International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 15, No. 2, July 2026, pp. 339∼349
ISSN: 2089-4864, DOI: 10.11591/ijres.v15.i2.pp339-349

Configurable embedded solution for multi-mode motor control

Ufuk Güner
Department of Electrical and Electronic Engineering, Faculty of Engineering and Architecture, Erzurum Technical University, Erzurum, Turkey

Article Info

Article history:

Received Dec 18, 2025
Revised Feb 23, 2026
Accepted May 30, 2026

Keywords:

Brushless direct current motor
Direct current motor
Motor type identification
Multi-mode motor drive
Stepper motor

ABSTRACT

Precise robotic systems often require multiple motor types, which increases hardware complexity, cost, and synchronization effort. This study presents an open-source multi-mode motor control platform based on four half-bridge power stages, enabling direct current (DC), brushless direct current (BLDC), and stepper motor control on a single hardware architecture. Unlike existing software-based multi-mode approaches, the proposed system introduces automatic motor type identification and safe connection verification at the hardware level, requiring only a microcontroller and a integrated power stage. This represents a key novelty of the platform. Experimental validation was performed using three different motor types. The system achieved correct motor classification over repeated identification tests, with no false detections. Position control experiments confirmed stable operation across DC, BLDC, and stepper motor. The results demonstrate that the proposed platform significantly reduces system complexity while providing reliable multi-motor operation in a compact and low-cost structure.

Corresponding Author:

Ufuk Güner
Department of Electrical and Electronic Engineering, Faculty of Engineering and Architecture
Erzurum Technical University
Erzurum, Turkey
Email: ufuk.guner@erzurum.edu.tr

1. INTRODUCTION

Direct current (DC)-driven actuators are used extensively in robotics, electric cars, and automation. The most commonly used actuators in these fields are DC, stepper, and brushless direct current (BLDC)/ permanent magnet synchronous motor (PMSM) motors. These motors require separate drive hardware and software for optimal performance due to different driving principles, current control strategies, and feedback requirements [1]. This situation leads to the construction of complex hardware structures, especially in prototyping and experimental studies where different types of motors are used together.

Hardware-wise, these motors can be driven using different half-bridge power switching unit topologies, disregarding motor power. While DC motors are driven by a simple full-bridge [2], PMSM/BLDC motors require precise current and position feedback along with three half-bridges [3]. On the other hand, stepper motors require two full-bridges and chopper current control [4]. Therefore, although the basic hardware is a half-bridge, each motor type requires different hardware optimization and control infrastructures. Additionally, power differences between the same motor types also necessitate a change in the power stage.

Research on motor drive systems has generally focused on developing control methods specific to

the motor type. Specifically, sensorless and adaptive control strategies for BLDC motors are comprehensively addressed with the goal of achieving high efficiency and dynamic performance [5], [6]. Sliding mode and combined control approaches are recommended for stepper motors [7]. Converter-based adaptive [8] and embedded control software is offered for DC motors [9]. These studies offer significant contributions in terms of control algorithms. However, the studies are based on drive architectures specific to motor type.

In studies on the control of multiple motors, the general aim is to synchronize the driving of multiple same type motors [10]–[12] with the same type system architectures. Such as, FPGA-based multi-motor control platforms for electric vehicles and industrial applications [13]. However, in these systems, it is also assumed that the motor types are homogeneous. Therefore, current multi-motor control studies do not address the issue of operating different motor types together.

A significant portion of the work on systems supporting different motor types is based on software libraries or FPGA-based reconfigurable architectures. Libraries such as SimpleFOC [14] and STM32 motor control SDK [15] offer common control algorithms for DC, BLDC, and stepper motors in microcontroller-based solutions. However, the power driver stages must be designed or added separately according to the motor type. In FPGA-based systems, hardware control cores can be implemented for different motor with selecting the power stage according to the motor type [16], [17]. Although these approaches offer flexibility in control design, critical aspects such as power level configuration, current limiting, motor type identification, and safe connection management are left entirely to the user.

This situation poses a significant limitation, especially for robotic systems where different types of actuators are used together, experimental control studies, and R&D applications requiring rapid prototyping. The requirement for separate drivers and different power topologies for each motor type extends system integration time, increases costs, and makes it difficult to compare and evaluate control algorithms. The best approach to solving this problem is multi-mode motor drive approximation on the same hardware. On the other hand, designing multi-mode motor control on the same platform also brings with it some problems. To ensure optimal control specific to the motor type, hardware deficiencies must be addressed through embedded software. Magnetic zero and direction determination are required for BLDC motors [18]. Current limiting capability must be provided for stepper motors. The control algorithm and motor type must be matched. Even if this match is determined by the user, incorrect entries should be considered. Applying the wrong control mode can lead to serious safety risks, including high current, overheating, and uncontrolled torque generation [19]. It is also quite difficult to support a wide range of motor power.

This study proposes a platform that utilizes an open-source multi-mode motor drive architecture built around a microcontroller. The proposed platform aims to control DC, BLDC, and stepper motors as servos on the same hardware. An integrated circuit is used for driving DC, BLDC, or stepper motors. The integrated circuit contains four half-bridge channels. The integrated circuit provides current feedback from each channel. In the study, moving average filtering is applied within direct memory access (DMA) for fast and effective current measurement. The power stage has a single current limit input for all bridges. A digital-to-analog converter (DAC) is used for dynamic current limiting. In software optimization, special algorithms have been developed for different motor types, considering hardware capabilities. An algorithm for determining the direction and magnetic zero for the BLDC motor was developed using feedback from the encoder. For the stepper motor, closed-loop control with encoder feedback was implemented to overcome the lack of a current chopper. An additional decision algorithm has also been integrated into the system to determine the motor type and activate the appropriate controller. Embedded software supports data collection, parameter adjustment, and real-time simulation integration. In this respect, the proposed system offers a unified hardware-software co-design architecture that supports heterogeneous motor types on a single power stage, in contrast to the common approach in the literature of a dedicated driver per motor type. Furthermore, it allows for the determination of motor type and active channels, which is a weakness of software-based multi-mode motor control solutions. The proposed platform provides a low-cost, fast, and reconfigurable solution for comparative testing of heterogeneous multi-motor mechatronic systems, educational laboratory infrastructure, and control algorithms.

2. HARDWARE CONFIGURATION

The core of the system is the STM32F405RB microcontroller unit (MCU), which provides high processing power and extensive peripheral support. The MCU is responsible for PWM generation, current and voltage measurement, data communication, and executing control algorithms. The system uses the DRV8962DDW

four half-bridge motor driver integrated circuit as the motor power stage [20]. The connections between the MCU and the power stage are arranged to include control and feedback signals. Figure 1 shows design of the multi-mode motor control platform. Motor supply is applied directly to the power stage.

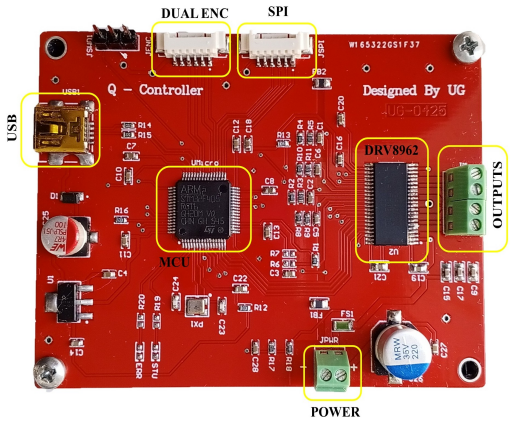


Figure 1. Design of the multi-mode driver

The MCU, on the other hand, is powered independently from the motor driver via USB. Feedback from the motors includes current and bus voltage, as well as position and speed information. There are two quadratic encoder inputs. Additionally, there is an extra input for encoders that has serial peripheral interface (SPI). Encoder data is used in the closed-loop control algorithms. The block diagram of the proposed multi-mode motor controller is shown in Figure 2.

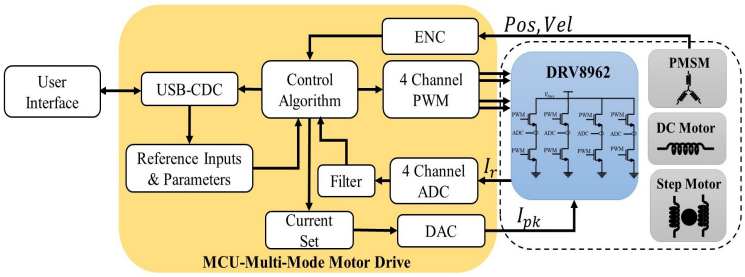


Figure 2. Block diagram of multi-mode motor control

The power stage provides current feedback from the high-side of each half-bridge [20]. The current feedback outputs are connected to the MCU's analog-to-digital converter (ADC) unit. The MCU performs ADC readings using DMA. Measurements are filtered with the moving average on the DMA. The current reference for the power stage is connected to the MCU's DAC output. The MCU can dynamically adjust the current limit via the DAC. The external unit connection of the system is achieved using the MCU's internal USB interface. With the USB connection, the MCU can be connected to MATLAB or similar simulation environments. In this study, a MATLAB user interface is used. Control parameters, target position, current limits, and other parameters can be sent to the platform via the USB interface. Real-time data is received over the same connection.

3. CONTROL ALGORITHMS

The algorithm initially waits for parameters and commands from the user. In this study, the MATLAB environment was used for user input. When commands and parameters arrive from the user, the algorithm first performs the function of determining the motor type and active channels. If there are no errors or incompatible motors at this stage, the motor rotation direction is determined according to the motor connections. Subsequently, the relevant motor control layer is activated. Motor control performs two time-dependent process. The

first process is motor control stage. It is different for each motor and repeats with a period of T_c at $timer_c$. The second function is the data transfer process. It is the same for each motor and repeats with a period of T_m at $timer_m$. The parameters T_c and T_m are user-adjustable.

3.1. Direct current motor control algorithm

Algorithm 1 shows the DC motor drive algorithm. The DC motor is driven by two pulse width modulation (PWM) outputs through the two half-bridges of the power stage. The direction of motor rotation is determined by the polarity of the PWM signals. A linear relationship can be established between motor current, PWM duty cycle, and torque [21]. The control algorithm is based on closed-loop position control. Speed and position information are obtained from encoders. In each control cycle, the difference between the target position (pos_{ref}) and the measured position (pos) is processed by a proportional integral derivative (PID) controller. The controller output is passed to the proportional integral (PI) current control layer as the current reference (I_{ref}). The PI controller output (u_I) and the source voltage (V_{bus}) determine the PWM duty cycle. In this structure, the internal current limit (I_{pk}) of the power stage is adjusted by the DAC. A short pulse and encoder response are used to determine the motor direction. All PID and PI parameters are user-adjustable.

Algorithm 1. DC motor drive

```
I_pk ←Set current limit with DAC
Initialize power stage
Initialize parameters
Detect rotation direction
loop
  if timer_c ≥ T_c  then
    (pos, vel) ← ReadENC()
    ( I_r[], V_bus) ← ReadADC_FDMA()
    I_ref ← CalcPIDPos(pos, pos_ref)
    u_I ← CalcPI(I_r[], I_ref)
    PWM_TIM2 ← CalcPWM(u_I, V_bus)
  end if
  if timer_m ≥ T_m  then
    SendUSBData(I_r[], pos, vel, vbus)
  end if
end loop
```

3.2. Stepper motor control algorithm

The stepper motor drive is shown in Algorithm 2. The hardware has the capability to drive a bipolar stepper motor. However it does not channel-based chopper control. To overcome this deficiency, position feedback and PID control are utilized. In stepper motor control, the four half-bridges of the power stage are connected to the two phases of the motor. Each phase current is modulated according to a sinusoidal reference profile. The motor windings are excited with a phase difference of 90 degrees [22]. It is possible to create micro-step by dividing the phase into smaller steps [23]. For this, a predefined sinusoidal microstep lookup table (LUT) is used. LUT is scalable from 4 to 128 microsteps by the user. The motor shaft angle and speed are measured. The motor angle (pos) and the reference position (pos_{ref}) are passed through the PID controller. The controller output provides the step speed reference (w_{ref}). By converting the speed reference to an angle command (θ_{cmd}), the LUT index (idx) is calculated [24]. The sinusoidal references (s_Q, c_Q) obtained from LUT are converted into PWM ratios. Control and LUT parameters are user-adjustable.

3.3. Brushless direct current motor control algorithm

BLDC control is the most advanced control layer of the system. The BLDC motor control is shown in Algorithm 3. The proposed platform is designed to be suitable for three-phase motor driving. However, the motor's magnetic zero and direction must be determined in the multi-mode control. This can be done in two ways. The encoder start can be mechanically aligned to magnetic zero, or a relative magnetic zero can be determined by applying a $2\pi/3$ radian phase difference pulse. For motor direction, a short pulse and encoder response are used. While the three half-bridges of the power stage directly drive the three phases, field-oriented control (FOC) with position feedback is implemented in the software [25]. FOC includes current and position loops with a cascaded structure. By applying Clarke–Park transformations, the phase currents are transformed to the d–q axis (i_d, i_q). PI-based current controllers generate voltage references (v_d, v_q) in the d-q axis. These

references are converted to three-phase PWM values using the inverse Park (v_α, v_β) transformation [26]. Rotor position and speed are obtained from encoders. All control parameters are user-adjustable.

Algorithm 2. Stepper motor drive

```
I_pk ←Set current limit with DAC
Initialize power stage
Initialize parameters
Generate LUT
Detect rotation direction
loop
  if timer_c ≥ T_c then
    (pos, vel) ← ReadENC()
    ( I_r[], V_bus) ← ReadADC_FDMA()
    ω_ref ← CalcPIDPos(pos, pos_ref)
    θ_cmd ← CalcAngleCMD(ω_ref);
    idx ← CalcLUTIndex(θ_cmd);
    s_Q, c_Q ← LUT(idx)
    PWM_TIM2 ←CalcPWM(s_Q, c_Q, gain, V_bus)
  end if
  if timer_m ≥ T_m then
    SendUSBData(I_r[], pos, vel, V_bus)
  end if
end loop
```

Algorithm 3. BLDC drive

```
I_pk ←Set current limit with DAC
Initialize power stage
Initialize parameters
Detect magnetic zero
Detect rotation direction
loop
  if timer_c ≥ T_c then
    ( I_r[], V_bus) ← ReadADC_FDMA()
    θ_m ← CalcMechanicalAngle()
    i_d_ref, i_q_ref ← CalcPIDPos(θ_m, θ_ref)
    FOC_Update(i_d_ref, i_q_ref)
      i_a, i_b ← I_r[0], I_r[1]
      i_α, i_β ←Clarke(i_a, i_b)
      θ_e ← CalcElectricalAngle()
      i_d, i_q ←Park(i_α, i_β, θ_e)
      v_d, v_q ←CalcPI(i_d_ref, i_q_ref)
      v_α, v_β ←InversePark(v_d, v_q, θ_e)
      PWM_TIM2 ←CalcPWM(v_α, v_β, V_bus)
    end FOC_Update
  end if
  if timer_m ≥ T_m then
    SendUSBData(I_r[], pos, vel, V_bus)
  end if
end loop
```

3.4. Motor type identification algorithm

For motor identification, short-duration and low-duty cycle test pulses are applied to the driver outputs. Feedback currents are measured. The power stage provides feedback from high side of half-bridge. Thus, it is possible to match active channels with current measurement. This current measurements are used to create the current matrix B_{ij}. The index i represents the channel being excited, and j represents the average current value measured from the channel. A relative threshold value ϵ is used for analysis.

$$\epsilon = \delta \cdot (I_{pk_{ref}}), \qquad \delta \in [0.5, 0.9] \tag{1}$$

where, δ is the proximity factor that is used to grantee. The binary response matrix B_{ij} dependent on the threshold value is defined as (2):

$$B_{ij} = \begin{cases} 1, & \max(I_r[j]_i) \geq \epsilon, \\ 0, & \max(I_r[j]_i) < \epsilon. \end{cases} \tag{2}$$

In this case, $B_{ij} = 1$ indicates a strong current response in channel j when channel i is stimulated. This reaction allows for motor identification and active channel determination with Algorithm 4. As a result of this identification process, the system dynamically activates the relevant control layer or the safe mode.

Algorithm 4. Motor identification algorithm

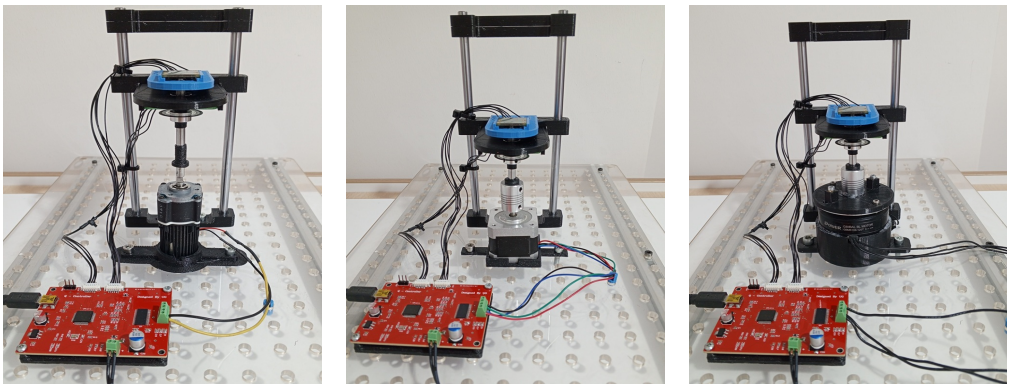
```
I_pk ←Set current limit with DAC
Initialize power stage
for i = 1 to 4 do
   PWM_TIM ← Set all channel to zero
   PWM_i ← Excite channel i
   ( I_r[], V_bus) ← ReadADC_FDMA()
   for j = 1 to 4 do
      if  max(I_r[j]) ≥ ϵ then  B_ij = 1  else B_ij = 0
   end for
end for
if Σ_{j=i} B_ij = 3  then
   Motor type ← BLDC, active channels i : B_ij = 1
else if Σ_{j=i} B_ij = 2 then
   Motor type ← DC, active channels i : B_ij = 1
else if Σ_{j=i} B_ij = 4  then
   Motor type ← stepper, active channels i : B_ij = 1
else if Σ_{j=i} B_ij = 0 then
   Open Mode ← Disable Driver, No connection
end if
if  Motor type ≠ User type then
   Safe Mode ← Disable Driver, Unmatched Motor
end if
```

4. VALIDATION OF PLATFORM

To verify the platform, three test setups were established. In the experimental setup, a 1000 pulses per revolution 3-phase optical disk encoder and an AS5048 magnetic encoder were located on the same shaft. The magnetic encoder provides absolute position measurement with 14-bit resolution. The optical encoder was connected to the quadratic encoder input of the MCU. The magnetic encoder was connected to the MCU via the SPI bus. The angular position of the motor shaft was measured with the magnetic encoder. The angular velocity of the motor shaft was measured with the optical encoder. A Faulhaber planetary geared motor was used as a sample DC motor. The DC motor geared shaft is directly connected to sensor shaft. Figure 3 presents the experimental test setups used to validate the proposed motor algorithm for different motor types. Figure 3(a) shows the DC motor test setup. Figure 3(b) demonstrates the stepper motor test bench. A NEMA 17 type bipolar motor was used for the stepper motor test. The BLDC motor test setup is shown in Figure 3(c). A GMB5108-120T gimbal motor was used as the FOC algorithm validation.

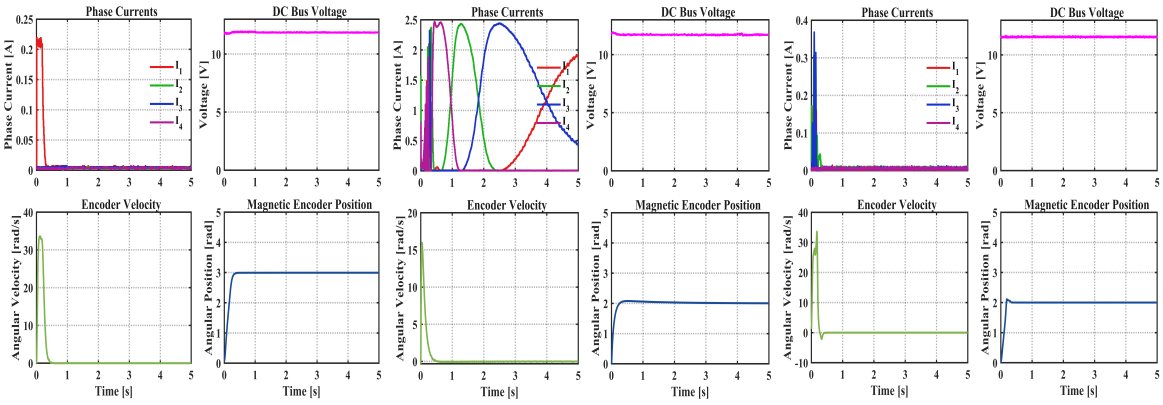
In platform validation experiments, the currents of each half-bridge (on high-side), the DC bus voltage, the optical and absolute encoder responses were measured and graphed. Figure 4 presents the responses of the control algorithms obtained during the experimental validation tests. Figure 4(a) shows the DC motor algorithm response with a reference input of 3 radian. The stepper motor was set to 128 microsteps and a 2 radian reference input was applied. Figure 4(b) shows the stepper motor algorithm response. For the BLDC motor, a reference of 2 radians was applied with relative magnetic zero detection. Figure 4(c) shows the FOC algorithm response. All parameters were adjusted using MATLAB interface and all response data were collected in MATLAB. Three performance metrics were considered to demonstrate the functionality of the algorithms: rise time, settling time, and steady-state error. The control parameters for each motor were manually adjusted experimentally by monitoring motor responses.

Table 1 summarizes the validation results of the proposed power stage for the sample motors. The steady-state error remains below 0.015 radians for all tested motors. The stepper motor exhibits the highest settling time and steady-state error, while the DC motor achieves the best dynamic performance. These results confirm that the proposed system can successfully drive different motor types, with the final performance depending on the selected control parameters and motor characteristics.



(a) (b) (c)

Figure 3. Test setup for; (a) DC, (b) stepper, and (c) BLDC motor



(a) (b) (c)

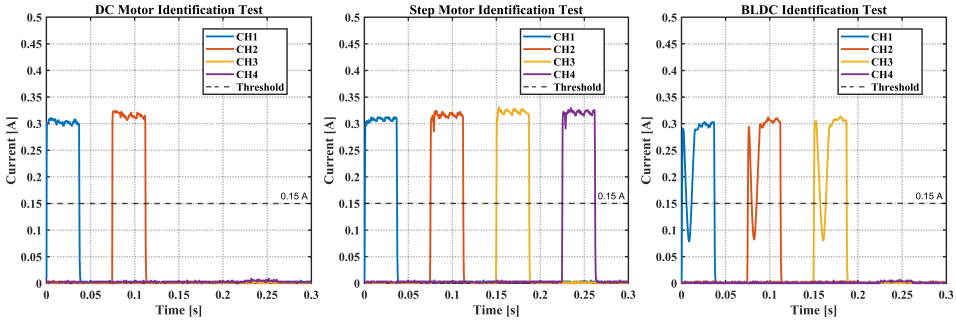
Figure 4. Control responses for; (a) DC, (b) stepper, and (c) BLDC motor

Table 1. The validation of control algorithms

Motor type	Rise time (s)	Settling time (s)	Steady state error (rad)
DC motor	0.32	0.254	0.0094
Stepper motor	0.22	0.853	0.0148
BLDC motor	0.18	0.383	0.0107

Experiments were conducted with the sample motors to demonstrate the robustness of the motor type identification algorithm. In the experiment, each half-bridge was stimulated sequentially for a duration of 37.5 ms. The channel being tested is active in the high side. Other channels are kept active on the low side. The current limit is set as 300 mA $\pm$ 12% and duty-cycle is tuned as 45% at 25 kHz. The threshold value is defined as 150 mA for $\delta = 0.5$. Figure 5 shows the current response for motor type and active channel experiments. The DC motor is connected to the channel one and two. Figure 5(a) shows the channel responses for DC motor. There are two short current pulses on channel one and two. Other channels are on the floor.

The active channel responses are approximately twice the threshold level. The stepper motor was connected on channel one, two, three, and four. Figure 5(b) shows the current response for the stepper motor. Four channels provide current pulses over than twice the threshold level. In the BLDC motor, three channels of driver were connected. Figure 5(c) exhibits the current response for the BLDC motor. There are three pulses over threshold level. Response of the BLDC motor is different from DC and stepper motor. There are sharp current drop at the beginning of the excitation. This situation is caused by the back-EMF that initially occurs in the active low-side connected windings of the star-wound. The identification algorithm searches the maximum response over than the threshold. Thus, the current drop in the BLDC motor does not affect the identification process. Repeated testing was conducted to demonstrate the robustness of the identification algorithm.



(a) (b) (c)

Figure 5. Motor type and active channel identification response for; (a) DC, (b) stepper, and (c) BLDC motor

Table 2 summarizes the identification performance. The identification algorithm was repeated ten times for each motor. The success count denotes the number of correct identifications. The peak-min value represents the average of the lowest responses above the threshold level, while the peak-max value denotes the average of the highest responses above the threshold level. Peak-Std defines the standard deviation of the peak responses. The root-mean-square (RMS) of the responses below the threshold level is denoted as the noise RMS. The signal-to-noise ratio (SNR) is calculated as the logarithmic ratio between the peak-max value and the noise RMS. The identification algorithm determines the motor type and active channels based on the peak-max values. For all tested motors, the success count reached ten out of ten, corresponding to a 100% identification success rate. The maximum Peak-Std value was limited to 0.0035, indicating good repeatability of the proposed method across repeated trials. Furthermore, the minimum SNR was observed in the BLDC motor as 25.078. All measured identification peaks are clearly distinguishable from the noise, which ensures accurate motor type identification in the experiments.

Table 2. Performance of identification algorithm

Motor	Success count	Active channels	Peak-min (A)	Peak-max (A)	Peak-Std	Noise RMS	SNR (dB)
DC motor	10	CH1	0.3065	0.3135	0.0018	0.0060	34.278
		CH2	0.3223	0.3293	0.0020	0.0051	36.121
Stepper motor	10	CH1	0.3048	0.3153	0.0031	0.0057	34.785
		CH2	0.3135	0.3258	0.0035	0.0058	34.843
		CH3	0.3223	0.3328	0.0028	0.0054	35.655
		CH4	0.3206	0.3328	0.0035	0.0051	36.065
BLDC motor	10	CH1	0.3030	0.3048	0.0008	0.0169	25.078
		CH2	0.3083	0.3118	0.0012	0.0165	25.497
		CH3	0.3048	0.3135	0.0031	0.0168	25.324

5. RESULTS AND DISCUSSION

Considering Table 1, while the BLDC motor has the highest rise time, the DC motor exhibits the lowest rise time. The main reason for this situation is that the measurement is taken from the gear output of the DC motor. The DC motor exhibits the best settling time, while the stepper motor has the worst. The microstep resolution of a stepper motor is set at 128. Therefore, increasing the number of control step causes

the stepper motor to settle down more slowly. Additionally, it should be considered that the motor performance, besides the control parameters, will depend on load, the mechanical structure, motor specifications (such as step resolution of stepper motors and pole-pair number of BLDCs), and feedback sensors. Therefore, Table 1 should be evaluated specifically in terms of validation. With the proposed platform, the user can customize the control performance depending on the motor characteristics, associated mechanical system, and load.

Motor type identification and active channel detection offer a feature that distinguishes the proposed system from software-based heterogeneous multi-mode motor control methods [14], [15]. This feature ensures the correct identification of the motor type and the active half-bridge channels. According to Table 2, it is seen that the stepper and DC motor have SNRs above 34 dB. On the other hand, it is observed that the SNR value of the BLDC motor is around 25 dB and its noise RMS is relatively high. The main reason for this is that the current drop in the motor response falls below the threshold value. Therefore, the RMS noise increases. However, this does not affect the identification algorithm. Because the algorithm considers the peak-max value.

One situation to consider is connecting two DC motors simultaneously. In this case, the identification process recognizes both DC motors as stepper motor. However, the user must also enter the motor type information to start the identification process. The algorithm compares the motor type with the user input. In case of a mismatch, the system enters safety mode and shuts down the power stage.

Connection issues, such as connection fault or motor winding problems, can be determined using the identification algorithm. In addition to algorithmic check, the channel responses shown in Figure 5 are transmitted to the user via the user interface. Therefore, the user can also identify faulty connections.

The proposed hardware can provide up to 300 watts of output power. The maximum supply voltage is 60 V and the maximum current per channel is 5 A. This limits the motors that the system can drive. On the other hand, many motors commonly used in robotics and precision electro-mechanical systems can be covered [27]. Additionally, a heat-sink is required if the current per channel exceeds 1.5 A.

The proposed platform uses an MCU with a 12-bit resolution ADC and DAC. Current measurement utilizes a high-side current sensor located within the integrated power stage. The current measurement accuracy for the current sensor is defined as $\pm 3\%$ for a maximum current of 5 A [20]. The current measurement resolution is calculated as 1.23 mA/LSB. The current limiter accuracy of the DRV8962DDW is typically within $\pm 8\%$ [20], it is assumed $\pm 12\%$ for worst-case conditions.

The MCU hardware and algorithms are open to implemented within higher power motor. However, hardware of the power stage needs to be redesign. For higher powers, it is necessary to construct 4 half-bridges that can operate under high power with thermal management. In addition to a suitable gate driver, analog signal conditioning circuits must be added for current limiting and current sensing. On the other hand, a pin-compatible 600 watt version of the integrated power stage (DRV8962DDV) is available and compatible with all algorithms in this study. However, due to the different package structure, it requires revision of the printed circuit board.

6. CONCLUSION

This study presents a unified, multi-mode motor control platform capable of controlling DC, BLDC, and stepper motors on the same hardware through motor type identification and optimized control algorithms. Studies on motor control in the literature are largely focused on a single type of motor. Software libraries or FPGA solutions are being developed in the context of multi-mode motor control systems. In the proposed system, software and hardware are combined for a multi-mode motor driving structure with four half-bridges. Unlike software-based solutions, algorithms were applied to detect the motor type and the motor connections. Additionally, a method is presented that is easier to configure than FPGA-based solutions. For experimental validation, position control of DC, BLDC, and stepper motors was performed. The validity of the motor type and active channel identification algorithms was tested through repeated experiments. No identification errors were observed during the experiments. The results demonstrate the potential of the proposed platform and algorithms as a reliable and flexible solution for medium-power multi-mode motor control.

FUNDING INFORMATION

Author state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ufuk Güner	✓	✓	✓	✓		✓			✓	✓				

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Author state no conflict of interest.

DATA AVAILABILITY

All code and hardware design files are openly available in https://github.com/ufukguner/FEMC_Mul ModMotorDrv with GPL3 licenses. They can be used and modified with referencing this study.

REFERENCES

[1] A. Veltman, D. W. J. Pulle, and R. W. De Doncker, *Fundamentals of Electrical Drives*, Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-29409-4.

[2] S.-H. Kim, *Electric Motor Control: DC, AC, and BLDC Motors*, Elsevier Science, 2017, ISBN: 978-0-12-812138-2.

[3] D. Mohanraj *et al.*, "A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications," *IEEE Access*, vol. 10, pp. 54833–54869, 2022, doi: 10.1109/ACCESS.2022.3175011.

[4] A. Morar, "Compact and intelligent Full/Half five-phase stepping Motor driver," *Procedia Technology*, vol. 12, pp. 730–739, Jan. 2014, doi: 10.1016/j.protcy.2013.12.556.

[5] M. Akrami, E. Jamshidpour, B. Nahid-Mobarakeh, S. Pierfederici, and V. Frick, "Sensorless Control Methods for BLDC Motor Drives: A Review," *IEEE Transactions on Transportation Electrification*, vol. 11, no. 1, pp. 135–152, Feb. 2025, doi: 10.1109/TTE.2024.3387371.

[6] D. Joshi, D. Deb, and A. K. Giri, "Metaheuristic Adaptive Input Output Feedback Linearization Control for BLDC Motor Drive," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 6120–6130, May 2025, doi: 10.1109/TCE.2025.3553388.

[7] S. Lixian and W. Rahiman, "A Compound Control for Hybrid stepper Motor Based on PI and Sliding Mode Control," *IEEE Access*, vol. 12, pp. 163536–163550, 2024, doi: 10.1109/ACCESS.2024.3490793.

[8] D. Montoya-Acevedo, W. Gil-González, O. D. Montoya, C. Restrepo, and C. González-Castaño, "Adaptive Speed Control for a DC Motor Using DC/DC Converters: An Inverse Optimal Control Approach," *IEEE Access*, vol. 12, pp. 154503–154513, 2024, doi: 10.1109/ACCESS.2024.3482982.

[9] J. A. Niembro-Ceceña, R. A. Gómez-Loenzo, and J. Rodríguez-Reséndiz, "SoftCtrlDC-M: Embedded control software for brushed direct current motors," *SoftwareX*, vol. 25, 2024, doi: 10.1016/j.softx.2024.101643.

[10] X. Zeng, C. Liu, X. Sheng, Z. Xiong, and X. Zhu, "Development and implementation of modular FPGA for a multi-motor drive and control integrated system," in *International Conference on Intelligent Robotics and Applications*, 2015, pp. 221–231, doi: 10.1007/978-3-319-22879-2_21.

[11] Z. Huang, S. Qiu, B. Wang, and Q. Liu, "High Precision and Fast Synchronization Fuzzy Position FPGA-Based Controller for Smart Multi-Motor System," *IEEE Transactions on Industry Applications*, vol. 61, no. 3, pp. 3886–3895, May–Jun. 2025, doi: 10.1109/TIA.2025.3536417.

[12] T. Lan, B. Liu, J. Wang, and Y. Xue, "Event-Triggered Control Strategy for Multi-Motor Systems Based on Iterative Learning," in *2025 2nd International Conference on Electrical Technology and Automation Engineering (ETAE)*, Guangzhou, China, 2025, pp. 324–329, doi: 10.1109/ETAE65337.2025.11089630.

[13] R. de Castro, R. E. Araujo, and H. Oliveira, "Control in multi-motor electric vehicle with an FPGA platform," in *2009 IEEE International Symposium on Industrial Embedded Systems*, Lausanne, Switzerland, 2009, pp. 219-227, doi: 10.1109/SIES.2009.5196218.

[14] A. Skuric, H. S. Bank, R. Unger, O. Williams, and D. González-Reyes, "A Field Oriented Control (FOC) Library for Controlling Brushless Direct Current (BLDC) and Stepper Motors," *Journal of Open Source Software*, vol. 7, no. 74, p. 4232, 2022, doi: 10.21105/joss.04232.

[15] "STM32 Motor Control Software Development Kit (SDK)," STMicroelectronics, [Online]. Available: https://www.st.com/en/embedded-software/x-cube-mcsdk.html#overview, (Accessed: Dec. 2025).

[16] Xilinx Inc., "Avnet-AES-S6MC1-LX75T-G User Guides: Getting Started Guide," Xilinx Documentation, 2014, [Online]. Available: https://www.avnet.com/fsp/opasdata/d120001/medias/docus/7/Avnet-AES-S6MC1-LX75T-G-User-Guides-Getting-Started-Guide.pdf, (Accessed: Dec. 2025).

[17] Microchip Technology Inc., "Motor Control Design Using SmartFusion2 and IGLOO2 Devices," AC445 Application Note, Jul. 2015, [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/SoC/microsemi_smartfusion2_igloo2_motor_control_design_application_note_ac445_v3.pdf, (Accessed: Dec. 2025).
[18] K. Kolano, "Determining the Position of the Brushless DC Motor Rotor," *Energies*, vol. 13, no. 7, pp. 1-9, Apr. 2020, doi: 10.3390/en13071607.
[19] E. Gundabattini, R. Kuppan, D. G. Solomon, A. Kalam, D. P. Kothari, and R. A. Bakar, "A review on methods of finding losses and cooling methods to increase efficiency of electric machines," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 497–505, 2021, doi: 10.1016/j.asej.2020.08.014.
[20] Texas Instruments Incorporated, "DRV8962 Four-channel Half-Bridge Driver with Current Sense Outputs," Technical Report, Oct. 2023, [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8962.pdf, (Accessed: Dec. 2025).
[21] Ş. Yıldırım, M. S. Bingöl, and S. Savaş, "Tuning PID controller parameters of the DC motor with PSO algorithm," *International Review of Applied Sciences and Engineering*, vol. 15, no. 3, pp. 281–286, 2024, doi: 10.1556/1848.2023.00698.
[22] K. M. Le, H. V. Hoang, and J. W. Jeon, "An Advanced Closed-Loop Control to Improve the Performance of Hybrid stepper Motors," *IEEE Transactions on Power Electronics*, vol. 32, no. 9, pp. 7244–7255, Sept. 2017, doi: 10.1109/TPEL.2016.2623341.
[23] S. Derammelaere *et al.*, "The Efficiency of Hybrid stepping Motors: Analyzing the Impact of Control Algorithms," *IEEE Industry Applications Magazine*, vol. 20, no. 4, pp. 50–60, Jul.–Aug. 2014, doi: 10.1109/MIAS.2013.2288403.
[24] J.-H. Ma, Y.-q. Li, G. Wang, and Y.-Y. Liu, "Design of adaptive control algorithm for automotive dashboard stepper motor pointer," in *Proceedings of the 2013 International Conference on Advanced Mechatronic Systems*, Luoyang, China, 2013, pp. 219–223, doi: 10.1109/ICAMechS.2013.6681781.
[25] H.-C. Liu *et al.*, "Micro controller unit based motor control system using field oriented control algorithm," in *2013 9th International Conference on Information, Communications & Signal Processing*, Tainan, Taiwan, 2013, pp. 1–5, doi: 10.1109/ICICS.2013.6782806.
[26] M. Hu, H. Ahn, J. Park, and K. You, "Novel rapid control prototyping for permanent magnet synchronous motor via model-based design and STM32 chip," *The International Journal of Advanced Manufacturing Technology*, vol. 135, pp. 1187–1204, 2024, doi: 10.1007/s00170-024-14579-4.
[27] Silicon Labs, "Software considerations for advanced motor control," White Paper, 2005, [Online]. Available: https://www.silabs.com/documents/public/white-papers/software-considerations-advanced-motor-control.pdf, (Accessed: Dec. 2025).

BIOGRAPHY OF AUTHOR

Ufuk Güner received the B.S. degree in 2005. He worked as an R&D engineer for more than eight years. He received the M.S. degree in Electrical Electronics Engineering from Ankara Yildirim Beyazit University in 2016 and Ph.D. degree in Control and Automation Engineering from Yildiz Technical University in 2023. Since 2023, he has been an assistant professor with the Department of Electrical and Electronics Engineering at Erzurum Technical University. His research interests include embedded systems, sensors, robotics, and control systems. He can be contacted at email: ufuk.guner@erzurum.edu.tr.