Development and Experimental Validation of a Low–Cost DC Motor Platform for Educational Control Applications

Huseyin Deniz Ozturk[1], Cagri Hindistan[1,*], Volkan Akar[2], Oyku Ozge Korkmaz[1], Erman Selim[1], Enver Tatlicioglu[1]

***Abstract*— In this study, a low–cost real-time DC motor experimental setup that could be utilized for educational purposes in control systems and robotics courses in engineering departments is developed. The experimental platform utilizes an STM32–based microcontroller to generate the control signals applied to the motor and to obtain encoder measurements. Real-time communication between the hardware and MATLAB/Simulink is established for monitoring and control implementation. The DC motor is mathematically modeled, and the effectiveness of the experimental setup is evaluated using proportional integral and state feedback controllers. The performance of the proposed system is validated through both simulation and experimental studies.**

I. Introduction

In engineering disciplines that feedback control courses are taught, particularly to equip the students with specifics of feedback and control concepts, hands-on training is essential. Specifically, conveying specifics of classical controllers, such as proportional (P), proportional integral (PI), proportional integral derivative (PID), and modern ones, e.g. linear quadratic regulator (LQR), experimentally may provide longstanding returns to relevant industrial control applications. Still, in most control curricula, theoretical knowledge is supported with idealized academic examples run in simulation environments. However, it is known that systems tested solely in simulation may fail to exhibit the desired responses in real-life applications, which eventually leads to an educational process that overlooks the behavior of real-time control systems.

To provide the engineering students with at least some sense of industrial control applications, experimental setups, such as linear pendulum [1], [2], [3], ball-on-plate [4], [5], DC motor [6], [7], [8], [9], targeting educational purposes have been developed. These platforms provide an environment in which engineering students can observe the characteristics and challenges of real-time control systems directly. Educational setups designed for this purpose contribute to an effective learning setting that facilitates the understanding of feedback control systems and enables students to experience these structures in a hands-on manner. For such experimental systems, being user-friendly and as independent as possible from a dedicated workstation are also important to improve usability. Therefore, the use of embedded-system-based software for these testbeds offers a more efficient and practical solution. On the other hand, the entire programming can be developed in MATLAB/Simulink, which provides a user-friendly platform for students.

Recent rapid advances in power electronics and microelectronics have enabled the application of modern control techniques to motor and motion control systems. Additionally, as DC motors are among the most frequently preferred actuators in industrial applications and also due to their simple and instructive dynamic characteristics, testbeds involving DC motors qualifies as a promising experimental system for control engineering education [10]. There are some relevant results in the literature. To name a few, in [9], an online laboratory featuring a DC motor test bench is presented for electrical engineering students. The study describes the lab's implementation, including its hardware setup, software frameworks, and user interface, and outlines its educational applications through several exercises. In [11], a low-cost DC motor control platform was presented, in which the real-time behavior of the system was monitored directly on a desktop environment. In [7], real-time DC motor speed control using the TMS320F28379D digital signal processor is addressed. In [12], the educational potential of a simple DC motor plant for modeling and control in a basic control course is presented. The DC motor laboratory setup is developed based on Arduino UNO and integrated with the MATLAB/Simulink environment for communication and control. Another line of past research was devoted to implementing linear controller formulations on DC motor systems. In [13], the improvement of the speed and torque dynamic responses of a DC motor using a neural network parameter tuner for a PI controller is proposed. In [14], a PI control loop is used to implement closed-loop speed regulation of a BLDC motor drive. In [15], a state feedback controller for the speed control system of a DC motor is designed and implemented using MATLAB/Simulink. The proposed approach consists of a state feedback controller with a state observer and an integrator.

In this study, design, development and experimental verification of a modular DC motor experimental setup is presented. The proposed system is built using off-the-shelf components such as an STM32F746ZG microcontroller, a

[1]H. D. Ozturk, C. Hindistan, O. O. Korkmaz, E. Selim, and E. Tatlicioglu are with Department of Electrical & Electronics Engineering, Ege University, 35040, Bornova, Izmir, Türkiye huseyin.deniz.ozturk@ege.edu.tr, cagri.hindistan@ege.edu.tr, 05220000630@ogrenci.ege.edu.tr, erman.selim@ege.edu.tr, enver.tatlicioglu@ege.edu.tr

[2]V. Akar is with Department of Electrical & Electronics Engineering, Dokuz Eylul University, 35390, Buca, Izmir, Türkiye volkan.akar@ogr.deu.edu.tr

*Corresponding author: C. Hindistan.

high-resolution AS5047D magnetic encoder, a bidirectional BTS7960 motor driver, and a JGB37-3530 geared DC motor. Bidirectional communication between MATLAB/Simulink and the STM32 was established via USB, allowing controller parameters defined by the user to be transmitted to the microcontroller in real-time. Afterwards, a state feedback controller is implemented on the developed DC motor experimental setup in order to investigate the real-time performance of state space-based control design. This approach provided a systematic framework for assigning the closed-loop dynamics and analyzing transient response characteristics such as settling time, overshoot, and stability margins under experimental conditions. By applying state feedback control directly on the hardware platform, the proposed setup also enabled students and researchers to observe the relationship between the mathematical model and the actual plant response in real-time. In addition, a PI controller was implemented as a benchmark control strategy due to its widespread use in DC motor speed regulation and its practical simplicity. The inclusion of PI control made it possible to compare a classical transfer function-based controller with the state space-based state feedback method on the same experimental platform. This comparison provided a clearer understanding of the advantages and limitations of both approaches in terms of tuning effort, robustness, and real-time response performance.

The rest of the paper is outlined as follows. In Section II, development of the DC motor test setup is presented. Section III provides the details of the system model and the controller development. In Section IV, simulation and experimental results of proposed test setup are provided. Finally, in Section V, concluding remarks are given.

II. Development of a DC Motor Test Setup

A. System Architecture

The overall architecture of the system consists of four main components: an embedded control unit, a sensing unit, a motor driver, and a DC motor actuation system. These components form a closed–loop control structure in which the motion of the motor can be measured and regulated in real-time.

In addition to the hardware components, the system is supported by a host computer running MATLAB/Simulink for real-time monitoring and controller parameter tuning. Bidirectional communication between the embedded hardware and the MATLAB environment enables users to observe the dynamic response of the system and adjust controller parameters during operation. The modular architecture of the setup allows the integration of different control strategies and facilitates its use as both an educational laboratory platform and a research test bench. The overall architecture of the developed system is illustrated in Fig. 1.

Beyond the electronic and software layers, the physical robustness and mechanical integrity of the setup are critical for reliable long–term operation in a laboratory setting. The 3D mechanical design of the system was entirely modeled using Autodesk Fusion 360 and is presented in Fig. 2.

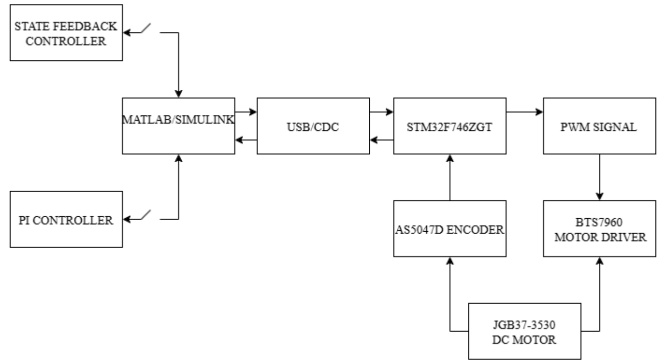

Fig. 1: Block diagram of the developed DC motor platform

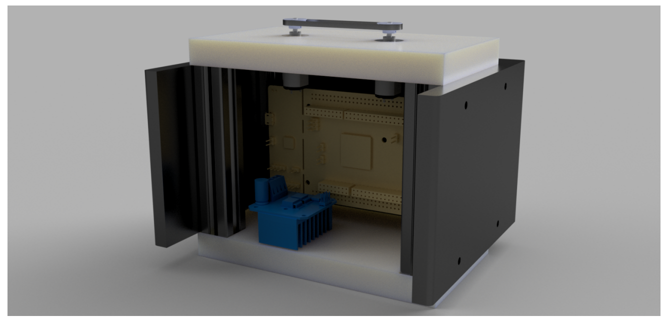
Fig. 2: 3D CAD model of the DC motor platform

To ensure structural rigidity and vibration dampening, the main mechanical chassis was manufactured from 5 mm thick aluminum plates utilizing a CNC machine. The mechanical design inherently incorporated dedicated mounting holes for the motor, encoder, and driver board, alongside strategically placed cable routing channels. The manufactured DC Motor Platform is shown in Fig. 3.

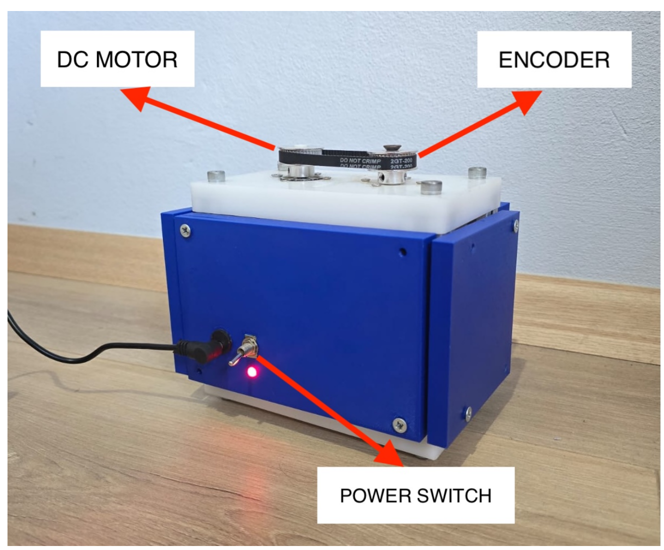

Fig. 3: Manufactured DC motor platform

B. Embedded Control Unit

The core processing unit of the developed system is the STM32F746ZG microcontroller. The microcontroller is based on the ARM Cortex–M7 architecture and operates at a clock frequency of up to 216 MHz. Thanks to its high computational capability and integrated peripheral interfaces such as timers, SPI communication, pulse width modulation (PWM) modules, and USB connectivity, the STM32F746ZG provides a suitable platform for embedded real-time motor control applications.

Within the developed platform, the microcontroller is responsible for generating PWM signals used to drive the motor, acquiring sensor measurements from the encoder, converts the control input received from the host computer into PWM signals, and transmits the measured speed data back to the computer. Hardware timer modules are used to generate PWM signals with precise duty cycle control, ensuring stable operation of the motor and reliable control performance.

C. Sensing Unit

Angular position feedback information of the motor is obtained using an AS5047D magnetic rotary encoder. This sensor provides absolute angular position measurements with a resolution of 14 bits and communicates with the microcontroller through the SPI interface. The high resolution of the encoder allows accurate measurement of the motor shaft position and enables reliable estimation of the motor speed. In the developed system, the angular velocity of the motor is calculated based on the difference between angular position measurements. This discrete differentiation approach provides an effective method for estimating rotational speed in real-time.

D. Motor Driver

The motor driver of the experimental setup is implemented using a BTS7960 high–current H–bridge motor driver module. This driver supports bidirectional control of the DC motor and is capable of handling relatively high current levels, making it suitable for laboratory–scale motor control applications.

E. Actuation System

The mechanical actuation of the system is provided by a JGB37–3530 geared DC motor. This motor was selected due to its low cost, stable torque characteristics and its suitability for low and medium speed control applications. The presence of an integrated gearbox allows the dynamic response of the system to be observed within a manageable rotational speed range, which is particularly advantageous for educational demonstrations and experimental analysis. The shaft of the DC motor is mechanically coupled with the magnetic encoder, allowing the rotational motion of the motor to be measured directly. This configuration enables the implementation of both speed and position control strategies within the same experimental platform.

F. MATLAB/Simulink Communication Interface

To enable interactive experimentation and real-time monitoring, a bidirectional communication interface is established between the STM32 microcontroller and the MATLAB/Simulink environment through USB–based serial communication. This communication structure allows system measurements such as motor speed to be sent to the host computer for visualization and control analysis, and control signals to be transmitted to the microcontroller for motor actuation.

III. System Modeling and Controller Design

A. Mathematical Modeling and Parameter Estimation

The dynamic behavior of the DC motor is governed by two fundamental differential equations, representing the electrical and mechanical dynamics in the following form [16]

$$v(t) = R\,i(t) + L\,\frac{di(t)}{dt} + K\,\omega(t), \tag{1}$$

$$K\,i(t) = J\,\frac{d\omega(t)}{dt} + b\,\omega(t), \tag{2}$$

where $v(t), i(t), \omega(t) \in \mathbb{R}$ are terminal voltage, armature current, and angular velocity, and $R, L, K, J, b \in \mathbb{R}_{>0}$ are resistance, inductance, back-EMF/torque constant, inertia, and viscous friction, respectively.

A two-stage identification procedure was employed. In the first stage, analytical parameter estimates were obtained from multi-level voltage excitation $v \in \{\pm 4, \pm 8\}$ V. The armature resistance was determined from peak current measurements under step excitation, while the back-EMF constant and viscous friction were identified from steady-state relations, and the inertia from the mechanical time constant. Since the electrical time constant $\tau_e = L/R \approx 51.8\,\mu$s requires a much higher sampling rate than the USB–CDC acquisition limit, the inductance was measured separately using an LCR meter, yielding $L = 0.462$ mH.

In the second stage, grey-box parameter refinement was performed by minimizing the normalized mean-square error over the full dataset using the Nelder–Mead simplex algorithm (MATLAB `fminsearch`) with the stiff solver `ode15s`. The final model achieved 99.0% and 95.3% fit for speed and current, respectively. The final parameters are listed in Table I.

TABLE I: Identified DC motor parameters

Parameter	Value	Unit
R	8.92	Ω
L	0.462	mH
K	0.01697	V·s/rad
J	2.95×10^{-6}	$\mathrm{kg} \cdot \mathrm{m}^2$
b	1.53×10^{-5}	N·m·s/rad

B. State Space Representation and Open-Loop Analysis

With state vector $x = [i, \omega]^\top$, input $u = v$, and output $y = \omega$, the model $\dot{x} = Ax + Bu$, $y = Cx$ gives

$$A = \begin{bmatrix} -R/L & -K/L \\ K/J & -b/J \end{bmatrix}, \quad B = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \tag{3}$$

which is evaluated numerically as

$$A = \begin{bmatrix} -19307 & -36.73 \\ 5752 & -5.186 \end{bmatrix}, \quad B = \begin{bmatrix} 2165 \\ 0 \end{bmatrix}. \tag{4}$$

After applying the MATLAB command of `ss2tf` to the identified state space model, the transfer function is obtained numerically as

$$G(s) = \frac{1.245 \times 10^7}{s^2 + 19313\, s + 311440}. \tag{5}$$

The open-loop eigenvalues are calculated as $s_1 \approx -19296$ rad/s (electrical, $\tau_e \approx 51.8\,\mu$s) and $s_2 = -16.14$ rad/s (mechanical, $\tau_m \approx 61.9$ ms), both real and negative, confirming open-loop stability. Since $|s_1|/|s_2| \approx 1195$, the electrical mode decays three orders of magnitude faster than the mechanical response and lies far outside the intended control bandwidth.

C. State Observer Design

Speed estimation by numerical differentiation of the 14-bit encoder output at 1 ms intervals yields quantization noise of approximately ± 200 RPM. To suppress this, a discrete-time Kalman filter was employed as a state observer for both control loops. The system matrices were discretized in MATLAB using `c2d` with zero-order hold at $T_s = 1$ ms. The process and measurement noise covariances were set to

$$\begin{aligned} Q &= \mathrm{diag}(0.01,\ 100), \\ R_n &= \left(200 \times \tfrac{2\pi}{60}\right)^2 \approx 439\text{rad/s}. \end{aligned} \tag{6}$$

where the noise amplitude is bounded by ± 200 RPM. It is noted that, modeling the noise as uniformly distributed gives a conservative assignment $\sigma_\omega = 200$ RPM ($R_n \approx 439$) to avoid over-trusting the model. The filter was implemented as a discrete-time Kalman Filter block in Simulink, reducing the effective speed noise to ± 15 RPM in both experiments. The observer introduces a latency of approximately 2 ms ($\approx 3.2\%$ of τ_m), which does not measurably affect the closed-loop transient response. The observer and state feedback gains are designed independently while preserving closed-loop stability.

D. State Feedback Controller Design

To achieve zero steady-state speed error, an integral state $\xi = \int(\omega_{\text{ref}} - \omega)\,dt$ is appended, forming $x_{\text{aug}} = [i, \omega, \xi]^\top$ with

$$A_{\text{aug}} = \begin{bmatrix} -19307 & -36.73 & 0 \\ 5752 & -5.186 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \quad B_{\text{aug}} = \begin{bmatrix} 2165 \\ 0 \\ 0 \end{bmatrix}. \tag{7}$$

The augmented system is shown to remain fully controllable. The desired closed-loop poles are considered as

$$\begin{aligned} &p_1 = -19291 \text{ rad/s } (\textit{i.e.}, \approx s_1), \\ &p_2 = -20 \text{ rad/s}, \quad p_3 = -12 \text{ rad/s}. \end{aligned} \tag{8}$$

Since relocating pole p_1 would require impractically large control effort, it is placed near the open-loop electrical eigenvalue. The integral pole p_3 is intentionally set slower at -12 rad/s, so that the integrator gain remains small and the ± 200 RPM encoder noise does not accumulate excessively. As a consequence, p_3 dominates the closed-loop settling time: $4/|p_3| \approx 0.33$ s. The gain matrix was computed in MATLAB using Ackermann's formula [17]

$$K_{\text{aug}} = \begin{bmatrix} 0.0048 & 0.0246 & -0.3718 \end{bmatrix}. \tag{9}$$

The control law $u = -K_{\text{aug}} x_{\text{aug}}$ expands to $u = -k_1 i - k_2\omega + |k_3|\xi$, confirming positive integral action as required for reference tracking. Since the proposed setup serves as an educational platform, students are expected to select their own target poles and recompute K_{aug} accordingly.

E. PI Controller Design

A PI controller serves as a classical benchmark against the state space approach. Since $\tau_e \ll \tau_m$, the electrical pole is transparent at PI bandwidth; the full-order transfer function (5) is used directly for loop-shaping. Gains were tuned using the Simulink PID Tuner, yielding

$$K_p = 0.00304, \qquad K_i = 0.04125. \tag{10}$$

The same Kalman observer described in Section III-C is applied to the PI loop. As this platform is intended for education, students are expected to retune both controllers for their own desired responses.

IV. Simulation and Experimental Results

All control algorithms were designed and validated in MATLAB/Simulink and subsequently deployed to the STM32F746ZG microcontroller via USB–CDC real-time communication. The effectiveness of the designed controllers was evaluated through both simulation and experimental studies conducted using MATLAB/Simulink. The duration of all experiments was set to 10 sec, and the reference speed was selected as 3000 rpm for all test scenarios.

A. Case I: State Feedback Controller

In the first case, the performance of the state feedback controller is investigated. Fig. 4 presents the simulation results of the state feedback controller. The results obtained in the simulation environment indicate that the system has a settling time of less than 0.5 second and reaches the reference speed without overshoot, exhibiting a critically damped characteristic. The error signal converges exactly to zero within the simulation time. Furthermore, the control signal remains within levels that are feasible for practical implementation in the experimental setup.

The experimental results corresponding to the state feedback controller are shown in Fig. 5. Overall, the experimental results demonstrate a high degree of agreement with the predictions of the simulation model. In particular, when the control input u is examined, it closely mirrors the smooth transient behavior observed in the simulation study. The controller effectively handles the large initial speed error without exhibiting any aggressive spikes, providing a clean and controlled increase in voltage during the start-up phase. As the motor reaches the reference speed, the control

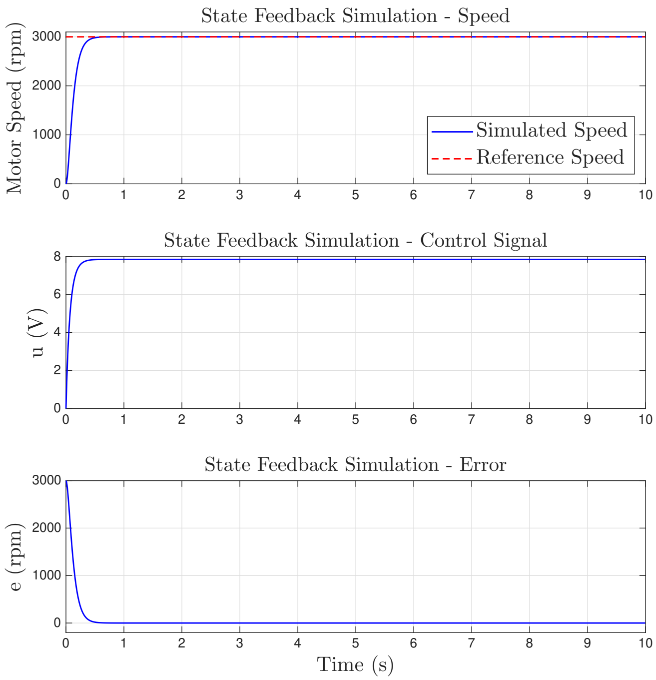

Fig. 4: Simulation results of state feedback controller

signal settles stably around its nominal steady-state value of approximately 7.5 V. In the error plot, a small oscillation within the band of approximately ± 15 rpm is observed. These fluctuations can be attributed to the quantization noise of the encoder, electrical imperfections in sensor measurements. Nevertheless, these effects remain within acceptable limits and do not significantly degrade the overall control performance of the system.

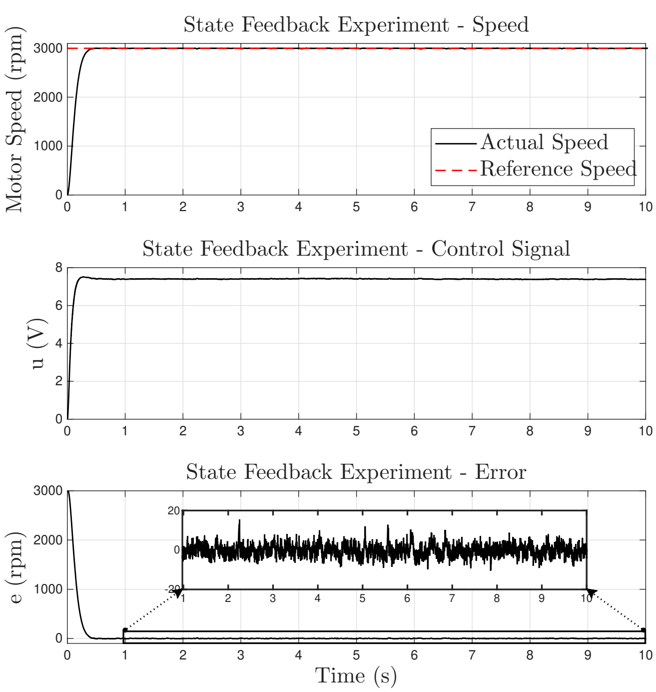

Fig. 5: Experiment results of state feedback controller

B. *Case II: PI Controller*

In the second case, a PI control algorithm is implemented to evaluate the system performance. Fig. 6 presents the simulation results obtained from the identified mathematical model of the developed DC motor experimental setup under the PI controller. Fig. 6 presents the simulation results obtained from the identified mathematical model of the developed DC motor experimental setup under the PI controller. The simulation results indicate that the motor speed follows the reference trajectory with high precision, demonstrating a well-damped transient response without any overshoot. The system reaches the reference speed rapidly, achieving a settling time of less than 0.5 seconds. The control signal u remains within the physical limits of the actuator.

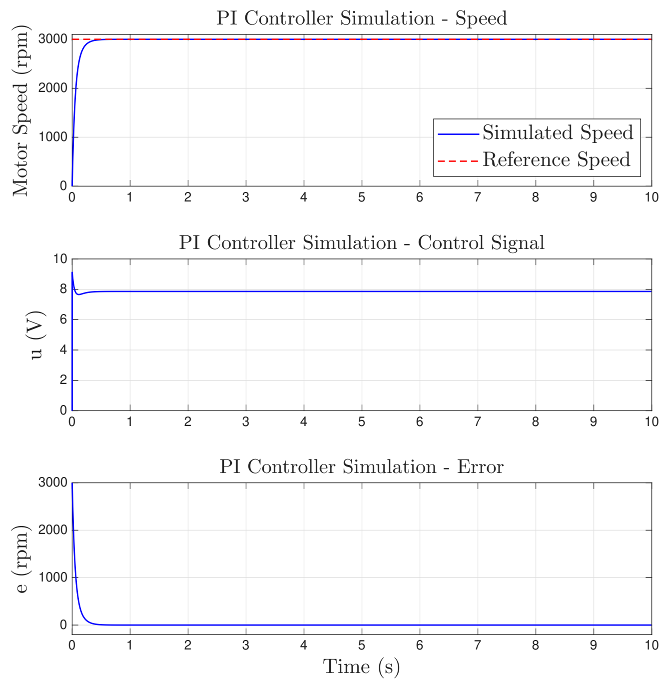

Fig. 6: Simulation results of PI controller

The experimental results obtained with the PI controller are illustrated in Fig. 7. The results confirm that the fast, non-oscillating, and well-damped response observed in the simulation is successfully reflected in the physical implementation. The slight phase and amplitude discrepancies between the experimental and theoretical responses can be attributed to static and dynamic friction parameters as well as mechanical delays that are not included in the mathematical model of the motor. From the perspective of the control signal u, the actuator is driven with a smoother acceleration profile, reaching a maximum value of approximately 9 V, and eventually settling at the steady-state voltage level of around 7.5 V. Although the error is theoretically expected to converge to zero, the experimental results show a small fluctuation of approximately ± 10 rpm due to hardware-related noise in the physical setup. Overall, both the PI controller and the state feedback controller achieve an average zero steady–state error and successfully drive the system to the desired reference speed.

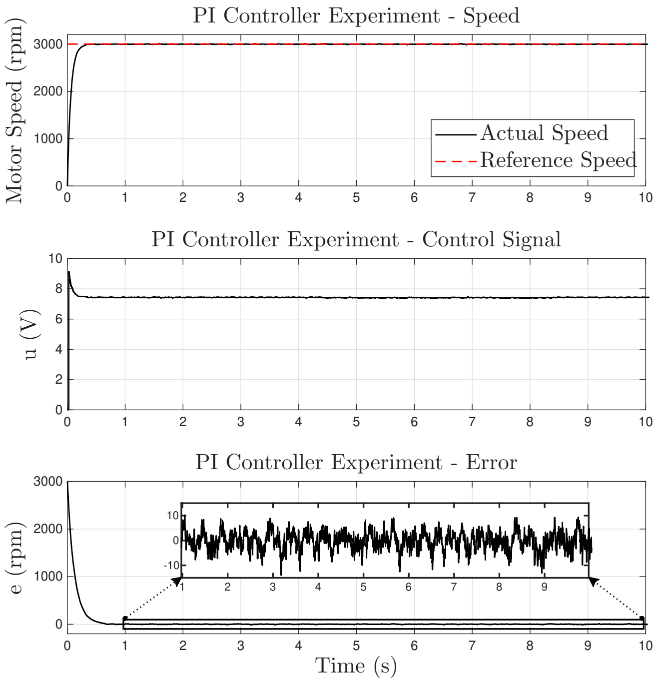

Fig. 7: Experimental Results of PI Controller

V. Conclusions

In this study, a low–cost DC motor experimental setup was developed targeting control systems education. The developed experimental platform was designed as a real–time DC motor control system that allows the implementation and evaluation of different control algorithms on a physical setup. The DC motor was mathematically modeled as a second–order system while its performance under both PI and state feedback controllers was investigated. Using the obtained model, the PI controller was tuned, and the gain matrix for the state feedback controller was calculated. The performance of the modeled system was first verified through simulations, confirming that the target speed could be reached. Subsequently, real-time experiments demonstrated that both control strategies were highly effective in achieving the desired reference speed, with the steady–state error converging to zero.

The developed experimental setup provides a practical and cost–effective tool for control systems courses and is intended to enhance students' hands–on learning experience. The system allows the comparative implementation of state feedback and PI controllers, while also enabling users to directly observe the effect of parameter changes on the dynamic response of the system. Furthermore, the modular structure of the setup provides an extensible platform that can be adapted to different microcontrollers, motor types, or control strategies. Through this system, students gain not only an understanding of controller structures, but also multidisciplinary skills related to embedded system design, sensor integration, signal processing, and real-time communication.

Compared with existing systems in the literature, the developed experimental setup offers a more compact learning environment in terms of resolution, computational capability, user interaction, modularity, and flexibility. It is suitable both for teaching fundamental control structures and for testing advanced controller algorithms in research projects.

Acknowledgment

This study was supported by Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the Grant Number 124E563. The authors thank to TÜBİTAK for their supports.

References

[1] G. H. Lee and S. Jung, "Reference compensation technique of a fuzzy controlled inverted pendulum system for control education," in *Int. Conf. on Control, Automation and Systems*, 2008, pp. 1704–1708.
[2] ——, "Design and control of an inverted pendulum system for intelligent mechatronics system control education," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2008, pp. 1254–1259.
[3] R. Ganganath, C. Ranganath, and B. Annasiwaththa, "Remotely operated rotary inverted pendulum system for online control engineering education," in *Int. Conf. on Electrical Engineering and Informatics*, 2022, pp. 137–142.
[4] B. I. Răzvan and R. Rusu-Both, "Ball-plate control system design and implementation for educational purposes," in *Int. Conf. on System Theory, Control and Computing*, 2025, pp. 836–841.
[5] C. J. Bay and B. P. Rasmussen, "Exploring controls education: A reconfigurable ball and plate platform kit," in *American Control Conf.*, 2016, pp. 6652–6657.
[6] H. Wang, J. Tian, and Y. Yang, "Design of feedforward pid control system for dc motor based on simulink and dsp," in *Int. Conf. on Electronics, Integrated Circuits and Communication Technology*, 2025, pp. 476–480.
[7] N. K. Alnagbi, A. S. Almazrouei, A. M. Alali, L. A. Tayara, L. B. H. Al Shaqfa, T. B. Hashfi, and A. Wahyudie, "Remote laboratory design for a control systems course: An example case of dc motors' speed control," in *Int. Conf. on E-Education, E-Business, E-Management and E-Learning*, 2025, pp. 81–86.
[8] K. Pandey, R. Mahadeva, and A. K. Goyal, "Design and implementation of a verilog based pid controller for dc motor speed control," in *Artificial Intelligence and Smart Technologies for Sustainability Conf.*, 2025, pp. 1–4.
[9] S. Baltayan, C. Kreiter, and A. Pester, "An online dc-motor test bench for engineering education," in *IEEE Global Engineering Education Conf.*, 2018, pp. 1484–1488.
[10] A. P. Moreira, P. Costa, J. Gonçalves, and B. M. Faria, "Dc motor educational kit: a teaching aid in control theory," in *Portuguese Conf. on Automatic Control*, 2016, pp. 879–889.
[11] 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.
[12] M. Hypiusová, M. Minár, and D. Rosinová, "Basic control course with dc motor," in *Cybernetics & Informatics*, 2022, pp. 1–6.
[13] Y. Naung, S. Anatolii, and Y. H. Lin, "Speed control of dc motor by using neural network parameter tuner for pi-controller," in *IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering*, 2019, pp. 2152–2156.
[14] B. Bairwa, M. Murari, M. Shahapur, K. MR, and M. F. Khan, "Speed control of bldc motor using pi controller," in *Int. Conf. for Advancement in Technology*, 2023, pp. 1–6.
[15] U. Manwong, S. Boonpiyathud, and S. Tunyasrirut, "Implementation of a dspace dsp-based state feedback with state observer using matlab/simulink for a speed control of dc motor system," in *Int. Conf. on Control, Automation and Systems*, 2008, pp. 2433–2436.
[16] M. Ruderman, J. Krettek, F. Hoffmann, and T. Bertram, "Optimal state space control of dc motor," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5796–5801, 2008.
[17] K. Ogata, "Modern control engineering," 1970.