

## REAL-TIME SIMULATION AND EXPERIMENTAL IMPLEMENTATION OF LUENBERGER OBSERVER-BASED SPEED SENSOR FAULT DETECTION OF BLDC MOTORS

Yousef AL-MUTAYEB\*<sup>ORCID</sup>, Moayed ALMOBAIED\*\*<sup>ORCID</sup>, Mohamed OUDA\*\*\*<sup>ORCID</sup>

\*Department of Engineering Sciences, University College of Science and Technology, Khan Younis Street, Kasba Tadla, Palestine

\*\*Department of Electrical Engineering, Islamic University of Gaza, Jamal Abdel Street, Al-Remal, Gaza, Gaza strip, Palestine

\*\*\*Department of Electrical Engineering, University of Doha for Science and Technology, 24449-Arab League Street, Doha, Qatar

[y.mtieb@ucst.edu.ps](mailto:y.mtieb@ucst.edu.ps), [malmobaied@iugaza.edu.ps](mailto:malmobaied@iugaza.edu.ps), [Mohamed.ouda@udst.edu.qa](mailto:Mohamed.ouda@udst.edu.qa)

*received 21 January 2023, revised 25 July 2023, accepted 11 August 2023*

**Abstract:** Modern control systems' dependability, safety and efficiency have all been improved by studying fault-tolerant control systems (FTCS). FTCS techniques can typically be active or passive controls. The fault detection and diagnosis (FDD) method is used in this study's active control branch to identify probable faults that could develop in the speed Hall sensors of brushless DC motors (BLDC). FDD methodologies can be categorised into two types, depending on the available data and the process involved: model-based methods and data-based methods. The proposed approach in this study explores the implementation of the Luenberger observer methodology as part of the model-based approach. The chosen methodology was practically implemented and subjected to experimental evaluation. The proposed observer relies on the residual signal, which displays the difference between the plant's observed and estimated speed signals and serves as a failure alert for the entire system. Given the increasing demand for BLDC motors in various industrial control applications, including medical fields, automation and robotics, this particular motor was selected as a benchmark to thoroughly evaluate and validate the proposed method. The primary contribution of this paper lies in the real-time application of model-based sensor fault detection methods to BLDC motors. The efficiency of the suggested method is showcased through extensive MATLAB simulations, where the obtained results confirm the successful detection of faults with a high level of responsiveness. As a result, the project was successfully implemented in real-time, and the experimental results exhibited a close correlation with the simulated outcomes. This consistency between simulation and practical implementation validates the accuracy and reliability of the proposed methodology for detecting faults in the BLDC motor speed sensor. The results underscore the heightened reliability and safety attained by promptly and accurately detecting sensor faults during the operation of the motor.

**Key words:** brushless DC motor, Hall sensors, faults detection, Luenberger observer

### 1. INTRODUCTION

Fault-tolerant control systems (FTCS) are a cohesive collection of advanced control functions designed to ensure exceptional system safety by operating seamlessly as a unified entity. Two types of FTCS can be distinguished: passive FTCS and active FTCS [1]. This paper focusses on the study of fault detection and diagnosis (FDD), which falls under the category of active fault-tolerant control methods. During the last two decades, fault tolerance control (FTC) has garnered significant attention in various critical industrial applications. The need to enhance the reliability and safety of electrical systems is on the rise in numerous industrial applications. Industrial applications encompass a variety of uses, such as electrical motors, industrial production lines and aircraft systems. The BLDC motor has gained significant popularity in recent times and has emerged as one of the most extensively utilised electrical motors in various industrial applications, including water pumps, hybrid vehicles and industrial robots [2]. In comparison to other motor types, BLDC motors have higher efficiency, lower noise, a higher speed range and greater durability [3]. The FDD is a technique used by researchers interested in fault tolerance control to increase electric motors' dependability and safety [5]. An abrupt system fault that occurs when a device (such as a motor) is in use could result in expensive downtime,

environmental harm or even danger to people [1]. Some faults can be detected early because they decay more slowly than others, enhancing safety, preventing system failures, minimising product damage and extending the overall equipment life [6]. Electrical and mechanical faults are the two categories of electrical motor faults. Electrical faults can be attributed to several causes, including rotor damage, stator winding short circuit, broken end rings and inverter faults. Mechanical faults can stem from various sources, such as shaft misalignment, gearbox problems and bearing damage [7, 8]. For motors to work properly, sensors are crucial. When investigating a motor fault and diagnostics, sensor faults pose a major challenge. Electric motors can be monitored, and faults can mainly be found using the model-based and data-based methods. The model-based approach compares the system's mathematical model with the actual plant's behaviour. The method utilises the residual signal, obtained by computing the difference between the actual process (measured signal) and the model's estimation (estimated signal). There are several methods for residual generation, including observer-based generation, parity equations and parameter estimation [9]. For fault detection, observer-based and parameter estimation techniques are most frequently utilised [10]. A data-driven method, on the other hand, is distinguished by the absence of the need for a dynamic model of the real process. The fault detection technique is entirely based on the utilisation of measured and analysed physical parameters

to extract fault signatures, enabling the detection and diagnosis processes [11]. The current issue faced by the industry in the context of BLDC motors is the reliability and accuracy of speed sensor fault detection. Traditional methods of detecting faults in speed sensors may not be suitable for real-time applications and may lead to inaccurate or delayed fault detection, potentially causing safety concerns and operational inefficiencies. The problem encountered in this research is to develop a robust and real-time speed sensor fault detection system for BLDC motors that can accurately and promptly detect any faults in the speed sensor. This problem poses a significant challenge due to the complex dynamics of BLDC motors and the need to ensure precise fault detection without compromising the motor's performance. The research aims to address this issue by proposing and implementing a Luenberger observer-based fault detection methodology. This method leverages the benefits of observer-based techniques to achieve real-time and reliable detection of speed sensor faults in BLDC motors, thus enhancing the overall system's safety and efficiency. Numerous significant contributions have been made in the area of BLDC motor fault detection. Skora and Kowalski [12] discuss the impact of a Hall-effect sensor defect for rotor position determination on motor operation. In Sova et al. [4], an algorithm for enabling the operation of a BLDC motor drive with one faulty Hall position sensor has been introduced. In Tashakori and Ektesabi [13], a fault-tolerant control system for BLDC motor has been presented, focussing on the Hall-effect position sensor failure. In Zandi and Poshtan [14], a method for detecting bearing faults in the BLDC motor was introduced, which involved the utilisation of Hall-effect sensors and wavelet transform. A new methodology for active fault-tolerant control has been developed in Allous and Zanzouri [15], addressing simultaneous faults in both the actuator and sensor. The innovative approach incorporates the integration of a Luenberger observer and a bond graph linear fractional transformation model as key elements of its working strategy. In this context, Saoudi et al. [16] introduce a novel fault detection approach employing a Luenberger observer, which is based on a bond graph model. This method is particularly applicable for constructing a full-order observer in the context of linear systems. In Al-Mutayeb and Almobaied [17], a Luenberger observer and MATLAB software method were utilised to simulate a speed sensor fault on a BLDC motor. This paper presents a significant contribution by applying model-based sensor fault detection methods to the MOONS' BLDC motor in real-time scenarios. Specifically, the study introduces a Luenberger observer to effectively detect speed sensor faults in the motor. The schematic diagram of the proposed method is illustrated in Fig. 1.

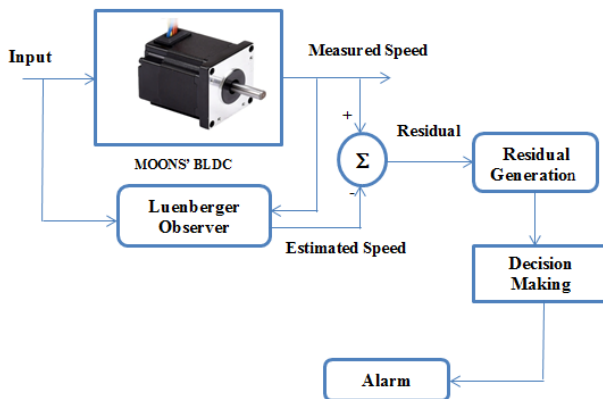


Fig. 1. The BLDC motor's Luenberger observer

The remainder of this paper is structured as follows: In Section 2, the mathematical model of MOONS' BLDC motor is presented in detail. Section 3 presents a comprehensive description of the Luenberger observer method. Section 4 presents hardware and software design and implementation. Section 5 provides simulation and experimental results of the suggested Luenberger observer approach. Finally, Section 6 presents the conclusion of the study and outlines potential areas for future research.

## 2. MATHEMATICAL MODEL

### 2.1 A BLDC motor overview

BLDC motor connections usually take the form of a star (Y) connection. However, the BLDC motor of MOONS' Company uses a delta ( $\Delta$ ) winding connection. The equivalent circuit of the BLDC motor demonstrates the delta winding connection, as depicted in Fig. 2.

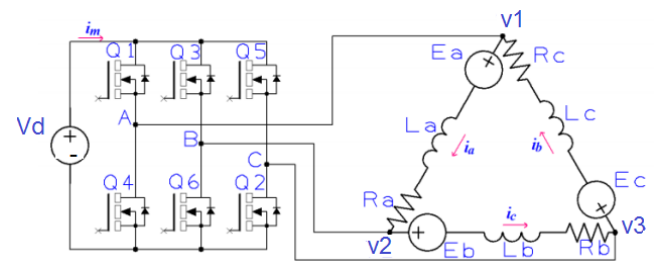


Fig. 2. Delta winding connection in BLDC motor circuit [18]

Fig. 3 illustrates the utilisation of a 30 W BLDC motor manufactured by MOONS' Company for industrial applications. Table 1 provides the parameter specifications extracted from the BLDC motor datasheet [19].



Fig. 3. The 42BL30L2-5 is a BLDC motor manufactured by MOONS' Company

Tab. 1. The Specifications of the 42BL30L2-5 MOONS' BLDC motor

Motor Details	Value
Drive voltage nominal $U_N$	24 V
Poles number	6
The number of phases	3
Rotational speed nominal $n_N$	4,000 RPM
Winding current nominal $I_N$	1.67 A
Resistance to ph/ph $R_a$	1.34 Ohm

Ph/ph inductance $L_a$	0.00115 H
Inertia of the rotor $J$	0.0388e-4 kg/m <sup>2</sup>
The coefficient of friction $B$	1.718e-4 Nm/rad/sec
Back EMF constant $K_b$	0.0281 Vrms/rad/sec
Torque constant $K_t$	0.043 Nm/Arms

## 2.2 Transfer Function

The transfer function represents “the ratio of angular velocity ( $\omega_m$ ) to the source voltage ( $V_s$ )”. It can be expressed as [20]:

$$T.F = \frac{\omega_m}{V_s} = \frac{1/K_b}{\tau_m \tau_e s^2 + \tau_m s + 1} \quad (1)$$

In modelling BLDC motors, the mechanical and electrical time constants serve as essential parameters to consider. The mechanical time constant ( $\tau_m$ ) and the electrical time constant ( $\tau_e$ ) of a BLDC motor are defined as follows [17]:

$$\tau_m = \frac{3 \cdot R_a \cdot J}{K_b \cdot K_t} \quad (2)$$

$$\tau_e = \frac{L_a}{3 \cdot R_a} \quad (3)$$

The mechanical time constant ( $\tau_m$ ), obtained from Eq. (2), is:

$$\tau_m = 0.0129 \text{ sec}$$

The electrical time constant ( $\tau_e$ ), obtained from Eq. (3), is:

$$\tau_e = 2.8606e^{-4} \text{ sec}$$

The resulting transfer function is as follows:

$$T.F = \frac{35.587}{3.690e^{-6} \cdot s^2 + 0.0129 \cdot s + 1}$$

## 2.3 Modelling the MOONS' BLDC Motor using State-Space Representation

The observer for fault detection in BLDC motors was created using state-space modelling. The general form of a state-space model is represented as follows [21]:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4)$$

$$y(t) = Cx(t) \quad (5)$$

Where  $u(t)$  is the input,  $x(t)$  represents the state,  $A$  is the state matrix,  $B$  is the input matrix,  $C$  is the output matrix and  $y(t)$  is the output.

To satisfy the requirements, the state vector  $x(t)$  has been carefully designed [22]:

$$x_1(t) = \omega(t) = \frac{d\theta(t)}{dt} \quad (6)$$

$$x_2(t) = i(t) \quad (7)$$

$$\dot{x}(t) = \begin{pmatrix} \dot{\omega} \\ \dot{i} \end{pmatrix} = \begin{pmatrix} -\frac{B}{J} & \frac{K_t}{J} \\ -\frac{K_b}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} x + \begin{pmatrix} 0 \\ \frac{1}{L_a} \end{pmatrix} u \quad (8)$$

$$y(t) = \omega(t) = (1 \ 0)x \quad (9)$$

where  $\omega(t)$  denotes angular velocity and  $i(t)$  denotes motor current.

Consequently, the matrices  $A$ ,  $B$  and  $C$  are defined as follows:

$$A = \begin{pmatrix} -\frac{B}{J} & \frac{K_t}{J} \\ -\frac{K_b}{L_a} & -\frac{R_a}{L_a} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{L_a} \end{pmatrix}, C = (1 \ 0) \quad (10)$$

Upon substituting the parameter values into Eq. (10), the matrices  $A$ ,  $B$  and  $C$  can be obtained as the following:

$$A = \begin{pmatrix} -44.28 & 1.108e^4 \\ -24.43 & -1165 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 869.6 \end{pmatrix}, C = (1 \ 0)$$

## 3. METHOD OF LUENBERGER OBSERVER

The basic idea underlying the Luenberger observer approach is to approximate the real system states based on the available measured data [23]. The residual signal is generated to compare the estimated states from the observer with the observed states of the actual system. Following that, the residue is computed by subtracting the observer's output signals from the measured signals [24]. Fig. 4 depicts a functional schematic of the real system with the Luenberger observer model, which is utilised for defect detection approaches. The state-space model serves as the foundation for the Luenberger observer design for defect detection.

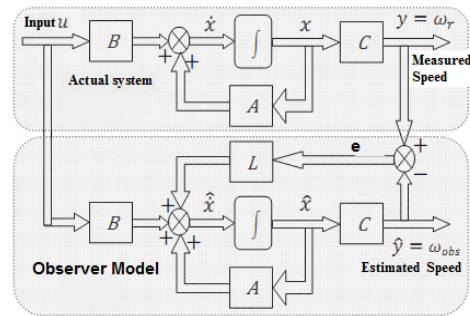


Fig. 4. Diagram of the functionality of the real system and observer models

The Luenberger observer can be constructed using the following equations [23]:

$$\hat{\dot{x}}(t) = A\hat{x}(t) + Bu(t) + Le(t) \quad (11)$$

$$\hat{y}(t) = C\hat{x}(t) \quad (12)$$

$$e(t) = y(t) - \hat{y}(t) \quad (13)$$

where  $\hat{x}(t)$  represents the estimated system state  $t$ ,  $\hat{y}(t)$  represents the estimated output,  $L$  is the observer gain matrix and  $e(t)$  is the output error that is the difference between the actual measured output  $y(t)$  and its estimated value  $\hat{y}(t)$ .

Substituting Eq. (13) in Eq. (11):

$$\hat{\dot{x}}(t) = [A - LC]\hat{x}(t) + Bu(t) + Ly(t) \quad (14)$$

The estimated error can be expressed as follows:

$$e(t) = x(t) - \hat{x}(t) \quad (15)$$

By computing the first derivative of  $e(t)$ , we can derive the rate of change of the output error, expressed as:

$$\dot{e}(t) = \frac{d}{dt}(x(t) - \hat{x}(t)) = Ax(t) + Bu(t) - A\hat{x}(t) - Bu(t) - LC(x(t) - \hat{x}(t)) \quad (16a)$$

The error's dynamics can be expressed as follows:

$$\dot{e}(t) = (A - LC)(x(t) - \hat{x}(t)) \quad (16b)$$

To achieve faster observer dynamics compared to the real system dynamics, the eigenvalues of the matrix (A-LC) are strategically placed to define the observer error dynamics [21].

The residual signal, denoted as r(t), is computed based on the measured and estimated speeds. The calculation of the residual r(t) is performed in the following manner:

$$r(t) = y(t) - \hat{y}(t) \quad (17)$$

Examination of the residual value enables the automatic detection of faults. Thus, when the residual signal approaches zero or becomes negligible, it indicates that the system is operating without any faults. Conversely, if the residual signal deviates significantly from zero, it signifies the occurrence of a specific fault in the system. The flowchart depicted in Fig. 5 outlines the implementation of the Luenberger observer for fault detection, as proposed in this study [25].

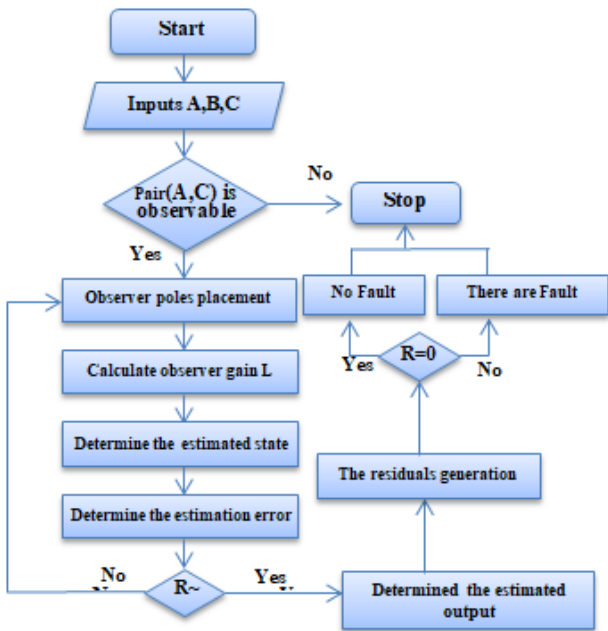


Fig. 5. Flowchart for fault detection using the Luenberger observer

#### 4. HARDWARE AND SOFTWARE DESIGN AND IMPLEMENTATION

The experimental setup in this paper comprises a diverse array of software and hardware components, as elaborated in detail throughout the study. The design of the drive circuits for the BLDC motor was carried out utilising the software Proteus 8.6. The Luenberger observer was also designed using Simulink, which is a special toolbox of the MATLAB numerical calculation software. The interface can be programmed with Simulink. The interfacing is done via the Arduino Mega 2560 board. With the Simulink support package for Arduino hardware, it becomes possible to develop and execute Simulink models directly on the Arduino Mega 2560 board, enabling seamless integration between Simulink and the physical hardware [26].

#### 4.1 BLDC Motor Design and Implementation for BLDC Motor Using Simulink

Since BLDC motors only work when the input voltage phase changes, phase signal generation is a part of the Simulink design for the BLDC motor. A signal with three different phase delays is required for a three-phase BLDC motor. A PWM generation block is used to generate these three-phase signals. Simulink generates PWM signals that are programmed for Arduino Mega 2560. The BLDC motor drive receives the PWM output from the programmed Arduino Mega 2560. Fig. 6 displays a PWM generation block that generates PWM signals and transmits these signals to digital pins (32, 34, 36, 38, 40, 42) in Arduino Mega 2560. Fig. 6 illustrates the internal Simulink design for PWM generation.

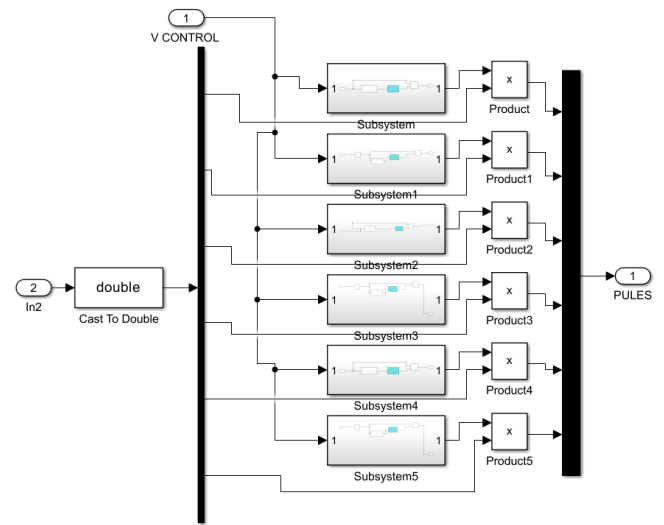


Fig. 6. Internal Simulink design of PWM

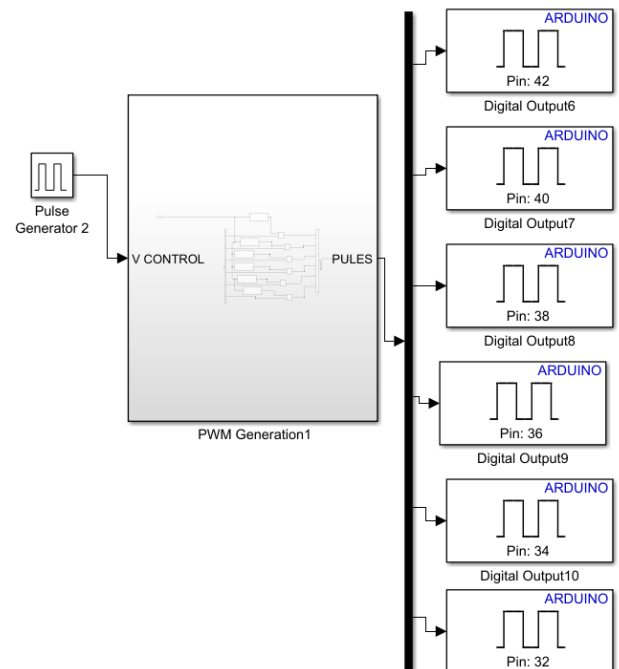


Fig. 7. PWM generation

All generations of PWM signals are similar, but the phase at the output of each pin varies (32, 34, 36, 38, 40, 42) for the PWM

signals generated as shown in Fig. 7. Inside the PWM block, a dead time block was integrated in. Its function was to allow the inverter arm switches to not operate together, ensuring that a key can open before another closes by changing the block's parameters [27]. The full parameterisation of this block is presented in Fig. 8. The dead time between the pulses given to two complementary switches is shown in Fig. 9.

In order to achieve precise control over the position and speed of the BLDC motor, it is essential to employ either Hall sensors or an EMF input signal to accurately determine the rotor's position and speed. The present research specifically utilises Hall sensors to measure and calculate the rotor speed. Fig. 10 depicts the software function designed to receive the input of Hall-effect sensors from the Arduino Mega pins (3, 4, 5).

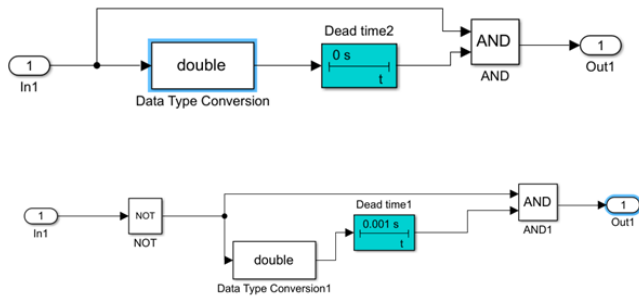


Fig. 8. Internal Simulink design of dead time

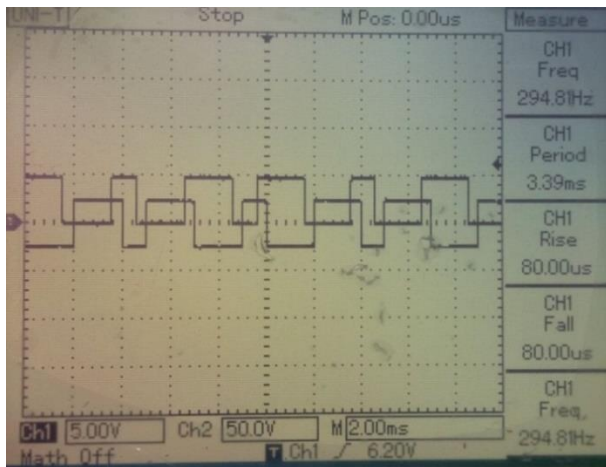


Fig. 9. Complementary PWM pulses with dead time

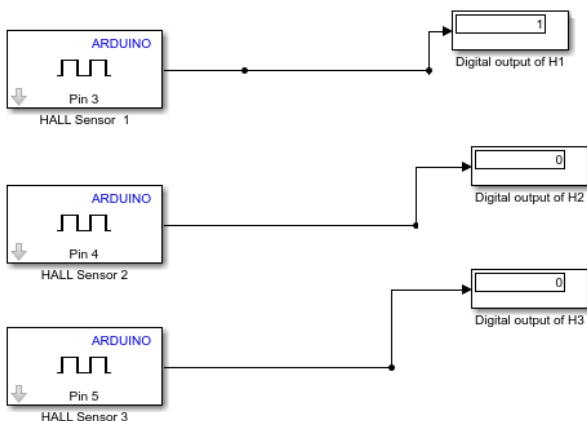


Fig. 10. Design of Hall-effect sensors in display

Fig. 11 displays the three signals obtained from the BLDC motor through the Hall-effect sensors, utilising the Arduino Mega as the interface.

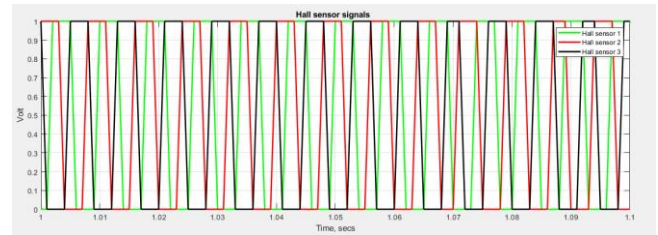


Fig. 11. Hall sensor signals

To achieve speed and position control of the BLDC motor, it is necessary for the motor to rotate in accordance with the inputs received from the Hall-effect sensor. Indirectly, the phase must change in comparison to the Hall sensor inputs [28]. It is known that the corresponding phase signals are solely activated for a specific pair of MOSFETs. Fig. 12 shows the pairs of MOSFET switches inverter.

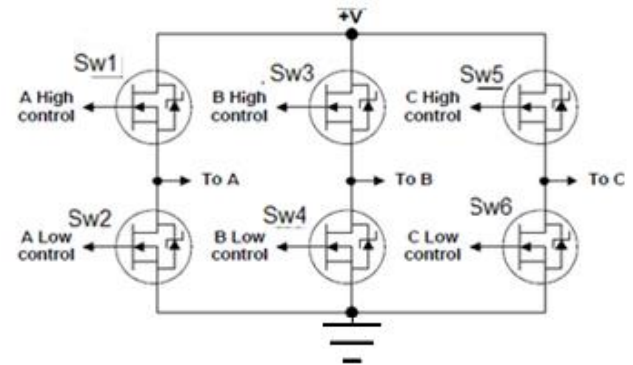


Fig. 12. MOSFET switches inverter

Therefore, we need to match the input of the Hall sensor with the states of a pair of MOSFETs in order to obtain the corresponding phase signals. Typically, the Hall-effect sensor input circuit consists of 101, 100, 110, 010, 011 and 001 [16]. The matching of the input and switching conditions of the Hall-effect sensor are intended to control the speed and position of the BLDC motor. This matching is achieved using two internal logic blocks; the first block is a decoder block, which receives the input signals of the Hall sensor and performs calculations. The decoder block is shown in Fig. 13. Fig. 14 depicts the internal Simulink design for the decoder block. The calculated outputs from the decoder block go to the comparison logic circuit for selecting the PWM signal in the second block, which is the gate block as shown in Fig. 15; and the internal Simulink design for gates block is shown in Fig. 16.

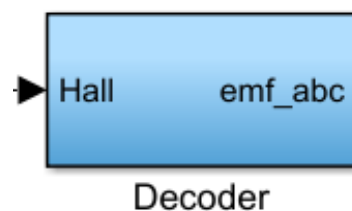


Fig. 13. Decoder block design

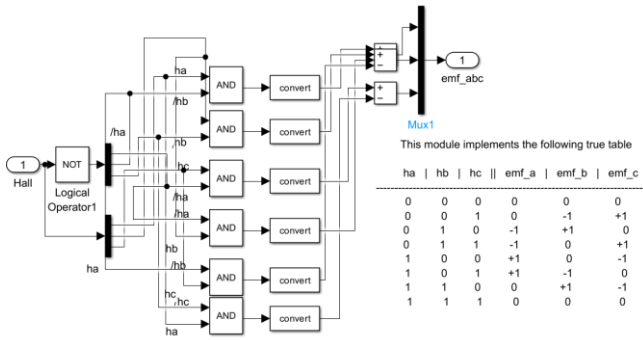


Fig. 14. Internal Simulink design for decoder block

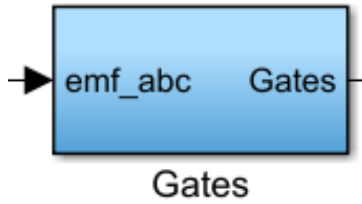


Fig. 15. Gates block design

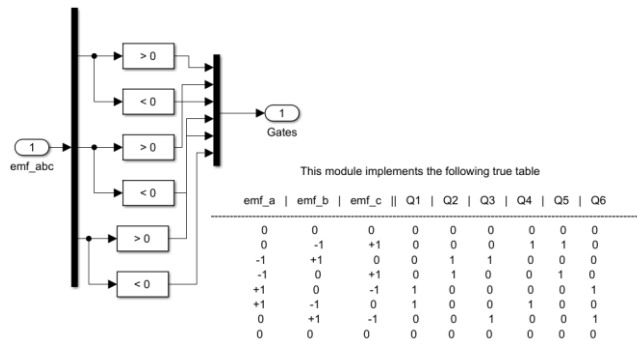


Fig. 16. The total internal Simulink design for gates block

The BLDC motor speed in RPM is calculated using the Hall sensor inputs. The following equations were used for Simulink block design (Hall to RPM) to read the speed of BLDC motor. The rotor's angular velocity can be expressed as follows:

$$\omega_{Hall} = \frac{d\theta_m}{dt} \quad (17)$$

where ( $\omega_{Hall}$ ) rotor angular velocity is calculated by Hall sensors.

The electrical rotor angle and the mechanical rotor angle are interrelated in the following manner [29]:

$$\omega_{Hall} = \frac{2}{p} \cdot \frac{d\theta_e}{dt} \quad (18)$$

where ( $\theta_e$ ) represents the electrical rotor angle [ $^\circ$  el], ( $\theta_m$ ) the mechanical rotor angle [ $^\circ$  el] and ( $p$ ) the number of poles.

The Hall sensor will change the state of each electric  $60^\circ$  and by measuring the time between each state change, we can obtain the angular velocity. Converting the angular velocity to RPM gives the final equation [30]:

$$\omega_{Hall-RPM} = \frac{60}{360} \cdot \frac{2}{p} \cdot \frac{60}{\Delta T_s} \quad (19)$$

where  $\omega_{Hall-RPM}$  represents the rotor speed calculated by Hall sensors in RPM and  $\Delta T_s$  the difference between the last two known time samples generated by Hall state switching.

Fig. 17 depicts the Simulink design used to determine the speed of a BLDC motor.

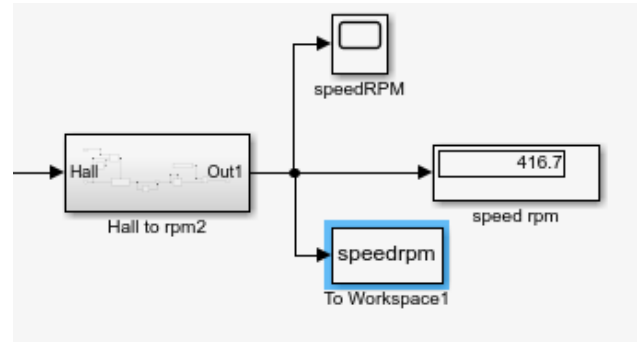


Fig. 17. Outer Simulink design to find speed in RPM

The Simulink internal design for determining the speed in RPM is illustrated in Fig. 18.

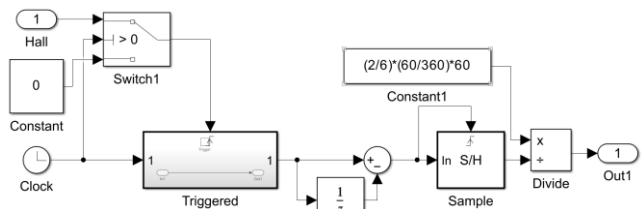


Fig. 18. Internal Simulink design to find speed in RPM.

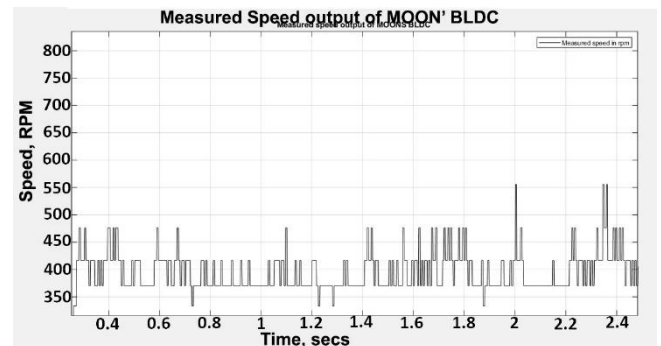


Fig. 19. Measured speed output of MOONS' BLDC motor in RPM

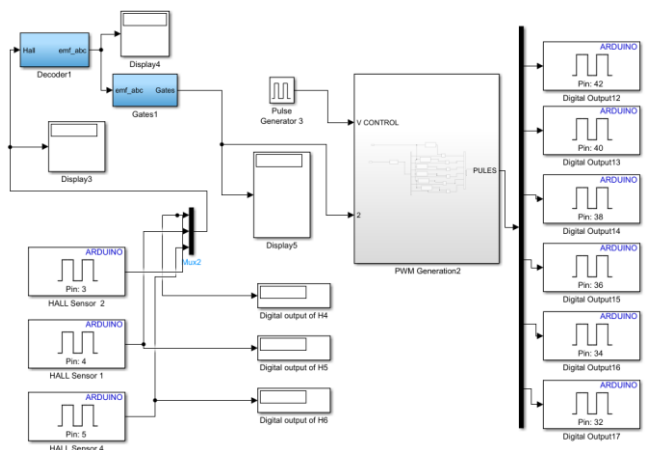


Fig. 20. Comprehensive Simulink design for BLDC motor

Fig. 19 shows the MOONS' BLDC motor's measured output speed in RPM.

By referring to Fig. 20, it can be observed that the comprehensive Simulink design enables the successful initiation of the BLDC motor and the effective control of its speed.

### 4.2 Setup for Experimental Testing of BLDC Motor

The measurement data are read into the PC by Arduino Mega 2560, which can also process data in real time. The interface can be programmed with Simulink, in MATLAB. The experimental setup consists of seven parts, namely MOONS' BLDC motor, power supply with three DC channel, Arduino Mega 2560, invert circuit, driver circuit, Hall sensors circuit and the host computer. The host computer is used to execute control algorithms and is connected to the BLDC motor. The Arduino Mega 2560 microcontroller is responsible for controlling and regulating the motor speed. The schematic diagram of the hardware implementation is illustrated in Fig. 21, and Fig. 22 shows the corresponding photo of the experimental setup.

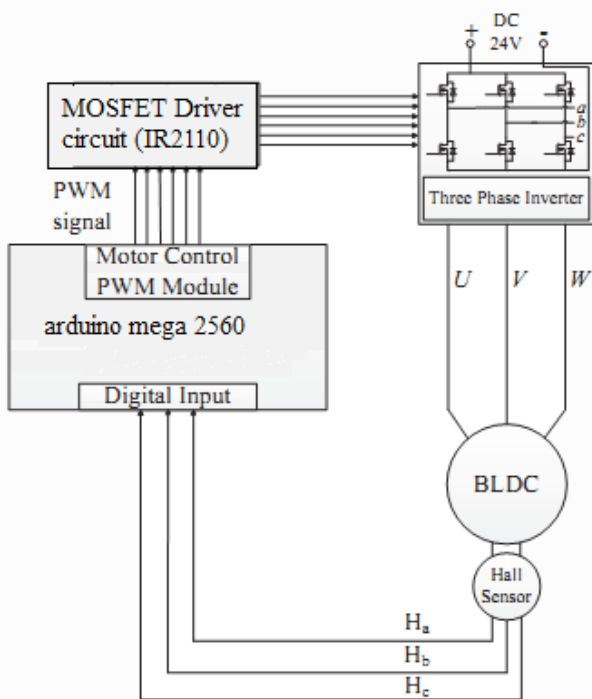


Fig. 21. Schematic diagram of the equipment setup

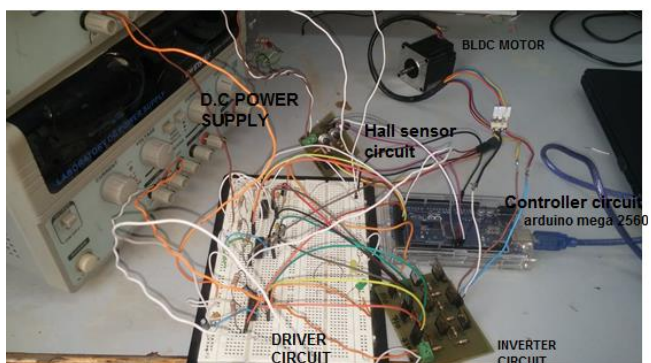


Fig. 22. The photo of the experimental setup of the BLDC motor

### 4.3 Hardware Implementation Details

#### 4.3.1 Arduino Mega 2560 (Motor controller)

The Arduino Mega 2560 serves as a hardware interface, facilitating communication and interaction between the computer, sensors and actuators. To acquire real input and output data from the physical system, the Arduino Mega 2560 is programmed using MATLAB's Arduino block set on Simulink. This enables the generation of a PWM signal that controls the six switches of the three-phase inverter. Next, the inverter is connected to the motor's phases (U, V and W) and receives speed measurement from the Hall sensors ( $H_1, H_2$  and  $H_3$ ). Fig. 23 illustrates a simplified block diagram depicting the motor controller.

#### 4.3.2 Inverter circuit design

Inverters are power electronic circuits that convert direct current (DC) to alternating current (AC) power sources. The inverter output has three alternating voltage signals with a delay of  $120^\circ$ . The inverter has a DC link that will be converted by switching MOSFETs into AC voltage. The BLDC motor is driven by three-phase inverters. In our system, we need to use a three-phase inverter with six switches (N-channel MOSFETs), which are divided into three up/down arms. Fig. 12 depicts a circuit diagram for an inverter; the upper arm includes upper bridge switches (A+, B+ and C+), and the lower arm includes lower bridge switches (A-, B- and C-). The inverter circuit design is accomplished using Proteus 8.6 software. In this circuit, the MOSFET (IRF740 N) is used. Fig. 24 shows the practical implementation of printed circuit board (PCB) for the inverter.

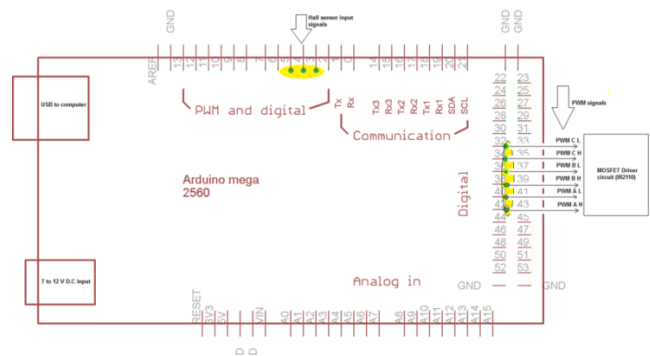


Fig. 23. Simplified diagram of the motor controller

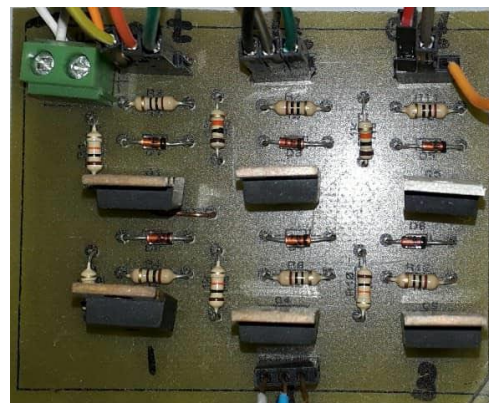


Fig. 24. The photo of the PCB board

When the bridge switch encounters a situation wherein, at the same time, one arm up and another down are turned on, a short circuit, resulting from serious damage to the circuit, will occur. As a result, there are six types of switching patterns based on the complementary switching principle of the up and down bridge. However, this solution results in the high-side gate driving problem. Special gate driving techniques are required for high-side n-channel switches. TLP250 isolated gate driving, transformer isolated gate driving, and bootstrap technique are the solutions to the problem [31]. The bootstrap technique is used in this paper because it is simple. The IR2110 integrated circuit is selected, and the datasheet for the same is available from the ALLDATASHEET website [32]. Additionally, the IR2110 driver itself aids in resolving this issue. In order to prevent the switchers from conducting at the same time as the IR2110, an internal delay is introduced to the structure of the device, as illustrated in Fig. 25.

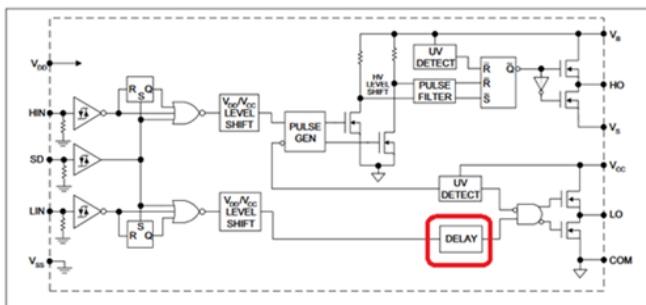


Fig. 25. Internal block diagram of the IR2110

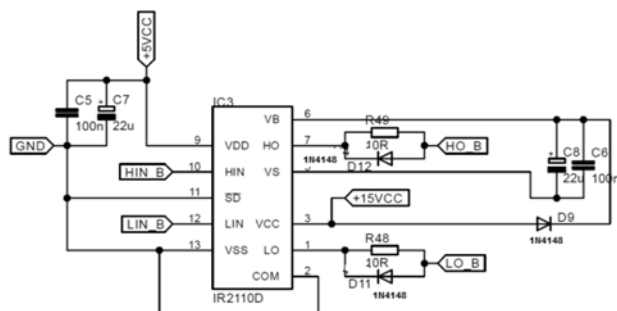


Fig. 26. The electronic block diagram of the IR2110 driver [34]



Fig. 27. The photo of the PCB board for the IR2110 driver circuit

Fig. 26 illustrates the electronic block diagram of the IR2110 drive circuit. The driver circuit is repeated thrice (each driver triggers the complete inverter arm). Additional details can be found in the study of Tahmid [33].

The drive circuit was designed using Proteus 8.6 software for the BLDC motor speed sensor test.

The practical implementation of the IR2110 driver circuit is depicted in Fig. 27 on the PCB. The technical specifications of components and driving parameters are presented in Table 2.

Tab. 2. Technical specifications of components and driving parameters

Component/parameter	Specification
Power source	24 V
Controller	Arduino Mega 2560
Inverter drive	IR2110
MOSFET	IRF740
Switching frequency	500 Hz

### 4.3.3 Hall-effect Sensors

The Hall-effect sensors (H\_1, H\_2 and H\_3) are also described using other terms (in relation to all sensors), for instance as “a transducer that changes the output voltage as a result of a change in the magnetic field”. Fig. 28 shows the Hall-effect principle.

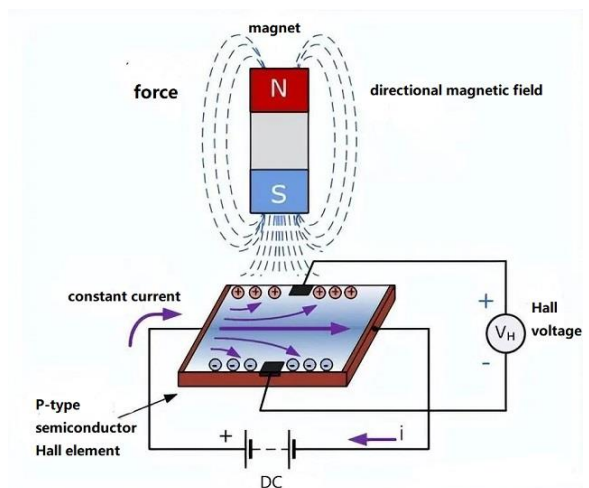


Fig. 28. Hall-effect principle [35]

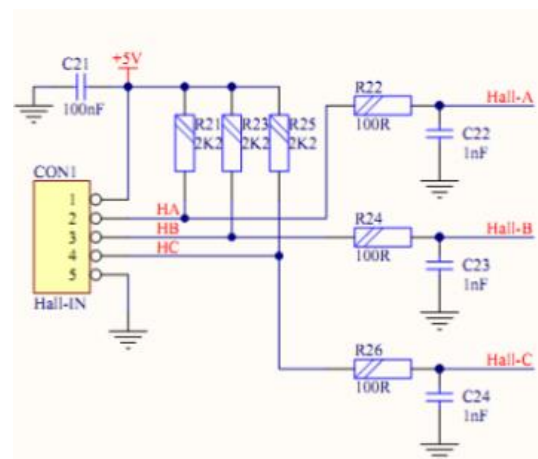


Fig. 29. The block diagram represents the circuitry of the Hall sensor





Fig. 30. The photo of the PCB board for Hall sensor

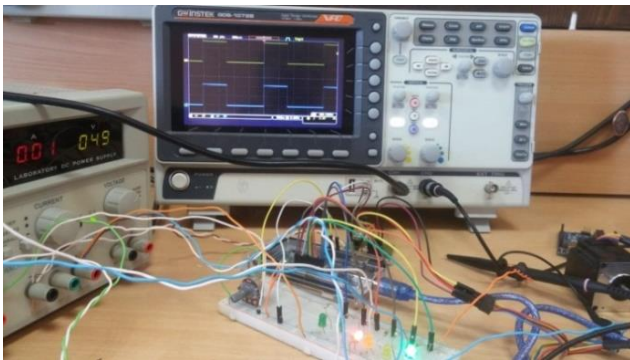


Fig. 31. Testing circuit for Hall sensor

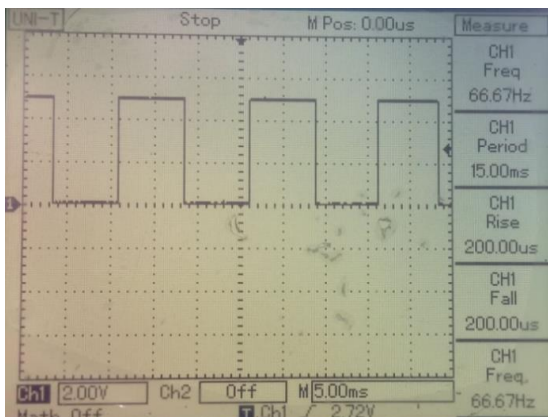


Fig. 32. Visualisation of Hall sensor output with a voltage scale of 2 V/div

Hall-effect sensors are utilised for both rotor position determination and embedded speed detection within the stator itself. This will send a digital signal (high or low), which will detect the position of the rotor [34]. Hall-effect sensors are placed at 120° electrical apart. For every 60°, one of the Hall sensors changes its state. The output of Hall-effect sensor is fed to the controller (Arduino Mega 2650) for the triggering of the MOSFET circuit, which requires a 5 V supply for its sensor. The connection of Hall sensor circuit is shown in Fig. 29.

The Hall-effect sensor circuit is designed using Proteus 8.6 software. Fig. 30 shows the practical implementation of PCB for the Hall-effect sensor circuit. A normally closed switch has been added to the circuit, to induce a fault in the Hall-effect sensor.

The BLDC motor speed sensor was experimentally tested by interfacing the Hall-effect sensor circuit with the Simulink models on the Arduino Mega 2560 board. Subsequently, the digital pins

(32, 34, 36, 38, 40, 42) were connected to six LEDs for monitoring and observation. Fig. 31 shows the testing circuit for the Hall sensor. Then we manually move the BLDC motor step by step at each step until two LED lights are lit.

The waveforms of Hall sensor signals were recorded and stored using a digital storage oscilloscope. Fig. 32 displays the output signal of one of the Hall sensors.

## 5. SIMULATION AND REAL-TIME RESULTS

The proposed Luenberger observer for the BLDC motor was thoroughly evaluated for its effectiveness and accuracy through a combination of simulation and real-time experiments. In the experimental setup, all tests were conducted in real-time, with a sampling time of  $T_s=1$  ms. All experiments for the MOONS' BLDC motor were conducted using an input voltage of 5 V, which corresponds to an operational speed of 416 RPM. The BLDC motor models were developed and simulated using MATLAB/Simulink. Various well-known faults were examined to assess the motor speed, as outlined in Farnen and Zarchi [36]:

- Additive fault: It arises from variations in internal temperature or calibration inaccuracies, leading to alterations in the motor's output behaviour. The additive fault manifests itself when an extra constant value is introduced to the speed sensor output. Two types of additive faults exist: abrupt faults that occur instantly, and intermittent faults that appear and disappear repeatedly. Abrupt faults are commonly attributed to hardware damage, while intermittent faults are often caused by partial wire damage [3]. In Simulink, the abrupt fault model can be simulated using a step function, whereas the intermittent fault model can be represented by a series of pulses with varying amplitudes.
- Multiplicative fault: This type of fault occurs when multiplier coefficients are applied to the sensor, resulting in distorted measurements or outputs. An incipient fault is an example of this type of fault, characterised by gradual parametric changes occurring over time. It is often attributed to the natural aging process of the system [3]. To model the incipient fault in Simulink, a ramp function can be utilised to accurately represent the gradual development of the fault over time.
- Sensor fault: This is a critical fault that occurs when a sensor fails at a specific time, typically due to disconnection, leading to a constant zero output after the fault occurrence [3]. To simulate a sensor fault in Simulink, the speed signal can be effectively modelled by multiplying it by zero. This fault was experimentally executed by briefly disconnecting the speed sensor from the motors and then re-connecting it, or by temporarily cutting off the power to the motors.

The fault detection thresholds are established by considering the highest values observed in the residuals from several experiments. The two threshold types are as follows: fixed threshold and adaptive threshold [37]. The paper utilised a fixed-type threshold, and several fault-free motor tests were conducted to determine the values of both the upper and lower thresholds. When the residual value exceeds a predefined threshold, the alarm is activated to signal the presence of a fault. The alarm was implemented using a buzzer device. Furthermore, the alarm is triggered upon the occurrence of this fault. In Fig. 33, the Luenberger observer for the BLDC motor is presented as part of the simulation design.

This work presents a real-time Simulink design of the fault detection method using the Luenberger observer for the BLDC motor, as depicted in Fig. 34. In this paper, we will demonstrate the remarkable consistency between simulation and practical implementation results for each type of fault applied.

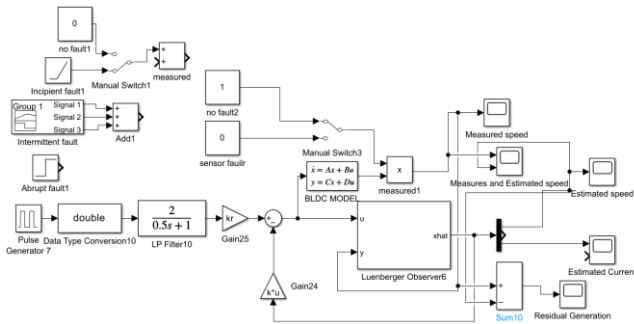


Fig. 33. Luenberger observer simulation design for BLDC motors

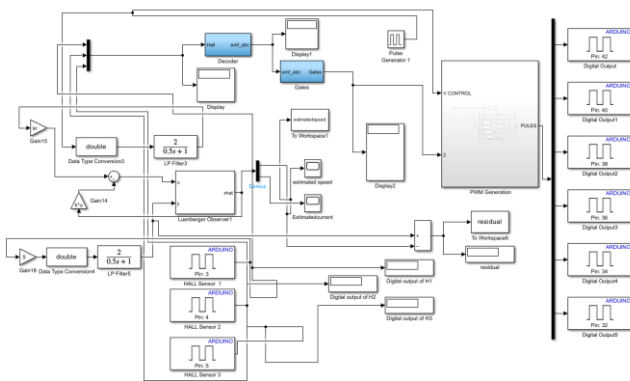


Fig. 34. Luenberger observer simulation (in real time) for BLDC motor

### 5.1 Simulation Results: Normal Operation of BLDC Motor without Faults

In this particular simulation, the speed of the BLDC motor is fault-free, meaning that no faults are intentionally introduced or present in the system during the test. Fig. 35 presents the simulated output and estimated speeds of the BLDC motor, while Fig. 36 illustrates the resulting residual output.

Fig. 36 clearly demonstrates that the residual generation signal is nearly zero, providing an accurate determination of the threshold value.

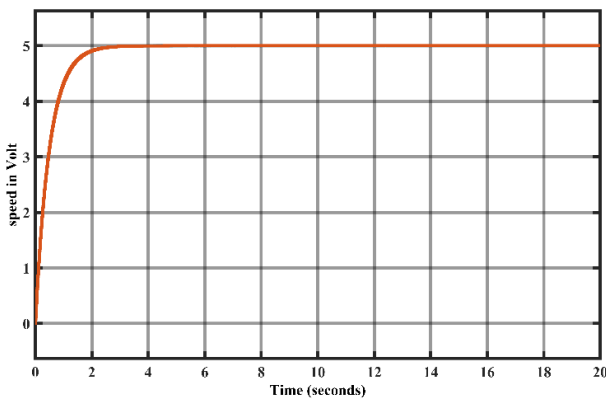


Fig. 35. Simulation of the BLDC motor's output and estimated speeds without fault

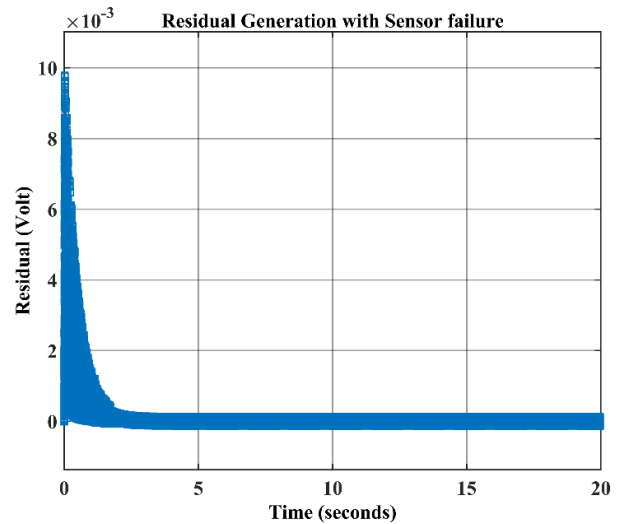


Fig. 36. Simulation of residual output for a BLDC motor without a fault

### 5.2 Experimental Results: Normal Operation of BLDC Motor without Faults

This experiment was conducted under normal conditions, with no fault present in the speed sensor of the BLDC motor. The implementation of this experiment was carried out in real-time, enabling online execution. In Fig. 37, the output speeds are illustrated in both their measured and estimated forms, with no faults detected.

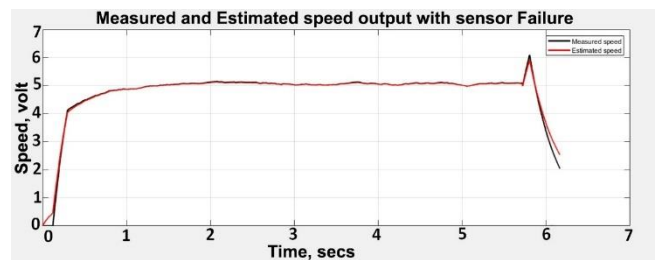


Fig. 37. Measured and estimated speeds with no fault.

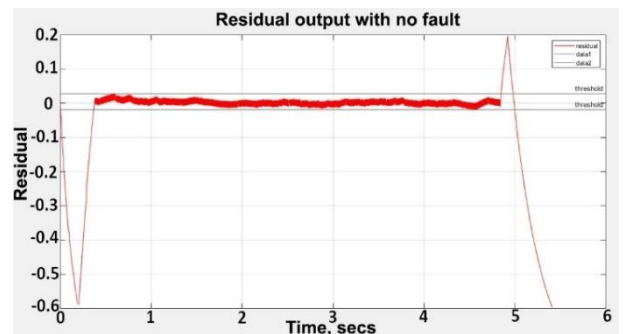


Fig. 38. Residual output with upper threshold value 0.05 and lower threshold value  $-0.01848$

The obtained results closely resembled the simulation results when no fault was present. Multiple experiments were conducted on the fault-free BLDC motor to establish the threshold value based on the residual generation. During the first test, the upper threshold value was set to 0.05, and the lower threshold value was  $-0.01848$ , as indicated in Fig. 38. In the second test, the

upper threshold value was adjusted to 0.0425, while the lower threshold value remained at  $-0.01848$ , as shown in Fig. 39. Accurate knowledge of the threshold values is crucial for designing the alarm system to detect faults in the motor speed.

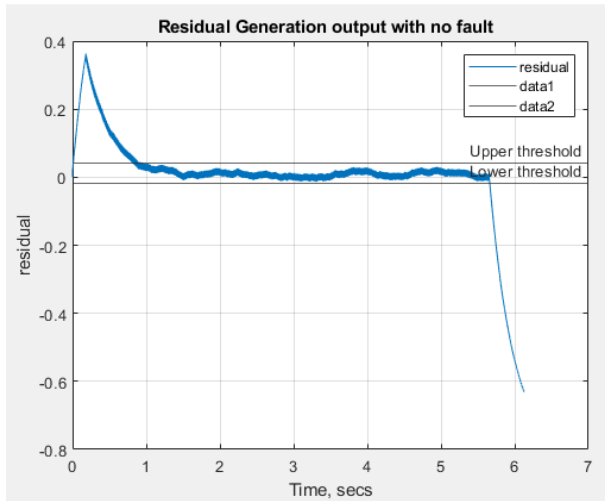


Fig. 39. Residual output with upper threshold value 0.0425 and lower threshold value  $-0.01848$

Upon examining Fig. 39, it becomes apparent that the non-zero residual generation is attributed to the presence of noise in the BLDC motor and potential errors in the observer design parameters. To mitigate false alarms, higher and lower threshold values of 0.05 and  $-0.01848$ , respectively, were chosen for the residual value. These threshold values were adopted for BLDC motor tests to implement the alarm.

### 5.3 Simulation Results: Abrupt Fault in BLDC Motor

An abrupt fault is induced in the speed sensor of the BLDC motor at the eighth second through the addition of a constant value to the sensor reading, as illustrated in Fig. 33. This fault did not work in real time for the BLDC motor. In Fig. 40, the output simulation is shown together with an estimate of the BLDC motor speeds, and Fig. 41 shows the residual output. The fault detection method proved to be successful at the time of its occurrence.

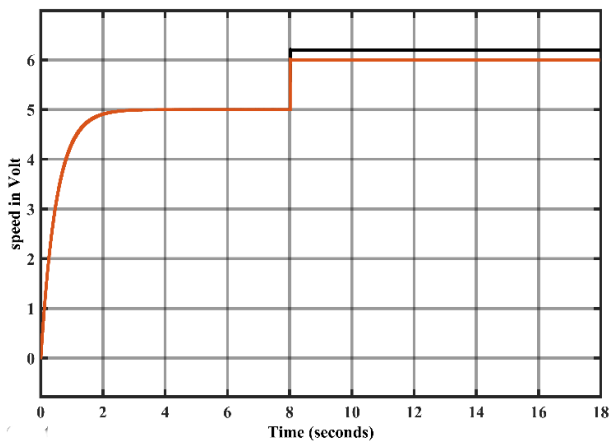


Fig. 40. Simulation of a BLDC motor's output and estimated speeds with an abrupt fault

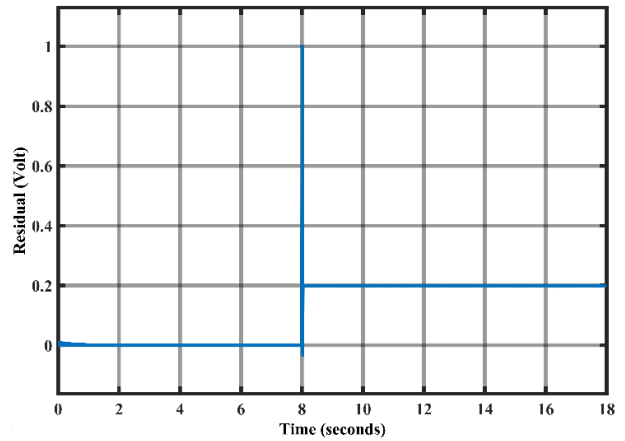


Fig. 41. Residual output simulation with an abrupt fault

Upon analysing the data presented in Fig. 41, it becomes evident that the residual value exceeds the established threshold level. The successful detection of the fault, as indicated by the observation, confirms the occurrence of a fault in the sensor's speed.

### 5.4 Analysis of Sensor Incipient Fault in BLDC Motor: Simulation Results

In Fig. 33, the application of an incipient fault to the speed sensor of the BLDC motor is depicted. The incipient fault was applied to the output speed at the 12th second. Simulink was utilised to simulate the fault applied to the BLDC motor. Fig. 42 displays the simulation output and estimated speeds of the BLDC motor, whereas Fig. 43 showcases the residual output. These figures provide evidence of the successful completion of the fault detection process at the time of its incidence.

In Fig. 43, it can be observed that the residual value exceeds the threshold value at the beginning of the 12th second, signifying a fault in the speed sensor.

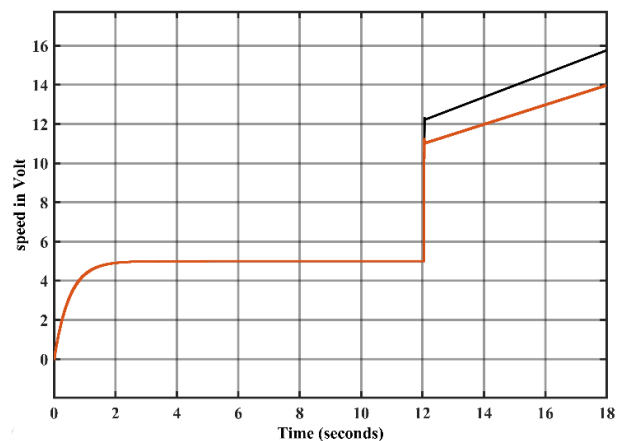


Fig. 42. Simulation of a BLDC motor with an incipient fault's output and estimated speed

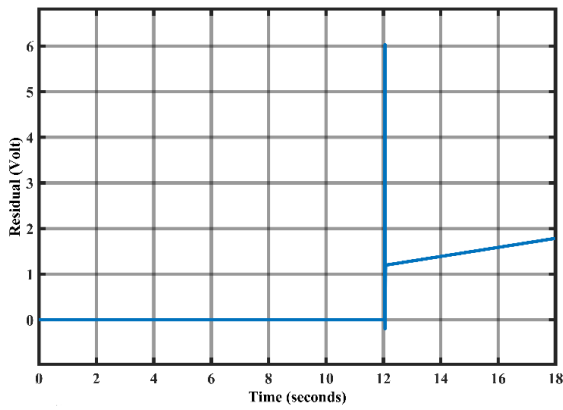


Fig. 43. Residual output simulation with an incipient fault

### 5.5 Simulation Results: Intermittent Fault in BLDC Motor

In this section, an intermittent fault is introduced to the speed of the BLDC motor, as illustrated in Fig. 33. In this implementation, a constant value is periodically added to the sensor reading. At the fifth, eighth and 11th seconds, intermittent faults with the amplitudes of 1, 1.5 and 2, respectively, were applied to the output speed. Each signal is characterised by a pulse width of 1 s. The implementation of this fault for the BLDC motor was not performed in real-time; instead, it was executed as a simulation using Simulink. Fig. 44 depicts the simulation results of the BLDC motor's output and estimated speeds, while Fig. 45 shows the residual output. The success of the fault detection technique is apparent.

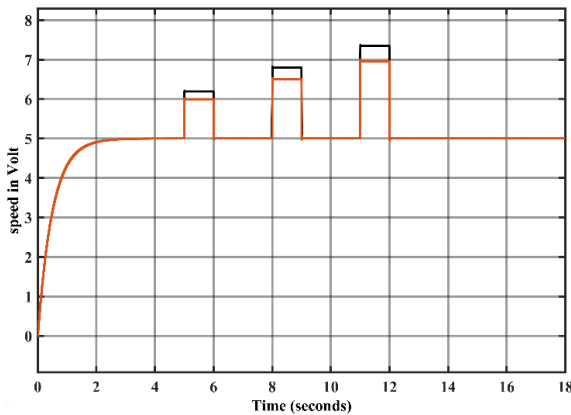


Fig. 44. Simulation of a BLDC motor's output and estimated speeds with an intermittent fault

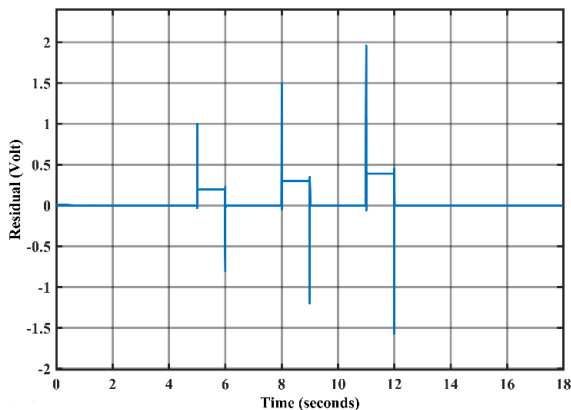


Fig. 45. Residual output simulation with an intermittent fault

In Fig. 45, the observed residual value exceeds the pre-established threshold level.

### 5.6 Analysis of Sensor Fault in BLDC Motor: Simulation Results

To simulate these faults, it is essential to apply a multiplication factor of zero to the speed signal. In Fig. 46, both the simulated and estimated speeds of the BLDC motor are illustrated, while Figure 47 displays the residual output.

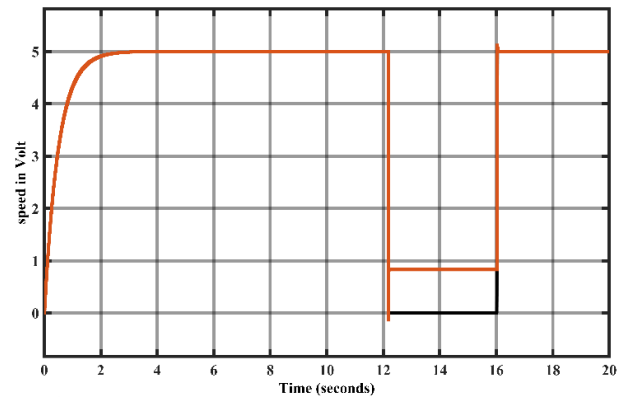


Fig. 46. Simulating BLDC motor speeds in the presence of sensor fault

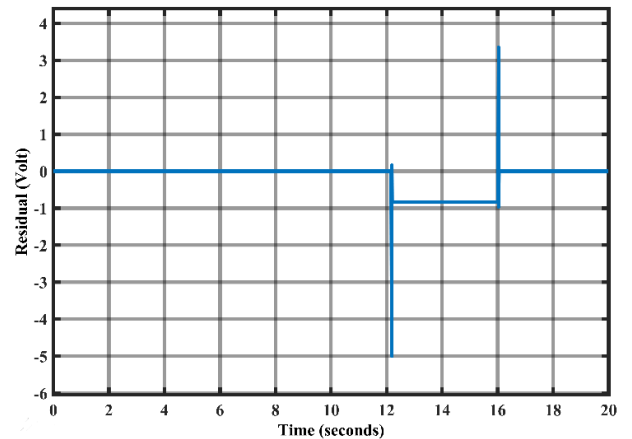


Fig. 47. Residual output simulation with sensor fault

As shown in Fig. 47, the residuals value at the moment of the fault falls below the threshold value, indicating successful fault detection.

### 5.7 Sensor Fault in BLDC Motor Experiment Results

A sensor fault occurs when the speed sensor is disconnected from the BLDC motor. The implementation was specifically conducted on the output speed at the 2nd second. After 1 s, the sensor was reconnected. This fault was implemented in real-time during online operation. In Fig. 48, the response of the measured and estimated speeds is depicted, while Fig. 49 illustrates the output of the residual generation. The fault detection process was successfully executed in this scenario.

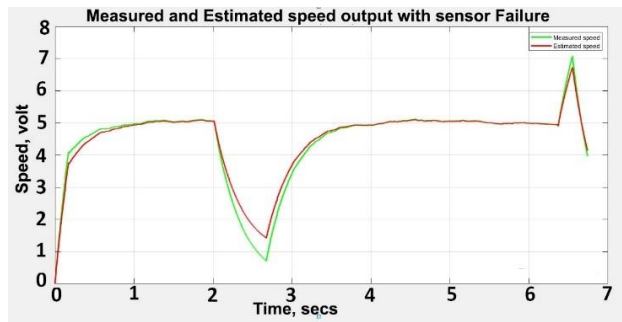


Fig. 48. Measured and estimated speeds with sensor fault

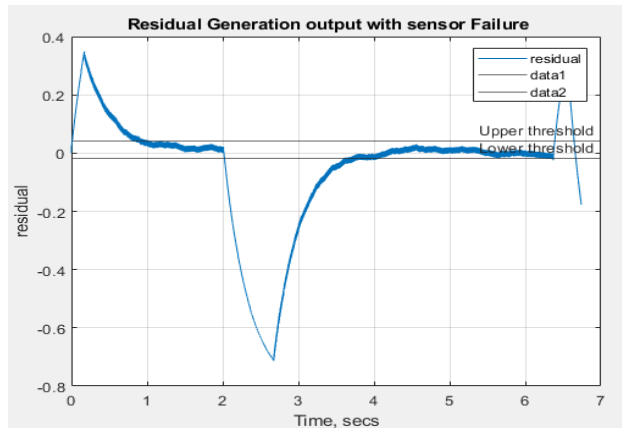


Fig. 49. Output residual with a faulty sensor

In Fig. 49, it can be observed that a sensor fault is detected when the residual value falls below the threshold at the time of the fault occurrence. Once the fault is detected, the alarm is activated as an indication of its presence.

## 6. CONCLUSION AND FUTURE WORK

This research explores the application of model-based methods, specifically the utilisation of Luenberger observers, to detect potential faults in the speed sensor of a BLDC motor. The proposed method's effectiveness was evaluated by subjecting the speed sensor of the BLDC motor to four different fault types: sensor fault, abrupt fault, intermittent fault and incipient fault. The proposed method involves the deployment of a threshold algorithm for generating a residual signal that serves as an indicator of sensor fault. The effectiveness of the proposed method is validated through a combination of simulation and real-time experiments. It is recommended that an adaptive threshold can be used for residual signal in a future work.

## REFERENCES

1. Gaeid, K.S. Fault Tolerant Control of Induction Motor. *Modern Applied Science*. 2011 Aug 3,5(4).
2. Macfos. Detailed Brushless DC Motor Working Principle and Applications [Internet]. Robu.in | Indian Online Store | RC Hobby | Robotics. 2020. Available from: <https://robu.in/brushless-dc-motor-working-principle-construction-applications/>
3. Brushless DC motor application, 8 BLDC Motor Application examples. SONTIAN Auto Parts. 2021. Available from: <https://sontianmotor.com/brushless-dc-motor-applications/>

4. Sova V, Chalupa J, Grepl R. Fault tolerant BLDC motor control for hall sensors failure. In: 2015 21st International Conference on Automation and Computing (ICAC). IEEE 2015.
5. Alkaya A. Novel data driven-based fault detection for electromechanical and process control systems. *Cukurova University of Natural and applied sciences*. 2012.
6. Miljković D. Fault detection methods: A literature survey. *Opatija: Proceedings of the 34th International Convention MIPRO*. 2011; 750-755.
7. Abed W. Robust Fault Analysis for Permanent Magnet DC Motor in Safety Critical Applications [Internet]. Plymouth University; 2015. Available from: <http://dx.doi.org/10.24382/1328>
8. Li L. Robust fault detection and diagnosis for permanent magnet synchronous motors. The Florida State University; 2006.
9. Gaied K. S, Ping H.W. Wavelet fault diagnosis and tolerant of induction motor: A review. *International Journal of Physical Sciences*. 2011; 358-376.
10. Isermann R, Ballé R. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*. 1997 May;5(5):709-19.
11. Isermann R. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. 2006<sup>th</sup> ed. Berlin, Germany: Springer 2006.
12. Skora M, Kowalski C.T. The influence of sensor faults on PM BLDC Motor Drive. In: 2015 International Conference on Electrical Drives and Power Electronics (EDPE). IEEE 2015.
13. Tashakori A, Ektesabi M. A simple fault tolerant control system for hall effect sensors failure of BLDC Motor. In: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA). IEEE 2013.
14. Zandi O, Poshitan J. Brushless DC motor bearing fault detection using hall effect sensors and a two-stage wavelet transform. In: 2017 Electrical Engineering (ICEE). Iranian Conference on. IEEE 2018.
15. Allous M, Zanzouri N. Fault tolerant control of system with simultaneous actuator and sensor faults. In: 2017 International Conference on Control, Automation and Diagnosis (ICCAD). IEEE 2017.
16. Saoudi G, El Harabi R, Abdelkrim M. N. Graphical Linear Observers for Fault Detection: The DC Motor Case Study. In: 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13). IEEE; 2021.
17. Al-Mutayeb Y, Almobaied M. Luenberger observer-based speed sensor fault detection of BLDC Motors. In: 2021 International Conference on Electric Power Engineering – Palestine (ICEPE- P). IEEE 2021.
18. Chan HL, Woo KT. Closed loop speed control of miniature brushless DC motors. *J Autom Contr Eng [Internet]*. 2015; 329-35. Available from: <http://dx.doi.org/10.12720/joace.3.4.329-335>.
19. MOONS – moving in better ways! [Internet]. Moonsindustries.com. 2019. Available from: <https://www.moonsindustries.eu/>
20. Oguntoyinbo O. PID control of brushless DC motor and robot trajectory planning simulation with MATLAB@SIMULINK@ Technology and Communication 2009 Acknowledgements [Internet]. Available from: <https://www.theseus.fi/bitstream/handle/10024/7467/Oludayo%20Oguntoyinbo.pdf?sequence=1>
21. Aghaee M, Jalali AA. BLDC Motor Speed Control Based on MPC Sliding Mode Multi-Loop Control Strategy – Implementation on Matlab and Arduino Software. In: *Electrical Engineering (ICEE), Iranian Conference on*. IEEE 2018.
22. Ridwan M, Yuniarto MN, Soedibyo. Electrical equivalent circuit-based modeling and analysis of brushless direct current (BLDC) motor. In: 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA) 2016.
23. Horváth Z, Molnárka G. Design Luenberger Observer for an Electro-mechanical Actuator. *Acta Technica Jaurinensis [Internet]*. 2014; 7(4). <http://dx.doi.org/10.14513/actatechjaur.v7.n4.313>
24. Sellami L. Simulink Model of a Full State Observer for a DC Motor Position, Speed, and Current. *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV) 2014*; 1-6.

25. Eissa MA, Ahmed MS, Darwish RR, Bassiuny AM. Model-based sensor fault detection to brushless DC motor using Luenberger observer. In: 2015 7th International Conference on Modelling, Identification and Control (ICMIC). IEEE 2015.
26. Sadun AS, Jalani J, Sukor JA. A comparative study on the position control method of dc servo motor with position feedback by using Arduino. In: 2015 Proceedings of Engineering Technology International Conference (ETIC-2015). 10-11.
27. Agrawal K, Gandhi A., Shah MT, Gojiya MV. Design, analysis and realization of SVPWM using embedded code generation technique for a three phase, two level inverter. 2016 International Conference on Electrical Power and Energy Systems (ICEPES); 2016.
28. Megalingam RK, Vadivel SR, Pula BT, Reddy Sathi S, Gupta US. Motor Control Design for position measurement and speed control. In: 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE 2019.
29. Tibor B, Fedak V, Durovsky F. Modeling and simulation of the BLDC Motor in Matlab Gui. In: 2011 IEEE International Symposium on Industrial Electronics. IEEE 2011.
30. Johansson M. Evaluation of Sensor Solutions & Motor Speed Control Methods for BLDCM/PMSM in Aerospace Applications [Internet]. 2017. <https://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-61437>
31. Balogh L. Design and application guide for high speed MOSFET gate drive circuits. In: 2001 Power Supply Design Seminar SEM-1400. Topic (Vol.2).
32. IR2110 datasheet(PDF) – international rectifier – Datasheet search site for Electronic Components and Semiconductors [Internet]. Available from: <https://www.alldatasheet.com/datasheet-pdf/pdf/82793/IRF/IR2110.html>
33. Tahmid P. Using the high-low side driver ir2110 - explanation and plenty of example circuits [Internet]. Available from: <http://tahmidmc.blogspot.com/2013/01/using-high-low-side-driver-ir2110-with.html>
34. Poovizhi M, Kumaran MS, Ragul P, Priyadarshini LI, Logambal R. Investigation of Mathematical Modelling of Brushless DC motor(bldc) drives by using MATLAB-Simulink. In: 2017 International Conference on Power and Embedded Drive Control (ICPEDC). IEEE 2017.
35. Storr, W. ElectronicsTutorials. Hall effect sensor and how magnets make it works [Internet]. Basic Electronics Tutorials. 2018. Available from: <https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html>
36. Farnen T, Zarchi M. Fault Diagnosis of an Electrical Winch based on Structural Analysis. University of Agder 2018.
37. Frank PM, Ding SX, Marcu T. Model-based fault diagnosis in technical processes. Transactions of the Institute of Measurement and Control [Internet]. 2000;22(1):57-101. Available from: <http://dx.doi.org/10.1191/014233100676786880>

Yousef Al-Mutayeb:  <https://orcid.org/0000-0002-5476-6488>

Moayed Almobaied:  <https://orcid.org/0000-0001-7912-2673>

Mohamed Ouda:  <https://orcid.org/0000-0002-4804-0015>