Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electrotechnique
Commandes Electriques

Réf. : Entrez la référence du document

Présenté et soutenu par :
Presented and supported by: ELHAMAMI Mariem & BECHA Roufaida

Le : Lundi 26 mai 2025

Optimization of BLDC motor based on HHO Algorithems

Jury :

Mme.	BACHAR Rahima	MAB	Universite de Biskra	Presedent
Mme.	RAHOUA Naima	MCB	Universite de Biskra	Superviseur
Mme.	KHELILI Fatiha	MCB	Universite de Biskra	Examinateur

Année universitaire : 2024-2025

Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electrotechnique

Commandes Electriques

Réf. : Entrez la référence du document

Optimization of BLDC motor based on HHO Algorithms

Le : : Lundi 26 mai 2025

Présenté par : ELHAMAMI Mariem & BECHA Roufaida
Avis favorable de l'encadreur : RAHOUA Naima

Signature Avis favorable du Président du Jury

Cachet et signature


بسم الله الرحمن الرحيم

Acknowledgement

Words cannot express my gratitude, and my deepest appreciation to everyone who had supported me and helped me along this journey.

First of all, and before anything, I ‘am deeply indebted to my advisor Ms. RAHOUA Naima, your encouragement, kindness, your dedication to education and unwavering belief in me have been the pillars of my strength and inspired me to Seek mastery. Your priceless guidance has played a pivotal role in influencing the trajectory and depth of my research, shaping it into what it is today.

I would like to extend my sincere thanks to the Faculty of Electrical Engineering at Mohamed Khider University - Biskra for providing a dynamic and enriching academic environment. Their dedication to fostering collaboration and intellectual growth has profoundly influenced my academic and professional development and expended my perspective and enriching my understanding for this field.

I'd like to acknowledge my friends for their support, respect, and uplifting companionship throughout this journey. Your encouragement and the moments we shared have made this experience both meaningful, enjoyable and memorable.

Lastly, I'd like to address a special thanks to my family, whose unwavering love and support have been the foundation of my strength, your belief in me, even during moments when I doubted myself, has been a guiding light through every challenge I’ve faced. All of the emotional and the practical support you gave me; my thanks will never be enough.

I am truly grateful to each of you, for your participation and encouragement and for sharing in this journey alongside me.

الإهداء

"تعلموا العلم وعلموه الناس، وتواضعوا لمن تعلمتم منه ولمن علمتموه" عمر بن الخطاب.

بسم الله الرحمن الرحيم، والحمد لله رب العالمين.

أتقدم بشكر جزيل وامتنان كثير لكل من ساهم في رحلتي العلمية هذه وقدم لي كل الدعم والحب.

هاهنا نحن اليوم نكتب اخر سطور رحلتنا، لا المشوار كان سهلا ولا الطريق كانت قصيرة، لكن بعد توفيق الله عز وجل وتيسيره لما كنا هنا، الحمد لله الذي بلغنا النهايات بفضله وكرمه الحمد لله، دوما وابدا.

الى قدوتي ومصدر فخري، الى ظهري وسندي، والدي الغالي رحمة الله عليه.

الى الشخص الذي احمل اسمه، وصفاته، انا هنا أقف وكم أتمنى وقوفك بجانبي لترى ما صنع لطفك ودعمك مني، مهندسة بدوري اليوم.

الى نور حياتي، وملكة قلبي، الصبورة المجاهدة، الحنونة الطيبة، الى امي اطال الله عمرها وحفظها لنا.

اهدي هذا التخرج لك غاليتي، انت من كنت الشمس المشرقة في أحلك ايامي، وجودك جانبي هو أساس ما انا عليه اليوم وما سأكونه غدا، شكرا يا امي، شكرا يا من علمني المبادئ والقيم ومن كان لي الناصح والصديق عند الشدائد.

إخوتي الأعزاء، حنان، يونس واحمد، شكرا لدعمكم وحبكم الصادق لي، لولا وجودكم لن تكون لحياتي معنى، ادامكم الله نعمة علي واطال اعماركم وافرحكم.

الى عمي وعمتي، لقد كنتم كأبوين لي، لن اوفي ابدا بمقدار صغير من الحب والعطف والدعم الذي قدمتموه لي، كم انا محظوظة لوجودكم معي.

أصدقائي الأوفياء، امينة، رفيدة، سولاف وخولة، اللواتي جعلن الطريق أسهل وأجمل شكرا لكم.

كما أتوجه بالشكر لأساتذتي الكرام وكل الطاقم التربوي لكلية العلوم والتكنولوجيا في جامعة محمد خيضر-بسكرة.

وأتقدم بشكر خاص الى المشرفة رهوة نعيمة، على كل الدعم والارشاد الذي قدمته لنا، لقد كنت مدرسة لنا، تعلمنا منك الكثير.

وأخيراً، أشكر نفسي على المثابرة والإصرار، فبفضل الله تعالى أولا ثم بعزيمتي وصبري وصلت إلى هذه المرحلة، و الحمد لله اوله واخره.

و السلام عليكم ورحمة الله وبركاته.

الهمامي مريم.

الإهداء

الى كل من كلل العرق جبينه ومن ان النجاح لا يأتي الا بالصبر والإصرار الى النور الذي انار دربي والسراج الذي لا ينطفئ نوره بقلبي ابدا، من بذل الغالي والنفيس واستمدت منه قوتي واعتزازي بذاتي، أبي.

الى من جعل الجنة تحت اقدامها وسهلت لي الشدائد بدعائها الى الانسانة العظيمة التي لطالما تمنت ان تقر عينها في يوم كهذا، امي رحمها الله .التي لطالما كان أحد احلامها مشاهدتي خريجة، النور الذي أنار دربي والسراج الذي لا ينطفئ نوره بقلبي توسدها التراب قبل أن يحقق أمنيتها.

إلى مناضلتي واجتهادي الحاضر بقلبي لا يغيب.

الى قرة عيني. اخواتي الغاليات.

لكل من كان عونا وسندا في هذا الطريق. للأصدقاء الأوفياء ورفقاء السنين لأصحاب الشدائد والأزمات الى من افاضني بمشاعره ونصائحه المخلصة. اليكم عائلتي.

اهديكم هذا الإنجاز وثمرة نجاحي الذي لطالما تمنيته ها انا اليوم أكملت واتممت اول ثمراته بفضله سبحانه وتعالى.

فالحمد لله على ما وهبني وان يجعلني مباركا وان يعنني أينما كنت فمن قال انا لها نالها وانا لها وان ابت رغما عنها اتيت بها فالحمد لله شكرا وحبا وامتنانا على البدء والختام.

واخر دعوانا ان الحمد لله رب العالمين.

بشة روفيدة.

Summary

This thesis focuses on empowering engineers and researchers to design and develop brushless DC motors with enhanced efficiency, thereby the primary objective of this project is to regulate the speed of a BLDC motor. The study introduces the fundamentals of the PID controller and calculates its parameters using two methods the Zeigler-Nichols second method and the MATLAB/APPS/ tuning method, which are then applied in the simulation. Following this, the principles and main features of Haris Hawk optimization (HHO) method. A Simulink model incorporating the BLDC motor, PID controller, and HHO controller is developed based on theoretical and mathematical modeling of the control elements. The speed simulation results of the BLDC motor under HHO algorithms reveal improved performance and meet the desired control objectives.

Key terms: BLDC, ESC, PID, Zeigler-Nichols, HHO algorithms, optimization, control, Simulink modeling.

الملخص

تركز هذه الاطروحة على تمكين المهندسين والباحثين على تصميم وتطوير محركات التيار المثمر بدون فرش(BLDC) بكفاءات معززة وشبه مثالية، بالتالي الهدف الأساسي من هذه الدراسة هو تنظيم سرعة المحرك بتقديم اساسيات المتحكم التناسبي-التكاملي-التفاضلي (PID)وحساب معاييره باستخدام ضبط التعرف على النظام باستخدام طريقتين الأولى هي القاعدة الثانية ل Zeigler-Nichols والأخرى باستخدام MATLAB/APPS، ولذلك لإيجاد خصائص المتحكم (PID)والتي يتم تطبيقها بعد ذلك على المحاكاة. يلي ذلك دراسة لمبادئ وخصائص لطريقة تحسين صقر هاريس(HHO) . يتم تطوير نموذج محاكاة Simulink يتضمن محرك (BLDC) ومتحكم(PID) قائم على نظام برمجة (HHO)بناء على النمذجة النظرية والرياضية لعناصر التحكم. نتائج محكاه التحكم في السرعة لمحرك(BLDC) تحت خوارزميات تحسين صقر هاريس (HHO) تكشف عن أداء محسن وتحقق اهداف التحكم المطلوبة.

الكلمات المفتاحية: محرك تيار مستمر بدون فرش(BLDC) ، وحدة التحكّم الإلكترونية بالسرعة(ESC) ، متحكم التناسب والتكامل والتفاضل(PID) ، منظم زيجلر نيكولز، خوارزميات تحسين صقر هاريس(HHO) ، التحسين، التحكم، النمذجة (Simulink).

Table of content

Chapter two: Controller of Harris hawk's

Chapter three :The Application

List of figures

Chapter one:

Chapter two:

Chapter three:

List of tables

List of symbols

V_a =voltage for phase A [V].

V_b = voltage for phase B [V].

V_c = voltage for phase C [V].

V_{ab} =line voltage [V].

V_{bc}= line voltage [V].

V_{ca} = line voltage [V].

e_a= back electromotive force (back-emf) for phase A [V].

e_b = back electromotive force (back-emf) for phase B [V].

e_c = back electromotive force (back-emf) for phase C [V].

T_a =Torque for phase A [N.m].

T_b = Torque for phase B [N.m].

T_c = Torque for phase C [N.m].

T_e = Electromagnetic Torque [N.m].

T_L = Load Torque [N·m].

i_a= phase A current [A].

i_b= phase B current [A].

i_c = phase C current [A].

$R_a = R_b = R_c = R_s = R$ = Stator resistance [Ω].

$L_a = L_b = L_c = L_s = L$ = Self-inductance [H].

M = Mutual inductance [H].

j= Rotor inertia [Kg.m^2].

β= Friction constante.

K_t = Torque constant.

k_e = Back-emf constant.

θ_m = Rotor angular displacement [Radian or Degrees].

ω_m= Rotor speed[rad/sec].

p = Pole number.

$F(\theta_e)$ = The function that gives the trapezoidal waveform of the back-emf.

U_d = Bus voltage [v].

r_a = Line resistance of windings [Ω].

L_a = Equivalent line inductance of windings [H].

i = Line current [A].

ω_n = Natural frequency of the second-order system.

ξ =Damping ratio of the second-order system.

P Kp= Proportional gain of PID controller.

Ki= Integral gain of PID controller.

Kd= Derivative gain of PID controller.

PID= Proportional–Integral–Derivative controller.

PID-ZN =Zegler-Nichols based PID.

$f_{obj}(X)$ =Objective function to optimize.

g(x)≤0 =Inequality constraint.

h(x)=0 = Equality constraint.

x= Current position of the hawk (candidate solution).

X_{rand}= Randomly selected hawk's position.

r_1, r_2 = Random numbers between 0 and 1.

E= Escape energy of the prey (controls exploration vs. exploitation).

E_0= Initial energy of the prey.

t =Current iteration.

T= Maximum number of iterations.

ΔX(t)= Difference between prey and hawk's positions at iteration.

J= Random jump strength of the prey.

ITAE= Integral of Time multiplied by Absolute Error.

ISE= Integral of Squared Error.

IAE= Integral of Absolute Error.

RPM= Rotations per minute (motor speed unit).

N·m= Newton meter (unit of torque).

A= Ampere (unit of electric current).

HHO = Harris Hawks Optimization.

General Introduction

Electric motors play a vital role in contemporary industrial systems and daily life. Among the various types, brushless DC (BLDC) motors are particularly valued for their high efficiency, minimal maintenance requirements, and extended lifespan. Nevertheless, precise control of BLDC motors is crucial to achieve optimal performance, which includes accurate regulation of speed and torque, enhanced efficiency, and minimized vibration and noise.

Historically, BLDC motor speed regulation has relied on conventional methods like PID (Proportional-Integral-Derivative) control. While PID controllers perform well in many scenarios, they often struggle with the nonlinearities and inherent complexities of BLDC motor systems, such as parameter variations and external disturbances. Recent advancements have shifted toward Haris Hawks optimization -based control strategies, which have gained traction for their ability to emulate human decision-making through linguistic rules. This approach provides adaptability in managing nonlinear dynamics and uncertainties, making it particularly suited for systems where precise mathematical models are challenging to define [1].

The primary objective of this work is to design and implement an advanced control system for a brushless DC (BLDC) motor. To achieve this, the following tasks will be undertaken:

- Develop a mathematical model of the BLDC and evaluate PID-based speed control performance.
- Design a Harris Hawks Optimization (HHO)-based PID controller in the BLDC system
- Implement the HHO on the BLDC motor and assess its effectiveness under different load conditions.
- Compare the results of classic PID control and HHO-PID based on response time, overshoot, and robustness.

This work is organized into three chapters:

The first chapter presents a detailed mathematical model of the brushless DC (BLDC) motor. It then examines a classic PID regulator-based speed control strategies for the BLDC motor and concludes with simulation results evaluating the control system's performance.

The second chapter will be designated to the presentation and evaluation of the Harris Hawks Optimization (HHO) method, with the aim of optimizing the gains of the PID regulator.

Chapter Three explores a real-life application of brushless DC (BLDC) motor speed control using an Arduino microcontroller board. The study focuses on identifying optimal speed regulator parameters through a systematic methodology to enhance control performance.

The work will conclude with a general conclusion.

Chapter I:

Generalities, Modeling & Controle of Brushless DC motor

Chapter I: Generalities, Modeling & Controle of Brushless DC motor

I.1 Introduction

For several years, the industry has utilized the Direct Current (DC) motor, which offers the primary advantage of being easily controllable due to the natural decoupling of flux and torque. However, the fragility of the brush-commutator system has always been a disadvantage of the DC motor, limiting its maximum power and speed, and presenting maintenance difficulties and operational interruptions.

This was one of the reasons why BLDC motors are widely recommended today for both high- and low-power applications thanks to their numerous advantages, including high efficiency, superior torque-to-inertia ratio, variable speed capabilities, quiet operation, compact design, and low maintenance [2].

This chapter will study the principles and mathematical modeling of Brushless DC motors. It further analyzes speed control using a classic PID regulator, supported by a MATLAB /Simulink & MATLAB /APPS/system identification simulation result to validate the effectiveness of the proposed control system.

I.2 History of Brushless DC motors

Traditional DC motors are limited by their dependence on mechanical brushes and commutators. The search for improved efficiency and reliability led to the development of brushless DC (BLDC) motors. Fig (I.1) shows a historical diagram tracing the evolution of BLDC motors over the past century [3].

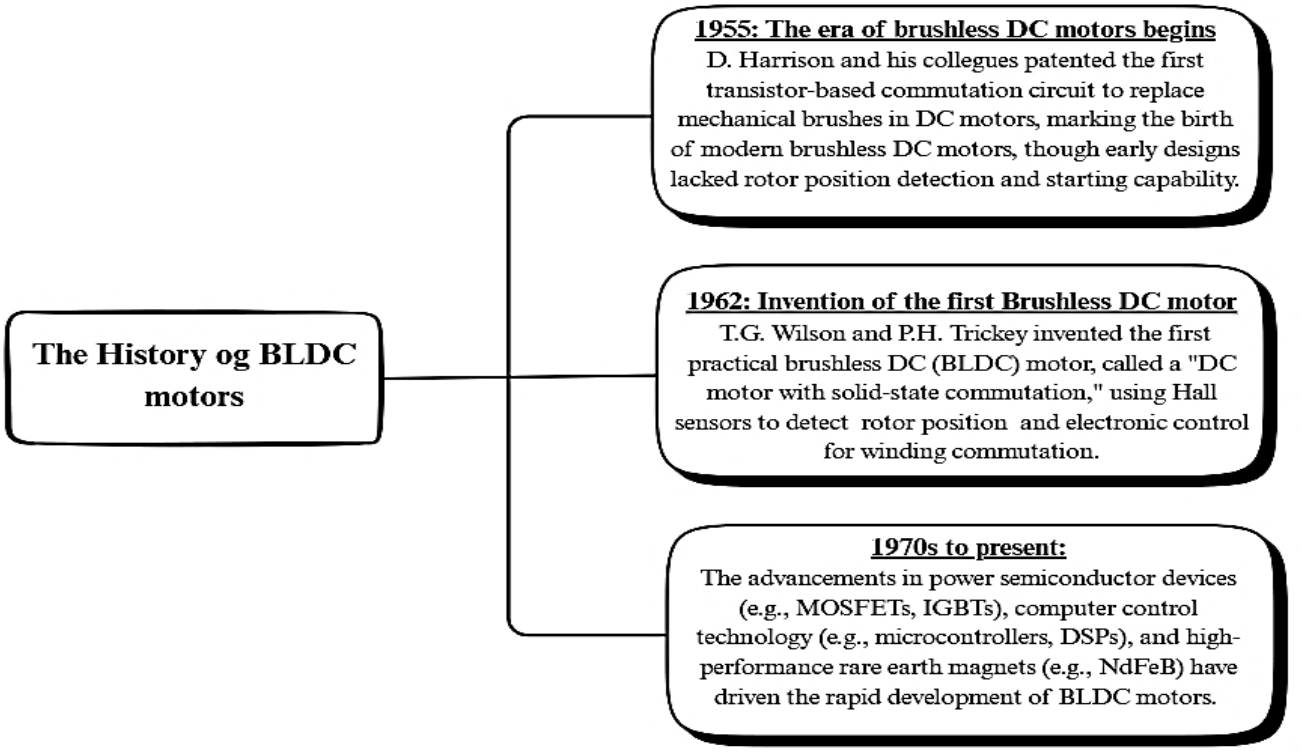


Fig (I.1): History diagram of the BLDC motor.

I.3 General overview of Brushless DC motor

A brushless DC (BLDC) motor is a type of permanent magnet synchronous machine that relies on rotor position detection feedback for operation, also, PMSMs exhibit a sinusoidal back EMF, while brushless DC (BLDC) motors display a trapezoidal form. These motors are typically controlled by a three-phase power semiconductor (e.g. MOSFET's, IGBT, GOT) bridge called electronic speed controllers (ESC) that we will detail following, and require a rotor position sensor like Hall effect sensor to initiate movement and ensure proper commutation sequence [4].

Commutation is crucial process in BLDC motor operation, it is produced by switching the current flow through the motor coils to maintain rotation. It is achieved through the Electronic Speed Controller (ESC) that it is mentioned already in the previous paragraph and will be detailed better in the content. It serves as the intermediary between the power source (typically a battery) and the motor, The system effectively transforms the DC power supply into the alternating current required to drive the motor, enabling continuous and controlled rotation.

Electronic commutation is a key feature of these motors, as it eliminates the need for mechanical brushes, and thereby this design significantly boosts the efficiency and lifespan of BLDC motors [5].

I.4 Structure of the Brushless DC motor

A Brushless DC (BLDC) motor contains three crucial components: the stator, the rotor, and the electronic control system (ESC), which is considered the reason why this machine is special. Each component plays a fundamental role in the motor's operation and performance.

It also has other components that are shown in the Fig (I.2) down below:

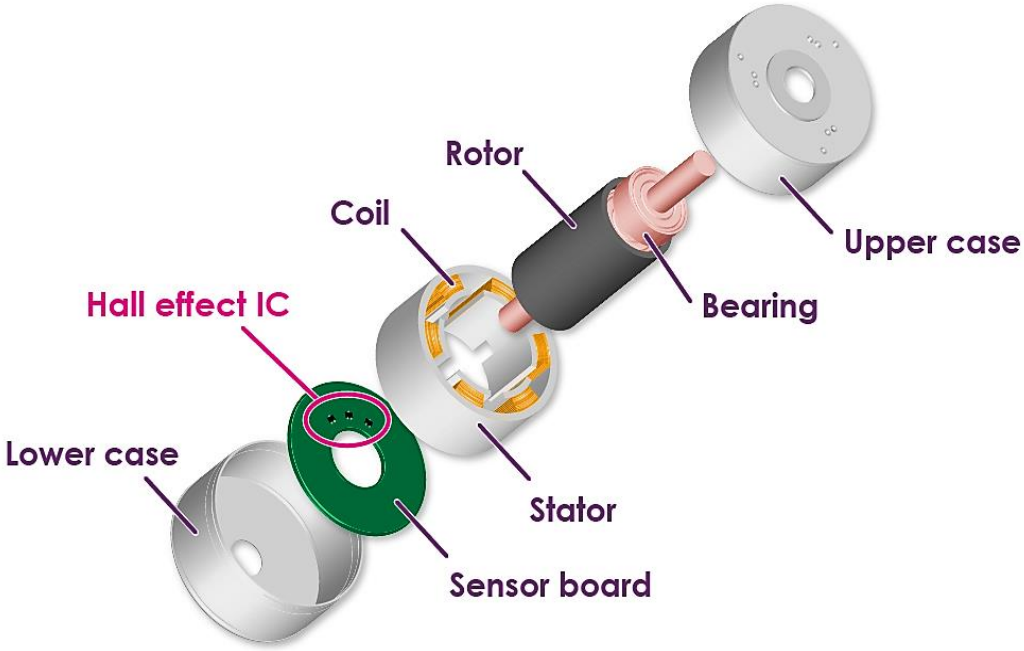


Fig (I.2): Brushless DC motor inner structure [6].

I.4.1 Structure of stator

The stator, a crucial component of brushless DC motors, houses the windings and generates the electromagnetic field essential for operation.it is made from four essential component like we see in the Fig (I.3) down below:

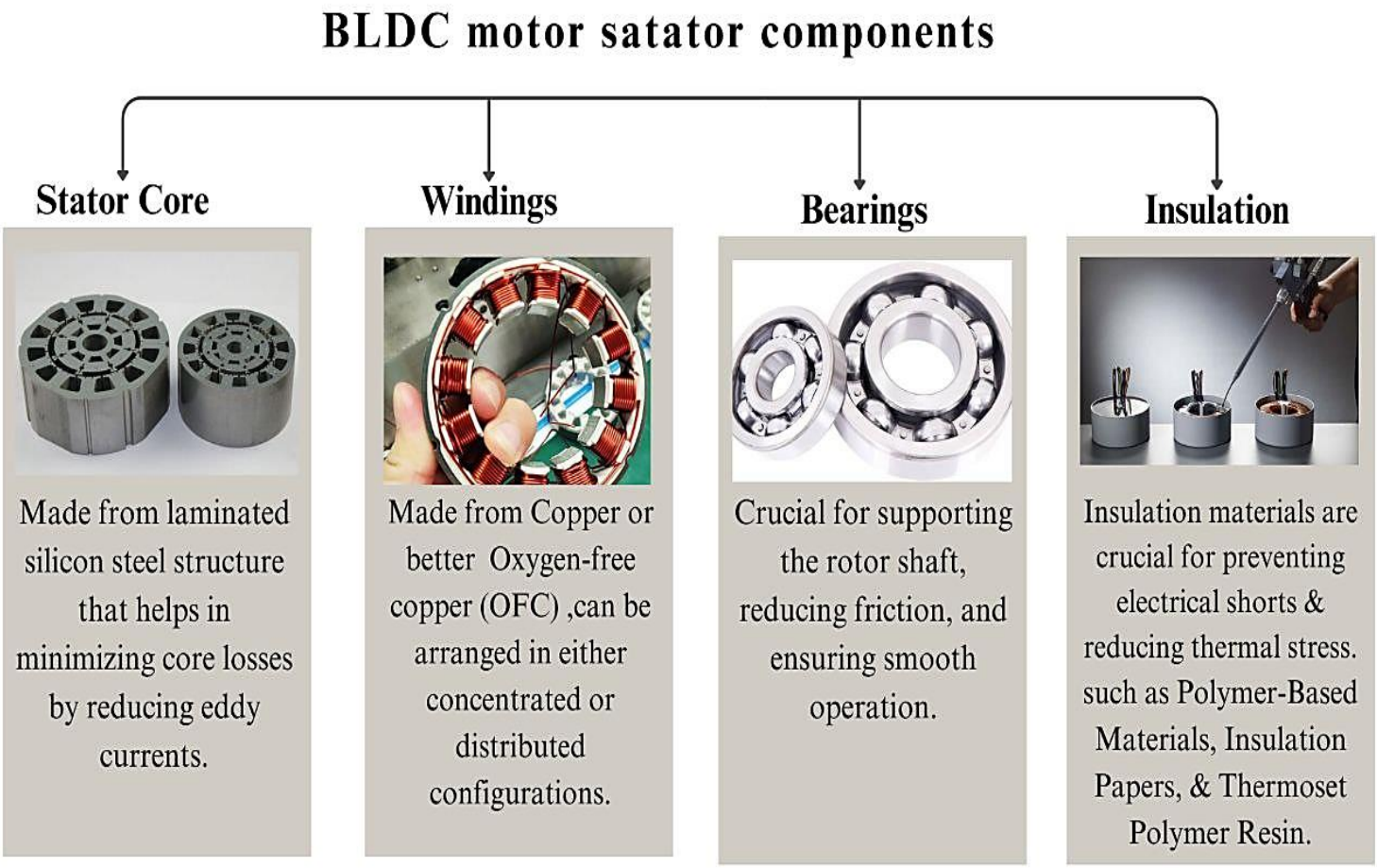


Fig (I.3): Stator structure of BLDC motor [16],[17],[18],[19].

I.4.2 Structure of rotor

In a brushless DC motor, the rotor is made of a high strength permanent magnets like neodymium-iron-boron (NdFeB) and samarium cobalt (SmCo) for the aim to generate a strong magnetic field to assure the continues rotation, also its right design, its core configuration and the pole pair numbers optimizes his performance in different applications [7].

Ther is two main types of rotors in the brushless dc motor as shown in the Fig (I.4) beneath:

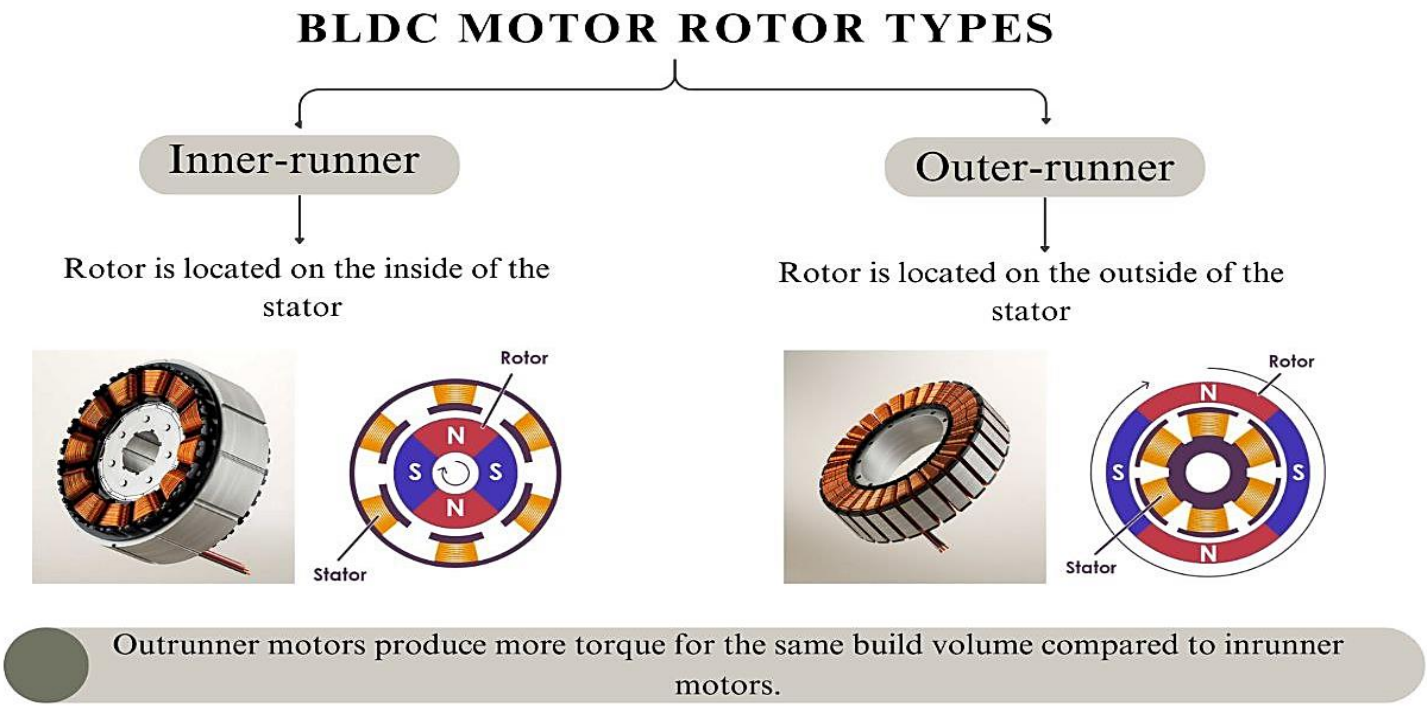


Fig (I.4): Rotor structure of BLDC motor [20],[21].

I.4.3 Electronic speed controller (ESC)

The Electronic Speed Controller (ESC), also called the inverter is key component in modern brushless DC (BLDC) motors making them unique and distinctive. Historically, the invention of electronic speed controller is not to no single inventor, it is described as the product of the development of: power electronics, microcontrollers, and essentially to the adaptation to the BLDC motors advancements [3],[27].

In the Fig (I.5) there is a real picture of how the ESC is connected with the BLDC motor:

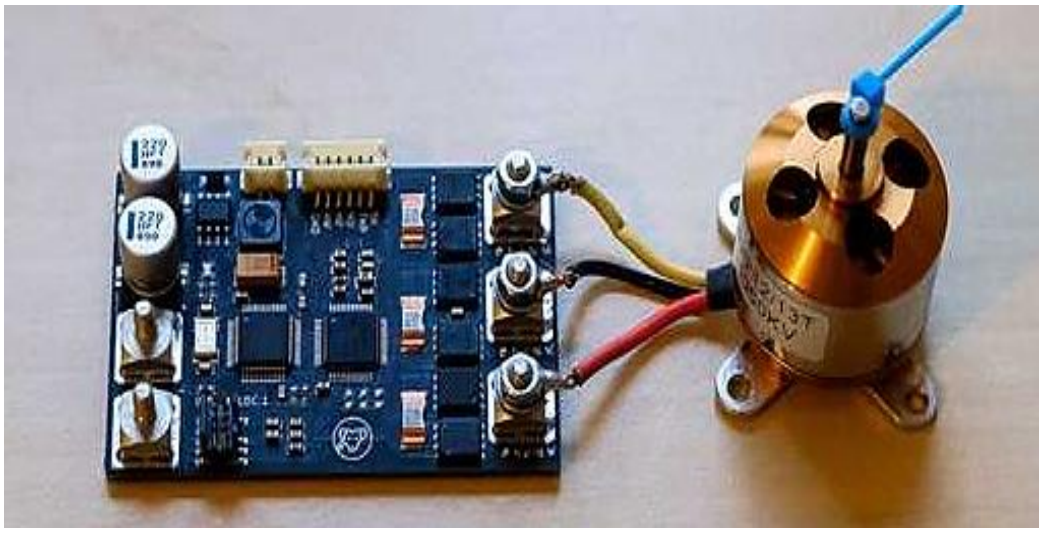


Fig (I.5): Real image of an ESC connected to BLDC motor [22].

Basically, the ESC serves as an interface between power source and the motor, it is used to control the motor speed by converting DC input from the power source (usually batteries) to a three phase AC trapezoidal waveform output [27], The ESC offers a precise speed and torque control, quiet and smooth operating, low maintenance and a light compact design.

I.4.3.1 Components of an ESC:

An electronic speed controller typically comprises five crucial components that can be resumed and explained in the Fig (I.6):

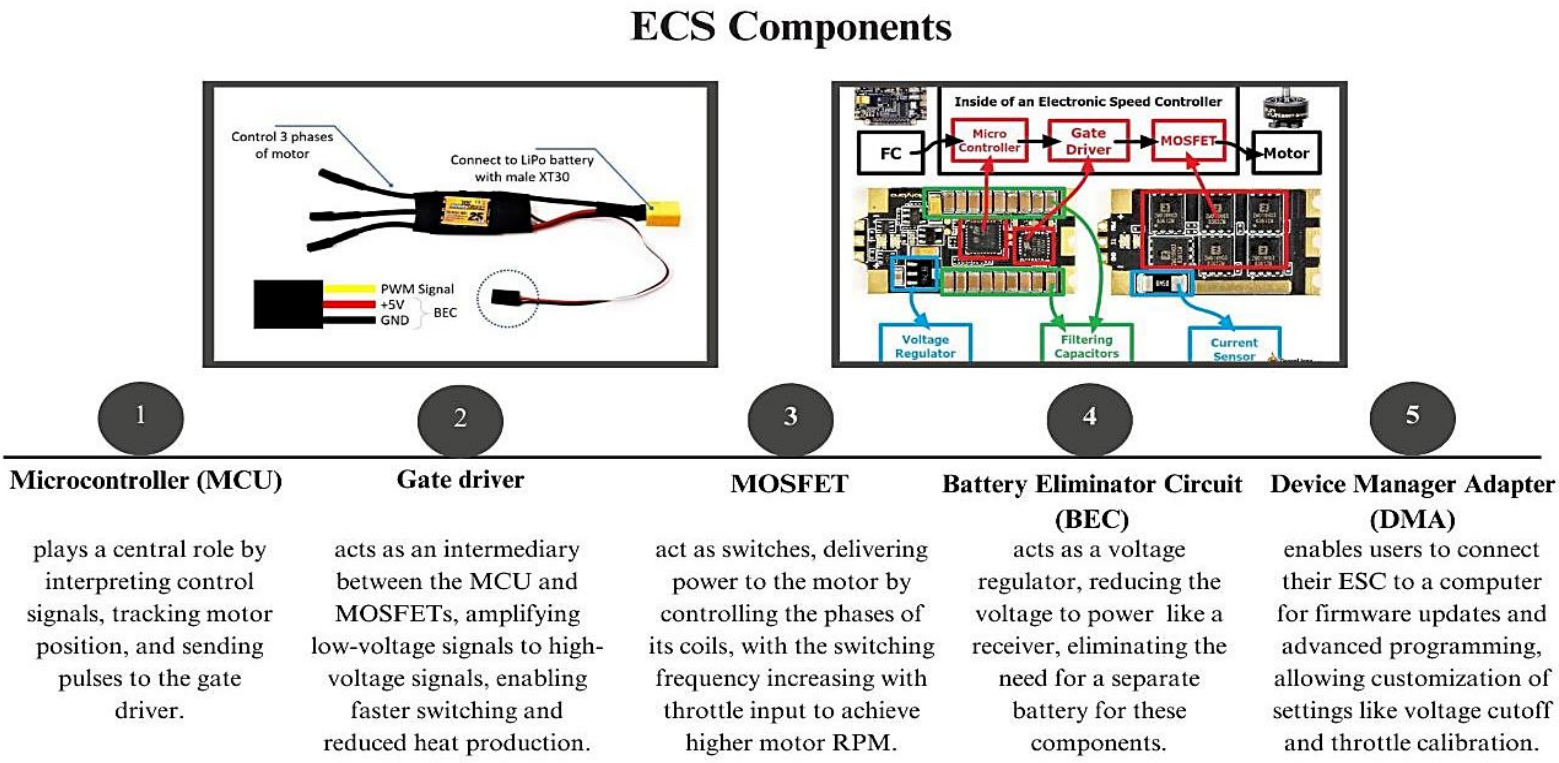


Fig (I.6): The BLDC motor ESC components [23],[24].

I.4.3.2 Six step Commutation of BLDC motor

Also known as trapezoidal commutation, it is a control method for BLDC motors, it operates by sequentially energizing two phases or two stator winding creating a stronger rotating magnetic field, while the third phase /windings remain unpowered.

Now, all of this powering and empowering of the phases is based on rotor position detection feedback and since there are two types of brushless DC motors based on sensor usage (sensor-based and sensorless), as shown in the Fig (I.7) in the content underneath, means there are also two corresponding types of commutation methods based on six step commutation [29],[30],[31] [32].

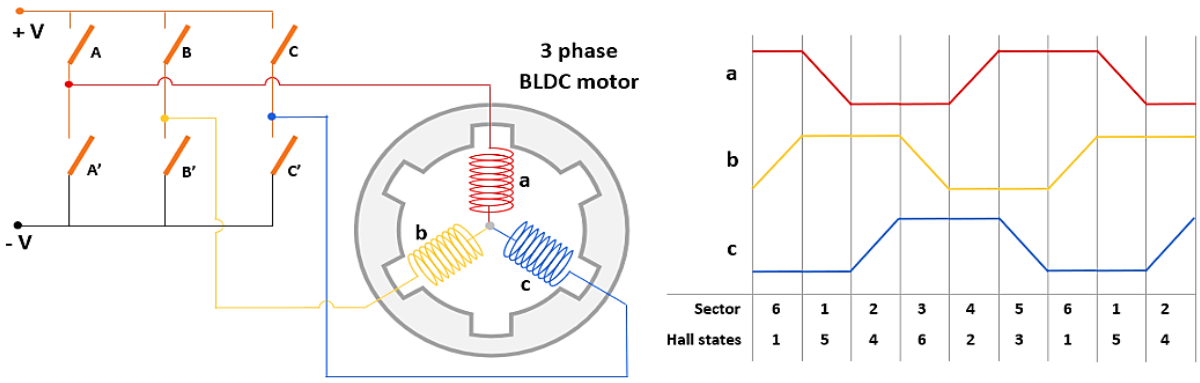


Fig (I.7): Six step commutation of a BLDC motor [28].

1- Sensor-Based Commutation:

- Uses three Hall Effect sensors positioned with 120° electrical degrees apart for precise rotor position detection.
- Offers smooth operation at low speeds and reliable starting under various loads.

2- Sensorless Commutation:

- Mostly uses what called zero-crossing detection method (ZCD) to detect the back-emf when the unpowered phase cross zero volts (i.e. matches the midpoint voltage of the DC bus).
- Once its detected, a delay usually of 30 ° electrical degrees are applied before the next commutation to optimally drive the motor [33].

This commutation can be better explained in the Fig (I.8):

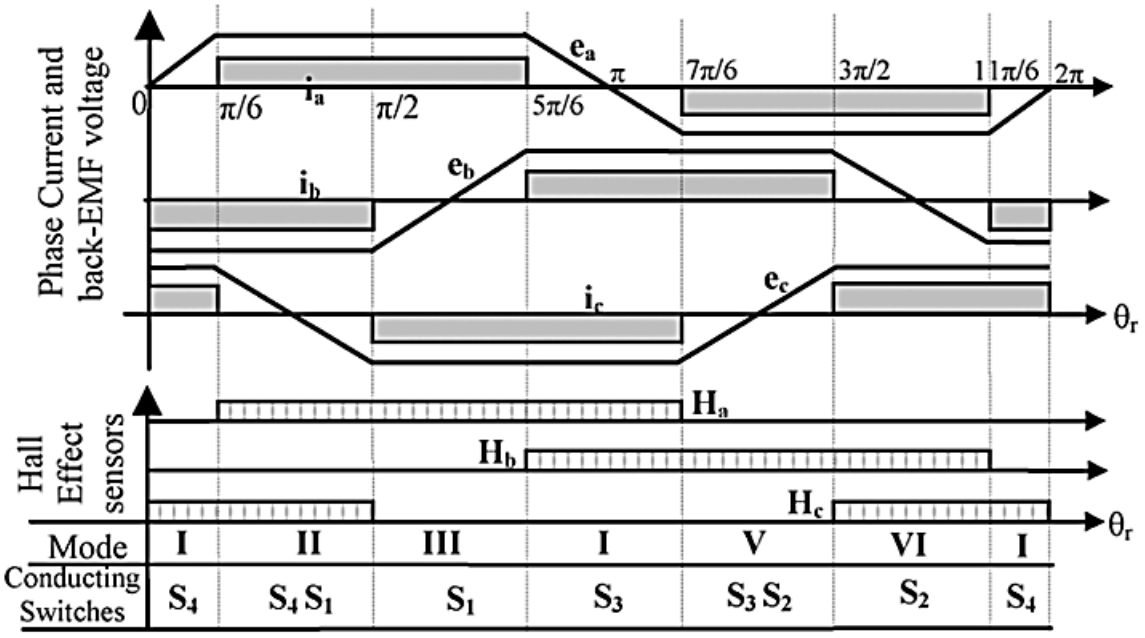


Fig (I.8): Six step commutation based on the two types of sensor usage for the BLDC rotor detection [26].

I.5 Operating principle of brushless DC motor

The operation of a Brushless DC (BLDC) motor involves special components and a sequence of coordinated steps that enable precise and efficient rotation as it is shown in the Fig (I.9).

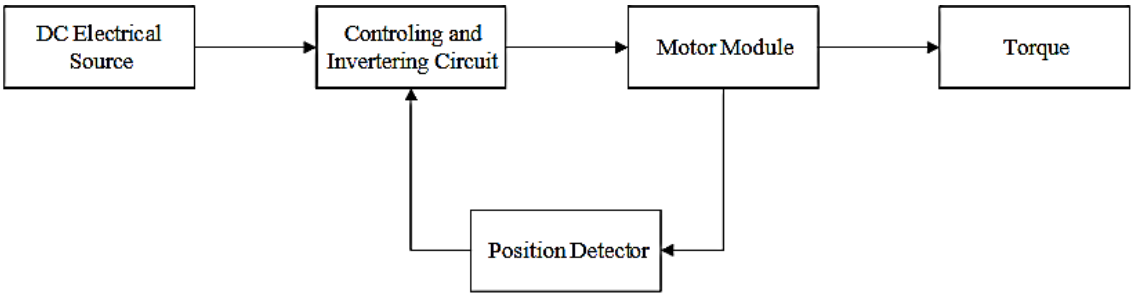


Fig (I.9): BLDC motor operating system [34].

Here is a step-by-step explanation of how a BLDC motor works:

Step 1: DC electrical power supply and electronic speed controller ESC integration [34]:

- A stable power supply that provides the necessary voltage and current.
- An ESC to convert the DC voltage input to a three phase AC voltage output to drive the motor.

Step 2: Stator and rotor configuration [13]:

- The stator three phase windings arranged in delta or star configuration; we energize two of them each time producing by that a rotating magnetic field.
- The rotor is made from strong permanent magnets that will synchronize with the rotating magnetic field generated by the stator rustling by that a torque production.

Step 3: Electronic commutation [35]:

- The ESC switches the currents in the stator windings based on the rotor position detection feedback.
- This method reduces maintenance problems, improve efficiency and guarantee smooth operating.

Step 4: Rotor position detection:

- It can be achieved by using Hall effect sensors or by detecting the back emf.
- By knowing where is the rotor its will be easier to commutate the BLDC motor.

Step 5: Speed and torque control:

- The ESC controls the speed and torque outputs of the motor simply by adjusting the timing, width and magnitude of the current's signals suppling the stator windings.

I.6 The advantages and disadvantages of BLDC motors

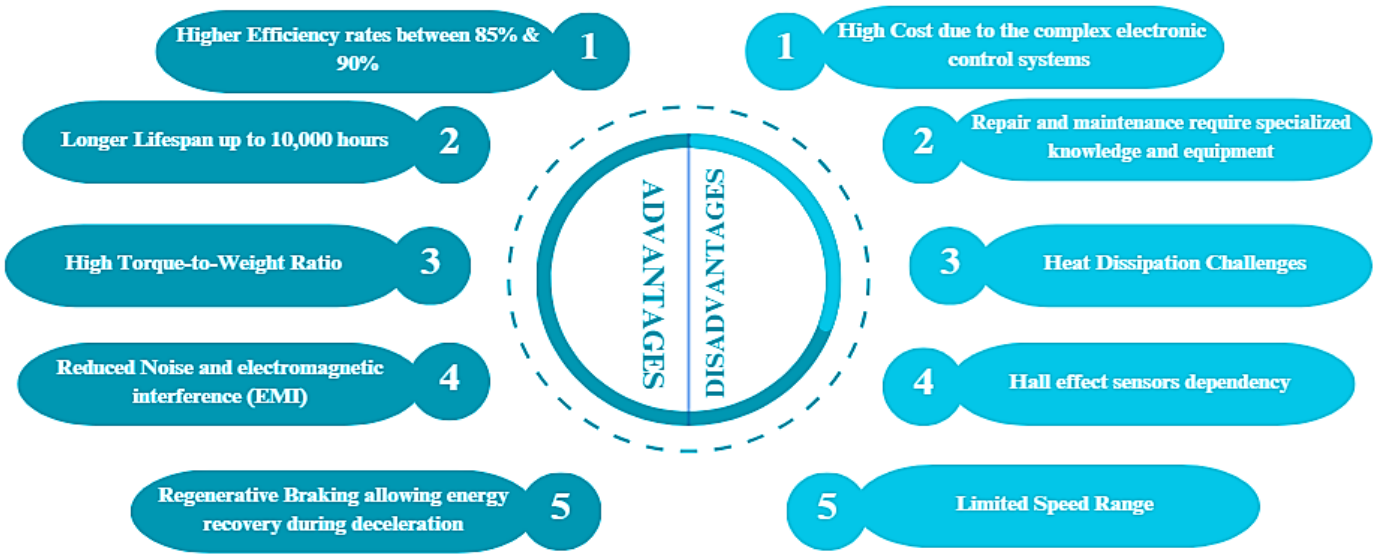


Fig (I.10): The advantages and disadvantages chart of a BLDC motor.

I.7 Applications of BLDC motors

The chart depicted in Fig (I.11) presents a comprehensive classification of BLDC motor applications across a diverse range of fields.

Fig (I.11): The applications of a BLDC motor for different fields [24].

I.8 Modeling and control of BLDC motor

I.8.1 Mathematical modeling of Brushless DC motor

The Fig (I.12) presents the three-phase equivalent circuit of the BLDC motor, from which we derived the electrical equations, torque equations, electrical angle, rotor speed, and the trapezoidal waveform function expressions.

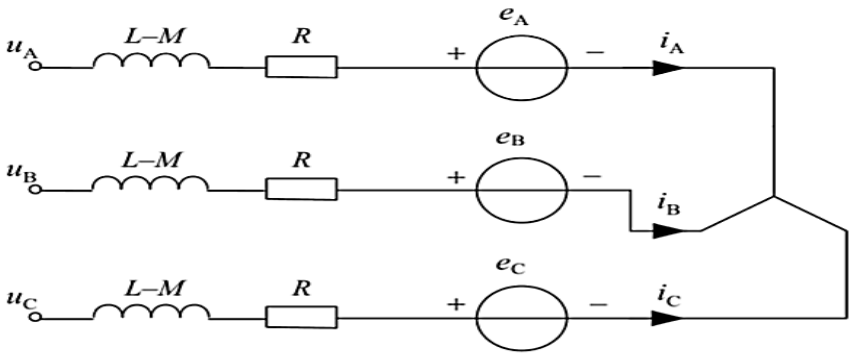


Fig (I.12): Equivalent circuit of a BLDC motor [13].

I.8.1.1 Voltage equations

- Voltage equations for each phase respectively of BLDC motor V_a , V_b, and V_c given in Volts[V]:

$$V_a = Ri_a + (L - M)\frac{di_a}{dt} + e_a \quad \text{(I.1)}$$

$$V_b = Ri_b + (L - M)\frac{di_b}{dt} + e_b \quad \text{(I.2)}$$

$$V_c = Ri_c + (L - M)\frac{di_c}{dt} + e_c \quad \text{(I.3)}$$

- Line voltage equations respectively of BLDC motor V_{ab}, V_{bc}, and V_{ca} given in Volts[V]:

$$V_{ab} = R(i_a - i_b) + (L - M)\frac{d}{dt}(i_a - i_b) + (e_a - e_b) \quad \text{(I.4)}$$

$$V_{bc} = R(i_b - i_c) + (L - M)\frac{d}{dt}(i_b - i_c) + (e_b - e_c) \quad \text{(I.5)}$$

$$V_{ca} = R(i_c - i_a) + (L - M)\frac{d}{dt}(i_c - i_a) + (e_c - e_b) \quad \text{(I.6)}$$

- Back Electromotive Force (back emf) equations for each phase of BLDC motor e_a, e_b, and e_c also given in Volts [V]:

$$e_a = K_e\, \omega_m F(\theta_e) \tag{I.7}$$

$$e_b = K_e\, \omega_m F\left(\theta_e - \frac{2\pi}{3}\right) \tag{I.8}$$

$$e_c = K_e\, \omega_m F\left(\theta_e - \frac{4\pi}{3}\right) \tag{I.9}$$

I.8.1.2 Torque equations

- Torque equations for each phase T_a, T_b, and T_c are given in Newton-Meter [N·m]:

$$T_a = K_t\, i_a F(\theta_e) \tag{I.10}$$

$$T_b = K_t\, i_b F\left(\theta_e - \frac{2\pi}{3}\right) \tag{I.11}$$

$$T_c = K_t\, i_c F\left(\theta_e - \frac{4\pi}{3}\right) \tag{I.12}$$

- Electromagnetic torque T_e is given also in Newton-Meter [N·m]:

$$T_e = T_a + T_b + T_c \tag{I.13}$$

$$T_e = 2p\omega_m i_a = K_t i \tag{I.14}$$

$$T_e - T_L = j\frac{d\omega_m}{dt} + \beta\omega_m \tag{I.15}$$

From the previous equation (I.15) we obtain the next one:

$$T_e = j\frac{d\omega_m}{dt} + \beta\omega_m + T_L \tag{I.16}$$

I.8.1.3 The electrical angle θ_e equation

$$\theta_e = \frac{p}{2}\theta_m \tag{I.17}$$

I.8.1.4 The Rotor speed ω_m equation

$$\omega_m = \frac{d\theta_m}{dt} \tag{I.18}$$

I.8.1.5 Trapezoidal form function $F(\theta_e)$

$$F(\theta_e) = \begin{cases} 1 & 0 \ll \theta_e \ll \frac{2\pi}{3} \\ 1 - \frac{6}{\pi}(\theta_e - \frac{2\pi}{3}) & \frac{2\pi}{3} \ll \theta_e \ll \pi \\ -1 & \pi \ll \theta_e \ll \frac{5\pi}{3} \\ -1 + \frac{6}{\pi}\left(\theta_e - \frac{5\pi}{3}\right) & \frac{5\pi}{3} \ll \theta_e \ll 2\pi \end{cases} \tag{I.19}$$

I.8.1.6 State-space form representation

- For a convenient performance, the voltage equations and the torque equation (I.4), (I.5), (I.6), (I.16) respectively, must be written in a state-space form, and must be modified to allow a state-apace representation.
- Since each voltage equation is a linear combination of the other two voltage equations are needed, by removing one equation and eliminating one variable using the current relationship as it shown below:

$$i_a + i_b + i_c = 0 \tag{I.20}$$

So, the new voltage equations must be:

$$V_{ab} = R(i_a - i_b) + (L - M)\frac{d}{dt}(i_a - i_b) + (e_a - e_b) \tag{I.21}$$

$$V_{bc} = R(i_a - 2i_b) + (L - M)\frac{d}{dt}(i_a - 2i_b) + (e_b - e_c) \tag{I.22}$$

- The complete state model is then:

$$\begin{pmatrix} i_a' \\ i_b' \\ \omega_m' \\ \theta_m' \end{pmatrix} = \begin{pmatrix} -R/(L-M) & 0 & 0 & 0 \\ 0 & -R/(L-M) & 0 & 0 \\ 0 & 0 & k\beta/j & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} i_a \\ i_b \\ \omega_m \\ \theta_m \end{pmatrix} + \begin{pmatrix} 2/3(L-M) & 1/3(L-M) & 0 \\ -1/3(L-M) & 1/3(L-M) & 0 \\ 0 & 1/j & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} V_{ab} - e_{ab} \\ V_{bc} - e_{bc} \\ T_e - T_L \\ 0 \end{pmatrix} \quad \text{(I.23)}$$

With:

$$\begin{pmatrix} i_a \\ i_b \\ i_c \\ \omega_m \\ \theta_m \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} i_a \\ i_b \\ \omega_m \\ \theta_m \end{pmatrix} \quad \text{(I.24)}$$

The rest of the symbols will be donated bellow:

i_a, i_b, i_c = each phase currents respectively [A].

$R_a = R_b = R_c = R_s = R$ = Stator resistance [Ω].

$L_a = L_b = L_c = L_s = L$ = Self-inductance [H].

M = mutual inductance [H].

j= rotor inertia [Kg.m^2].

β= friction constante.

K_t = torque constant.

k_e = back emf constant.

θ_m = rotor angular displacement [Radian or Degrees].

ω_m= Rotor speed[rad/sec].

T_L =the load torque [N·m].

p = pole number.

$F(\theta_e)$ = the function that gives the trapezoidal waveform of the back-emf.

I.8.2 Brushless DC motor transfer function

The transfer function is a convenient mathematical way to represent a linear, time-invariant system in term of the relationship between the input and output in the Laplace domain, assuming zero initial conditions. The transfer function simplifies differential equations into mathematical ones [36].

When a three-phase brushless DC motor in the Fig (I.12) operates under full-bridge drive control using two-phase conduction mode, specific electromagnetic conditions arise during each commutation phase. During the excitation of phases ***A*** and ***B*** windings (while phase ***C*** remains inactive), we obtain the following operational characteristics:

$$i_a = -i_b = i \tag{I.25}$$

$$\frac{di_a}{dt} = -\frac{di_b}{dt} = \frac{di}{dt} \tag{I.26}$$

Therefore, we can write the line voltage equation V_{ab} (I.4):

$$V_{ab} = 2Ri + 2(L - M)\frac{di}{dt} + (e_a - e_b) \tag{I.27}$$

Thus,

$$e_a = e_b \tag{I.28}$$

So, equation (I.28) can be expressed as:

$$V_{ab} = U_d = 2Ri + 2(L - M)\frac{di}{dt} + 2e_a = r_a i + L_a\frac{di}{dt} + K_e\omega_m \tag{I.29}$$

As it is shown in the BLDC motor equivalent circuit with two phase windings exited in the Fig (I.13), Where:

U_d = DC bus voltage [v].

$r_a = 2R$ = line resistance of windings [Ω].

$L_a = 2(L - M)$ = equivalent line inductance of windings [H].

i = line current [A].

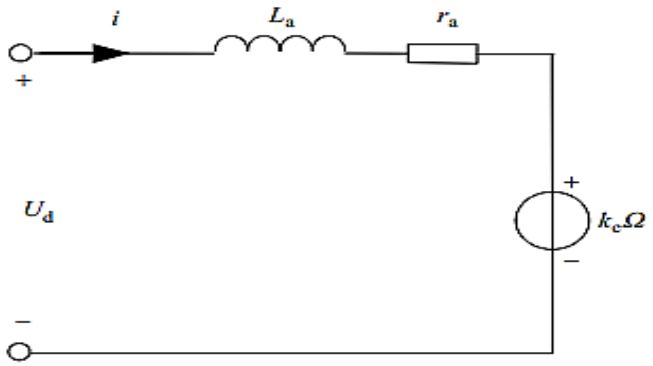


Fig (I.13): Equivalent circuit of the BLDC motor with two phase windings excited [13].

So, by replacing the equation (I.15) in (I.16) we obtain:

$$K_t i - T_L = j\frac{d\omega_m}{dt} + \beta\omega_m \quad \text{(I.30)}$$

The current equation below is given when the BLDC motor runs with no load:

$$i = \frac{j}{K_t}\frac{d\omega_m}{dt} + \frac{\beta}{K_t}\omega_m \quad \text{(I.31)}$$

By exchanging the equation (I.31) in (I.29) we obtain:

$$U_d = r_a\left(\frac{j}{K_t}\frac{d\omega_m}{dt} + \frac{\beta}{K_t}\omega_m\right) + L_a\frac{d}{dt}\left(\frac{j}{K_t}\frac{d\omega_m}{dt} + \frac{\beta}{K_t}\omega_m\right) + K_e\omega_m \quad \text{(I.32)}$$

Using Laplace transformation of equation (I.32) directly yields the transfer function representation of the BLDC motor we obtain:

$$G_U(s) = \frac{\omega_m(s)}{U_d(s)} = \frac{K_t}{L_a j s^2 + (r_a j + L_a\beta)s + (r_a\beta + K_e K_t)} \quad \text{(I.33)}$$

Thus, results in that the BLDC motor can be seen as a second-order system, so the equation (I.33) can be reformatted as:

$$G_U(s) = \frac{K_t\omega_n^2}{r_a\beta+K_eK_t(s^2+2\xi\omega_n+\omega_n^2)} \quad \text{(I.34)}$$

Where:

$\omega_n = \sqrt{\frac{r_a\beta+K_eK_t}{L_aj}}$ = Natural frequency of the second-order system. (I.35)

$\xi = \frac{1}{2}\frac{r_aj+L_a\beta}{\sqrt{L_aj}\sqrt{r_a\beta+K_eK_e}}$ = Damping ratio of the second-order system. (I.36)

An open-loop regulation in BLDC motors, derived from system transfer functions (I.33), operates without feedback and relies on predefined command signals, offering simplicity and cost-effectiveness. However, this method exhibits suboptimal performance under dynamic operating conditions, including poor load regulation, limited precision, and inefficiency at low speeds, as empirically validated in Fig (I.14):

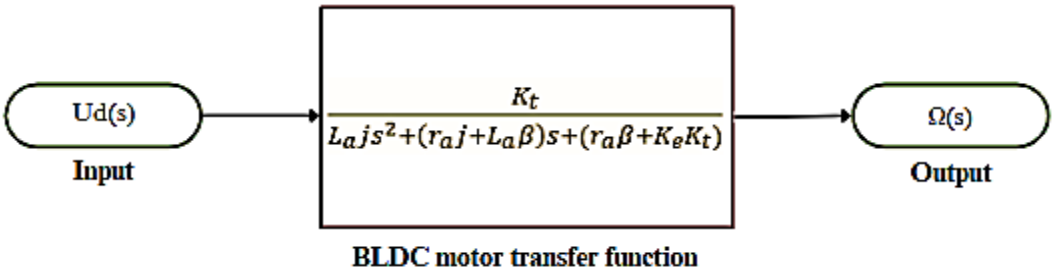


Fig (I.14): Block diagram of BLDC motor open-loop regulation.

I.8.2.1 Controle and regulation

- Closed loop regulation: it is a crucial control method for BLDC motors it relies on the real-time feedback mechanism (i.e. the Hall effect sensors ,decoder, or back emf detection provides us with real time data on rotor position ,speed and currents and also the ESC possesses the feedback and compares it to the wanted setpoint speed and adjusts the PMW signals to control the motor speed as so),this method is known of being efficient, precise ,adaptable and robust .
- PID regulator: a proportional(P)-integral(I)-derivative(D) regulator is a closed loop control system used to regulate processes by minimizing the error between the setpoint (e.g. speed in our case) and the measured real variable value, the PID regulator contains three essential control actions [38]:

- Proportional(P): it adapts the error output proportionally upgrading responsiveness, but can cause oscillations if the K_p is too high, where:

$$P = K_p * e(t) \quad \text{(I.37)}$$

- Integral (I):it eliminates the steady-state error by integrating past error over time, but the excessive K_i can augment overshot, where:

$$I = K_i * \int_0^t e(t)dt \quad \text{(I.38)}$$

- Derivative(D):it enhances the stability, reduces settling time and most important is that it predicts future errors by using the error rate of change, where:

$$D = K_d * \frac{de(t)}{dt} \quad \text{(I.39)}$$

The total PID controller is the sum of Equations (I.37), (I.38), (I.39):

$$PID = K_p * e(t) + K_i * \int_0^t e(t)dt + K_d * \frac{de(t)}{dt} \quad \text{(I.40)}$$

In the Fig (I.15), there is a BLDC PID closed-loop condole block diagram:

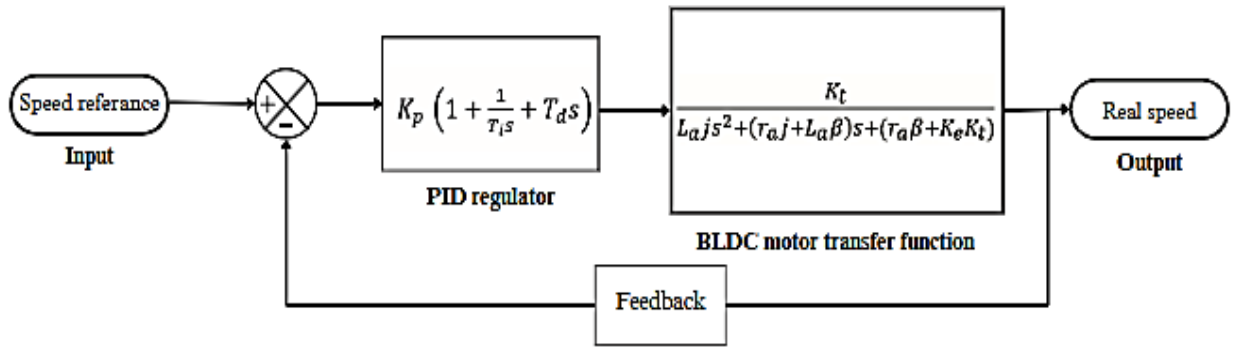


Fig (I.15): BLDC PID closed-loop control bloc diagram.

I.8.2.2 PID parameter identification methods

The determination of PID controller parameters (Kp,Ki, and Kd) is necessary for effective system regulation. This study utilizes two methods for parameter identification: the analytical *second methos of Ziegler-Nichol's* technique and the *System Identification* tool in MATLAB/APPS.

- **Ziegler-Nichol's method**

Introduced in 1942, the Ziegler-Nichols methods are foundational empirical techniques for tuning proportional-integral-derivative (PID) controllers. Renowned for their simplicity and reliability. The first and the second methods are the two principal approaches of Ziegler-Nichol's technique, they are extensively utilized in industrial control applications and serve as reference standards for contemporary tuning strategies. enables the determination of PID parameters (Kp,Ki, and Kd) based on the dynamic characteristics of the process [42].

In the Ziegler-Nichols second method, the controller is initially set to proportional mode only with the integral and derivative actions disabled ($Ki = 0$, $Kd = 0$) according to the Fig (I.16) and the Table (I.1) down below [43][44]:

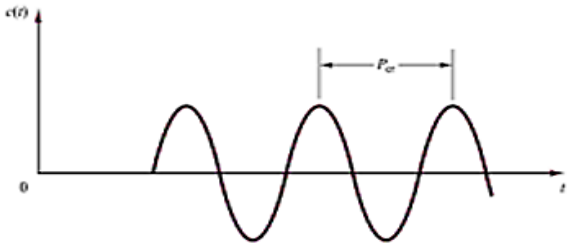


Fig (I.16): Persistent oscillatory response characterized by the period P_{Cr} [42.]

Table (I.1): Ziegler-Nichols's (second method) PID tuning rule based on critical gain $K_{Cr}$and critical period P_{Cr}.

Type of controller	$\boldsymbol{Kp}$	$\boldsymbol{Ki}$	$\boldsymbol{Kd}$
P	$0.5\ K_{Cr}$	∞	0
PI	$0.45\ K_{Cr}$	$\frac{1}{1.2}\ P_{Cr}$	0
PID	$0.6K_{Cr}$	$0.5\ P_{Cr}$	$0.12\ P_{Cr}$

The PID controller values were determined using the Ziegler-Nichols closed-loop method according to the Table (I.1), where:

$$PID - ZN = 51.36 * e(t) + 2153.45 * \int_0^t e(t)dt + 0.306 * \frac{de(t)}{dt} \quad \text{(I.41)}$$

- **System Identification method:**

Before applying this method, two key conditions must be satisfied: the system must be set to open-loop operation, and it should function without any external load. The procedure is outlined in the following steps:

Step 1: define where to put the input and the output data in the system, because the system identification app enables us to identify from a measured input-output data the models of dynamic systems [39], as it shown in Fig (I.17) (see Annex A).

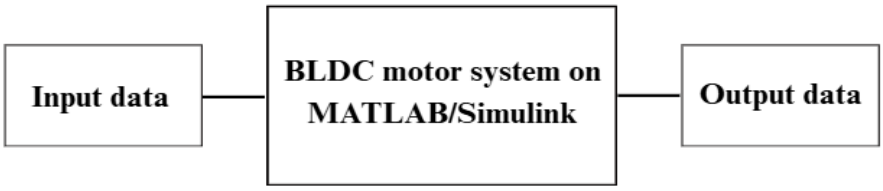


Fig (I.17): BLDC MATLAB/APPS/ System Identification block diagram explanation.

Step 2: open to MATLAB/APPS/System identification and follow the steps in the next Fig (I.18), (I.19), (I.20)

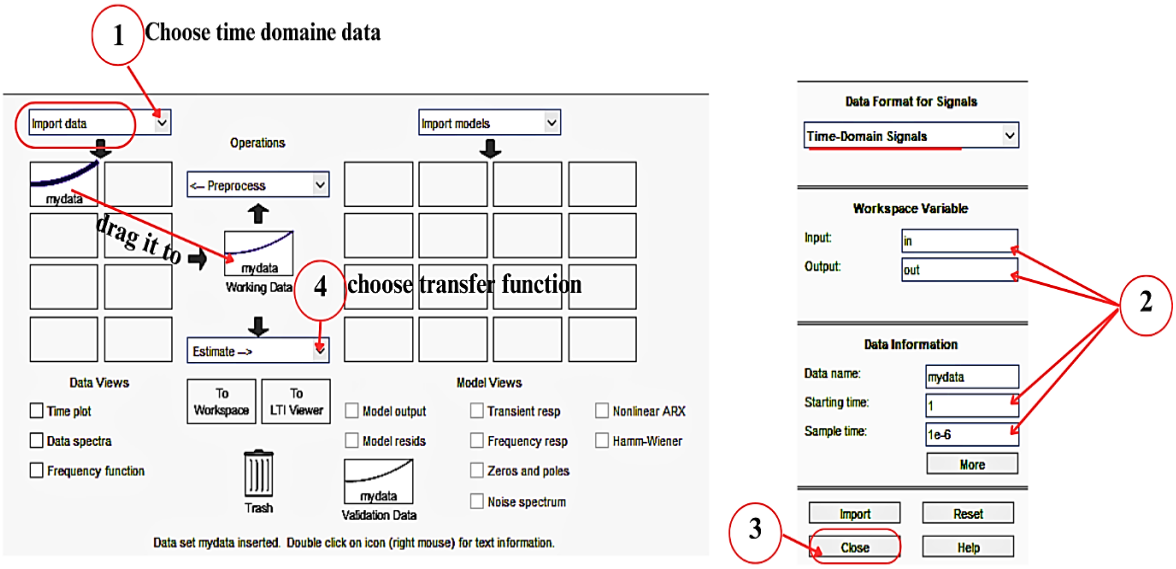


Fig (I.18): System transfer function identification step 1.

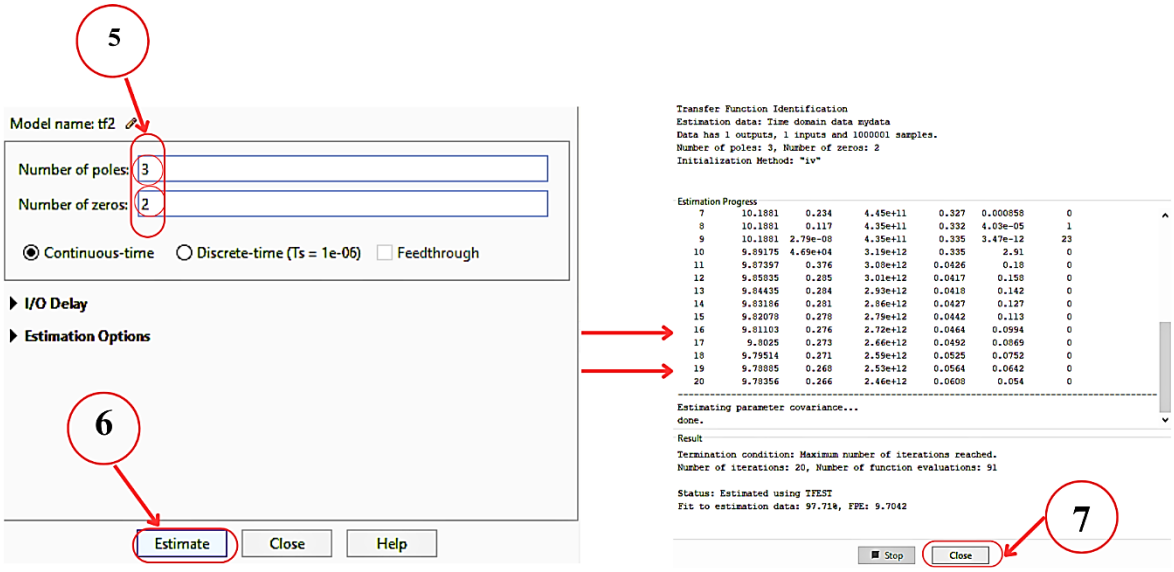


Fig (I.19): System transfer function identification step 2.

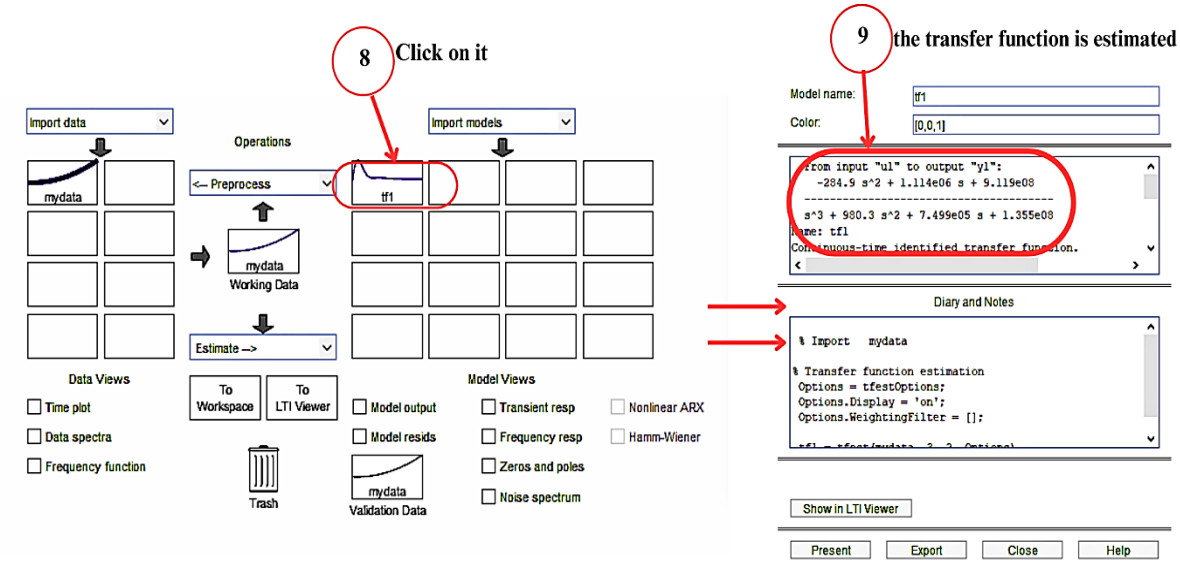


Fig (I.20): System transfer function identification step 3.

It is established in control theory that the overall system transfer function is obtained by multiplying the transfer function of the BLDC motor with that of the PID controller, as illustrated in Fig (I.15). This results in a third-order system, which justifies selecting three poles and two zeros (see Fig (I.19)). This approach enables the estimation of the complete system transfer function and facilitates the determination of the appropriate PID parameters respectively, as demonstrated in Fig (I.20).

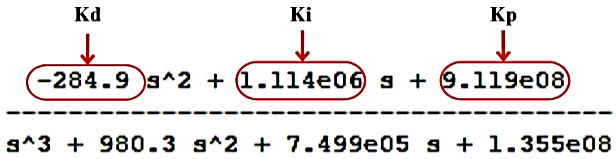


Fig (I.21): Estimated BLDC system transfer function with the PID parameters.

After injecting the PID controller parameters derived from the estimated system transfer function to our system, the following results were obtained.

$$PID - APPS = 9.119e08 * e(t) + 1.114e06 * \int_0^t e(t)dt - 284.9 * \frac{de(t)}{dt} \quad \text{(I.42)}$$

I.9.2 Results & simulation

I.9.2.1 Uncontrolled simulation (open-loop)

Open-loop simulations of the BLDC motor system were conducted in MATLAB/Simulink model (see Annex B) under no-load and loaded conditions. The tests, parameterized as detailed aimed to analyze dynamic response and validate system performance across operational scenarios.

utilizing: voltage supply $V_{DC} = 300V$

- Test 1: With no load ($TL = 0\ N.m$)

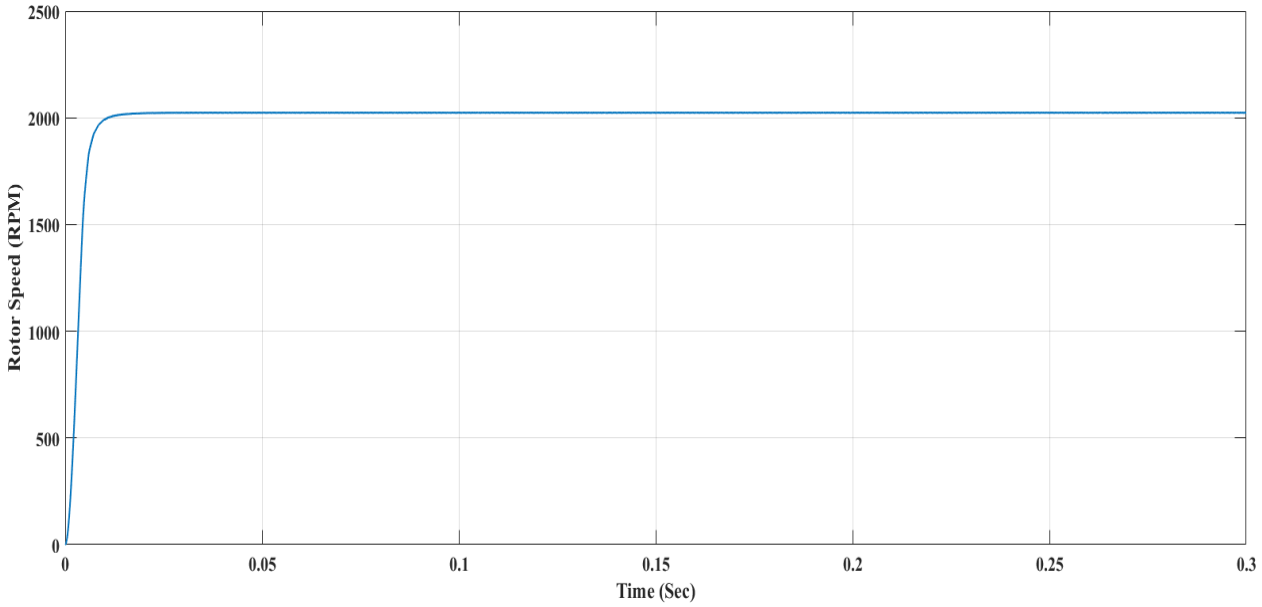


Fig (I.22): Rotor speed response with no load.

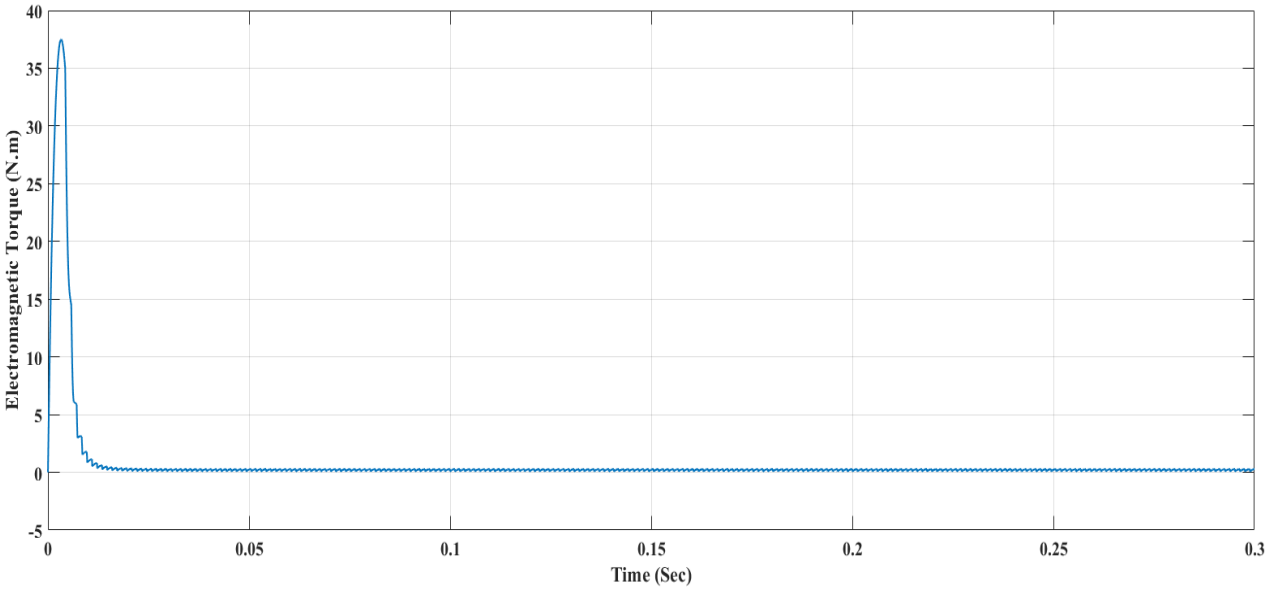


Fig (I.23): Electromagnetic Torque response with no load.

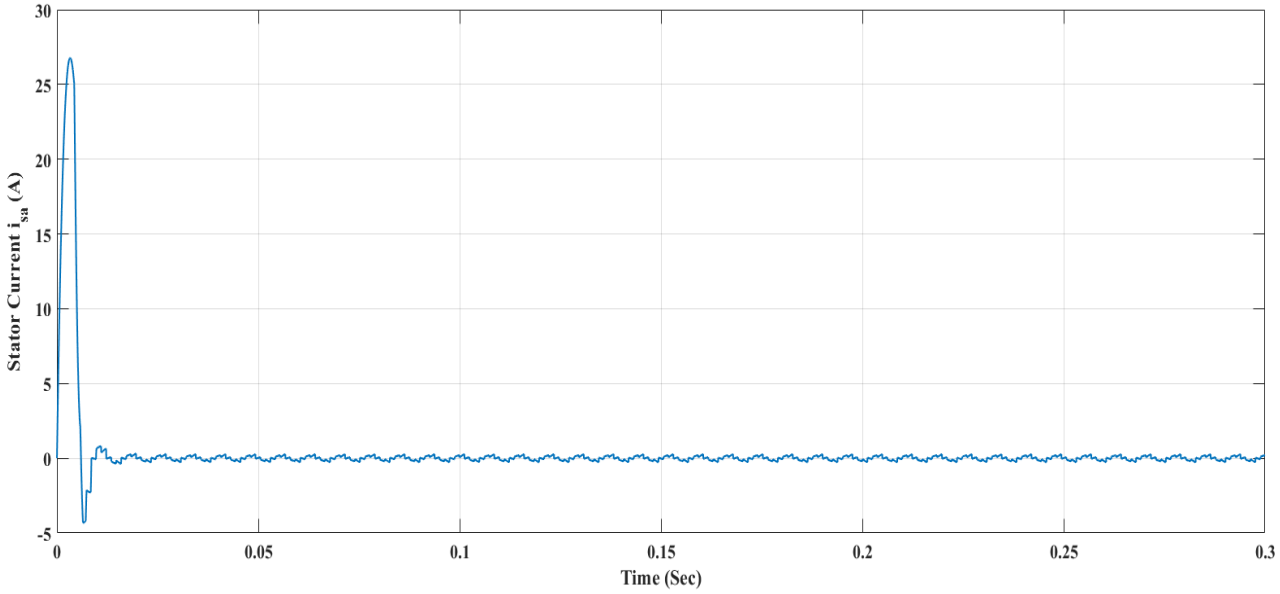


Fig (I.24): Stator current response with no load.

- Test 2: with load: where $TL = 5(N.m)$ added at $t = 0.15s$

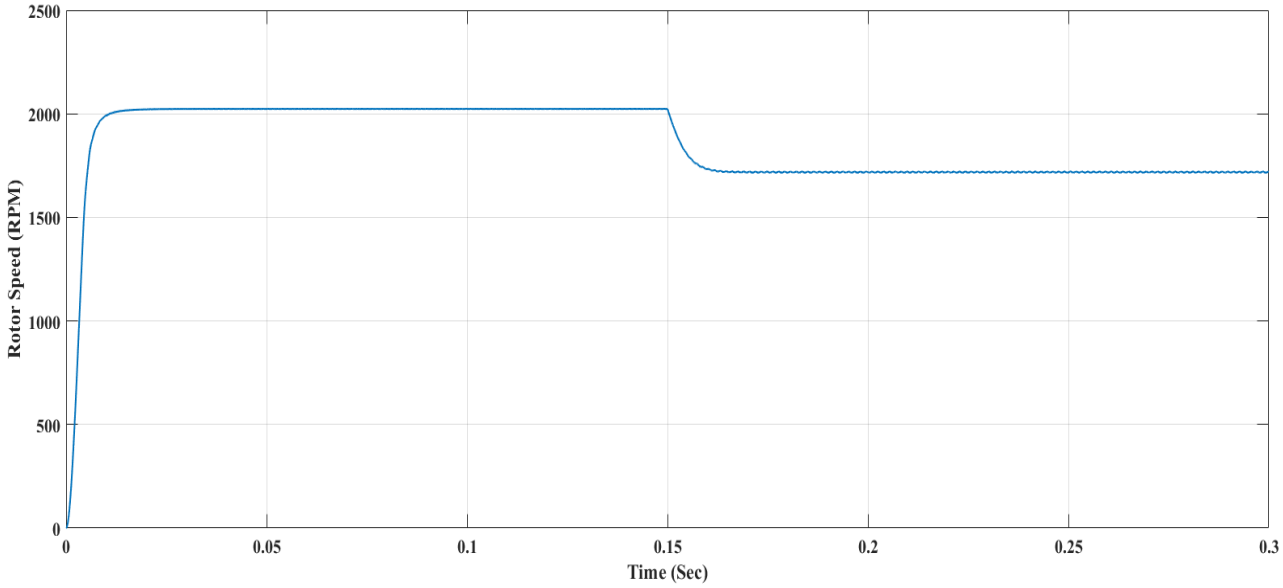


Fig (I.25): Rotor Speed response with load.

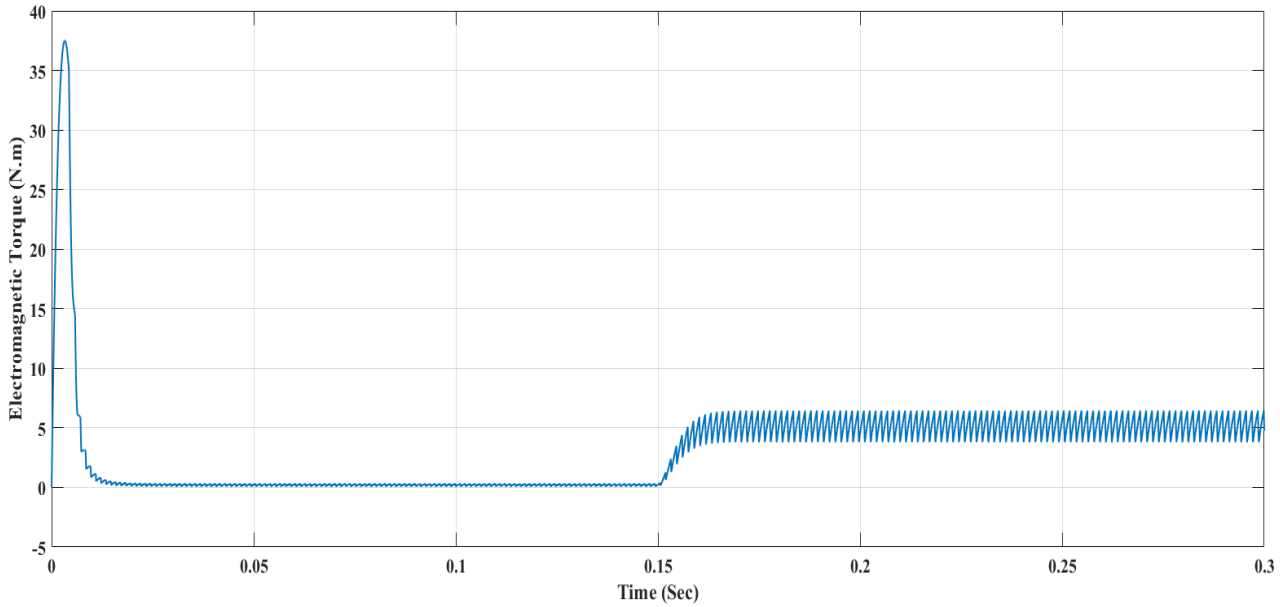


Fig (I.26): Electromagnetic Torque response with load.

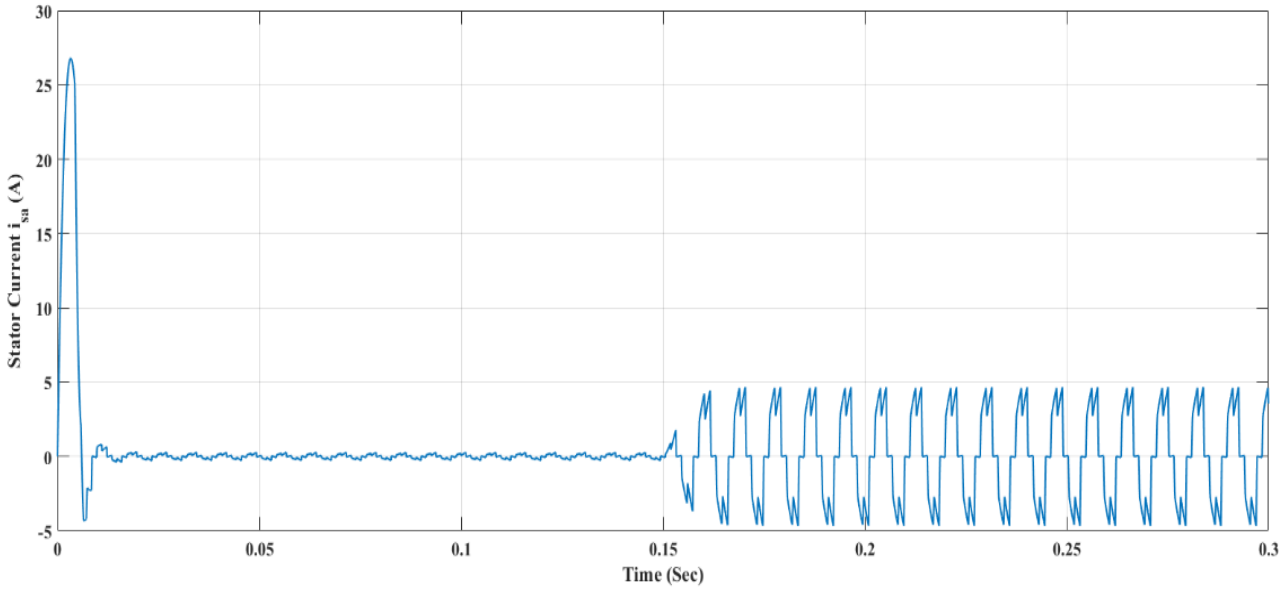


Fig (I.27): Stator current response with load.

I.9.2.2 Controlled simulation (Closed-loop)

A simulation model of a tuned PID-controlled BLDC motor system was developed in MATLAB/Simulink (see Annex C), three operational scenarios: no-load, loaded, and speed inversion, were simulated to validate the tuning methodology and evaluate dynamic response characteristics. Under the following parameters

Voltage supply $V_{DC} = 300V$,speed set point is $1000\ RPM$, $T_L = 5$(N.m)

- Test 1: with no load ($T_L = 0$(N.m))

  Over Speed: The motor executed speed step change was applied, increasing the rotational velocity from $1000\ RPM$ to $1500\ RPM$ at $t = 0.15s$ .This transition enabled the assessment of the system's dynamic response to a sudden increase in setpoint speed.

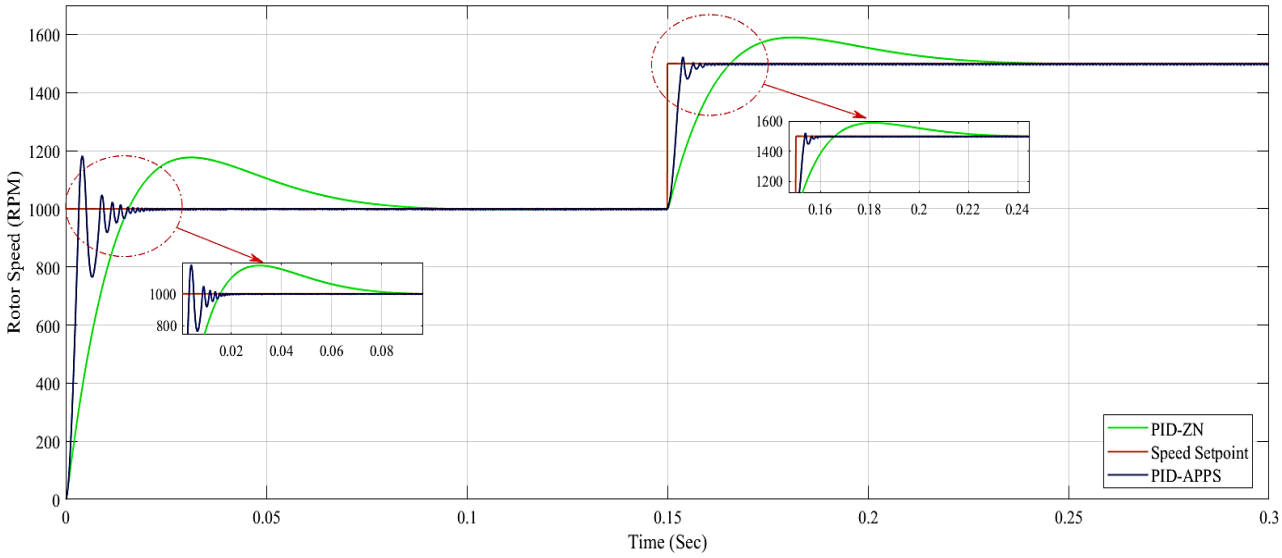


Fig (I.28): Rotor Speed response with no load.

- Rotor speed inversion: The motor executed a directional speed reversal from $+1000\ RPM$ to $-1000\ RPM$ at $t = 0.15s$ .This transition enabled the assessment of the system's dynamic response to a sudden decrease in setpoint speed.

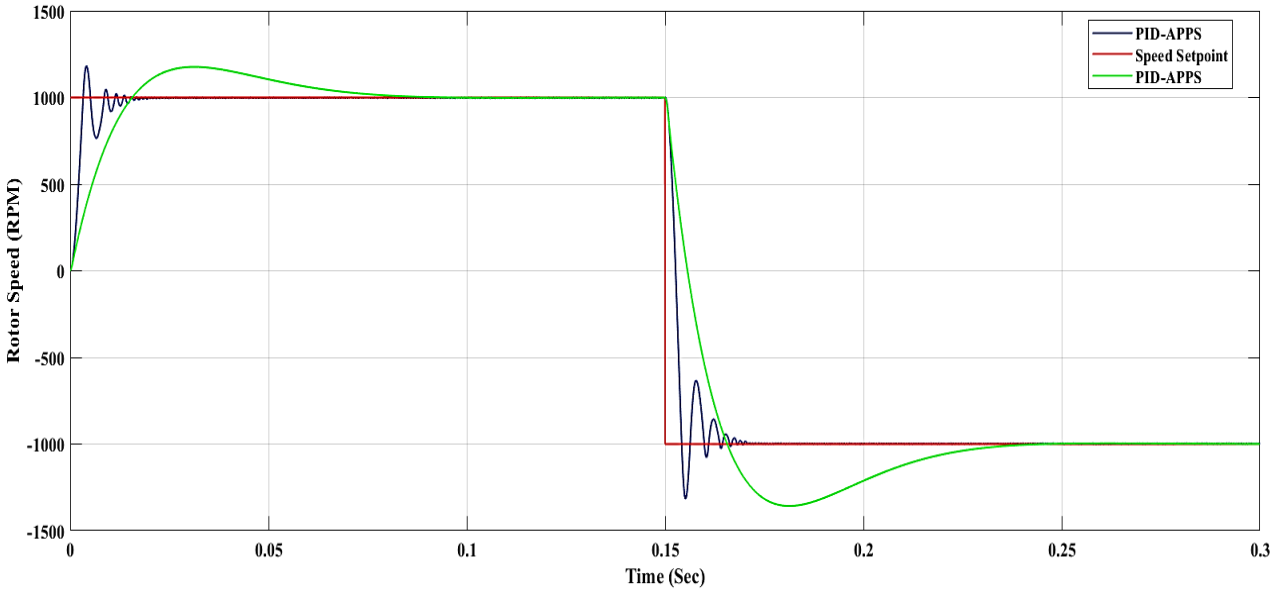


Fig (I.29): Rotor Speed inversion response with no load.

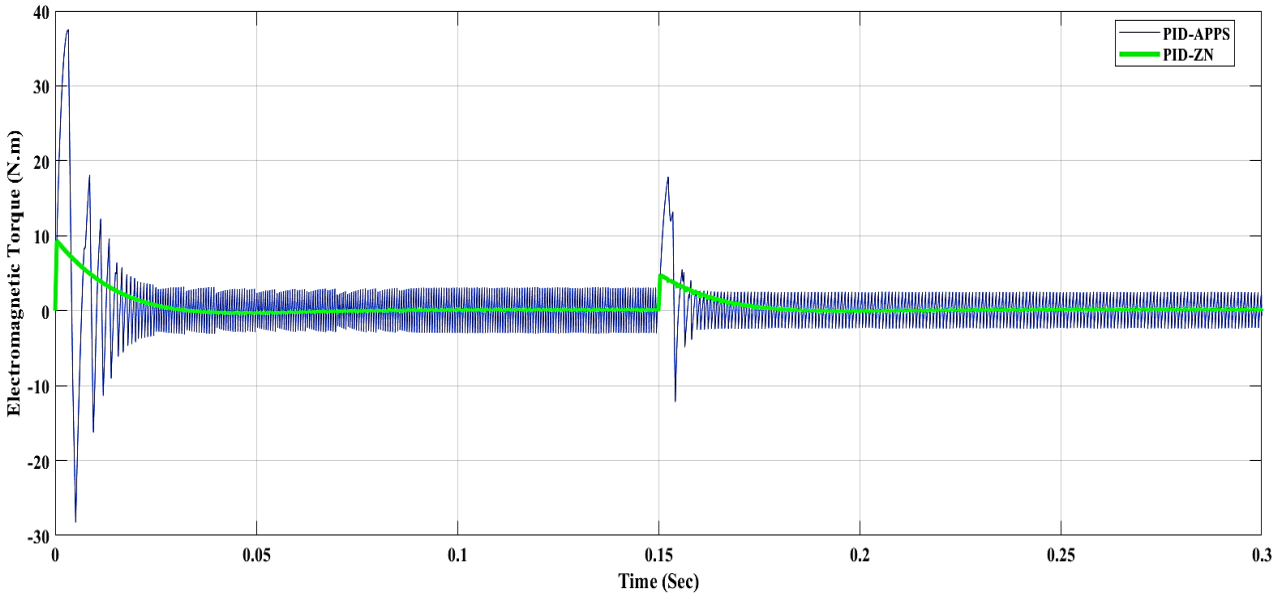


Fig (I.30): Electromagnetic torque response with no load.

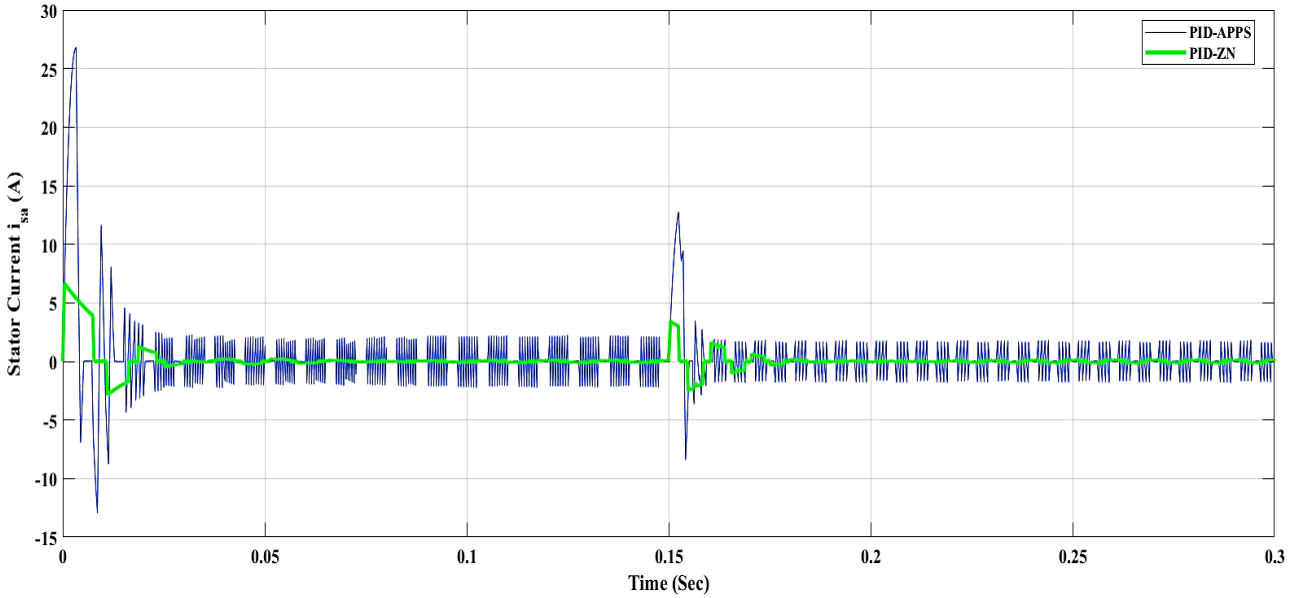


Fig (I.31): Stator current response with no load.

➢ Test 2: with load: where $T_L = 5$(N.m) added at $t = 0.15$(s), speed reference is $1000RPM$

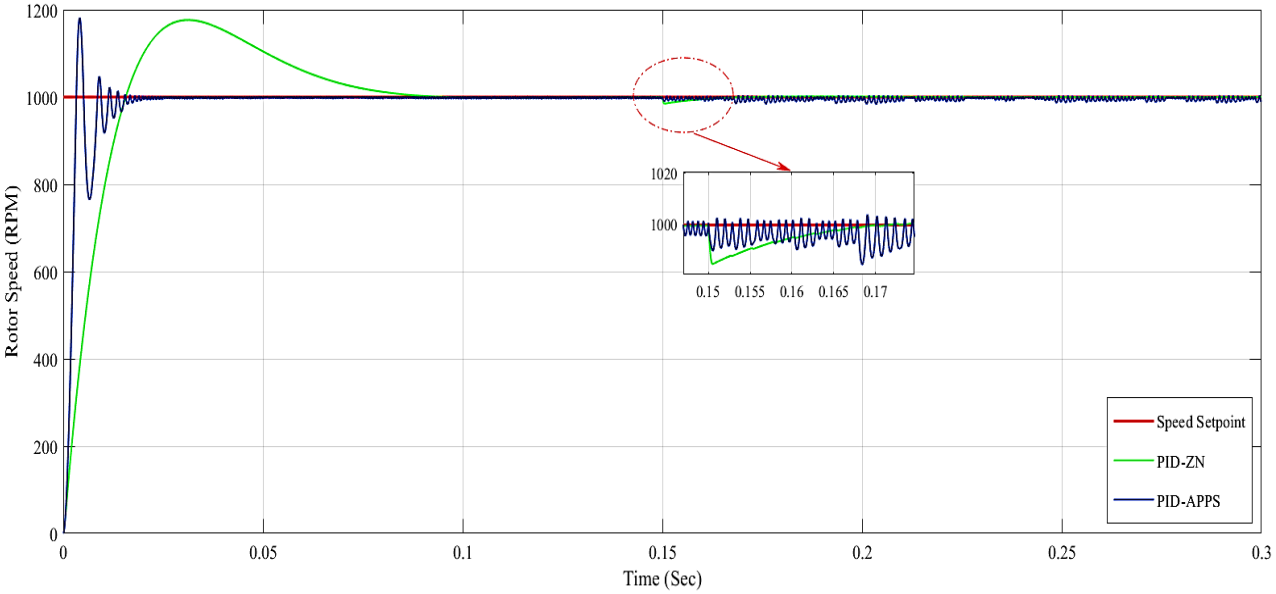


Fig (I.32): Rotor Speed response with load.

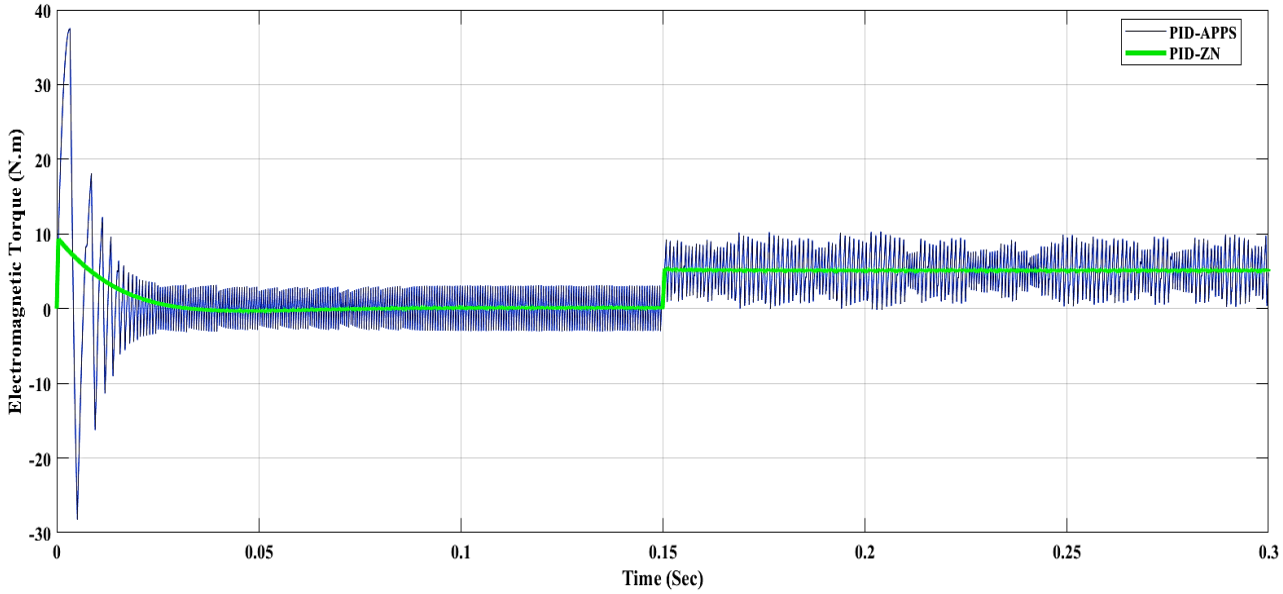


Fig (I.33): Electromagnetic torque response with load.

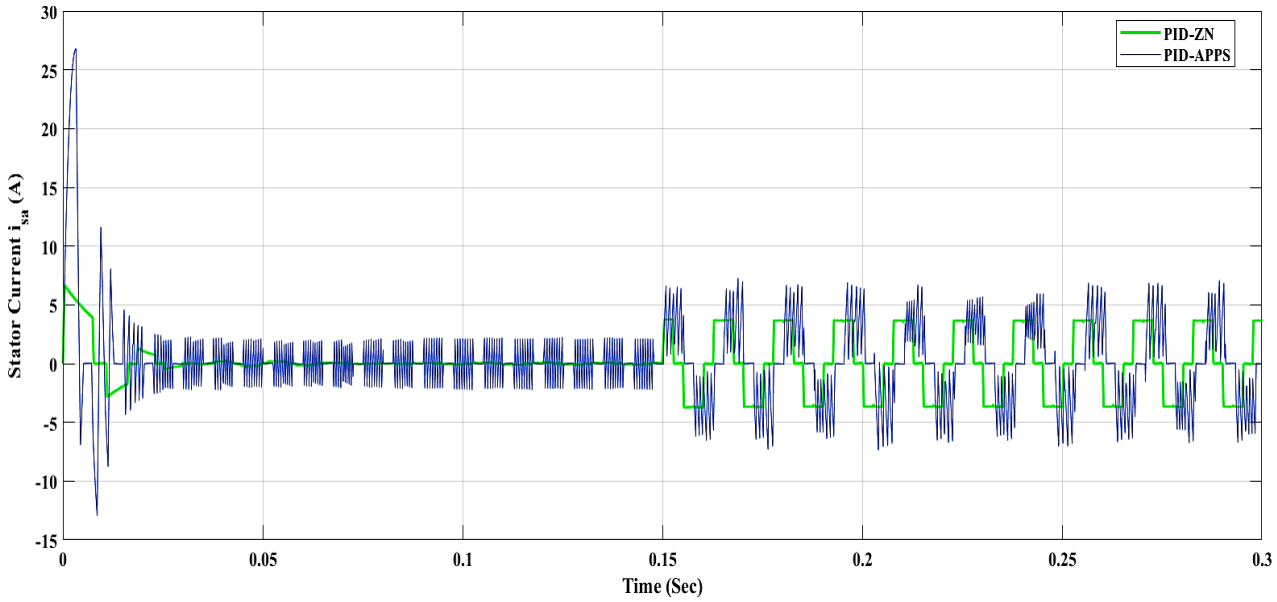


Fig (I.34): Stator current response with load.

I.9.3 Interpretation and comments

➢ Test one: with no load

- Figure (I.22) illustrates the open-loop rotor speed response of a BLDC motor under no-load conditions, demonstrating steady-state operation at approximately $2024\ RPM$. The motor accelerates rapidly from $0\ RPM$ and reaches steady-state, indicating a fast dynamic response. The smooth curve, with no overshoot or oscillations, reflects a well-damped and stable system.
- Figure (I.23) illustrates the open-loop electromagnetic torque response of a BLDC motor under no-load conditions. At startup, the torque exhibits a sharp transient peak of $37.42\ N \cdot m$ to overcome rotor inertia, followed by a rapid decline and stabilization around $0\ N \cdot m$ within a few milliseconds. This behavior is consistent with BLDC motor dynamics and reflects efficient startup performance.
- Figure (I.24) illustrates the open-loop stator current response for the first phase (i_{sa}) of a BLDC motor under no-load conditions. The current exhibits a transient peak of $26.72\ A$ at startup due to inrush, followed by a rapid decay into a steady-state oscillatory regime centered near zero.

➢ Test 2: with load

- Figure (I.25) illustrates the open-loop rotor speed response of a BLDC motor when subjected to a load torque applied at 0.15 s. Initially, the motor accelerates and stabilizes at approximately $2024\ RPM$. Upon load application, the speed drops to $1719\ RPM$ and stabilizes at this lower value, as dictated by the increased torque demand. This behavior highlights the inherent limitation of open-loop systems in compensating for load changes, though the smooth transition reflects stable dynamic performance under load conditions.

- Figure (I.26) illustrates the open-loop electromagnetic torque response of a BLDC motor under load conditions. At startup, the torque peaks at $37.44\ N \cdot m$ to overcome inertia, then rapidly declines near zero as the motor reaches no-load steady-state. When a load torque of $5\ N \cdot m$ is applied at $t = 0.15s$, the electromagnetic torque rises accordingly and stabilizes around this value, accompanied by small oscillations. these oscillations reflect inherent torque ripple caused by commutation effects and non-ideal flux distribution, demonstrating the motor's ability to match load demand while maintaining system stability.

- Figure (I.27) illustrates the open-loop stator current response of a BLDC motor for the first phase (i_{sa}) under load conditions. Initially, the current exhibits a transient peak of $26.74\ A$, then stabilizes near zero in a low-amplitude oscillatory steady-state. Upon load application at $t = 0.15s$, the stator current shows a significant increase in both amplitude and fluctuation, reflecting the motor's response to the added torque demand.

➢ Test one: with no load

- Figure (I.28) illustrates the closed-loop rotor speed response of a BLDC motor under no-load conditions, comparing two control strategies: PID-ZN (Ziegler–Nichols) and PID-APPS. Initially, the rotor accelerates to a setpoint of approximately $1000\ RPM$, followed by a rise to $1500\ RPM$ at $t = 0.15s$. Both transitions exhibit minor overshoot and brief oscillations. The PID-ZN controller achieves a faster rise time but displays significant overshoot and slower settling with sustained oscillations. In contrast, the PID-APPS controller provides a more stable and accurate response with minimal overshoot and quicker damping. This comparison highlights the effectiveness of closed-loop control in achieving fast setpoint tracking and the superior performance of PID-ZN in delivering smoother, more stable rotor speed regulation.

- Figure (I.29) illustrates the closed-loop rotor speed response of a BLDC motor focusing on a speed inversion scenario from $+1000\ RPM$ to $-1000\ RPM$ at $t = 0.15s$. This abrupt reversal induces transient oscillations and overshoot during both deceleration and acceleration phases. The figure also compares the performance of PID-ZN and PID-APPS controllers. While the PID-ZN controller shows a faster initial response, it suffers from higher overshoot and slower settling. In contrast, the PID-APPS controller achieves smoother transitions, reduced oscillations, and quicker stabilization. Overall, the system demonstrates effective control, with PID-ZN offering superior dynamic performance and stability during rapid speed inversion.
- Figure (I.30) illustrates the closed-loop electromagnetic torque response of a BLDC motor under no-load conditions, highlighting both transient behavior and controller performance. Initially, the torque exhibits significant oscillations, peaking, indicating an underdamped response. Around $t = 0.15\ s$, a sharp disturbance triggers another set of oscillations, which gradually diminish as the system stabilizes. The figure compares PID-ZN and PID-APPS controllers: while PID-APPS shows initially higher oscillations than PID-ZN, it stabilizes more quickly and returns to steady-state faster after the setpoint change. In contrast, PID-ZN demonstrates slower damping and less precise settling. Overall, PID-ZN provides better dynamic performance and more effective torque transient suppression and less torque ripple.
- Figure (I.31) illustrates the closed-loop stator current response of a BLDC motor under no-load conditions, highlighting both transient dynamics and control performance. Initially, the current shows high-amplitude oscillations, peaking and dipping, before settling near zero with smaller oscillations. Around $t = 0.15s$, a disturbance associated by damped oscillations returning to the steady-state pattern. The figure compares PID-ZN and PID-APPS controllers: while both experience transient currents, PID-APPS stabilizes more quickly and maintains consistent oscillations, whereas PID-ZN displays slower recovery and greater overshoot. This demonstrates PID-ZN superior dynamic response and robustness under load disturbances.

➢ Test 2: with load

- Figure (I.32) illustrates the closed-loop rotor speed response of a BLDC motor to a speed setpoint of $1000\ RPM$ under load conditions. The rotor speed rises rapidly from zero, briefly overshoots the setpoint with minor oscillations, and stabilizes near the target within approximately at $t = 0.15s$,even with a load torque of $5\ N \cdot m$ applied. The figure compares the PID-ZN and PID-APPS controllers: PID-ZN exhibits significant overshoot and slower settling both before and after the load application, while PID-APPS reaches the setpoint more smoothly and recovers quickly with minimal deviation. This highlights the control system's effectiveness and demonstrates PID-ZN superior speed regulation, stability, and robustness under dynamic load disturbances.
- Figure (I.33) presents the closed-loop electromagnetic torque response of a BLDC motor under load conditions. Initially, the system shows large but rapidly damped oscillations, indicating strong stability. At $t = 0.15\ s$, a load is applied, causing a spike in oscillation amplitude and a shift in the torque pattern. Comparing PID-ZN and PID-APPS controllers, PID-APPS exhibits faster damping and quicker torque alignment, while PID-ZN responds more conservatively with slower adjustment. Overall, PID-ZN demonstrates superior responsiveness to dynamic load changes and less torque ripple.
- Figure (I.34) shows the closed-loop stator current response of a BLDC motor under load. After an initial spike above 25 A, the current quickly damps and settles into an oscillatory pattern. At t = 0.15 s, a load disturbance increases oscillation amplitude, revealing the system's dynamic behavior. Comparing controllers, PID-APPS offers smoother, more sinusoidal currents and better transient handling, while PID-ZN results in more abrupt, step-like responses. Overall, PID-ZN demonstrates superior performance under dynamic load conditions.

I.10 Conclusion

This chapter presents an overview of brushless DC (BLDC) motors, including their historical background, structural design, operational principles, and mathematical modeling, as well as a study of open-loop and closed loop behavior using MATLAB/Simulink. It also discusses the advantages, disadvantages, and diverse applications of BLDC motors. Additionally, the chapter introduces the use of the Ziegler-Nichols second method and also MATLAB/APPS/ System Identification Tool for PID controller tuning.

The next chapter will focus on closed-loop speed regulation using PID control and further enhance performance by applying the Harris Hawks Optimization algorithm (HHO) to minimize settling time, overshoot, and steady-state error.

Chapter II

Controller of Harris Hawk’s

Chapter II: Controller of Harris Hawk's

II.1 Introduction

Classic PID controller, while widely used in industrial applications, often demonstrate limited effectiveness when applied to Brushless DC (BLDC) motors, primarily due to the motors' nonlinear and dynamic behavior. To overcome these challenges and enhance controller performance, advanced optimization techniques such as the Harris Hawks Optimization (HHO) algorithm have been introduced. The success of HHO in such applications depends significantly on the accurate adjustment of system parameters [45]. As energy efficiency and performance optimization become increasingly important, the use of intelligent algorithms has gained prominence. The Harris Hawks Optimization (HHO) algorithm, inspired by the cooperative hunting behavior of Harris's hawks, is particularly suited for complex optimization problems due to its strong balance between global exploration and local exploitation. This chapter provides an in-depth overview of the HHO algorithm, detailing its operational mechanism and its role in optimizing the control of BLDC motors. It also introduces the concept of metaheuristic optimization methods and presents a schematic representation of the HHO algorithm's structure.

II.2 Optimization

II.2.1 Definition

An optimization problem is defined as finding the minimum or maximum (optimum value) of a given function. Optimization problems can also be found in which the variables of the function to be optimized are constrained to evolve in a specific part of the search space; these are optimization equations.

$$f_{obj}(X) = \left\{ \begin{array}{c} \min f_{obj}(X) \\ subject\ to: h_k(X) = 0 \\ g_j(X) \leq 0 \end{array} \right\} \quad \text{(II.1)}$$

$f_{obj}(X)$: represents the objective function.

$g_j(X)$: an inequality constraint.

$h_k(X)$: an equality constraint function.

II.2.2 Bases of Optimization

In operations research, it is not always possible to find an exact method for solving optimization problems. In such cases, researchers first attempt to identify an equivalent problem that has already been solved. If no solution is available, they turn to heuristics simple and intuitive algorithms that provide approximate solutions based on observation and logical reasoning. These algorithms often explore multiple solutions and retain the best one. Randomness may be introduced to widen the search space, requiring several runs to approach the optimal solution [46].

Some heuristics are categorized as metaheuristics, which are general strategies adaptable to various optimization problems. The simplest and most widely used heuristic is stochastic descent, particularly for minimization problems [48]:

- ✓ An initial solution is chosen.
- ✓ A neighboring solution is selected randomly.
- ✓ If the new solution has a lower objective function value, it becomes the new reference point, and its neighbors are explored.
- ✓ If not, another neighbor is selected.
- ✓ The process stops when no better solution can be found.

II.2.3 Classification of optimization methods

Classification of optimization can be divided into several categories depending on their characteristics are deterministic approaches stochastic techniques stochastic learning approaches this is in Fig (II.1):

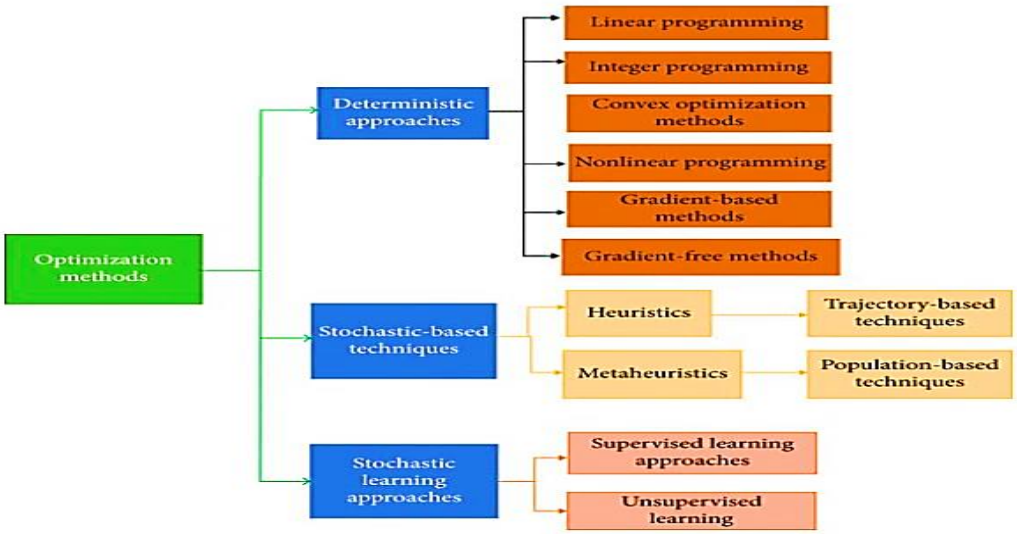


Fig (II.1): Classification of optimization [51].

II.3 Metaheuristics

II.3.1 Definition of Metaheuristics

A metaheuristic is a high-level, problem-independent algorithmic framework that offers guidelines or strategies for designing heuristic optimization algorithms. When a heuristic optimization algorithm is developed following these guidelines for a specific problem, it is also called a metaheuristic. The term "metaheuristic" was introduced by Glover in 1986 and combines the Greek prefix meta- (meaning "beyond" or "higher-level") with heuristic (derived from the Greek ‘’heuriskein’’ or ‘’euriskein’’, meaning "to search) [49]. Unlike heuristic that often problem specific in return then metaheuristic is general more flexible and complex optimization problem.

II.3.2 Types of Metaheuristics

Many optimization algorithms have been inspired by the behaviours observed in natural systems, including both biological and physical phenomena. As shown in Fig (II.2), these nature-inspired algorithms are generally classified into several categories. The heuristic category typically focuses on single-solution methods that use specific rules to guide the search process; notable examples include Simulated Annealing (SA) and Hill-Climbing (HC). In contrast, metaheuristic algorithms, often referred to as population-based methods, involve multiple interacting candidate solutions and are capable of adapting to a variety of problems through mechanisms like parameter tuning. These metaheuristics are further divided into four major groups: (1) Evolutionary

Algorithms (e.g., GA, GP, DE); (2) Human-based techniques (e.g., TS, TLB); (3) Physics-inspired methods (e.g., GSA, CFO, BBBC); and (4) Swarm-based algorithms, which imitate the collective behaviour of animals, such as HHO, MFO, and CSA [50].

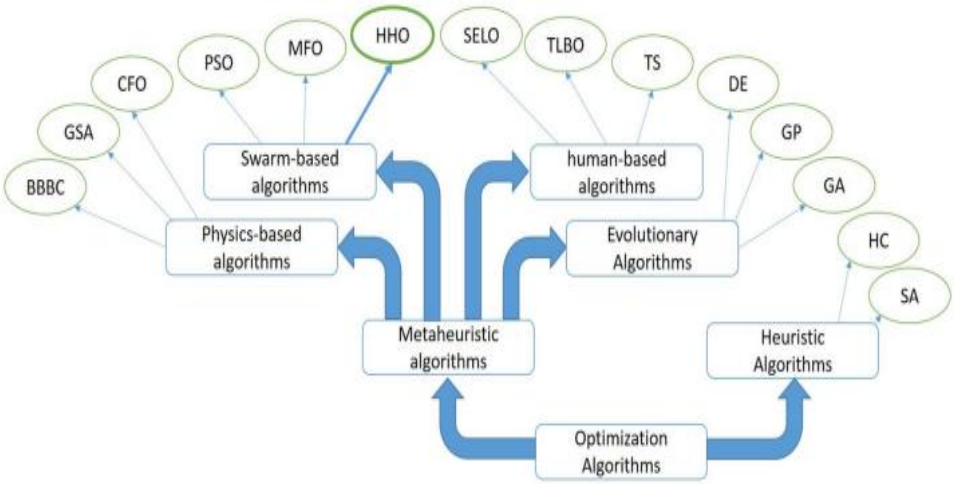


Fig (II.2): Optimization algorithms [52].

II.4 Harris Hawks Optimization

II.4.1 Overview and inspiration from nature

In 1997, Louis Lefebvre introduced a method to evaluate avian intelligence by observing changes in feeding behaviors. His findings indicated that hawks are among the most intelligent bird species. One species in particular, the Harris's hawk, stands out for its unique group hunting strategy and highly social behavior. Native to the southern regions of Arizona (USA), this bird of prey is renowned for its ability to coordinate with others during hunts a trait uncommon among raptors, which generally hunt alone. Behavioral studies have shown that all group members participate in hunting, especially outside the breeding season. The typical hunting approach involves a sudden ambush, as illustrated in Fig (II.3). This cooperative behavior inspired the development of the Harris Hawks Optimization (HHO) algorithm, a nature-inspired metaheuristic that effectively addresses complex optimization tasks by balancing the exploration and exploitation phases [50].

Fig (II.3): Harris's hawk and their behaviours [53].

II.4.2 Phases of HHO

The Harris Hawks Optimization algorithm comprises two fundamental phases: Exploration, Exploitation as depicted in Fig (II.4). Each phase emulates a distinct behavioral pattern of Harris hawks in the course of hunting:

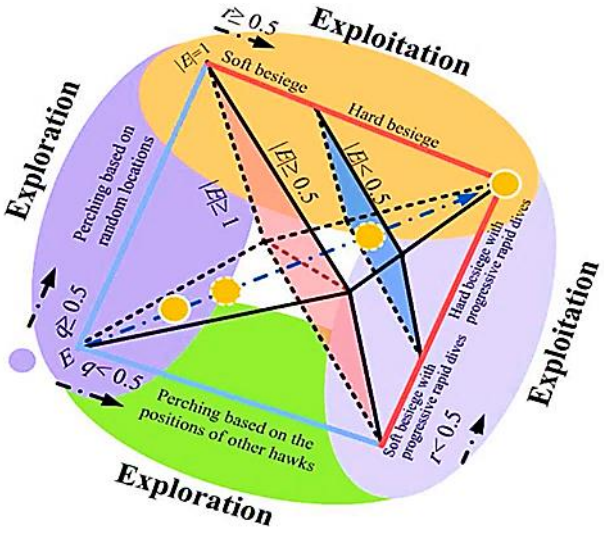


Fig (II.4): Several Phases of HHO [52].

II.4.2.1 Exploration Phase

The exploration phase of HHO is a stage that looks for potential areas in the solution space. It resembles how Harris hawks behave when they are looking for prey across a wide area:

- Random Perching: The hawks, or potential solutions, perch in the search space at random while exploring. The population is diversified as a result of this randomness, which is crucial for

preventing an early convergence to local optimality [55]. The following equation describes the position update during this phase:

$$X(t+1) = X_{rand}(t) - r_1 . |X_{rand}(t) - 2r_2 . X(t)| \quad \text{(II.2)}$$

where $X(t)$ is the current position.

$X_{rand}(t)$ is a randomly selected Hawk.

r_1, r_2 are random numbers in $[0, 1]$.

- Escape Energy: The escape energy (E), which diminishes over iterations, regulates the change from exploration to exploitation. The method stays in the exploration stage when $|E| \geq 1$.
- Random Locations: To broaden the variety of their search, hawks also utilize other hawks' locations to locate new investigation spots [56].

II.4.2.2 Exploitation Phase

The goal of the exploitation phase is to refine the This phase simulates the behavior of hawks when they get close to the prey and start hunting. Depending on the prey's escape energy and the hawks' attack strategy, the exploitation is divided into four tactics [57].

1. Soft Besiege: The hawks employ a soft besiege technique when the prey has sufficient energy to flee $|E| \geq 0.5$. The following provides the position update:

$$X(t+1) = \Delta X(t) - E . |J . X_{rand}(t) - X(t)| \quad \text{(II.3)}$$

where (J) is the prey's random leap strength and $\Delta X(t)$ is the difference between the prey's position and the current Hawk's position.

2. Hard Besiege: Hawks employ this tactic to forcefully grab prey when their energy level is low $|E| < 0.5$:

$$X(t+1) = X_{rand}(t) - E . |\Delta X(t)| \quad \text{(II.4)}$$

3. Progressive Rapid Dives and Soft Besiege: This tactic is employed when the prey is likely to flee but the hawks are still wary. The hawks use progressive dives to modify their positions in response to the movements of their prey [58].

4. Hard Besiege with Progressive Rapid Dives: The hawks employ a hard besiege with rapid dives to guarantee capture once the prey is worn out. This approach is more forceful and concentrates on taking advantage of the most well-known answer [59].

II.4.2.3Transition Between Phases:

The escape energy (E), which is determined as follows, controls the change from exploration to exploitation:

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad \text{(II.5)}$$

where (t) is the current iteration, (T) is the maximum number of iterations, and E_0 is the initial energy. The algorithm investigates when $|E| \geq 1$ and exploits when $|E| < 1$ [13].

II.5 Organigrams of HHO algorithm

The accompanying flowchart is in Fig (II.5) which outlines the operation of HHO:

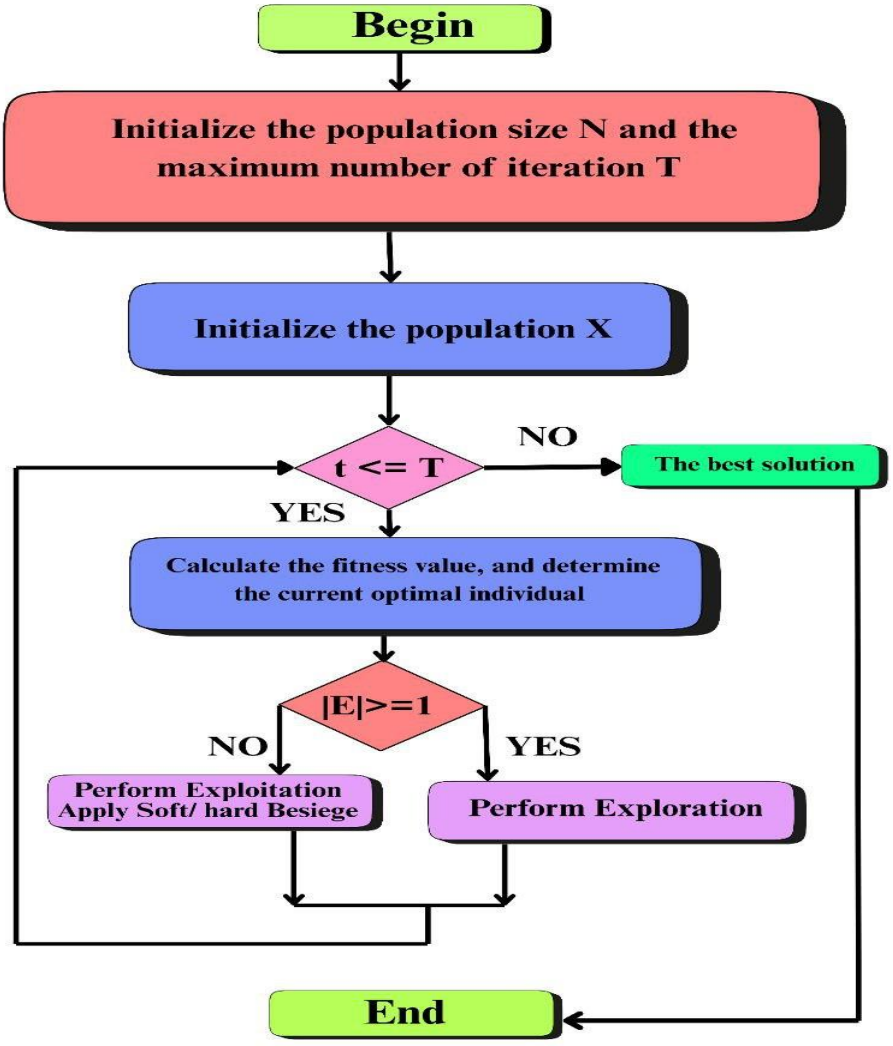


Fig (II.5): Organigrams of HHO algorithm [54].

II.6 HHO-Based Optimization of BLDC Motor Performance

The HHO algorithm provides a flexible and effective tool for enhancing BLDC motor performance in a number of areas, including as fault detection, speed control, torque ripple reduction, efficiency enhancement, and parameter identification [56].

II.7 Objective function (fitness function)

The objective function was used to improve system performance through the Harris Hawks Optimization (HHO) algorithm by tuning the PID controller parameters. The ITAE criterion (Integral of Time multiplied by Absolute Error) was adopted as the performance index, as it gives higher weight to errors occurring at later times, thereby reducing the settling time and enhancing the dynamic response of the system. The ITAE is defined by the following equation and other equation of error:

- Integral of Time multiplied by Absolute Error (ITAE):

$$I_{ITAE} = \int_0^T t|e(t)|\,dt \tag{II.6}$$

- Integral of Squared Error (ISE):

$$I_{ISE} = \int_0^\infty e^2(t)\,dt \tag{II.7}$$

- Integral of Absolute Error (IAE):

$$I_{IAE} = \int_0^\infty |e(t)|\,dt \tag{II.8}$$

However, the ITAE was chosen in this study due to its effectiveness in minimizing long-term errors and improving overall control performance.

II.8 BLDC Motor modeling and control using HHO

In this study we utilized the HHO optimization technique. The diagram of this method of optimization in a BLDC motor control at no load, on load and also for the reversal of rotation direction with over speed is shown in Fig (II.6), and the error is (ITAE).

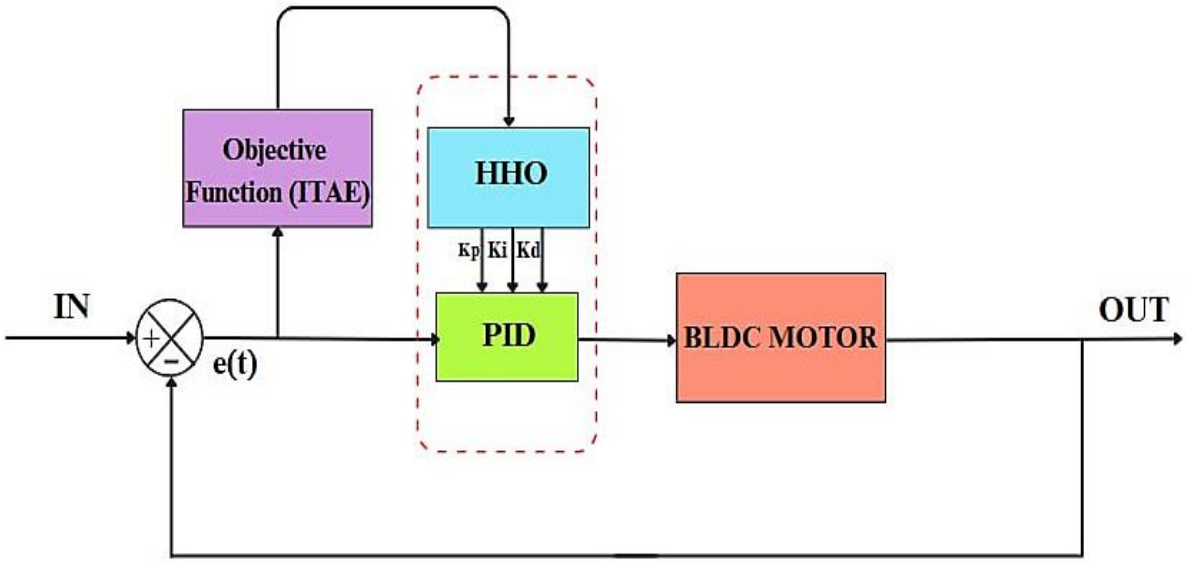


Fig (II.6): Optimizing PID Controllers using HHO for Brushless DC (BLDC) Motor [1].

II.9 Simulation and results

A simulation model of a tuned PID-controlled BLDC motor system was developed in MATLAB/Simulink (see Annex D), three operational scenarios: no-load, loaded, and speed inversion.

Table (II.1)**:** Parameter of HHO

Min	Max	N_P	$N_{iteration}$	K_p	K_i	K_d
[0.1 0.1 0.01]	[500 3000 0.5]	40	40	39.3862	13.9856	0.00224

Based on the data presented in Table (II.1) we obtained the convergence curve in the Fig (II.7):

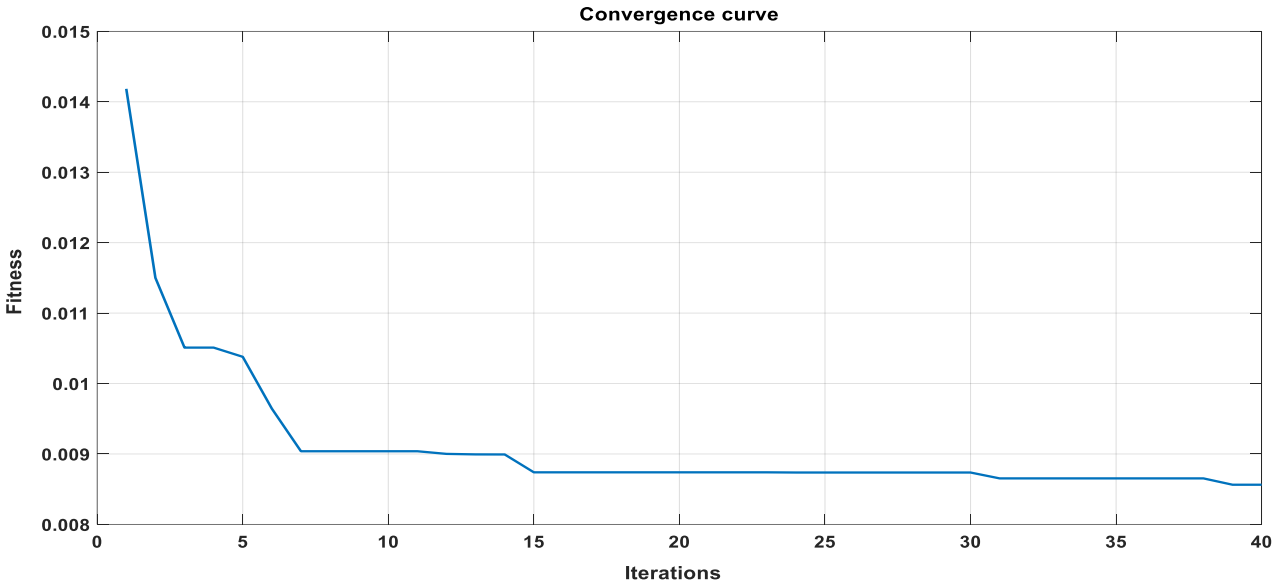


Fig (II.7): Convergence curve of HHO.

➢ Test 1: with no load ($T_L = 0$(N.m))

- Over Speed: The motor executed speed step change was applied, increasing the rotational velocity from $1000\,RPM$ to $1500\,RPM$ at $t = 0.15s$ .This transition enabled the assessment of the system's dynamic response to a sudden increase in setpoint speed.

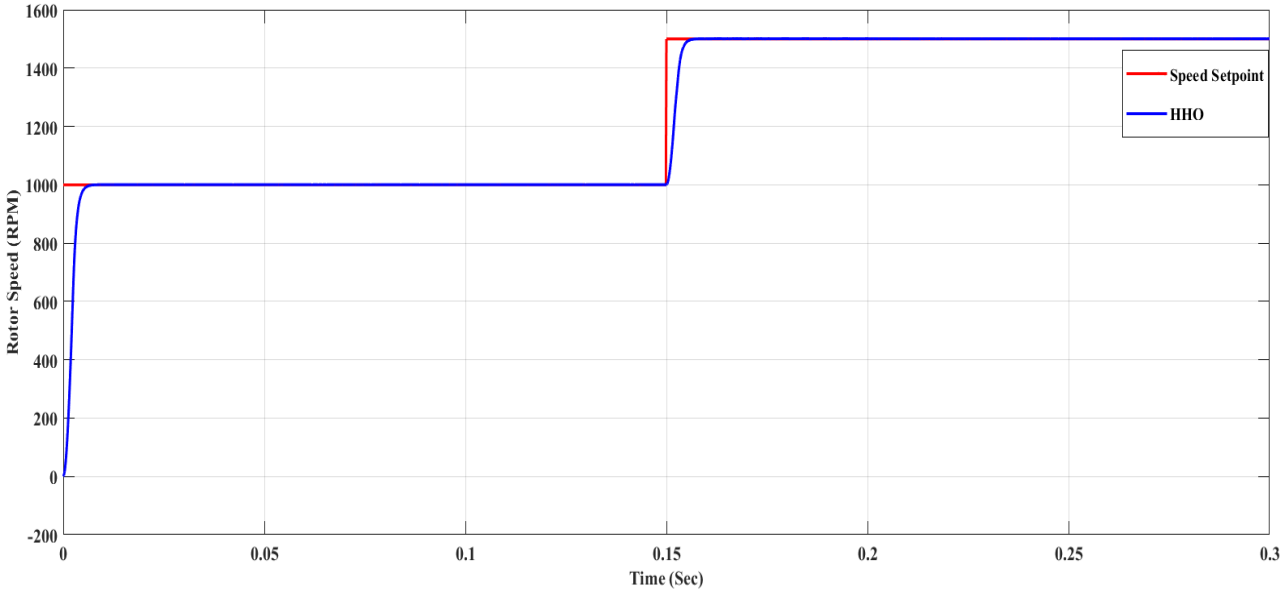


Fig (II.8): Rotor Speed response with no load.

- Rotor speed inversion: The motor executed a directional speed reversal from $+1000\ RPM$ to $-1000\ RPM$ at $t = 0.15s$ .This transition enabled the assessment of the system's dynamic response to a sudden decrease in setpoint speed.

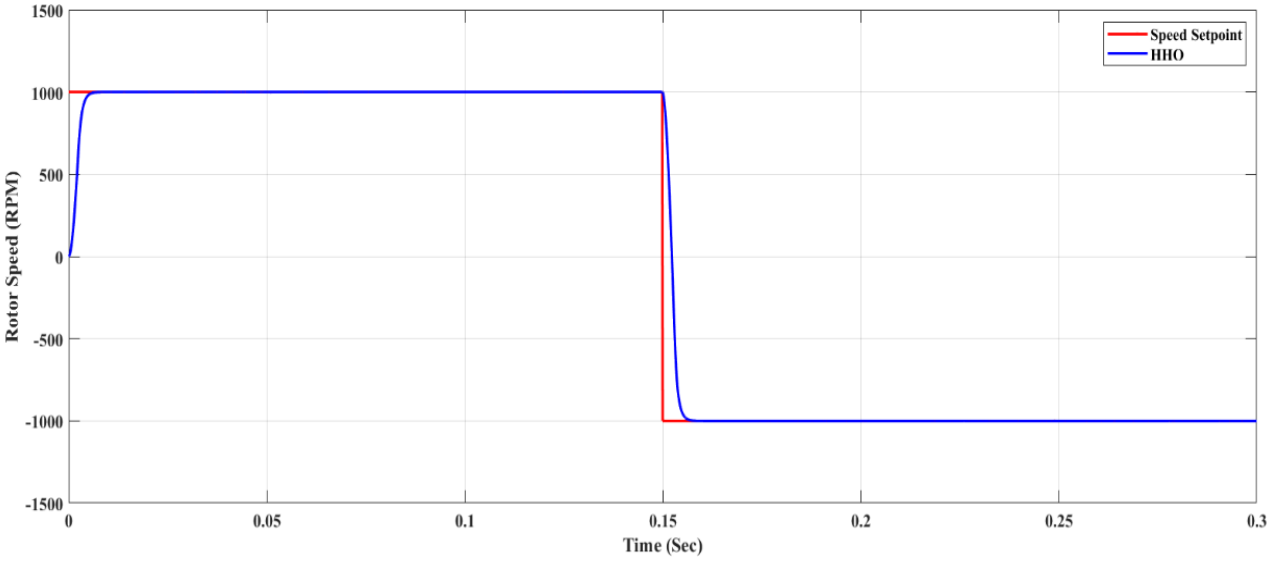


Fig (II.9): Rotor Speed inversion response with no load.

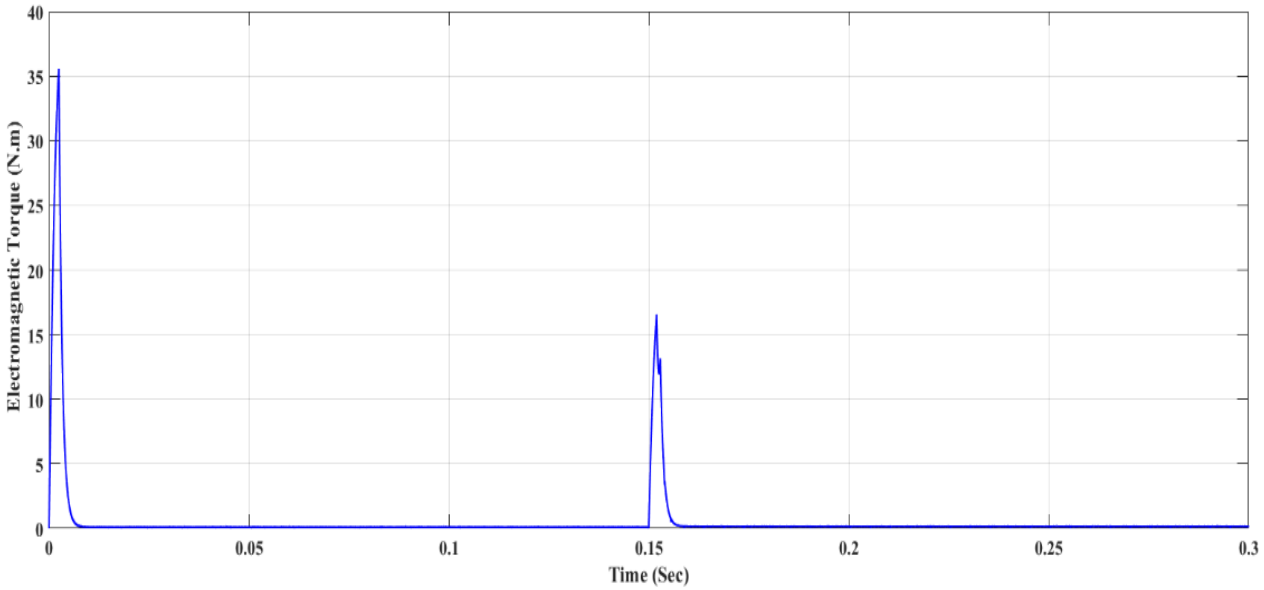


Fig (II.10) Electromagnetic torque response with no load.

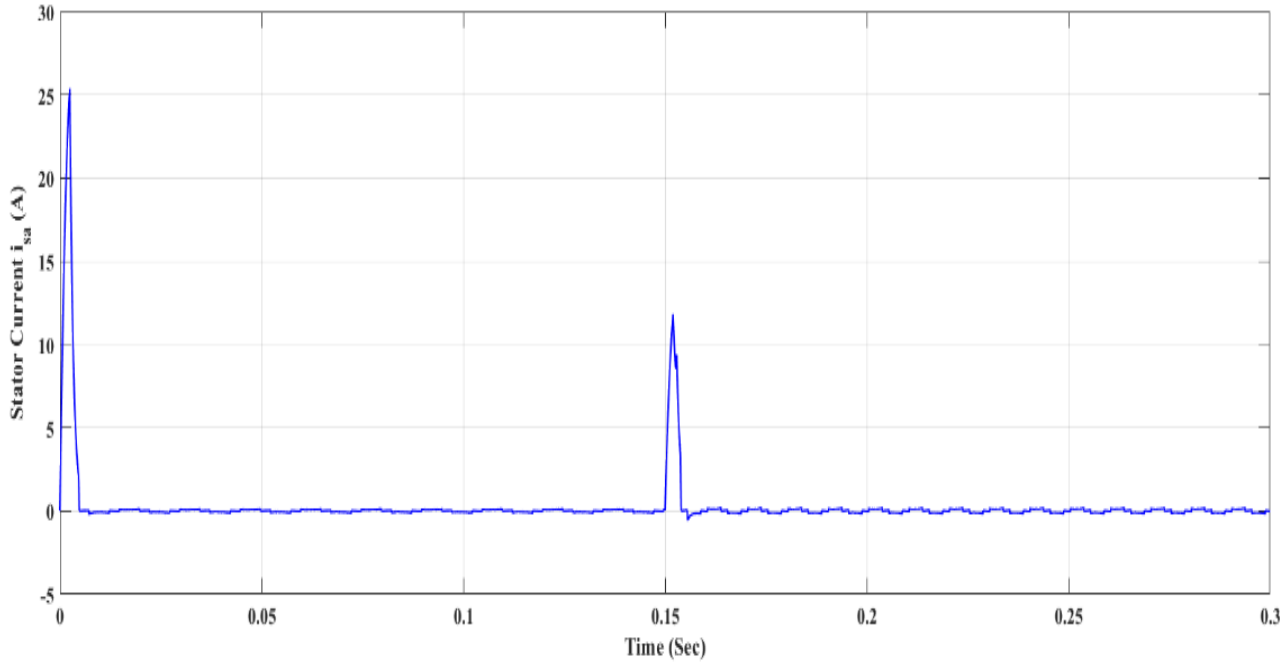


Fig (II.11): Stator current response with no load.

- Test 2: with load: where $T_L = 5$(N.m) added at $t = 0.15$(s), speed reference is $1000RPM$.

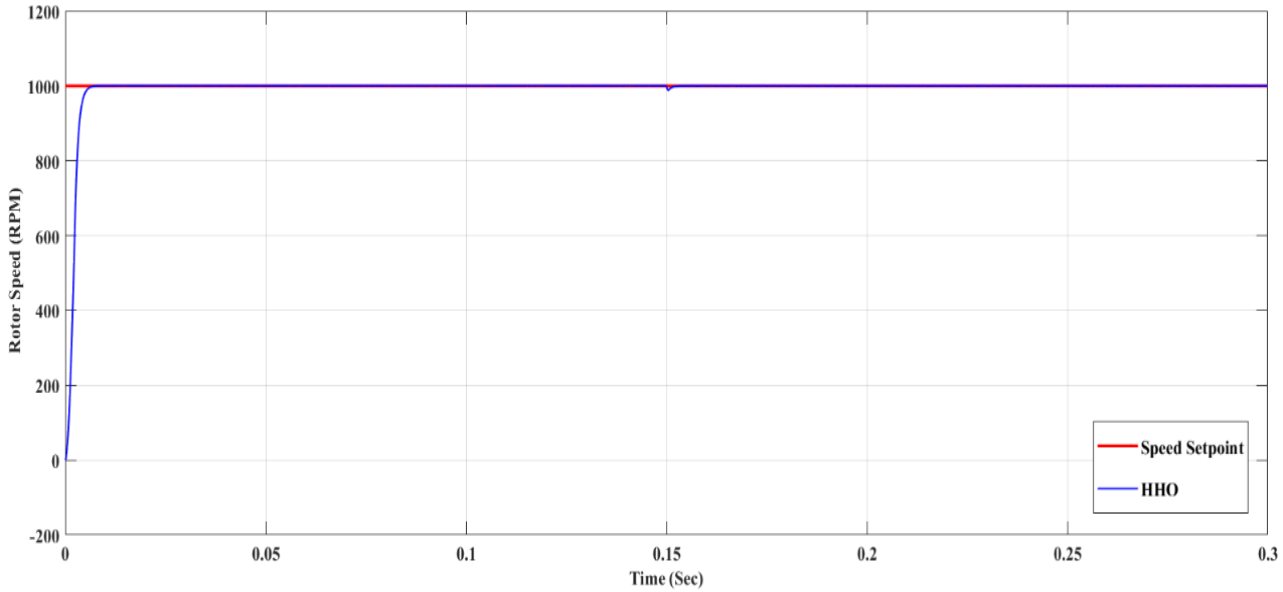


Fig (II.12): Rotor Speed response with load.

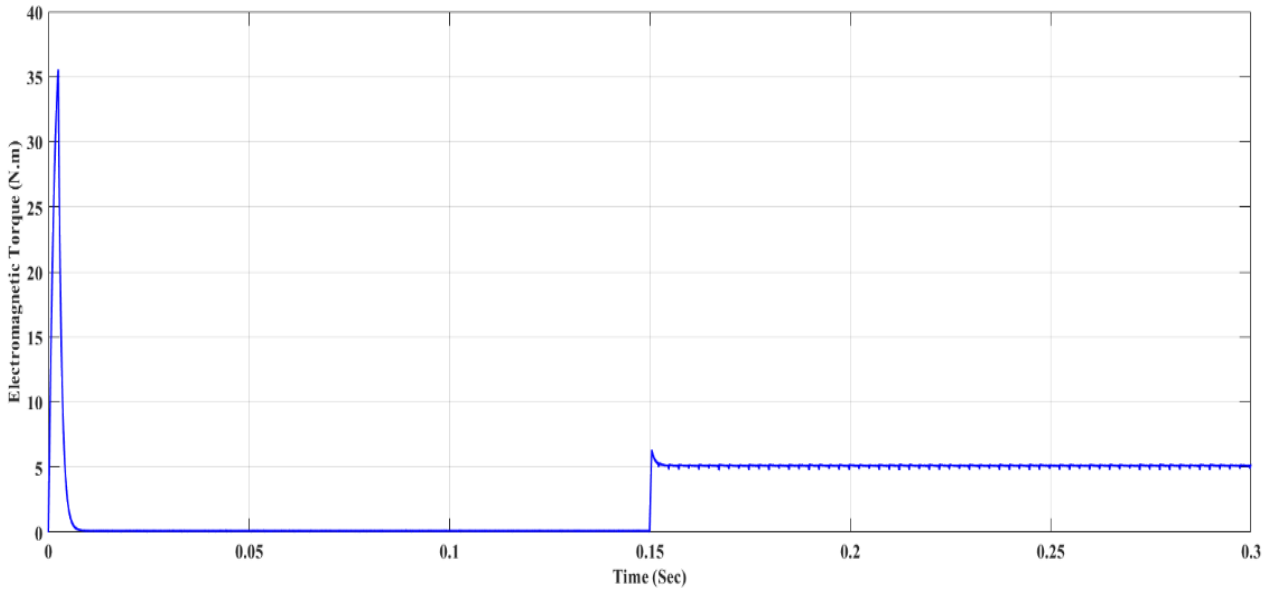


Fig (II.13): Electromagnetic torque response with load.

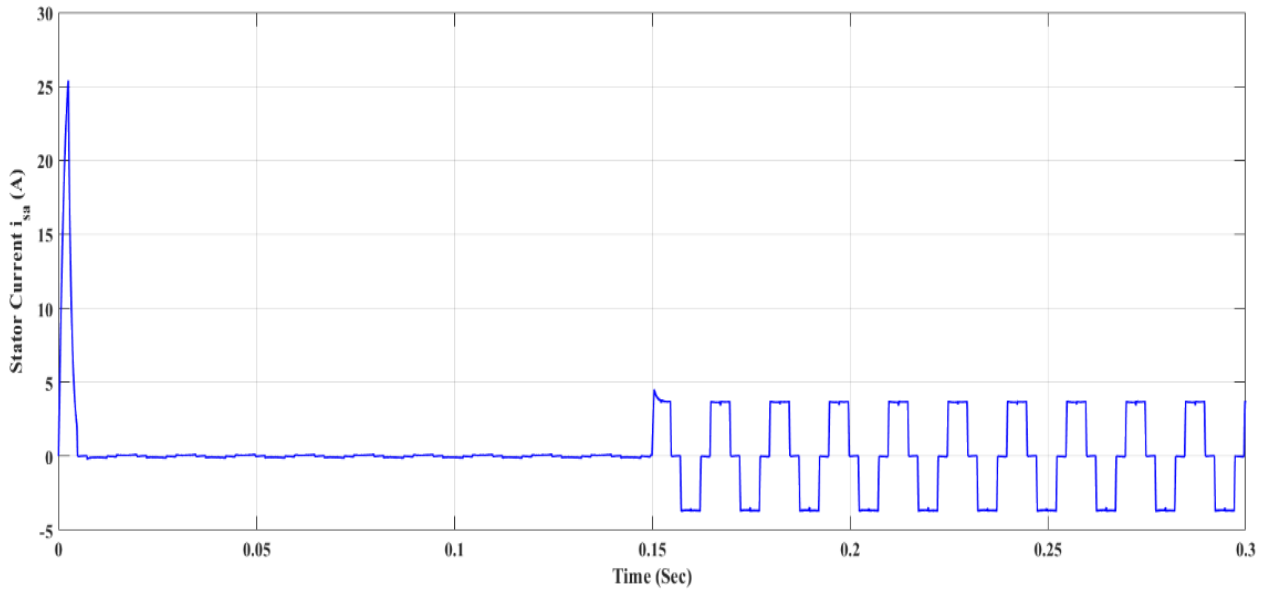


Fig (II.14): Stator current response with load.

II.9.1 Comments and interpretations:

- Test one: with no load
  - As shown in Fig (II.8) Speed Response (No Load – HHO) the motor rapidly reaches the reference speeds (1000 RPM then 1500 RPM) with minimal overshoot and smooth transitions, demonstrating accurate and stable speed tracking by the HHO-optimized PID controller under no-load conditions.
  - As shown in Fig (II.9) Speed Response during Speed Inversion the motor speed shifts rapidly from $+1000$ RPM to -1000 RPM around 0.15 seconds. The inversion is smooth and fast, with no overshoot or oscillation, reflecting effective control performance during direction reversal.
  - As shown in Fig (II.10) Electromagnetic Torque (No Load – HHO) Initial torque peaks to overcome inertia, then quickly drops near zero. A second peak occurs during the speed step, indicating fast dynamic response in no-load operation.
  - As shown in Fig (II.11) Stator Current (No Load – HHO) the current peaks briefly at startup and during the speed change, then stabilizes with low ripple, confirming efficient current regulation without external load torque.

- Test two: with load
  - As shown in Fig (II.12) Speed Response (With Load) the BLDC motor speed rises rapidly to reach 1000 RPM without noticeable overshoot and remains stable at this reference value despite the applied load. This demonstrates the effectiveness of the HHO algorithm in providing accurate speed tracking and maintaining stable performance under loaded conditions.
  - As shown in Fig (II.13) Electromagnetic Torque (With Load) the Electromagnetic torque initially peaks at approximately 35 N·m to overcome the inertia and external load. After about 0.15 seconds, it stabilizes around 5 N·m, indicating the system's ability to deliver the required torque and operate smoothly under load.
  - As shown in Fig (II.14) Stator Current (With Load) the stator current rises sharply to nearly 28 A at startup, then gradually decreases and stabilizes around 5 A, with visible ripple due

to inverter switching. This reflects effective current control and stable operation of the motor drive under loaded conditions.

II.9.2 Comparison between HHO, PID-APPS and PID-ZN

In this section, a test under torque load was carried out on a BLDC motor to compare the performance of the conventional PID-APPS and the PID-HHO and PID-ZN.

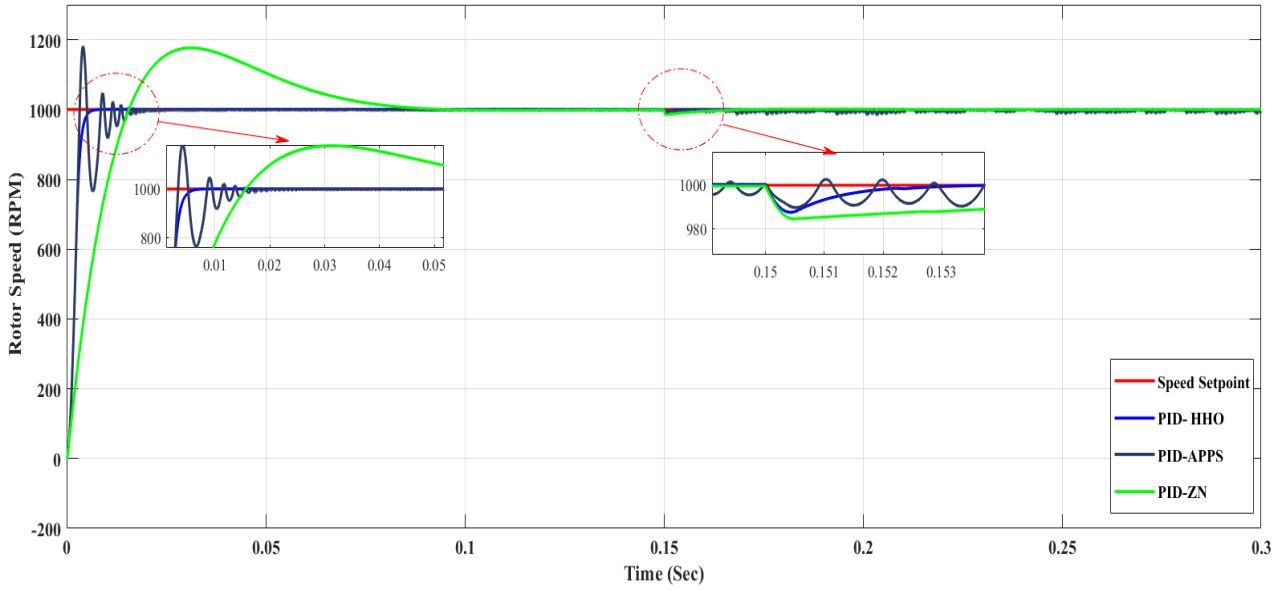


Fig (II.15) :Speed Response under Torque Load (PID vs. HHO vs ZN).

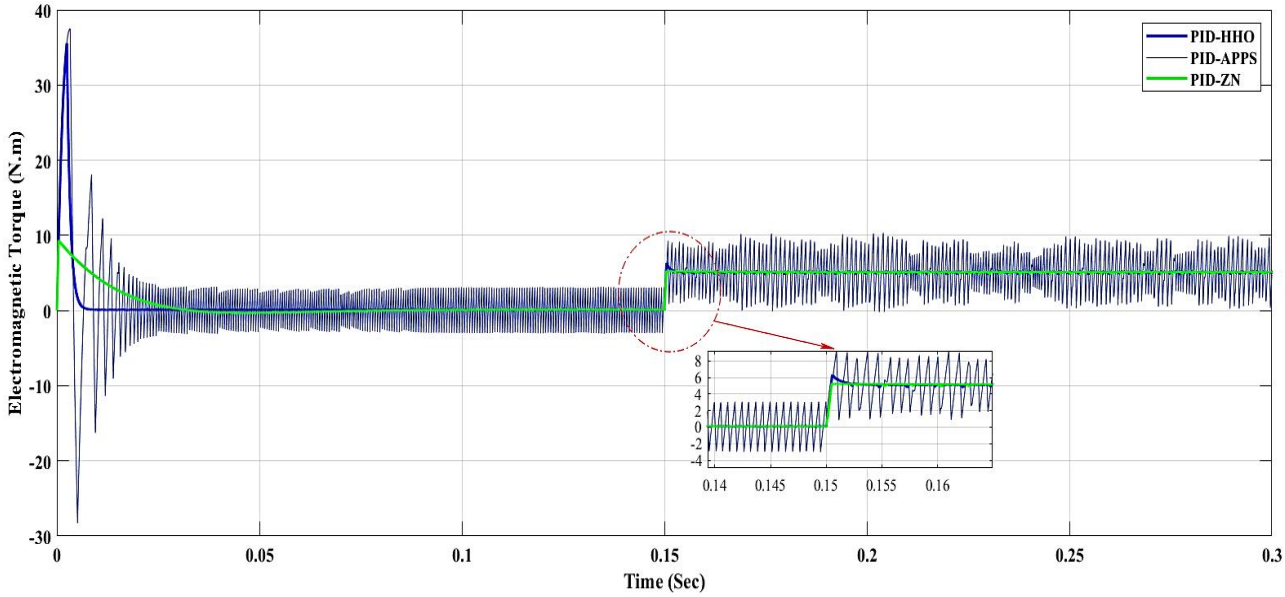


Fig (II.16): Electromagnetic Torque Response (PID vs. HHO and ZN).

II.9.3 Comments and interpretations

- This Fig (II.15) illustrates the speed response of a BLDC motor under a sudden torque load using three PID tuning methods: HHO, APPS, and Ziegler-Nichols (ZN). The response using the HHO algorithm demonstrates superior performance, with the fastest rise to the reference speed, minimal overshoot, and rapid settling without noticeable oscillations. This indicates the effectiveness of HHO in optimizing the dynamic performance of the system.
- The second Fig (II.16) shows the electromagnetic torque response under the same conditions. It is evident that the ZN and APPS methods suffer from pronounced and persistent ripples after the load disturbance. In contrast, the HHO-based controller exhibits a significantly smoother torque response with reduced oscillations. These ripples in the torque signal are undesirable as they can lead to mechanical stress and inefficiencies.

From both figures, it is clear that the HHO algorithm achieves better overall system performance, providing faster and more stable responses in both speed and torque dynamics, making it a more robust tuning strategy compared to ZN and APPS methods.

Table (II.2): Performances of each control technique.

Control techniques	Rise time (sec)	Settling time (sec)	Overshoot (%)
PID-HHO	0.0027	0.0049	0.0480
PID-APPS	0.0022	0.0148	19.5101
PID-ZN	0.0116	0.0744	17.6130

In comparison between the PID-HHO method and the other control methods (PID-APPS and PID-ZN), and as shown in the Fig (II.17) and Fig (II.18) above and in Table (II.2), it can be concluded that the PID-HHO technique provides better performance. It achieves the smallest rise time, settling time, and has the lowest overshoot. Moreover, it demonstrates a more stable response

and is less affected by the torque load variations compared to the conventional PID techniques, which show higher rise time, greater overshoot, and more sensitivity to load disturbances.

II.10 Conclusion

At the end of this chapter, we presented an overview of the Harris Hawks Optimization (HHO) algorithm, which is inspired by the cooperative hunting strategy of Harris hawks in nature. Through simulations, we compared the HHO-based PID tuning to conventional methods such as Ziegler–Nichols (ZN) and APPS. The HHO approach demonstrated significantly better performance in terms of rise time, settling time, and overshoot.

In the next chapter, we will apply this optimized control strategy to a Brushless DC (BLDC) motor, using an Arduino and an Electronic Speed Controller (ESC), aiming to ensure adaptive and real-time response for improved motor performance.

Chapter III
Application

Chapter III : Application

III.1 Introduction

Brushless DC (BLDC) motors are widely used in modern automation and robotics due to their efficiency, reliability, and precise control capabilities. The combination of microcontroller platforms, such as Arduino, provides an accessible and flexible approach for developing and testing BLDC motor control systems.

This chapter introduces the general process of implementing and simulating BLDC motor control using Arduino Mega 2650 and Proteus. It highlights how these tools can be used to design, simulate, and validate motor control strategies, offering valuable experience for both educational and practical engineering applications.

III.2 Tools and devises

In this project, we used various tools and devices, which we categorized into hardware components, such as the BLDC motor, Arduino boards, ESC, and potentiometer, and software tools, including the Arduino IDE and Proteus simulation software, to design, control, and test the system effectively.

III.2.1 Hardware components

The hardware utilized in this project consists of the A2212/13T/930KV BLDC motor, Arduino Mega and Uno microcontrollers, a 30A ESC, a potentiometer, a laboratory DC power supply, and a breadboard for assembling and connecting the circuit. Each of these components will be described in detail below.

III.2.1.1 Cables and wiring

A cable is an assembly of one or more insulated conductors designed to transmit electrical power or data signals between devices or subsystems. It serves as a crucial medium for ensuring signal integrity and reliable energy distribution in electrical and electronic infrastructures [54].

To connect our project component (the BLDC motor, Arduinos, DC voltage source, and potentiometer) we used quality cables (male/male 20 cm long) and reliable connectors to ensure

efficient energy transmission and minimal signal loss, which helped maintain the overall performance and stability of the circuit. Like it shown in the Fig (III.1) down below:

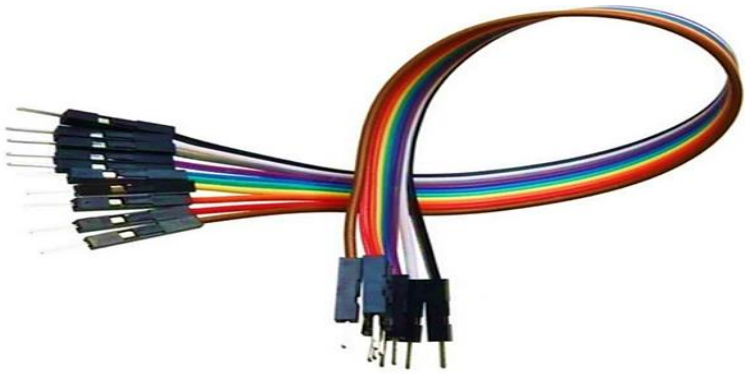


Fig (III.1): male/male 20 cm cables [55].

III.2.1.2 Potentiometer

A potentiometer is a three-terminal analogue device functioning as a variable voltage divider. It enables real-time manual adjustment of voltage levels in a circuit and is widely employed in instrumentation, signal conditioning, and user-interface control applications due to its simplicity and reliability [56].

We used the potentiometer to control the BLDC motor speed by connecting it to an analog input on the Arduino, allowing us to vary the voltage signal as we turned the knob, and adjusted the motor's speed accordingly through the motor controller.

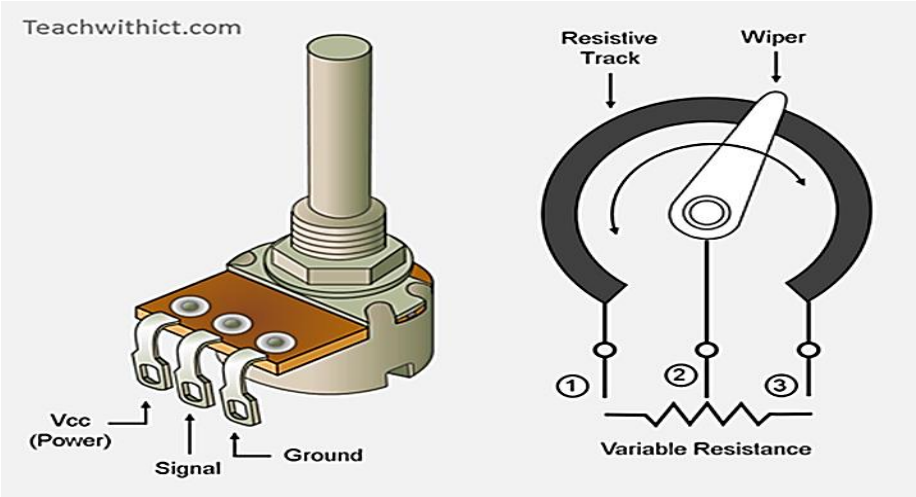


Fig (III.2): Structure and working principal of a Potentiometer [57].

III.2.1.3 DC Power Supply

Reliable DC voltage sources are fundamental in today's electronics, providing the stable and adjustable power needed for devices, testing, and research. Modern DC variable power supplies allow users to precisely set and control DC voltage levels, making them essential for circuit prototyping, experimentation, and troubleshooting [58].

We used the lab DC power supply in our project to provide a stable and adjustable voltage source for powering the BLDC motor through the ESC, ensuring the motor received sufficient and consistent energy for reliable operation and precise speed control via the Arduino and potentiometer.

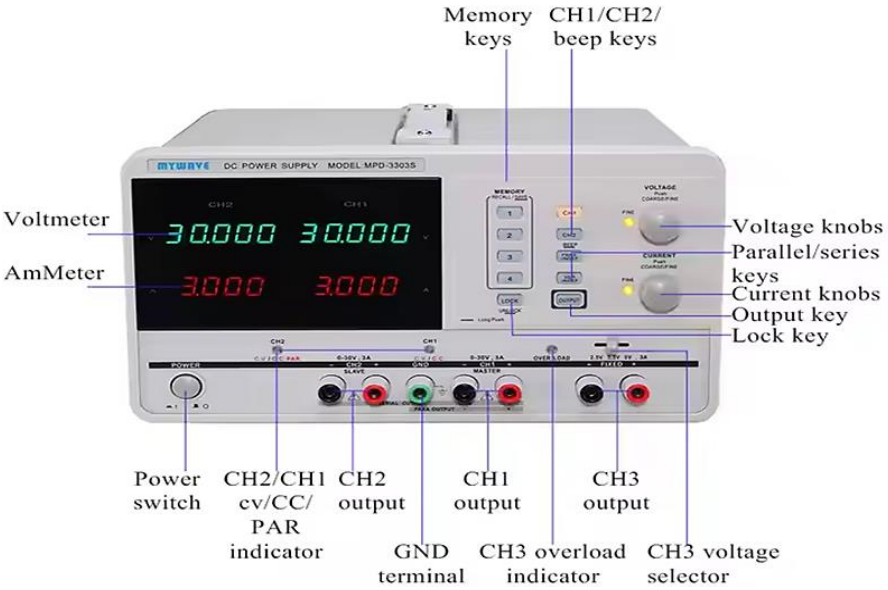


Fig (III.3): Lab DC source and its components [59].

III.2.1.4 Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 chip. It offers 54 digital input/output pins, 16 analogue inputs, 4 UARTs (hardware serial ports), and a large memory capacity, making it ideal for projects requiring a large number of connections and sensors. It is commonly used in complex robotics, automation, and embedded system applications [60].

We used the Arduino Mega in our project to control the BLDC motor by generating PWM signals through its digital pins, which were sent to the ESC to regulate motor speed and direction.

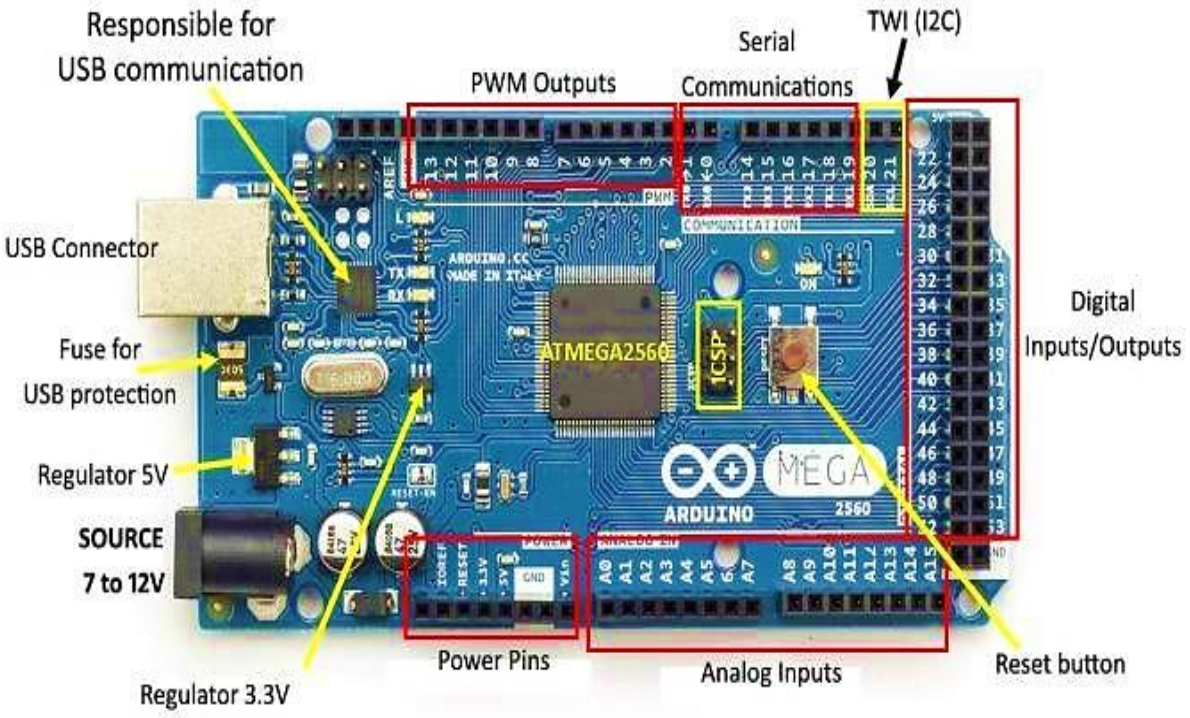


Fig (III.4): Arduino Mega 2560 and its components [61].

III.2.1.5 Arduino Uno R3

The Arduino Uno is one of the most widely used and popular development boards, powered by the ATMega328P microcontroller. It is favored by the maker and developer community due to its affordability, ease of use, and the wide range of readily available modules that simplify the process of building prototypes and new projects. The board features 14 digital I/O pins, including 6 PWM capable 8-bit outputs, 6 analog inputs with 10-bit resolution, and supports essential communication interfaces such as SPI, I2C, and UART.

Although many variants of the Arduino Uno are available globally, a large number of them are clones or replicas of the original version. As a result, their appearance or color may differ from the official board shown above [62].

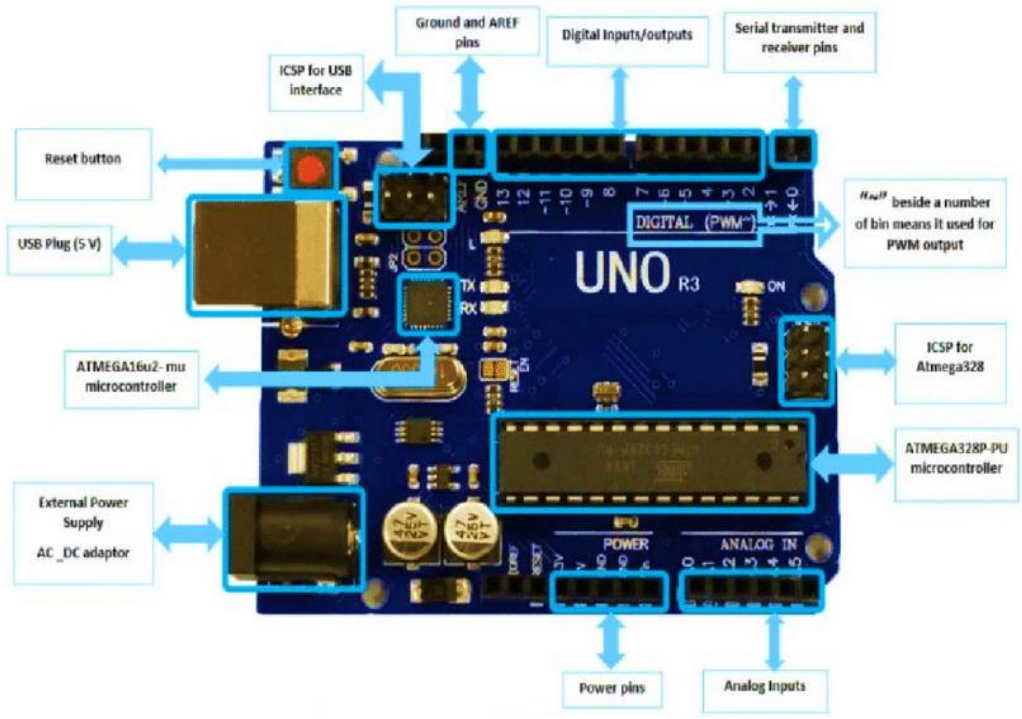


Fig (III.5): Arduino Uno R3 and its components [63].

III.2.1.6 Breadboard

A breadboard is a tool used for prototyping electronic circuits without components like resistors soldering. It consists of a grid of interconnected holes that allow, capacitors, and integrated circuits (ICs) to be placed and connected using jumper wires. Breadboards are essential for testing and iterating circuit designs before finalizing them on a printed circuit board (PCB) [64].

We used the breadboard in our project to create a flexible and accessible platform for connecting the Arduino, ESC, potentiometer, and BLDC motor driver components, allowing us to easily prototype, modify, and troubleshoot the control circuit for the BLDC motor without soldering, while ensuring stable connections for power, signal, and feedback lines.

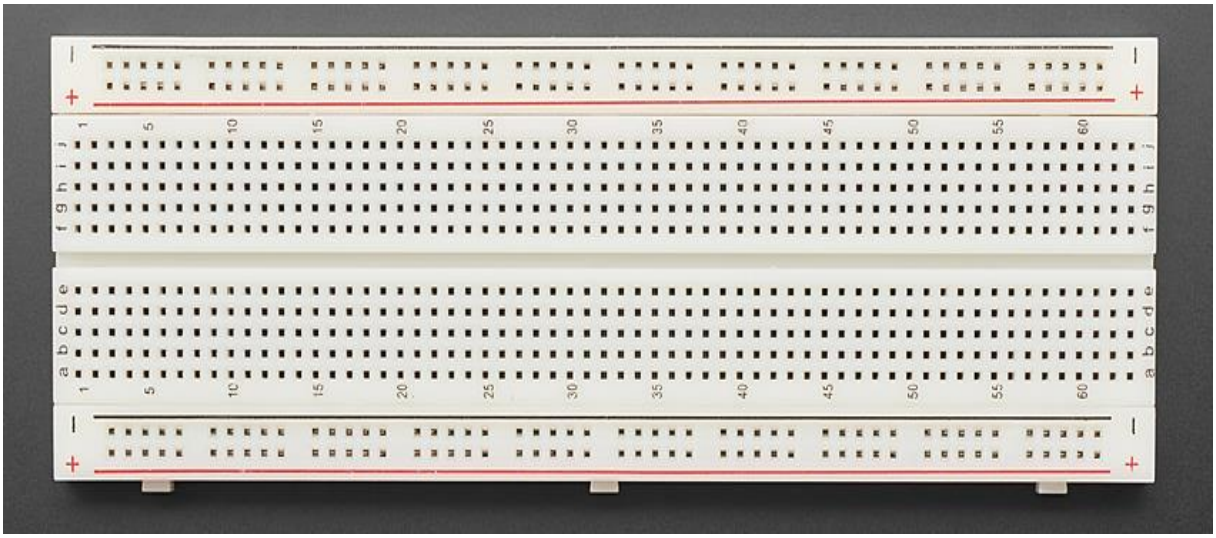


Fig (III.6): Breadboarded [65].

III.2.1.7 Electronic Speed Controller (ESC)

An Electronic Speed Controller (ESC) is an electronic circuit that controls the speed, direction, and braking of an electric motor. In the case of BLDC motors, an ESC is used to provide the correct signals to the motor for precise control over its operation. It is an essential component in drones, RC vehicles, and other motorized applications [66].

We used the ESC 30A in our project to precisely control the speed of the BLDC motor by receiving PWM signals from the Arduino, efficiently regulating power delivery to the motor while providing stable operation and protection features such as overcurrent and thermal management.

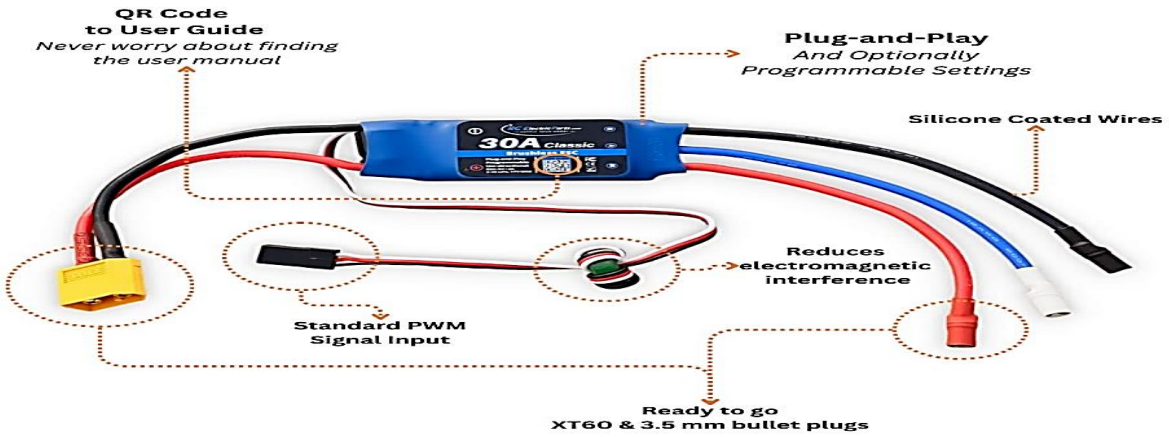


Fig (III.7): Electronic Speed Controller (ESC)and its components [67].

III.2.1.8 Brushless DC motor (BLDC)

A Brushless DC (BLDC) motor is an electric motor that operates without brushes and commutators. It uses electronic controllers to drive the motor, offering higher efficiency, reliability, and longevity compared to brushed DC motors. BLDC motors are widely used in applications requiring high-performance, such as robotics, drones, and electric vehicles [68].

In this project, we used the A2212/13T/930KV BLDC motor, selected for its reliability and suitable speed-torque characteristics, and referenced its datasheet down in the following to ensure proper integration and safe operation within our circuit.

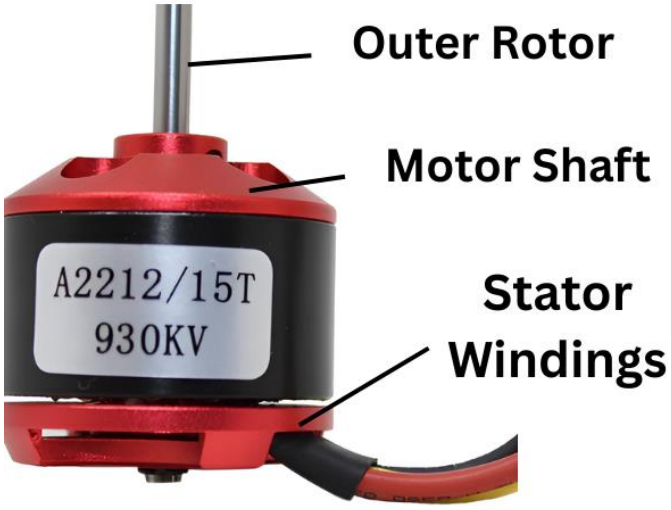


Fig (III.8): A2212/13T/930 BLDC motor [69].

Table (III.1): A2212/13T/930 BLDC motor Datasheet.

Parameter	**Specification**
Model	$A2212/13T/930KV$
Motor Type	Brushless DC (BLDC)
KV Rating	930 KV
Max Current	13 A
Max Voltage	11.1 V $(3S\ LiPo)$
Shaft Diameter	3.17 mm

Weight	$50\ g$
Max Power	$220\ W$
Recommended Prop	$8x4.5, 9x4.7, 10x4.5$

III.2.2 Software's

The software tools employed in this project include Arduino IDE for programming the microcontrollers and Proteus Professional for simulating and testing the circuit; each software will be discussed in detail in the following sections.

III.2.2.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software platform used to write, compile, and upload programs to Arduino microcontroller boards. The IDE simplifies the process of creating embedded systems by providing a user-friendly interface with built-in libraries for controlling hardware components. It supports multiple programming languages, primarily C and C++, and allows users to develop and test code in a variety of environments. The Arduino IDE is widely popular in both academic and hobbyist communities for learning embedded systems and rapid prototyping [70].

We used Arduino IDE 2.3.4 in our project to write, compile, and upload code to the Arduino board, enabling precise control and coordination of the BLDC motor, ESC, and potentiometer based on our project requirements that will be detailed down below.

The Fig (III.9) shows the Arduino IDE App logo, and the Fig (III.10) shows the Arduino IDE interface:

Fig (III.10): Arduino IDE interface [71].

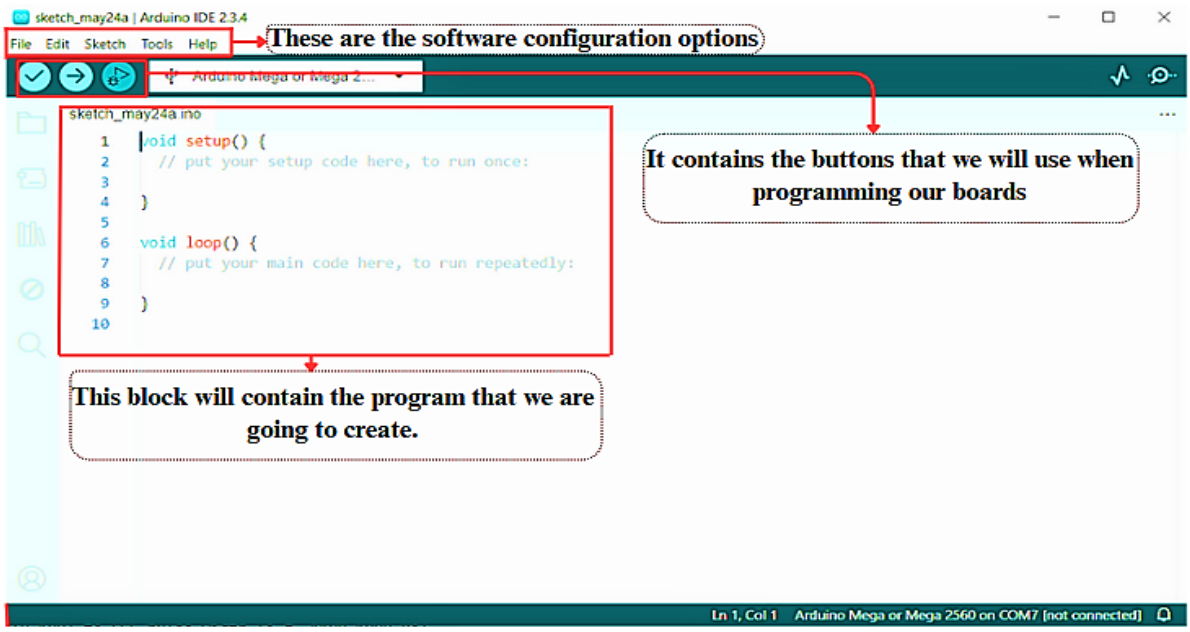


Fig (III.9): Arduino IDE App logo.

III.2.2.2 Proteus professional 8. 8.17

Proteus 8.17 is a version of the Proteus Design Suite, a comprehensive software used for the simulation and design of electronic circuits. It provides tools for schematic capture, PCB layout, and the simulation of microcontroller-based systems. The software allows users to design, test, and validate circuits and embedded systems in a virtual environment before physical implementation. Proteus also supports simulation of various components like sensors, motors, and displays, and is widely used in academic and professional environments for rapid prototyping and learning [72].

We used Arduino Proteus Professional 8.17 in our project to simulate the entire circuit by designing the schematic, integrating the Arduino-generated .hex file, and testing real-time interactions to validate performance before physical implementation.

The Fig (III.11) shows the Proteus Professional App logo, and the Fig (III.12) shows the Proteus Professional 8.17 interface.

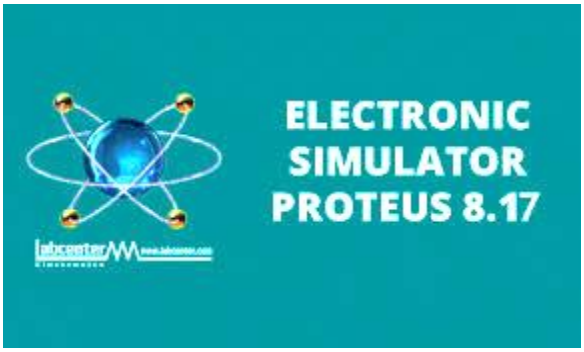


Fig (III.11): Proteus Professional App logo.

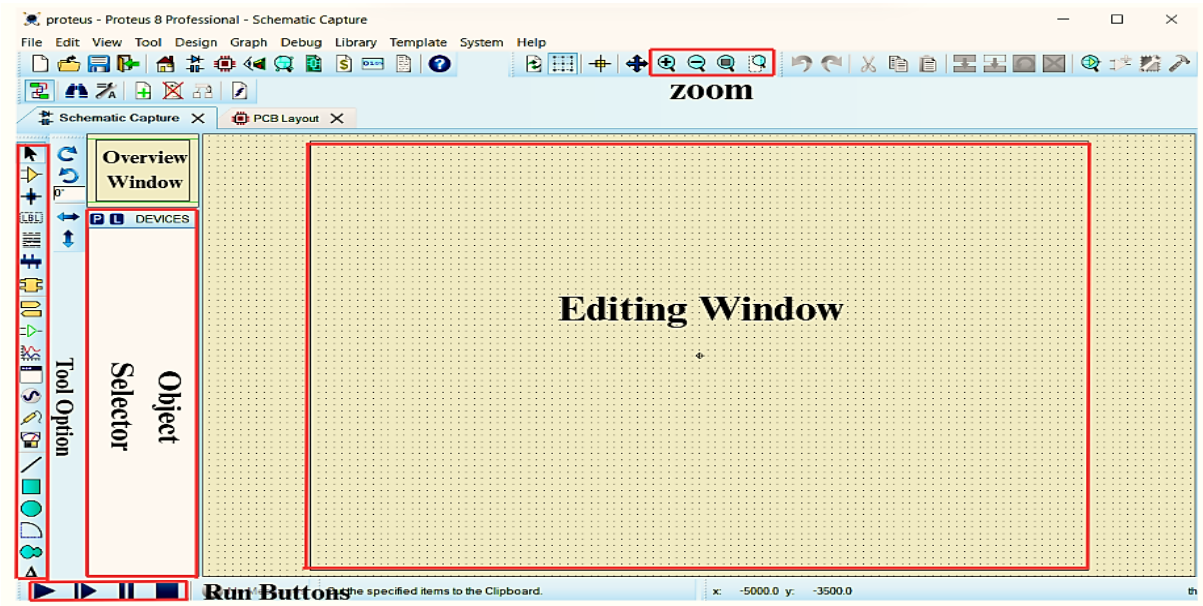


Fig (III.12): Proteus Professional 8.17 interface.

- **Proteus simulation circuit**

In this Proteus simulation, we built the full schematic of our BLDC motor control system based on how we assembled it in real life.

- We used the Arduino Mega 2560 to generate the PWM signals and run our PID algorithm, just like in our physical setup.
- To handle the motor's commutation, we added logic gates and MOSFET driver circuits, which helped us control the signals sent to the motor phases.
- On the right side, we placed the BLDC motor (M1) and connected it to an H-bridge, which we designed to switch the motor coils correctly.

- The PWM signals coming from the Arduino are processed and sent through the H-bridge to drive the motor.

Running this simulation helped us a lot it allowed us to see how the signals behaved, test our logic, and make sure everything was working before applying it to the real circuit, the step was very useful, especially given the limited materials we had, and it gave us confidence that our control system would perform as expected once built.

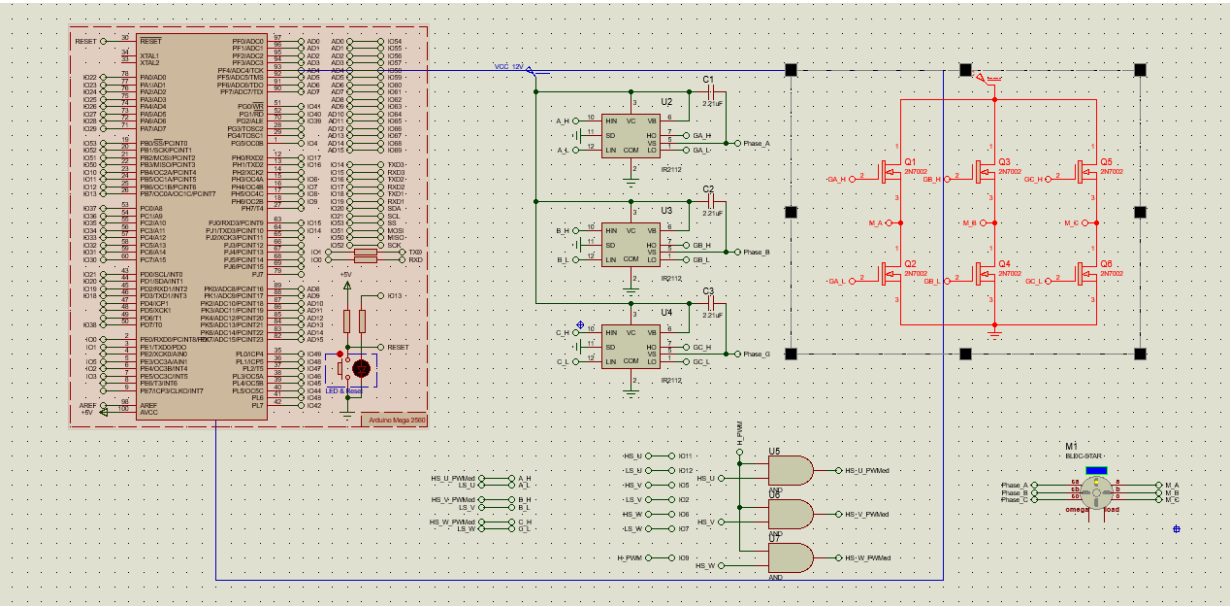


Fig (III.13): Proteus Simulation of the BLDC Motor Control System.

III.3 The Base circuit

This circuit diagram illustrates a setup for controlling a BLDC (Brushless DC) motor using an Arduino-based system. The circuit uses two Arduino boards:

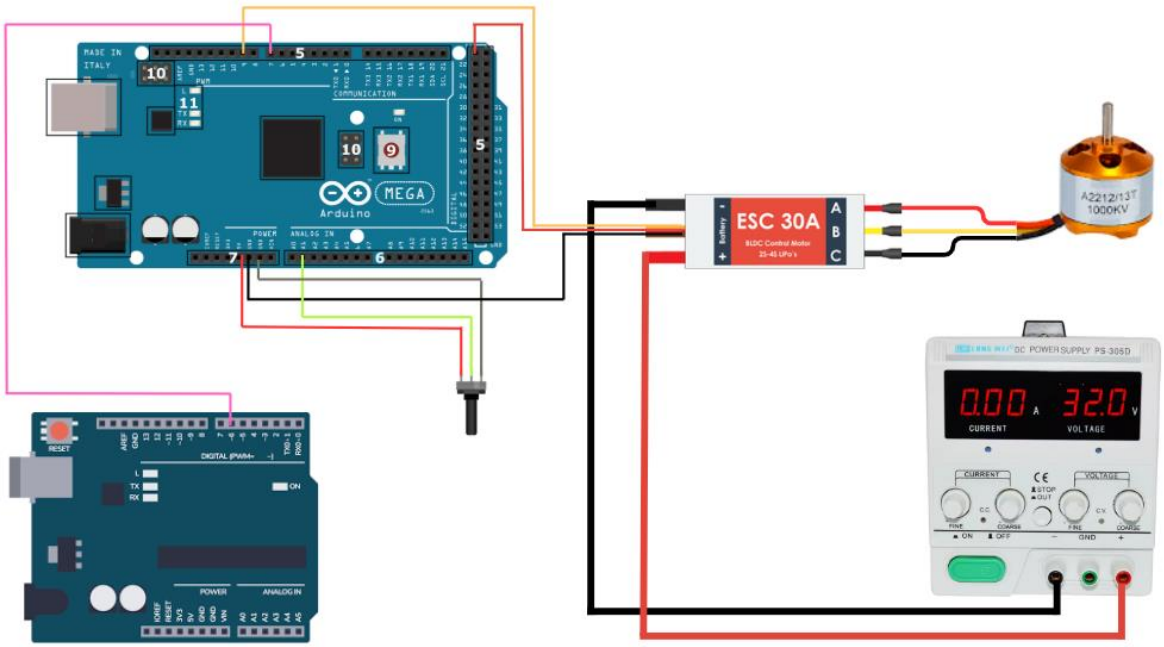


Fig (III.14): The Base circuit diagram.

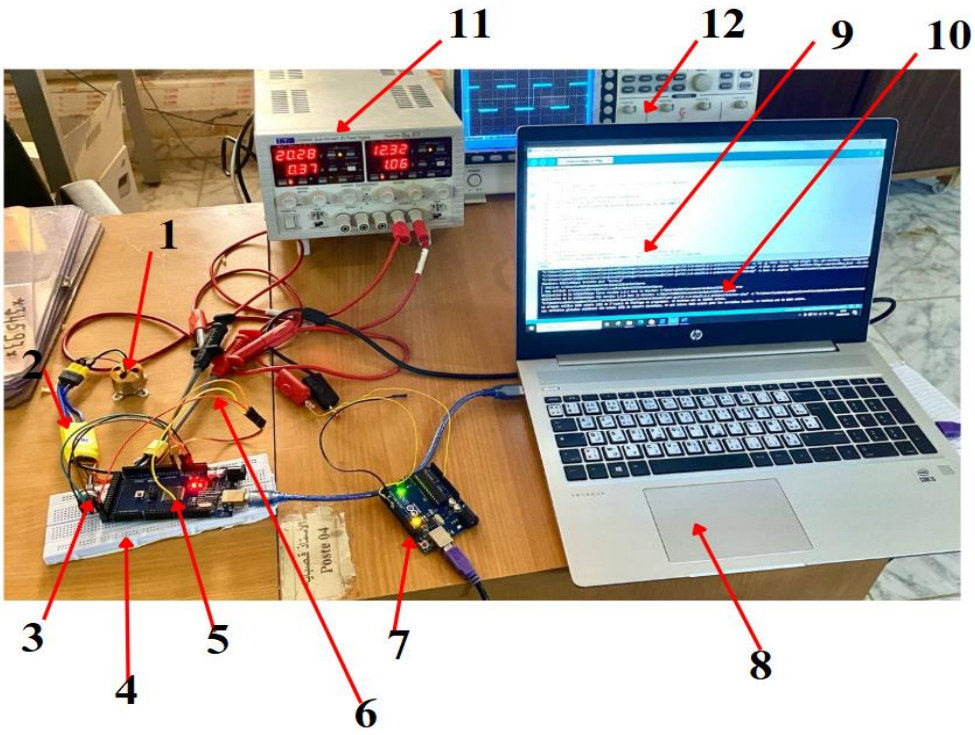
- Arduino Mega 2650: appears to serve as the primary controller, processes this input and generates a corresponding PWM signal through its digital pin 9, which were sent to the ESC to regulate motor speed and direction, while leveraging the board's computational power to process sensor feedback (or simulated signals) and maintain stable motor performance.
- Arduino UNO R3: Due to limited resources, we used the Arduino Uno to simulate a speed sensor for the BLDC motor, generating control signals to provide fictional speed feedback in the code (referred to as currentSpeed). In a real system, current Speed would represent the actual motor speed measured in RPM, RPS, or counts per second using sensors like Hall effect sensors, rotary encoders, optical sensors, or ESC telemetry. However, since we lacked these sensors, we programmed the Arduino Uno to act as a virtual sensor by running a loop every $20\ ms$, pin 6 is set HIGH for one second, then set LOW for one second, and finally set HIGH again, repeating this sequence continuously, allowing the PID controller to compare this simulated speed with the target setpoint and adjust the ESC signal accordingly.

- ESC 30A (Electronic Speed Controller): The ESC is connected between the Arduino and the BLDC motor. It receives PWM (Pulse Width Modulation) signals from the Arduino to regulate the speed and direction of the BLDC motor.
- BLDC Motor (A2212/13T/1000KV): The motor is powered and controlled via the ESC. The three wires from the ESC connect to the three terminals of the BLDC motor, enabling three-phase operation.
- DC Power Supply: The power supply provides the necessary voltage and current to the ESC and, consequently, the BLDC motor. The positive and negative terminals of the power supply are connected to the ESC's power input.
- Potentiometer: A potentiometer is connected to Analog Pin A0 on Arduino Mega 2560. It allows the user to vary the voltage signal sent to the Arduino, which is then used to adjust the PWM signal output to the ESC, thus controlling the motor speed.

III.4 Application results

This section presents our experimental project setup for controlling a BLDC motor using Arduino, with all components clearly labeled in the Fig (III.15). a breadboard, and connecting cables. A DC power supply delivers the required power, while a PC is used to upload and run the control code. An oscilloscope is used to monitor the output signals.

This configuration demonstrates the practical implementation of our control system and highlights the integration between hardware and software.



1- BLDC Motor
2- ESC
3- Potentiometer
4- Breadboard
5- Arduino Mega 2560
6- Cables
7-Arduino Uno R3
8- PC
9- Code of Arduino
10- Run of code
11- DC Power Supply
12- Oscilloscope

Fig (III.15): Photo of the assembled Set-up.

- This Fig (III.16) shows the oscilloscope display capturing the output waveform from Arduino Uno R3, which simulates the current speed feedback of a BLDC motor. Due to the absence of actual speed sensors such as Hall effect sensors or encoders, the Arduino Uno was programmed to generate a virtual speed signal. Specifically, pin 6 of the Arduino is toggled too HIGH for one second, then LOW for one second, and HIGH again repeating this loop every 20 ms to simulate a speed signal. The square waveform observed on the oscilloscope corresponds to this behavior, representing a basic on-off pulse that the PID controller uses as a reference for "current Speed." This artificial signal allows the control system to compare it with a target set point and modulate the ESC accordingly, emulating real-time speed regulation. Although it's a simplified simulation, this approach allows for effective testing of the control algorithm in the absence of physical speed sensing hardware.

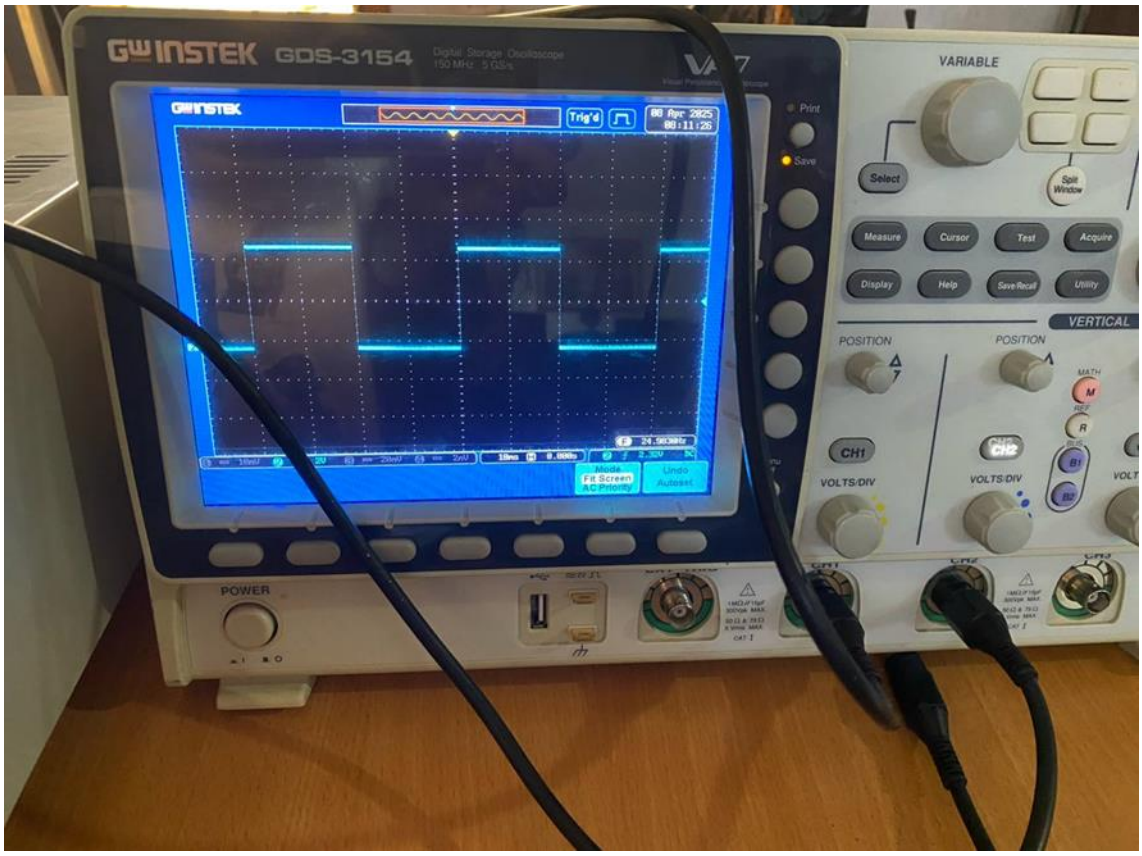


Fig (III.16): Observation of the virtual current speed sensor Signal.

Since we couldn't obtain the actual speed curve due to the absence of a physical speed sensor, we adopted an alternative approach to estimate the motor speed. By using the voltage applied to the motor and referring to the motor's datasheet, which provides the relationship between voltage and RPM, we can approximate the motor's rotational speed. This method allows us to infer the motor's performance and validate the control system behavior despite the lack of direct sensor feedback. It serves as a practical solution for estimating speed in sensorless experimental setups.

- BLDC motor speed estimation Based on Voltage-RPM Characteristics. We know that:

$$1V = 930KV \quad \text{(III.1)}$$

Thus,

$$930 * 12.32 = 11457{,}6\ RPM$$

We conclude that the estimated motor speed is: ***11457.6 RPM***

III.5 Interpretation and comments

Throughout the development of this project, several key observations were made that highlighted the importance of closed-loop control:

- After implementing the Arduino Uno loop, we observed effective speed regulation, simulating feedback in the absence of real sensors.
- In open-loop mode, the motor speed varied randomly and chaotically, with no consistency or stability. With the introduction of the closed-loop control using PID, the motor speed stabilized and fixed around the calculated reference value.
- During initial open-loop tests, there were noticeable fluctuations in current and voltage, affecting performance.
- Once the closed-loop was activated, both current and voltage values stabilized, aligning with the desired levels and improving overall system efficiency.

III.6 Conclusion

In conclusion, this project successfully bridged the gap between theoretical knowledge and practical application. By carefully selecting and integrating both hardware components such as the BLDC motor, ESC, and Arduino boards, and software tools like Arduino IDE and Proteus, we were able to build a fully functional control system. Although we faced significant challenges, particularly a lack of time and limited materials, our determination and problem-solving approach allowed us to overcome these obstacles. Ultimately, we achieved our main objective and gained valuable hands-on experience in system design, programming, and motor control.

Conclusion

This document offers a comprehensive study of brushless motors, organized into three main chapters. The first chapter introduces brushless motors by defining their key components and explaining their operating principles. It highlights their advantages over traditional motors, such as higher efficiency and longer lifespan, along with their broad industrial applications. The chapter also covers fundamental concepts, mathematical modeling, and simulation techniques, focusing on how differential equations describing motor behavior are implemented in MATLAB/Simulink. The role of the classic PID controllers, particularly those tuned via the MATLAB/APPS/System identification tool, is emphasized for effective speed control.

The second chapter explores the Harris Hawks Optimization (HHO) algorithm, a modern metaheuristic inspired by the hunting tactics of Harris hawks. This method was applied to optimize PID controller parameters for the motor system, demonstrating improved speed response, reduced speed drops under load, and minimized torque ripple compared to traditional PID control.

The third chapter focuses on a practical application of BLDC motors, showcasing the integration of microcontroller platforms like Arduino Mega 2560 with Proteus simulation software. This combination facilitates the implementation, simulation, and validation of BLDC motor control strategies, offering a versatile and accessible solution for both educational purposes and engineering development.

Additionally, this work can be expanded through further research aimed at improving BLDC motor control. Potential directions include:

- Experimental validation of the proposed control strategies.
- Applying the HHO optimization method to other BLDC motor control techniques.
- Exploring and adapting alternative optimization criteria.

Future work:

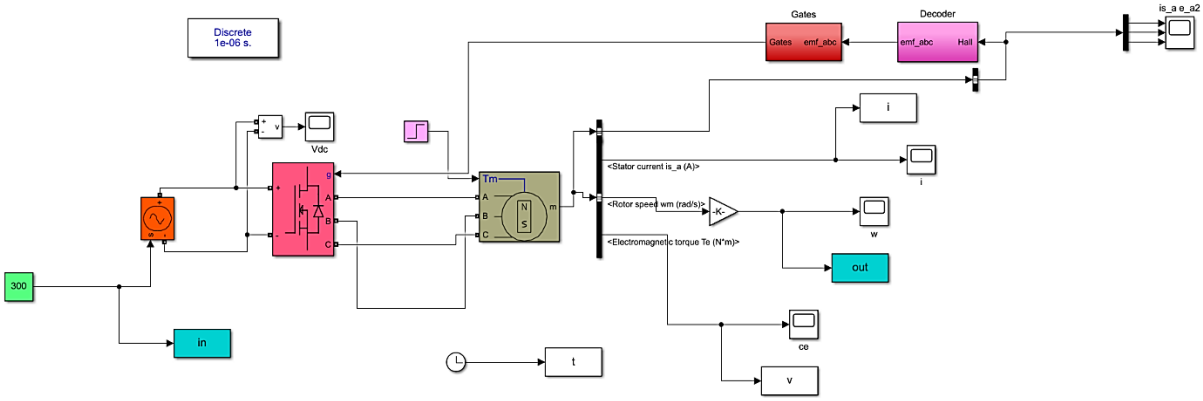
The research underscores the significant potential of HHO algorithms for BLDC motor speed control. Future investigations could further advance this area by:

- Genetic algorithm (GA) fused with HHO to optimize controller gains and reduce torque ripple.
- Hybrid HHO with PSO (Particle Swarm Optimization) or ACO (Ant Colony Optimization) for faster and better optimization.
- Self-tuning fuzzy controllers where HHO adjusts rules and membership functions.
- Adapting HHO-based control strategies for use in high-precision medical devices that utilize BLDC motors, such as robotic surgical tools or prosthetics.

➢ Annex A :

In the figure below there is a simulation of the BLDC motor in Open-loop for System identification APP.

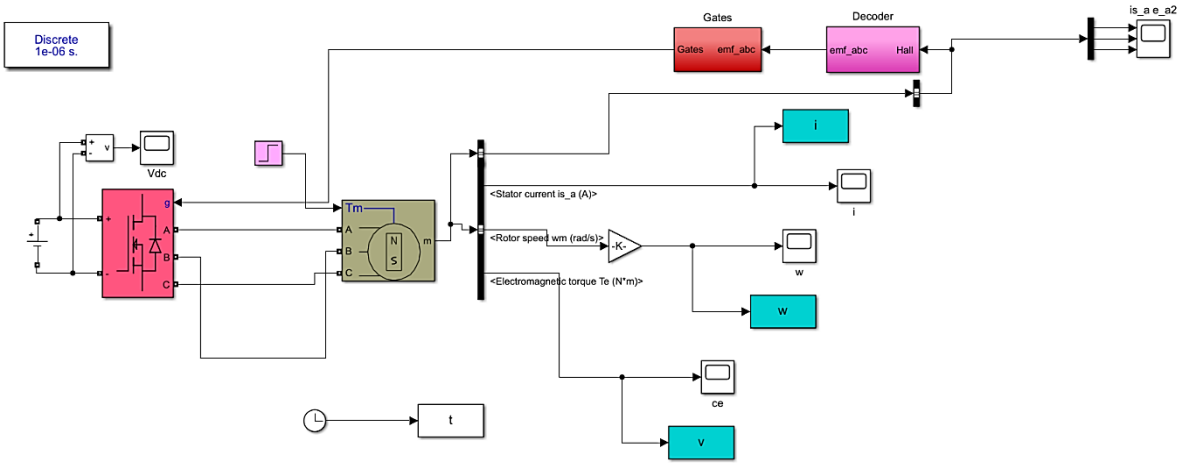
We created in MATLAB/Simulink an open loop simulation in the aim to estimate the system transfer function in MATLAB/APPS/System identification:



➢ Annex B :

In the figure below there is a simulation of the BLDC motor in Open-loop.

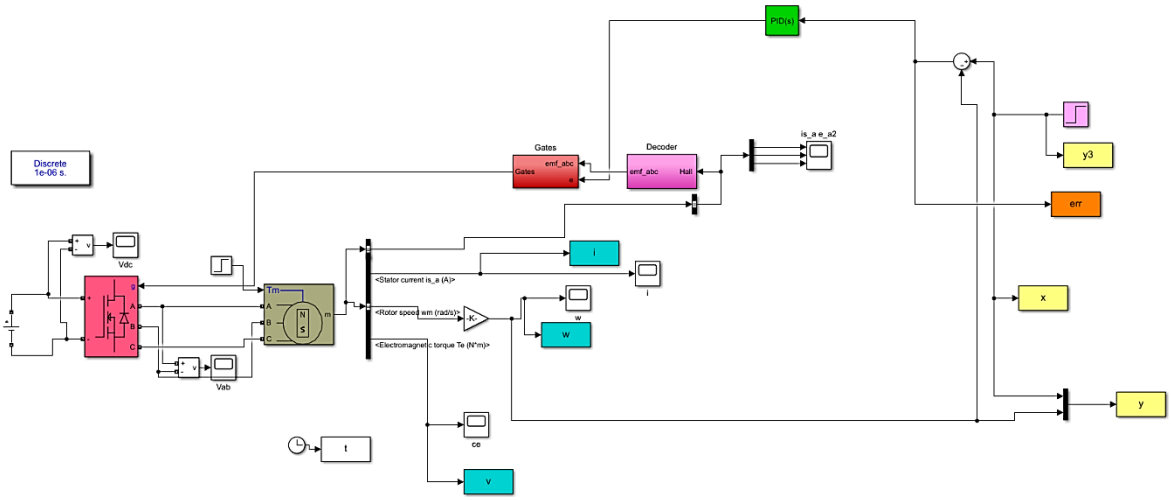
We created in MATLAB/Simulink an open loop simulation in the aim to see the system dynamic response:



- Annex C:

In the figure below there is a simulation of the BLDC motor in closed-loop.

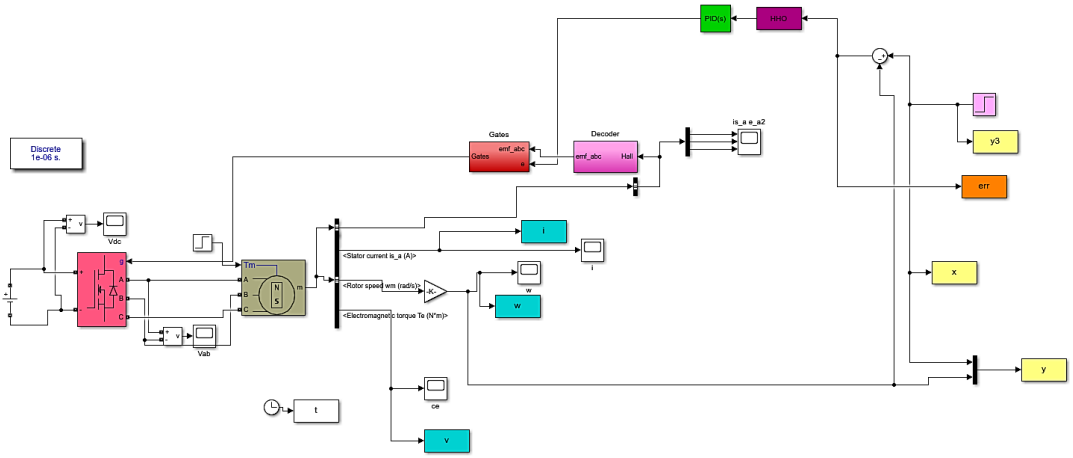
We created in MATLAB/Simulink a closed-loop simulation in the aim to control the speed of the BLDC motor using PID-ZN/PID-APPS:



- Annex D

In the figure below there is a simulation of the BLDC motor in closed-loop using HHO based controller.

We created in MATLAB/Simulink a closed-loop simulation in the aim to control the speed of the BLDC motor using HHO based PID controller (PID-HHO):



REFERENCES

[1] Z. E. Kribaa, *Regulation optimized of speed for the DC Motor by the HHO technique*, Master's thesis, Mohamed Khider Univ., Biskra, Algeria, 2023.

[2] M. Murali and R. Arulmozhiyal, "Investigation on modeling and simulation BLDC motor fed universal actuation system," *Métodos Numéricos para Cálculo y Diseño en Ingeniería: Revista Internacional*, vol. 37, no. 1, pp. 1–8, 2021.

[3] D. Mohanraj, R. Aruldavid, R. Verma, et al., "A review of BLDC motor: state of art, advanced control techniques, and applications," *IEEE Access*, vol. 10, pp. 54833–54869, 2022.

[4] S. Rambabu, *Modeling and Control of Brushless DC Motor*, M.S. thesis, Dept. Elect. Eng., National Institute of Technology, Rourkela, India, [no year specified].

[5] S. Usha, P. M. Dubey, R. Ramya, and M. V. Suganyadevi, "Performance enhancement of BLDC motor using PID controller," *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 3, pp. 1335–1344, Sep. 2021.

[6] ABLIC Inc., "ZCL Hall Effect ICs ideal for BLDC motors," [Online]. Available: https://www.ablic.com/en/semicon/applications/bldc-motor/. [Accessed: 25-May-2025].

[7] Raj Borkar, "Material selection for BLDC Motor Components: An Engineering Perspective," MECHTEX, Feb. 05, 2025. [Online]. Available: https://mechtex.com/blog/material-selection-for-bldc-motor-components-an-engineering-perspective/. [Accessed: 25-May-2025].

[8] Brushless.com, "What are the Components of Brushless DC Motor?" [Online]. Available: https://www.brushless.com/what-are-the-components-of-brushless-dc-motor. [Accessed: 25-May-2025].

[9] M. Yaz and E. Cetin, "Brushless Direct Current Motor Design and Analysis," *COJ Electronics & Communications*, vol. 2, no. 2, Sep. 2021.

[10] MECHTEX, "Types of BLDC Motors," [Online]. Available: https://mechtex.com/blog/types-of-bldc-motors. [Accessed: 25-May-2025].

[11] S. Fu, "Types of Brushless DC(BLDC) Motor," BeUDMKE, Oct. 29, 2024. [Online]. Available: https://beudmke.com/types-of-brushless-dc-motors/. [Accessed: 25-May-2025].

[12] SONTIAN, "Types of Brushless DC Motors," [Online]. Available: https://sontianmotor.com/types-of-brushless-dc-motors/. [Accessed: 25-May-2025].

[13] D. T. Toro, "Introduction to Brushless DC Motors," Diodes Incorporated, AN1164. [Online]. Available: https://www.diodes.com/assets/Uploads/AN1164-BLDC-Motors.pdf. [Accessed: 25-May-2025].

[14] C.-L. Xia, *Permanent Magnet Brushless DC Motor Drives and Controls*, John Wiley & Sons, 2012.

[15] RobotMotor, "The History of the Development of Brushless DC Motors," [Online]. Available: https://www.robotmotor.com/news/the-history-of-the-development-of-brushless-dc-motors/. [Accessed: 25-May-2025].

[16] M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam, and A. N. Nordin, "Control BLDC Motor Speed using PID Controller," Int. Islamic Univ. Malaysia, [unpublished].

[17] Magnetic Innovations, "The Direct Drive Motor Company," [Online]. Available: https://www.magneticinnovations.com/fr/. [Accessed: 25-May-2025].

[18] Changzhou Rosetta Stone, "Sheet metal stamping BLDC motor rotor stator core," [Online]. Available: https://www.czrosettastone.com/Lamination-Core/51.html. [Accessed: 25-May-2025].

[19] Google Images, "BLDC Motor Cross-section," [Online]. Available: https://images.app.goo.gl/yWGvGRwrxY41jnK7A. [Accessed: 25-May-2025].

[20] Amazon.fr, "BERWENNY Roulement à billes à gorge profonde," [Online]. Available: https://www.amazon.fr/BERWENNY-roulement-profonde-roulements-vitesse/dp/B0CC28WYHK. [Accessed: 25-May-2025].

[21] Google Images, "Brushless DC Motor Rotor," [Online]. Available: https://images.app.goo.gl/TZ95bEnKiXo2LnSX7. [Accessed: 25-May-2025].

[22] Gian Transmission, "A Comprehensive Guide to Brushless DC Motor," [Online]. Available: https://www.gian-transmission.com/a-comprehensive-guide-to-brushless-dc-motor/. [Accessed: 25-May-2025].

[23] Google Images, "BLDC Motor Diagram," [Online]. Available: https://images.app.goo.gl/skPuQ3Swn5CXePCR6. [Accessed: 25-May-2025].

[24] Proteshea, "Brushless DC Motor Control with ESC and Arduino Uno," [Online]. Available: https://proteshea.com/brushless-dc-motor-control-with-esc-and-arduino-uno/. [Accessed: 25-May-2025].

[25] RCDRONE, "ESC Anatomy: Understanding the Components," [Online]. Available: https://rcdrone.top/blogs/articles/esc-anatomy-understanding-the-components. [Accessed: 25-May-2025].

[26] D. Mohanraj et al., "A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications," *ResearchGate*, [Online]. Available: https://www.researchgate.net/publication/360585129. [Accessed: 25-May-2025].

[27] P. Abinaya, M. S. Preetha, and N. S. Harivignesh, *BLDC motor control using ESC and CCPM consistency master controller*, M.S. thesis.

[28] M. Kannan, "Electronic Speed Controllers – A Review," in *Proc. Int. Conf. Recent Adv. Eng. Technol.*, 2022. [Online]. Available: https://www.ijert.org/research/electronic-speed-controllers-a-review-IJERTCONV10IS11031.pdf. [Accessed: 25-May-2025].

[29] MathWorks, "Six-step commutation - MATLAB & Simulink," [Online]. Available: https://www.mathworks.com/help/mcb/ref/sixstepcommutation.html. [Accessed: 25-May-2025].

[30] MathWorks, "Get Started with Six-Step Commutation," [Online]. Available: https://www.mathworks.com/help/mcb/gs/six-step-commutation.html. [Accessed: 25-May-2025].

[31] F. Pangerang, F. A. Samman, Z. Zainuddin, and R. S. Sadjad, "Variable loaded brushless DC motor with six step commutation PID-based speed controller optimized by PSO algorithm," *Bull. Electr. Eng. Inform.*, vol. 14, no. 1, pp. 99–107, 2025, doi: 10.11591/eei.v14i1.8618.

[32] H. Hadeed and A. Gottscheber, "Six Step Control vs Direct Torque Control: Comparative evaluation for BLDC drive," in *Proc. 5th Int. Conf.*, 2017.

[33] S. Hong, M. Luo, and X. Wang, "Control of BLDC Motor Drive with Single Hall Sensor Considering Angle Compensation," *Power Electronics and Drives*, vol. 8, pp. 299–309, 2023, doi: 10.2478/pead-2023-0019.

[34] Y. Ling, Y.-H. Wang, and L.-L. Zhu, "Research of Zero-crossing detection for sensorless BLDC motor," in *Proc. MEIC*, 2015, doi: 10.2991/meic-15.2015.269.

[35] J. Ye and Z. Huang, "Design and Realization of Control System of Brushless DC Motor Based on ARM," *J. Phys.: Conf. Ser.*, vol. 1087, p. 042070, 2018, doi: 10.1088/1742-6596/1087/4/042070.

[36] Monolithic Power Systems, *Brushless DC Motor Fundamentals*. [Online]. Available: https://www.monolithicpower.com/media/document/Brushless_DC_Motor_Fundamentals.pdf. [Accessed: 25-May-2025].

[37] MathWorks, "Transfer Function - MATLAB & Simulink," [Online]. Available: https://www.mathworks.com/discovery/transfer-function.html. [Accessed: 25-May-2025].

[38] M. Nguyen, "BLDC Motor PID Controller," unpublished article, 2021.

[39] MathWorks, "System Identification App - MATLAB & Simulink," [Online]. Available: https://www.mathworks.com/help/ident/ref/systemidentification-app.html. [Accessed: 25-May-2025]

[40] A. Irawan, *et al.*, "Pneumatic servo position control optimization using adaptive-domain prescribed performance control with evolutionary mating algorithm," *Results in Control and Optimization*, vol. 15, p. 100434, 2024.

[41] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*, Athena Scientific, 2012.

[42] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[43] K. Sörensen and F. Glover, "Metaheuristics," *Encyclopedia of Operations Research and Management Science*, vol. 62, pp. 960–970, 2013.

[44] Z. Menacer, *Modeling and Control of BLDC Motor*, Master's thesis, Dept. of Electrical Engineering, Mohamed Khider Univ. 2024.

[45] M. Shehab, *et al.*, "Harris Hawks Optimization Algorithm: Variants and Applications," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5579–5603, Jul. 2022, doi: 10.1007/s11831-022-09780-1.

[46] W. K. Mashwani, R. Haider, and S. B. Belhaouari, "A multiswarm intelligence algorithm for expensive bound constrained optimization problems," *Complexity*, vol. 2021, Article ID 5521951, 2021.

[47] "Harris Hawks Optimization," *ScienceDirect*. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/harris-hawks-optimization. [Accessed: 25-May-2025].

[48] Aliasgharheidaricom, "Harris Hawks Optimization Algorithm and Applications," *GitHub*. [Online]. Available: https://github.com/aliasgharheidaricom/Harris-Hawks-Optimization-Algorithm-and-Applications. [Accessed: 25-May-2025].

[49] Z. E. Kribaa, *Regulation optimized of speed for the DC Motor by the HHO technique*, Master's thesis, Mohamed Khider Univ., Biskra, Algeria, 2023.

[50] A. A. Heidari, *et al.*, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[51] M. Abd Elaziz, D. Yousri, and S. Mirjalili, "A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics," *Advances in Engineering Software*, vol. 154, p. 102973, 2021.

[52] A. Abbasi, B. Firouzi, and P. Sendur, "On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks," *Engineering with Computers*, vol. 37, pp. 1409–1428, 2021.

[53] H. Chen, *et al.*, "multi-population differential evolution-assisted Harris hawks' optimization: Framework and case studies," *Future Generation Computer Systems*, vol. 111, pp. 175–198, 2020.

[54] T. Yang, J. Fang, C. Jia, Z. Liu, and Y. Liu, "An improved Harris Hawks Optimization algorithm based on chaotic sequence and opposite elite learning mechanism," *PLoS ONE*, vol. 18, no. 2, p. e0281636, 2023. doi: 10.1371/journal.pone.0281636.

[55] “10 Jumper Cables dupont Arduino Macho- M / Hembra-h / Macho - H,” *Electrosena*. [Online]. Available: https://www.electrosena.com/10-jumper-cables-arduino-macho-m-hembra-h-macho-h. [Accessed: 25-May-2025].

[56] R. Boylestad and L. Nashelsky, *Electronic Devices and Circuit Theory*, 11th ed., Pearson, 2012.

[57] “How to attach a Potentiometer to a micro:bit,” *TeachComputing*. [Online]. Available: https://www.teachwithict.com/potentiometer.html. [Accessed: 25-May-2025].

[58] L. Udawat, H. Rodage, C. Raut, M. Waghade, D. Pawar, S. Shekhdar, and P. Kadam, “DC Variable Power Supply: Review Paper,” Jagadambha College of Engineering and Technology, Yavatmal, India.

[59] S. Hussain, “Automated Color Recognition System for Visually Challenged and Achromatopsia People using Arduino and Mobile App,” *Int. J. Eng. Res. Technol.*, vol. 4, 2015.

[60] Shenzhen Bodunbaili Electronic Co., Ltd., “MW MPD-3303 30V 3A Affichage Numérique Multicanal Alimentation Réglable,” *Alibaba*. [Online]. Available: https://french.alibaba.com/product-detail/MW-MPD-3303-30V-3A-Multi-1600698663967.html. [Accessed: 25-May-2025].

[61] Arduino, “Arduino Mega 2560 Rev3,” 2021. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega2560. [Accessed: 25-May-2025].

[62] CircuitDigest, “Different Types of Arduino Boards – Quick Comparison on Specification & Features,” Apr. 22, 2022. [Online]. Available: https://circuitdigest.com/. [Accessed: 25-May-2025].

[63] F. Aswin, I. Dwisaputra, and R. Afriansyah, “Online vibration monitoring system for rotating machinery based on 3-axis MEMS accelerometer,” *J. Phys.: Conf. Ser.*, vol. 1450, p. 012109, 2020, doi: 10.1088/1742-6596/1450/1/012109.

[64] D. J. Bates, *Practical Electronics for Inventors*, 4th ed., McGraw-Hill Education, 2016.

[65] Adafruit Industries, “Full sized premium breadboard - 830 Tie points.” [Online]. Available: https://www.adafruit.com/product/239. [Accessed: 25-May-2025].

[66] J. W. Dixon and A. L. K. Yung, *Fundamentals of Electric Drives*, 2nd ed., Springer, 2010.

[67] "RC Electric Parts 30A Brushless Motor Speed Controller ESC BEC with XT60 & 3.5mm bullet plugs," *Amazon.sa*. [Online]. Available: https://www.amazon.sa/-/en/RC-Electric-Parts-Brushless-Controller/dp/B071GRSFBD?th=1. [Accessed: 25-May-2025].

[68] M. R. Zolghadri, "Brushless DC Motors and Controllers," 2018. [Online]. Available: https://www.sciencedirect.com/topics/engineering/brushless-dc-motor. [Accessed: 25-May-2025].

[69] DSM Electro, "Brushless DC Motor A2212/15T 930KV RacerStar Compatible." [Online]. Available: https://www.dsmonline.in/product/brushless-dc-motor-a221215t-930kv-racerstar-compatible-khbzg. [Accessed: 25-May-2025].

[70] Labcenter Electronics, *Proteus Design Suite*, Version 8.17, 2021. [Online]. Available: https://www.labcenter.com/. [Accessed: 25-May-2025].

[71] Arduino, "Arduino IDE for Software Development," 2021. [Online]. Available: https://www.arduino.cc/en/software. [Accessed: 25-May-2025].

[72] Ee-Diary, "Download Proteus Professional V8.17 SP2 free, a Must-Have tool for electronics enthusiasts," *ee-diary*, Jul. 09, 2024. [Online]. Available: https://www.ee-diary.com/2024/04/download-proteus-professional-v817-sp2.html. [Accessed: 25-May-2025].