İSTANBUL UNIVERSITY
CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2025; 25: 1-15

LQR-Based PID Optimization for Model-Based Control of Brushless DC Motors in HEVs

Sedat İn[1], Berna Örs[2], Hasan Tiryaki[3]

[1]Department of Electrical-Electronics Engineering, İstanbul University-Cerrahpaşa Faculty of Engineering, İstanbul, Türkiye
[2]Department of Electronics and Communication Engineering, İstanbul Technical University Faculty of Electrical and Electronics Engineering, İstanbul, Türkiye
[3]Department of Electrical-Electronics Engineering, Kırklareli University Faculty of Engineering, Kırklareli, Türkiye

Cite this article as: S. İn, B. Örs and H. Tiryaki, "LQR-based PID optimization for model-based control of brushless DC motors in HEVs," *Electrica*, 25, 0186, 2025. doi: 10.5152/electrica.2025.24186.

WHAT IS ALREADY KNOWN ON THIS TOPIC?

- *Motor control is performed using different methods.*
- *Determining feasible and optimised controller coefficients is critical for motor control performance.*
- *Processors for motor control are coded using various programming languages (C, C++, etc.).*

WHAT DOES THIS STUDY ADD ON THIS TOPIC?

- *Thanks to the model-based approach, the parameters required for motor control are obtained with higher accuracy.*
- *By using the optimisation method, the most applicable controller coefficients are determined.*
- *With the code generation technique, the coding of the motor control was performed using block diagrams. In this way, the code development process is accelerated and the debugging process is made more efficient.*

Corresponding Author:
Sedat İn

E-mail:
sedat.in@iuc.edu.tr

Received: December 3, 2024
Revision requested: January 5, 2025
Last revision received: January 7, 2025
Accepted: February 10, 2025
Publication Date: April 14, 2025

DOI: 10.5152/electrica.2025.24186

ABSTRACT

In this study, a control brushed direct current motor is selected, and a modeling study is carried out. All parameters used in the modeling are presented in detail. A controller is designed for the modeled motor. Traditional methods (Ziegler–Nichols Open Loop Response and Ziegler–Nichols Closed Loop Response), computer-aided calculation methods (MATLAB (matrix laboratory) tuning), and coefficient determination methods obtained by optimization (Linear Quadratic Regulator) are investigated to determine the parameters of the proportional integral derivative controller. The controllers obtained by these methods are applied separately to the system. The conversion of the Simulink project into code for a card with an embedded ARM (Acorn RISC Machine) processor is mentioned. How to create the project to be converted into code is explained. Motor speed and applied voltage graphs are analyzed in detail. The controller coefficients obtained with MATLAB tuning, a computer-aided calculation method, showed a high overshoot in the speed graph. Therefore, the MATLAB tuning method is not suitable for the motor control used in this study. Traditional methods are not applicable either. The controller whose parameters were generated with Ziegler–Nichols Open Loop Response showed too much speed drop when the motor was under load. The controller with parameters generated by Ziegler–Nichols Closed Loop Response has very high motor speed oscillation. The reasons why traditional methods are not applicable are explained in detail in this study. It is decided that the appropriate method for this motor is a Linear Quadratic Regulator. How to determine the controller parameters using a Linear Quadratic Regulator is explained. The applicability and comparison with other methods are presented.

***Index Terms*—**Code generation, hybrid electric vehicle, linear quadratic regulator, proportional integral derivative, modeling, Ziegler–Nichols methods

I. INTRODUCTION

The electric motor is an energy conversion device invented by Michael Faraday in 1821 [1]. Electric motors are basically devices designed to convert electrical energy into mechanical energy. Electric motors can be divided into two types: direct current (DC) and alternating current (AC). Alternating current electric motors are mostly used in areas where high power is required [2]. In this type of motor, losses increase as the speed control voltage decreases [3]. Direct current electric motors can be preferred in areas with lower power consumption as well as in areas with high power consumption. Direct Current electric motors are divided into two types, brushed and brushless. To control brushed DC motors (BLDC), it is sufficient to keep the supply voltage or current stable. Control brushed DC motors can be controlled by different methods for different needs. Commonly used control studies for BLDC motors are speed or torque control studies. There are studies that try to control both speed and torque.

Control brushed DC motors can be specially manufactured according to the required application. There are many BLDCs with different types and different requirements, such as high-speed low torque, and low-speed high torque. An example of design visualization for BLDC is shown in detail in Fig. 1 [4].

The electric motor shown in Fig. 1 is an internal rotor motor. The rotor is the area where the magnets are located and move. With the stator windings on the outside, a magnetic field is applied to the magnets on the rotor, and rotation is achieved with thrust. The part that processes this control algorithm is the motor drive. Motor drive structures should be designed in accordance with the motor that is expected to be driven. It is important to design a motor driver that can meet the performance requirements of the motor. To prevent damage to the motor driver structure, all structural parameters of the motor to be driven must be known. It is possible to model the motor using the motor parameters. A control system can then be developed for the modeled motor.

The most commonly used control method for motor control is the proportional integral derivative (PID) controller. To achieve PID control, three different control coefficients must be determined. These control coefficients can be obtained by different methods. The Ziegler–Nichols method, which is the simplest method of parameter determination, can quickly create a controller. Controller parameters can be determined by various methods (e.g., computer-aided tuning method). In addition to all these methods, it is also possible to obtain the coefficients using the optimization method [5]. There are studies in the literature where the controller parameters of PID used for motor control are determined by Ziegler–Nichols methods [6]. In addition, there are studies in the literature where the control system is created with a separate proportional integral (PI) controller controlling the PID block [7]. This reduces the overshoot rate. A genetic algorithm was also used to determine the PID coefficients. It has been observed that the controllers obtained by Genetic Algorithm studies provide better control than conventional PID methods [8]. For the motor control system, a study was also carried out in which an adaptive speed was first generated with a PID controller and then an adaptive voltage was generated using this adaptive speed [9]. There are also studies that aim to improve the system response by adding different methods to the PID control. One of them is the study where a hysteresis current control block is added [10]. In this study, a PI controller is used, and overshoots are prevented by a saturation block. Another method of preventing overshoot is to use a 'compensator.' By using the compensator, an attempt is made to capture the unit step response without overshooting by adding the required number of zeros and poles to the root locus curve of the system. In the studies, the use of either the PID block or the Compensator block alone is not considered sufficient, and it is recommended to use both methods together to achieve good results [11]. Dynamic modification of the PID coefficients with fuzzy logic has been investigated in studies [12][13][14][15].In control studies with fuzzy logic, it is important to create the rule table correctly [16].

As part of this study, a control study was carried out on a motor drive intended for use in hybrid electric vehicles. Efficiency is very important for hybrid electric vehicles. For this reason, the system should produce an output that is as close as possible to the desired output value. In order to realize this system, firstly the motor model was created. How the motor model is created is explained in detail in the second section. Then, the coefficients for the controller suitable for the model were obtained using various methods mentioned in section three. In section four, how to generate code from the project is explained in detail, and the notes to be considered for code generation are mentioned. The obtained controller coefficients are simulated in the motor model, and the simulation results are explained in detail in section five. In the literature, motor control is possible with many different methods. The Linear Quadratic Regulator (LQR) method emphasized in this study has shown more successful results in terms of applicability compared to all other methods. The results compared in the sixth section are explained in detail. In previous research, motor control using this method has not been encountered much. This shows that the proposed method is both efficient and innovative.

II. MODEL OF CONTROL BRUSHED DIRECT CURRENT MOTOR

Electric motors vary according to their application. In order to work with a BLDC motor, it is necessary to know the manufacturing parameters of the motor. When the design parameters of the motor are known, it is possible to create a model of the motor in the simulation environment. Creating the model of the motor is very important for analyzing the operation of the motor. The outputs produced by the referenced inputs of the motor can be analyzed thanks to this model.

The BLDC motor used in this study consists of a rotor and a stator with three phases. The motor phases can be connected in a star or delta configuration. The motor to be modeled is a star-connected BLDC motor. The electrical circuit of the motor is shown in Fig. 2.

Fig. 2 In the equivalent circuit shown in Fig. 2, *ra, rb,* and r_c are the stator phase resistances; L_a, L_b and L_c are the stator phase inductances; L_{ab}, L_{bc} and L_{ca} are the average inductance between the stator phase windings; $e(t)_a$, $e(t)_b$ and $e(t)_c$ are the opposing electromotive forces;

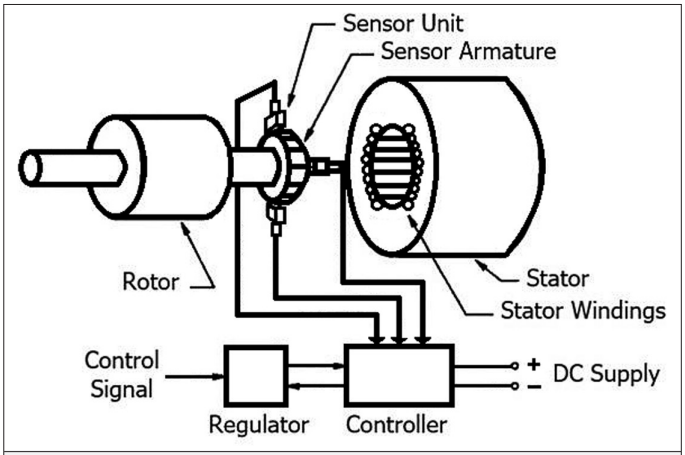


Fig. 1. Control brushed direct current motor.

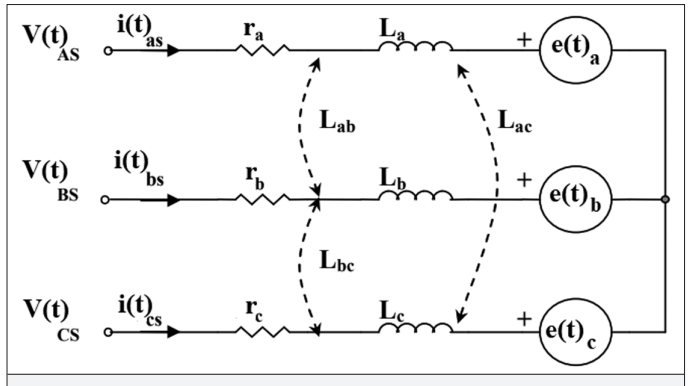


Fig. 2. Equivalent circuit of a star-connected contro brushed direct current motor.

$V(t)_{AS}$, $V(t)_{BS}$ and $V(t)_{CS}$ are the stator phase voltages; $i(t)_{as}$, $i(t)_{bs}$ and $i(t)_{cs}$ are the stator phase currents [17].

For the motor whose equivalent circuit is shown in Fig. 2, the back-emf value "$e(t)_{abc}$" is shown in terms of the back-emf constant "K_e" and motor angular speed "$\omega(t)$" with (1).

$$e(t)_a = K_e\omega(t) \tag{1}$$

In the equation shown in (1), the back emf value is expressed, and the electromagnetic torque value can also be expressed by a constant of its own. The equation for the electromagnetic torque value is shown in (2).

$$T_e = K_t i(t) \tag{2}$$

In the equation presented in (2), the electromagnetic torque equation is obtained with the torque constant "K_t" and current "$i(t)$". The electromagnetic torque equation of the motor is shown in (3).

$$T_e - T_L = J\frac{d\omega(t)}{dt} + D\omega(t) \tag{3}$$

In the equation given by (3), J represents the inertia value and D represents the viscous damping.

The voltage equation of the circuit diagram shown in Fig. 2 is given by (4) for any phase.

$$u(t) = r_x i(t) + L_x\frac{di(t)}{dt} + K_e\omega(t) \tag{4}$$

In equation (3), $u(t)$ is the battery voltage, $i(t)$ is the current to the windings, r_x is the phase resistance, *and* L_x is the phase inductance.

In the motor model, it is desired that the input is voltage and the output is angular velocity, so that the speed control of the motor can be performed more easily. In order to create this model, the transfer function should have a structure given in (5) [3].

$$G(s) = \frac{\omega(t)}{u(t)} = \frac{\omega c}{s^2 + as + b} \tag{5}$$

In equation (2), $i(t)$ is left alone. Using equation (4), the current value can be expressed in terms of rotor inertia, viscous damping, motor angular velocity, and torque constant. If the current value obtained is substituted into the term in equation (4) and Laplace transformed, the voltage expression of the motor is obtained in terms of motor variables. If this equation is transformed into the form shown in (5), the transfer function shown in (6) is obtained [18].

$$G(s) = \frac{K_t}{(L_x s + r_x)(Js + D) + K_t K_e} \tag{6}$$

The transfer function of the BLDC motor is shown in (6). If the parameters in this equation are known, the model of the motor is created. The parameters of the motor used in this study are shown in Table I.

If the values given in Table I are substituted in (6), the mathematical model of the motor is obtained. The obtained motor transfer function is presented in (7).

TABLE I. CONTROL BRUSHED DIRECT CURRENT MOTOR PARAMETERS

Motor Phase Resistance	0.00856537 Ω
Motor Phase Inductance	0.000156957 H
Motor Torque Constant	1.16
Motor Back-EMF[1] Constant	0.11828
Rotor Inertia	0.000386 kg.m^2
Viscous Damping	0.787

[1]Electro magnetic force.

$$G(s) = \frac{1.16}{(0.000156957s + 0.00856537)(0.000386s + 0.787) + 0.1372048} = \frac{Y}{U} \tag{7}$$

III. DESIGN OF CONTROLLERS

There are various methods for motor control. The most widely used control type is PID. Motor control with PID is easier to design than many other methods. A PID controller is based on controlling the system with three different coefficients. The block diagram of the PID structure is shown in Fig. 3.

In the structure shown in Fig. 3, $C(S)$ is the control block, $e(t)$ is the error of the system, and $u(t)$ is the input applied to the system. The mathematical expression of the block shown in Fig. 3 is given by (8).

$$u(t) = K_p e(t) + K_i\int_0^t e(t)dt + K_d\frac{de(t)}{dt} \tag{8}$$

In equation (7), K_p, *Ki,* and K_d are the control parameters of the controller. There are various methods to obtain these parameters. In this study, the controller parameters are calculated separately by the Ziegler–Nichols method, the Linear Quadratic Regulator (LQR), and MATLAB supported Tuning method.

A. Ziegler–Nichols (Open-Loop)

In order to determine the controller coefficients with this method, the unit step input is applied to the equation presented in (6). The system response is monitored. As given in Fig. 4, a tangent line is drawn parallel to and passing through the response curve. The points where the tangent line crosses the time axis and the gain line are marked, and the output graph of the system with dead time is observed. Here, the distance between the leftmost cursor and the left axis is the dead time, T_1. The time from the dead time until the

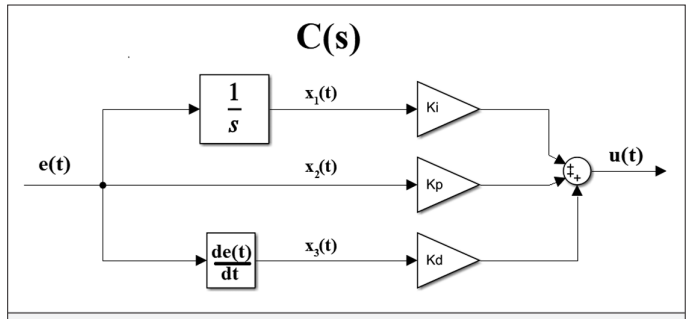


Fig. 3. Proportional integral derivative block internal structure.

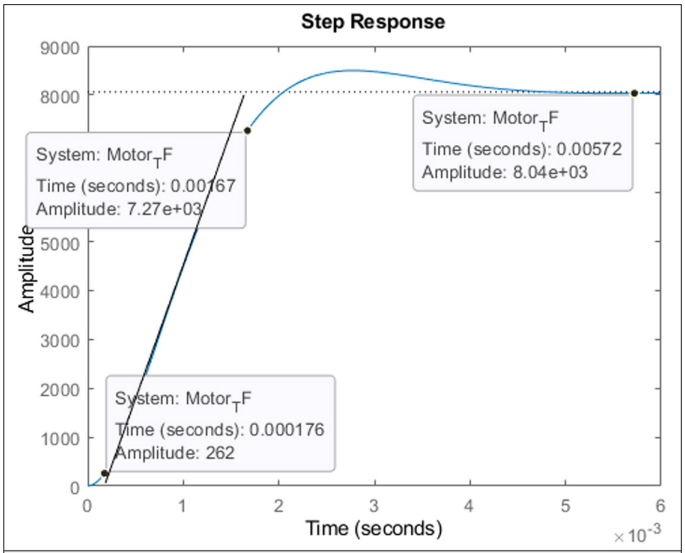


Fig. 4. Ziegler–Nichols open-loop response.

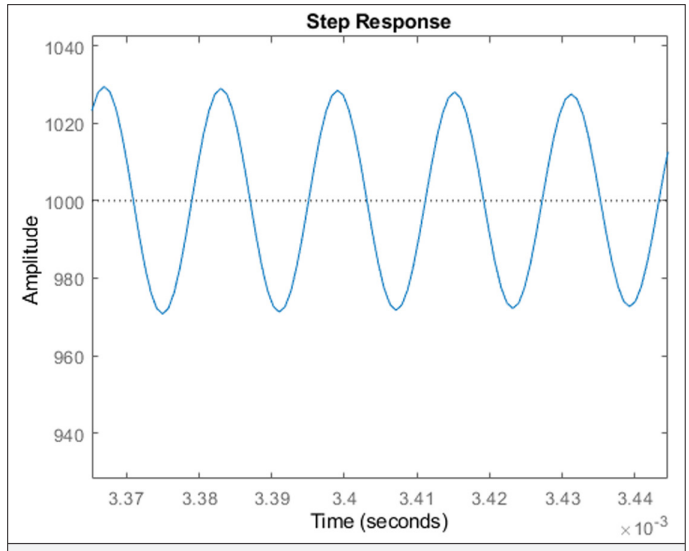


Fig. 5. Ziegler–Nichols closed-loop oscillation signal.

gain value of the system response is reached is expressed as T_2. The gain value of the system response is denoted by K.

Controller coefficients were calculated with $T_1 = 0.000176$ ms, $T_2 = 0.001494$ ms, and $K = 8040$ values obtained from the open loop response. Proportional integral derivative parameters for the Ziegler–Nichols Open Loop method can be calculated by means of equations (9), (10), and (11) [19].

$$K_p = \frac{1.2T_2}{KT_1} = 0.0013 \tag{9}$$

$$K_i = \frac{K_p}{2T_1} = 3.5993 \tag{10}$$

$$K_d = K_p \frac{T_1}{2} = 1.1150 \times 10^{-7} \tag{11}$$

Given in (9), (10), and (11), the controller parameters obtained by Ziegler–Nichols Open-Loop are calculated for the controller.

B. Ziegler–Nichols (Closed-Loop)

In order to determine the parameters of the controller with the Ziegler–Nichols Closed-Loop method, the set value is applied as a closed loop to the equation presented in (7). The gain K_p of the input value applied with Closed-Loop is increased until the step response oscillates. Integral and derivative gains are taken as 0. When the system reaches oscillation, the oscillation period is expressed by "P_u." The value of "K_p" when oscillation is reached is expressed by "K_u." The oscillation signal is shown in Fig. 5.

For the PID controller, parameters K_p, K_i, and K_d must be obtained. The P_u value is calculated from the signal shown in Fig. 5, as shown in (12).

$$P_u = 0.0034155 - 0.003399 = 0.000156 \tag{12}$$

The K_p value of the input applied to the system to obtain the oscillation signal shown in Fig. 5 is 798.3. The controller parameters are obtained from equations (13), (14), and (15) [19].

$$K_p = 0.6K_u = 478.98 \tag{13}$$

$$K_i = 2\frac{K_p}{P_u} = 6140769.2307 \tag{14}$$

$$K_d = K_p \frac{P_u}{8} = 0.00934 \tag{15}$$

C. MATLAB Tuning

The MATLAB program allows the user to automatically tune the controller coefficients using the Tuning Toolbox provided within the program. The controller coefficients were obtained using the tuning structure provided by the MATLAB Simulink interface. The obtained controller coefficients are given in (16), (17), and (18).

$$K_p = 0.27037 \tag{16}$$

$$K_i = 282.408 \tag{17}$$

$$K_d = 6.39956 \tag{18}$$

D. Linear Quadratic Regulator (Proposed Method)

Linear Quadratic Regulator is an optimal control method used in many different engineering sciences [20]. In the PID block shown in Fig. 3, $x_1(t)$, x2(t), and $x_3(t)$ are shown. These are the state variables of this system. The reference input applied to the system does not affect the controller design, so the reference input is assumed to be $r(t) = 0$[21]. Thus, the error value for this closed-loop system is obtained as given in (19).

$$e(t) = -y(t) \tag{19}$$

The equation given structurally by (20) is obtained by inserting the error value given by (19) into the transfer function given by (6).

$$G(s) = \frac{c}{s^2 + as + b} = \frac{-E(s)}{U(s)} \tag{20}$$

Adding c and $U(s)$ in equation (20) gives equation (21).

$$-cU(s)=\left[s^2+as+b\right]E(s) \tag{21}$$

If the equation presented by equation (21) is rewritten in the time domain, (22) is obtained.

$$-cu(t)=\ddot{e}(t)+a\dot{e}(t)+b(e(t) \tag{22}$$

Combining the state variables shown in Fig. 3 with (22) gives (23).

$$-cu(t)=\dot{x}_3(t)+ax_3(t)+bx_2(t) \tag{23}$$

The state space model of equation (23) is shown in (24).

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & -b & -a\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ -c\end{bmatrix}u(t) \tag{24}$$

The equation given by (7) is again obtained by setting the coefficient of s^2 to 1. Substituting the values *a*, *b*, and *c* of the new transfer function into the equation given by (24), the equation given by (25) is obtained.

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & -2092.7545 & -2374979.3695\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ -19145073.4445\end{bmatrix}u(t) \tag{25}$$

The equation given by (25) can be considered a form of a classical state space model. A classical state space model is given by (26).

$$\dot{x}=Ax+Bu \tag{26}$$

Taking (25) and (26) together, matrix A and matrix B of the system can be obtained. To determine the coefficients with LQR, the Riccati equation given by (27) is used [3].

$$-\dot{P}=A^TP+PA-PBR^{-1}B^TP+Q \tag{27}$$

In the given equation (27), *Q* and *R* matrices are weight matrices. The Q matrix is a positive or positive semidefinite, real, symmetric, constant matrix with dimensions 3 × 3. It is composed of diagonal elements. The *R* matrix is a positive, real, real-definite, symmetric control vector weight matrix with dimensions 1×1. The matrices *A* and *B* are the state matrices given by (25). To obtain the *Q* matrix, the flowchart shown in Fig. 6 was run in the MATLAB environment.

The parameters of the Q-matrix were obtained using the flowchart shown in Fig. 6. The Q-matrix obtained is shown in (28).

$$Q=\begin{bmatrix}100 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{28}$$

The only unknown value in the Ricatti equation is the value of *P*. The value of *P* is solved with the help of other known values, and the *K* vector containing the input signal *u*(*t*) and the controller coefficients can be obtained as mentioned in [3]. The *K* vector obtained is a vector containing the control coefficients of the system. The *K* vector is given in (29). The values given in (29) are the controller coefficients obtained by the LQR method for the control of the system.

$$K=\begin{bmatrix}k_i & k_p & k_d\end{bmatrix}=\begin{bmatrix}10 & 5.3546 & 0.9999\end{bmatrix} \tag{29}$$

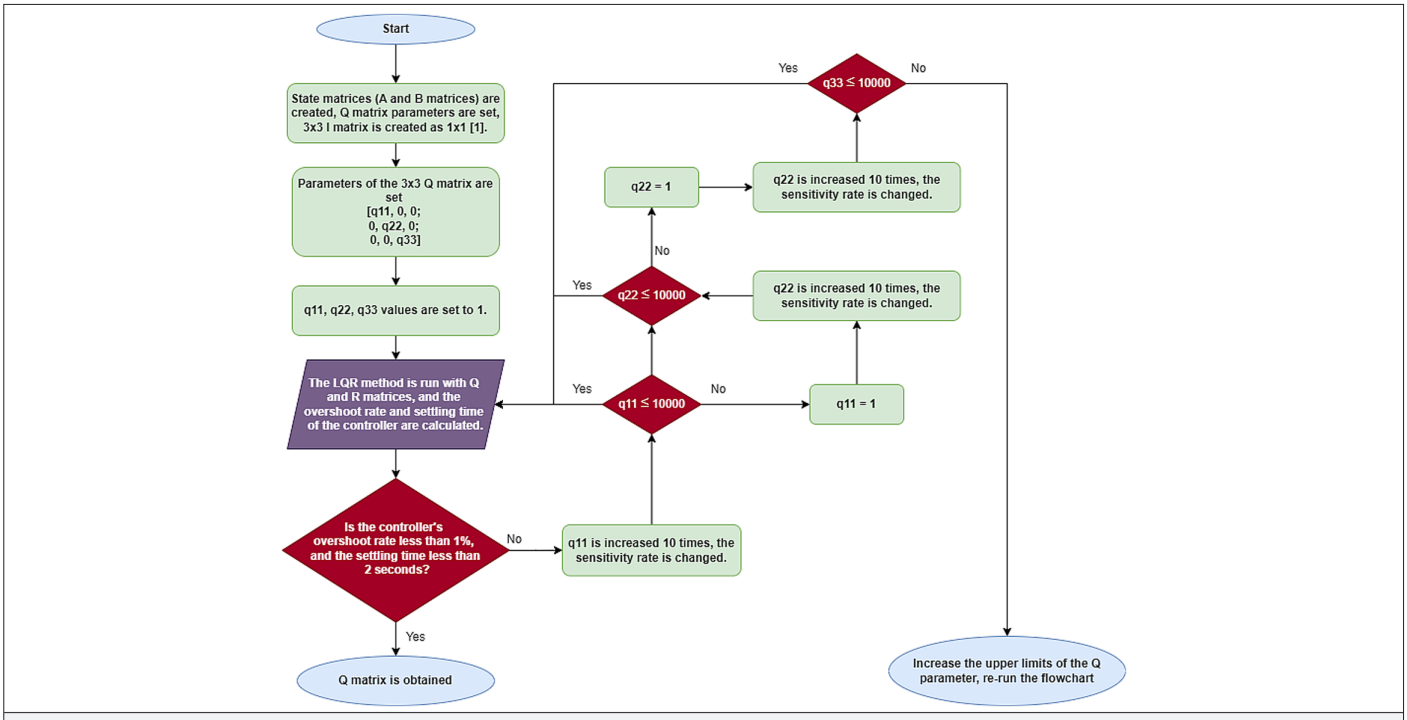


Fig. 6. Flowchart for obtaining the parameters of the *Q*-matrix.

IV. CODE GENERATION

The model-based design of the motor drive system in the Simulink environment was performed after the control system required for the motor drive structure was designed. The STM32-MAT target was used for the design in the Simulink environment [3]. The microprocessor used was an ARM-based STM32F407VGTx. Firstly, a configuration file for the project was created from the STMCubeMX interface. Fig. 7 shows the pinout of the processor.

All pins that are actively used for microprocessor configuration and can be used in future studies have been activated. "Channel 1," "Channel 2," and "Channel 3" are enabled for the ADC (Analog-Digital Converter) structure. "Channel 1" and "Channel 3" are created to read a single data point over a single input. "Channel 2" is configured to read three different analog data points from three different inputs, "IN1," "IN3," and "IN5." To enable pin activation for reading multiple analog data points on a single ADC channel, it is necessary to separate the analog channel with "Rank." The parameter settings for the "Rank"-separated ADC configuration are given in Table II. The ADC configuration is completed by activating the ADC channel for "Channel 1" and "Channel 3."

The clock frequency of the microprocessor used is 168 MHz. The clock settings have been made accordingly. Timer 3 (TIM3) is integrated into "APB1" and Timer 1 (TIM1) is integrated into "APB2" [22]. The timer configuration is shown in Table III.

The clock frequency for "APB1" is 42MHz [22]. In the configuration shown in Table III, the choice of prescaler values for "APB1" and "APB2" is very important. If they were lower, the maximum frequency for "APB1" would be exceeded. The system would not work correctly.

The configuration file created with the STM32CubeMX was included in the Simulink project. The motor control system was divided into subsystems, and blocks were created within these subsystems. The model in which the motor control system software is created is shown in Fig. 8.

The configuration block is subsystemized in the Simulink project with its input and output blocks. Read and write blocks collected under a single subsystem are activated according to the pinout shown in Fig. 7. The instantaneous speed information of the motor is important information for motor control. The instantaneous speed information of the motor is provided by a tachometer that produces an analog output. The tachometer can read up to 10000 rpm with a voltage in the range of 0V - 10V. The maximum target speed value is 1000 rpm. Considering that the ADC of the microprocessor has a maximum power input of 3V, the tachometer produces a maximum voltage of 1V for the motor to be measured. The speed information of the motor is therefore received via the ADC channel. To achieve the motor speed desired by the driver, the speed information was obtained from the driver using a potentiometer. The resulting Motor RPM (Revolutions Per Minute) subsystem is shown in Fig. 9.

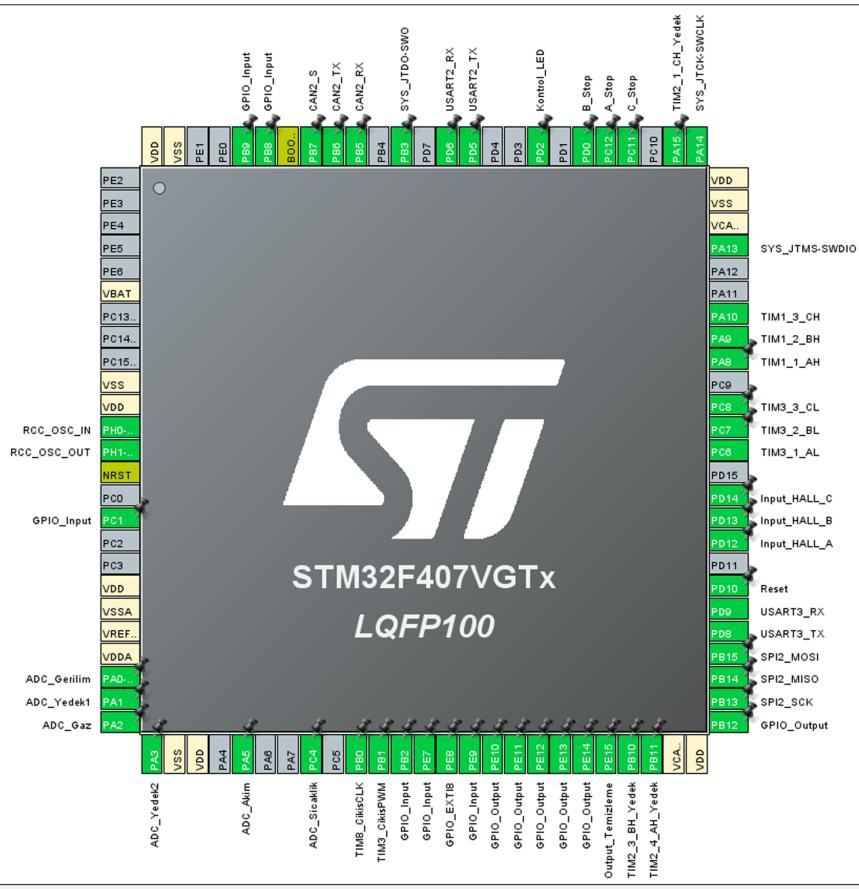


Fig. 7. STMCubeMX microprocessor pinout.

TABLE II. ADC CONFIGURATION PARAMETERS SETTING

ADCs Common Settings	Mode	Independent mode
ADC[1] Setting	Clock Prescaler	PCLK2[2] divided by 2
	Resolution	12 bits (15 ADC Clock cycles)
	Data Alignment	Right alignment
	Scan Conversion Mode	Enabled
	Continuous Conversion Mode	Disabled
	Discontinuous Conversion Mode	Disabled
	DMA[3] Continuous Requests	Disabled
	End Of Conversion Selection	EOC[4] flag at the end of signal channel conversion
ADC Regular Conversion Mode	Number Of Conversion	4
	External Trigger Conversion Source	Regular Conversion launched by software
	External Trigger Conversion Edge	None
	Rank	4
ADC Injected Conversion Mode	Number Of Conversions	0
WatchDog	Enable Analog WatchDog Mode	Disabled

[1]Analog digital converter.
[2]Pixel clock.
[3]Direct access memory.
[4]End of conversion.

The data received from the subsystem shown in Fig. 9 is sent to the Motor Driver Control subsystem together with the data received from the Hall Sensor subsystem. The Motor Driver Control subsystem is a subsystem that generates phase triggers with the received speed and sensor data. The Motor Driver Control subsystem is shown in Fig. 10.

The motor driver control subsystem (MDCS) is a subsystem that contains several subsystems. Hall sensor information is given as input to a subsystem called the "Decoder" in the MDCS, which is used for decoding. The Decoder subsystem produces EMF (Electro-Magnetic Force) outputs of the instantaneous values of the phases. These outputs are then passed to the "Gates" subsystem. The Gates subsystem is a subsystem that transfers the EMF outputs to the output as shared in Table IV.

TABLE III. TIMER CONFIGURATION PARAMETERS SETTING

High-Speed Clock	168MHz
APB1[1]	Prescaler → /4
APB2	Prescaler → /4
Counter Prescaler	420
PWM[2] Pulse	100
Counter Period	65535

[1]Advanced peripheral bus 1, advanced peripheral bus 2.
[2]Pulse width modulation.

The gate outputs obtained by the Gates subsystem are controlled by the ADC Control subsystem, which processes the motor data and outputs Logic1 when appropriate. Current, temperature, and battery voltage are controlled by the ADC Control subsystem. If the current value exceeds 350A, the temperature value exceeds 80°C, or the battery voltage drops below 100V, it is designed to switch off the motor commutation and stop the motor operation. The ADC Control subsystem is shown in Fig. 11.

In the subsystem shown in Fig. 11, the "u1" input to the condition block is the current input. The CR5220(s) current sensor is used to read the "Channel 1" channel of the "ADC2" port. The 4-20 mADC output of the sensor is followed by a 100-ohm resistor and brought to a voltage level in the range of 0.4V - 2V. According to the catalogue information of the sensor, the temperature on the resistor does not affect the reading [23]. Voltage information can be read with a 100-ohm resistor. With a current flow of 0.1A through the transmission line, the sensor produces 0.4V, and with a current flow of 600A, it produces a 2V output value. This shows that approximately 600A will be measured in the 1.6V voltage range. If the 0.4V voltage is subtracted from the value read via the "ADC2-CH1" channel and the remaining expression is multiplied by the 600A/1.6V coefficient, the ampere equivalent of the analog value read is calculated. The "u1" input is designed in this way.

Another condition block in the ADC Control subsystem is the "u2" input. The "u2" input is designed to read the voltage level of the battery. The battery voltage divided by the voltage divider is taken via channel 5 of the "ADC2" port. The battery voltage is formed by four 22-ohm resistors and one 2.2-ohm resistor. The value coming from

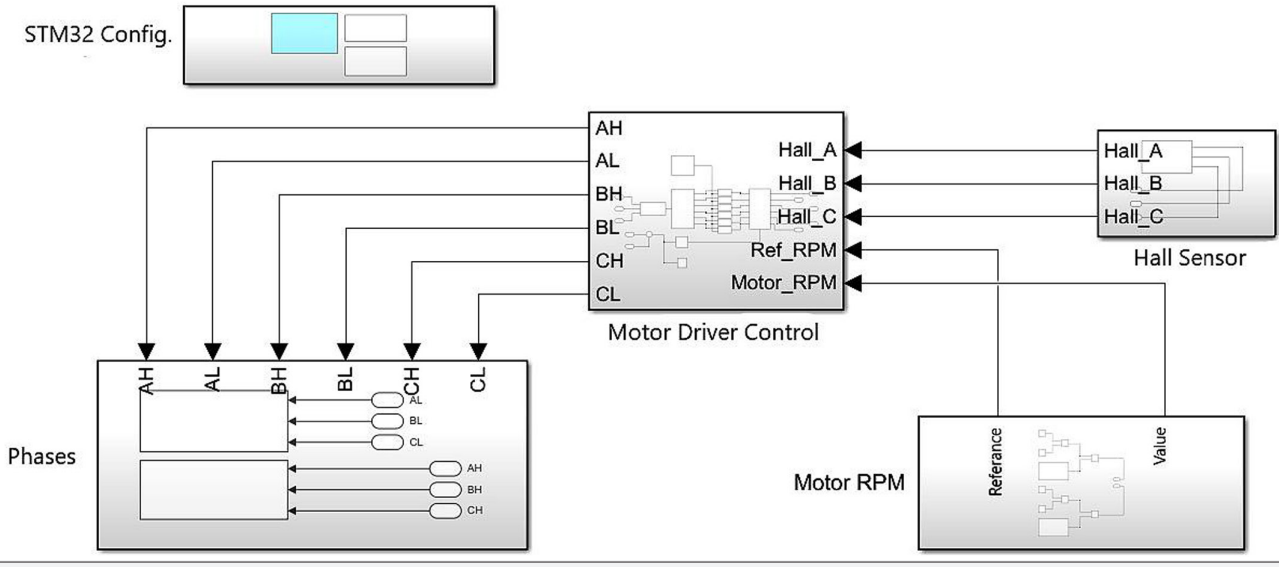


Fig. 8. Motor project block model.

the voltage divider is expanded with 90.2 ohm/2.2 ohm = 41, and the voltage value of the battery is obtained.

The last analog value, i.e., the input where the temperature information of the motor is obtained, is the input obtained with "Channel 14" of the "ADC2" port, which is entered into the condition block with "u3". The temperature sensor connected to channel

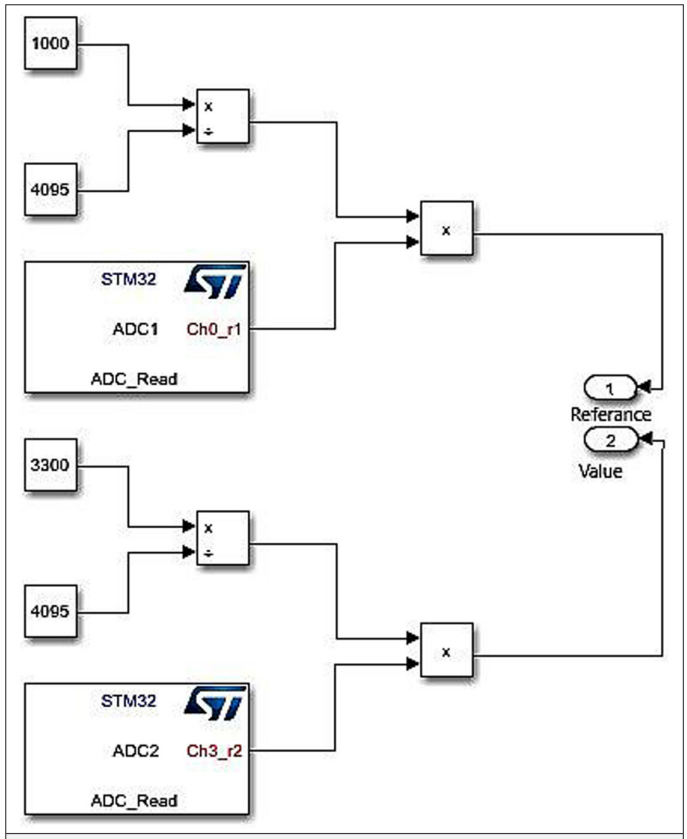


Fig. 9. Motor RPM subsystem.

"ADC2-CH14" is the LM35 (Texas Instruments, TO-220) temperature sensor. The temperature sensor requires a supply voltage in the range of 0-5V. However, the output value produced by the sensor for 150°C is given as 1.5V in the catalog information. Therefore, the analog value will not exceed the range of 0-3.3V and will not damage the processor. The analog value read from the sensor was processed according to the resolution of the ADC channel. This is how the sensor output voltage is calculated. According to the catalog information, the read voltage value is multiplied by 1000 to obtain the decimal temperature value, and the actual temperature value can be obtained by dividing the decimal value obtained by 10 [24]. For this reason, the analog sensor data read in the 0-3.3V range was multiplied by a factor of 100 to obtain the temperature value read by the sensor.

As the ADC block is configured as 12-bit, the values read across the channels are in the range 0-4095. All incoming analog data was divided by 4095 and multiplied by a voltage value of 3.3V. This gives the voltage level at the analog input [12].

In the pulse generator subsystem shown in Fig. 12, trigger information is generated via MOSFETs (Metal–Oxide–Semiconductor Field-Effect Transistor). The occupancy rate at which the generated PWM (Pulse-Width Modulation) signal is obtained from the output of the PID block, which is input to number 7. The outputs generated by the Gates subsystem are processed with the outputs of the ADC Control subsystem, and the control is provided and included in the Pulse Generator subsystem. Input number 7 processes the occupancy rate and phase triggering information after the control and passes it to the PWM generator blocks. This determines which MOSFETs are triggered at which duty cycle. The outputs of the Pulse Generator subsystem are also the outputs of the MDCS.

The outputs from the MDCS are passed to the Phases subsystem shown in Fig. 8. This subsystem activates the TIMER outputs according to the given phase occupancy rates. The "TIM1" channel is used for high-level MOSFETs, and the "TIM3" channel is used for low-level MOSFETs. The configurations of "TIM1" and "TIM3" channels are shown in Table III.

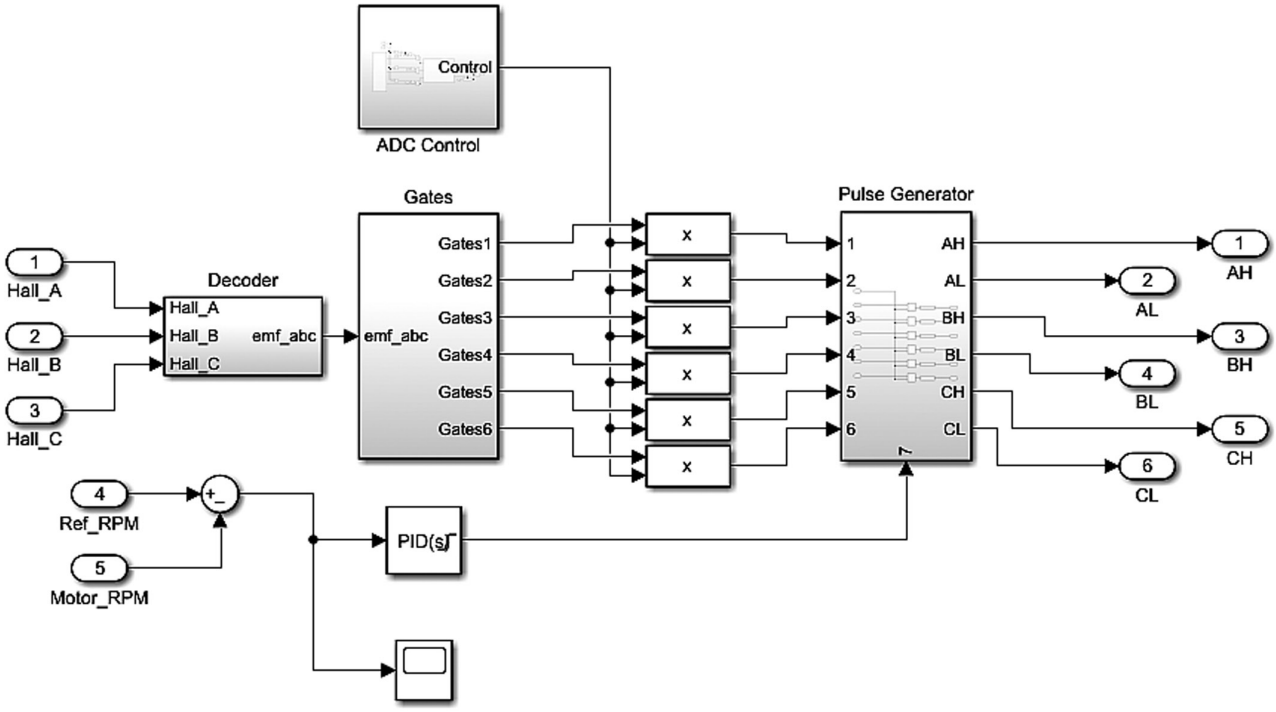


Fig. 10. Motor driver control subsystem.

The generated project was converted into code using STM32-MAT Target. The steps for converting the generated project into code are described in detail in [3].

V. SIMULATION RESULTS AND DISCUSSION

The BLDC motor parameters shown in Table I are controlled in the MATLAB/Simulink interface with the block diagram shown in Fig. 13. For the control, the controller parameters obtained by 4 different methods were separately applied to the system, and the results were monitored. All the control parameters obtained for the motor model are shown in Table V.

Torque (TLoad) is used as the disturbance in the block diagram shown in Fig. 13. To calculate the torque value, parameters such as vehicle weight, rolling resistance, wind friction, speed, frontal area of the vehicle, acceleration time between 0 km/h and 100 km/h, and gradient are required. As mentioned in [3], all these values were obtained specifically for the vehicle structure, and the torque was calculated with these values. As mentioned in [3], the torque required by the vehicle model was determined to be 70 Newtons. The load determined as 70*N* was applied to the model from the fourth second.

TABLE IV. RELATING EMF AND GATES

EMF A	EMF B	EMF C	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	0	0	0
0	−1	+1	0	0	0	1	1	0
−1	+1	0	0	1	1	0	0	0
−1	0	+1	0	1	0	0	1	0
+1	0	−1	1	0	0	0	0	1
+1	−1	0	1	0	0	1	0	0
0	+1	−1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0

The four methods mentioned in the control studies were applied separately to the model, and the speed control of the motor model was performed. The speed control plots obtained using the four different methods are shown in Fig. 14.

If the graph shown in Fig. 14 is examined for all methods, the blue color is used for the Ziegler–Nichols Open Loop method. As can be seen from the blue control signal, the motor speed has reached the set value after approximately 0.3 seconds. The motor has settled to the desired setpoint and is rotating at a speed of 1000 rpm. A 0.01% overshoot and oscillation have occurred.

The control signal generated using the Ziegler–Nichols Closed Loop method coefficients is shown in yellow in Fig. 14. The control signal shown in yellow has reached the desired speed much faster than the Open Loop method. However, the obtained motor speed was generated with very high oscillation compared to the speed information obtained with the Open Loop method. The motor structure, which is expected to have a speed of 1000 rpm, oscillates in the range of 1107 ~ 779 rpm. This result shows that the control signal obtained with Ziegler–Nichols Closed Loop deviates from the desired motor speed value by approximately +10.7% and −22.1%. The results show that the control signal generated by the Ziegler–Nichols Closed Loop method does not really try to settle at 1000 rpm, but oscillates around 940 rpm.

The control signal obtained by tuning is shown in green in Fig. 14. The control signal shown in green indicates that the motor reached

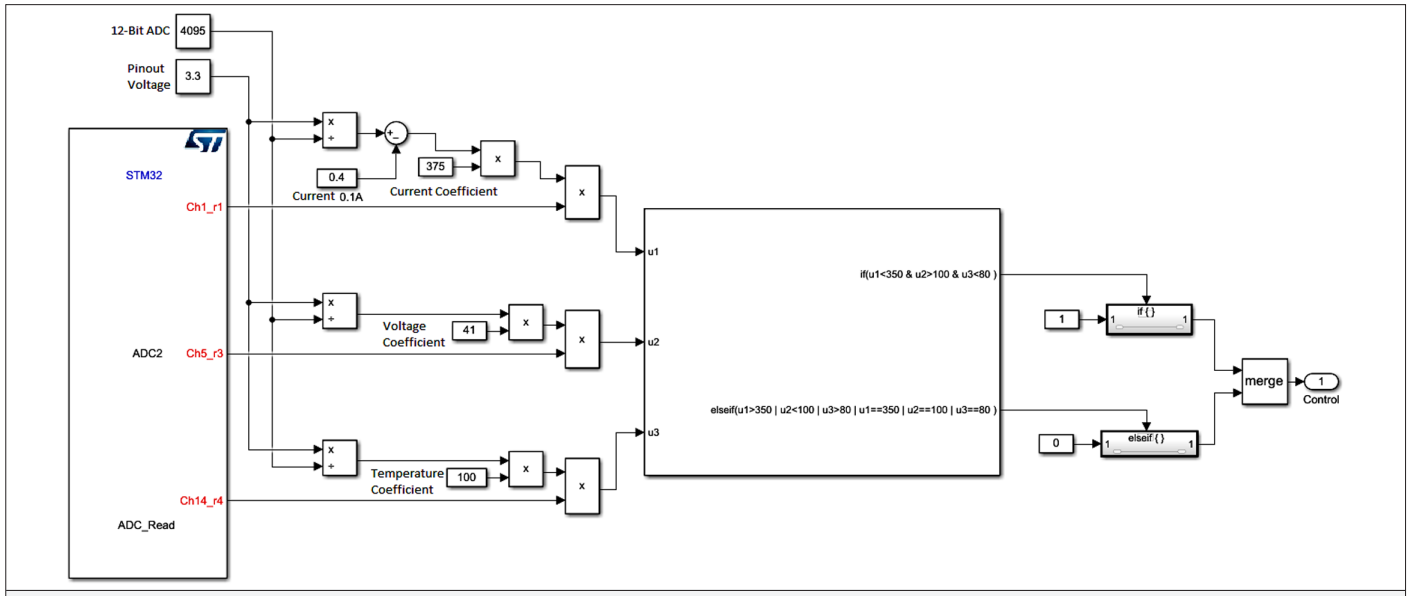


Fig. 11. ADC control subsystem.

1000 rpm after 0.02 seconds, but it reached 1000 rpm with a 15% overshoot. The overshoot rate was high, but the oscillation rate remained low. The speed output of the motor structure was controlled with an oscillation rate of 0.1%.

The control signal generated by LQR is shown in purple in Fig. 14. The purple control signal reached the desired speed more slowly than the Ziegler–Nichols method. The motor speed obtained with LQR oscillated 10 times more compared to the Ziegler–Nichols Open Loop method and 164 times less compared to the Ziegler–Nichols Closed Loop method. The motor structure, expected to have a speed of 1000 rpm, oscillates in the range of 1001 ~ 999 rpm. This result shows that the control signal obtained with LQR deviates from the desired motor speed value by approximately 0.1%.

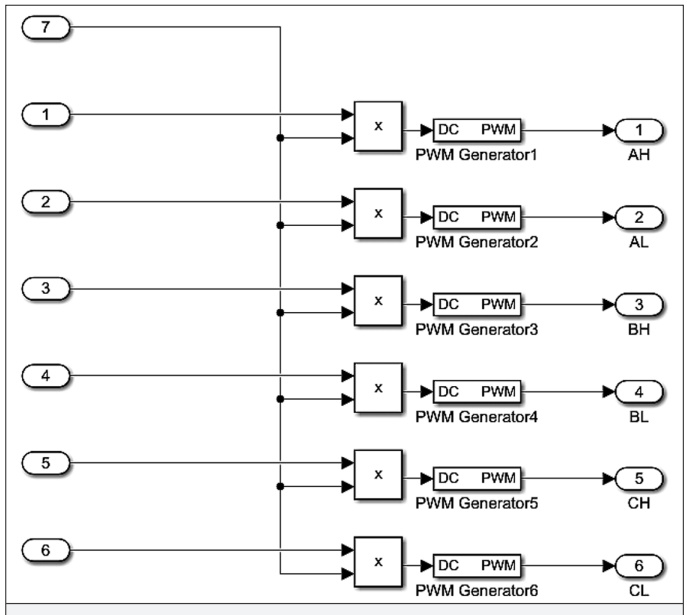


Fig. 12. Pulse generator subsystem.

For all four methods, the time taken for the motor to reach the required speed and the accuracy rate at the required speed are shown in Table VI.

The simulation of the motor model was analyzed not only in the no-load case but also in the case of a disturbing load input to the motor. In the diagram shown in Fig. 13, it is mentioned that the disturbance input to the motor is provided by TLoad. The oscillation on the graph caused by the disturbance applied at the 4th second is shown in detail for all controller methods in the graph shown in Fig. 15.

Looking at the graph in Fig. 15, the 70*N* torque applied to the motor as a disturbance effect at the 4th second was controlled within approximately 0.07 seconds by Tuning, LQR, and Ziegler–Nichols Open Loop methods, and the motor speed remained at 1000 rpm with the same error rate. As can be seen in Fig. 15, the LQR method reacted faster to the disturbance effect.

The results of the load condition of the three methods are shown in Table VII. The Ziegler–Nichols Closed Loop method is not shown in Table VII due to its high oscillation.

While the speed control of the BLDC motor model is provided, the voltage applied to the motor generated by the controller signal is also a value that must be taken into consideration. This applied voltage must remain within the limits of the battery voltage. The voltages applied to the system to obtain the 1000 rpm speed value are shown in a single graph for each control method in Fig. 16.

In the graph shown in Fig. 16, the light blue marker represents the applied voltage curve of the Ziegler–Nichols Open Loop method. For the Ziegler–Nichols Open Loop method, 1000rpm was achieved by applying 124V from the battery to the controller. The 1000rpm value was achieved by staying within the battery voltage limits. In the graph shown in Fig. 16, the yellow color represents the applied voltage graph of the Ziegler–Nichols Closed Loop method. Using this method, it can be seen that continuous oscillation in the range of 0-135V is required to reach 1000rpm and realize the motor drive. This voltage signal is not a feasible control signal. In the graph shown

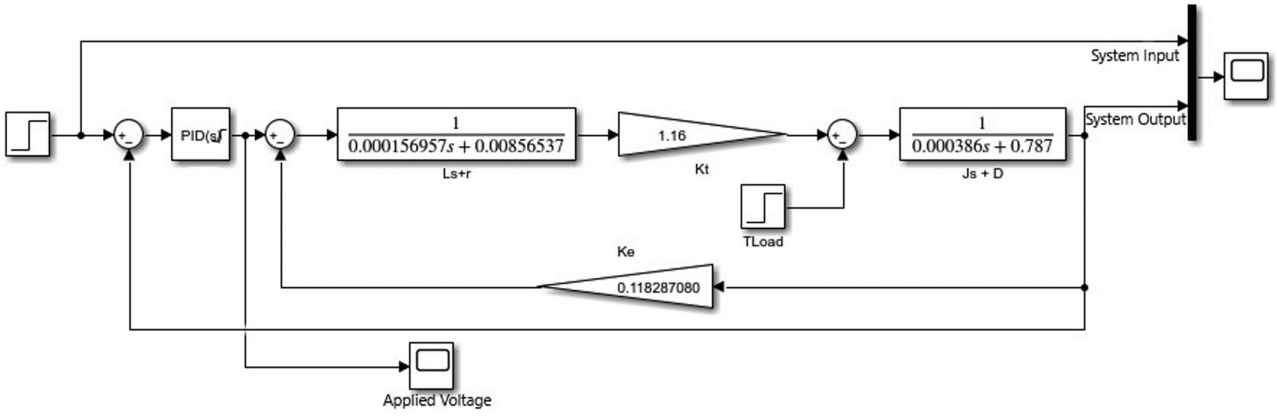


Fig. 13. Control brushed direct current motor simulation block diagram.

TABLE V. CONTROL BRUSHED DIRECT CURRENT MOTOR CONTROLLER PARAMETERS

Methods	K_p	K_i	K_d
Z-N[1] Open-Loop	0.0013	3.5993	1.1150×10^{-7}
Z-N Closed-Loop	478.98	6140769.2307	02.00934
LQR[2]	5.3546	10	0.9999
Tuning	0.27037	2882.408	6.39956

[1]Ziegler–Nichols.
[2]Linear quadratic regulation.

in Fig. 16, the sign shown in purple color represents the applied voltage graph of the LQR method.

The applied voltage required by the LQR control oscillated slightly under load. The voltage curve of the tuning method shown in green is similar to the voltage curve of the Ziegler–Nichols Closed Loop method. In the Ziegler–Nichols Closed Loop method, the oscillation is much higher. In the tuning method, the oscillation is less for the required voltage. However, the oscillation in the range of 105 ~ 130V reduces the applicability.

The motor model reaching the set value, the ability of the control model to maintain system stability under the disturbance effect, and the applicability of the voltage value required by the control signal will show us the most successful method.

VI. CONCLUSIONS

Hybrid and electric vehicles have been in high demand in recent years. The demand for these vehicles creates new opportunities for the development and improvement of subsystems. The BLDC motor type enables electric vehicles to operate quietly and with low maintenance costs. Therefore, BLDC motor structures are an important subsystem for both hybrid and electric vehicles. In this study, where the BLDC motor structure is developed, the focus is on improving the system and discovering the most efficient method. Motor modeling was carried out for the BLDC motor structure. Then, controller studies suitable for the model were conducted. The speed control of the motor was achieved using four different methods.

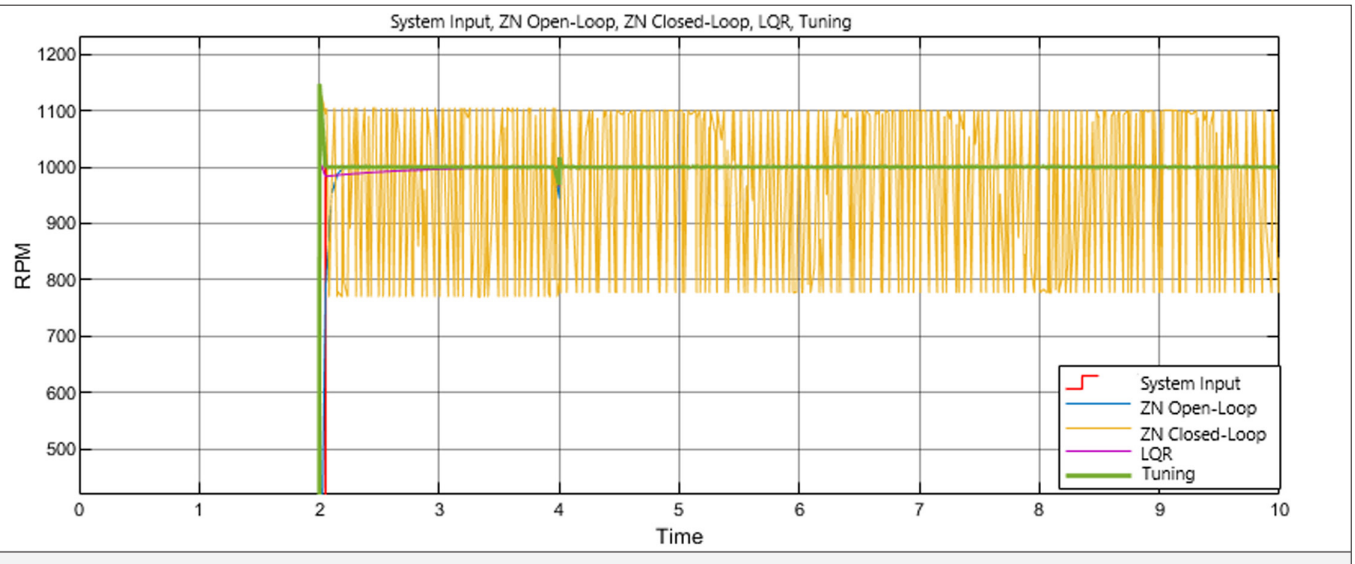


Fig. 14. Control brushed direct current motor RPM response.

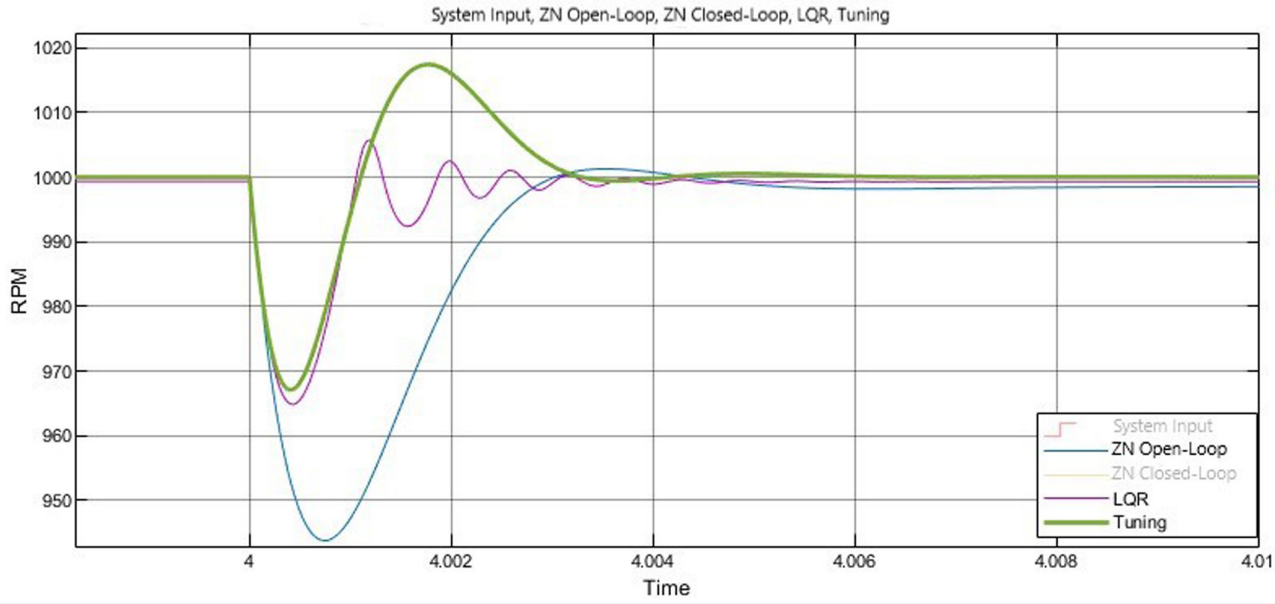


Fig. 15. Control brushed direct current motor applied load.

TABLE VI. COMPARISON OF CONTROLLERS ON SPEED OUTPUT

	Z-N Open-L	Z-N Closed-L	LQR	Tuning
Overshoot	0.01%	10.7%	0.1%	15%
Oscillation	0.01%	16.4%	0.1%	0.1%
Settling Time (sec)	0.3	0.1	1.8	0.02
Settling Value (rpm)	1000	943	1000	1000

TABLE VII. COMPARISON OF SPEED OUTPUTS OF CONTROLLERS AT LOAD CONDITION

	Z-N Open-Loop	LQR	Tuning
Overshoot	0%	0.7%	1.8%
Oscillation	5.5%	3.5%	3.3%
Settling Time (sec)	0.07	0.07	0.07
Settling Value (rpm)	1000	1000	1000

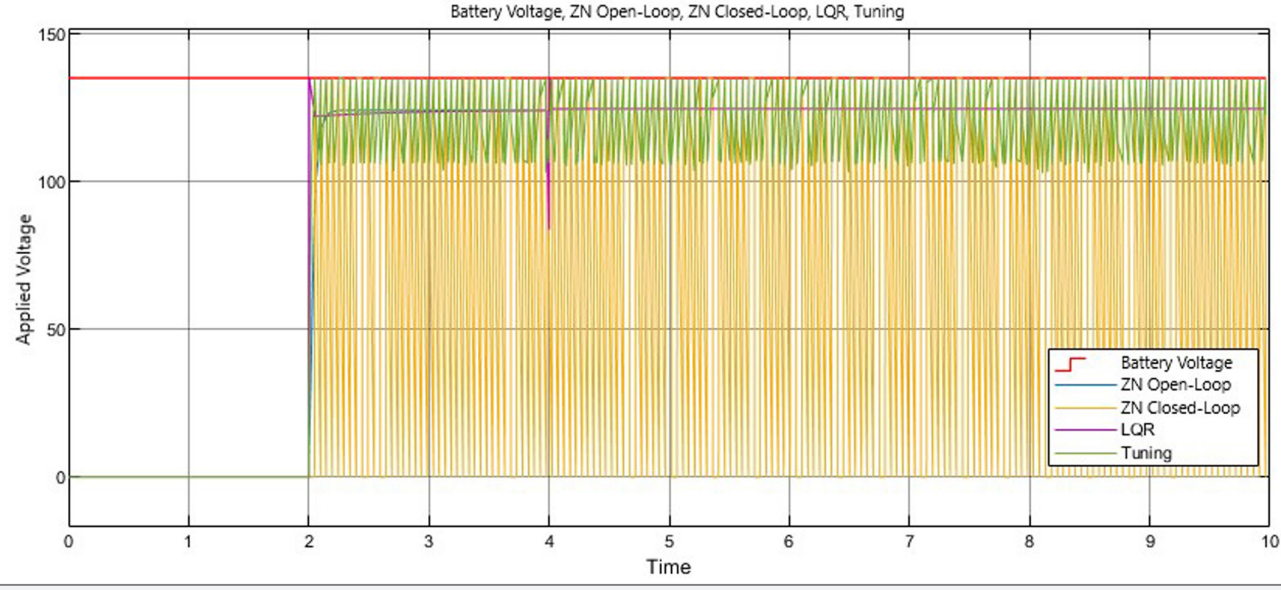


Fig. 16. Control brushed direct current motor applied voltages.

TABLE VIII. PERFORMANCE COMPARISON OF METHODS

	Settling Time (sec)		Accuracy Rate at Settling Value (rpm)		Overshoot		Oscillation	
Methods	**Under Load**	**No Load**	**Under Load**	**No Load**	**Under Load**	**No Load**	**Under Load**	**No Load**
Z-N Open-Loop	0.07	0.3	100%	100%	0%	0.01%	5.5%	0.01%
Z-N Closed-Loop	Too much	0.1	Too little	94.3%	Too much	10.7%	Too much	16.4%
LQR	0.07	1.8	100%	100%	0.7%	0.1%	3.5%	0.1%
Tuning	0.07	0.02	100%	100%	1.8%	15%	3.3%	0.1%

The results obtained for four different speed controllers are shown in Table III. Considering the minimum oscillation and overshoot rate at the desired speed value of the motor, it can be seen that the most efficient methods are the Ziegler–Nichols Open Loop Response, LQR, and Tuning methods. While Z-N Open Loop Response works with a margin of error of 0.01%, Ziegler–Nichols Closed Loop works with a margin of error of 16.4%, LQR works with a margin of error of 0.1%, and Tuning works with a margin of error of 0.1%. In addition, the Ziegler–Nichols Closed Loop method realized these oscillations 5.7% below the desired speed value. The controllers obtained with LQR, Ziegler–Nichols Open Loop method, and Tuning method realize the oscillation at the desired value of 1000 rpm. However, the controller produced by the Ziegler–Nichols Open Loop method has a disadvantage in terms of settling time compared to the LQR and Tuning methods. The Ziegler–Nichols Open Loop method settles to the desired speed value with a slowness that is 3 to 15 times slower than the Ziegler–Nichols Closed Loop and Tuning methods. All these comments can be made for the motor running at no load. In order to evaluate the motor control under load, it is necessary to consider the analysis results presented in Table VII. In Table VII, although all three methods reached the desired 1000 rpm value at the same time, the LQR method started to oscillate quickly above 1000 rpm and settled quickly at 1000 rpm. The signal obtained with Tuning showed a higher overshoot compared to LQR and Ziegler–Nichols Open Loop. Tuning and LQR methods settled at 1000 rpm with a speed reduction close to each other. The Ziegler–Nichols Open Loop method showed approximately 1.61 times more speed reduction compared to these two methods. The results of the Ziegler–Nichols Closed Loop method are not shown in Table VII due to the high level of oscillation. The performance comparison for both under load and no load conditions is presented in Table VIII. The voltage value applied to the motor should not show high oscillation and should be realistic. Looking at the applied voltage values shown in Fig. 16 in the simulation studies of the motor structure, it can be seen that the oscillation values of the voltages required by the controller created by the Tuning method and the Ziegler–Nichols Closed Loop method are very high. It is not possible or healthy to apply the battery voltage to the system at this level of oscillation. Therefore, the Tuning method and the Ziegler–Nichols Closed Loop method are not advantageous for this model in terms of applicability. When the Ziegler–Nichols Open Loop method and the LQR method are analyzed, the fast response of the Ziegler–Nichols Open Loop method reveals the idea that the Ziegler–Nichols method is the best method, but the LQR method is the method that best preserves the speed of the motor. As shown in Table VIII, the LQR method is the method that achieves the speed with the least reduction in speed compared to the Ziegler–Nichols Open Loop method when the motor is under load. The LQR method was most successful in maintaining the motor speed.

As a result, the LQR method is found to be the most successful method for BLDC motors used in hybrid electric vehicles. In the future, machine learning-based studies are planned for the control of BLDC motors used in hybrid electric vehicles and the optimization of controller parameters.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – S.I., B.O., H.T.; Design – S.I.; Supervision – B.O., H.T.; Resources – H.T., B.O.; Materials – S.I., B.O., H.T.; Data Collection and/or Processing – S.I.; Analysis and/or Interpretation – S.I., B.O., H.T.; Literature Search – S.I., H.T.; Writing – S.I.; Critical Review – S.I., B.O., H.T.

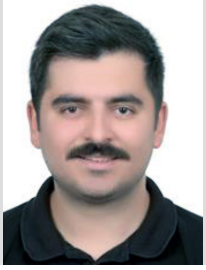
Declaration of Interests: The authors have no conflict of interest to declare.

Funding: The authors declared that this study has received no financial support.

REFERENCES

1. P. Day, *The Philosopher's Tree*. Boca Raton, United States of America: CRC Press, 2020. [CrossRef]
2. M. Miyamasu, and K. Akatsu, "Efficiency comparison between Brushless dc motor and Brushless AC motor considering driving method and machine design," in IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, IEEE, Nov. 2011, pp. 79–86. [CrossRef].
3. S. İn, *Model-based hardware and software mutual design of A motor control system*. İstanbul: Istanbul Technical University, 2023. [Accessed: Feb. 17, 2024]. [Online]. Available: https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=klrldtdJ31bRgjb6fHvMUY7abrClAEf0LkX3d2wg4wwLFsD0fTQ21nKTXr0qo8VS.
4. A. Kusko, and S. M. Peeran, "Definition of the brushless DC motor," in Conference Record of the 1988 IEEE Industry Applications Society Annual Meeting, IEEE, 1988, pp. 20–22. [CrossRef]
5. E. Landolfi, A. Salvi, A. Troiano, and C. Natale, "Model-based design and processor-in-the-loop validation of a model predictive control for coupled longitudinal-lateral vehicle dynamics of connected and automated vehicles," in 2021 29th Mediterranean Conference on Control and Automation (MED), IEEE, Jun. 2021, pp. 699–705. [CrossRef]
6. H. T. Al-Fiky, M. Sh. Asfoor, M. I. Yacoub, and A.-H. Sharaf, "Speed control modeling for in-wheel permanent magnet brushless DC motors for electric vehicles," in 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR), IEEE, Aug. 2019, pp. 438–443. [CrossRef]
7. A. Ulasyar, H. Sheh Zad, and A. Zohaib, "Intelligent speed controller design for brushless DC motor," in 2018 International Conference on

Frontiers of Information Technology (FIT), IEEE, Dec. 2018, pp. 19–23. [CrossRef]
8. U. Ansari, S. Alam, and S. M. un N. Jafri, "Modeling and control of three phase BLDC motor using PID with genetic algorithm," in2011 UkSim 13th International Conference on Computer Modelling and Simulation, IEEE, Mar. 2011, pp. 189–194. [CrossRef]
9. M. Mahmud, S. M. A. Motakabber, A. H. M. Zahirul Alam, and A. N. Nordin, "Adaptive PID controller using for speed control of the BLDC motor," in 2020 IEEE International Conference on Semiconductor Electronics (ICSE), IEEE, Jul. 2020, pp. 168–171. [CrossRef]
10. S. K. Awaze, "Four Quadrant operation of BLDC motor in MATLAB/SIMULINK," in 2013 5th International Conference on Computational Intelligence and Communication Networks, IEEE, New York: IEEE, Sep. 2013, pp. 569–573. [CrossRef]
11. E. Anitha, and P. Vairaprakash, "Design of dynamic states and digital speed control of BLDC motor," in 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE, Mar. 2016, pp. 1–6. [CrossRef]
12. E. Monmasson, and M. N. Cirstea, "FPGA design methodology for industrial control systems—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007. [CrossRef]
13. P. Suganthi, S. Nagapavithra, and S. Umamaheswari, "Modeling and simulation of closed loop speed control for BLDC motor," in 2017 Conference on Emerging Devices and Smart Systems (ICEDSS), IEEE, Mar. 2017, pp. 229–233. [CrossRef].
14. K. S. Devi, R. Dhanasekaran, and S. Muthulakshmi, "Improvement of speed control performance in BLDC motor using fuzzy PID controller," in 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE, May 2016, pp. 380–384. [CrossRef]
15. N. N. Baharudin, and S. M. Ayob, "Brushless DC motor drive control using Single Input Fuzzy PI Controller (SIFPIC)," in 2015 IEEE Conference on Energy Conversion (CENCON), IEEE, Oct. 2015, pp. 13–18. [CrossRef]
16. W.-J. Tang, and S.-Y. Cao, "A fast realization method of fuzzy PID control for DC motor," in 2018 37th Chinese Control Conference (CCC), IEEE, Jul. 2018, pp. 5131–5135. [CrossRef]
17. T. Yıldız, *Fırçasız doğru akım Motorlarının Modellenmesi ve kontrolü*. Sakarya: Sakarya University, 2009. [Accessed: May 31, 2024]. [Online]. Available: https://acikerisim.sakarya.edu.tr/bitstream/handle/20.500.12619/80948/T04016.pdf?sequence=1.
18. T. Krishna, N. Gowri, G. Babu, B. Akshita, G. Agarwal, and M. Netha, "Modelling and development of controller for BLDC motor," *Int. J. Electr. Eng. Technol. (IJEET)*, vol. 12, pp. 200–210, 2021. [CrossRef]
19. A. Abdulameer, M. Sulaiman, M. S. M. Aras, and D. Saleem, "GUI based control system analysis using PID controller for education," *IJEECS*, vol. 3, no. 1, pp. 91–101, Jul. 2016. [CrossRef].
20. J. Borggaard, and L. Zietsman, "The Quadratic-Quadratic Regulator Problem: Approximating feedback controls for quadratic-in-state nonlinear systems," in 2020 American Control Conference (ACC), IEEE, Jul. 2020, pp. 818–823. [CrossRef]
21. J. B. He, Q. G. Wang, and L. Tong-Heng, "PI/PID controller tuning via LQR approach," in Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171), FL: IEEE, 1998, pp. 1177–1182. [CrossRef]
22. STMicroelectronics, "STM32F405xx STM32F407xx – production data," STMicroelectronics. [Accessed: Dec. 03, 2024]. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f405rg.pdf.
23. CRMagnetics, "DC Current Transducer - CR5200 Series Datasheet," CRMagnetics. [Accessed: Sep. 06, 2024]. [Online]. Available: https://www.crmagnetics.com/Assets/ProductPDFs/CR5200%20Series%201.pdf.
24. TexasInstruments, "LM35 precision centigrade temperature sensors - Production data," Texas Instruments Incorporated. [Accessed: Oct. 05, 2024]. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm35.pdf.



Sedat İn, MSc, is a Research Assistant at İstanbul University-Cerrahpaşa Electrical and Electronics Engineering Department. He received his master's degree from İstanbul Technical University, Department of Electronic Engineering. He continues his PhD education at İstanbul University-Cerrahpaşa. His research interests are Embedded System Design and Control Systems. He continues his studies on electronic board design, modeling, and control of electronic systems.

Berna Örs received the Electronics Communication Engineering degree and the MSc degree in 1995 and 1998, respectively, both from İstanbul Technical University (İTÜ), Turkey. She received the Electrical Engineering degree in applied sciences from Katholieke Universiteit Leuven, Belgium, in 2005. Currently, she is a Professor at İTÜ. Her main research interests include cryptography, embedded systems, and side-channel attacks.

Assoc. Prof. Dr. Hasan Tiryaki is the Head of the Department of Electrical and Electronic Engineering at Kırklareli University. He received his Ph.D. from İstanbul University, Department of Electrical and Electronic Engineering. His research interests include Control Systems, Artificial Intelligence, Thermoelectricity, Renewable Energy Systems, and Railway Systems. He has worked as an executive in projects funded by Tübitak and the European Union. He holds several patents on automotive technologies.