of choice for developing control algorithms. Writing programs specifically for a microcontroller is a lot slower than developing open source programming. Because of the complexity of controlling the resources available to a microcontroller, building a control algorithm on one could seem daunting to novice users. The authors suggest a hybrid system in which the control algorithm is written in high-level language and the Arduino microcontroller is used to communicate with the real hardware. Quick events, such interrupts and speed sensor readings, are handled by the microcontroller. The Universal Serial Bus (USB) interface is the means by which Python and Arduino communicate with one another. This article details the process of creating, deploying, and evaluating an embedded control system for the purpose of studying DC motor speed control strategies. Hardware that measures the DC motor's speed is best accomplished using the widely used Arduino microcontroller boards. To regulate the motor, a general-purpose computer is used. Python or another machine-independent interpreter-based language is used to implement the controller. In addition, a local web server makes the Python integrated development environment (Jupyter Notebook) available for remote access. You may access the embedded control system over the local area network or the internet, depending on how you set up port forwarding. In this article, the following structure is used. In Section II, we detail the system, including all of the hardware platform information. The DC motor's mathematical analysis is presented in Section III. The examination of the controller is presented in Section IV, while the findings and analysis are presented in Section V. The last thoughts are presented in Section VI.

II. SYSTEM DESCRIPTION

The embedded platform of the DC motor control setup is shown in Fig. 1, which is an overall block diagram. The hardware platform is linked to the PC using a USB connection. The Python kernel, which is executed on this PC, is responsible for communicating with the Arduino microcontroller in the hardware arrangement. On top of that, the computer is running a Jupyter Server, which provides a web interface for accessing the Python Kernel. Opening a new application in the Jupyter IDE really allows the user to create a new instance of the Python kernel. You may use this Jupyter IDE on any machine on your local area network. But there are a number of methods to set up the Jupyter server so that it can be accessible from outside the local network, which is useful for remote work. It is possible to set up the Jupyter server to listen to a publicly accessible static IP address on a PC. An extra layer of user-friendliness may be achieved by configuring a domain name with a static IP. The Jupyter server is accessible directly from any remote user's to use the Jupyter IDE on their machines after successfully authenticating. • A Dynamic DNS service may be used when a Static IP is not available; it monitors the PC's Dynamic IP that is linked to the hardware and controls the connection to the PC using a set domain or subdomain name. By configuring a Virtual Private Network (VPN), individuals in different locations may virtually access the same network via their internet connections. In this way, people located outside of the physical location of the organization may access the computer that is part of the experiment as if it were part of the local network.

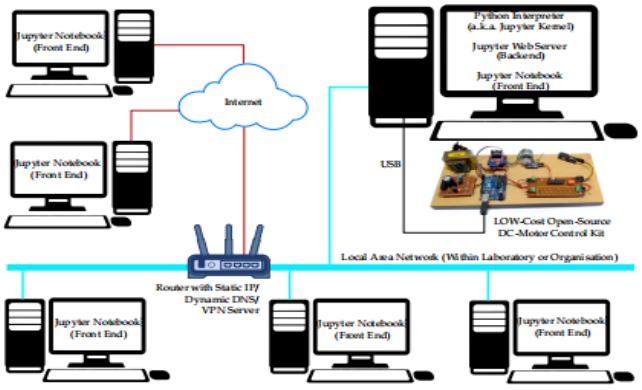


Fig. 1. Block Diagram of the Embedded Closed-Loop Set-up for remote access of the dc-motor control platform

The hardware platform that was constructed is shown in Figure 3, and Figure 2 shows the block design of the DC motor control arrangement. It is clear from the hardware configuration that two microcontrollers are used. The primary microcontroller, or Arduino Uno, and the secondary microcontroller, or bare-bone Arduino, shall be called such from now on. With just the bare minimum—a 16 MHz crystal oscillator, two 16 pF paper/ceramic capacitors, a reset button, and two LEDs (one for power and one for testing)—soldering the microcontroller IC onto a general purpose breadboard—the bare-bone Arduino is the most environmentally friendly way to build an Arduino Uno development board. The primary microcontroller generates pulse width modulation

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

(PWM) signals for speed control of the direct current (DC) motor and communicates with the personal computer. The output of the microcontroller is used by the dual H-bridge motor driver (L293D) to produce the signals that control the speed and direction of the DC motor. The L293D can provide driving currents of up to 1 A over a voltage range of 4.5 V to 36 V. By adjusting the pulse width modulation (PWM) signal, the DC motor's speed may be modulated. To get an analog signal, the auxiliary microcontroller measures the speed of the DC motor. How quickly

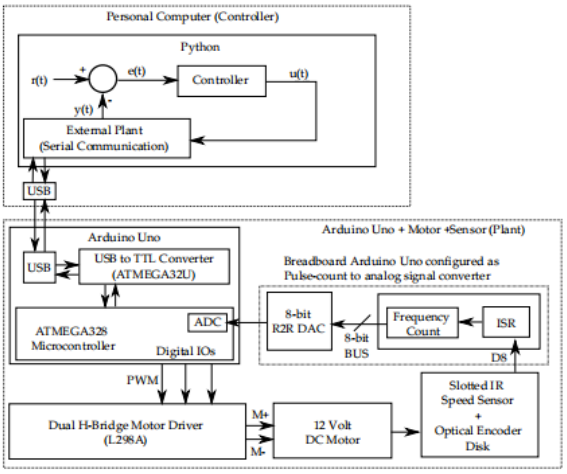


Fig. 2. Block diagram schematic for closed loop control of DC Motor speed using Arduino and Python

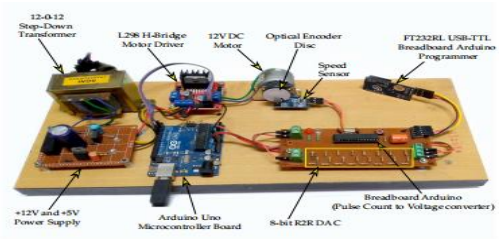


Fig. 3. Hardware setup for Speed control of a DC Motor using Python and Arduino

a magnetic encoder or an optical encoder disk fastened to the shaft of a DC motor is used to measure the DC motor. Since optical encoders are less expensive than magnetic ones, they are taken into consideration in this work [20]. When a DC motor is turned on, a slotted infrared optical sensor in an optical encoder disk produces pulses. No matter what the speed encoders you choose, the ratio of the produced pulses per second to the motor's speed remains constant. The motor speed may be determined by frequency-based speed measurement with an optical encoder by

$$\omega_m = \frac{2\pi}{T}\frac{n}{N} \qquad (1)$$

for where T is the specified fixed time, N is the number of pulses in one rotation, and n is the number of pulses counted. However, low-speed applications are not a good fit for this technology. When measuring low speeds, the technique that is based on periods is used [21], [22]. One method of measuring speed that relies on frequency is the frequency counter. Interrupts are used to implement the frequency counter in microcontroller programming. The microcontroller software has a subprogram or function called an Interrupt Service Routine (ISR) that runs whenever the monitored pin receives an interrupt. The speed sensor's pulses are linked to the pin here. The Arduino's capacity to communicate with the PC was limited while the DC motor was operating at high speeds due to the excessive amount of interruptions generated by the speed sensor's pulse output. The primary microcontroller is responsible for reading the speed sensor's readings, while the secondary microcontroller is in charge of actually computing the speed. An 8-bit number between zero and two hundred and fifty is assigned the frequency of the pulses by the auxiliary microcontroller. An 8-bit R-2R ladder-based DAC receives this 8-bit value. The DAC converts the frequency measurement into an analog signal, the voltage of which may range from zero to five volts.

III. MODELING AND ANALYSIS

In Figure 4, we can see the schematic of the permanent magnet DC motor. The model is run on the premise that the magnetic field is constant, the system is linear, and the susceptibility is also constant. Figure 4 shows the electrical parameters of an armature, including its resistance (ra), inductance (La), electromagnetic torque (Te), back EMF, and torque constant (ka). Since the permanent magnet's susceptibility is constant, the field it creates is also constant. Consequently, ka remains constant. Both the load torque (TL) and the moment of inertia (J) are defined. Both the armature voltage (ua) and current (ia) are defined. The rotational speed of the motor is ör.

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

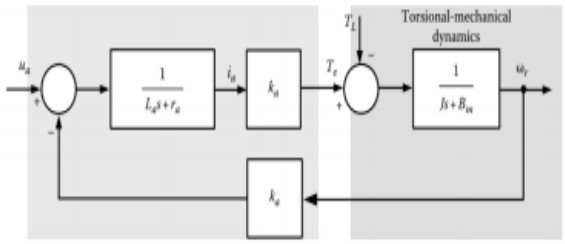


Fig. 4. Block diagram of permanent magnet DC motor

A DC motor's dynamic performance may be described by a linear differential equation, which is

$$\frac{di_a}{dt} = -\frac{r_a}{L_a} i_a - \frac{k_a}{L_a}\omega_r + \frac{1}{L_a}u_a$$
$$\frac{d\omega_r}{dt} = \frac{k_a}{J} i_a - \frac{B_m}{J}\omega_r - \frac{1}{J}T_L$$

The motor manufacturer's datasheet contains the motor's parameters. To get the approximated parameters, system identification may be used in cases when the correct parameters are not available.

IV. CONTROL LOOP

Figure 5 shows the plant's closed-loop control mechanism as a simplified block diagram. By using an optical encoder and the main microcontroller, the motor's speed may be detected. on order to provide the processed speed data to the controller running on a personal computer, the secondary microcontroller takes data from the primary microcontroller. A primary microcontroller and an H-bridge motor driver are used to transmit the controller's actuation signal to the DC motor. Deadzone and saturation are two examples of the many nonlinearities present in a physical plant. Within a certain low voltage range, called the deadzone, the motor stops rotating. Maintaining the control signal within the saturation limitations is of the utmost importance. Signal reconstruction using a digital-to-analog converter makes advantage of the zero-order hold (ZOH).

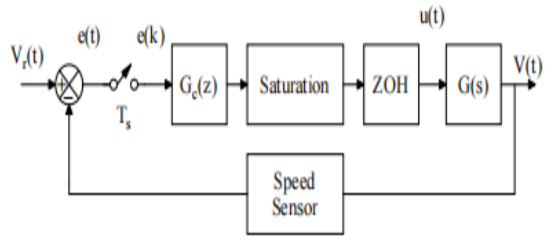


Fig. 5. Block diagram of closed-loop control of plant

When expressed in continuous form, the PID controller may be

$$u(t) = K\left[e(t) + \frac{1}{\tau_i}\int_0^t e(t)\,dt + \tau_d\frac{d}{dt}e(t)\right] \qquad (3)$$

The error signal e (t) and the controller's output u (t) are represented in Equation (3), where K is the gain, τi is the integral time constant, and τd is the derivative time constant.
One way to describe the PID controller's laplace transform is

$$G_c(s) = \frac{u(s)}{e(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \qquad (4)$$

The controller's continuous transfer function is discretized using the Backward Euler form for derivatives and the Trapezoidal approach for integration. A PID controller's discrete representation is

$$G_c(z) = \frac{u(z)}{e(z)} = \frac{az^2 + bz + c}{z(z-1)} \qquad (5)$$

where,

$$a = K_p + \frac{K_i T_s}{2} + \frac{K_d}{T_s}$$
$$b = -K_p + \frac{K_i T_s}{2} - \frac{2K_d}{T_s} \qquad (6)$$
$$c = \frac{K_d}{T_s}$$

The open-loop transfer function is

$$G_{ol}(z) = G_c(z)G(z) = G_c(z)\,z\left(\frac{1 - e^{-sT_s}}{s}G(s)\right) \qquad (7)$$

Optimal controller gains are computed using the Ziegler-Nichols tuning criteria. In order to get the PID controller settings, the quarter decay ratio approach is used. Ziegler-Nichols tuning with a quarter decay ratio yields values of Kp=0.5K, Ti=0.5T, and Td=0.125T. The DC motor's speed may be detected by the speed sensor and sent to the microcontroller. The PWM signal, which may have varying duty cycles, is generated by the Arduino microcontroller. The primary control loop is handled by the Python program. Through USB serial connection, it takes readings of the DC motor speed from the Arduino Uno microcontroller, calculates the control signal, and sends the corresponding duty cycles back to the microcontroller. You may see the flowchart in Figure 6.

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

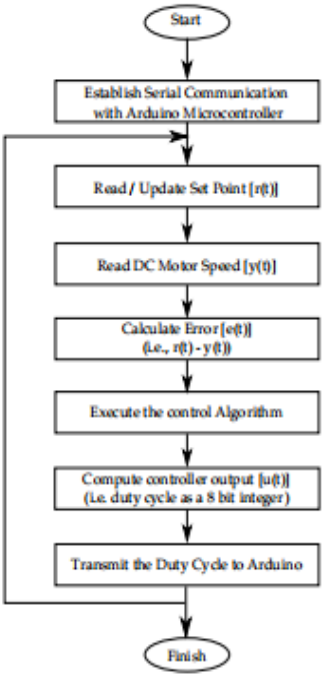


Fig. 6. Flow chart for DC Motor Speed Control in Pyt

V. EXPERIMENTAL VERIFICATION

A. System Identification

Observing the DC motor's static characteristics, such as its no-load speed which is independent of the applied voltage, is the first step in experimental verification in order to determine the transfer function.

TABLE I
STATIC CHARACTERISTICS OF DC MOTOR

Voltage (V)	Current (A)	No Load Speed (RPM)
4	0.013	16
5	0.014	21
6	0.014	28
7	0.016	36
8	0.017	42
9	0.019	47
10	0.02	52
11	0.022	59
12	0.024	64

By varying the PWM (8 bit) (0 to 255) in ascending sequence and measuring the corresponding speed (RPM), the static characteristics of the DC motor are illustrated in Fig. 7. The no-load speed (RPM) is measured by varying the DC motor's voltage and current (Table I). The values of process gain K, process time constant τ, and dead time L are determined from the steady-state analysis. Instead of a linearly rising voltage, a consistently stimulating pseudo-random binary sequence (PRBS) signal is fed into the DC motor in order to discover its intrinsic nonlinear dynamics. A serial-in parallel-out (SIPO) shift register (74164) and an XOR gate are used to create the PRBS signal.

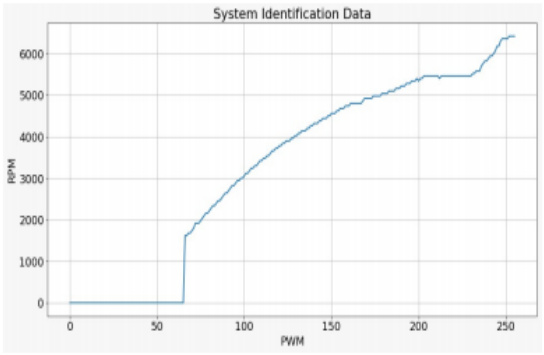
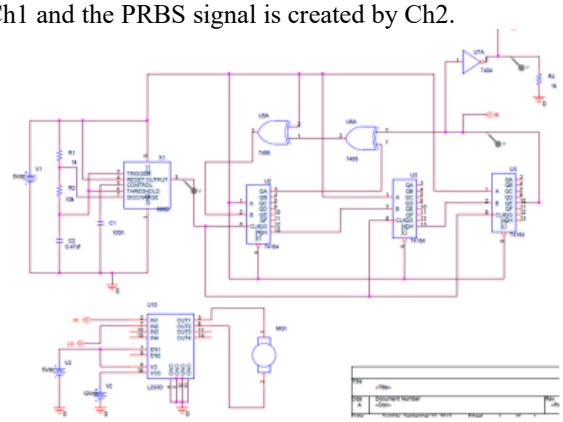


Fig. 7. The static characteristics of DC motor

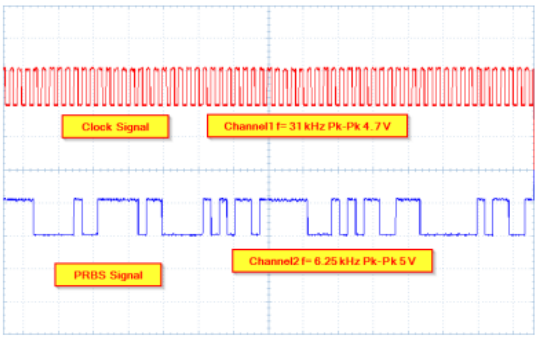
Figure 8(a). The PRBS circuit's response is shown in Figure 8(b), where the clock signal is supplied by Ch1 and the PRBS signal is created by Ch2.



(a)

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026



(b)

Fig. 8. System Identification using PRBS (a) PRBS circuit with motor driver L293D, (b) Ch-1 Clock signal (Red), Ch-2 PRBS Signal (Blue)

The system identification allows us to estimate the DC motor's transfer function as

$$G(s) = \frac{1832}{s^2 + 15.5s + 127} \quad (8)$$

B. Controller Performance Evaluation

The system is tested experimentally by measuring the embedded platform's reaction in regulatory performance and servo performance. One half second is the controller's sampling time. In this study, we evaluate the performance using two distinct signals, (i.e. Signal that is pulsed • A random step signal Figure 9 displays the DC motor's reaction to a square wave, specifically a duty cycle variation of 40% to 80%. A square wave signal with a duration of three seconds was used for testing purposes. The results demonstrate that, during the transition from low to high rpm, the DC motor's reaction is delayed relative to the input change.

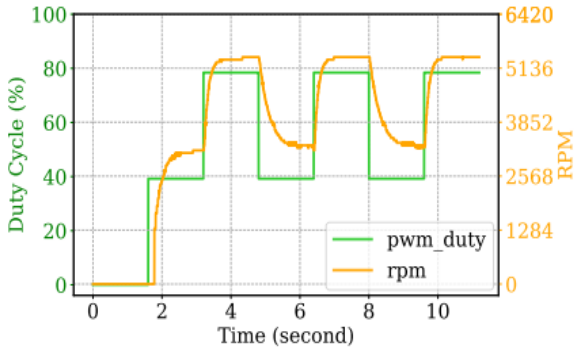


Fig. 9. DC Motor Speed Response for a pulse signal

Figure 10 displays the DC motor's reaction to a randomly generated step signal. During the transition from high to low steps, the graph clearly shows that the dc-motor damps.

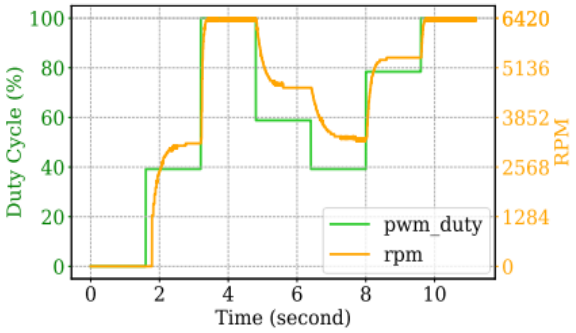


Fig. 10. DC Motor Speed Response for an arbitrary step signal

VI. CONCLUSION

An in-depth examination of the design, build, and testing of the embedded platform for DC motor speed control is presented in this work. The embedded platform combines a microcontroller-based setup with a Python-based controller, linked via the USB protocol, to provide a remote access facility. For this experiment, we've created the necessary apparatus, including a 12 V DC motor. The DC motor's reaction to various signals is now available. Multiple users may access the physical system using the remote access capability and operate on distant desktops.

REFERENCES

[1] R. Heradio, L. De La Torre, D. Galan, F. J. Cabrerizo, E. HerreraViedma, and S. Dormido, "Virtual and remote labs in education: A bibliometric analysis," *Computers & Education*, vol. 98, pp. 14–38, 2016.

[2] D. Grimaldi and S. Rapuano, "Hardware and software to design virtual laboratory for education in instrumentation and measurement," *Measurement*, vol. 42, no. 4, pp. 485–493, 2009.

[3] J. R. Brinson, "Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research," *Computers & Education*, vol. 87, pp. 218–237, 2015.

[4] J. Cavalcanti, L. F. Figueredo, J. Y. Ishihara, M. C. Bernardes, P. H. Santana, A. N. Vargas, and G. A. Borges, "A real-time web-based networked control system education platform," *The International Journal of Electrical Engineering & Education*, vol. 55, no. 2, pp. 130–141, 2018.

[5] L. Costas-Perez, D. Lago, J. Farina, and J. J. Rodriguez-Andina, "Opti- mization of an industrial sensor and data acquisition laboratory through time sharing and remote access," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2397–2404, 2008.

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

[6] R. Mondal, A. Mukhopadhyay, and D. Basak, "Embedded system of dc motor closed loop speed control based on 8051 microcontroller," *Procedia Technology*, vol. 10, pp. 840–848, 2013.
[7] M. Gunasekaran and R. Potluri, "Low-cost undergraduate control systems experiments using microcontroller-based control of a dc motor," *IEEE Transactions on Education*, vol. 55, no. 4, pp. 508–516, 2012.
[8] H. Wu, X. Chen, and L. Hu, "Embedded system of dc motor speed control based on arm," in *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 2. IEEE, 2008, pp. 123–126.
[9] E. Guzman-Ramirez, I. Garcia, E. Guerrero, and C. Pacheco, "An educational tool for designing dc motor control systems through fpgabased experimentation," *International Journal of Electrical Engineering Education*, vol. 52, no. 1, pp. 22–38, 2015.
[10] T. P. Cabre, A. S. Vela, M. T. Ribes, J. M. Blanc, J. R. Pablo, and F. C. ´ Sancho, "Didactic platform for dc motor speed and position control in z-plane," *ISA transactions*, vol. 118, pp. 116–132, 2021.
[11] R. Ganganath, C. Ranganath, and D. C. Jayasekara, "Closed-loop dc motor embedded control platform based on arm® for distance learning experiments," in *2022 7th International Conference on Information Technology Research (ICITR)*. IEEE, 2022, pp. 1–6.
[12] H. Wong and V. Kapila, "Internet-based remote control of a dc motor using an embedded ethernet microcontroller," in *Proceedings of the ASEE 2004 Annual Conference and Exposition,"Engineering Researchs New Heights*, vol. 9, 2004, pp. 8199–8214.
[13] P. Jacko, M. Beres, I. Kov ˇ a´cov ˇ a, J. Moln ´ ar, T. Vince, J. Dziak, B. Fecko, ´ S. Gans, and D. Kov ˇ a´c, "Remote iot education laboratory for microcon- ˇ trollers based on the stm32 chips," *Sensors*, vol. 22, no. 4, p. 1440, 2022.
[14] A. J. Chandra and C. Venugopal, "Novel design solutions for remote access, acquire and control of laboratory experiments on dc machines," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 349–357, 2011.
[15] C. I. Muresan, S. Folea, G. Mois, and E. H. Dulf, "Development and implementation of an fpga based fractional order controller for a dc motor," *Mechatronics*, vol. 23, no. 7, pp. 798–804, 2013.
[16] W. G. Soliman, B. K. Priya, D. A. Reddy, P. Anusha, and D. R. K. Reddy, "Reconfigurable microarchitecture-based pmdc prototype development for iot edge computing utilization," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 2334–2345, 2020.
[17] P. Bhandari and P. Z. Csurcsia, "Digital implementation of the pid controller," *Software Impacts*, vol. 13, p. 100306, 2022.
[18] K. Saurav and R. Potluri, "Sensorless speed control of a permanent magnet dc motor by compensating the plant nonlinearities," in *2013 IEEE International Symposium on Industrial Electronics*. IEEE, 2013, pp. 1–4.
[19] F. Alam, M. Ashfaq, and S. S. Zaidi, "Fpga implementation of a sensorless speed control architecture for the pmdc motor," in *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE, 2018, pp. 347–352.
[20] M. Aguado-Rojas, W. Pasillas-Lpine, A. Loria, and A. De Bernardinis, "On the compensation of incremental encoder imperfections for motion control: a dc motor case-study," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 627–632, 2018.
[21] A. Top and M. Gokbulut, "A novel period–based method for the ¨ measurement direct current motor velocity using low-resolver encoder," *Transactions of the Institute of Measurement and Control*, vol. 45, no. 4, pp. 711–722, 2023.
[22] C. O. Gokc¸e, M. E. ¨ ˙Ipek, M. Dayıoglu, and R. ˘ Unal, "Parameter ¨ estimation and speed control of real dc motor with low resolution encoder," *Results in Control and Optimization*, p. 100549, 2025.