

## DC motor position control with Raspberry Pi on Simulink wireless external mode

Erdem Ilten

*Electrical-Electronics Engineering, Balikesir University, Turkey*

*erdemilten@balikesir.edu.tr*

**Abstract** – In recent years, the use of Raspberry Pi as a microcontroller in real-time applications has become widespread. In this study, Raspberry Pi is used for DC motor position control system. In addition, wireless communication is provided between the external control panel prepared in the Matlab/Simulink program and the Raspberry Pi. Thanks to this control panel, the system can be turned on and off, the controller gains can be changed, and the position information of the motor can be displayed on the scope in real-time.

**Keywords** – *Raspberry PI, Position Control, PI Control, Simulink, External Mode*

### I. INTRODUCTION

DC (Direct Current) motors are widely used in various applications, ranging from robotics and industrial automation to consumer electronics and automotive systems [1]. To achieve precise control over the position of a DC motor, a position control system is employed. This control system ensures that the motor rotates to and maintains a desired position accurately, enabling precise motion control and automation. The position control of a DC motor involves monitoring the angular position of the motor's shaft and adjusting the motor's input voltage or current to achieve the desired position. The control system typically consists of a combination of hardware components and software algorithms. At the heart of the position control system is a feedback loop that continuously measures the actual position of the motor and compares it to the desired position. The most used feedback device for position control is an encoder, which provides accurate feedback about the motor's rotational position. The encoder generates pulses that correspond to the motor's angular displacement, allowing the control system to calculate the current position and make necessary adjustments.

To control the position of the DC motor, a control algorithm, such as a PID (Proportional-Integral-Derivative) controller, is employed [2]. The controller takes the difference between the desired position and the measured position (known as the error) and calculates an appropriate control signal to adjust the motor's input voltage or current. The PID controller uses proportional, integral, and derivative terms to determine the control signal, providing a balance between stability, responsiveness, and steady-state accuracy.

Raspberry Pi, a versatile and affordable single-board computer, has gained immense popularity among hobbyists, educators, and professionals alike. While it is commonly known for its capabilities as a small, low-cost computer, Raspberry Pi can also be utilized effectively as a microcontroller, offering a wide range of possibilities for various embedded systems and IoT (Internet of Things) projects. Traditionally, microcontrollers have been the first choice for controlling and monitoring devices in embedded systems [3]. These microcontrollers are specifically designed for tasks requiring real-time control and interaction with the physical world. However, Raspberry Pi's computational power and expansive

set of features make it an attractive alternative for microcontroller applications. Raspberry Pi offers a broad range of connectivity options, including USB, Ethernet, Wi-Fi, and Bluetooth, enabling seamless integration with other devices and the internet. This connectivity, combined with the onboard GPIO (General-Purpose Input/Output) pins, allows you to interface with various sensors, actuators, and electronic components, making it an ideal choice for building interactive prototypes or IoT projects.

Simulink, a popular tool for modeling and simulating dynamic systems, offers a powerful feature called External Mode. External Mode allows you to interactively connect Simulink models with external hardware or software in real-time, enabling rapid prototyping, testing, and validation of control algorithms and embedded systems. External Mode serves as a bridge between the Simulink model running on a host computer and the target hardware or software system. It establishes a communication link that allows you to monitor and tune parameters, visualize signals, and collect data from the running simulation or control algorithm in real-time [4]. Simulink External Mode supports various communication interfaces, including serial, TCP/IP, and CAN (Controller Area Network), providing flexibility in connecting with a wide range of hardware platforms or external software systems. It also integrates seamlessly with popular hardware support packages, allowing you to deploy and test control algorithms on real-time hardware platforms, such as microcontrollers, DSPs (Digital Signal Processors), or FPGA (Field-Programmable Gate Array) devices [5].

## II. MATERIALS AND METHOD

Closed-circuit voltage equation of DC motor is given in Eq. (1) [6].

$$V_a = E_c + I_a R_a \quad (1)$$

In Eq. (1)  $V_a$  is the supply voltage,  $E_c$  is the back EMF,  $R_a$  is the resistance of the armature winding. Back EMF equation is follows:

$$E_c = K_1 \phi_f \omega \quad (2)$$

In Eq. (2)  $K_1$  is a constant depends on motor parameters such as number of armature conductors and number of poles.  $\phi_f$  is the field flux and  $\omega$  is the

angular speed of the armature. Torque equation can be defined as Eq. (3).

$$T_o = \frac{P_o}{\omega} \quad (3)$$

In Eq. (3)  $P_o$  is the power obtained from the motor shaft and  $\omega$  is the angular speed as rad/s.

Block diagram of DC motor position control system is illustrated in Fig. 1. The experimental setup of the system is presented in Fig. 2.

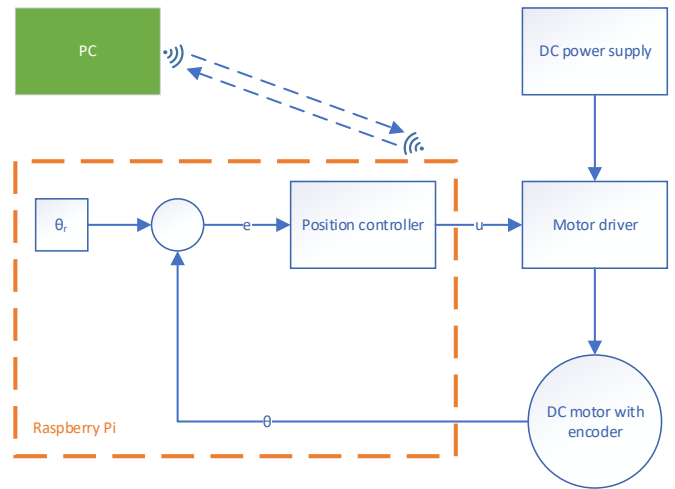


Fig. 1 Block diagram of the system

In Fig. 1  $\theta_r$  is the reference position,  $\theta$  is the actual position,  $e$  is the position error between  $\theta$  and  $\theta_r$  and  $u$  is the output of the speed controller. PID method is used for position controller and its equation is given in Eq. 4 [7].

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d e(t) \frac{d}{dt} \quad (4)$$

In Eq. 4  $K_p$ ,  $K_i$  and  $K_d$  are proportional, integral, and derivative controller gains, respectively.

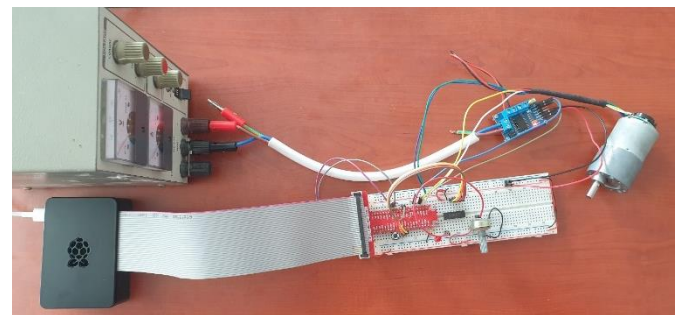


Fig. 2 Experimental setup

The experimental setup consists of a 12 V brushed DC motor with an integrated quadrature encoder with 64 counts per revolution (CPR), a Raspberry Pi 4B board, a DC motor control board, a 12 V<sub>dc</sub> power supply and a PC.

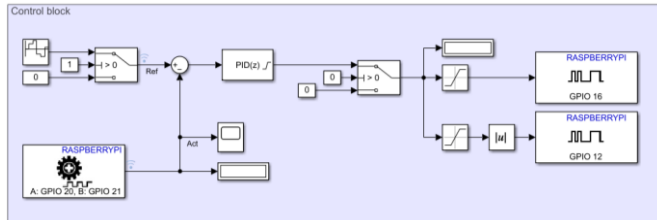


Fig. 3 Simulink control block

*Simulink Support Package for Raspberry Pi Hardware* [8] toolbox is used for creation of the block diagram is shown on Fig 3.

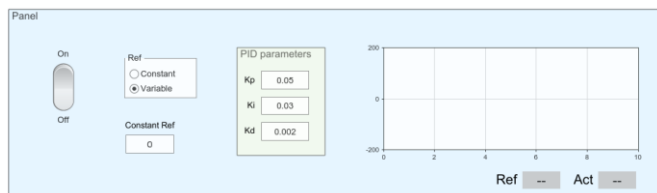


Fig. 4 External mode control panel

Simulink dashboard library is used in the design of the panel in Figure 4. Through this panel, the system can be turned on and off, constant reference or variable reference position can be selected, PID controller gains can be changed, reference position and actual position values can be observed on the scope in real time.

### III. RESULTS

The system is tested on the experimental setup and the results are given in Fig 5. In Fig. 5 the block diagram is runs on Raspberry Pi and controls the position of DC motor with using encoder and motor driver. Thanks to external mode, position controller gains can be changed, actual position can be seen on scope while the system is running real-time.

### IV. DISCUSSION

Thanks to the designed block diagram and control panel, the PID controller structure and its application to the DC motor system are shown in a simple way. Thanks to this system, it is aimed that undergraduate students can learn the PID controller structure practically.

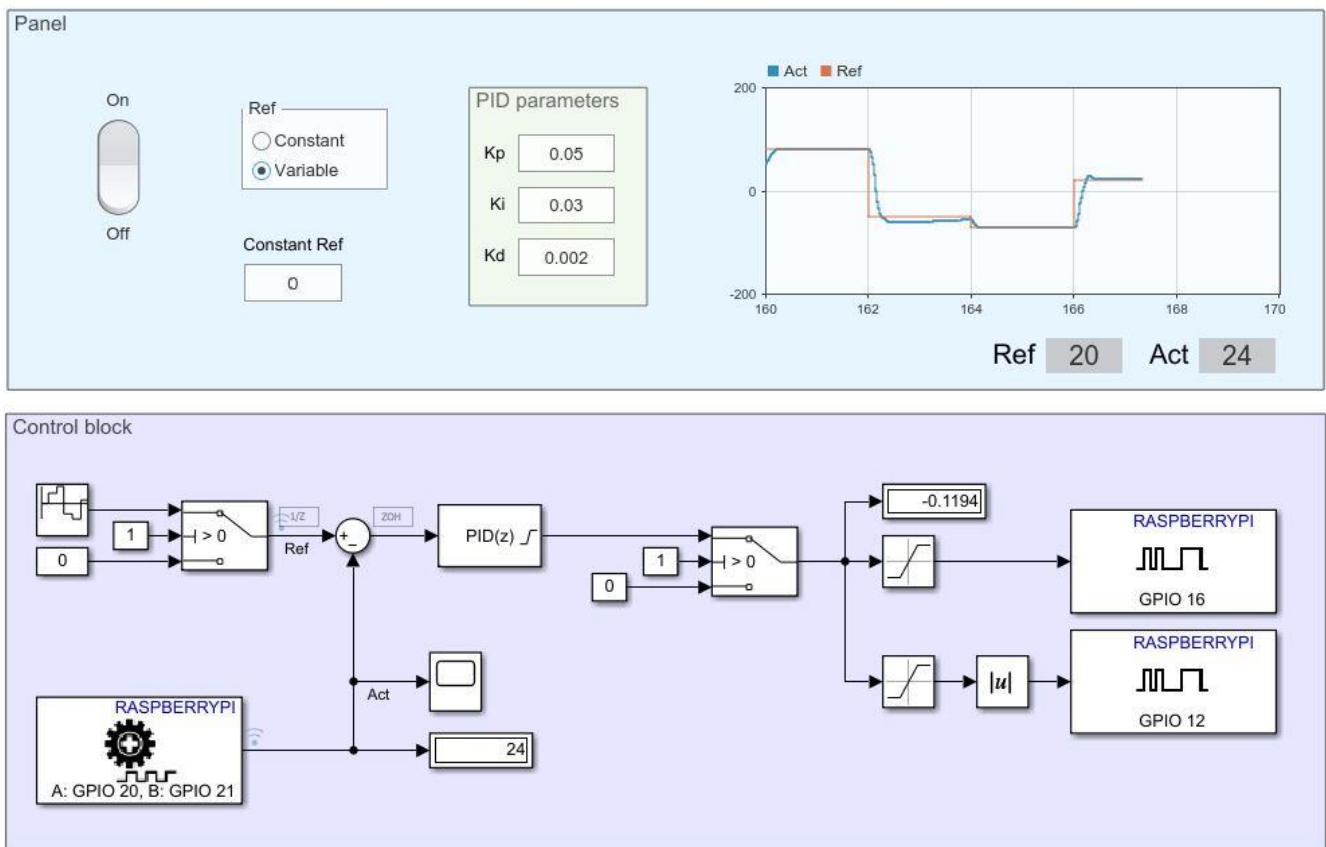


Fig. 5 Test results

## V. CONCLUSION

In this study, DC motor position control application is made in real-time via Simulink external mode using Raspberry Pi. Position control block diagram and user panel designed for this purpose are explained. As a result of the tests, it is seen that the PID controller successfully tracks the reference position.

## REFERENCES

- [1] I. Petráš, "Fractional-order feedback control of a DC motor," *J. Electr. Eng.*, vol. 60, no. 3, pp. 117–128, 2009.
- [2] K. Kaplan, M. Kuncan, H. Polat, T. Burak, and H. M. Ertunç, "PID ve Bulanık Mantık Tabanlı DC Motorun Gerçek Zamanlı Konum Kontrolü," *J. Inst. Sci. Technol.*, vol. 10, no. 2, pp. 900–916, 2020.
- [3] M. Šustek, M. Marčaník, P. Tomášek, and Z. Úředníček, "DC motors and servo-motors controlled by Raspberry Pi 2B," 2017.
- [4] E. Ilten, "Induction Motor Speed Controller Interface Design Via Simulink External Mode," in *II. International Joint Conference on Engineering, Science and Artificial Intelligence, 2022*, pp. 59–63.
- [5] E. Ilten, "Conformable Fractional Order Controller Design and Implementation for Per-Phase Voltage Regulation of Three-Phase SEIG Under Unbalanced Load," *Electr. Power Components Syst.*, vol. 50, no. 11–12, pp. 636–648, Nov. 2022.
- [6] G. Bal, "Doğru akım makinaları ve Sürücüleri," 2014.
- [7] E. Ilten, "Conformable fractional order controller design and optimization for sensorless control of induction motor," *COMPEL-The Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 41, no. 5, pp. 1528–1541, 2022.
- [8] MathWorks, "MATLAB documentation." The MathWorks, 2023.