ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

Testing a DC motor's embedded speed control system via a remote closed-loop control system

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

[1,2,3,4]Student, Department of EEE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.
[5]Assistant Professor, Department of EEE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

Abstract—

This article's goal is to evaluate the performance of an embedded platform for a virtual laboratory that can be accessed remotely. The common problem of regulating the speed of DC motors is what we will be examining in this lab. The embedded platform is connected to a single-board computer via USB protocol, which runs a control algorithm based on Python. This computer may be accessed remotely from many sources. This article provides a comprehensive overview of the embedded platform, including experimental results and analysis. The index makes use of the following terms: embedded platform, speed control, Python, DC motor, remote access laboratory, and DC motor.

I. INTRODUCTION

In engineering laboratory courses, students get a solid grounding in both the theory and practice of engineering. Any system's underlying principles may be shown in an engineering laboratory's physical installations at different operating points or in distinct modes. Proper operation of conventional hands-on laboratories need ample space, specialized apparatus, and qualified staff. The biggest problem with conventional laboratories is that only students from one institution get to access the top facilities and technology. The conventional, hands-on lab setting doesn't help a lot of pupils fully grasp the physical system. Improvements in processing power, specialized software, networking protocols, and communication protocols with improved bandwidth have brought most independent systems online. A trend toward VRLs, or virtual and remote labs, has emerged, replacing more conventional approaches. One of the main advantages of VRLs is the access they provide to both physical and digital resources, both locally and remotely [1]. In [2], we discussed the process of designing the software and hardware components of a virtual laboratory. The findings of conventional VRLs and conventional hands-on labs are detailed in [3]. Virtual reality loops rely on a network control system because of the network that links the controller to the physical process. Topics covered in the literature include VRL time-sharing remote access systems [5] and web-based network control laboratories [4]. In mechatronics, control systems, and electromechanical systems, the DC motor is one of the most basic components. The main goal of the benchmark challenge is to set the DC motor at a consistent speed. In order to regulate the motor's speed experimentally, one needs a DC motor, a load, speed sensors, and an embedded control system. A number of authors in the control literature have developed embedded platforms for controlling the speed of DC motors. The dsPIC30F4012, MCP2551, ARM, and FPGA are examples of such systems. The authors of [10] suggest a didatic platform that enables discrete control of a DC motor's speed and position. Along with the more conventional, experimental approaches, there are VRLs for controlling the speed of DC motors in the literature. A VRL may be built to control the speed of a DC motor using one of three microcontrollers: ARM, TINA, or STM32 [11–13]. Controls for DC motor speed that include field-programmable gate arrays (FPGAs) have been created in addition to microcontrollers [14], [15]. The authors of [16] developed a DC motor that has a dynamic pipeline system and a changeable microarchitecture. An edge-computing based control approach is used in this work. Detailed in [17] are the challenges of incorporating a PID controller into a digital setting. Other studies have looked at DC motors with sensorless speed control. When a speed sensor is not available, the speed estimator may be used to

estimate the motor's speed from the armature voltage and current[18]. The FPGA code for sensorless speed control of the DC motor is available in [19]. The embedded control platform may be built using a variety of programming languages, from high-level ones like MATLAB or LabVIEW to low-level ones like Assembly or C. These days, it's common advice for new programmers to begin their control algorithm development journey using an open-source language like Python. It takes much longer to produce open source programming than it does to write programs tailored to a microcontroller. New users may feel overwhelmed by the prospect of developing a control algorithm for a microcontroller due to the complexity of managing the resources accessible to such a device. The authors propose a mixed-language system where the control algorithm is coded in high-level languages and the actual hardware is communicated with by means of the Arduino microcontroller. The microcontroller takes care of sudden occurrences like interruptions and information from the speed sensor. Arduino and Python are able to talk to one another via the USB interface, which stands for Universal Serial Bus. In this article, we will go over the steps involved in developing, implementing, and testing an embedded control system to investigate different tactics for controlling the speed of DC motors. Arduino microcontroller boards are the most popular choice for hardware that measures the speed of the DC motor. The motor is controlled by a general-purpose computer. For the controller's implementation, Python or another language based on interpreters that are machine-independent is employed. As an added bonus, a local web server enables remote access to the Python integrated development environment, namely Jupyter Notebook. Through proper configuration of port forwarding, the embedded control system may be accessed via either the local area network or the internet. The following format is used in this piece. Section II provides comprehensive system details, including information on the hardware platform. The mathematical analysis of the DC motor is given in Section III. Section IV details the controller's inspection, whereas Section V details the results and analysis. The final ideas are laid forth in Section VI.

II. SYSTEM DESCRIPTION

Figure 1 shows a schematic view of the DC motor control setup's embedded platform. A USB cable connects the physical platform to the personal computer. The hardware configuration relies on the Python kernel running on this PC to communicate with the Arduino microcontroller. Moreover, a Jupyter Server is operating on the machine, giving users a web interface to access the Python Kernel. The Jupyter IDE really lets users launch fresh instances of the Python kernel whenever they open an application. This Jupyter IDE is available for use on any computer in your local area network. To facilitate remote work, however, the Jupyter server may be configured in a variety of ways to make it available from networks other than the local one. If you have a static IP address that is available to the public on your computer, you may configure the Jupyter server to listen to that address. Adding a static IP to a domain name configuration makes the system even more user-friendly. After successfully authenticating, users from any distant location may access the Jupyter server and utilize the Jupyter IDE on their own workstations. • In the absence of a static IP address, a Dynamic DNS service may be used; this service keeps tabs on the hardware-linked dynamic IP address of the computer and manages the connection to the computer via a predetermined domain or subdomain name. Individuals in various places may virtually join to the same network via their internet connections by setting up a Virtual Private Network (VPN). So, those not physically present at the organization may nevertheless access the experiment computer as if it were connected to the local network.

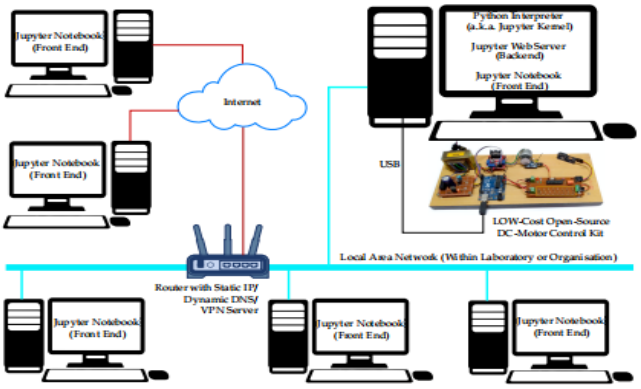


Fig. 1. Block Diagram of the Embedded Closed-Loop Set-up for remote access of the dc-motor control platform

Figure 2 depicts the block architecture of the DC motor control arrangement, and Figure 3 displays the developed hardware platform. The use of two microcontrollers is evident from the hardware configuration. From here on, the main

ISSN2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

microcontroller will be referred to as Arduino Uno, while the auxiliary microcontroller will be termed bare-bone Arduino. For the most eco-conscious build, go with 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 makes it a barebone Arduino. The main microcontroller controls the speed of the DC motor and interacts with the computer using pulse width modulation (PWM) signals. The dual H-bridge motor driver (L293D) takes its cues for regulating the DC motor's speed and direction from the microcontroller's output. It is within the L293D's voltage range of 4.5 V to 36 V that driving currents of up to 1 A are provided. You may control the speed of the DC motor by modifying the pulse width modulation (PWM) signal. The auxiliary microcontroller detects the DC motor's speed and converts it to an analog signal. Just how fast

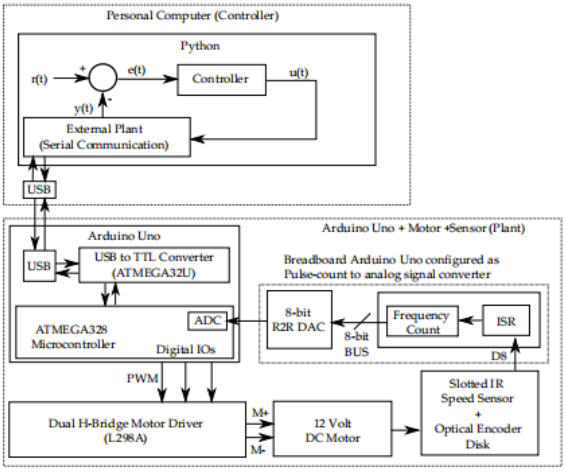


Fig. 2. Block diagram schematic for closed loop control of DC Motor speed using Arduino and Python

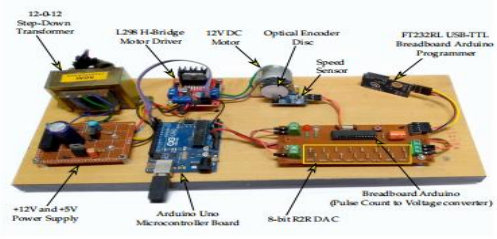


Fig. 3. Hardware setup for Speed control of a DC Motor using Python and Arduino

One way to measure a DC motor is using an optical encoder disk or a magnetic encoder that is attached to the motor shaft. This study takes optical encoders into account since they are more cost-effective than magnetic ones [20]. An optical encoder disk with a slotted infrared optical sensor generates pulses whenever a DC motor is activated. The relationship between the motor's speed and the number of pulses per second stays the same regardless of the speed encoders you choose. A frequency-based speed measurement using an optical encoder may be used to find the motor speed by

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

for N is the number of pulses in one rotation, n is the number of pulses counted, and T is the given fixed duration. Having said that, this technique isn't ideal for low-speed applications. The method based on periods is used for the measurement of low speeds [21], [22]. An example of a device that uses frequency to measure speed is the frequency counter. In microcontroller code, the frequency counter is implemented via interrupts. When the monitored pin experiences an interrupt, a certain subprogram or function inside the microcontroller's software known as an Interrupt Service Routine (ISR) begins running. This pin is connected to the pulses of the speed sensor. The DC motor was running at high speeds, but the Arduino's ability to interact with the PC was hindered by the large number of interruptions caused by the pulse output of the speed sensor. The speed sensor readings are handled by the main microcontroller, while the speed computations are handled by the secondary microcontroller. The frequency of the pulses is assigned by the auxiliary microcontroller, which is an 8-bit integer between zero and 250. This 8-bit value is sent to a DAC that is based on an R-2R ladder. An analog signal, with a voltage that could be anything from zero to five volts, is produced by the DAC from the frequency measurement.

III. MODELING AND ANALYSIS

The DC motor with a permanent magnet is shown schematically in Figure 4. The model is tested under the assumptions of a constant magnetic field, a linear system, and a constant susceptibility. There are a number of electrical characteristics of an armature that are shown in Figure 4. These include the torque constant (ka), back EMF, electromagnetic torque (Te), inductance (La), and resistance (ra). A constant field strength is produced by a permanent magnet since its susceptibility is likewise constant. As a result, ka is unchanging. Load torque (TL) and moment of inertia (J) are both precisely determined.

ISSN2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

The armature current (ia) and voltage (ua) are both specified. The number ór represents the motor's rotational speed.

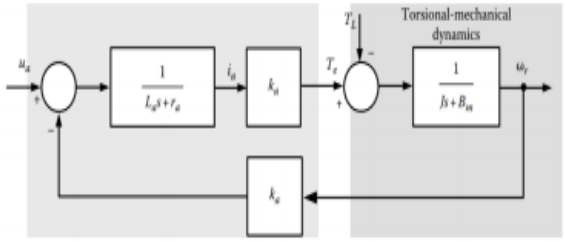


Fig. 4. Block diagram of permanent magnet DC motor

An equation that describes the dynamic performance of a DC motor may be expressed as

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

The specifications of the motor may be found in the datasheet provided by the manufacturer. System identification may be used to get the estimated parameters in situations when the actual values are unavailable.

IV. CONTROL LOOP

The simplified block diagram of the plant's closed-loop control system is shown in Figure 5. The speed of the motor may be monitored by integrating an optical encoder with the main microcontroller. The data is taken from the main microcontroller by the secondary microcontroller so that it may provide the processed speed data to the controller operating on a personal computer. The controller's actuation signal is sent to the DC motor via an H-bridge motor driver and a main microcontroller. A physical plant has several nonlinearities, such as deadzone and saturation. The deadzone is the voltage range in which the motor will not spin. Crucial is keeping the control signal within the saturation limits. One use of the zero-order hold (ZOH) is signal reconstruction via the use of digital-to-analog converters.

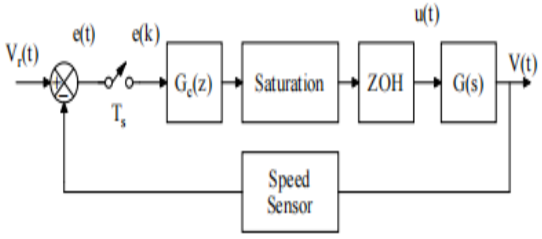


Fig. 5. Block diagram of closed-loop control of plant

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 gain, τi, is the integral time constant, and τd, the derivative time constant, and Equation (3) represents both the error signal e (t) and the controller's output u (t).

The laplace transform of the PID controller may be described in several ways.

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

We discretize the continuous transfer function of the controller with respect to derivatives in the Backward Euler form and integration in the Trapezoidal technique. The discrete form of a PID controller 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

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

By using the Ziegler-Nichols tuning criterion, the best gains for the controller may be calculated. The quarter decay ratio method is used to get the PID controller parameters. Kp=0.5K, Ti=0.5T, and Td=0.125T are the numbers that result from tuning the Ziegler-Nichols system with a quarter decay ratio.

ISSN2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026

The speed sensor may pick up the DC motor's speed and relay that information to the microcontroller. The Arduino microcontroller produces the pulse width modulation (PWM) signal, the duty cycle of which may be adjusted. The Python program manages the main control loop. It reads the speed of the DC motor from the Arduino Uno microcontroller using a USB serial connection, computes the control signal, and returns the relevant duty cycles to the microcontroller. Figure 6 displays the flow diagram.

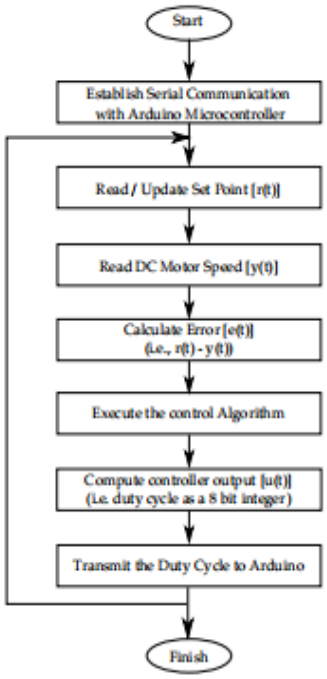


Fig. 6. Flow chart for DC Motor Speed Control in Pyt

V. EXPERIMENTAL VERIFICATION

A. System Identification

An experimental verification procedure to ascertain the transfer function involves keeping an eye on the DC motor's static properties, including its no-load speed, which is unaffected by the supplied voltage.

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

Figure 7 shows the DC motor's static characteristics as a function of the rising sequence of PWM (8 bits) values (0 to 255) and the matching RPM values. By adjusting the DC motor's voltage and current, we can determine the no-load speed (RPM) (Table I). Findings from the steady-state study are used to establish the process gain K, process time constant τ, and dead time L. To uncover the DC motor's inherent nonlinear dynamics, a stimulatory pseudo-random binary sequence (PRBS) signal is used in place of a linearly increasing voltage. Building the PRBS signal requires an XOR gate and a serial-in parallel-out (SIPO) shift register (74164).

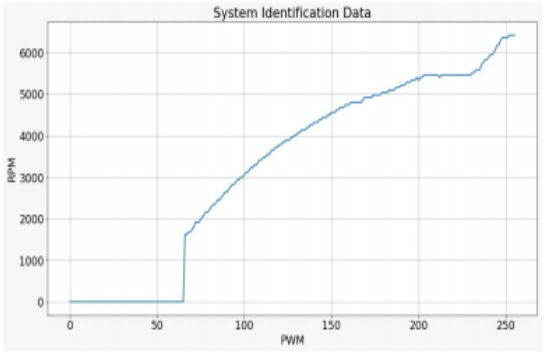


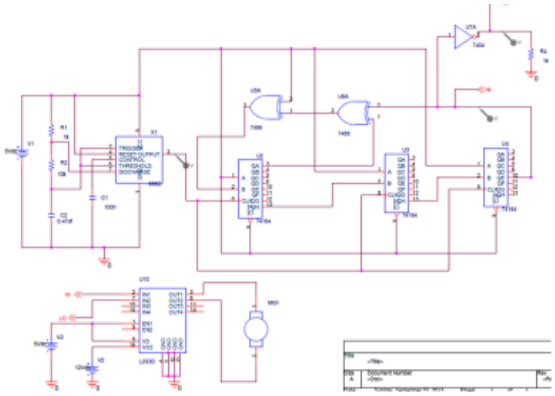
Fig. 7. The static characteristics of DC motor

Fig. 8(a). The response of the PRBS circuit is shown in Figure 8(b), where Ch2 generates the PRBS signal and Ch1 supplies the clock signal.

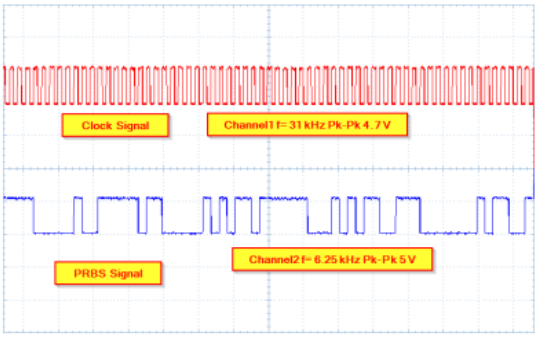
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940 www.ijasem.org

Vol 20, Issuse.2 April 2026



(a)



(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)

By identifying the system, we can estimate the transfer function of the DC motor as

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

B. Controller Performance Evaluation

By monitoring the embedded platform's response in regulatory performance and servo performance, the system is experimentally evaluated. The sampling time of the controller is half a second. For the purpose of this research, we use two separate signals—a pulsed signal and a random step signal—to measure performance. The response of the DC motor to a square wave, seen in Figure 9, is a duty cycle fluctuation of 40% to 80%. For this test, we used a three-second square wave stimulus. The findings show that the DC motor's response is proportionally slower to the input change while going from low to high rpm.

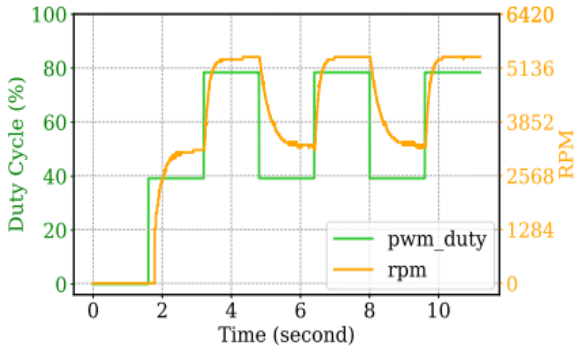


Fig. 9. DC Motor Speed Response for a pulse signal

In Figure 10, we can see how the DC motor responds to a step signal that is created at random. The graph depicts the dc-motor damping as it moves from high to low steps.

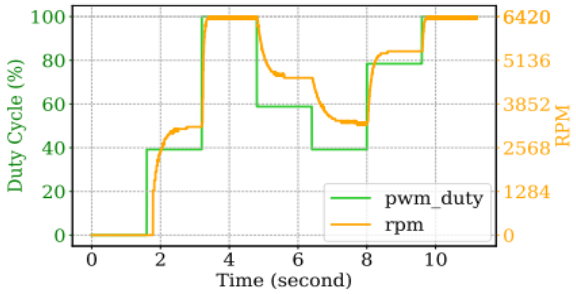


Fig. 10. DC Motor Speed Response for an arbitrary step signal

VI. CONCLUSION

This study presents an exhaustive analysis of the design, construction, and testing of the embedded platform for controlling the speed of DC motors. A remote access capability is provided by the embedded platform, which combines a microcontroller-based setup with a Python-based controller. The two are connected via the USB protocol. We have built the required equipment, which includes a 12 V DC motor, for this experiment. You can now see how the DC motor responds to different signals. The remote access functionality allows several users to access the physical system and work from distant PCs.

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.
[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.