الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة العربي التبسي – تبسة

Larbi Tébessi University -Tébessa

Faculty of Sciences and Technology

Departement of electrical engineering

Thesis

Presented for obtaining the academic master's degree in

Automatic

Option: Automatic and Systems

Candidates:
Mr. GHOUL Mounir
Mr. AIMENE Mohammed

Supervisor: Dr. AMIEUR Toufik

Modelling and Fractional Order Control of DC motor

Presented and supported publicly, in 11/06/2022 in front of the jury composed of:

Dr. YOUSFI Laatra	MCA	President
Dr. AMIEUR Toufik	MCA	Supervisor
Dr. OUNNAS Djamel	MCA	Examiner

Promotion: 2021/2022

Dedication

I am very pleased to dedicate this thesis: to my dear father ***Ghoul Alaoua****, may God have mercy on him, to my mother* ***Benmhania Fatima****, who did not spare me her prayers and constant encouragement, with all my gratitude to all my dear brothers* ***"Ahmed, Abderraouf*** *and* ***Haithem"****, to my dear sisters and my brothers' wives, to my nephews* ***"****Ritadj, Ibrahim, Sara, Youssef, Mouhamed, Ali, Roudaina, Razane, Yassin and Abd Errahime****"****, to all members of my large family especially* ***"*** *Hazhouz Mourad, Harmal Fawzi, Boukrine Hassan, Grid Mouhsen, Benmhania Azzeddine, Belakehal Aymen****"****, to all my friends, especially:* ***"****Sadi kais, Amrane Naceur, Fetni Ibrahim, Djabali Salem, Benmhania Ziad, Hadhoud Zakaria, Necib Imed, Abaidia Haithem,, Benzine Zakaria, Benmhania Hilal, Benmhania Nabil, Benmhania Abdelaziz, Maifi Abd ennour, Maifi Achref, Karkoub Amin, Beghil Najmeddine, Maalem Haithem, Gassoumi Ayoub, Daani Yahia, Tebbassi Baha, Labidi Abdelmadgid, Kafi Redouane, Labidi Walid, Lamouchi Aymen, Gastel Aymen, Zitoun Abdelouahed, Benmhania Fares, Benmhania Soufiane, Ghouzlane Yasser, Ghouzlane Samir, Trad Rafik, Khadhraoui Ayoub, Aimene Imed, Soultani Hama, Zerguine Saif, Bougrouz M. Elmotassim, Fares Haithem, Bouchoucha Mouhamed Bachir, baji Oussama, Oumairi Nawfel, Chebira Abdelkarim, Zarai Fathi, Zaouai Mouhammed, Brahmi Ayoub, Hail Soufiane, Hail Ziad, Orgdi Ilyasse, Orgdi Saif, Orgdi Jijo Chatouh Taleb, Ghouzlane Saif, Ghouzlane M. Ziad, Ghouzlane Adham, Ghouzlane Ayoub, Ghouzlane Mabrouk, Hadhoud Adlene, Chatouh Akrem, Ghouzlane Faiz****"****, to all my relatives* ***"****Ghoul Yahia, Ghoul Raid, Ghoul louai, Ghoul Maher, Ghoul Naceur, Ghoul Kaddour, Tria Fares, Tria Saif, Tria Abdelkader, Mansouri Chadi, Ghoul Adem, Ghoul Samir, Ghoul Fouad and Ghoul djihad****"*** *and all the family GHOUL especially* ***"****Achour, Mouhamed, Saddam, Adel, Fawzi, Maamar, Salah, Houssem, , Mourad, Ramy, Djalel, Khalifa, Oussama, Fares, Yassin, Khaled, Issam, Hamza, Islam, Nasereddine, Imad****"*** *, to all the* ***"****Automatic and systems promotion 2021/2022****"*** *classmates, and to my companion in thesis* ***"Aimene Mohammed"****.*

Dedication

I am very pleased to dedicate this thesis: To my father ***Aimene Laabidi*** *and my mother* ***Ferd Bahja*** *who never stopped encouraging me throughout my studies to make it to this day. They are dearest to my heart, with all my gratitude to all my dear brothers* ***"Hodheifa, Choaib and Dhirar"****, to my dear sisters and my beautiful angel, and to all members of my large family, my grandfather, grandmother, uncles and aunts, to all my relatives and cousins especially:* ***"****A. Imade, A. Djemoui, A. kader, A. Walid, A. Saddam, A. Wael, A. Aniss, A. Saif, A. Saber, A. Halim, A. Mounir, F. Khalifa, F. Nouar, M. Zakaria****"****, to all my friends, especially:* ***"****A. Naceur, F. Ibrahim, S. Kais, S. Hama, A. Amara, B. Mansour, Y. Laiche, M. Majed, L. Madjid, A. Issal, D. Salem, T. Rafik, K. Ayoub, Z. Saif, M. Abdelouahab, S. Nasro****"****. to all the* ***"****Automatic and systems****"*** *classmates and all the promotion 2021/2022. to everyone I know, To my brother and companion in this work* ***"Ghoul Mounir"****.*

Acknowledgement

At the end of this work, we would like to thank all the people who contributed to the realization of this thesis. We extend special thanks to **Mr. Amieur Toufik** *for suggesting this work, following up and directing it. We would like to express our sincere gratitude to him for his relevant advice, constructive guidance and the valuable time he has devoted to us. Without forgetting* ***Mr. Jamal Taibi****, and we also extend a special thanks to* Brother **Ibrahim Ayeb.** *To all my friends and all the people who helped us make this work happen. Finally, we would like to thank all the doctors like:* ***Yousfi, Ounnas, Thelaijia, Djari, Boukadoum, Bougerne, Lemita, metatla, guiza*** *.... etc, who followed us throughout our university journey.*

Figures List :

Chapter I:

Description and modelling of DC motor

Chapter II:

Fundamentals of fractional calculus and systems

Chapter III:
Fractional order controller

Chapter IV:
Tuning of fractional order PID controller by PSO algorithm for DC motor speed regulation

Tables List :

Abbreviations list :

DC *: direct current*

PMDC*: permanent magnet*

R-L *: Riemann-Liouville*

G-L *: Grünwald-Letnikov*

LTI *: linear time-invariant*

FO *: fractional order*

PID *: proportional integrator derivative*

FOPID *: fractional order proportional integrator derivative*

PSO *: particle swarm optimisation*

GA *: genetic algorithm*

ISE *: integral of square error*

IAE *: integral of absolute error*

ITSE *: integral of time square error*

ITAE *: integral of time absolute error*

Abstract :

The aim of this work is to modelling a DC motor and control it by a fractional order PID controller, in order to improve the performance of this type of motors, tracking the reference speed signal, minimisation of measurement noise and rejection of torque load effect. To achieve this goal, the five controller parameters have been optimized by the metaheuristic optimization algorithm PSO (Particle Swarm Optimization), which ensures to find the optimal position of the parameters (particles) within the search space. Based on the minimisation of the objective function which is the integral of time absolute error $(ITAE)$. The simulations carried out show that the planning strategy used gives promising results considering the parameters of the fractional order FOPID controller. A comparison between the performance of classical PID controller and fractional order PID controller demonstrate the superiority of the latter applied to the speed regulation of the DC motor.

Keywords :

DC motor. Fractional calculus. Fractional systems. Classical PID Controller. Fractional Order PID controller. Particle swarm optimization. Integral of time absolute error.

Résumé :

Le but de ce travail est de modéliser un moteur à courant continu et de contrôler par un contrôleur PID fractionnaire, Afin d'améliorer les performances de ce type de moteur, suivi du signal de référence de minimiser le bruit de mesure et d'annuler l'effet de couple résistant. Pour atteindre ce but, les cinq paramètres de régulateur $FOPID$ ont étaient optimisés par l'algorithme d'optimisation métaheuristique PSO (Optimisation des Essaims de Particules), ce dernier assure de découvert la position optimale des paramètres(particules) à l'intérieur de l'espace de recherche. Baser sur la minimisation de la fonction objectif, qui est intégrale du temps d'erreur absolue ($ITAE$). Les simulations effectuées montrent que la stratégie de planification utilisée donne des résultats prometteurs considérant les paramètres du contrôleur fractionnaire $FOPID$. Une comparaison entre les performances des contrôleurs PID et $FOPID$ fractionnaire nous démontre la supériorité de ce dernier appliqué à une régulation de vitesse d'un moteur à courant continu.

Mots clés :

Moteur à courant continu. Calcul fractionnaire. Systèmes fractionnaires. Contrôleur PID classique. Contrôleur PID d'ordre fractionnaire. Optimisation des essaims de particules. Intégrale de l'erreur absolue de temps.

الملخص :

الهدف من هذا العمل هو نمذجة محرك التيار المستمر (DC motor) والتحكم فيه بواسطة المتحكم PID الجزئي، وذلك من اجل تحسين أداء هذا النوع من المحركات، تتبع إشارة السرعة المرجعية, تقليل ضوضاء القياس وإلغاء تأثير عزم الحمل. لتحقيق هذا الهدف، تم تحسين معاملات المتحكم $FOPID$ الخمس بواسطة خوارزمية التحسين (PSO)، والتي تضمن ايجاد المعاملات (الجسيمات) المحسنة داخل فضاء البحث. بناءا على تصغير دالة الهدف وهي الخطأ المطلق الضمني مع الوقت $(ITAE)$.تظهر عملية المحاكاة التي تم اجراؤها ان استراتيجية التحسين تعطي نتائج واعدة مع الاخذ بعين الاعتبار اعدادات جهاز التحكم الجزئي $FOPID$. المقارنة بين المتحكمين PID و FOPID المحسنين أثبتت فعالية هذا الأخير في التحكم في سرعة محرك التيار المستمر.

الكلمات المفتاحية :

محرك التيار المستمر. حساب التفاضل والتكامل الكسري. أنظمة كسور. المتحكم PID الكلاسيكي. المتحكم PID الجزئي. تحسين سرب الجسيمات. تكامل الوقت في الخطأ المطلق.

Contents:

Chapter I:

Description and modelling of DC motor

Chapter II:

Fundamentals of fractional calculus and systems

Chapter III:

Fractional order controller

Chapter IV:

Tuning of fractional order PID controller by PSO algorithm for DC motor speed regulation

GENERAL INTRODUCTION :

An industrial motor is an electric motor to modify electricity into mechanical energy. Motors produce forces, including linear or rotary. They are typically powered by alternating current (AC) resources such as power grids or generators; however, some of them may be supplied by direct current (DC) resources like batteries.

The DC motor, by its very nature, has a high torque vs. speed characteristic, enabling it to deal with high resistive torques and absorb sudden rises in load effortlessly; the motor speed adapts to the load. In addition, DC motors are an ideal way of achieving the miniaturization that is so desirable to designers, since they offer a high efficiency as compared with other technologies.

A mathematical model is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed mathematical modelling. Mathematical models are used in the natural sciences (such as physics, biology, earth science, chemistry) and engineering disciplines (such as computer science, electrical engineering), as well as in non-physical systems such as the social sciences (such as economics, psychology, sociology, political science). A model help to explain a system and to study the effects of different components, and to make predictions about behaviour. [1]

For many decades, proportional - integral - derivative (PID) controllers have been the most widely used technique in industry process control, The main reasons for its widespread acceptance in the industry are due to its ability to control the majority of processes, well-defined actions and relatively simple implementation. Their merit consists in simplicity of design and good performance, such as low percentage overshoot and small settling time (which is essential for slow industrial processes)., there are several techniques for tuning the parameters of the PID controller this technique remains less effective in the case of complex and nonlinear systems.

Fractional calculus is a branch of mathematical analysis that studies the possibility of taking real number powers or complex number powers of the differentiation operator. Generally speaking, we know the n the derivative of y with regard to x. What does it mean if we take n to a fractional number? This is a very important and interesting question asked by many mathematicians. In the history, fractional calculus has long been a pure theoretical problem. However, in the recent years, calculus has been successfully applied to many areas such as automatic control and signal processing [2.3]. Despite the applications of fractional calculus, it is hard to develop numerical methods for fractional derivatives due to its complex definitions.

Fractional-order proportional-integral-derivative $(FOPID)$ controllers have received a considerable attention in the last years both from academic and industrial point of view. In fact, in principle, they provide more flexibility in the controller design, with respect to the standard PID controllers, because they have five parameters to select (instead of three). However, this also implies that the tuning of the controller can be much more complex. In order to address this problem, different methods for the design of a $FOPID$ controller have been proposed in the literature. However, the implementation of a $FOPID$ has been generally made via an appropriate approximation of those fractional-orders, namely Oustaloup, Matsuda, Carlson, or via the so-called exact analytical formula (i.e., without any approximation) [4.5].

So, the objective of this work is to modelling an DC motor and control it by a fractional order PID, in order to improve the performance of this type of motor, tracking the reference speed signal, minimisation of measurement noise and rejection of some external influences such as the torque load and internal influences such as the parametric variation.

In computational science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its local best-known position, but is also guided toward the best-known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions [6].

In this work, the metaheuristic optimization algorithm (Particle Swarm Optimization) is used to determine the optimal parameters of the $FOPID$ controller. The process based on the minimization of objective function which is the integral of time absolute error.

This work will be organized into four-chapter structure as follows:

- ***Chapter I :***

This chapter is devoted to talking about a DC motor, and in this context, it has been divided into two parts. The first part dealt with the basics about a DC motor. It's construction and classification. As for the content of the second part, it talked about the mathematical model, state space and transfer function of DC motor.

- ***Chapter II :***

This chapter take Recalls some of the definitions of Fractional calculus and fractional order operators, some main properties and also the LaPlace transform of derivatives and integrals of fractional order. Part is also assigned to present some methods of numerical approximation of fractional order operators.

- ***Chapter III :***

This chapter is devoted to introducing the PID and $FOPID$ controllers, as it included in its content a historical background for both, with the definition of the common types of these controllers, were also presented some algorithms and methods of tuning and the optimization of their settings, for example, we mentioned (ZN) method and (PSO) algorithm. After that, a comparison been made between Classical PID Controller and Fractional Order $PI^{\lambda}D^{\mu}$ Controller**.** In the end, the Performance criterion was defined and some of its common types were presented، from which we mentioned the following: $ITAE, IAE, ISE, ITSE.$

- ***Chapter IV :***

We studied and track the speed of a DC motor in a closed loop with the $FOPID.$ And to verify the effectiveness of the $FOPID$ controller, we tracked its effect and its ability to change the DC motor speed and the accuracy of its tracking for different speeds (constant signal, triangular signal, rectangular signal and sinusoidal signal). After that, We studied the ability of the controller to eliminate the external influences(torque load), internal influences (parametric variation), and minimization of the measurement noise, Finally, to prove the superiority of the $FOPID$ controller over the PID controller in accuracy, efficiency, robustesse and effectiveness, we made a comparison in terms of accuracy of tracking speed and the ability to rejection of the external and the internal influences.

Chapter I :

Description and modelling of DC motor

1. Introduction :

Electric motors are the driving force for many industrial applications. Therefore, it is important to study the performance and characteristics of these motors so that they can be used best according to the nature of the load. DC Machines are one of the most important types as they are widely used in all industries because of their ease of control and high torque, especially when starting movement. Also, they are one of the most important machines in the most control systems such as electrical systems in home electrical, vehicles and trains [7].

A DC machine is an electromechanical energy alteration device. The working principle of a DC machine is when electric current flows through a coil within a magnetic field, and then the magnetic force generates a torque that rotates the dc motor. The DC machines are classified into two types such as DC generator as well as DC motor [7].

The DC motor uses electricity and a magnetic field to produce torque, which causes it to turn. It requires two magnets of opposite polarity and an electric coil, which acts as an electromagnet. The repellent and attractive electromagnetic forces of the magnets provide the torque that causes the motor to turn. It also consists of one set of coils, called armature winding, inside a set of permanent magnets, called the stator. Applying a voltage to the coils produces a torque in the armature, resulting in motion. Basically, there are three types of DC motors. The two commonly criteria usually use in classifying them are their characteristics and the connection of their exciting windings or circuits. Based on these criteria, the three common types are: PMDC motor, Separately-existed DC motor and self-existed DC motor (shunt, series and compound motors).

It is well known that the mathematical model is very crucial for a control system design. For a DC motor, there are many models to represent the machine behaviour with a good accuracy. However, the parameters of the model are also important because the mathematical model cannot provide a correct behaviour without correct parameters in the model.

Our study in this chapter is about the DC motor, in this context, this chapter of the research is devoted to talking about its description and modelling. Where we will divide this chapter into two parts, in the first part we will talk about the construction of the DC Motor and its classification, in the second part we will talk about the mathematical model, stat space and transfer function of DC motor.

2. Construction of a DC Motor :

The DC Motor consists of two parts: One part is rotating, called rotor and the other part is stationary, called stator [8].

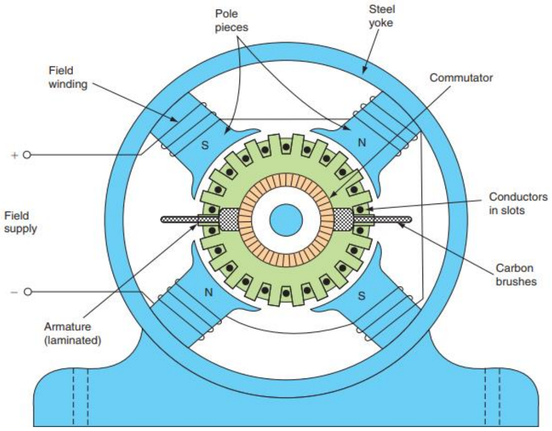


Figure I.01: *Cross-section of DC Motor***.**

The major components of a DC Motor are:

2.1. Magnetic Frame or yoke :

It is the stationary part of the Motor in the shape of hollow cylinder. Poles are fixed at the inner periphery of the yoke.

It acts as the outer cover or frame for the entire Motor and serves two main purposes: It is used to carry the magnetic flux produced by the poles. It acts as mechanical support for the Motor.

Yoke is usually made of cast iron for small Motor, because of its cheapness. But for large Motors, it is made of cast steel or rolled steel, due to its high permeability.

The lifting eye, feet and the terminal box are welded to the frame afterwards [8].

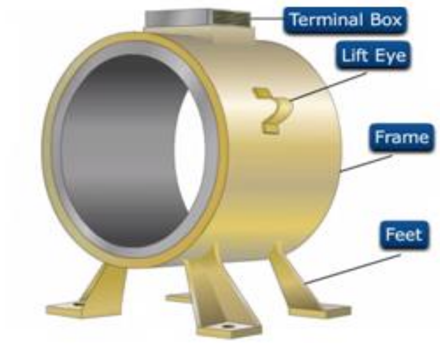


Figure I.02: *Yoke of DC Motor*.

2.2. Pole Cores and Pole Shoes :

Pole cores are the projecting rectangular parts, which produce magnetic flux needed for the Motor, when it is excited by the field winding. It is fitted to the yoke or frame by means of bolts and nuts or rivets.

The pole shoes are located at the end of pole core. The purpose of providing pole shoe in the poles is to make the magnetic field uniform on the surface of the armature.

The main function of the poles is acts as a mechanical support to the field coil, they reduce the reluctance of the magnetic path and they guide and spread out the flux in the air gap [8].

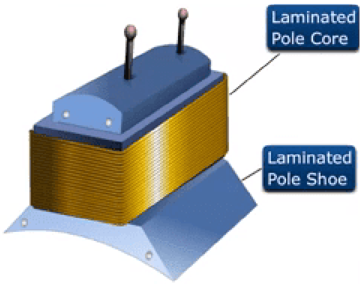


Figure I.03: *Pole Cores and Pole Shoes****.***

2.3. Field coil or winding :

Field coil is made up of copper. They are mounted on the pole core and carry the dc current. The field coils are connected in such a way that adjacent poles have opposite polarity. When the coils carry dc current, the pole core become an electromagnet and produces the magnetic flux. The magnetic flux passes through the pole core, the air gap, the armature and the yoke. The number of poles in a DC Motor depends on the speed of the Motor and the output for which the Motor is designed. There are several field constructions are adopted according to the type of excitation. In shunt field, a greater number of turns with small cross sectional are used, in series field only a few turns of large cross-sectional area are used and in compound field, both shunt and series field winding are used [8].

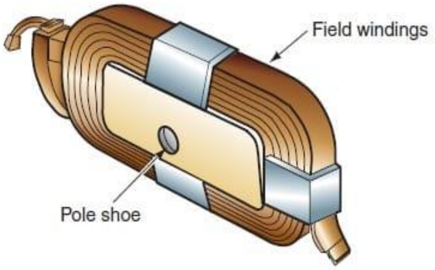


Figure I.04: *Field coil.*

2.4. Armature core and winding :

Armature core is designed as the rotating part and is built in cylindrical or drum shape with slots on its outer periphery. The purpose of armature is to house the winding and to rotate the conductors in the uniform magnetic field. It is mounted on the shaft.

It is build-up of steel lamination which are insulated by each other by thin paper or thin coating of varnish as insulation. The thickness of each lamination is about 0.5 mm. These laminations will reduce the eddy current loss. If silicon sheet is used for armature core, the hysteresis loss will also reduce.

The armature winding or coil is placed on slots available on the armature's outer periphery. The ends of the coils are joined with commutator segments. Insulated higher conductivity copper wire is used for making the coils. There are two types of winding. lap winding is used for high current, low voltage Motors and Wave winding is used for high voltage, low current Motors [9].

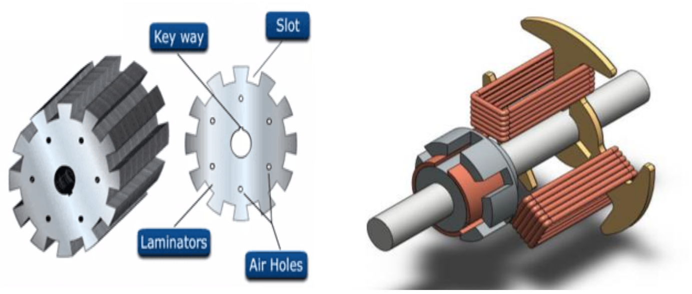


Figure I.05: *Armature core and Armature winding.*

2.5. Commutator :

The commutator provides the electrical connection between the rotating armature coil and the stationary external circuit. It is essentially a cylindrical structure and is built up of wedge-shaped copper segments insulated from each other by mica sheets and mounted on the shaft of the Motor.

The commutator is a mechanical rectifier which converts the alternating emf Motor in the armature winding into direct voltage across the brushes. The ends of the armature coil or winding are connected to commutator segments [9].

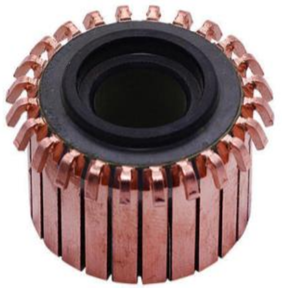


Figure I.06: *Commutator**.***

2.6. Brushes :

The function of brush is to collect the current from the commutator and supply it to the external load circuit. The brushes are manufactured in a variety of compositions to suit the commutation requirements. It is made of carbon, graphite metal graphite or copper and is rectangular in shape.

The brushes are placed in the brush holders which is mounted on rocker arm. The brushes are arranged in rocker arm in such a way that, it touches the commutator.

Brush pressure is adjusted by means of adjustable springs. If the bush pressure is high, the friction produces heating of the commutator and the brushes. If the pressure is too weak, the imperfect contact with the commutator may produce spark [9].

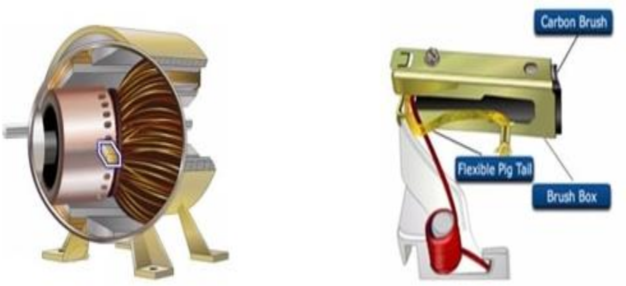


Figure I.07: *Brushes.*

2.7. Bearings and Shaft :

For construction of smaller DC Motor, ball bearings are used at both the ends of the shaft but for larger Motors, roller bearings are used at the driving end and ball bearings are used at the non-driving end of the Motor.

The shaft is made up of mild steel having maximum breaking strength. It is used to transfer the mechanical power from or to the Motor. All the rotating parts including the armature core, commutator, cooling parts and mounted and keyed to the shaft [9].

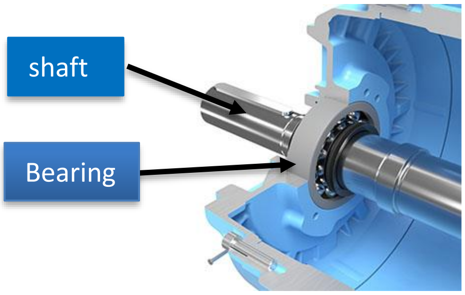


Figure I.08: *Bearings and shaft.*

3. Classification of a DC Motor :

The DC motors are classified as shown in (Figure I.9), the classification based on the method in which the field windings are excited [8]:

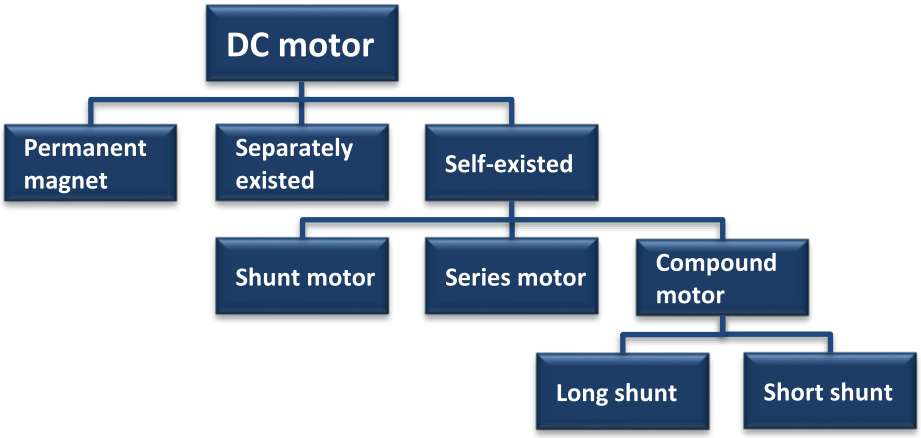


Figure I.09: *Types of DC motor.*

3.1. Permanent Magnet DC Motor :

As the name indicates, the motor has a permanent magnet on the inner periphery of the stator. The necessary magnetic field required to produce the rotating torque is developed by the set of permanent magnets [10].

The magnets are radially magnetized inside the yoke. The stator is used to carry the magnetic field produced by the magnets. The rotor consists of armature core, armature winding, commutator and brush arrangements to carry the current into the Motor [10].

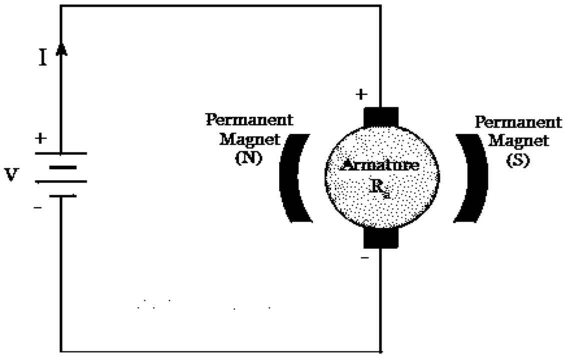


Figure I.10: *The circuit of PMDC Motor.*

Because of the absence of field winding, the PMDC Motor has more advantages. It does not require an excitation current for field winding, which reduces the loss. PMDC Motors are smaller in size and it is cheaper in cost [10].

Magnetic material like samarium cobalt, neodymium magnets having high residual flux and high coercively is used as a permanent magnet. This will greatly reduce the demagnetization of permanent magnets due to excessive armature current [10].

PMDC Motors are used extensively in automobiles for wipers, washers, blowers, air conditioners. It is also used in disc drives in personal computers [10].

3.2. Separately Excited DC Motor :

In a separately excited dc Motor, a separate DC supply is given to both the field winding and armature winding. Both are electrically isolated and have separate voltage ratings [11].

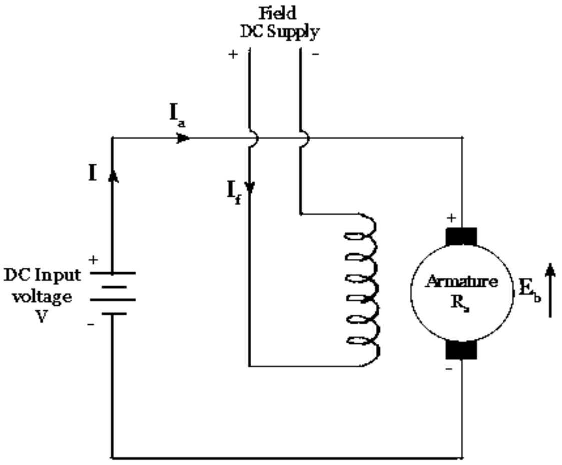


Figure I.11: *The circuit of separately excited DC Motor.*

From the circuit: Armature current (I_a) = Line current (I)

Back emf developed:

$$E_b = V + I_a R_a \qquad (I.1)$$

Power drawn from the supply:

$$P = VI = VI_a \qquad (I.2)$$

Mechanical Power developed:

$$P_m = Input\ power\ –\ Losses\ in\ the\ armature$$

$$P_m = VI_a - I_a^2 R_a = I_a(V - I_a R_a) = E_b I_a \qquad (I.3)$$

3.3. Self-Excited DC Motors :

The field winding is energized by the residual magnetism and then by the emf induced in the Motor. Because of the residual flux, the Motor starts to rotate.

The self-excited Motors are classified into three types, based on the way of connecting the field and armature winding: Shunt Motor, Series Motor and compound Motor [12].

3.3.1. DC Shunt Motor :

The word ‘shunt’ means ‘parallel’. It is because the field winding is connected in parallel with the armature winding. The field winding has a greater number of turns with thin wire to provide high resistance. Thus, the field current is much less compared to the armature current [12].

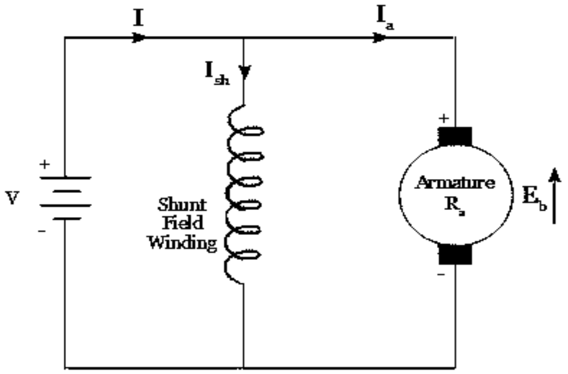


Figure I.12: *The circuit of DC Shunt Motor.*

From the circuit the current and voltage equations are:
Input current:

$$I = I_a + I_{sh} \tag{I.4}$$

Current through field winding:

$$I_{sh} = \frac{V}{R_{sh}} \tag{I.5}$$

Back emf developed:

$$E_b = V - I_a R_a \tag{I.6}$$

Power drawn from the supply:

$$P = VI \tag{I.7}$$

Mechanical Power developed:

$$P_m = Input\ power – Losses\ in\ the\ armature\ and\ field\ winding$$

$$P_m = VI - VI_{sh} - I_a^2 R_a \tag{I.8}$$

$$P_m = E_b I_a \tag{I.9}$$

3.3.2. DC Series Motor :

The field winding is connected in series with the armature winding to make DC series Motor. The field winding has a lesser number of turns with thick wire. It offers low resistance for the flow of current through the armature [12].

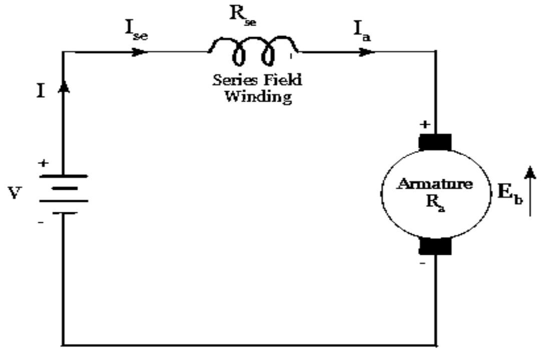


Figure I.13: *The circuit of DC Series Motor.*

Hence:

$$I = I_{se} = I_a \qquad (I.10)$$

Back emf developed: 0

$$E_b = V - I_{se}R_{se} - I_aR_a = V - I(R_{se} + R_a) \qquad (I.11)$$

Power drawn from the supply:

$$P = VI \qquad (I.12)$$

Mechanical Power developed:

$$P_m = Input\ power - Losses\ in\ the\ armature\ and\ field\ winding$$

$$P_m = VI - I_{se}^2R_{se} - I_a^2R_a \qquad (I.13)$$

$$P_m = I\big(V - I(R_{se} + R_a)\big) = E_bI \qquad (I.14)$$

3.3.3. DC Compound Motors :

Compound DC Motor is a type of DC Motor, which has both the shunt field winding and series field winding. Among the fluxes produced by both these windings, the shunt field flux is stronger than the series field winding. If the flux produced by the series field winding strengthens the shunt field flux, it is called a differential compound Motor. The flow of direction of current is the same in both the field windings. On the other side, if the series field flux weakens the shunt field flux, it is called cumulatively compound Motor. For this case, the flow of current is opposite in both the windings [12].

DC Compound Motor can be further classified into two different types based on the way of connection [12].

3.3.3.1. Long Shunt Compound Wound DC Motor :

The shunt field winding is connected in parallel with a combination of series field winding and armature conductors [12].

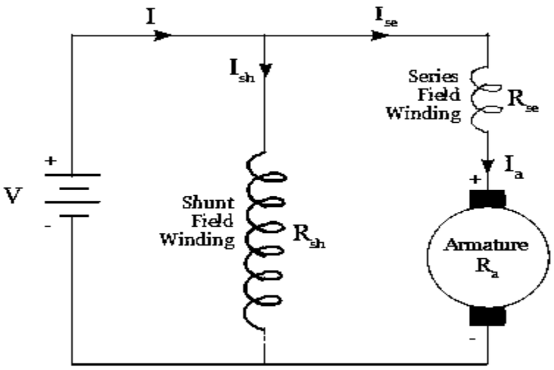


Figure I.14: *The circuit of Long Shunt Compound Wound DC Motor.*

The line current:

$$I = I_{sh} + I_{se} \quad And \quad I_{se} = I_a \tag{I.15}$$

Current through field winding:

$$I_{sh} = \frac{V}{R_{sh}} \tag{I.16}$$

Power drawn from the supply:

$$P \ = \ VI \tag{I.17}$$

Back emf developed:

$$E_b = V - I_{se}R_{se} - I_aR_a = V - I_a(R_{se} + R_a) \tag{I.18}$$

3.3.3.2. Short Shunt Compound Wound DC Motor :

In short shunt DC compound Motor, the armature winding and the shunt field winding are connected in parallel. This pair is again connected in series with the series field winding to construct the Motor [12].

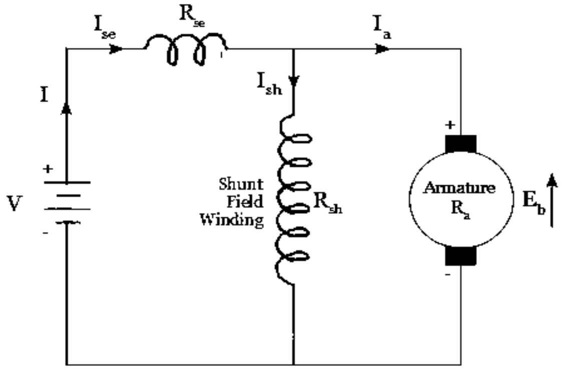


Figure I.15: *The circuit of Short Shunt Compound Wound DC Motor.*

Similar to the long shunt compound Motor the equations are derived as follows.

Line current is given by:

$$I = I_{se} = I_{sh} + I_a \tag{I.19}$$

Power drawn from the supply:

$$P \ = \ VI \tag{I.20}$$

Back emf developed:

$$E_b = V - I_{se}R_{se} - I_aR_a = V - IR_{se} - I_aR_a \tag{I.21}$$

4. DC Motor Modelling :

Torque of a DC motor is proportional to its armature current and field current. The field winding of a DC motor gets excited from a separate power supply, which is independent of its armature current. To achieve a linear variation of torque, one current (either field or armature) is kept constant while varying the other and is possible in a DC motor because its armature and field currents are independent of each other. Therefore, DC motors can be divided into two types depending up on how its torque is being controlled, namely armature-controlled DC motors and field-controlled DC motors [13].

As the discussion about the field-controlled DC Motor is out of scope of this memoir, we shall continue our discussion with armature-controlled DC motors [13].

In an armature-controlled DC motor, the excitation for the field winding is kept constant and the torque is varied by varying the supply voltage connected to the armature. In some cases, a permanent magnet is used instead of field winding to produce the magnetic flux which is again independent of the armature current. Such motors are called Permanent magnet DC motors [13].

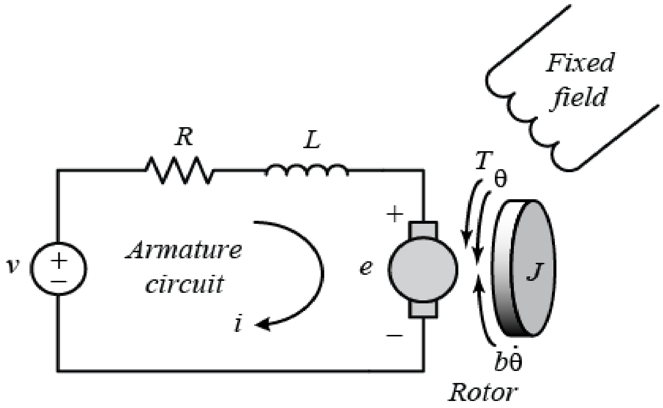


Figure I.16: *Armature-Controlled DC Motor.*

By Kirchhoff voltage law,

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} = v_a(t) - e_b(t) \qquad (I.22)$$

Where $e_b(t)$ is the back EMF produced in the DC motor which is proportional to derivative of angular position of the shaft nothing but speed or angular velocity.

$$e_b(t) = K_e \omega(t) \qquad (I.23)$$

Torque of DC motor can be expressed in terms of product of field flux and armature current. Since in armature-controlled DC Motor field flux is constant, Torque varies only with respect to Armature current. Such Torque expression can be given as:

$$T(t) = K_t i_a(t) \qquad (I.24)$$

The mathematical expression presiding over this DC Motor is given by:

$$J \frac{d\omega(t)}{dt} + B\omega(t) = T(t) - T_L(t) \qquad (I.25)$$

where: $$\begin{cases} R_a : \text{is the armature resistance } (\Omega) \\ L_a : \text{ is armature inductance } (H) \\ J\text{: is the equivalent moment of inertia of the motor } (kg.m^2/s^2) \\ B\text{: is the viscous friction coefficient of the motor } (N.m.s/rad) \\ K_t\text{: } is\ the\ torque\ constant\ (Nm/A) \\ K_e\text{: } is\ the\ back\ emf\ constant\ (V/rad/s) \\ i_a(t)\text{: } is\ the\ armature\ current\ (A) \\ v_a(t)\text{: } is\ the\ armature\ voltage\ (V) \\ \mathrm{\omega(t)}\text{: is the angular speed (rad/s)} \\ \mathrm{T_L(t)}\text{: is the load torque across the motor shaft (Nm)} \end{cases}$$

4.1. Stat space of armature-controlled DC motor :

From the equations (EQ.I.22) and (EQ.I.23):

$$\frac{di_a(t)}{dt} = \frac{R_a}{L_a} i_a(t) - \frac{K_e}{L_a} \omega(t) + \frac{1}{L_a} v_a(t) \qquad (I.26)$$

From the equations (EQ.I.24) and (EQ.I.25):

$$\frac{d\omega(t)}{dt} = -\frac{B}{J} \omega(t) + \frac{K_t}{J} i_a(t) - \frac{1}{J} T_L(t) \qquad (I.27)$$

To build the state-space MIMO model, the input, state and the output vectors are defined:

a. Input vector $u(t)$, whose two components are represented by armature voltage $v_a(t)$ and load torque $T_L(t)$, that is:

$$u(t) = \begin{bmatrix} v_a(t) \\ T_L(t) \end{bmatrix} \qquad (I.28)$$

b. State vector $x(t)$, whose three components are represented by armature current $i_a(t)$, angular position $\theta(t)$ and angular velocity $\omega(t)$:

We put:

$$\begin{cases} x_1(t) = \omega(t) \\ x_2(t) = i_a(t) \end{cases} \qquad (I.29)$$

So:

$$\begin{cases} \dot{x_1}(t) = \dfrac{d\omega(t)}{dt} \\ \dot{x_2}(t) = \dfrac{di_a(t)}{dt} \end{cases} \qquad (I.30)$$

From equations (EQ.I.26) and (EQ.I.27):

$$\begin{cases} \dot{x_1}(t) = -\dfrac{B}{J}x_1(t) + \dfrac{K_t}{J}x_2(t) - \dfrac{1}{J}T_L(t) \\ \dot{x_2} = -\dfrac{K_e}{L_a}x_1(t) + \dfrac{R_a}{L_a}x_2(t) + \dfrac{1}{L_a}v_a(t) \end{cases} \qquad (I.31)$$

c. Output vector $y(t)$, whose two components are represented by angular position $\theta(t)$ and angular velocity $\omega(t)$

Finally [14]:

$$\begin{cases} \begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{B}{J} & \dfrac{K_t}{J} \\ -\dfrac{K_e}{L_a} & \dfrac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & -\dfrac{1}{J} \\ \dfrac{1}{L_a} & 0 \end{bmatrix} \begin{bmatrix} v_a(t) \\ T_L(t) \end{bmatrix} \\ \quad y(t) \quad = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(t) \\ T_L(t) \end{bmatrix} \end{cases} \qquad (I.32)$$

4.2. Transfer function of armature-controlled DC motor :

From the equations (EQ.I.22), (EQ.I.23), (EQ.I.24) and (EQ.I.25) we obtained:

$$\begin{cases} L_a\dfrac{di_a(t)}{dt} + R_a i_a(t) + K_e\omega(t) \ = v_a(t) \\ J\dfrac{d\omega(t)}{dt} + B\omega(t) - K_t i_a(t) = -T_L(t) \end{cases} \qquad (I.33)$$

Assume that all the initial conditions are zero. Taking the Laplace transform of equations (I.33) results in:

$$\begin{cases} L_a sI_a(s) + R_a I_a(s) + K_e\Omega(s) = V_a(s) \\ Js\Omega(s) + B\Omega(s) - K_t I_a(s) = -T_L(s) \end{cases} \qquad (I.34)$$

1. The transfer function relating the armature voltage $V_a(s)$ and angular velocity $\Omega(s)$, with $T_L(s) = 0$:

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_t}{L_a J s^2 + (L_a B + R_a J)s + R_a B + K_t K_e} \qquad (I.35)$$

2. The transfer function relating the load torque $T_L(s)$ and angular velocity $\Omega(s)$, with $V_a(s) = 0$:

$$\frac{\Omega(s)}{T_L(s)} = -\frac{L_a s + R_a}{L_a J s^2 + (L_a B + R_a J)s + R_a B + K_t K_e} \qquad (1.36)$$

Note that the two transfer functions have the same denominator, which is characteristic of the system. The order of the characteristic polynomial implies that the system is second-order [15].

4.3. Block diagram of armature-controlled DC motor [16] *:*

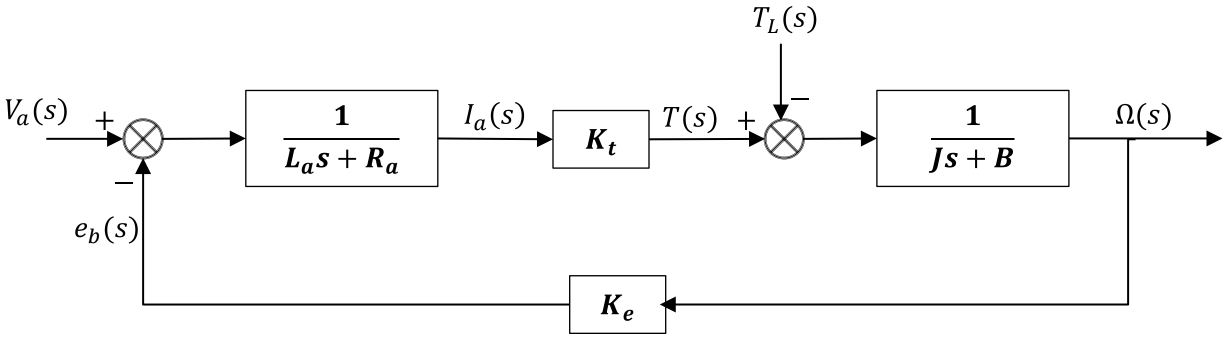


Figure I.17: *Block diagram of armature-controlled DC motor.*

5. Simulation of armature-controlled DC motor :

5.1. DC Motor parameter used :

We simulate a DC motor that characterised by :

$$\begin{cases} V_a = 120\ v \\ R_a = 1.5\ \Omega \\ L_a = 0.2\ H \\ J = 0.0988\ kg.m^2/s^2 \\ B = 0.000587\ N.m.s/rad \\ K_e = 0.67609\ (V/rad/s) \\ K_t = 0.67609\ N.m/A \end{cases}$$

5.2. Simulation without measurement noise and load torque:

We simulate this DC motor which powered by constant voltage $V_a = 120\ v$ as shown in figure I.18 to obtain the curve of speed change as shown in figure I.19.

5.2.1. Diagram on Simulink :

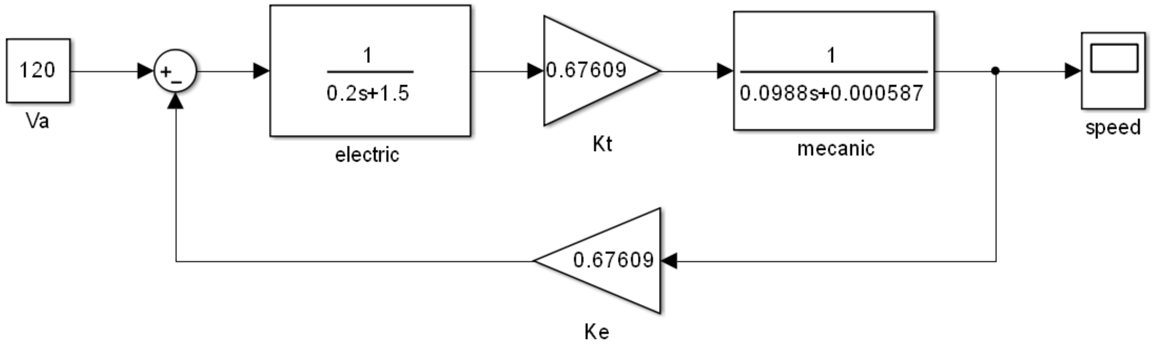


Figure I.18: *Simulink model of open-loop block diagram.*

5.2.2. Simulation results :

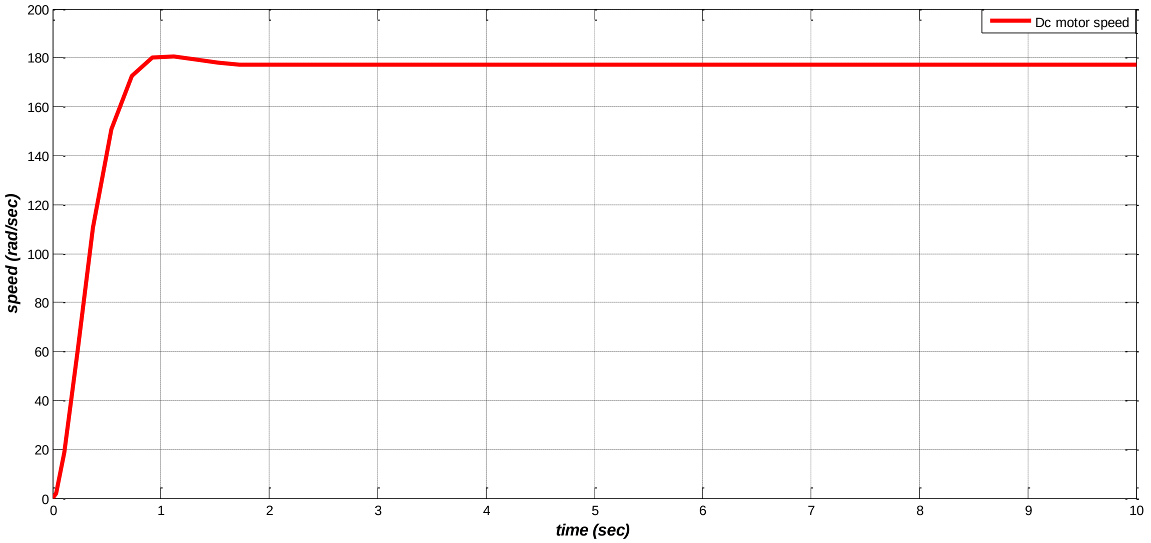


Figure I.19: *simulation result in speed scope*

5.3. Simulation with measurement noise and load torque :

We apply a torque load $T_L = 10\ N.m$ at the moment $t = 2.5\ sec$ and a measurement noise ranges from $[-0.5; 0.5]\ rad/sec$ start $t = 5\ sec$, we obtain the curve of speed change as shown in figure I.23

5.3.1. Diagram on Simulink :

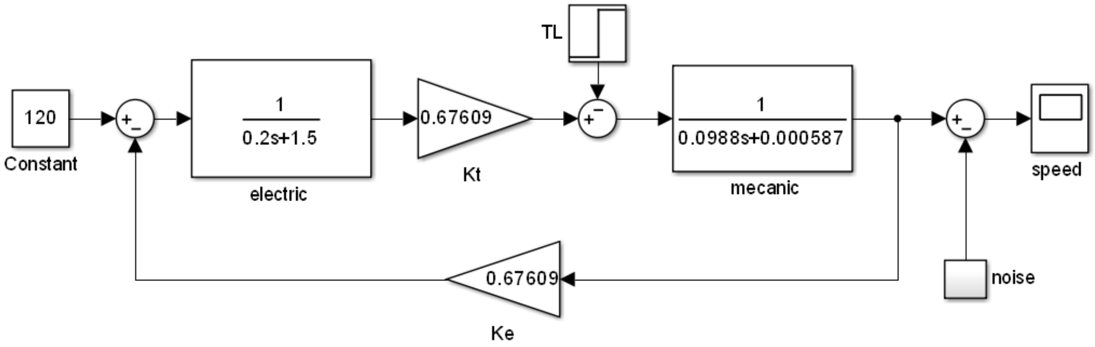


Figure I.20: *Simulink model of block diagram with torque load and measurement noise.*

5.3.2. Simulation results :

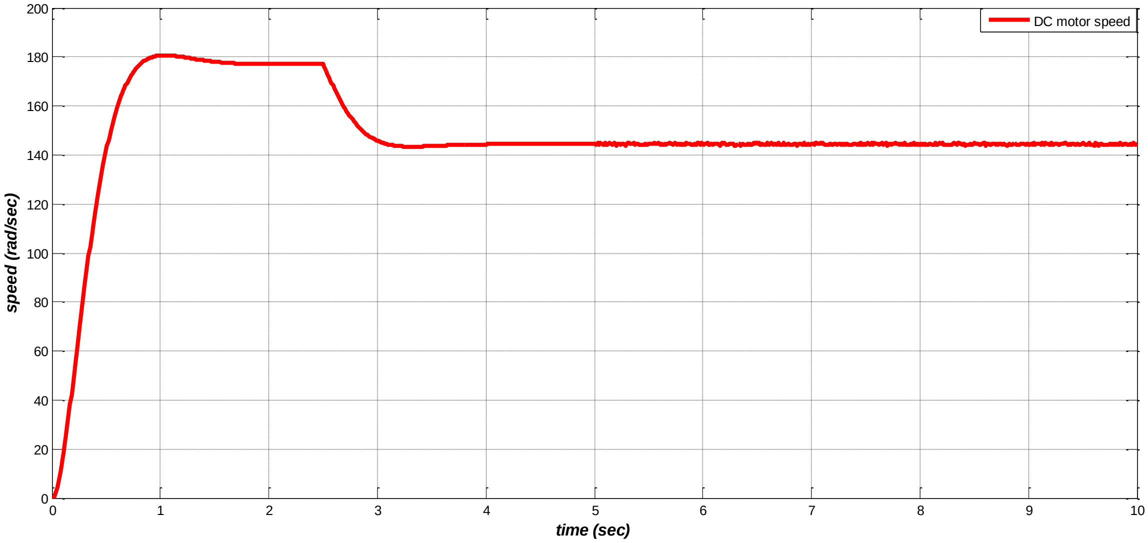


Figure I.21: *simulation result in speed scope.*

Observation :

We noticed that the speed of the DC motor was affected by the presence of the load torque and decreased by about 33 (rad/s) we also note that there is uncertainty in the measured speed due to the measurement noise, and this effect is due to the lack of a controller.

6. Conclusion :

In this chapter, we talked about the DC motor, its construction and classification, then we did the mathematical model, the stat space and transfer function, after that, we used the SIMULINK MATLAB to make a simulation of the DC motor.

We noticed that the DC motor speed does not follow the reference speed and is affected by the load torque and the measurement noise, so we must use a controller for the rejection of the load torque or any other external effect, and the minimization of the measurement noise, to make the DC motor rotate at the speed that we want.

Chapter II :

Fundamentals of fractional calculus and systems

1. Introduction :

Fractional calculus is known as a mathematical tool for generalizing conventional integrals and derivatives, the topic attracted the attention of mathematical giants such as Euler, Laplace, Fourier, Abel, Laurent, Hardy and Littlewood. In the 19th cycle and thanks to the main contributions of Liouville [17].

Grünwald, Letnikov and Riemann a whole complete theory adapted to modern mathematical developments has been formalized. Nowadays, fractional calculus is a well-established theory with strong mathematical bases**.** The main reason for the vast spreading of fractional calculus is that it actually provides excellent capabilities to accurately model several physical processes [17].

In this chapter, we will talk about the basic definitions of fractional calculus along with its mathematical preliminaries, the representation and analysis of fractional order systems and the application fractional calculus.

2. History :

Fractional calculus was started by some research works of G. W. Leibniz (1695, 1697) and L. Euler (1730), then it was developed until the middle of the 20th century by numerous researchers such as: P. S. Laplace (1812), S. F. Lacroix (1819), J. B. J. Fourier (1822), J. Liouville (1832-1873), B. Riemann (1847), H. Holmgren (1865-1867), A. K. Grunwald (1867-1872), A. V. Letnikov (1868-1872), A. Erdelyi (1939-1965), H. Kober (1940), D. V. Widder (1941), M.Riesz (1949), W.Feller (1952) [18].

Fractional calculus appeared again with the 1st book published in 1974 by K. B. Oldham and J. Spanier. The fractional calculus continues its growing development stimulated by several applications that they find in different fields of applied sciences such as physics and engineering, possibly including fractional phenomena. Now there are more books, proceedings and special issues of journals published that refer to applications of fractional calculus. applications of fractional calculus in various scientific fields including special functions, control theory, chemical physics, random processes, anomalous diffusion, rheology. Several special issues have been published in the contain selected and improved papers, presented at conferences and graduate schools, on various conferences and graduate schools, concerning various applications of fractional calculus technology. For several years now, there have been two international journals devoted almost exclusively to the subject of fractional calculus. almost exclusively to the subject of fractional calculus: Journal of Fractional Calculus (editor: K. Nishimoto, Japan) started in 1992, and Fractional Calculus and Applied Analysis (Managing Editor: V. Kiryakova, Bulgaria) started in 1998 [18].

3. Mathematical preliminaries :

Fractional calculus is a generalization of integration and differentiation to the non-integer order fundamental operator ${}_{t_0}D_t^{\alpha}$, where t_0 and t are the limits of the operation. The continuous integro-differential operator is defined as [19]:

$$ {}_{t_0}D_t^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{\alpha}}, & \Re > 0 \\ 1, & \Re = 0 \\ \int_{t_0}^{t} (d\tau)^{-\alpha} & \Re < 0 \end{cases} \qquad (II.1) $$

Where: α is the order of the operation, generally $\alpha \in \mathbb{R}$ but could also be a complex number.

3.1. Useful Mathematical Functions :

Before looking at the definition of the Riemann-Liouville Fractional Integral or Derivative, we will first talk about some useful mathematical definitions that are inherently tied to fractional calculus and will commonly be encountered. These include the Gamma function, the Beta function, the Error function, the Mittag-Leffler function, and the Mellin-Ross function [20].

3.1.1. The Gamma Function :

The most basic interpretation of the Gamma function is simply the generalization of the factorial for all real numbers. Its definition is given by [20]:

$$ \Gamma(x) = \int_0^{\infty} e^{-t}\, t^{x-1} dt\,, \quad x \in \mathbb{R}^{+} \qquad (II.2) $$

The Gamma function has some unique properties. By using its recursion relations, we can obtain formulas:

$$ \Gamma(x+1) = x\Gamma(x), \quad x \in \mathbb{R}^{+} \qquad (II.3) $$

$$ \Gamma(x) = (x-1)!\ , \quad x \in \mathbb{N} \qquad (II.4) $$

From equation (EQ.II.4) we note that $\Gamma(1) = 1$. We now show that $\Gamma(1/2) = \sqrt{\pi}$.

By definition (EQ.II.2) we have:

$$ \Gamma(1/2) = \int_0^{\infty} e^{-t}\, t^{1/2} dt \qquad (II.5) $$

If we let $t = y^2$, then $dt = 2ydy$ and we now have:

$$ \Gamma(1/2) = 2\int_0^{\infty} e^{-y^2} dy \qquad (II.6) $$

Equivalently, we can write (EQ.II.6) as:

$$ \Gamma(1/2) = 2\int_0^{\infty} e^{-x^2} dx \qquad (II.7) $$

If we multiply together (EQ.II.6) and (EQ.II.7) we get:

$$[\Gamma(1/2)]^2 = 4\int_0^\infty \int_0^\infty e^{-(x^2+y^2)}dx\,dy \qquad (II.8)$$

Equation (EQ.II.8) is a double integral over the first quadrant, and can be evaluated in polar coordinates to get:

$$\left[\Gamma\left(\frac{1}{2}\right)\right]^2 = 4\int_0^{\frac{\pi}{2}} \int_0^\infty e^{-r^2} r dr\, d\theta = \pi \;\Rightarrow\; \Gamma(1/2) = \sqrt{\pi} \qquad (II.9)$$

The incomplete Gamma function is a closely related function defined as:

$$\Gamma^*(v,t) = \frac{1}{\Gamma(v)t^v}\int_0^t e^{-x}\, x^{v-1}dx\,, \quad \Re\{v\} > 0 \qquad (II.10)$$

3.1.2. The Beta Function :

Like the Gamma function, the Beta function is defined by a definite integral. Its definition is given by [20]:

$$B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt\,, \quad x,y \in x \in \mathbb{R}^+ \qquad (II.11)$$

The Beta function can also be defined in terms of the Gamma function:

$$B(x,y) = \frac{\Gamma(\mathrm{x})\Gamma(\mathrm{y})}{\Gamma(\mathrm{x}+\mathrm{y})}, \quad x,y \in x \in \mathbb{R}^+ \qquad (II.12)$$

3.1.3. The Error Function :

The definition of the Error function is given by [20]:

$$Erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt\,, \quad x \in \mathbb{R} \qquad (II.13)$$

The complementary Error function (Erfc) is a closely related function that can be written in terms of the Error function as:

$$Erfc(x) = 1 - Erf(x) \qquad (II.14)$$

As a result of (EQ.II.13) we note that $Erf(0) = 0$ and $Erf(\infty) = 1$.

3.1.4. The Mittag-Leffler Function :

The Mittag-Leffler function is named after a Swedish mathematician who defined and studied it in 1903. The function is a direct generalization of the exponential function, e^x, and it plays a major role in fractional calculus. The one and two-parameter representations of the Mittag-Leffler function can be defined in terms of a power series as [20]:

$$E_\alpha(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k+1)}, \quad \alpha > 0 \qquad (II.15)$$

$$E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k+\beta)}, \quad \alpha > 0, \quad \beta > 0 \qquad (II.16)$$

The exponential series defined by (EQ.II.16) gives a generalization of (EQ.II.15). This more generalized form was introduced by R.P. Agarwal in 1953 [20].

As a result of the definition given in (EQ.II.16), the following relations hold [20]:

$$E_{\alpha,\beta}(x) = \frac{1}{\Gamma(\beta)} + xE_{\alpha,\beta+1}(x) \qquad (II.17)$$

And:

$$E_{\alpha,\beta}(x) = \beta E_{\alpha,\beta+1}(x) + \alpha x \frac{d}{dx} E_{\alpha,\beta+1}(x) \qquad (II.18)$$

Observe that (EQ.II.18) implies that:

$$\frac{d}{dx} E_{\alpha,\beta+1}(x) = \frac{1}{\alpha x}\left[E_{\alpha,\beta}(x) - \beta E_{\alpha,\beta+1}(x)\right] \qquad (II.19)$$

So:

$$\frac{d}{dx} E_{\alpha,\beta}(x) = \frac{1}{\alpha x}\left[E_{\alpha,\beta-1}(x) - (\beta-1) E_{\alpha,\beta}(x)\right] \qquad (II.20)$$

We now prove (EQ.II.17).

By definition (EQ.II.16):

$$\begin{aligned} E_{\alpha,\beta}(x) &= \sum\nolimits_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k + \beta)} \\ &= \sum\nolimits_{k=-1}^{\infty} \frac{x^{k+1}}{\Gamma(\alpha(k+1) + \beta)} \\ &= \sum\nolimits_{k=-1}^{\infty} \frac{xx^k}{\Gamma\big(\alpha k + (\alpha+\beta)\big)} \qquad (II.21) \\ &= \frac{1}{\Gamma(\beta)} + x\sum\nolimits_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha(k+1) + \beta)} \\ &= \frac{1}{\Gamma(\beta)} + xE_{\alpha,\alpha+\beta}(x) \end{aligned}$$

Note that $E_{\alpha,\beta}(0) = 1$. Also, for some specific values of α and β, the Mittag-Leffler function reduces to some familiar functions. For example:

$$E_{1,1}(x) = \sum\nolimits_{k=0}^{\infty} \frac{x^k}{\Gamma(k+1)} = \sum\nolimits_{k=0}^{\infty} \frac{x^k}{x!} = e^x \qquad (II.22)$$

$$E_{1/2,1}(x) = \sum\nolimits_{k=0}^{\infty} \frac{x^k}{\Gamma(k/2+1)} = e^{x^2} Erfc(-x) \qquad (II.23)$$

$$E_{1,2}(x) = \sum\nolimits_{k=0}^{\infty} \frac{x^k}{\Gamma(k+2)} = \frac{1}{x}\sum\nolimits_{k=0}^{\infty} \frac{x^{k+1}}{(k+1)!} = \frac{e^x - 1}{x} \qquad (II.24)$$

3.1.5. The Mellin-Ross Function :

The Mellin-Ross function, $E_t(v,a)$, arises when finding the fractional integral of an exponential, e^{at}. The function is closely related to both the incomplete Gamma and Mittag-Leffler functions. Its definition is given by [20]:

$$E_t(v,a) = t^v e^{at}\Gamma^*(v,a) \qquad (II.25)$$

We can also write:

$$E_t(v,a) = t^v \sum_{k=0}^{\infty} \frac{(at)^k}{\Gamma(k+v+1)} = t^v E_{1,v+1}(at) \qquad (II.26)$$

3.2. Definition of fractional operator :

Up to now, there are more than 10 types of definitions for fractional order integrals and differentiations [21]. For researchers' convenience, several commonly used definitions are briefly listed below [22].

3.2.1 Riemann-Liouville definition of fractional order integration :

The Riemann-Liouville (R-L) definition of fractional order integration is:

$$_0D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} \left(\frac{f(\tau)}{(t-\tau)^{1-\alpha}} \right) d\tau \qquad (II.27)$$

where, $0 < \alpha < 1$ and $\Gamma(.)$ is the Gamma function, When the $\Gamma(x) = \int_0^\infty e^{-u} u^{x-1} du$ initial integral limit changes from 0 to an arbitrary point t_0, this definition is generalized to the Weyl definition of fractional order integral:

$$_{t_0}D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} \left(\frac{f(\tau)}{(t-\tau)^{1-\alpha}} \right) d\tau \qquad (II.28)$$

3.2.2 Riemann-Liouville definition of fractional order differentiation :

The R-L definition of fractional order differentiation is based on the fractional integral and the ordinary derivatives:

$$_0D_t^{-\alpha} f(t) = \frac{d}{dt} \int_{t_0}^{t} \left[{_0D_t^{-(1-\alpha)}} f(t) \right] d\tau \qquad (II.28)$$

More specifically, there are left R-L and right R-L definitions for fractional order differentiation by distinguishing the lower and upper limits of the integration.

$$_{t_0}D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt} \right)^n \int_{t_0}^{t} (t-\tau)^{a-\alpha-1} f(\tau) d\tau \qquad (II.29)$$

$$_tD_{t_1}^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(-\frac{d}{dt} \right)^n \int_{t}^{t_1} (t-\tau)^{a-\alpha-1} f(\tau) d\tau \qquad (II.30)$$

The complete Riemann-Liouville definition for the integration or differentiations of a fractional order α for a function $y(t)$ is given by:

$$ {}_{t_0}D_t^{\alpha} y(t) = \begin{cases} \frac{1}{\Gamma(-\alpha)} \int_{t_0}^{t} (t-\tau)^{-\alpha-1} y(t) d\tau \,, & \text{if } \Re(\alpha) < 0 \\ y(t) \,, & \text{if } \Re(\alpha) = 0 \\ D^n\left[{}_{t_0}D_t^{\alpha-n} y(t)\right], \; n = \min\{k \in \aleph : k > \Re(\alpha)\}\,, & \text{if } \Re(\alpha) > 0 \end{cases} \qquad (II.31) $$

where $\alpha \in \mathbb{C}\,, n$ is a positive integer and y is a locally integrable function defined on $[t_0\,, \infty]$.

3.2.3 Caputo definition of fractional order differentiation :

The Caputo definition of fractional order differentiation takes the integer order differentiation of the function first and then take a fractional order integration:

$$ {}_0^C D_t^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t} \left(\frac{f'(\tau)}{(t-\tau)^{\alpha}} \right) d\tau \qquad (II.32) $$

Under this definition, D and ${}_0D_t^{-(1-\alpha)}$ do not commute because the initial value needs be considered:

$$ {}_0^C D_t^{\alpha} f(t) = {}_0D_t^{-(1-\alpha)} f(t)[Df(t)] + \frac{f(0^+)t^{-\alpha}}{\Gamma(1-\alpha)} \qquad (II.33) $$

3.2.4. Grünwald-Letnikov definition :

The Grünwald-Letnikov (G-L) definition defines the fractional integration and differentiation in a unified way, as positive values of α give fractional differentiation and negative values of α give fractional order integration.

$$ {}_{t_0}D_t^{\alpha} f(t) = \lim_{h\to 0} \left(\frac{1}{h^{\alpha}}\right) \sum_{j=0}^{\left[\frac{(t-t_0)}{h}\right]} (-1)^j \binom{\alpha}{j} f(t-jh) \qquad (II.34) $$

Where: $\binom{\alpha}{j}$ represents the binomial coefficient, which is described in terms of the Gamma function.

3.3. Proprieties of fractional order operator :

For the fractional derivative of a time function $f(t)$ exists, it is sufficient that $f(t)$ can be written in the form [21]:

$$ f(t) = (t-t_0)^{\lambda} \eta(t-t_0) with \begin{cases} \lambda \in C\,, \\ \Re(\lambda)\,, \\ \eta(t)\,, \quad \text{analytic function of } C \text{ for } t \geq 0 \end{cases} \qquad (II.35) $$

And so that its fractional order integral exists, it is sufficient that $f(t)$ is piecewise continuous on $]t_0, +\infty[$ and integrable on $[t_0, t]$ for any $t > t_0$.

3.3.1. Properties of fractional order integral :

The fractional order integral operator meets the semi-group property, namely [23]:

$$I_{t_0}^{v_1} \circ I_{t_0}^{v_2} = I_{t_0}^{v_1+v_2} \; with \begin{cases} \Re(v_1) > 0 \\ \Re(v_2) > 0 \end{cases} \qquad (II.36)$$

And thus, the commutativity property:

$$I_{t_0}^{v_1} \circ I_{t_0}^{v_2} = I_{t_0}^{v_2} \circ I_{t_0}^{v_1} \qquad (II.37)$$

3.3.2. Properties of fractional order differentiation :

The fractional-order differentiation has the following properties:

1. The fractional-order differentiation ${}_0D_t^{\alpha} f(t)$, with respect to t of an analytic function $f(t)$, is also analytical [24]**.**
2. The fractional-order differentiation is exactly the same with integer-order one, when $\alpha = n$ is an integer. Also ${}_0D_t^{0} f(t) = f(t)$.
3. The fractional-order differentiation is linear; i.e., for any constants a, b, one has,

$$ {}_0D_t^{\alpha} D[af(t) + bg(t)] = a\,{}_0D_t^{\alpha} f(t) + b\,{}_0D_t^{\alpha} g(t) \qquad (II.38)$$

4. Fractional differentiation operators satisfy the commutative-law, and also satisfy:

$$ {}_0D_t^{\alpha}\left[{}_0D_t^{\beta} f(t)\right] = {}_0D_t^{\alpha+\beta} \qquad (II.39)$$

3.4. Laplace transforms of fractional operator :

Laplace transform (denoted as $\mathcal{L}$) is one of the most mathematical tools in dynamic system and control engineering. For this reason, we will give here the equation of these transforms for the defined fractional order integrals and differentiations.

3.4.1 Laplace transform of fractional order integral :

Writing of equation (EQ.II.1) using the convolution product, the Laplace transform of fractional integral is:

$$\mathcal{L}\{{}_0D_t^{-\alpha} f(t)\} = \mathcal{L}\left\{\frac{t^{\alpha-1}u(t)}{\Gamma(\alpha)} * f(t)\right\} = \mathcal{L}\left\{\frac{t^{\alpha-1}u(t)}{\Gamma(\alpha)}\right\}\mathcal{L}\{f(t)\} = s^{-\alpha}F(s) \qquad (II.40)$$

where, $u(t)$ is the unit step function, $R(\alpha) > 0$, $F(s) = \mathcal{L}\, f(t)$ and $s = (\sigma + j\omega)$ is the Laplace operator.

This is a remarkable result that generalizes the well-known formula of the integral Laplace transform in the integer case.

3.4.2 Laplace transform of fractional order differentiation :

The Laplace transform of the fractional derivative of a causal time function is given by:

$$\mathcal{L}\{\,D^{\alpha}f(t)\} = s^{\alpha}F(s) - \; D^{\alpha-1}f(0) \qquad (II.41)$$

This is a remarkable result that generalizes the well-known formula of the derivative Laplace transform in the integer case.

In particular, if the derivatives of the function $f(t)$ are all equal to 0 at $t = 0$, one has [25]:

$$\mathcal{L}[{}_0D_t^{\alpha}f(t)] = s^{\alpha}\mathcal{L}[f(t)] \qquad (II.42)$$

(Table II.1) gives a brief summary of some useful Laplace transform pairs. We will frequently refer to this Table. Notice that the Mittag-Leffler function is very prominent.

As will become more evident later on, this function plays an important role when solving fractional differential equations [20].

$Y(s)$	$\gamma(t) = \mathcal{L}^{-1}\{Y(s)\}$
$\frac{1}{s^{\alpha}}$	$\frac{t^{\alpha-1}}{\Gamma(\alpha)}$
$\frac{1}{(s+a)^{\alpha}}$	$\frac{t^{\alpha-1}}{\Gamma(\alpha)}e^{-at}$
$\frac{1}{s^{\alpha}-a}$	$t^{\alpha-1}E_{\alpha,\alpha}(at^{\alpha})$
$\frac{s^{\alpha}}{s(s^{\alpha}+a)}$	$E_{\alpha}(-at^{\alpha})$
$\frac{s^{\alpha}}{s^{\alpha}}$	$1 - E_{\alpha}(-at^{\alpha})$
$\frac{a}{s(s^{\alpha}+a)}$	$t^{\beta-1}E_{\alpha,\beta}(at^{\alpha})$
$\frac{1}{s^{\alpha}(s-a)}$	$t^{\alpha}E_{1,\alpha+1}(at\;)$
$\frac{s^{\alpha-\beta}}{s^{\alpha}-a}$	$t^{\beta-1}E_{\alpha,\beta}(at^{\alpha})$
$\frac{s^{\alpha-\beta}}{(s-a)^{\alpha}}$	$\frac{t^{\beta-1}}{\Gamma(\beta)}F_1(\alpha;\beta;at)$
$\frac{1}{(s-a)(s-b)}$	$\frac{1}{a-b}(e^{at}-e^{bt})$

Table II.1: *Laplace transform pairs.*

3.4.3 Example of Laplace Transform :

Let's solve $D^{2/3}y(t) = ay(t)$, where a is a constant.

Since $0 > v = 2/3 > 1$, we will use (EQ.II.41). Taking the Laplace transform of both sides of the equation we have:

$$\mathcal{L}\{D^{2/3}y(t)\} = a\mathcal{L}\{y(t)\} \qquad (II.43)$$

which implies that:

$$s^{2/3}Y(s) - D^{-(1-2/3)}y(0) = aY(s) \qquad (II.44)$$

The constant: $D^{-(1-2/3)}y(0) = D^{-1/3}y(0)$ is the value of $D^{-1/3}y(t)$ at $t = 0\ sec$

If we assume that this value exists, and call it c_1, then (EQ.II.44) becomes:

$$s^{2/3}Y(s) - c_1 = aY(s) \qquad (II.45)$$

Solving for $Y(s)$ we obtain:

$$Y(s) = \frac{c_1}{s^{2/3} - a} \qquad (II.46)$$

Finally, using (Table II.1) we find the inverse Laplace of $Y(s)$, and conclude that:

$$y(t) = \mathcal{L}^{-1}\left\{\frac{c_1}{s^{2/3} - a}\right\} = c_1 t^{-1/3} E_{2/3,2/3}(at^{2/3}) \qquad (II.47)$$

In Example 1 (and other similar situations) one may wonder whether the existence of $D^{-1/3}y(0)$ implies that its value is actually as assumed. We will show that indeed that is the case [20].

Again, using the Laplace transform (EQ.II.40) we note that:

$$\mathcal{L}\{D^{-1/3}y(t)\} = s^{-1/3}Y(s) \qquad (II.48)$$

Since:

$$Y(s) = \frac{c_1}{s^{2/3} - a} \qquad (II.49)$$

Then:

$$\mathcal{L}\{D^{-1/3}y(t)\} = \frac{c_1 s^{-1/3}}{s^{2/3} - a} \qquad (II.50)$$

So:

$$D^{-1/3}y(t) = \mathcal{L}^{-1}\left\{\frac{c_1 s^{-1/3}}{s^{2/3} - a}\right\} = c_1 E_{2/3,1}(at^{2/3}) \qquad (II.51)$$

At $t = 0$:

$$D^{-1/3}y(0) = c_1 E_{2/3,1}(0) = c_1 \qquad (II.52)$$

4. Representations and Analysis of Fractional Order Systems :

4.1. Fractional Order differential equation :

In the fields of dynamical systems and control theory, a fractional-order system is a dynamical system that can be modelled by a fractional differential equation containing non-integer operators. Such systems are said to have fractional dynamics. A general continuous-time dynamical linear system of fractional order can be described by the following form [26]:

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) = \\ b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t) \qquad (II.53)$$

where $y(t)$ and $u(t)$ are functions of the fractional derivative operator D of orders α_i, β_j $(i, j = 1,2,3 \dots)$, that can be arbitrary real numbers, i:e:$(\alpha_i, \beta_j) \in \mathbb{R}_+^2$ and $(a_i, b_j) \in \mathbb{R}^2$ If the $(\alpha_i$ and $\beta_j)$ are integer multiples of a common factor q, the system is called having commensurate-order, if $\left(\alpha_i, \beta_j\right) = kq, q \in \mathbb{R}^+$. The system can then be expressed as:

$$\sum_{k=0}^{n} a_k D^{kq} y(t) = \sum_{k=0}^{m} b_k D^{kq} u(t) \qquad (II.54)$$

If in (EQ.II.54) the order is $q = (1/r), r \in \mathbb{Z}^+$, the system will be of rational order; and is of $non - commensurate - order$ if no common factor exists [27].

This way, linear time-invariant systems can be classified as given in (Figure II.1):

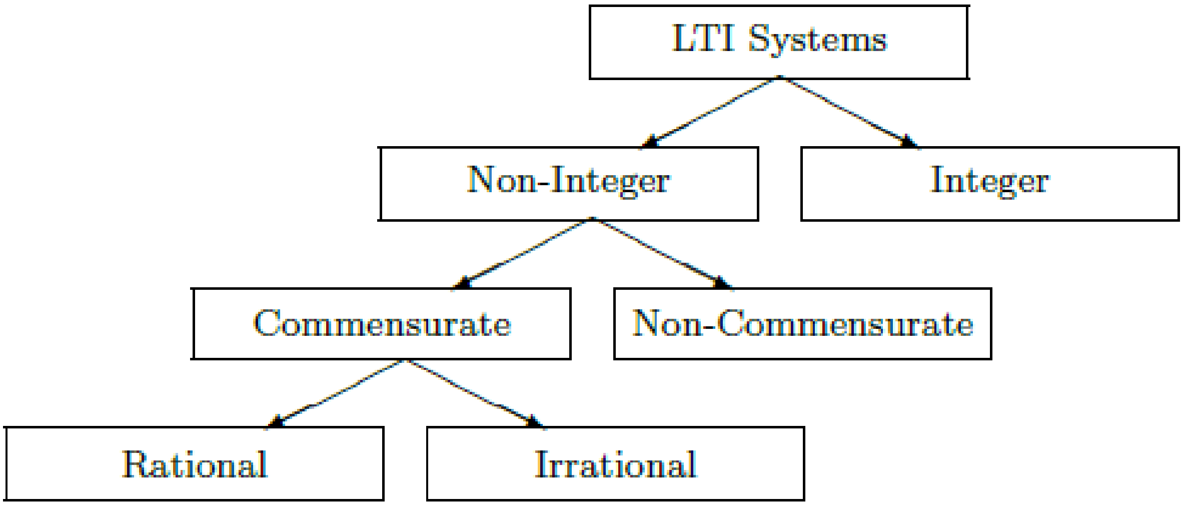


Figure II.1: *Classification of LTI systems*

4.2. Fractional Order Transfer Functions :

Applying the Laplace transform (EQ.II.41) to (EQ.II.53) with zero initial conditions, the input-output representation of the fractional-order system can be obtained in the form of a transfer function:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{b_m s^{\beta_m}+b_{m-1}s^{\beta_{m-1}}+\cdots+b_0 s^{\beta_0}}{a_n s^{\alpha_n}+a_{n-1}s^{\alpha_{n-1}}+\cdots+a_0 s^{\alpha_0}} \qquad (II.55)$$

We shall call the number of fractional poles in (EQ.II.55) the pseudo-order of the system. In the case of a system with commensurate order q, we may take $\sigma = s^q$ and consider the continuous-time pseudo-rational transfer function:

$$G(\sigma)=\frac{\sum_{k=0}^{m} b_k\sigma^k}{\sum_{k=0}^{n} a_k\sigma^k} \qquad (II.56)$$

4.3. Time and Frequency Domain Analysis of Fractional Systems :

To analyses the dynamical properties of a system, the time and frequency domain can be used. First, we begin with the frequency domain analysis.

4.3.1. Frequency domain analysis :

Frequency domain response may be obtained by substituting $s=j\omega$ in (EQ.II.55). The complex response for a frequency $\omega\ \in (0;\ \infty)$ can then be computed as follows:

$$G(j\omega)=\frac{Y(j\omega)}{U(j\omega)}=\frac{b_m(j\omega)^{\beta_m}+b_{m-1}(j\omega)^{\beta_{m-1}}+\cdots+b_0(j\omega)^{\beta_0}}{a_n(j\omega)^{\alpha_n}+a_{n-1}(j\omega)^{\alpha_{n-1}}+\cdots+a_0(j\omega)^{\alpha_0}} \qquad (II.57)$$

where j is the imaginary unit.

$$j^{\mu}=\cos\left(\frac{\mu\pi}{2}\right)+j\,sin\left(\frac{\mu\pi}{2}\right) \qquad (II.58)$$

In addition, consider the useful relation in (EQ.II.58) for a non-integer power of the imaginary unit j

4.3.2. Time domain analysis :

Another solution involves numerical computation of fractional-order derivatives which is carried out by means of a revised Grünwald-Letnikov definition (EQ.II.34) rewritten as:

$$ {}_{t_0}D_t^{\alpha}f(t)=\lim_{h\to 0}\left(\frac{1}{h^{\alpha}}\right)\sum_{j=0}^{\left[\frac{(t-t_0)}{h}\right]}\omega_j^{(\alpha)}f(t-jh) \qquad (II.59)$$

Where: h is the computation step-size and $\omega_j^{(\alpha)}=(-1)^j\binom{\alpha}{j}$ can be evaluated recursively from:

$$\omega_0^{(\alpha)}=1,\quad \omega_j^{(\alpha)}=\left(1-\frac{\alpha+1}{j}\right)\omega_{j-1}^{(\alpha)},\qquad j=1,2,\ldots \qquad (II.60)$$

To obtain a numerical solution for the equation in (EQ.II.53) the signal $\hat{u}(t)$ should be obtained first, using the algorithm in (EQ.II.60), where:

$$\hat{u}(t) = b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t) \qquad (II.61)$$

The time response of the system can then be obtained using the following equation:

$$y(t) = \frac{1}{\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}}} \left[\hat{u}(t) - \sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^{\left[\frac{(t-t_0)}{h}\right]} \omega_j^{(\alpha)} f(t - jh) \right] \qquad (II.62)$$

The presented method is a fixed step method. The accuracy of simulation therefore may depend on the step size [28,29,30]:

If the equation (EQ.II.53) has time delay L, the resulting delayed response $y_d(t)$ with $y_d(0) = 0$ is obtained such that:

$$y_d(t) = \begin{cases} y(t-L), & t > L \\ 0, & otherwise. \end{cases} \qquad (II.63)$$

4.4. Discrete Models of Fractional Order Systems :

Discrete models of fractional order systems can be obtained using discrete approximations of the fractional integrals and derivatives operators. Thus, a general expression for the discrete transfer function $G(z)$ of the fractional system $G(s)$, can be obtained as [27]:

$$G(z) = \frac{b_m(\omega(z^{-1}))^{\beta_m} + b_{m-1}(\omega(z^{-1}))^{\beta_{m-1}} + \cdots + b_0(\omega(z^{-1}))^{\beta_0}}{a_n(\omega(z^{-1}))^{\alpha_n} + a_{n-1}(\omega(z^{-1}))^{\alpha_{n-1}} + \cdots + a_0(\omega(z^{-1}))^{\alpha_0}} \qquad (II.64)$$

where, $(\omega(z^{-1}))$ denotes the discrete equivalent of the Laplace operator s, expressed as a function of the complex variable z or the shift operator z^{-1}.

As can be seen in the former equations, a fractional-order system has an irrational-order continuous transfer function in Laplace's domain or a discrete transfer function in the z Domaine of infinite order. In other words, a fractional-order system has an unlimited memory, and obviously the systems of integer-order are just particular cases of this general case in which the memory is limited. It is clear that only in the case of integer order it is possible to realize a transfer function exactly by using conventional lumped elements (resistances, inductances, and capacitors, in the case of analogue realizations), or procedures (finite order difference equations or digital filters in the case of discrete realizations). Because of this, and taking into account that the final step for applying a fractional controller demands a realizable form of it, in this work some continuous and discrete integer order approximations of fractional-order operators are presented.

4.5. fractional order state space :

Many modern control concepts and methodologies are still applicable to the dynamic systems possessing “FO” behaviours. The State-Space (S-S) representation is such a powerful tool. It can be generally expressed as the following by defining appropriate state variables:

$$\begin{cases} {}_0D_t^{\alpha}x(t) = f(x, u, t) \\ \quad y(t) = g(x) \end{cases} \qquad (II.65)$$

Where: $x \in \mathbb{R}^n$ is the state vector of dimension n, and $0 < \alpha < 2$ is the common factor of the differentiation orders. For linear fractional differential equations in (EQ.II.53), the (EQ.II.65) can be simplified as:

$$\begin{cases} {}_0D_t^{\alpha}x(t) = Ax(t) + Bu(t) \\ \quad y(t) = Cx(t) + Du(t) \end{cases} \qquad (II.66)$$

For integer order state space models, the exponential matrix, $\Phi(t) = e^{At}$ is known as the state transition matrix. It can be analogized accordingly that the generalized exponential matrix using Mittag-Leffler, $E_{\alpha}(At^{\alpha})$, plays the same role for fractional order state space models. It is called the fractional order state transition matrix and can be obtained by [26]:

$$\Phi(t) = \mathcal{L}^{-1}\{(s^{\alpha}I - A)^{-1}\} \qquad (II.67)$$

4.5.1. Pseudo state-space representation :

If the fractional differential equation (EQ.II.53) has commensurate order then, for null initial conditions, it admits a pseudo state space representation of the form:

$$\begin{cases} \dfrac{d^{\gamma}}{dt^{\gamma}}x(t) = Ax(t) + Bu(t) \\ \quad y(t) = Cx(t) + Du(t) \end{cases} \qquad (II.68)$$

Where: $x \in \mathbb{R}^n$ is the pseudo state vector, $\gamma = 1/q$ is the fractional order of the system and $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}\ and\ D \in \mathbb{R}^{p\times m}$ are constant matrices.

Remark :

Another way to obtain a fractional order transfer function is to convert from fractional order pseudo state-space expressions [26]. Consider the pseudo-S-S representation in (EQ.II.66) assuming zero initial conditions, and taking Laplace transformation gives:

$$s^{\alpha}X(s) = AX(s) + BU(s) \qquad (II.69)$$

$$Y\ (s) = CX(s) + DU(s) \qquad (II.70)$$

Then, the transfer function $G(s) = Y\ (s)/U(s)$ can be derived through:

$$G(s) = C(s^{\alpha}I - A)^{-1}B + D \qquad (II.71)$$

If B is a multi-column matrix and/or C is a multi-row matrix, then the resulting $G(s)$ is an FO transfer function matrix rather than a single FO transfer function.

4.6. Observability and Controllability :

Controllability and Observability represent two major concepts of modern control system theory. For the fractional order systems, the following two results can be demonstrated as their similar in the integer order case [31.32].

Theorem II.1 (Observability) :

A commensurable fractional order system of the form (EQ.II.68) is observable if and only if the Observability Matrix given by (EQ.II.72) is a full rank matrix:

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{L-1} \end{bmatrix} \qquad (II.72)$$

Where: L is the number of state variables.

Theorem II.2 (Controllability) :

A commensurable fractional order system represented by the state equation system (EQ.II.68) is controllable if and only if the Controllability Matrix given by (EQ.II.73) has a full rank:

$$[B \quad AB \quad \cdots \quad A^{L-1}B] \qquad (II.73)$$

Where: L is the number of state variables.

4.7. Stability Analysis :

System stability is always a big concern in control theory due to its importance. There are numerous notions and criteria for different kinds of stabilities, such as bounded-input bounded-output (BIBO) stability, exponential stability, asymptotic stability, Lyapunov stability, robust stability, etc. For FO systems, these criteria need to be extended, and new stability types are proposed.

In order to study stability of a fractional system given by (EQ.II.53) we consider the following theorem: [33.34]

Theorem II.3 (Matignon's stability theorem) :

A fractional order transfer function $G(s) = Z(s)/P(s)$ is stable if and only if the following condition is satisfied in σ-plane:

$$|Arg(\sigma)\,| > q\frac{\pi}{2}\,, \qquad \forall\sigma\epsilon C, \qquad P(\sigma) = 0 \qquad (II.74)$$

Where: $0 < q \leq 1$ and $\sigma = s^q$ When $\sigma = 0$ is a single root of $P(s)$, the system cannot be stable. For $q = 1$, this is the classical theorem of pole location in the complex plane: no pole is in the closed right plane of the first Riemann sheet.

The algorithm for checking the stability of the system in (EQ.II.55) can be summarized as follows:

1) Find the commensurate order q of $P(s)$, find $a_1, a_2, \ldots., a_n$ an in (EQ.II.56);

2) Solve for σ the equation $\sum_{k=0}^{n} a_k \sigma^k = 0$.

3) If all obtained roots satisfy the condition in (Theorem II.3), the system is stable.

Stability regions of a fractional-order system are shown in (Figure II.2).

Note that there are currently no polynomial techniques, either Routh or Jury type, to analyse the stability of fractional-order systems.

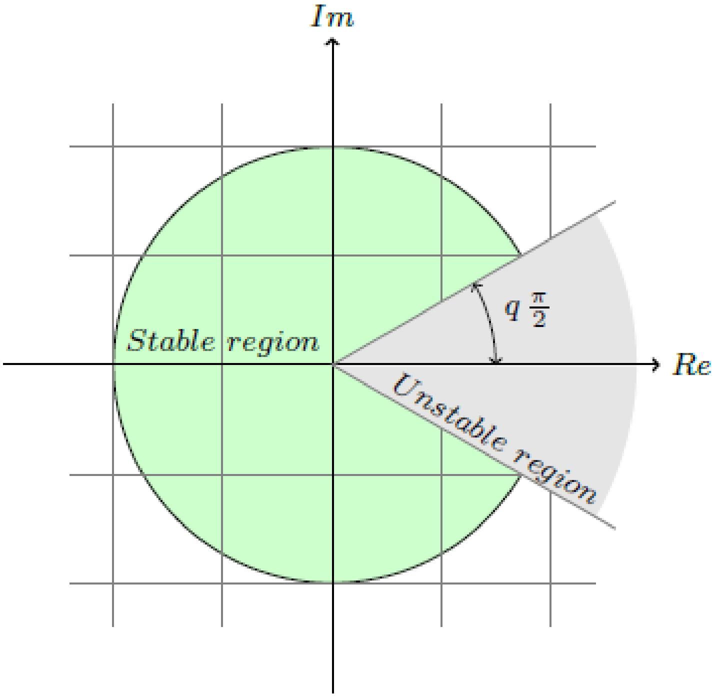


Figure II.2: *LTI fractional-order system stability region for* $0 < q \leq 1$

5. Approximations using curve fitting or identification techniques :

Generally, any method available for frequency domain identification can be applied in order to obtain a rational function, whose frequency response fits the frequency response of the original irrational transfer function [35].

5.1. The Oustaloup method :

Oustaloup method [36] is based on the continuous-time approximation of the fractional order operator $G(s) = s^{\alpha}, \alpha \in R$, by a rational function using a recursive distribution of zeros and poles of non-integer order, distributed in a frequency band limited.

Thus, the approximation of the operator in a frequency band, is given by:

$$G(s) = s^{\alpha} \approx \hat{G}(s) = C_0 \prod_{i=1}^{N} \left(\frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}} \right) \qquad (II.75)$$

with :

$$\begin{cases} si\ \alpha > 0: \begin{cases} z_1 = \omega_b\sqrt{\eta} \\ p_i = z_i\lambda \quad , \qquad i = 1,2,\dots N \\ z_{i+1} = p_i\eta \quad , \qquad i = 1,2,\dots N-1 \\ \omega_h = p_N\sqrt{\eta} \end{cases} \\ si\ \alpha < 0: \begin{cases} p_1 = \omega_b\sqrt{\eta} \\ z_i = p_i\lambda \quad , \qquad i = 1,2,\dots N \\ p_{i+1} = z_i\eta \quad , \qquad i = 1,2,\dots N-1 \\ \omega_h = z_N\sqrt{\eta} \end{cases} \end{cases} \qquad (II.76)$$

$$N = integer\left\{\frac{log\left(\frac{\omega_h}{\omega_b}\right)}{log(\lambda\eta)}\right\}, \quad \lambda = \left(\frac{\omega_h}{\omega_b}\right)^{|\alpha|/N}, \quad \eta = \left(\frac{\omega_h}{\omega_b}\right)^{\left(\frac{1-|\alpha|}{N}\right)} \qquad (II.77)$$

The rational model is then obtained by replacing each fractional order operator of the original system by its rational approximation.

5.2. Charef method :

This method, proposed in [35] which is very close to Oustaloup method, is based on the approximation of a function of the form:

$$H(s) = \frac{1}{\left(1 + \frac{s}{P_T}\right)^{\alpha}} \qquad (II.78)$$

by a quotient of the polynomials in in a factorized form:

$$\hat{H}(s) = \frac{\prod_{i=0}^{n-1}\left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^{n}\left(1 + \frac{s}{p_i}\right)} \qquad (II.79)$$

where the coefficients are calculated to obtain a maximum deviation of the frequency response in dB.

In this method, we define the following parameters:

$$a = 10^{y/_{10(1-\alpha)}}, \quad b = 10^{y/10\alpha}, \quad ab = 10^{y/10\alpha(1-\alpha)} \qquad (II.80)$$

The poles and zeros of the approximate rational function are obtained by applying the following formulas:

$$P_0 = P_T\sqrt{b}, \quad P_i = P_0(ab)^i, \quad z_i = aP_0(ab)^i \qquad (II.81)$$

The number of poles and zeros is related to the desired bandwidth and the error criteria employed by the expression:

$$N = integer\left(\frac{log\left(\frac{\omega_{max}}{p_0}\right)}{log(ab)}\right) + 1 \qquad (II.82)$$

Where: y is the approximation error and is the maximum frequency in the approximation band.

6. Applications of Fractional Calculus :

The concept of fractional calculus has great potential to change the way we see, model and analyse complex systems. We can say that ignoring fractional calculus is just like ignoring fractional, irrational or complex numbers. It provides good opportunity to researchers and engineers for revisiting the origins. The theoretical and practical interests of using fractional order operators are increasing. The application domain of fractional calculus is ranging from accurate modelling of the microbiological processes to the analysis of astronomical images. we will give here the most prominent areas of application of fractional calculus [17]:

6.1. Control Theory and Engineering :

The accuracy and robustness of control systems are becoming imperative these days. The dynamic nature of control systems requires them to be modelled using the fractional calculus. In control engineering the concept of the fractional operations is mostly used in fractional system identification, biomimetic (bionics) control, feedback control systems, trajectory control of redundant manipulators, temperature control, Model Reference based adaptive control, fractional PI^{λ} controller and fractional PD^{α} controllers [17].

6.2. Signal Processing :

In the last decade, the use of fractional calculus in signal processing has tremendously increased. In signal processing, the fractional operators are used in the design of differentiator and integrator of fractional order, fractional order FIR differentiator, IIR type digital fractional order differentiator and for modelling the speech signal [17].

6.3. Image Processing :

In image processing, fractional calculus (fractional differentiation) is used for enhancing image quality, image restoration and edge detection. In particularly fractional calculus is used in satellite image classification and astronomical image processing [17].

6.4. Electromagnetic Theory :

The use of fractional calculus in electromagnetic theory has emerged in the last two decades. In 1998, Engheta introduced the concept of fractional curl operators and this concept is extended by Naqvi. Engheta’s work gave birth to the new field of research in Electromagnetics, namely, ‘Fractional Paradigms in Electromagnetic Theory’. Nowadays fractional calculus is widely used in Electromagnetics to explore new results, for example: Faryad used fractional calculus for the analysis of a Rectangular Waveguide [17].

6.5. Communication :

Chaotic Communication and Chaos synchronization are becoming very popular nowadays. The concept of fractional calculus is recently introduced for secure chaotic communication and very satisfactory results have been achieved. Also concept of fractional calculus are used for informational network traffic modelling [17].

6.6. Probability Theory :

G. Cottone used fractional operators for the probabilistic characterization of random variable [17].

6.7. Biology :

In Biology, fractional calculus is used in neuron modelling, biophysical processes, modelling of complex dynamics of tissues, modelling of infectious diseases [17].

7. Conclusion :

In this chapter, we talked about the fundamental concept of the fractional calculus. we started by the useful mathematical functions then the definition of fractional operator, its proprieties and the Laplace transforms of fractional operator, after that, we talked about the different representation of fractional order systems, its analysis and the applications of fractional calculus.

The aim of this chapter is to identify these basic concepts in order to apply them in control engineering and to preface to explain the fractional order PID controller, which we will talk about it in the next chapter.

Chapter III :

Fractional order controller

1. Introduction :

“PID” is an acronym for “proportional, integral, and derivative.” A PID controller is a controller that includes elements with those three functions. In the literature on PID controllers, acronyms are also used at the element level: the proportional element is referred to as the “P element,” the integral element as the “I element,” and the derivative element as the “D element.” The PID controller was first placed on the market in 1939 and has remained the most widely used controller in process control until today. An investigation performed in 1989 in Japan indicated that more than 90% of the controllers used in process industries are PID controllers and advanced versions of the PID controller. [37]

The fractional order PID controller denoted by $PI^{\lambda}D^{\mu}$, or FO-PID controller, where λ and μ are two additional parameters to the integral and the derivative components of the classical PID controller.

The PID controllers are one of the most important popular controllers used in industry because of their remarkable effectiveness, simplicity of implementation and broad applicability. However, tuning of these controllers is time consuming, not easy and generally lead to poor performance especially with non-linear systems. Many intelligence algorithms are proposed to tuning the PID controllers’ parameters.

There are many methods to tuning classical PID controller, for example, we have Trial and Error Method and Zeigler-Nichols Method (first and second methods), as for Tuning $FOPID$ parameters we mention the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm.

In This chapter we will introducing the PID and $FOPID$ controllers, in this context we will make a historical background for both, with the definition of the common types of these controllers, and some algorithms and methods of tuning and the optimization of their settings. After that we will make a comparison between the Classical PID Controller and the Fractional Order $PI^{\lambda}D^{\mu}$ Controller**.** In the end, we will define the Performance criterion and some of its common types.

2. Feedback Configuration :

Let consider the feedback control system shown in (Figure III.1) [38]:

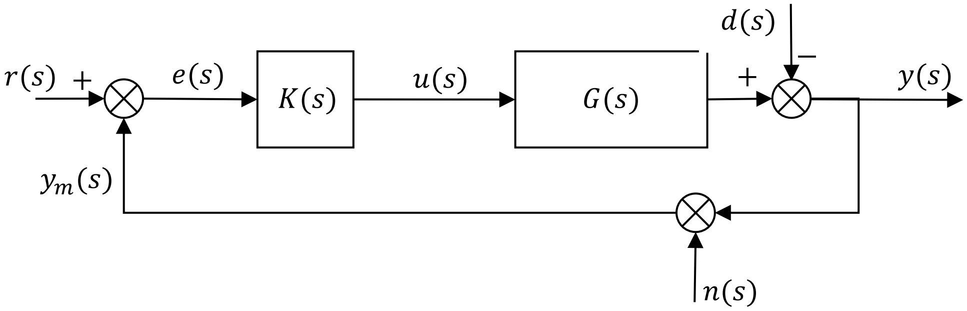


Figure III.1: *Block diagram of one degree-of-freedom feedback control system.*

Where: $\begin{cases} r(s): \textit{is the reference signal} \\ d(s): \textit{is the disturbane signal} \\ n(s): \textit{is the measurement noises signal} \\ y_m(s): \textit{is measured output signal} \\ y(s): \textit{is the output signal} \\ e(s): \textit{is the error signal} \\ u(s): \textit{is the control signal} \\ K(s): \textit{is the controller transfer function} \\ G(s): \textit{is the system transfer function} \end{cases}$

2.1. One degree-of-freedom controller :

The input to the controller: $K(s)$ is $r(s) - y_m(s)$

And:

$$y_m(s) = y(s) + n(s) \qquad (III.1)$$

Thus, the input to the plant is:

$$u(s) = K(s)\big(r(s) - y(s) - n(s)\big) \qquad (III.2)$$

The objective of control is to manipulate $u(s)$ design K such that the control error $e(s)$ remains small in spite of disturbances $d(s)$ The control error $e(s)$ is defined as [38]:

$$e(s) = r(s) - y(s) \qquad (III.3)$$

Note that we do not define $e(s)$ as the controller input $r - y_m$ which is frequently done.

2.2. Close-loop transfer functions :

The plant model is written as follows:

$$y(s) = G(s)u(s) + d(s) \qquad (III.4)$$

And for a one degree-of-freedom controller the substitution of (EQ.III.2) into (EQ.III.4) yields:

$$y(s) = G(s)K(s)(r(s) - y(s) - n(s)) + d(s) \qquad (III.5)$$

$$\left(1 + G(s)K(s)\right)y(s) = G(s)K(s)r(s) - G(s)K(s)n(s) + d(s) \qquad (III.6)$$

And hence the closed-loop response is:

$$y(s) = \frac{G(s)K(s)}{1 + G(s)K(s)}r(s) - \frac{G(s)K(s)}{1 + G(s)K(s)}n(s) + \frac{1}{1 + G(s)K(s)}d(s) \qquad (III.7)$$

We note that:

The sensitivity function:

$$S(s) = \frac{1}{1 + G(s)K(s)} \qquad (III.8)$$

The complementary sensitivity:

$$T(s) = \frac{G(s)K(s)}{1 + G(s)K(s)} \qquad (III.9)$$

It is important to note that $T(s)$ and $S(s)$ are algebraically related by the expression: [38]

$$S(s) + T(s) = 1 \qquad (III.10)$$

So, the closed-loop response can be expressed as:

$$y(s) = T(s)r(s) - T(s)n(s) + S(s)d(s) \qquad (III.11)$$

The control error is:

$$e(s) = r(s) - y(s) = S(s)r(s) + T(s)n(s) - S(s)d(s) \qquad (III.12)$$

where we have used the fact $T(s) - 1 = -S(s)$ The corresponding plant input signal is:

$$u(s) = K(s)S(s)r(s) - K(s)S(s)d(s) - K(s)S(s)n(s) \qquad (III.13)$$

3. Classical PID Controller :

3.1. Historical background of Classical PID Controller :

In 1788, James Watt included a flyball governor, the first mechanical feedback device with only a proportional function, into his steam engine. The flyball governor controlled the speed by applying more steam to the engine when the speed dropped lower than a set point, and vice versa. In 1933, the Taylor Instrumental Company introduced the first pneumatic controller with a fully tuneable proportional controller. However, a proportional controller is not sufficient to control speed thoroughly, as it amplifies error by multiplying it by some constant (K_p). The error generated is eventually small, but not zero. In other words, it generates a steady state error each time the controller responds to the load. Around 1930s, control engineers discovered that steady state error can be eliminated by resetting the set point to some artificial higher or lower value, as long as the error nonzero. This resetting operation integrates the error, and the result is added to the proportional term; today this is known as Proportional-Integral controller. In 1934-1935, Foxboro introduced the first PI controller. However, PI controllers can over-correct errors and cause closed-loop instability. This happens when the controller reacts too fast and too aggressively; it creates a new set of errors, even opposite to the real error. This is known as “hunting” problem. In 1920s, there were suggestions of including the rate of change of error in conjunction with PI controller. In 1940, Taylor Instrument Companies successfully produced the first PID pneumatic controller; the derivative action was called “pre-act”. With an extra derivative action, problems such as overshoot and hunting are reduced. However, issues like finding the appropriate parameter of PID controllers were yet to be solved. In 1942, Taylor Instrument Company's Ziegler and Nichols introduced Ziegler-Nichol’s tuning rules. Their well-known paper “Optimum settings for automatic controllers”, presented two procedures for establishing the appropriate parameters for PID controllers. However, the PID controller was not popular at that time, as it was not a simple concept; the parameters the manufacturers required to be tuned did not make much sense to the users. In the mid 1950's, automatic controllers were widely adopted in industries. A report from the Department of Scientific and Industrial Research of United Kingdom state, “Modern controlling units may be operated mechanically, hydraulically, pneumatically or electrically. The pneumatic type is technically the most advanced and many reliable designs are available. It is thought that more than 90 percent of the existing units are pneumatic.” The report indicated the need to implement controllers in electrical and electronic form. In 1951, The Swartwout Company introduced their first electronic PID controller, based on vacuum tube technology. Around 1957, the manufacturers started to realize the possibility of implementing the controllers in transistors. In 1959, the first solid-state electronic controller was introduced by Bailey Meter Co [39].

3.2. Classical PID Controller :

A PID controller is a simple three terms equation. Two terms are functions of the error between the measured output signal and the reference signal, the third term is just a constant. Each term is multiplied in a parameter [40].

These three parameters are called proportional, integral, and derivative parameters. They are denoted, K_p, K_i, and K_d respectfully. [40]:

3.3. Types Of Classical PID Controller :

There are three most commonly used classical PID controller, namely parallel, ideal or ISA and series or interacting PID controller [41]:

3.3.1. Parallel PID :

proportional, integral and derivative actions are working separately with each other and combine effect of these three actions are act in the system [41].

- The parallel PID controller provides a control effort u(t) given by:

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \qquad (III.14)$$

- The corresponding controller transfer function is defined as the ratio of the controller output U(s) and error E(s) as:

$$C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \qquad (III.15)$$

- The parallel structure of a classical PID controller.

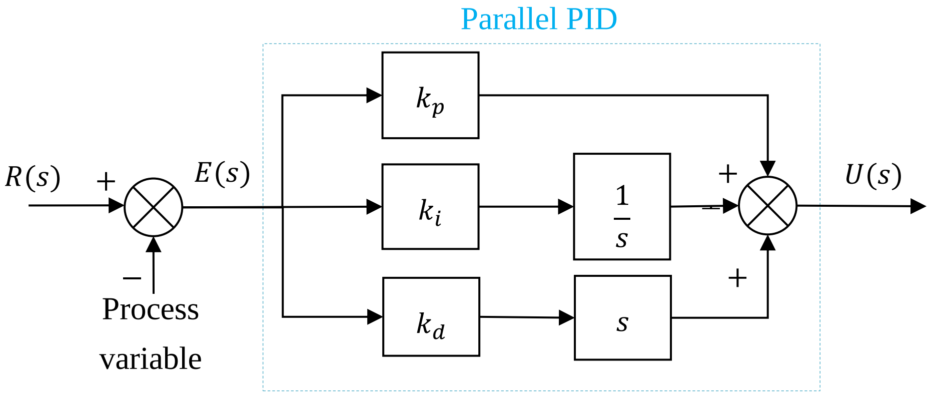


Figure.III.2: *parallel structure of a classical PID controller*

3.3.2. Ideal PID :

The gain constant k_p is distributed to all term. So, changes in kp affects all other terms.

- The ideal PID controller provides a control effort u(t) given by [41]:

$$u(t) = k_p \left(e(t) + T_i \int e(t)dt + T_d \frac{de(t)}{dt} \right) \qquad (III.16)$$

- The corresponding controller transfer function is defined as the ratio of the controller output $U(s)$ and error $E(s)$ as:

$$C(s) = \frac{U(s)}{E(s)} = k_p \left(1 + \frac{T_i}{s} + T_d s \right) \qquad (III.17)$$

- The ideal structure of a classical PID controller.

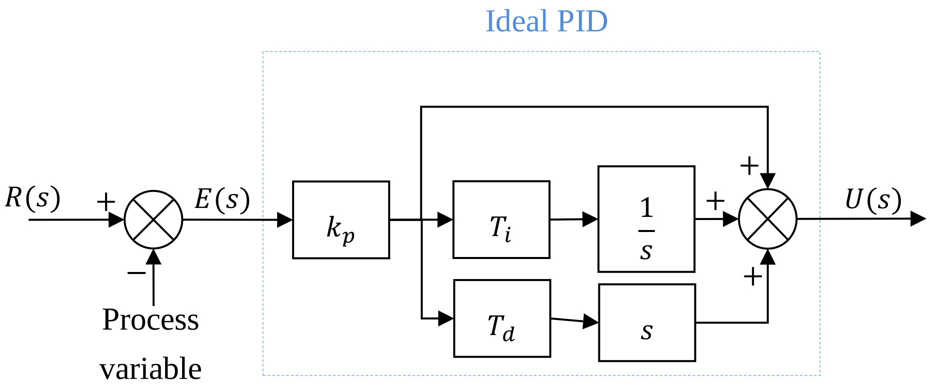


Figure.III.3: *Ideal structure of a classical PID controller*

3.3.3. Series PID :

The gain constant kp is distributed to all terms same as ideal PID, but in this form integral and derivative constant have an effect on proportional action [41].

- The series PID controller provides a control effort u(t) given by:

$$u(t) = k_p \left(e(t) + T_i \int e(t)dt \right) \left(e(t) + T_d \frac{de(t)}{dt} \right) \qquad (III.18)$$

- The corresponding controller transfer function is defined as the ratio of the controller output $U(s)$ and error $E(s)$ as:

$$C(s) = \frac{U(s)}{E(s)} = k_p \left(1 + \frac{T_i}{s} \right) (1 + T_d s) \qquad (III.19)$$

- The series structure of a classical PID controller [41]:

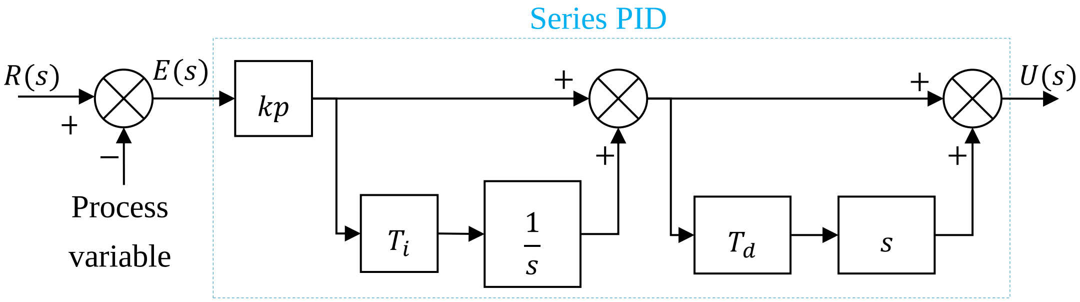


Figure.III.4: *series structure of a classical PID controller*

Remark :

"Approximate derivative." The approximate derivative $D(s)$ is used in place of the pure derivative s, because the latter is impossible to realize physically. In (EQ.II.20), τ is a positive parameter, which is referred to as "derivative gain." The response of the approximate derivative approaches that of the pure derivative as τ increases. It must be noted, however, that the detection noise, which has strong components in the high frequency region in general, is superposed to the detected signal in most cases, and that choosing a large value of τ increases the amplification of the detection noise, and consequently causes malfunction of the controller [39].

This means that the pure derivative is not the ideal element to use in a practical situation. It is usual practice to use a fixed value of τ, which is typically chosen as 10 for most applications. However, it is possible to use τ as a design parameter for the purpose of, for instance, compensating for a "zero" of the transfer function of the process [39].

$$D(s) = \frac{s}{1+\tau s} \qquad (III.20)$$

Hence instead of pure derivative s in (EQ.II.15), (EQ.II.17) and (EQ.II.19) we put (EQ.II.20) to get:

Parallel PID controller with approximate derivative:

$$C(s) = k_p + \frac{k_i}{s} + k_d \frac{s}{1+\tau s} \qquad (III.21)$$

Ideal PID controller approximate derivative:

$$C(s) = k_p \left(1 + \frac{T_i}{s} + T_d \frac{s}{1+\tau s}\right) \qquad (III.22)$$

Series PID controller approximate derivative:

$$C(s) = k_p \left(1 + \frac{T_i}{s}\right)\left(1 + T_d \frac{s}{1+\tau s}\right) \qquad (III.23)$$

3.4. PID Tuning Method :

The determination of corresponding PID parameter values for getting the optimum performance from the process is called tuning. This is obviously a crucial part in case of all closed loop control systems [42].

There are number of tuning methods have been introduced to obtain fast and acceptable performance. The steps involved in these methods include experimental determination of the dynamic characteristics of the control loop and estimating the controller tuning parameters that produce a desired performance for the dynamic characteristics determined. Some of these PID tuning methods are given below [42].

3.4.1. Trial and Error Method :

This is the simple method of tuning a PID controller. Once we get the clear understanding of PID parameters, the trial-and-error method become relatively easy.

- Set integral and derivative terms to zero first and then increase the proportional gain until the output of the control loop oscillates at a constant rate. This increase of proportional gain should be in such that response the system becomes faster provided it should not make system unstable [42].
- Once the P-response is fast enough, set the integral term, so that the oscillations will be gradually reduced. Change this I-value until the steady state error is reduced, but it may increase overshoot [42].
- Once P and I parameters have been set to a desired values with minimal steady state error, increase the derivative gain until the system reacts quickly to its set point. Increasing derivative term decreases the overshoot of the controller response [42].

3.4.2. Zeigler-Nichols Method :

It is another popular method for tuning PID controllers. Ziegler and Nichols presented two classical methods for determining values of proportional gain, integral time and derivative time based on transient response characteristics of a given plant or system [42].

3.4.2.1. First Method :

- Obtain a unit step response of the plant experimentally and it may look ‘s’ shaped curve as shown in figure below. This method applies, if obtained response exhibit s-shaped curve for unit step input otherwise it cannot be applied. This curve can also be obtained by dynamic simulation of the plant [42].

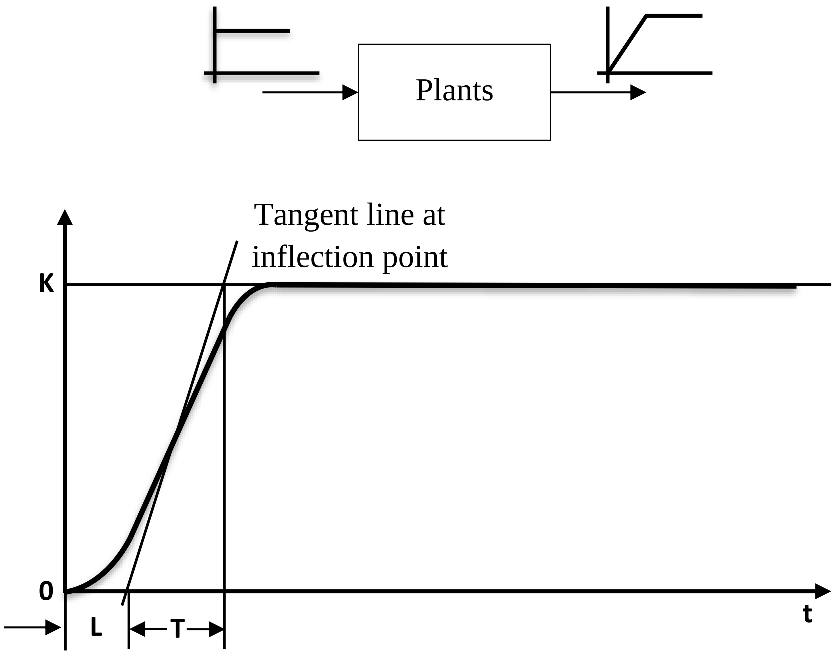


Figure.III.5: *Unit step response of a plant and its s-shaped curve.*

- Obtain two constants, delay time L and time constant T by drawing a tangent line at the inflection point of the s-shaped curve.
- Set the parameters of Kp, Ti, and Td values from the table given below for three types of controllers.

Type of controller	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9\,T/L$	$L/0.3$	0
PID	$1.2\,T/L$	$2L$	$0.5L$

Table.III.1: *The parameters of k_p, T_i and T_d of the P, PI and PID by first method of Ziegler Nichols.*

3.4.2.2. Second Method :

- It is very similar to the trial-and-error method where integral and derivative terms are set to the zero, i.e., making T_i infinity and T_d zero.

- Increase the proportional gain such that the output exhibits sustained oscillations. If the system does not produce sustained oscillations, then this method cannot be applied. The gain at which sustained oscillations produced is called as critical gain.
- Once the sustain oscillations are produced, set the values of Ti and Td as per the given table for P, PI and PID controllers based on critical gain and critical period [51].

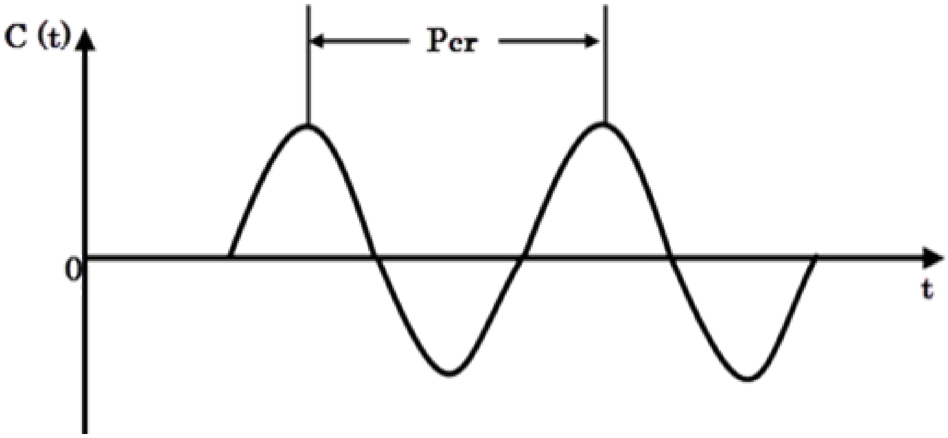


Figure.III.6: *The sustained oscillations with period Pcr.*

Type of controller	K_p	T_i	T_d
P	$0.5\ K_{cr}$	∞	0
PI	$0.45\ K_{cr}$	$1/1.2P_{cr}$	0
PID	$0.6\ K_{cr}$	$0.5\ P_{cr}$	$0.125\ P_{cr}$

Table.III.2: *The parameters of k_p, T_i and T_d of the P, PI and PID by second method of Ziegler Nichols.*

4. Fractional Order $PI^{\lambda}D^{\mu}$ Controller :

4.1. Historical background of Fractional-Order $PI^{\lambda}D^{\mu}$ Control :

The very first theoretical contributions to the field of fractional-order calculus were made by Euler and Lagrange in the 1800s, while Abel was the first to use the fractional-order calculus on an application in 1823. The first systematic studies have been done between 1900 and 1950 by Liouville, Riemann and Holmgren. The nth-order series has been defined by Liouville who has also expanded the functions in a series of exponentials. Riemann presented a different approach which involved a definite integral. After that, Grunwald and Krug unified the results of Liouville and Riemann (Oldham and Spanier, 2006; Miller and Ross, 1993) [43].

The need for solving a major design problem of a feedback amplifier was the key step towards engineers introducing fractional-order calculus methods. Bode presented an elegant solution for design a feedback loop for the amplifier so that the performance of the closed-loop will resist the changes in the gain of the amplifier (Monje *et al.*, 2010). The solution was called "*ideal cut-off characteristic"* by Bode himself, which is known as "*Bode's ideal loop transfer function"* nowadays. The characteristic of this frequency is very useful in the robustness of the system to parameter changes or uncertainties [43].

The step Bode took has encouraged other engineers and curious mathematicians to adopt the concept of fractional-order and helped to motivate new contributions in FOC systems, including both theory and applications. Over the last decades of the twentieth century, there was a growth of the practical application of fractional calculus, mainly in the engineering fields of feedback control, signal processing and system theory [43].

Manabe (1961) introduced a new application of FOC. Oustaloup has studied the algorithms of FOC of the dynamic systems and developed a PID controller called "$CRONE$" (Command Robust d'Ordre non-Entier) which means non-integer order robust control (Oustaloup, 1991). A generalisation of PID control has been presented by Podlubny (Podlubny, 1999). He was the first one to come up with the general form of $PI^{\lambda}D^{\mu}$, where the integrator and the differentiator come with the order of λ and μ, respectively. Also, Podlubny has demonstrated a comparison in terms of response between fractional-order PID against classical PID, as used to control fractional-order systems (Podlubny, 1999). In the next section, we shall understand why we need to use fractional-order controllers instead of conventional integer-order ones [43].

4.2. Fractional order PID controller :

Fractional Order PID controller denoted by $PI^{\lambda}D^{\mu}$ was proposed by Igor Podlubny in 1997.It is an extension of Conventional PID Controller where λ and μ have fractional values.

4.3. Type Of Fractional Order $PI^{\lambda}D^{\mu}$ Controller :

There are two most commonly used Fractional Order PID controller, namely parallel and series $FOPID$ controller:

4.3.1. Series FO-PID:

The series FO-PID controller called also SFO-PID becomes the general form the classical series PID controller [44.45].

- The SFO-PID controller provides a control effort $u(t)$ given by [53.54]:

$$u(t) = k_p e(t) + k_p k_i D^{-\lambda} e(t) + k_p k_d D^{\mu} e(t) + k_p k_i k_d D^{-\lambda+\mu} e(t) \qquad (III.24)$$

- The corresponding controller transfer function is defined as the ratio of the controller output $U(s)$ and error $E(s)$ as [44.45]:

$$C(s) = \frac{U(s)}{E(s)} = k_p + k_p k_i s^{-\lambda} + k_p k_d s^{\mu} + k_p k_i k_d s^{-\lambda+\mu} \qquad (III.25)$$

Or

$$C(s) = \frac{U(s)}{E(s)} = k_p\left(1 + k_i s^{-\lambda}\right)(1 + k_d s^{\mu}) \qquad (III.26)$$

Where: $\begin{cases} \lambda, \mu\text{: are positive real numbers} \\ k_p, k_i, k_d\text{ : are the controller gains.} \end{cases}$

- The series structure of a fractional order PID controller [53.54]:

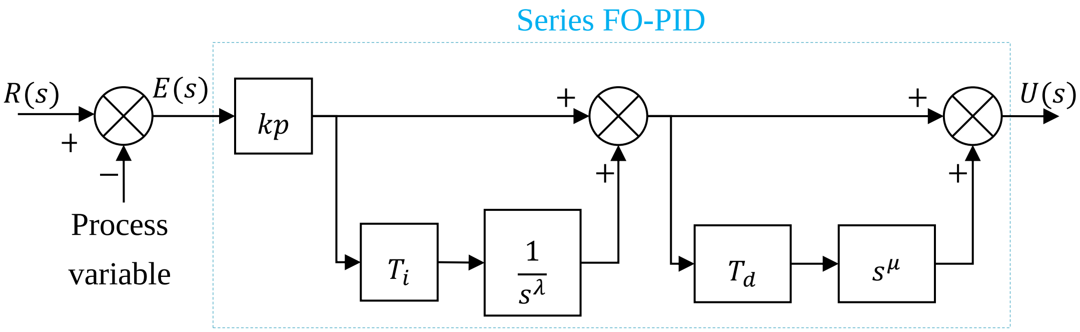


***Figure.III.7:** series structure of a fractional order PID controller*

4.3.2. Parallel FO-PID :

The parallel FO-PID controller called also PFO-PID is the general case of the classical parallel integer one. In time domain [17].

- The controller provides a control effort $u(t)$ given by [17]:

$$u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^{\mu} e(t) \qquad (III.27)$$

• The corresponding controller transfer function is defined as the ratio of the controller output $U(s)$ and error $E(s)$ as [17]:

$$C(s) = \frac{U(s)}{E(s)} = k_p + k_i s^{-\lambda} + k_d s^{\mu} \qquad (III.28)$$

Where: $\begin{cases} \lambda, \mu: \text{ are positive real numbers} \\ k_p, k_i, k_d : \text{are the controller gains.} \end{cases}$

• The parallel structure of a fractional order PID controller [17]:

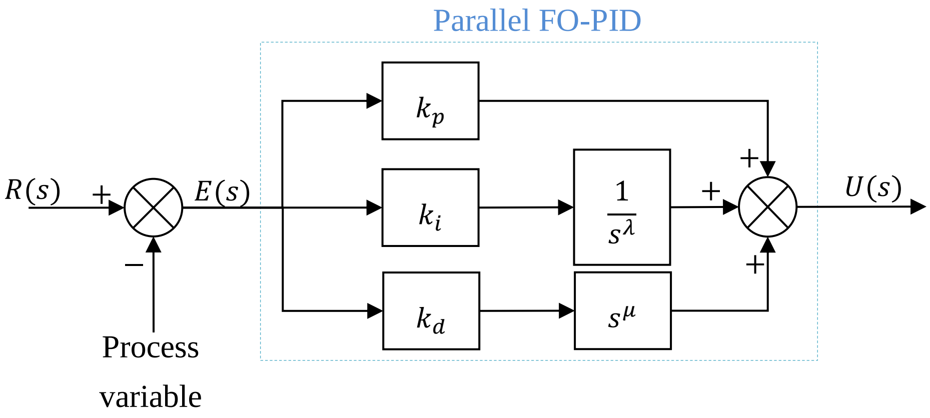


Figure.III.8: *parallel structure of a fractional order PID controller*

4.4. Fractional Order $PI^{\lambda}D^{\mu}$ tuning :

In recent years, automatic control system has assumed an increasingly important role in the development and advancement of modern civilization and technology [46].

Fractional PID controller is one of the most important controllers popular in the automatic system. The setting and optimization of PID parameters are always the important study topics in the automatic control field. The control effect depends on proportion differential and integral which are the parameters of PID controller. Original optimization method is a time-consuming method and cannot get satisfied control effect.

Many intelligence algorithms are proposed to tuning the $FOPID$ parameters. Tuning $FOPID$ parameters by the optimal algorithms such as the Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) algorithm. However, it is slow to search the best solution. (PSO), first introduced by Kennedy and Eberhart, is one of the modern metaheuristic's algorithms. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous non-linear optimization problems [46].

4.4.1. Particle Swarm Optimization (PSO) :

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 (Eberhart, and Kennedy, 1995). The inspiration underlying the development of this algorithm was the social behaviour of animals, such as the flocking of birds and the schooling of fish, and the swarm theory. It has been proven to be efficient in solving optimization problem especially for nonlinearity and non-differentiability, multiple optimums and high dimensionality (Chang and Shih, 2010; Chiou and Liu, 2009) [47].

➢ The PSO concept consists of changing the velocity (or accelerating) of each particle toward its p_{best} and the g_{best} position at each time step. Each particle tries to modify its current position and velocity according to the distance between its current position and p_{best}, and the distance between its current position and g_{best} as shown in the following. At each step n, by using the individual best position, p_{best}, and global best position, g_{best}, a new velocity for the i^{th} particle is updated by [48.49]:

$$V_i(n) = w * V_i(n-1) + c_1 r_1 \left(P_{best_i} - P_i(n-1)\right) + c_2 r_2 \left(g_{best_i} - P_i(n-1)\right) \qquad (III.29)$$

Where :

w: is called the constriction factor.

P_i: the position vector.

r_1,r_2: are random numbers between [0 ; 1].

c_1,c_2: are positive constant learning rates, called self-confidence and swarm confidence respectively [50].

Each of the three terms of the velocity update equation has different roles in the PSO algorithm:

• The first term $w * V_i(n)$ is the inertia component, responsible for keeping the particle moving in the same direction it was originally heading. The value of the inertial coefficient w is typically between 0.8 and 1.2, which can either dampen the particle's inertia or accelerate the particle in its original direction (Yuhui and Russell). Generally, lower values of the inertial coefficient speed up the convergence of the swarm to optimal, and higher values of the inertial coefficient encourage exploration of the entire search space [51].

• The second term $c_1r_1\ (P_{best_i} - P_i(n-1))$, called the cognitive component, acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. The cognitive coefficient c_1 is usually close to 2, and affects the size of the step the particle takes toward its individual best candidate solution p_{best_i} [51].

• The third term $c_2r_2(g_{best_i} - P_i(n-1))$, called the social component, causes the particle to move to the best region the swarm has found so far. The social coefficient c_2 is typically close to 2, and represents the size of the step the particle takes toward the global best candidate solution g_{best_i} the swarm has found up until that point [51].

• The random values r_1 in the cognitive component and r_2 in the social component cause these components to have a stochastic influence on the velocity update. This stochastic nature causes each particle to move in a semi-random manner heavily influenced in the directions of the individual best solution of the particle and global best solution of the swarm [51].

• In order to keep the particles from moving too far beyond the search space, we use a technique called velocity clamping to limit the maximum velocity of each particle (Frans, 2001). For a search space bounded by the range $[p_{min}, p_{max}]$, velocity clamping limits the velocity to the range $[V_{min}, V_{max}]$, where $V_{max} = k \times (p_{max} - p_{min})/2$. The value represents a user-supplied velocity clamping factor, $0.1 \leq k \leq 1$ [51].

➢ Based on the updated velocity, each particle changes its position as following:

$$P_i(n) = P_i(n-1) + V_i(n) \qquad (III.30)$$

• The position is confined within the range of $[p_{min}, p_{max}]$. If the position violates these limits, it is forced to its proper values. Changing position by this way enables the i^{th} particle to search around its local best position, p_{best}, and global best position, g_{best} [51].

$$P_i = \begin{cases} P_{min} & if \quad P_i < P_{min} \\ P_i & if \quad P_{min} < P_i < P_{max} \\ P_{max} & if \quad P_i > P_{max} \end{cases} \qquad (III.31)$$

The PSO is an algorithm with population. It starts with a random initialization of the swarm in the space of research. With each iteration of the algorithm, each particle is moved according to the equations of motion given by (EQ.III.29) and (EQ.III.30).

4.4.1.1. Flow chart of PSO algorithm

The following flow chart, summarize the steps for implementing the PSO algorithm:

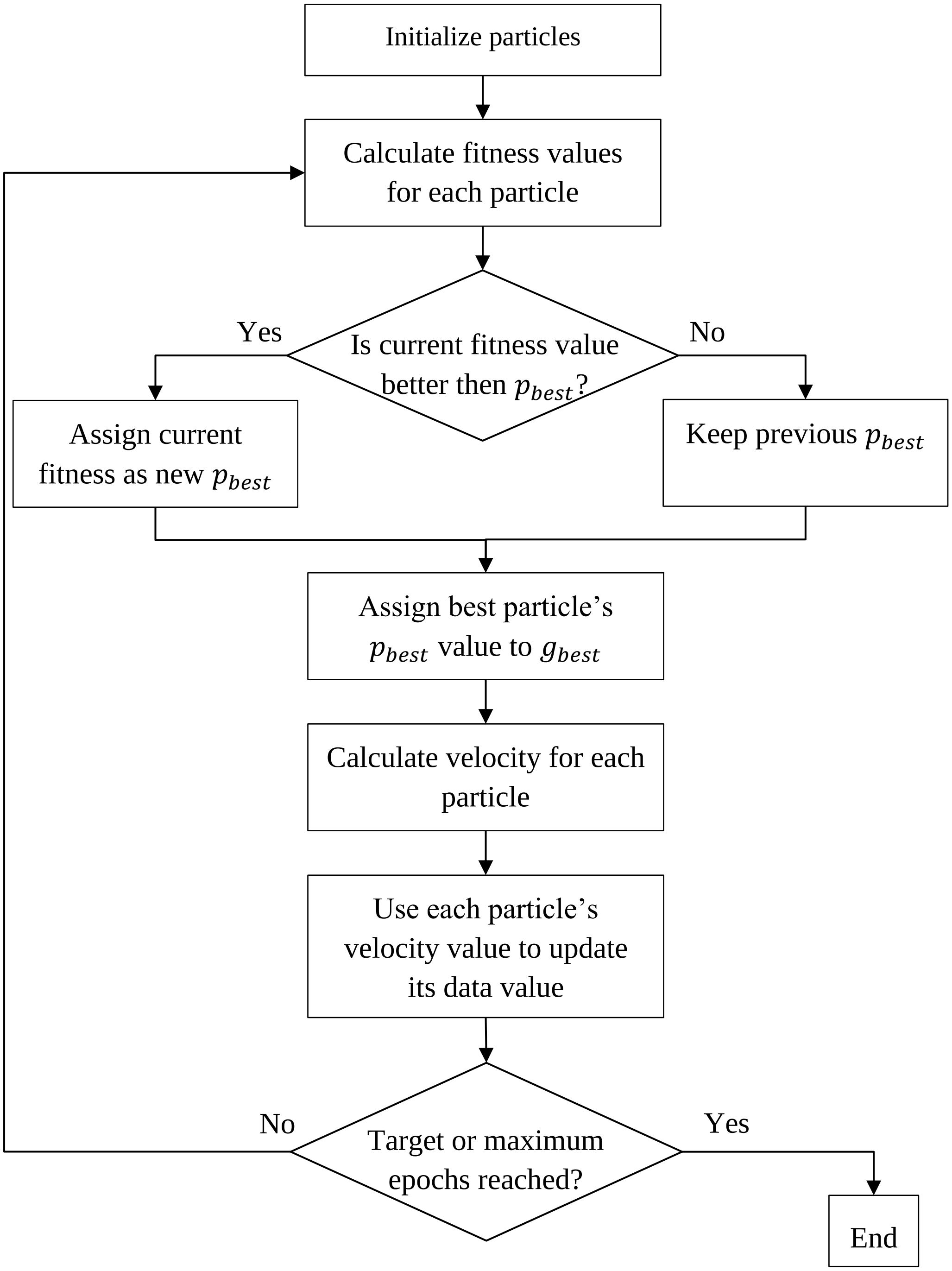

Initialize particles
Calculate fitness values for each particle
Is current fitness value better then p_{best}?
Yes
No
Assign current fitness as new p_{best}
Keep previous p_{best}
Assign best particle's p_{best} value to g_{best}
Calculate velocity for each particle
Use each particle's velocity value to update its data value
Target or maximum epochs reached?
No
Yes
End

Figure.III.9: *flow chart of PSO algorithm.*

4.4.1.2. Performance criterion :

The transient error can be used to characterize the quality of a given servo system.

The quality evaluation of a given servo system is made by evaluating a quality criterion or performance index.

The evaluation is mainly based on the step response of the servo system under consideration.

Among the performance indices often used, we find :

- The criterion of ISE (Integral of Square Error).
- The criterion of IAE (Integral of Absolute Error).
- The criterion of $ITAE$ (Integral of Time Multiplied Absolute Error).
- The criterion of $ITSE$ (Integral of Time Multiplied Square Error).

In practice, the steady-state error is never zero. Integrating a function of this error over an infinite interval gives the performance index used an infinite value. To overcome this drawback, the integration is often done over a finite interval [0, T], whose upper limit T of the interval is chosen to be greater than the response time of the system [52].

4.4.1.2.1. The ISE criterion:

This performance criterion is the most popular. It is defined by:

$$J_{ISE} = \int_0^T e^2(t)dt \qquad (III.32)$$

The presence of $e^2(t)$ in the criterion allows the highlighting of transient differences of high amplitude [52].

4.4.1.2.2. The IAE criterion :

The IAE performance criterion is defined by:

$$J_{IAE} = \int_0^T |e(t)|\, dt \qquad (III.33)$$

This criterion is especially used when the controlled system has a little oscillating transient answer with an average damping [52].

4.4.1.2.3. The $ITAE$ criterion :

It is sometimes advantageous to highlight the values at the end of the transient state. The $ITAE$ performance benchmark allows for this. Such a criterion is defined by [52]:

$$J_{ITAE} = \int_0^T t\,|e(t)|dt \qquad (III.34)$$

4.4.1.2.4. The $ITSE$ criterion :

The $ITSE$ performance criterion makes it possible to slightly weight the beginning of the transient state and simultaneously strongly weight the error values at the end of the transient state. Such a criterion is defined by [52]:

$$J_{ITSE} = \int_0^T te^2(t)dt \qquad (IV.35)$$

5. Comparison between Classical PID and Fractional Order $PI^{\lambda}D^{\mu}$ Controller :

The FO-PID can be seen as a generalization of the PID controller from "points to a plane". (Figure III.2), shows the schematic representation of the PID and the FO-PID controller on the λ, μ plane [17].

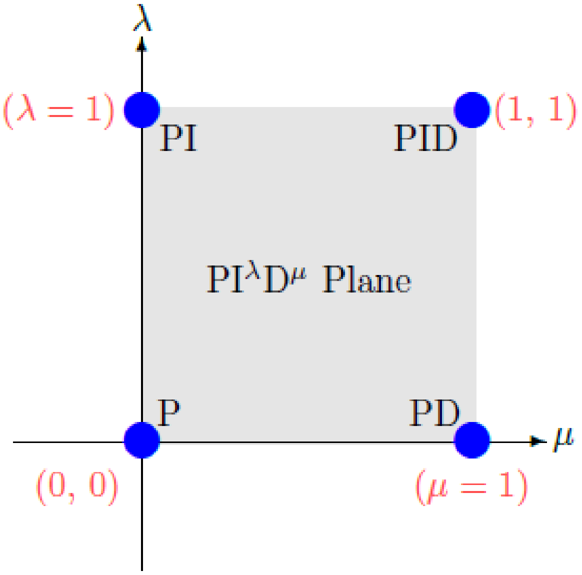


Figure III.10: *Generalisation of the PID controller from point to plane.*

From the figure we obtain particular cases of the $PI^{-\lambda}D^{\mu}$ controller:

- If $\lambda = 0$ and $\mu = 0$ we obtain classical P controller.
- If $\lambda = 1$ and $\mu = 0$ we obtain classical PI controller.
- If $\lambda = 0$ and $\mu = 1$ we obtain classical PD controller.
- If $\lambda = 1$ and $\mu = 1$ we obtain classical PID controller.
- If $0 < \lambda < 1$ and $\mu = 0$ we obtain $FO - PI$ controller.
- If $\lambda = 0$ and $0 < \mu < 1$ we obtain $FO - PD$ controller.
- If $0 < \lambda < 1$ and $0 < \mu < 1$ we obtain $FO - PID$ controller.

It is evident that the P, PI, PD and PID controllers are just four points on the plane and the FO-PID controller can have any value in the plane. Thus, the designer has a higher degree of freedom and can use these additional tuning knobs to fine tune his controller design for specific applications [17].

It is quite natural to conclude that by introducing more general control actions of the form $PI^{\lambda}D^{\mu}$, one could achieve more satisfactory performances between positive and negative effects of classical PID, and combining the fractional order actions one could develop more powerful and flexible design methods to satisfy the controlled system specifications. In other words, the generalization to non-integer-orders of traditional controllers or control schemes translates into more tuning parameters and more adjustable time and frequency responses of the control system, allowing the fulfilment of robust performance [17].

6. Conclusion :

We introduced the PID and $FOPID$ controllers. Where, this chapter included some sub-titles represented in: A historical background of the classical PID and fractional order $FOPID$ controllers, with the definition of the common types of these controllers, we also presented some algorithms and methods of tuning and the optimization of their settings, for example, we mentioned the (ZN) method and (PSO) algorithm. After that, we made a comparison between Classical PID Controller and Fractional Order $PI^{\lambda}D^{\mu}$ Controller**.** In the end, we define the Performance criterion and some of its common types were presented، from which we mentioned the following: $ITAE, IAE, ISE, ITSE$.

The aim of this chapter is to introduce the controllers (PID controller and $FOPID$ controller) as well as their basics and types.

As for the methods and algorithms for tuning and improving their settings is for the search for ways to improve the performance of the system for which they were developed.

Chapter IV :

Tuning of fractional order PID controller by PSO algorithm for DC motor speed regulation

1. Introduction :

DC motor is a power actuator, which converts electrical energy into rotational mechanical energy. DC motors are widely used in industry and commercial application such as tape motor, disk drive, robotic manipulators and in numerous control applications. Therefore, its control is very important. In this chapter, DC motor is controlled by a fractional order PID (FOPID) control. The parameters of the FOPID controller are optimally learned by using particle swarm optimization (PSO), and the optimization performance target is chosen as the Integral Time Absolute Error (ITAE). Their performance and robustness the FOPID controller are compared in time domain by the classical PID controller.

2. Modelling of the used DC motor :

We have a DC motor characterised by: armature resistance $\mathrm{R_a} = 1.5\ \Omega$, armature inductance $\mathrm{L_a} = 0.2$ H, equivalent moment of inertia $\mathrm{J} = 0.0988\ \mathrm{kg.m^2/s^2}$, viscous friction coefficient $\mathrm{B} = 0.000587\ \mathrm{N.m.s/rad}$, back emf constant $\mathrm{K_e} = 0.67609\ \mathrm{V/rad/s}$, torque constant $\mathrm{K_t} = 0.67609\ \mathrm{N.m/A}$ and nominal voltage $\mathrm{V_a} = 120$ v [53].

2.1. Stat space :

From (EQ.I.33) we get:

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -0.00594 & 6.84302 \\ -3.38045 & 7.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & -10.12 \\ 5 & 0 \end{bmatrix} \begin{bmatrix} v_a(t) \\ T_L(t) \end{bmatrix} \\ y(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_a(t) \\ T_L(t) \end{bmatrix} \end{cases} \quad (IV.1)$$

2.2. Transfer function :

1. The transfer function relating the armature voltage $\mathrm{V_a(s)}$ and angular velocity $\Omega(\mathrm{s})$, with $\mathrm{T_L(s)} = 0$:

$$\frac{\Omega(s)}{V_a(s)} = \frac{0.67609}{0.01976s^2 + 0.14832s + 0.45798} \quad (IV.2)$$

2. The transfer function relating the load torque $\mathrm{T_L(s)}$ and angular velocity $\Omega(\mathrm{s})$, with $\mathrm{V_a(s)} = 0$:

$$\frac{\Omega(s)}{T_L(s)} = -\frac{0.2s + 1.5}{0.01976s^2 + 0.14832s + 0.45798} \quad (IV.3)$$

2.3. Block diagram :

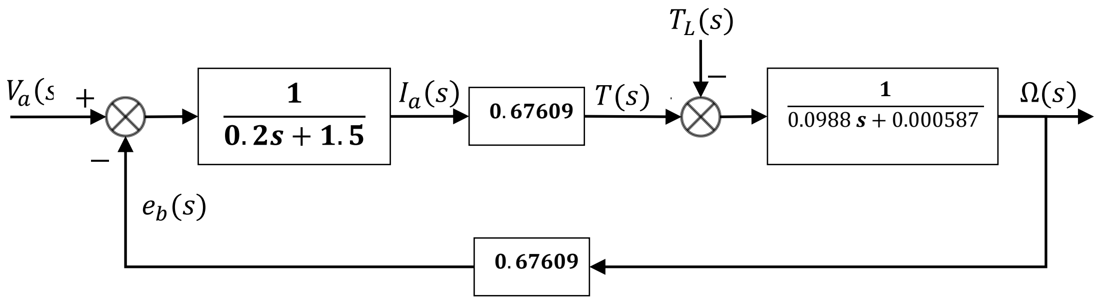


Figure IV.1: *block diagram of the used DC motor.*

3. Tuning of FO-PID by PSO algorithm :

Optimization by *PSO* consists of designing the optimization goal, i.e., the fitness function and then encoding the parameters to be searched. The *PSO* algorithm runs until the stop condition is satisfied. The best particle's position gives the optimized parameters.

The Parameters to be Optimized The $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller has five unknown parameters to be tuned, viz. $\{\mathrm{K_p}, \mathrm{K_i}, \mathrm{K_d}, \lambda, \mu\}$. Hence the present problem of controller tuning can be solved by an application of the *PSO* algorithm for optimization on a five-dimensional solution space, each particle having a five-dimensional position and velocity vector.

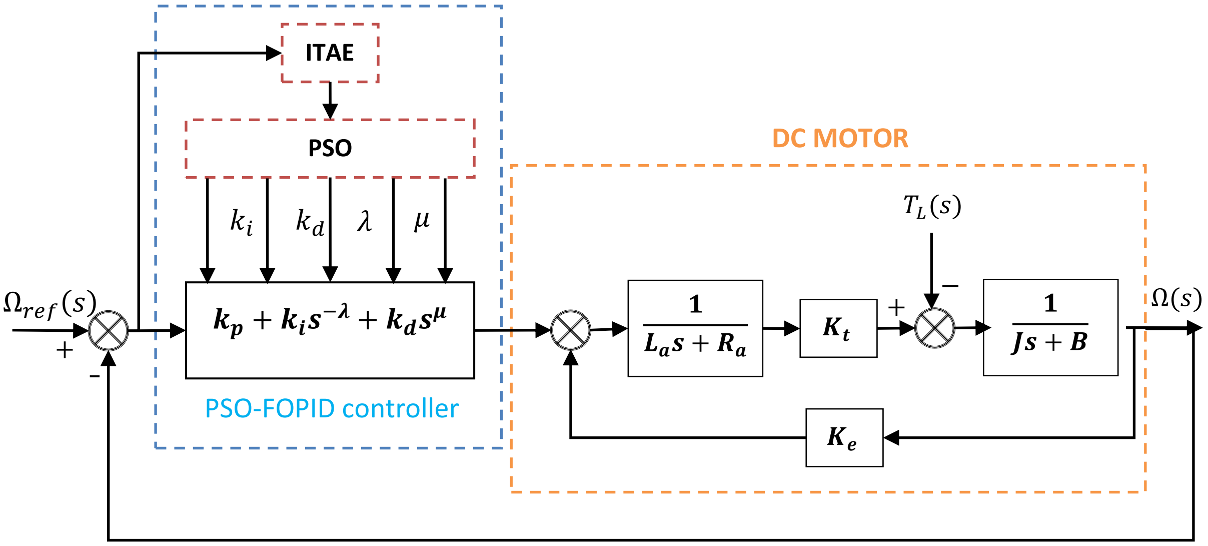


Figure IV.2: *Feedback control system based on PSO-FOPID controller and DC motor model.*

3.1. PSO Factors :

We use this factor: number of populations 50, the constriction factor $\mathrm{w} = 0.73$, self-confidence $\mathrm{c}_1 = 2.05$ and swarm confidence $\mathrm{c}_2 = 2.05$.

The velocity for the i^{th} particle is updated by:

$$\mathrm{V_i(n) = 0.73 * V_i(n-1) + 2.05 * r_1 \left(P_{best_i} - P_i(n-1)\right) + 2.05 * r_2 \left(g_{best_i} - P_i(n-1)\right)} \qquad (IV.4)$$

The position for the i^{th} particle is updated by:

$$\mathrm{P_i(n) = P_i(n-1) + V_i(n)} \qquad (\mathrm{IV.5})$$

The limits on positions and velocities of the parameters are summarized:

$Parameters$	$Limit\ of\ position\ vector$ $(LB\ to\ UB)$	$Limit\ of\ Velocity\ vector$ $(LB\ to\ UB)$
$\mathrm{K_p}$	0.001 to 50	-1 to 1
$\mathrm{K_i}$	0.001 to 100	-1 to 1
$\mathrm{K_d}$	0.001 to 10	-1 to 1
λ, μ	0.001 to 0.99	-1 to 1

Table IV.1: *Fractional order controller ranges of particles.*

3.2. Objective function :

As already mentioned, the fitness function to be minimized is the ITAE performance criterion. The integral of the absolute magnitude of error (ITAE) criterion is defined as

$$\mathrm{ITAE} = \int_0^{\mathrm{T}} \mathrm{t}|\,\mathrm{e(t)}|\,\mathrm{dt} \qquad (IV.6)$$

The $ITAE$ performance index has the advantages of producing smaller overshoots and oscillations than the IAE (integral of the absolute error) or the ISE (integral square error) performance indices. In addition, it is the most sensitive of the three, i.e., it has the best selectivity. The $ITSE$ (integral time-square error) index is somewhat less sensitive and is not comfortable computationally [54.55]. Since it is not practicable to integrate up to infinity, the convention is to choose a value of T sufficiently large so that $\mathrm{e(t)}$ for $\mathrm{t} > \mathrm{T}$ is negligible. We used $\mathrm{T} = 10$ seconds.

3.3. Stop Criterion :

The stop criterion used was the one that defines 15 number of generations or fitness value less than 0.1 to be produced.

3.4. Flow chart of tuning of FO-PID by PSO algorithm :

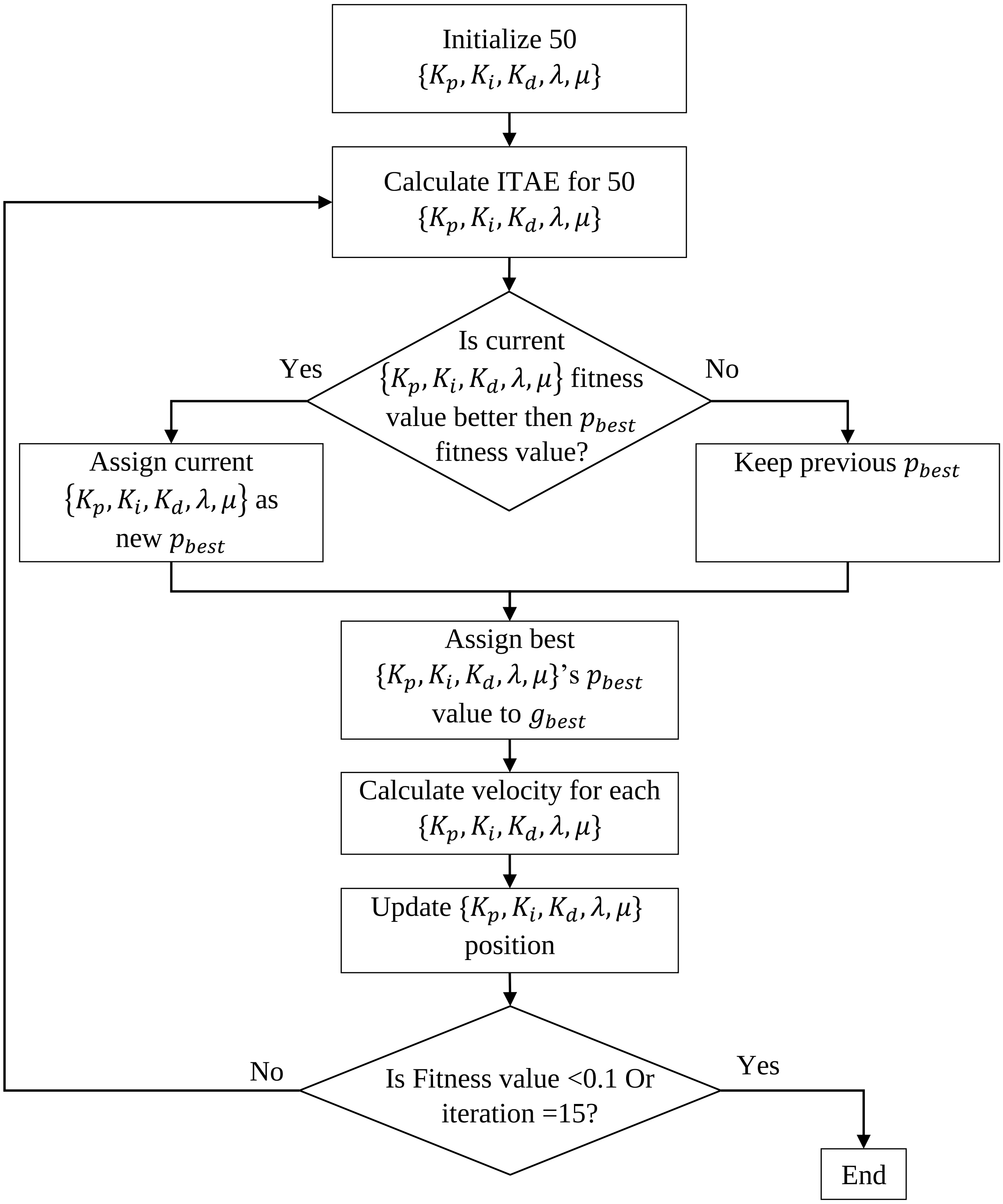

Initialize 50 $\{K_p, K_i, K_d, \lambda, \mu\}$
Calculate ITAE for 50 $\{K_p, K_i, K_d, \lambda, \mu\}$
Is current $\{K_p, K_i, K_d, \lambda, \mu\}$ fitness value better then p_{best} fitness value?
Yes
No
Assign current $\{K_p, K_i, K_d, \lambda, \mu\}$ as new p_{best}
Keep previous p_{best}
Assign best $\{K_p, K_i, K_d, \lambda, \mu\}$'s p_{best} value to g_{best}
Calculate velocity for each $\{K_p, K_i, K_d, \lambda, \mu\}$
Update $\{K_p, K_i, K_d, \lambda, \mu\}$ position
Is Fitness value <0.1 Or iteration =15?
No
Yes
End

***Figure* IV.3:** The *flow chart of FO-PID tuning by PSO algorithm.*

4. Result of PSO algorithm :

For 15 iterations the results are fixed on the optimal parameters of the $FOPID$ controller given by (table IV.2), (Figure IV.4) shown the fitness value in terms of iterations number, also (Figure IV.5) shown the FOPID parameters in terms of iterations number.

K_p	K_i	K_d	λ	μ
24.7288	100	3.8088	0.99	0.99

Table IV.2: *Optimal parameters of the FOPID controller.*

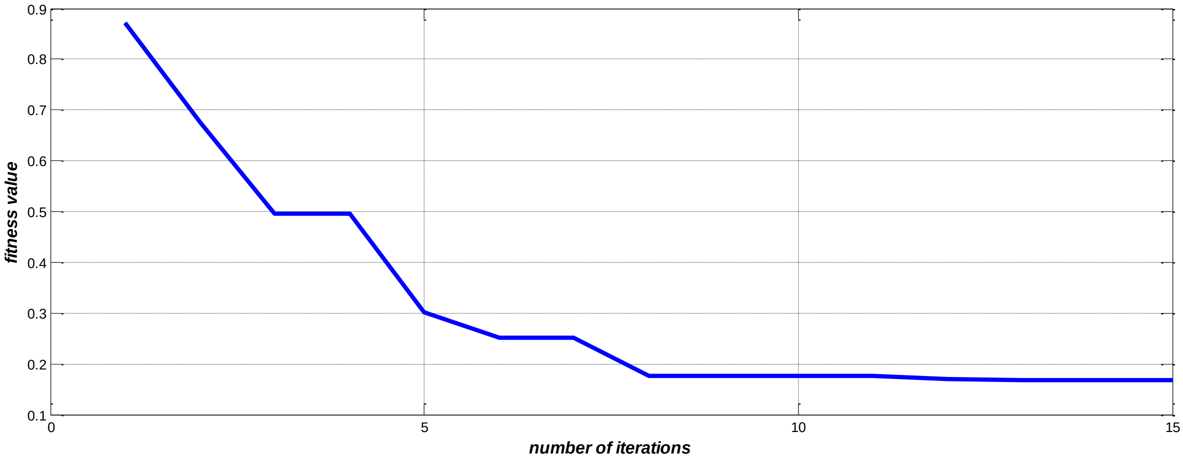
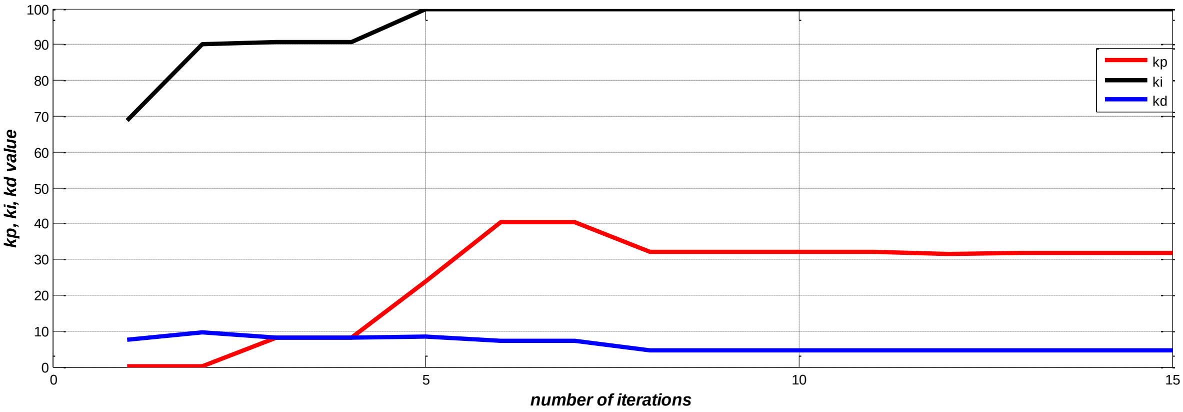
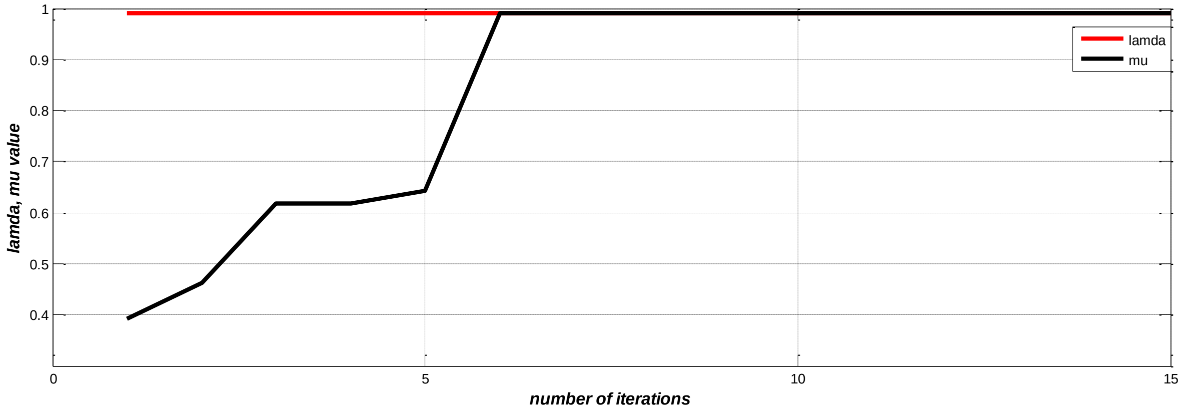


***Figure* IV.4:** *The best provided minimization using the PSO of the fitness value.*



(a)



(b)

***Figure* IV.5. ((a) and (b)):** *Parameters of the FOPID controller in term of iteration.*

So, $FOPID$ controller transfer function is:

$$C_{FOPID}(s) = 24.7288 + 100s^{-0.99} + 3.8088s^{0.99} \qquad (IV.7)$$

5. Simulation and results :

We done in Simulink the following feedback control system contain the DC motor model and the $FOPID$ controller its optimal parameters given by (EQ.IV.7).

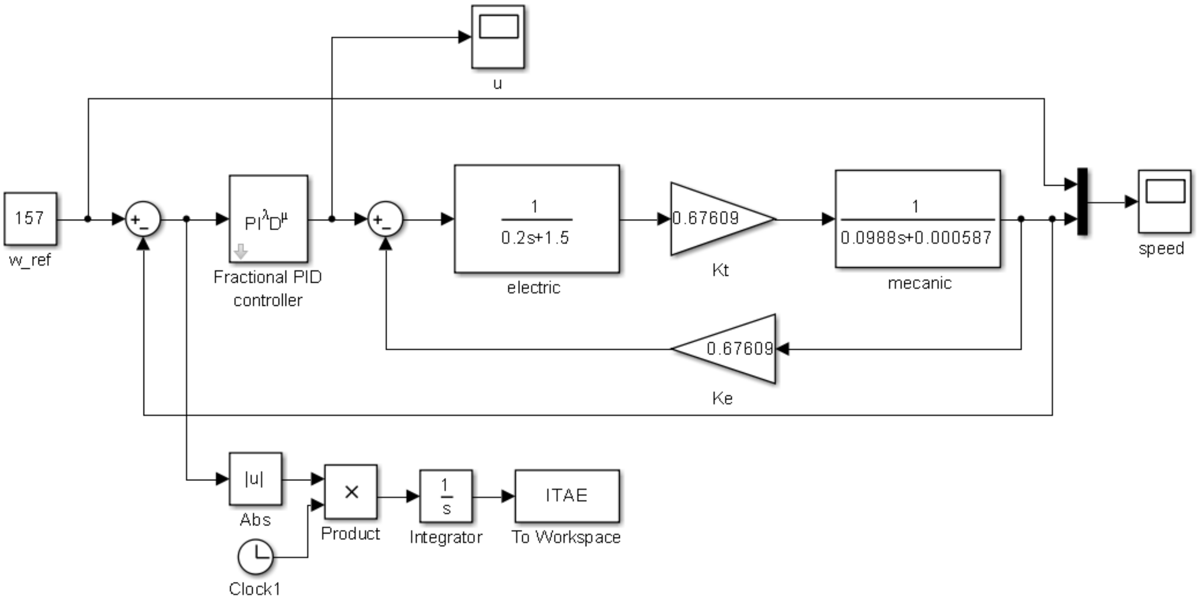


***Figure* IV.6:** *Feedback control system based on the FOPID controller and DC motor model.*

5.1. Tracking :

Each motor has a reference speed it rotates, but in some special cases we need to change this speed to a desired speed while trying to maintain the motor's performance at a good pace.

In this part, we will try to change the speed of DC motor into several forms (constant signal, rectangular signal, sinusoidal signal and triangular signal).

5.1.1. Tracking echelon signal reference speed :

The nominal reference speed of this motor is 157 rad/sec (120 is the supply voltage).

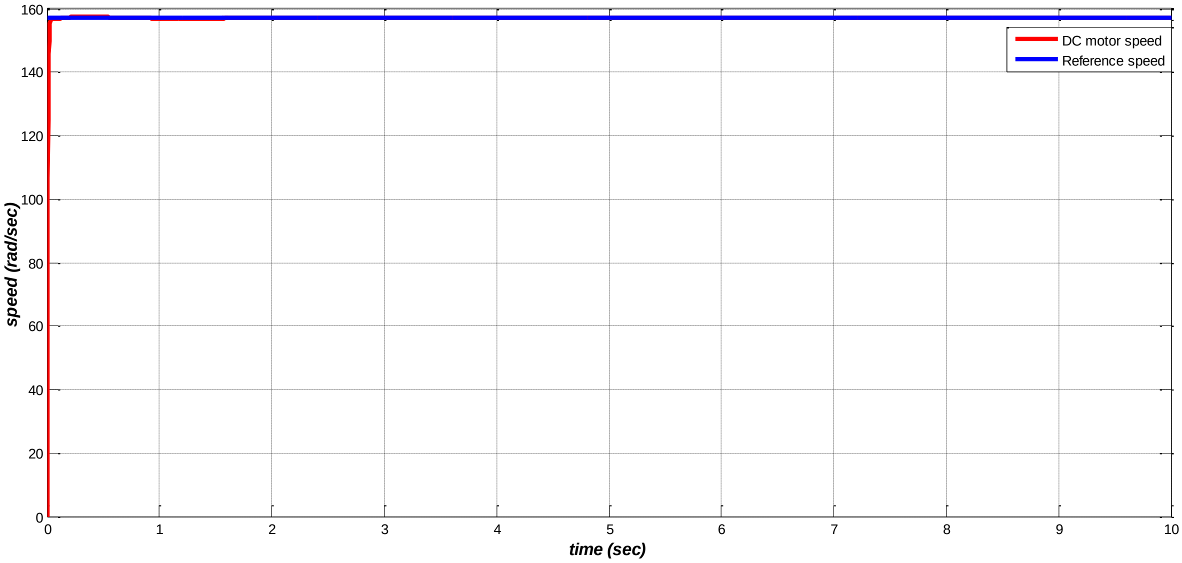


***Figure* IV.7:** *The given speed for reference speed 157 rad/sec.*

Observation :

From the (Figure IV.7), we conclude that the $FOPID$ controller allows tracking the reference speed with high accuracy.

5.1.2. Tracking rectangular signal reference speed :

Used to excite the feedback control system. The input is assumed by:

$$\omega_{ref} = \begin{cases} +157\ rad/ssec & : 0\ \ sec \leq t < 2.5\ \ sec\ \ and \quad 5.5\ \ sec < t \leq 10\ sec \\ +167\ rad/sec & : 2.5\ \ sec \leq t \leq 5.5\ sec \end{cases}$$

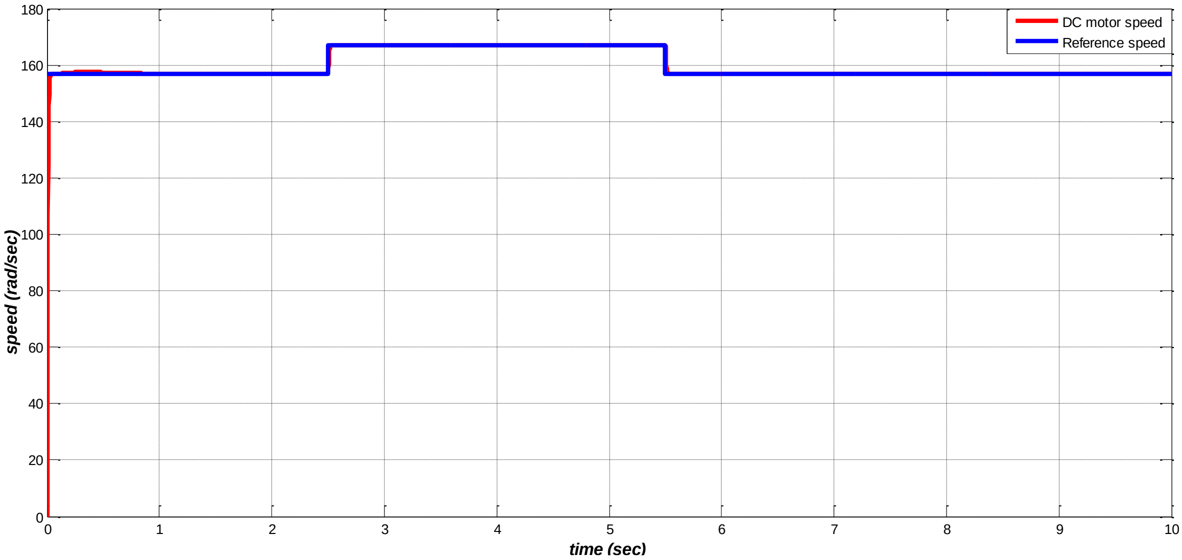


***Figure* IV.8:** *The given speed for reference speed rectangular signal.*

5.1.3. Tracking sinusoidal signal reference speed :

The reference speed input is assumed to be a sinusoidal signal with the amplitude $A = 157$ and the period $T = 3.14\ sec$.

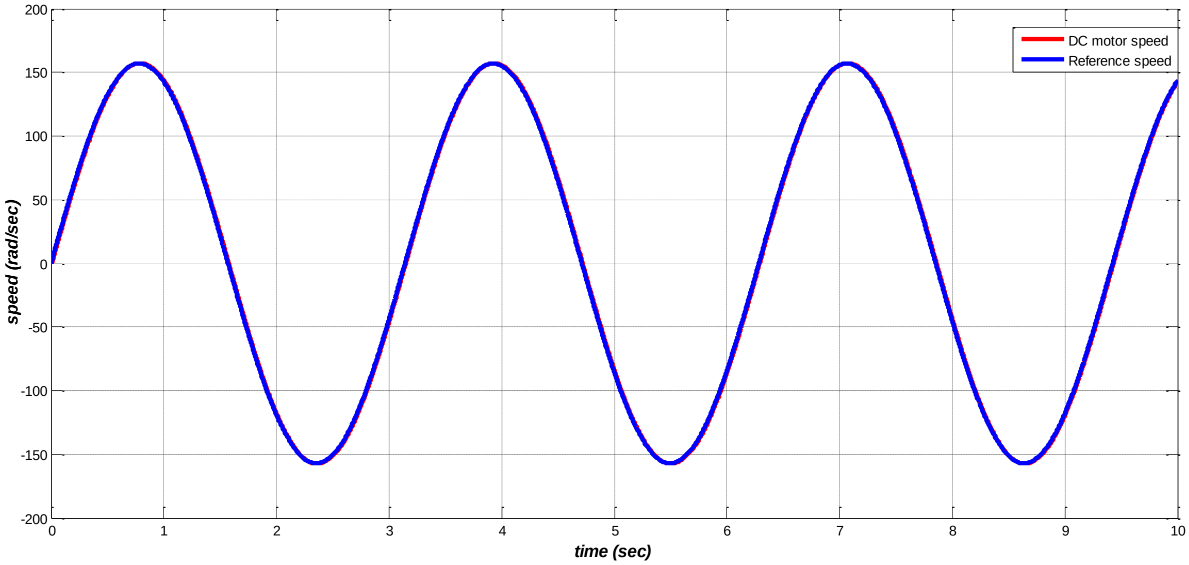


***Figure* IV.9:** *The given mechanical speeds for the reference speed sinusoidal signal.*

5.1.4. Tracking triangular signal reference speed :

The second reference speed input is assumed to be a triangular signal with the amplitude $A = 157$ and the period $T = 2\ sec$.

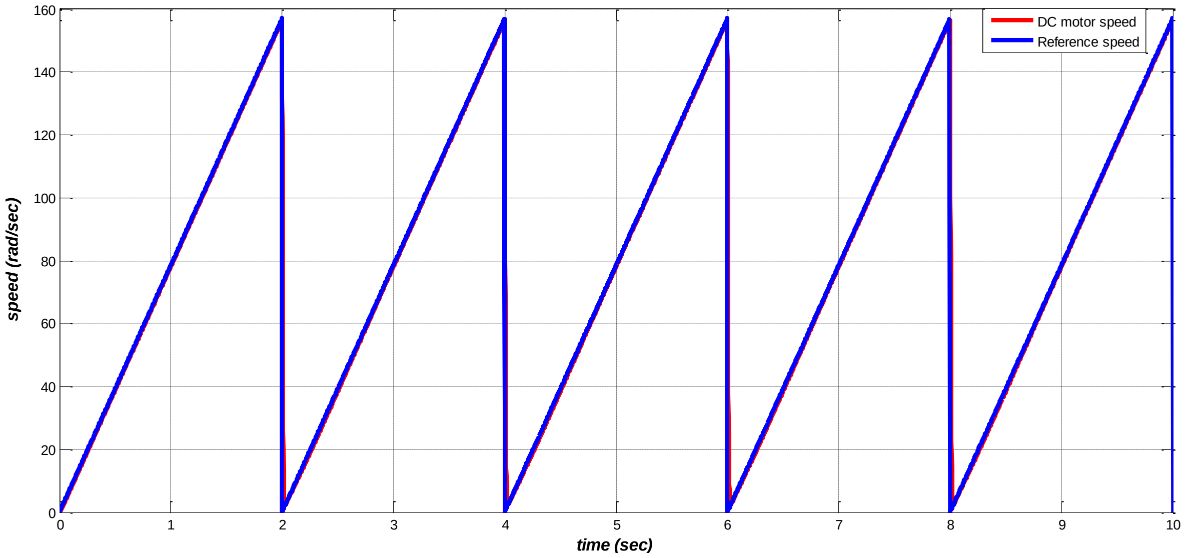


***Figure* IV.10:** *The given speed for reference speed triangular signal.*

Observation :

Despite changing the speed to several forms (rectangular signal, sinusoidal signal and triangular signal), we note that the $FOPID$ controller tracks it with high accuracy.

5.2. disturbances rejection :

During its rotation, the dc motor is exposed to many factors that affect its performance Among these factors, we find some external influences such torque load and internal influences such as the parametric variation. Taking into account the exposure of the sensor to the measurement noise.

In this part, we will subject the DC motor to internal influences (parametric variation) and external influences (torque load), exposing the sensor to measurement noise.

5.2.1. Torque load rejection :

These time responses are given in the presence of the load torque input $\mathrm{T_L} = 50\ \mathrm{N \cdot m}$, applied at the starting time $t = 3\ sec$.

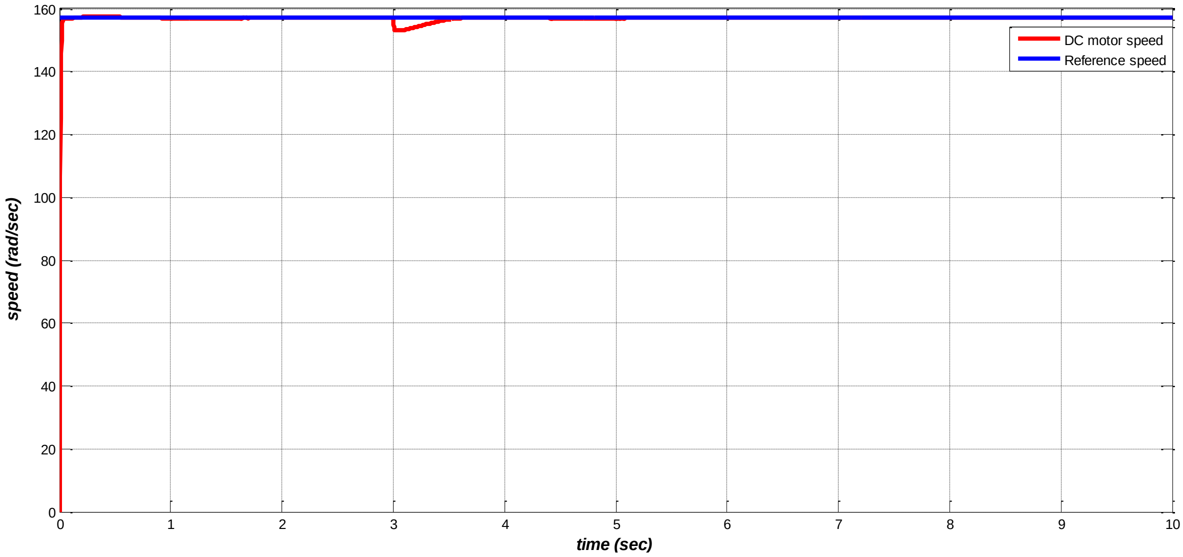


***Figure* IV.11:** *The given speed with torque load presence.*

5.2.2. Parametric variation effect rejection :

The DC motor parameters were varied at $\mathrm{t} = 4.5\ \mathrm{sec}$ where $\mathrm{R_a} = 2.25\ \Omega, \mathrm{L_a} = 0.3\ \mathrm{H}$,
$\mathrm{J} = 0.1\ \mathrm{kg.m^2/s^2}$ and $\mathrm{B} = 0.001\mathrm{N.m.s/rad}$.

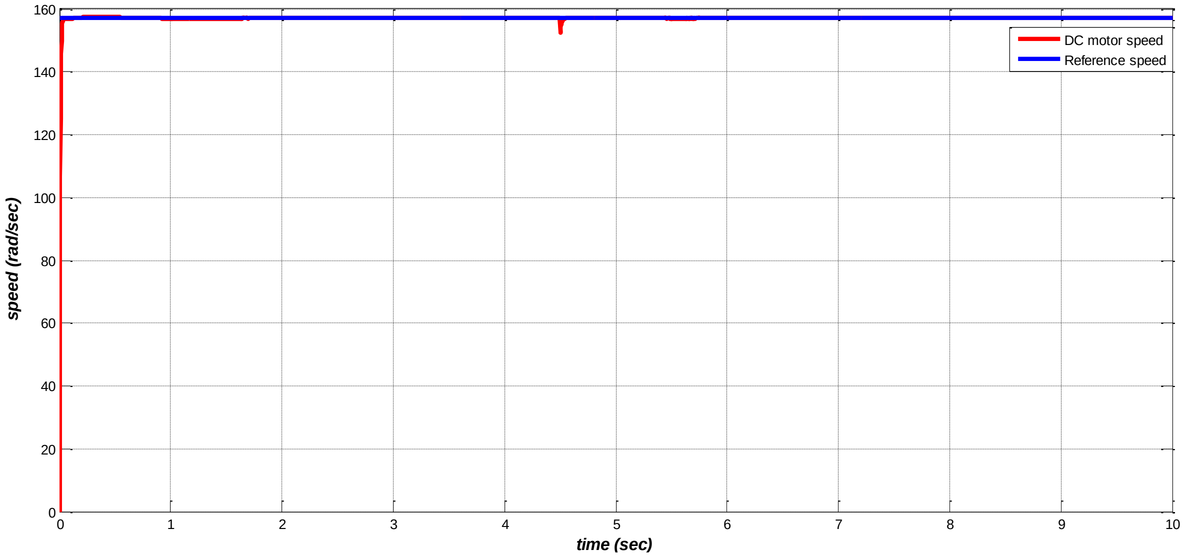


***Figure* IV.12.** *The given speed with variation parametric happening.*

5.2.3. minimisation of measurement noise :

The measurement noise ranges between $[-0.5; 0.5]$ $\mathrm{rad/sec}$ start at $\mathrm{t} =\ 7$ sec.

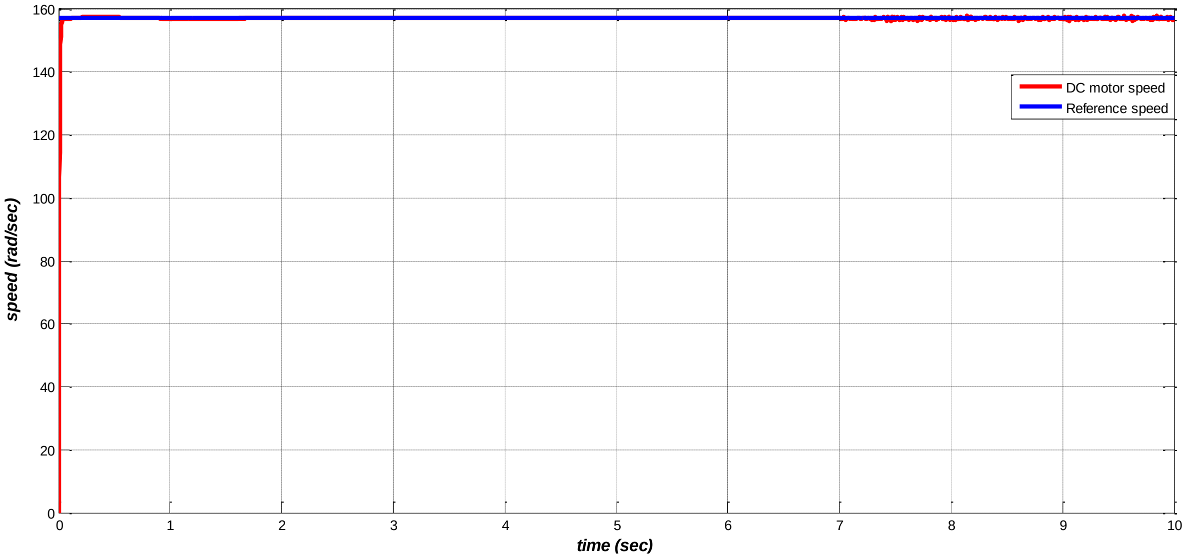


***Figure* IV.13:** *The given speed with measurement noise presence.*

Observation :

Through the (Figure IV.12), we conclude that the $FOPID$ controller rejected the external effect caused by the torque load within a short time, and the (Figure IV.13) shows that the latter also rejected the internal effect caused by parametric variation, while in (Figure IV.14) it minimized the measurement noise.

5.3. Comparison between FOPID speed control and PID speed control :

In the context of comparing the PID controller and $FOPID$ controller, we will put them in a closed loop with a DC motor with settings as shown in (Table IV.4).

Where, we will compare these two controllers in terms of accuracy of tracking speed and ability to rejection external influences.

	$\mathrm{k_p}$	K_i	$\mathrm{k_d}$	λ	μ	N
PID	9.4828	72.0302	27.8308	/	/	527
FOPID	24.7288	100	3.8088	0.99	0.99	/

Table IV.3: *Optimal parameters of PID and FOPID controller.*

5.3.1. Comparison from the side of tracking :

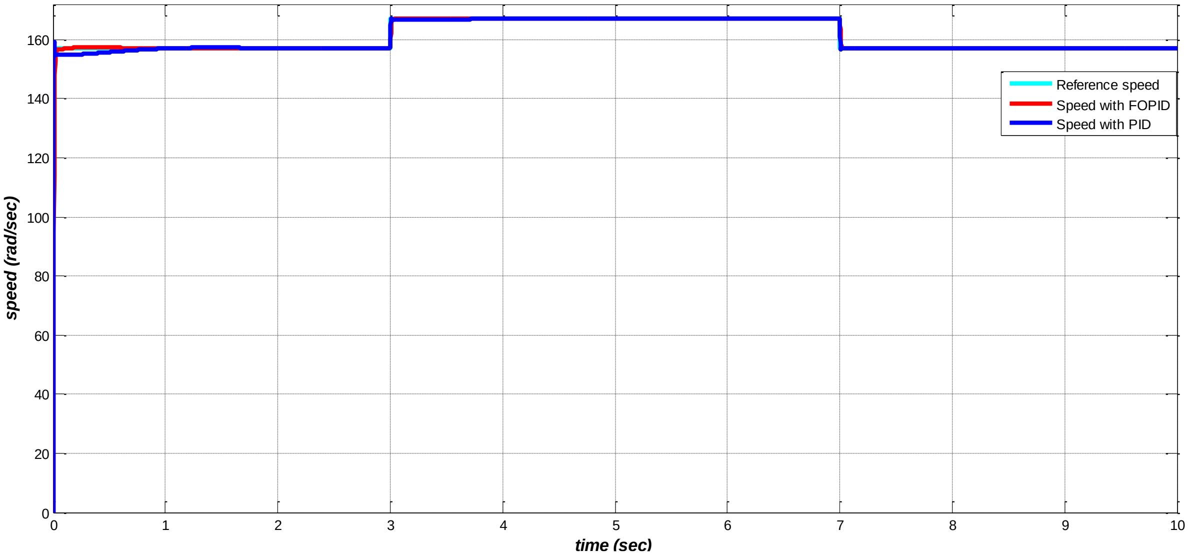
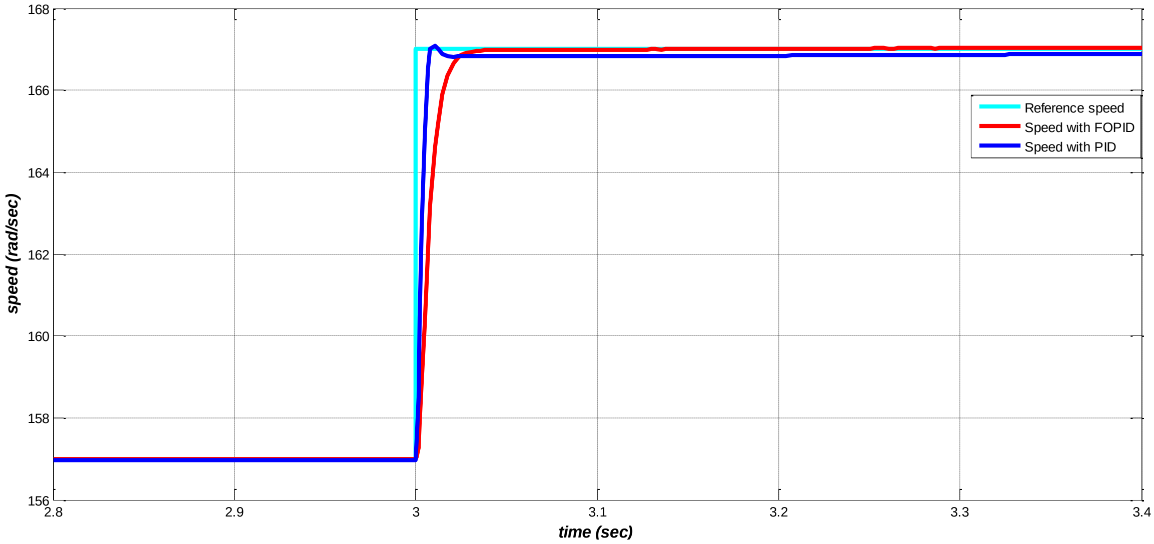
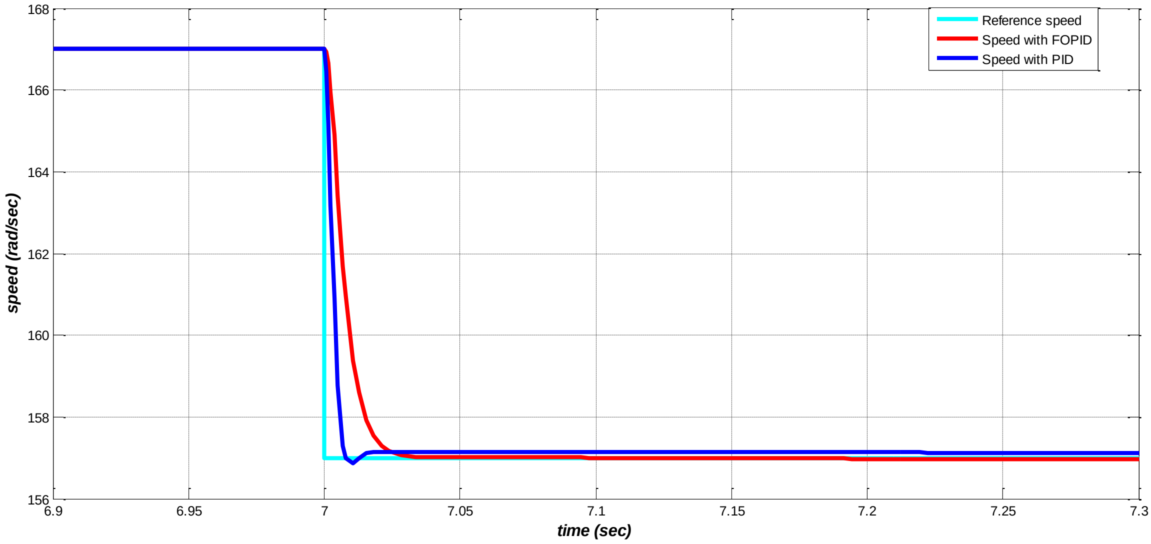


Figure IV.14: *The given speed for reference speed rectangular signal.*



(a)



(b)

Figure IV.15. (a) and (b)): *Zoom of the given speed for reference speed rectangular signal.*

Observation :

According to (Figure IV.14) and (Figure IV.15), it can be stated that the proposed $FOPID$ speed controller allows better than the proposed PID controller ensuring simultaneously, the better reference speed tracking, characterized by an almost negligible overshoot and reduced steady-state error

5.3.2. Comparison from the side of disturbance rejection and measurement noise minimisation :

The torque load input $C_t = 50\ \mathrm{N \cdot m}$, applied at the starting time $t = 2.5$ sec.

And measurement noise is ranges between [-0.5;0.5] at the starting time $t = 7.5$ sec.

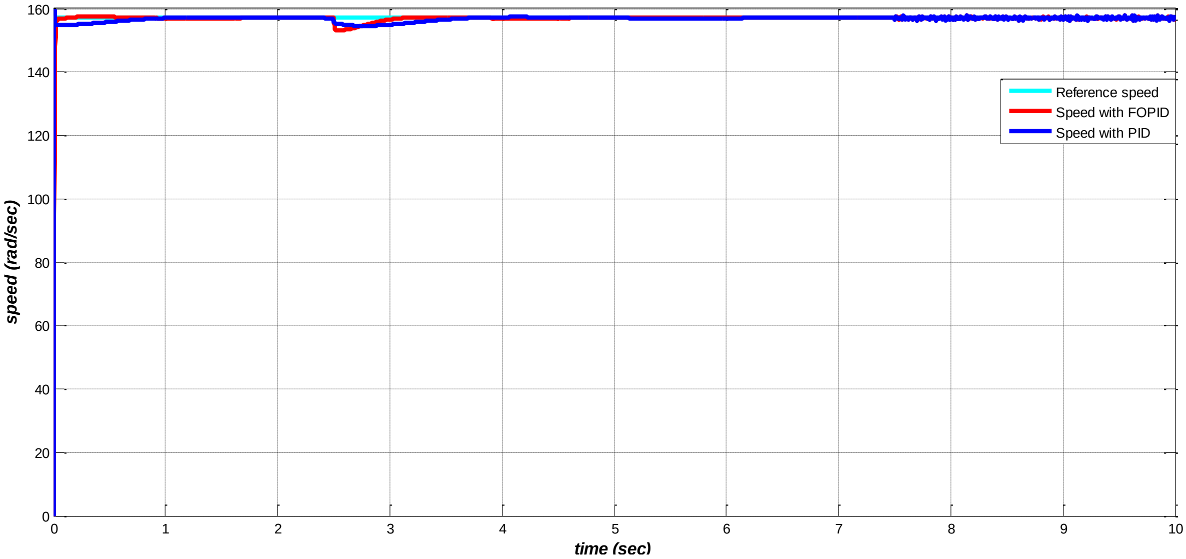
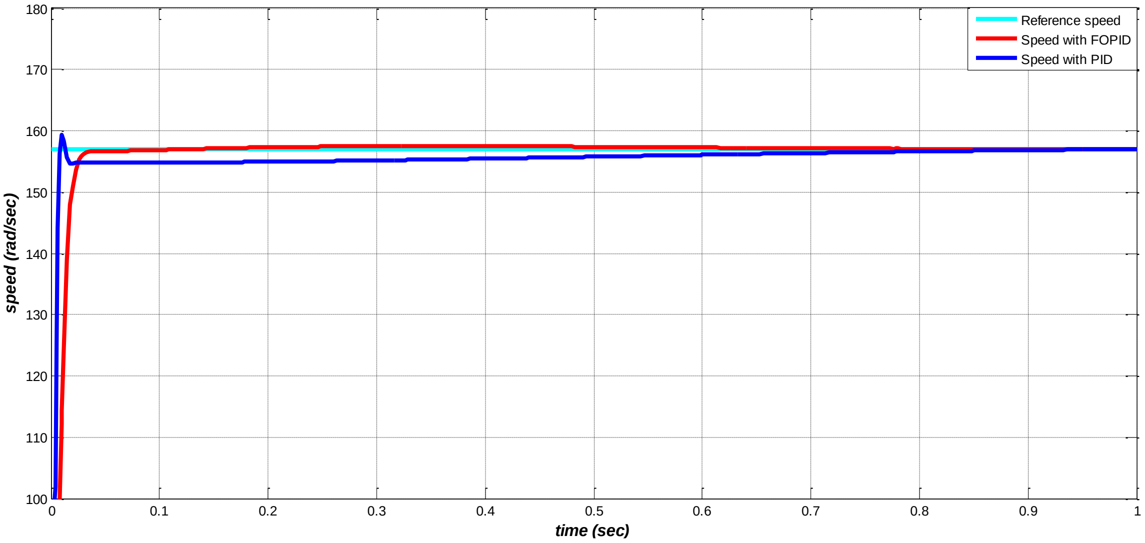
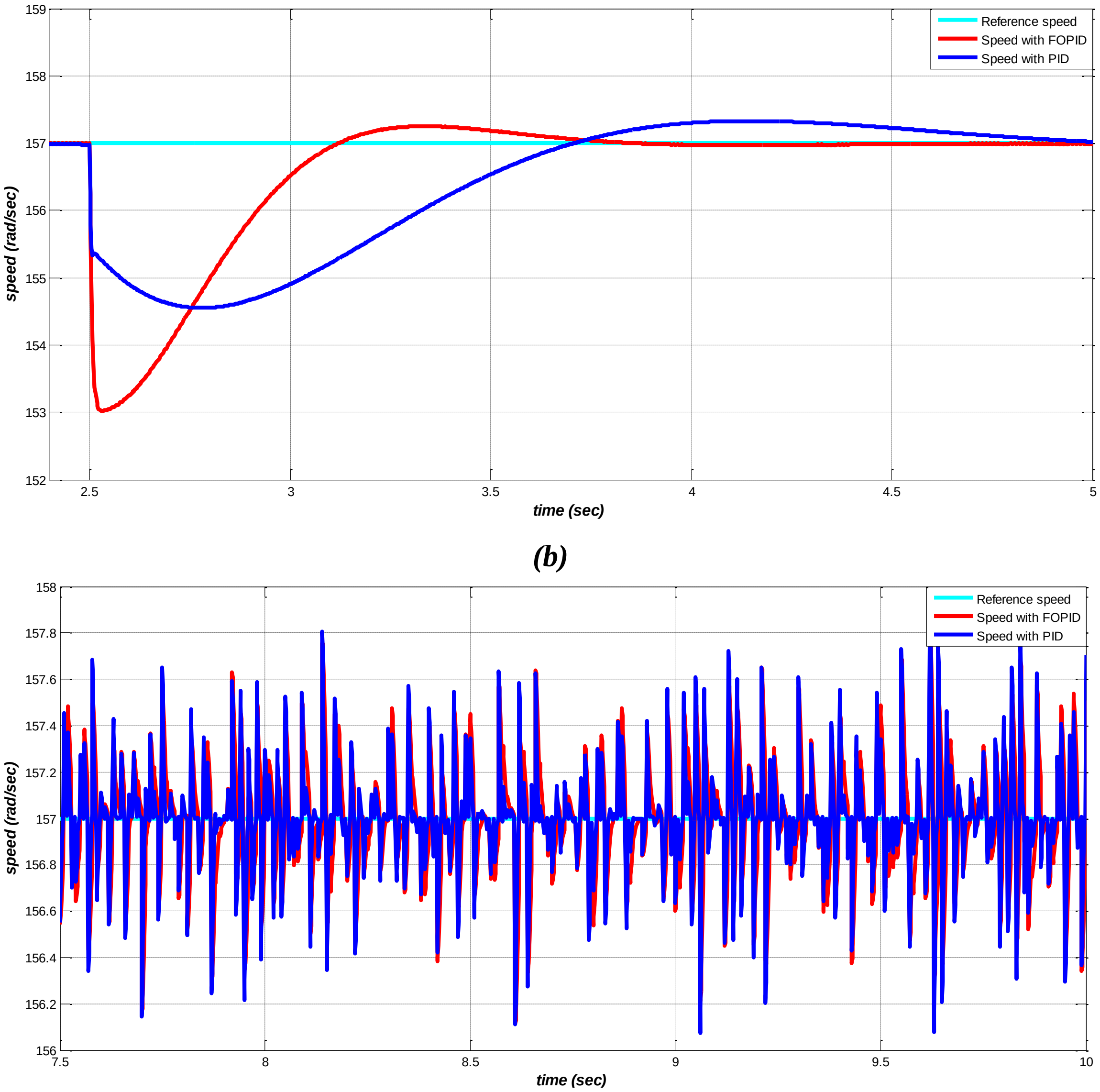


Figure IV.16: *The given speed with torque load and measurement noise presence.*



(a)



(b)

(c)

Figure IV.17. ((a) and (b) and (c)): *Zoom of the given speed with torque load and measurement noise presence.*

Observation :

According to (Figure IV.16) the time responses obtained with the $FOPID$ controller is significantly better than PID controller. To better verify of two outputs obtained, the latter is represented in a time interval $0 < t < 1$ see (Figure IV.17. (a)).

Also, $FOPID$ given good attenuation of load disturbances as shown in (Figure IV.17. (b)). finally, the minimization of measurement noise which is illustrated in (Figure IV.17. (c)). is very satisfactory.

	$Rise\ time\ (sec)$	$Overshoot\ D\ (\%)$	$Torque\ load\ rejection\ time\ (sec)$
PID	2.31	9.35	1.2082
FOPID	0.89	0.49	0.6216

Table IV.4: *DC motor performance given by the two controllers.*

Observation :

From the (Table IV.4), we conclude that the fractional order PID controller is better than the classical PID controller in improving engine performance in terms of the rise time, overshoot(D (%)) and Torque load rejection time.

6. Conclusion :

In this chapter, we presented a DC motor speed regulation with a fractional order PID controller, for the optimization of parameters this controller we used the particle swarm optimization PSO algorithm. It is a very simple and efficient algorithm which gave optimal parameters of fractional order PID controller, based on the $ITAE$ criterion.

We study the speed of a DC motor in a closed loop with $FOPID$ controller. First, we subjected the dc motor to different speeds (constant signal, rectangular signal, sinusoidal signal and triangular signal), we noticed the ability of the FOPID controller to change the dc motor speed to follow these speeds with high accuracy in a short time. Secondly, we studied the efficiency of the FOPID controller in rejection the external influences (torque load), the internal influences (parametric variation) and minimization of the measurement noise. Finally, we compared the performance of the PID and FOPID controllers to confirm the superiority and efficiency of the FOPID controller in tracking accuracy and the ability to rejection the internal and external influences.

Based on these results, it can be said that the FOPID controller is very effective and reliable in controlling the speed of the DC motor.

GENERAL CONCLUSION :

There are many types of the DC motor, including the separately existed DC motor, where it is concerned with its simplicity and ease of control by the fixing of the field tension and control the armature tension we call these armature-controlled DC motor, where the DC motor model becomes a linear model.

Most systems need a controller in order to work with a certain performance. The most well-known controller in control engineering is a PID controller, this controller is simple and easy to adjust. The PID controller is a special case of FOPID, where FOPID gives more ranges to control systems performance, but it is difficult to find the perfect parameters, for this improvement is used in order to find the optimal parameters of the FOPID controller.

The objective of this work is to modelling a DC motor and control it by a fractional order PID, in order to improve the performance of this type of motor, tracking the reference speed signal, minimisation of measurement noise and rejection of some external influences such as the torque load and internal influences such as the parametric variation.

For determine the optimal parameters of the FOPID controller we using the metaheuristic optimization algorithm PSO (Particle Swarm Optimization), the advantage of using PSO tuning FOPID is the computational efficiency, because it is very easy of the implementation and the computation processes is very fast, comparing with conventional methods.

The process of determine the optimal parameters of the FOPID controller $\{K_p, K_i, K_d, \lambda, \mu\}$ based on the minimization of objective function which is the integral of time absolute error (ITAE), Therefore, it can be inferred that the PSO algorithm is very effective in optimizing the FOPID controller which ensure excellent speed behaviour the DC motor.

From the simulation and results it can be concluded that the proposed FOPID controller improves the performance characteristics and robust in tracking speed reference signal with high accuracy, also in the external and internal influences rejection and the measurement noise minimization, as compared to the conventional PID controller applied to DC motor.

As perspectives, we suggest the following:

- Use the field-controlled DC motor for modelling the DC motor.
- Use another optimization algorithms like: Genetic Algorithm (GA)
- Use another objective function like: ISE, ITE, ITSE, ITAE + ITSE…
- Use of other types of controllers. For examples, TID, TPID, TFOPID.

Bibliography:

[1] Andres Kornai, “Mathematical linguistics (Advanced Information and Knowledge Processing)”, *Springer nature*, 22nd October 2010.

[2] I. Podlubny; “Fractional Differential Equations, Mathematics in Science and Engineering V198”, *Academic Press* 1999.

[3] Rose-Hulman, Undergrad, “Fractional Calculus and the Taylor-Riemann Series”, *J. Math*. Vol.6(1), 2005.

[4] I. Podlubny, “Fractional-order systems and $PI^{\alpha}D^{\beta}$ controllers”, *IEEE Transactions on Automatic Control*, 1999.

[5] Y. Luo, Y. Q. Chen, C. Y. Wang, and Y. G. Pi, “Tuning fractional order proportional integral controllers for fractional order systems”, *Journal of Process Control*, 2014.

[6] M. R. Bonyadi, Z. Michalewicz, “Particle swarm optimization for single objective continuous space problems: a review”, *Evolutionary Computation*, 2017.

[7] Sadik Kara, “A Roadmap of Biomedical Engineers and Milestones”, *BOD-books on demande*, 2012.

[8] A.F.M. Sajidul Qadir, “Electro-Mechanical Modelling of SEDM (Separately Excite DC Motor) & Performance Improvement Using Different Industrial Controllers”, 2013.

[9] R.K. Rajput, “Basic Electrical and Electronics Engineering”, *LAXMI publications (p) LTD,* 2007.

[10] Peter Philips, “Electrical Principles (Electrical Skills Series)”, *Cengage AU*, 2019.

[11] Mark Brown, Jawahar Rawtani, Dinesh Patil, “Practical Troubleshooting of Electrical Equipment and Control Circuits”, *Elsevier,* 2004.

[12] U.A. Bakshi, Dr. M.V. Bakshi, “Electric Motors”, *Technical Publications,* 2020.

[13] D. Ganesh Rao, K. Chennavenktesh, “Control Engineering”, dorling Kindersley (India) *Pvt, Ltd, licencees of Pearson Education in South Asia,* February 2017.

[14] Farzin Asadi, “State-Space Control Systems, THE MATLAB/SIMULINK (Synthesis Lectures on Control and Mechatronics)”, *Morgan & Claypool PUBLISHERS,* 2020.

[15] Ramin S. Esfandiari Bei Lu, CRC Taylor & Francis Group, “Modelling and Analysis of dynamic systems”, *Boca Rotan, CRC Press,* 19 May 2014.

[16] Rory A Cooper, “Rehabilitation Engineering Applied to Mobility and Manipulation, (medical science series)”, *CRC Press*, 1995.

[17] J1 Keziz, B., Djouambi, A. & Ladaci, S, “A new fractional order controller tuning method based on Bode’s ideal transfer function”, *Int. J. Dynam. Control* ,2020.

[18] J. A. Tenreiro Machado, Virginia Kiryakova, Francesco Mainardi, “Fractional calculus and applied Analysis”, *Institute of Mathematics and Innformations*, 2011.

[19] A. Oustaloup et al. “Frequency-band complex noninteger differentiator: characterization and synthesis”. *In: IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2000.

[20] Kimeu, Joseph M., "Fractional Calculus: Definitions and Applications". Masters Theses & Specialist Projects, http://digitalcommons.wku.edu/theses/115, 2009.

[21] K.S. Miller and B. Ross. “An Introduction to the Fractional Calculus and Fractional Differential Equations, *Willey, New York. In: NY. Zbl0789 26002*, 1993.

[22] R.L. Magin, “Fractional Calculus in Bioengineering”. *Begell House Redding*, 2006.

[23] S.G. Samko, A.A. Kilbas, and O.I. Marichev, “Fractional integrals and derivatives”. *Gordon and Breach Science Publishers, Yverdon-les-Bains, Switzerland*, 1993.

[24] I. Podlubny, “Analogue realizations of fractional-order controllers In: *Nonlinear dynamics”*, 2002.

[25] K.B. Oldham and J. Spanier. “The Fractional Calculus of Mathematics in science and engineering”. Academic Press, New York, London, 1974.

[26] C.A. Monje, “Fractional-Order Systems and Controls”, *Fundamentals and Applications Springer Science & Business Media*, 2010.

[27] B.M. Vinagre, “Some Approximations of Fractional Order Operators Used in Control Theory and Applications”. *In: Fractional calculus and applied analysis*, 2000.

[28] S.E. Hamamci, “An Algorithm for Stabilization of Fractional-Order Time Delay Systems Using Fractional-Order PID Controllers”, *In: IEEE Transactions on Automatic Control*, 2007.

[29] H. Malek, Y. Luo, and Y.Q. Chen, “Tuning Fractional Order Proportional Integral Controllers for Time Delayed Systems with a Fractional Pole”, *In: ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers Digital Collection*, 2011.

[30] F. Padula, R. Vilanova, and A. Visioli, “H1 Model Matching PID Design for Fractional FOPDT Systems”, *In: American Control Conference (ACC). IEEE*, 2012.

[31] D. Valério, “Fractional robust system control”, *In: Universidade Técnica de Lisboa*, 2005.

[32] Tepljakov, A. Alagoz, B. B. Yeroglu, C. Gonzalez, E. Hossein Nia, S. H. and Petlenkov, “FOPID controllers and their industrial applications: a survey of recent results”, *IFAC-Papers Online*, 2018.

[33] Y.Q. Chen, I Petráš and D. Xue, “Fractional order control-a tutorial”, *American control conference. IEEE*. 2009.

[34] Denis Matignon, “Generalized fractional differential and difference equations: stability properties and modelling issues”. *In: Mathematical Theory of Networks and Systems symposium*,1998.

[35] A. Charef, H. H. Sun, Y. Y. T Sao and B. Onaral. “Fractal system as represented by singularity function”. *IEEE Transactions on Automatic Control*, 1992.

[36] A. Oustaloup. “La commande CRONE”. *Editions Hermes, Paris*, 1991.

[37] Araki M, “control systems, robotics, and automation”, *EOLSS Publications*, 2009.

[38] Sigurd Skogestad, “multivariable feedback control”, Analysis and design, *Norwegian University of Science and Technology, Ian Postlethwaite, Wiley*, 2005.

[39] M. Zamani, M. Karimi-Ghartemani and N. Sadati, ‘’FOPID controller design for robust performance using particle swarm optimization. *Fractional calculus and applied analysis’’*, 2007.

[40] A. Ali, A. Mohamedy, A. Salimz, E. El-Aminx and O. Ahmed, “Design and Implementation of Two-Wheeled Self-Balancing Robot Using PID Controller”, *International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), University of Leicester*, 2020.

[41] Zhang, J., & Guo, L, “Theory and design of PID controller for nonlinear Uncertain system”. *IEEE control system letters* http://doi.org/10.1109/lcsys.2019.2915306, 2019.

[42] Tepljakov, A. Petlenkov, E. Belikov and J. Finajev,). “Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB “, In IEEE *conference on computer aided control system design (CACSD)*, 2013.

[43] Bertram Ross, “Historixa Mathematica (the development of Fractional Calculus 1695 – 1900)”, *University of New Haven,* CONN, G6516, 1977.

[44] C. Zhihuan, Y. Xiaohui, J. Bin, W. Pengtao and T. Hao, “Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II”, *Energy Conversion and Management*, 2014.

[45] K. Zhou, J.C. Doyle, “Essential of Robust Control, New Jersey”, *Prentice-Hall*, 1998.

[46] Zwe-Lee Gaing, “A particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR (Automatic Voltage Regulator) System”, *IEEE Transactions on Energy Conversion*, 2004.

[47] Djari Abdelhamid, Bouden Toufik and Blas M. Vinagre, Optimal, “Fractional-order Sliding Mode Controller Design for a class of Fractional-order Nonlinear Systems using particle swarm optimization Algorithm”, *Printed in Romania, CEAI,* 2016.

[48] Z. Bingül and K. Oguzhan, “A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control”, *Expert Systems with Applications*, 2011.

[49] H. Gozed, T. Cengiz, I. Kocaarslan and Ertugrul, “PSO based load frequency control in a single area power system”, University of Psitesti, *Scientific Bulletin,* 2008.

[50] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer”, *Proc. of IEEE Congress on Evolutionary Computation*,1998.

[51] Djari Abdelhamid, Bouden Toufik and Abdesselem Boulkroune, “Design of Fractional-order Sliding Mode Controller (FSMC) for a class of Fractional-order Non-linear Commensurate Systems using a Particle Swarm Optimization (PSO) Algorithm”, *Control Engineering and Applied Informatics,* 2014.

[52] E. K. Boukas, “Systèmes asservis”, Presses inter Polytechnique, 1995.

[53] F. Aashoor, “Maximum power point tracking techniques for photovoltaic water pumping system”, 4 Jan 2016.

[54] W. C. Schultz and V. C. Rideout, “Control system performance measures: Past, present and future”, *IRE Trans. on Automatic Control*, 1961.

[55] I.J. Nagrath and M. Gopal, “Control Systems Engineering, Fifth Edition”, ISBN *New Age International Publishers*, 2007.