الجمهورية الجزائرية الديموقراطية الشعبية

**République Algérienne Démocratique et Populaire**

وزارة التعليم العالي والبحث العلمي

**Ministère de l'enseignement supérieur et de la recherche scientifique**

**Université Mohamed khider – Biskra**
**Faculté des Sciences et de la Technologie**
**Département de Génie électrique**
**Anne : 2024/2025**

جامعة محمد خيضر-بسكـــرة
كلية العلـــوم و التكنـــولوجيـــا
قسم الهندسة الكهربائية
سنة:2025/2024

# Thèse de doctorat

## Spécialité : Electromécanique.

## Option : Electromécanique.

# Thème :

# Développement des techniques De Commande Performantes Pour Un Système De Conversion Basé Sur La Machine Synchrone à Aimant Permanent

## Présentée par :

✓ **Nouaoui Tahar**

**Soutenue publiquement le 11 / 05 /2024**

**Devant le jury composé de :**

| | | |
|---|---|---|
| Mr. Magherbi Ahmed Chaouki | Professeur. Université de biskra | Président |
| Mr. Abdelhakim Dendouga | Professeur. Université de biskra | Encadreur |
| Mr. Abdelmalik Bendaikha | MCA. Université de batna | Co-encadreur |
| Mr. Betka Achour | Professeur. Université de biskra | Examinateur |
| Mr. Sebti Belkassem | Professeur. Université de batna | Examinateur |

# Keywords

Interval type-2 Fuzzy logic controller (**FLC2**); Permanent Magnet Synchronous Motor (PMSM); Processor-in-the loop (PIL) Co-simulation; Adaptive robust control**;** metaheuristic optimization algorithm.

# Abstract

This thesis's main goal is to employ modern intelligent methodologies such as fuzzy logic controllers (FLC) and metaheuristic algorithms (MHA) to improve the performance of field-oriented-control (FOC) of the permanent magnet synchronous motor (PMSM). First, PMSM mathematical model was described in details along with the foundational parts of FOC such as feed forward compensation (FFC) decoupling technique. Then, a set of proportional-integral (PI) controllers optimized using classical tuning techniques (CT) were presented and compared to a set of PI controllers tuned by a new hybrid metaheuristic algorithm known as symbiotic-particle-sewing (SPS). In this comparison, both sets of controllers were used for FOC of PMSM, and upon analysis of this comparison results, SPS based PI (SPS-PI) proved to be far better than CT based PI (CT-PI). However, SPS-PI performance was still considered unsatisfactory due to its unprecise transient response, and high steady state error (SSE) when a load charge is applied. Further, to address the downsides of SPS-PI, an interval Type-1 FLC with tuned membership functions (MFS) denoted by (F1MF) was designed and optimized using SPS algorithm for speed control of PMSM with the currents remaining controlled by SPS-PI, and then compared to CT-PI and SPS based conventional FLC with unoptimized MFS denoted by (FC1). Furthermore, F1MF enhanced tolerance to load charges and unmatched transient response proved its superiority over FC1 and CT-PI controllers. However, although F1MF's settling time (ST) and overshoot (OS) rejection were very acceptable, its overall SSE minimization performance fell short when high load charges were applied. Therefore, an interval Type-2 based FLC (FLC2) with tuned MFS denoted by (F2MF), was presented and tuned using SPS algorithm. F2MF outmatched F1MF SSE minimization in the presence of high load charges but at the cost of a significant downgrade in its transient response ST and OS rejection compared to F1MF. Additionally, in order to address F2MF transient response problems while maintaining its high disturbances tolerance, it was paired with a fractional derivative (FD) and an adaptive low pass filter (ALF) combination denoted by (ALD). Finally, through MATLAB/SIMULINK and processor in the loop (PIL) co-simulation. The new controller constructed of F2MF and ALD (F2MF-ALD) was proven to be functional and far superior to its simpler variants such as F1MF, and F2MF when used for speed control of PMSM.

# ملخص

الهدف الرئيسي لهذه الأطروحة هو استخدام منهجيات ذكية حديثة مثل وحدات التحكم المنطقية الضبابية (FLC) وخوارزميات الميتا هيوريست (MHA) لتحسين أداء التحكم الموجه بالمجال (FOC) للمحرك المتزامن المغناطيسي الدائم (PMSM). أولاً، تم وصف النموذج الرياضي PMSM بالتفصيل جنبًا إلى جنب مع الأجزاء الأساسية لـ FOC مثل تقنية فصل التعويض الأمامي للتغذية (FFC). بعد ذلك، تم تقديم مجموعة من وحدات التحكم التكاملية النسبية (PI) المُحسّنة باستخدام تقنيات الضبط الكلاسيكية (CT) ومقارنتها بمجموعة من وحدات التحكم PI المُضبوطة بواسطة خوارزمية ميتا هيوريستية هجينة جديدة تُعرف باسم الخياطة التكافلية للجسيمات (SPS). في هذه المقارنة، تم استخدام كلتا مجموعتي وحدات التحكم للتحكم الموجه بالمجال في PMSM، وعند تحليل نتائج هذه المقارنة، ثبت أن PI القائم على SPS (SPS-PI) أفضل بكثير من PI القائم على CT (CT-PI). ومع ذلك، لا يزال أداء SPS-PI يعتبر غير مرضٍ بسبب استجابته غير الدقيقة للزمن الانتقالي، وخطأ الحالة المستقرة المرتفع (SSE) عند تطبيق حمل مقاوم. علاوة على ذلك، لمعالجة الجوانب السلبية لـ SPS-PI، تم تصميم FLC من النوع 1 (FLC1) مع وظائف العضوية المضبوطة (MFS) المشار إليها بواسطة (F1MF) وتم تحسينها باستخدام خوارزمية SPS للتحكم في سرعة PMSM مع بقاء التيارات خاضعة لسيطرة SPS-PI، ثم تمت مقارنته بـ CT-PI وFLC التقليدي القائم على SPS مع MFS غير المحسن المشار إليه بواسطة (FC1). علاوة على ذلك، عززت استجابة F1MF السريعة ة قدرته على التعامل مع الحمل المقاوم تفوقه على وحدات التحكم FC1 وCT-PI. ومع ذلك، على الرغم من أن وقت الاستقرار (ST) و تقليل تجاوز الحد الأقصى (OS) لـ F1MF كان مقبولاً للغاية، إلا أن أداء تقليل SSE الإجمالي كان أقل من المطلوب عند تطبيق حمل مقاوم ذو قيمة عالية. لذلك، تم تقديم FLC من النوع 2 (FLC2) القائم على الفاصل الزمني مع MFS المضبوط الذي يشار إليه بـ (F2MF)، وتم ضبطه باستخدام خوارزمية SPS. تفوق F2MF على تقليل F1MF SSE في وجود حمل مقاوم عالي ولكن على حساب خفض كبير في استجابته العابرة ST و قدرته على تقليل OS مقارنةً بـ F1MF. بالإضافة إلى ذلك، من أجل معالجة مشاكل الاستجابة العابرة لـ F2MF مع الحفاظ على تحمله العالي للاضطرابات، تم إقرانه بمجموعة مشتق كسري (FD) ومرشح تمرير منخفض متكيف (ALF) يشار إليه بـ (ALD). أخيرًا، من خلال محاكاة MATLAB/SIMULINK والمعالج في الحلقة (PIL)، ثبت أن وحدة التحكم الجديدة المصنوعة من F2MF وALD (F2MF-ALD) وظيفية ومتفوقة بكثير على المتغيرات الأبسط مثل F1MF وF2MF عند استخدامها للتحكم في سرعة PMSM.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| MHA | Metaheuristic algorithms |
| FOC | Field-oriented-control |
| PMSM | Permanent magnet synchronous motor |
| SVPWM | Space vector pulse width modulation |
| FFC | Feed forward compensation |
| PI | Proportional-integral |
| CT | Classical tuning techniques |
| SPS | Symbiotic-Particle-Sewing |
| SSE | Steady state error |
| ST | Settling time |
| MFS | Membership functions |
| OS | Overshoot |
| FLC | Fuzzy logic controller |
| FLC1 | Interval Type-1 FLC |
| FC1 | FLC1 with unoptimized (normalized) MFS |
| F1MF | Interval Type-1 FLC with optimized MFS |
| FLC2 | Interval Type-2 based FLC |
| F2MF | Interval Type-2 based FLC with tuned MFS |
| UMF | Upper bound membership function |
| LMF | Lower bound membership function |
| FD | fractional derivative |
| LD | low pass filter |
| ALD | adaptive low pass filter |
| OBJ | objective function |
| TRA | Type-reduction algorithm |
| THD | Total harmonics distortion |

# List of Notations

| | |
|---|---|
| $abc$ | Three phase coordinates axis |
| $V_{abc}$ | Three phase voltages |
| $i_{abc}$ | Three phase currents |
| $\varphi_{abc}$ | $abc$ axis stator fluxes |
| $\varphi_f$ | Permanent magnet flux linkage |
| $R_s$ | Stator phase resistance |
| $L_{s0}$ | Phase constant inductance |
| $L_{s1}$ | Phase variable inductance |
| $M_{s0}$ | Mutual inductance |
| $\theta$ | Rotor electrical angular position |
| $\omega_r$ | Motor mechanical speed |
| $\omega$ | Motor electrical speed |
| $\omega_{r(ref)}$ | Speed reference |
| $T_e$ | Motor electromagnetic torque |
| $T_r$ | Load charge torque |
| $J$ | Inertia |
| $B$ | Vicious friction |
| $dq$ | Rotating reference axis |
| $V_d, V_q$ | $dq$ axis voltages |
| $i_d, i_q$ | $dq$ axis currents |
| $L_d, L_q$ | $dq$ axis inductances |
| $P$ | Number of pair of poles |
| $V_r$ | Voltage reference vector |
| $\alpha\beta$ | Fixed reference axis |
| $V_\alpha, V_\beta$ | $\alpha\beta$ axis voltages |
| $k_p, k_i$ | PI parameters |

# Dedication

In this dedication, I wish to express my sincerest gratitude to supervisors for their invaluable guidance, and finally, love and gratitude to my family for their unwavering support throughout my entire life.

# General introduction

## 1.1 OVERVIEW

Electrical motors are a type of machines that works on transforming electrical energy into mechanical energy [1]. Since the fabrication of one of the first and most powerful electrical motors at the time by Moritz von Jacobi back in 1834, interest in electrical rotating machinery have grown significantly which played a major role in shaping the future of science and technology. There are two main types of electrical motors [2], which can also branch into even more types but with less differences. The first main type is the direct current (DC) motor [3], which obviously works using DC and can provide a decent torque/size ratio [4], but it requires a frequent maintenance by replacing its brushes and commutators which can be costly and time consuming, thus making alternating current (AC) motors more preferable due to their lack of need for commutators to operate [5]. AC motors use alternative current to work which is a major advantage due to the ease of generating and transporting AC compared to DC [6], and they can be classified into two types of motors: induction motors (IM) [7], which has a short-circuited copper windings in the rotor, and synchronous motors (SM) [8], which usually has a pair or more of permanent magnets in the rotor depending on its targeted nominal speed.

Unlike the DC motors, AC motors input source voltages are supplied to the stator rather than the rotor. In both the IM and the SM, the alternative voltages supplied to the stator produces a rotating magnetic field with a rotational speed called synchronous speed [9]. However, in the case of IM, the stator rotating magnetic field interacts with the rotor windings which based on the principle of electromagnetic induction creates a current and an electrical magnetic force (EMF) in the rotor [10]. Then, the stators rotating magnetic field pushes the EMF of the rotor causing it to rotate with a speed slightly slower than the synchronous speed. In the case of permanent magnet synchronous motor (PMSM), the rotating magnetic field of the stator is synchronized with the magnetic field of the rotor magnets, causing the rotor to rotate at a speed that exactly matches the synchronous speed [11]. The employment of PMSM in electrical applications have known a rapid growth recently compared to DC motor and IM [12], which is due to PMSM simple model compared to IM, and its superior viability and low cost maintenance compared to DC motor [13]. Moreover, unlike IM, vector control techniques are made easier when implemented for the control of PMSM [13]. Among these control techniques is direct torque control (DTC), which is a technique that aims at maintaining the torque and flux of the motor at desired values by selecting the appropriate voltage vectors [14]. The main advantages of DTC are the unnecessity of using coordinate transformations or pulse-width modulation (PWM) algorithms, and the simplicity of the control part of the system since DTC only needs one linear controller to be able operate [15]. However, despite DTC remarkable feats, sometimes it falls short when the desired speed is small, and it also experiences a relatively high torque ripples compared to field-oriented-control (FOC) technique.

Overall, FOC is the most complete control technique of PMSM due to its high compatibility with most real-life electrical motor's applications such as speed and position tracking, or traction and transportation especially when modified to work with a non-classical controller [16]. The most utilized non-classical controllers are sliding mode controller (SMC) [17], fractional order proportional-integral-derivative (FPID) controller [18], and artificial intelligence (AI) based controllers such as fuzzy logic controller (FLC) [19].

## 1.2 LITRETURE REVIEW AND PROBLEM STATEMENT

In the literature of FOC of PMSM, classical controller such us proportional-integral (PI) are still the most common due to their linear function, simplicity, and ease of usage. To tune and implement PI controllers for FOC of PMSM, usually, the speed PI controller is tuned using poles imposition technique [20], whereas direct and quadratic axis currents PI controller are tuned via poles compensation method [21]. Additionally, these classical techniques are capable of providing a decent tuning results. However, they often fall short and can't match the capabilities of modern tuning methods such metaheuristic algorithms (MHA) based tuning of PI controllers [22, 23]. Nevertheless, despite the lift in PI controller performance when tuned using MHA, it still lacks the additional pivotal features such as fast and precise response that non-classical controllers can provide. In the literature, numerous non-classical controllers have already been employed to control both the speed and currents of PMSM. However, each one of them comes with its own pros and cons. For example, SMC can handle any type of disturbances and still maintain a satisfying speed tracking. However, its switching on and off properties can produce high ripples in the actual values of torque and currents [24]. FPID is a flexible controller, but it lacks the ability to adapt to variations in system parameters. FLC, on the other hand, can tolerate internal disturbances such as motor parametric variations, but since it lacks an integration operator, external disturbance can cause a significant steady-state-error (SSE) in its produced mechanical speed [19]. This SSE can be reduced by using a hybrid controller combined of a PI and a FLC parts (FLC-PI) [25]. However, in this case, FLC produces two outputs which complicates its optimization and design due to the intense amount of inference rules required for the controller to be able to function properly. Additionally, for a less complicated approach, a single output FLC with MHA based membership functions (MFS) parameters can significantly enhance the transient response and minimization of SSE capability of FLC, as proposed in [26]. However, that was still not enough mitigate the effect of high external disturbances.

Ideally, the best FLC for speed control of PMSM is the one that can deliver a fast transient response, and an almost non-existent overshoot (OS) and SSE in the actual value of the mechanical speed without normalizing or limiting the output of the controller using techniques such as anti-wind up [27], or MATLAB/SIMULINK saturation tool which limits the controller flexibility and doesn't show its real performance [28]. Unfortunately, such ideal fuzzy logic controller is very rare in this area of research. In [29], authors proposed a fractional order backstepping based FLC with seemingly ideal performance. However, in addition to not presenting a reliable tuning method, authors didn't investigate this controller tolerance against inductances parametric variations which is the most dangerous type of electrical motor parameters deviations [30]. Further, in [31], an adaptive type-fuzzy logic controller was proposed with the ability to perform a high tolerance speed tracking. However, its transient response was relatively slow compared to similar research. Furthermore, in [25], a MHA based hybrid fuzzy-PI controller was proposed with the ability to handle any type of external disturbances, but it failed to fully suppress speed tracking OS despite only applying small changes in the desired speed reference. Interval-type 2 FLC (FLC2) and a fractional order proportional-derivative (PD) with a feedforward fractional order PI was introduced in [32]. The controller was capable of responding fast with an acceptable precision. However, the presented test didn't include enough diversity and changes in the performed tests to provide a thorough analysis of this controller. In [33], A hybrid controller combining an FLC2 and an SMC was proposed, and an overall inspection of this controller performance shows a promising results. However, its design and tuning approach was very vague, and its tolerance to parametric changes was not investigated. Finally, in [34], a fuzzy sliding mode controller was utilized for PMSM direct drive application. This controller was able to deliver a fast transient response, but at the cost of a high OS.

Thus, this thesis aims at developing an intelligent fuzzy controller capable of providing the best possible overall performance. The desired controller can only be considered satisfactory if it proficiently mitigates the negative repercussions caused by high internal and external disturbances, while carrying out an optimal speed control with a sufficient transient response, fast settling time (ST), and negligible SSE and OS at any given speed set-point reference within the nominal velocity of the motor. The rest of the thesis goes as follows:

- **Chapter 1:** This chapter delves into the details of PMSM mechanical structure and mathematical model by presenting the most famous types of PMSM and their properties. It then introduces PMSM three-phase mathematical equations and explains how to use Park transformation and its variants to reduce the nonlinearity of these equations. Moreover, to enable vector control of PMSM, the main concepts of space vector pulse width modulation (SVPWM) are explained in detail, so it can be utilized in Chapter 2.

- **Chapter 2:** This chapter serves as an introduction to FOC of PMSM, and how to perform it using classical PI controllers with decoupling techniques such as feed forward compensation (FFC). Further, FOC reference voltages are generated using the combination of SVPWM algorithm and voltage source inverter (VSI). Moreover, classical PI controller tuning techniques are showcased and compared to modern MHA-based tuning technique using a newly proposed hybrid algorithm known as Symbiotic-Particle-Sewing (SPS). Simulation results demonstrate the superiority of the proposed SPS based PI (SPS-PI), proving it to be better than classical tuning-based PI (CT-PI).

- **Chapter 3:** This chapter aims to make use of AI based controller interval Type-1 FLC (FLC1) by first explaining the most important details of fuzzy logic control and MAMDANI based inference engine. And then implements SPS to extract the best performance of FLC1 by tuning its MFS and scaling factors. Further, a thorough simulated comparison was carried out to prove the advantages of the proposed FLC1 with SPS based MFS denoted by F1MF compared to SPS-PI and SPS based FLC1 with unoptimized MFS (FC1).

- **Chapter 4:** This chapter presents the most important contributions of this thesis by far, which is an interval Type-2 FLC with SPS based MFS denoted by F2MF combined with a newly proposed versatile fractional integration-based combination of a fractional derivative and a flexible low pass filter denoted by (ALD). The resulted proposed controller is denoted by F2MF-ALD. Additionally, the chapter start by explaining the main differences between FLC1 and FLC2, and what makes the latter better. Then, fractional integration is defined along with how to integrate it into the combination of ALD. Further, the design and tuning of F2MF-ALD is explained and demonstrated. Furthermore, an all-encompassing and in-depth comparative simulation is showcased to prove the advantages of F2MF-ALD compared to its variants and state of the art speed control of PMSM fuzzy controllers. Finally, this chapter ends by performing a successful processor-in-the loop (PIL) co-simulation of speed control of PMSM using F2MF-ALD.

# Chapter 1: Modeling of PMSM and SVPWM

In this chapter, permanent magnet synchronous motor (PMSM) dynamic model in the form of three phase mathematical equations is presented. Then, Park and inverse Park transformations are addressed. Park transformation can be used to simplify the balanced three phase model of PMSM into an equivalent two orthogonal axis system, while inverse Park transformation can be utilized to execute space vector pulse width modulation (SVPWM) algorithm and generate pulses for the voltage source inverter (VSI) that produces voltages for PMSM. Furthermore, SVPWM is explained and exploited for the control of PMSM using a three phases VSI.

## 1.1 PERMANENT MAGNET SYNCHRONOUS MOTOR

Due to the rapid growth in the development of microprocessors and advanced electronic devices such as sensors and semiconductors, interest in the research of more sophisticated and enhanced PMSM control techniques has also grown significantly. Advanced PMSM control mainly consist of a variety of controllers and power electronics-based converters. Previously, the access to PMSM was very limited due to the high prices of permanent magnets materials. However, recently, with the extended usage and enhancement of new materials such as NdFeB, their prices have known a significant decrease, hence facilitating the access to permanent magnet-based machines such as PMSM. NdFeB is not cheap. However, it dominates the market of permanent magnets due to its remarkable characteristics such linear demagnetization with very high coercivity and remanence granting its corresponding PMSM a very high torque density.

PMSM is known for its high torque/size ratio, simple dynamics compared to induction motor, low noise emission and rotor inertia, and most importantly, it doesn’t require copper windings in the rotor which significantly reduces its heat emissions and power losses. PMSM have been adopted in a variety of fields such as robotics, transportation, and aviation [35-39]. Furthermore, in terms of volume and weight, PMSM offers up to 70% less volume and 60% less weight for a 5.0 HP PMSM compared to its 5.0 HP IM counterpart as can be seen in Figure 1.1. These significant differences in physical characteristics makes the PMSM the optimal choice for high-efficiency drive applications that require reliable and excellent performance under dynamic loads with minimal energy losses. Moreover, the robust construction of PMSM ensures extended durability and lower maintenance needs, further reinforcing its suitability for demanding conditions. Collectively, these advantages position PMSM as the leading option in applications that prioritize efficiency, reliability, and long-term performance.

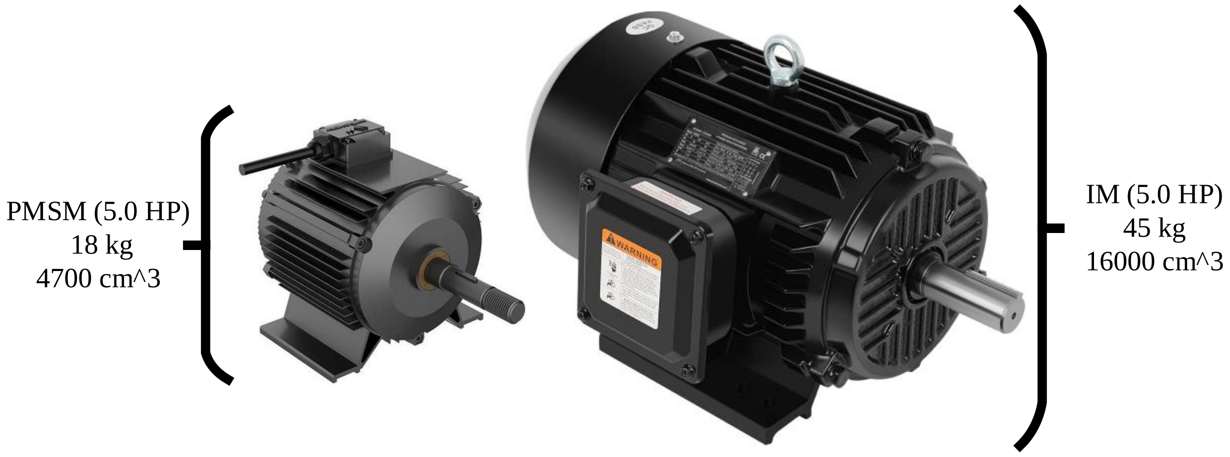


**Figure 1.1 weight and volume comparison between 5.0 HP PMSM and 5.0 HP IM.**

Advanced PMSM control techniques require a precisely constructed system that consists of three highly indispensable parts which are the power source, control system which usually an algorithm/circuit, and the motor itself. All of these parts are interconnected as illustrated in Figure 1.2.

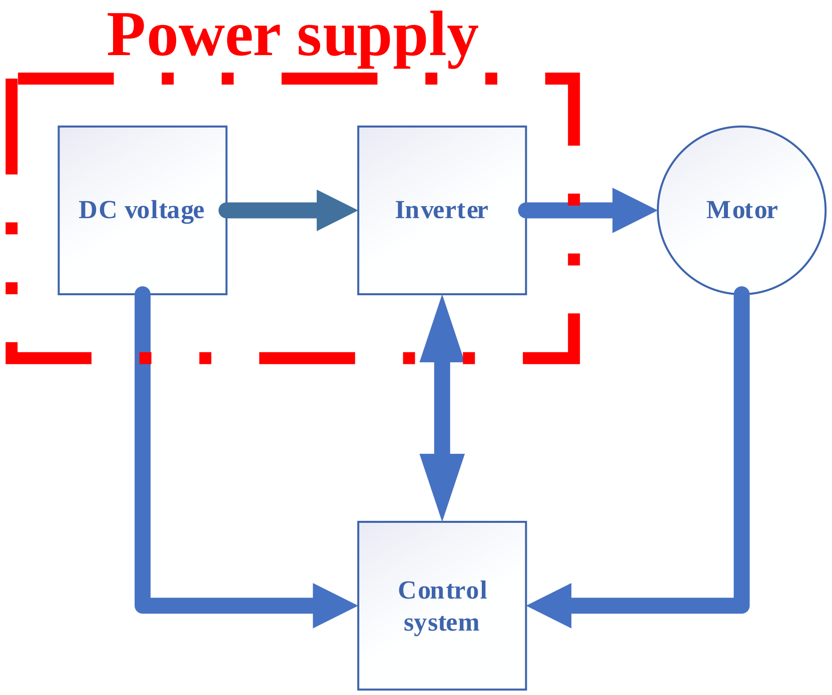


**Figure 1.2 Control of PMSM system.**

For such system to be handled and controlled properly, analyzing and studying the behavior of PMSM is very much needed, which requires a thorough knowledge of the latter dynamic model, this knowledge grants engineers the ability to predict how the controlled system would react to various operating conditions. Three phases PMSM usually rotates with a velocity proportional with the supply alternating current frequency. Voltages supplied to the stator have equal frequency and amplitude, but their phases are $120^o$ apart of each other. The rotor permanent magnets produce a constant magnetic field, which interacts with the stator rotating magnetic field causing the rotor to spin with a synchronous speed. Depending on the structure of the rotor magnets, PMSM can be classified into several types, each type comes with different features such as inductances that vary based on the rotor position, different torque characteristics, as well as a variety of preferred operating applications and conditions.

## 1.2 PMSM TYPES

### 1.2.1 Surface permanent magnet synchronous motor (SPMSM)

This type of PMSM has its permanent magnets placed on the outer surface of the rotor, resulting in a simple magnetic circuit design, and a linear characteristic of speed, torque, and voltages. However, this simplicity comes at the cost of a relatively narrow operating range, sometimes limited to high-speed applications, and a less robust structure compared to other types of PMSM [40]. SPMSM q-axis and d-axis inductances are constant, equal to each other, and independent of the rotor position. Moreover, the continuous centrifugal force applied on the magnets can sometimes damage their attachment to the rotor. Figure 1.3 depicts a rotor of a SPMSM.

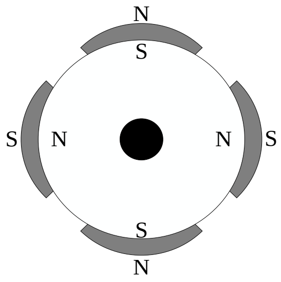


**Figure 1.3 Rotor of SPMSM**

### 1.2.2 Inset permanent magnet synchronous motor (IPMSM)

IPMSM permanent magnets are planted on the surface of its rotor, this structure provides strong compactness, and a remarkable flux-weakening credentials [41]. The d-axis and q-axis inductances values of this motor are slightly different from each other. Overall, IPMSM is one of the best candidates when it comes to precise speed control, and robust handling of various types of load charges and applications such as robotics, HVAC systems, and electrical vehicles systems [42]. IPMSM rotor structure can be seen in Figure 1.4.

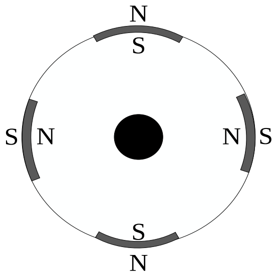


**Figure 1.4 Rotor of IPMSM**

### 1.2.3 Concentrated-flux PMSM (CFPMSM)

In this type of PMSM, a set of ferrite magnets are buried inside the rotor in a flared-shape position, resulting in a concentrated magnetic flux, which assures higher inductance, more efficient electromagnetic torque production, and lower acoustic noise [43]. Additionally, just like IPMSM, this motor permanent magnets are well protected against demagnetization and mechanical disturbances such as vibrations. Moreover, the inductance of the q-axis is higher than that of the d-axis. Concentrated-flux PMSM rotor structure can be seen in Figure 1.5.

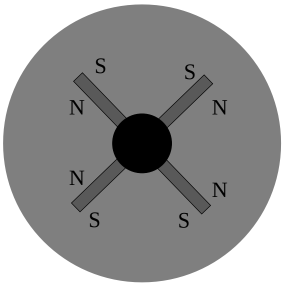


**Figure 1.5 Rotor of Concentrated-flux PMSM**

### 1.2.4 Interior permanent-magnet synchronous motor (INPMSM)

This type of PMSM has its permanent magnets buried deep inside the rotor as can be seen Figure 1.6. the d-axis and q-axis inductances of this motor have different values and varies based on the rotor position. The main pros of this motor are its remarkable torque, superior density, and its wide range of operating speeds [44]; however, INPMSM is notably sensitive to high temperatures [45]. When the temperature of this motor increases, the eddy current loss also increases, which sometimes can cause an irreversible demagnetization to the permanent magnets.

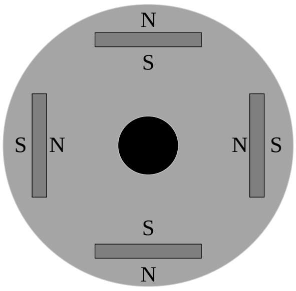


**Figure 1.6 Rotor of INPMSM**

## 1.3 MODELING OF PMSM

PMSM can be modeled by accounting for its electrical, mechanical, and magnetic dynamics. A dynamic model of PMSM consists of a set of electromechanical differential and algebraic equations that relates flux linkages, currents, torque, voltages, and rotating speed presenting a fast-functioning set of state variables that require a rapid and delicate simulation calculations due to electrical machines variables small time constants.

Detailed modeling of PMSM is essential for an accurate prediction of its produced speed, torque, and currents during both transient and steady-state operations [46, 47].

In order to simplify PMSM mathematical model, several hypotheses are taken into account:

- An optimally designed PMSM is still under to risk of experiencing saturation in some areas of the stator and rotor iron core resulting in an undesirable outcome in the overall PMSM control performance. The higher the energy production of the permanent magnets the higher the risk of saturation. Nevertheless, physical modeling of PMSM ignores magnetic saturation which is also the case in this thesis.
- PMSM experiences the occurrence of iron losses due to eddy and hysteresis currents phenomenon which are related to the supply voltage frequency, and the magnetic flux in the machine iron core. Most control technique ignore the presence of iron losses to focus on solving other more important issues. Therefore, the effects and presence of hysteresis and eddy currents is ignored here.
- Throughout long term operation of PMSM, temperature effect may occur in its physical form causing variations on its crucial internal parameters such as the rotor winding resistance and such. These effects are generally ignored in this thesis.
- To establish a smooth rotating magnetic field during steady-state operation, PMS machines employ stator windings distributed sinusoidally around the stator core's circumference. Despite this design intent, magnetic flux fidelity is compromised by stator slot harmonics. Physical models deliberately exclude these harmonics, approximating each phase's stator winding as a continuous sinusoidal distribution.

### 1.3.1 Equations of PMSM in the three-phase axis ($abc$)

In most cases, the windings of the 3-phases of PMSM stator are connected in star configuration due to its simplicity, equal current distribution, and high efficiency.

Figure 1.7 depicts the electrical circuit of a 3-phases PMSM stator connected in star configuration. $V_a$, $V_b$, and $V_c$ are the source voltages fed to PMSM and $i_a$, $i_b$, and $i_c$ are their respective alternating currents. $\vartheta$ is the phase difference between every two phases where $\vartheta = 2\pi/3$ [48].

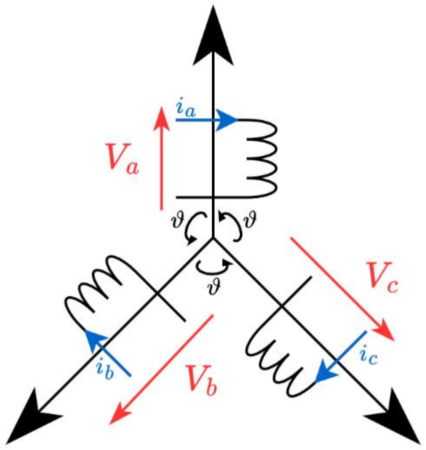


**Figure 1.7 PMSM stator windings connected in star configuration**

The dynamics of PMSM can be comprehensively defined by two types of equations which are:

- Electrical equations.
- Mechanical equations.

### 1.3.1.1 PMSM electrical equations

The 3-phase voltages fed to PMSM can be expressed as in (1-1):

$$V_{abc} = R_s i_{abc} + \frac{d\varphi_{abc}}{dt} \qquad (1\text{-}1)$$

Where $V_{abc} = [V_a\ V_b\ V_c]^T$ and $i_{abc} = [i_a\ i_b\ i_c]^T$ as mentioned before are the stator voltages and currents. $R_s$ in (1-2) represents the stator resistance. $\varphi_{abc}$ represents the stator fluxes and it can be written as in (1-3) [48]:

$$R_s = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \qquad (1\text{-}2)$$

And:

$$\varphi_{abc} = L_{ss} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \varphi_{af} \\ \varphi_{bf} \\ \varphi_{cf} \end{bmatrix} \qquad (1\text{-}3)$$

Where:

$$\begin{bmatrix} \varphi_{af} \\ \varphi_{bf} \\ \varphi_{cf} \end{bmatrix} = \varphi_f \begin{bmatrix} \cos(\theta) \\ \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \qquad (1\text{-}4)$$

Here $\theta$ is the electrical angular position of the rotor. $\varphi_{af}$, $\varphi_{bf}$, and $\varphi_{cf}$ represents the fluxes created by the permanent magnets and interacts with the stator windings. $\varphi_f$ is the flux linkage of the rotor magnets.

$L_{ss}$ is a combination of constant and variable inductances where the constants are denoted by $L_{s0}$, and the variables are denoted by $L_{s1}$. Overall, $L_{ss}$ can be written as in (1-5):

$$L_{ss} = L_{s0} + L_{s1} \tag{1-5}$$

Where:

$$L_{s0} = \begin{bmatrix} L_0 & M_{s0} & M_{s0} \\ M_{s0} & L_0 & M_{s0} \\ M_{s0} & M_{s0} & L_0 \end{bmatrix} \tag{1-6}$$

$$L_{s1} = \begin{bmatrix} \cos(2\theta) & \cos(2\theta - \frac{2\pi}{3}) & \cos(2\theta - \frac{4\pi}{3}) \\ \cos(2\theta - \frac{2\pi}{3}) & \cos(2\theta - \frac{4\pi}{3}) & \cos(2\theta) \\ \cos(2\theta - \frac{4\pi}{3}) & \cos(2\theta) & \cos(2\theta - \frac{2\pi}{3}) \end{bmatrix} \tag{1-7}$$

Here $M_{s0}$ is the mutual inductance between two phases where $M_{s0} = -\frac{1}{2}L_{s0}$. $L_0$ is each phase own inductance and its value is a positive constant dependent on the machine itself.

#### 1.3.1.2 PMSM mechanical equations

The last equation that completes the mathematical model of PMSM is the mechanical equation as in (1-8). Inherently, it describes the dynamics of PMSM rotor and how it functions based on the motor mechanical and electrical parameters:

$$J\frac{d\omega_r}{dt} = T_e - T_r - B\omega_r \tag{1-8}$$

Where:

$\omega_r$: Mechanical speed of the motor.

$T_e$: induced electromagnetic torque of PMSM.

$T_r$: Resistive torque (Load charge).

$J$: Machine moment inertia.

$B$: Vicious friction.

Ideally, $B$ and $J$ are positive constants and their values are completely based on the machine mechanical structure. $T_r$ is basically an external disturbance and it can be variable or constant depending on the type of the load charge. $T_e$ value adapts to match $T_r$ and its mathematical form is dependent on the d and q-axis currents and fluxes which will be presented in next sections.

### 1.3.2 Park transformation

Park transformation is a mathematical transformation used to simplify the dynamics of AC motors such as PMSM [49]. It converts the signals of the voltages, currents, and fluxes from time-domain three-phase coordinates $(abc)$ to a rotating reference coordinate denoted by $(dq0)$. This transformation in coordinates simplifies calculations and facilitates the control and prediction of said machine behavior. Moreover, an anti-Park transformation is also necessary to retrieve original three-phase signals values. Park transformation and its anti-version can be seen in (1-9), and (1-10) respectively:

$$\begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -sin(\theta - \frac{2\pi}{3}) & -sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (1\text{-}9)$$

$$\begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos(\theta - \frac{2\pi}{3}) & -sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & -sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} \quad (1\text{-}10)$$

Here $X$ is either $V$, $i$, or φ. $d$ and $q$ are the direct and quadratic coordinates respectively. $X_0$ (Zero component) is negligible and often ignored.

### 1.3.3 Inverse Park transformation

Similar to park transformation, Inverse Park is also a mathematical transformation and it converts the rotating frame reference $dq$ of the entered signals to the stationary stator two frame reference denoted by alpha-beta $(\alpha\beta)$ [50]. This transformation is useful and necessary for space vector pulse width modulation (SVPWM) of three-phase inverters. Inverse Park transformation can be utilized using (1-11):

$$\begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X_d \\ X_q \end{bmatrix} \quad (1\text{-}11)$$

### 1.3.4 Application of Park transformation on the model of PMSM

By applying Park transformation on PMSM mathematical model, we can obtain a new set of simplified equations fixed to the machine rotor and synchronous rotating flux of the stator. The new equations of voltages can be seen in (1-12), and (1-13):

$$V_d = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \tag{1-12}$$

$$V_q = R_s i_q + L_q \frac{di_q}{dt} + \omega L_d i_d + \omega \varphi_f \tag{1-13}$$

Here $\omega$ is the electrical velocity where $\omega = P_n * \omega_r$, and $P_n$ is the number of pair of poles.

The transmitted power $P(t)$ can be expressed as in (1-14):

$$P(t) = \frac{3}{2}(V_d i_d + V_q i_q) \tag{1-14}$$

We replace $V_d$ and $V_q$ in (1-14) by their expressions in (1-12) and (1-13) to get:

$$P(t) = \frac{3}{2}\left[R_s\left(i_d^2 + i_q^2\right) + \left(L_d \frac{di_d}{dt} i_d + L_q \frac{di_q}{dt} i_q\right) + \frac{d\theta}{dt}((L_d i_d + \varphi_f) i_q - L_q i_q i_d)\right] \tag{1-15}$$

- $\frac{3}{2} R_s\left(i_d^2 + i_q^2\right)$: represents the ohmic losses of the motor.
- $L_d \frac{di_d}{dt} i_d + L_q \frac{di_q}{dt} i_q$: represents the variation in stator windings magnetic energy.
- $(L_d i_d + \varphi_f) i_q - L_q i_q i_d$: represents the electromagnetic power.

It’s known that:

$$T_e = \frac{3}{2} P\left[(L_d i_d + \varphi_f) i_q - L_q i_q i_d\right] \tag{1-16}$$

Which also can be written as:

$$T_e = \frac{3}{2} P\left[\left(L_d - L_q\right) i_d i_q + \varphi_f i_q\right] \tag{1-17}$$

## 1.4 IMPLEMENTATION OF SVPWM BASED INVERTER FOR THE CONTROL OF PMSM

### 1.4.1 Voltage source inverter

In advanced control techniques, the use of an inverter for the control of PMSM is essential and indispensable [51]. Inverters are electrical converters capable of transforming direct state of voltages or currents into alternative state [52]. This is achieved using diodes and semiconductors that can be controlled by various algorithms such as SVPWM [53], or carrier-based PWM [54]. Inverters can generate output voltages or currents with a wide range of frequencies and amplitudes depending on the desired speed or load charge requirements. In this particular research, we use voltage source inverter (VSI) to feed a three-phase PMSM. Figure 1.8 depicts the schemes of PMSM controlled by SVPWM based VSI.

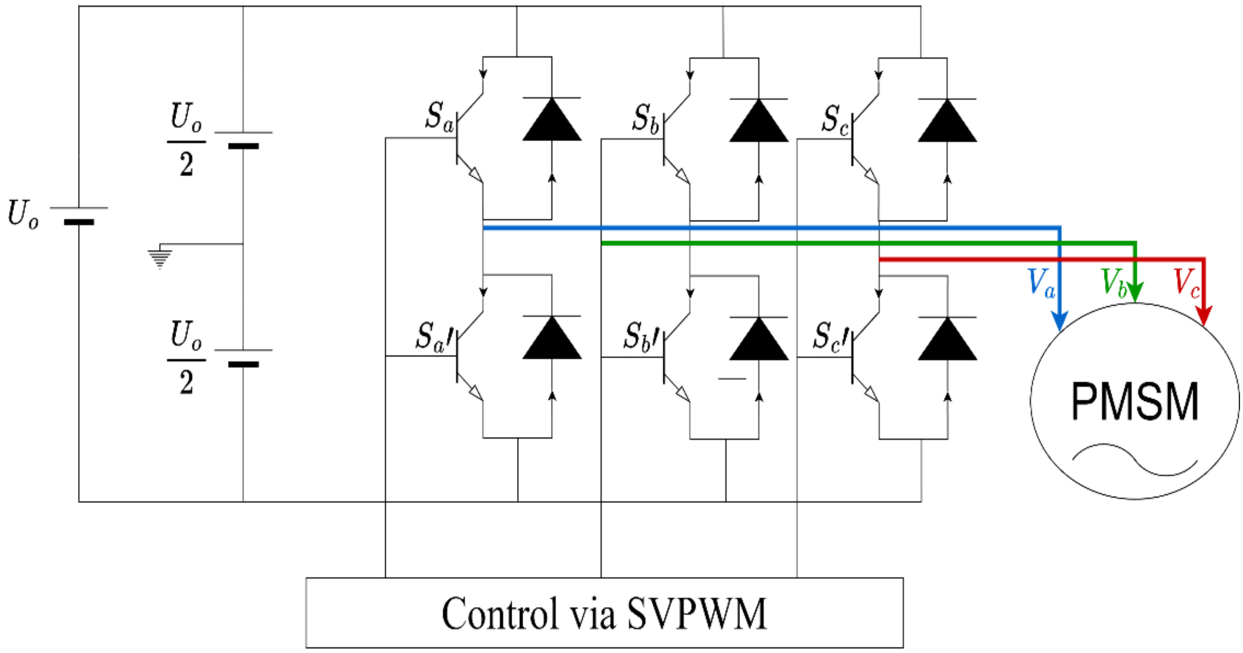


**Figure 1.8 Control of PMSM via SVPWM/VSI**

Considering the motor is wired in star configuration, all its phases interact in a neutral common point $n$. Here $S_\alpha$, $S_\alpha'$, $S_b$, $S_b'$, $S_c$, and $S_c{}'$ are inverter gates, when a pulse is supplied to these gates, their respective transistor closes and permits the passage of current. The duty cycle of how long and when a gate is activated can be determined using SVPWM which will be explained in the next section.

As can be seen in Figure 1.8, each phase voltage is supplied from between a duo of transistors, these duos are called legs, each leg transistors cannot be activated at the same time. Based on that, there exist only 8 possible sets of what gates are activated. Moreover, the state of each leg can be written in binary sequences, and their respective voltages can be described by vectors. Further, the binary sequence consists of three digits $[D_a D_b D_c]$, each digit describes the status of the upper transistor of each leg where $D_a$, $D_b$, $D_c$ are digits of the legs of $V_a$, $V_b$, and $V_c$ voltages respectively. These digits can either be 0 or 1, when a digit is equal to 1, it means that the upper gate of its respective leg is activated, but when that digit is 0 it means that the lower gate of its respective leg is the one that is activated. Considering $U_o = V_{dc}$, Table 1 shows the switching states and their respective binary sequence, space vector voltages, and phase voltages.

**Table 1 the respective space vector and phase voltages of each switching state**

| Vector | *binary sequance* | $V_a$ | $V_b$ | $V_c$ |
|---|---|---|---|---|
| $\overrightarrow{V_0}$ | [000] | 0 | 0 | 0 |
| $\overrightarrow{V_1}$ | [100] | $\frac{2V_{dc}}{3}$ | $\frac{-V_{dc}}{3}$ | $\frac{-V_{dc}}{3}$ |
| $\overrightarrow{V_2}$ | [110] | $\frac{V_{dc}}{3}$ | $\frac{V_{dc}}{3}$ | $\frac{-2V_{dc}}{3}$ |
| $\overrightarrow{V_3}$ | [010] | $\frac{-V_{dc}}{3}$ | $\frac{2V_{dc}}{3}$ | $\frac{-V_{dc}}{3}$ |
| $\overrightarrow{V_4}$ | [011] | $\frac{-2V_{dc}}{3}$ | $\frac{V_{dc}}{3}$ | $\frac{V_{dc}}{3}$ |
| $\overrightarrow{V_5}$ | [001] | $\frac{-V_{dc}}{3}$ | $\frac{-V_{dc}}{3}$ | $\frac{2V_{dc}}{3}$ |
| $\overrightarrow{V_6}$ | [101] | $\frac{V_{dc}}{3}$ | $\frac{-2V_{dc}}{3}$ | $\frac{V_{dc}}{3}$ |
| $\overrightarrow{V_7}$ | [111] | 0 | 0 | 0 |

### 1.4.2 Space vector modulation

SVM is a modulation scheme used to adjust the frequency and average value of voltages fed to AC machines such as PMSM and induction motor. First, a desired value of $\alpha\beta$ voltages frame reference denoted by $V_r$ is supplied to SVM algorithm, which deduces a certain pattern of pulses that are fed to VSI transistors to regulate the three phase voltages fed to the motor just so they reach as close of value to the desired $V_r$ as possible where:

$$V_r = V_{\alpha(r)} + jV_{\beta(r)} = \sqrt{V_{\alpha(r)}^2 + V_{\beta(r)}^2} \qquad (1\text{-}18)$$

Here $V_{\alpha(r)}$ and $V_{\beta(r)}$ are in the fixed frame reference. During operation, $V_r$ rotates around a hexagon where its value changes based on the electrical angle $\theta$. This hexagon can be divided into 6 sectors, each sector is bounded by two space vectors as can be seen in Figure 1.9.

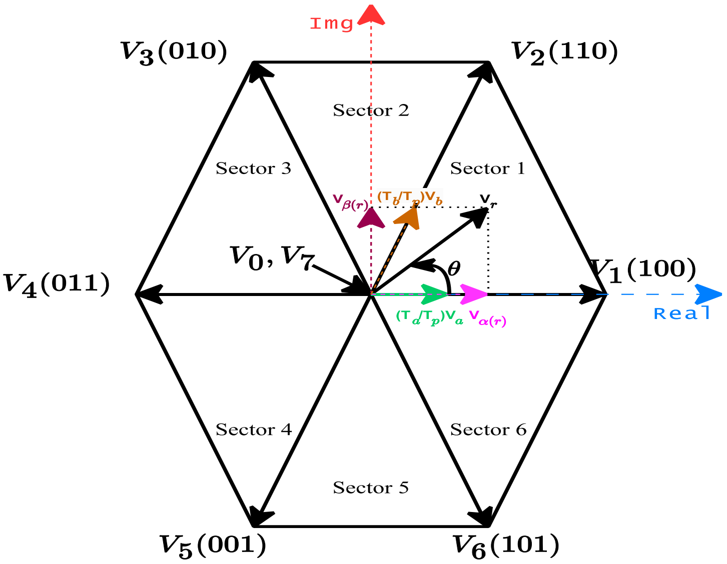


**Figure 1.9 Representation of the switching hexagon and the voltage reference vector**

Depending on the value of $\theta$, SVM utilizes the space vectors $V_a$ and $V_b$ of $V_r$ current sector $(i)$, and the null space vectors $V_0$ and $V_7$. We can use (1-19) to find $\theta$ and the current active sector where $V_r$ is located.

$$\theta = tan^{-1}(\frac{V_{\alpha(r)}}{V_{\beta(r)}}) \tag{1-19}$$

After choosing a switching period $T_p$ so small that $V_r$ is considered constant during it, we activate the null and the current sector space vectors one at time each for a specific duration during $T_p$ just so the average of the output voltages during $T_p$ is equal to $V_r$. The relation between $T_p$ and the durations of how long each space vector is activated can be seen in (1-20).

$$T_p = T_{null} + T_a + T_b \tag{1-20}$$

Where:

$$T_{null} = T_0 + T_7; V_{null} = V_0 + V_7 \tag{1-21}$$

For a guaranteed equality between $V_r$ and the average of the generated voltages during $T_p$ we use (1-22):

$$\frac{1}{T_p}\int_t^{t+T_p} \bar{V_r}\,dt = \frac{1}{T_p}\left[\int_t^{t+T_a} \bar{V_a}\,dt + \int_t^{t+T_a+T_b} \bar{V}_b\,dt + \int_{t+T_a+T_b}^{t+T_p} \bar{V}_{null}\,dt\right] \tag{1-22}$$

A simpler expression of (1-22) can be seen in (1-23):

$$\overrightarrow{V_r}T_p = T_a\overrightarrow{V_a} + T_b\overrightarrow{\bar{V}_b} + T_{null}\vec{V}_{null} \tag{1-23}$$

After an active sector is found, we use its two vectors $V_a$ and $V_b$ as a projection for $V_r$ on the real and imaginary axis where $|V_a| = |V_b| = \frac{2}{3}V_{dc}$. When $V_r$ is located is section (1), projection results can be obtained as in (1-24), and (1-25).

$$\text{Real: } V_r\cos(\theta)\,T_p = V_1\cos(0)\,T_a + V_2\cos\left(\frac{\pi}{3}\right)T_b = \frac{2}{3}V_{dc}T_a + \frac{1}{3}V_{dc}T_b \tag{1-24}$$

$$\text{Img: } V_r\sin(\theta)\,T_p = V_1\sin(0)\,T_a + V_2\sin\left(\frac{\pi}{3}\right)T_b = \frac{1}{\sqrt{3}}V_{dc}T_b \tag{1-25}$$

We can easily obtain $T_b$ from (1-25) where:

$$T_b = V_r\sin(\theta)\,T_p\frac{\sqrt{3}}{V_{dc}} \tag{1-26}$$

$T_a$ can be also obtained by replacing (1-26) in (1-24). Finally, $T_{null}$ can be deduced from (1-20).

A generalized formula of how to obtain $T_a$, and $T_b$ based on the number of $V_r$ current sector ($i$) can be seen in (1-27), and (1-28).

$$T_a = \frac{\sqrt{3}\, V_r\, T_p}{V_{dc}} \left( \sin\left(\frac{i}{3}\pi - \theta\right) \right) \tag{1-27}$$

$$T_b = \frac{\sqrt{3}\, V_r\, T_p}{V_{dc}} \left( \sin\left(\theta - \frac{(i-1)}{3}\pi\right) \right) \tag{1-28}$$

Where $i$ is the number of the current sector $(i)$.

### 1.4.3 Space vector pulse width modulation algorithm

SVM can be put to use using a variety of techniques, the most common one is SVPWM scheme which uses pulse width modulation to generate the desired pulses. This can be done by comparing SVM algorithm generated reference to a triangular wave carrier. Whenever the reference exceeds the carrier triangular wave, the corresponding VSI upper gate closes and its lower counterpart opens and vice versa. Figure 1.10 shows an example of a triangular wave compared to switching reference of upper gate Sa. The shape and amplitude of the triangular wave and the reference signal may defer based on the structure and design of SVPWM Simulink model. However, it is worth noting that the maximum amplitude of the triangular wave must always be equal or higher than the maximum amplitude of the reference signal. Finally, equations (1-29), (1-30), and (1-31) presents the resulting phase voltages which obviously are dependent mainly on the DC voltage source magnitude and the VSI active legs.

$$V_{an} = U_{dc}\left(\tfrac{2}{3}S_a - \tfrac{1}{3}S_b - \tfrac{1}{3}S_c\right) \tag{1-29}$$

$$V_{an} = U_{dc}\left(-\tfrac{1}{3}S_a + \tfrac{2}{3}S_b - \tfrac{1}{3}S_c\right) \tag{1-30}$$

$$V_{an} = U_{dc}\left(-\tfrac{1}{3}S_a - \tfrac{1}{3}S_b + \tfrac{2}{3}S_c\right) \tag{1-31}$$

Where $S_a, S_b$, and $S_c$ are either 0 or 1.

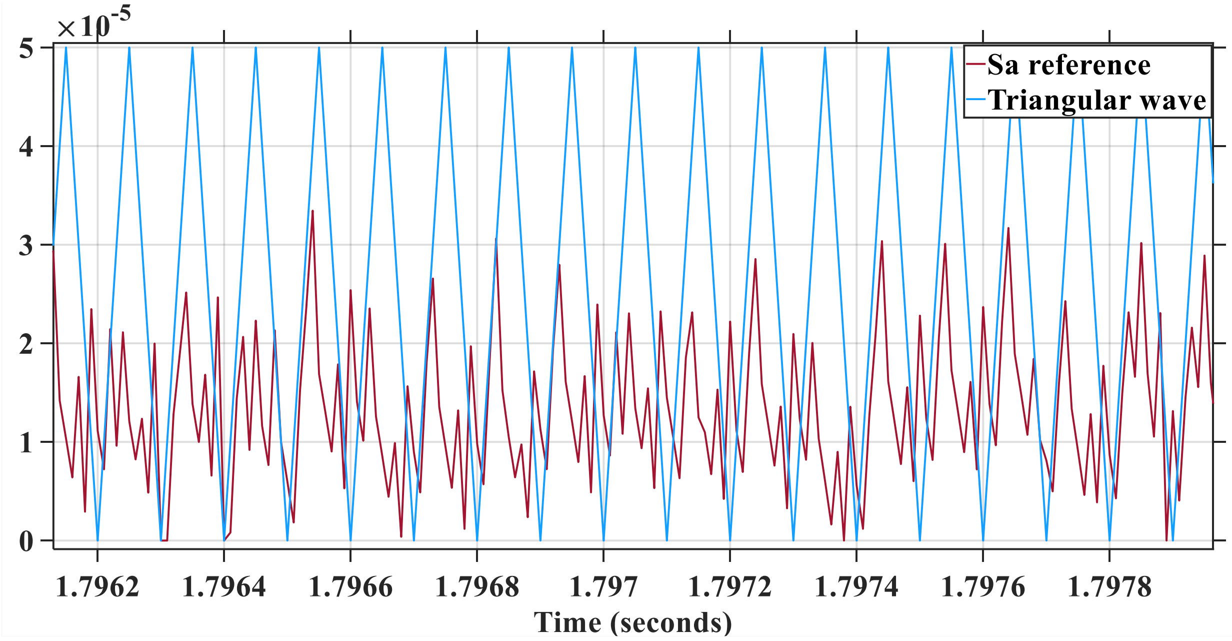


**Figure 1.10 Pulses generation via SVPWM**

## 1.5 CONCLUSION

This chapter showcased a detailed definition and presentation of PMSM structure and working phenomenon, as well as its different classifications depending on the structure of its rotor. Moreover, PMSM detailed mathematical model was also presented in a set of electrical and mechanical equations, these equations were simplified and reduced from three frame reference $(abc)$ to two frame reference $(dq)$ using Park mathematical transformation. Furthermore, VSI was presented and defined along with SVPWM that controls its output voltages.

# Chapter 2: Field oriented control and SPS based tuning of PI

In this chapter, field-oriented control (FOC) was showcased, defined, and implemented for the control of permanent magnet synchronous motor (PMSM) [55]. FOC is a popular vectorial control of AC motors such as three phase induction motor and PMSM. It is most known for its capability of efficiently handling the non-linearity of PMSM by simplifying the latter mathematical model. This can be done by forcing the value of the direct axis current $i_d$ to remain very small and negligible within a tight interval around zero. Hence simplifying the expression of the produced electromagnetic torque resulting in a torque control similar to that of a direct current motor. FOC utilizes a set of controllers to keep the controlled motor mechanical speed, electromagnetic torque, and phase currents at desired values. One of the most common controllers used in FOC is the classical proportional-integral (PI) controller [56]. PI has two tunable gains that can be adjusted using a variety of classical techniques. However, in order to achieve better results, metaheuristic algorithms (MHA) key concepts are explained and implemented to tune the gains of PI. Moreover, feed forward compensation (FFC) technique was also defined and used for a direct FOC of PMSM [57].

## 2.1 FOC OF PMSM DETAILS

Mathematically, the torque developed by electrical motors is proportional to the product of two vectors of flux linkages. The goal of FOC is to make these flux linkages perpendicular as much as possible.

FOC of PMSM main basis is the separation of the components of stator currents electromagnetic torque and flux, and as stated earlier, this can be achieved by setting a reference value for $i_d$ equal to zero. In order to further improve the dynamic behavior of FOC, $i_d$ and $i_q$ currents must be rendered completely independent of each other by using decoupling techniques such as FFC which will be introduced in later section.

There are two types of FOC, direct FOC [58], and indirect FOC [58], and the difference between them is that indirect FOC does not require the utilization of FFC or feedback linearization to completely severe the link between $dq$ current components [59], hence less sensors are needed for real time implementation. However, direct FOC is more robust and much more reliable than indirect FOC. In this research, we use direct FOC method. Furthermore, regardless of what type of FOC is implemented, an accurate rotor position knowledge is always necessary.

When a successful regulation and separation between $i_q$ and $i_d$ is achieved, $i_d$ flux vector in the electromagnetic torque equation can be neglected due to the negligible value of $i_d$, hence reducing (1-17) to (2-1).

$$T_{em} = \frac{3}{2} P \varphi_f i_q \tag{2-1}$$

Since $\varphi_f$ is constant, (2-1) can also be written as:

$$T_{em} = k_t i_q \tag{2-2}$$

Where:

$$k_t = \frac{3}{2} P_n \varphi_f \tag{2-3}$$

Knowing that the rotor flux is constant, this makes $i_q$ the torque component since its orthogonal to the rotor magnetic field and it is also responsible for regulating the value of the produced electromagnetic torque.

In FOC, both $i_q$ and $i_d$, as well as the motor mechanical speed needs to be regulated using controllers such as PI controller, which provides an output based on a closed loop feedback and a setpoint reference of the controlled values.

The global schemes of FOC of PMSM with feedforward compensation (FFC), SVPWM, classical PI controllers, and VSI can be seen in Figure 2.1.

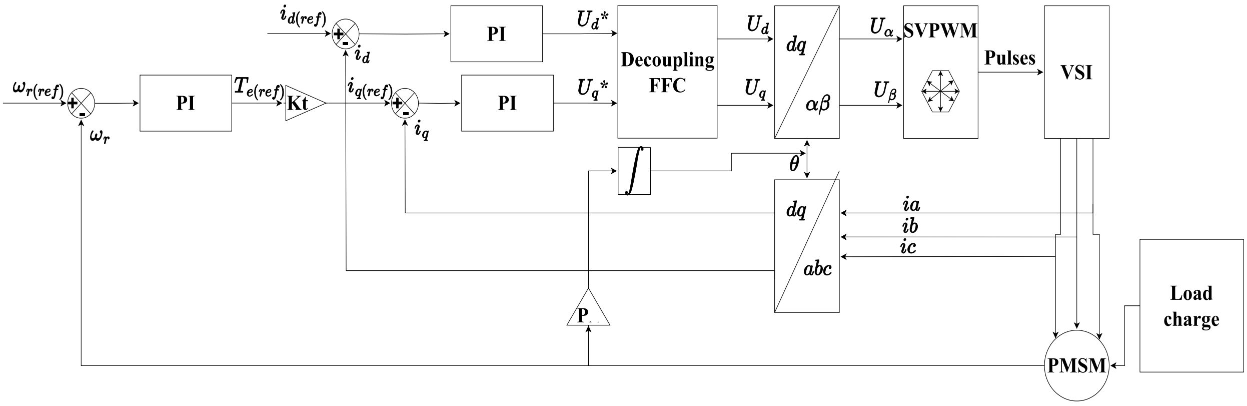


**Figure 2.1 FOC of PMSM full depiction**

### 2.1.1 Feedforward compensation

The aim of FFC in the contest of FOC of PMSM is to reduce the inherent coupling between $i_d$ current that controls the motor magnetic flux, and $i_q$ current that produces its electromagnetic torque [60]. Achieving a perfect decoupling is a very difficult task due to the limitations of electrical motors nonlinearities, parametric changes, and sensors deviations. However, FFC is still essential for a minimized interactions between $dq$ currents, and an overall better motor dynamic response and robustness. The implementation of FFC requires precise knowledge of motor electrical parameters, $dq$ currents, and mechanical speed. Using these parameters and variables, a feedforward signals are produced and injected into the output of $dq$ currents controllers, resulting in an enhanced counterreaction to the coupling effect. Additionally, FFC also helps adjusting the parameters of PI controllers when using classical tuning methods.

The main principal of FFC is the creation of new two control variables presented in Figure 2.2 and denoted by $\varepsilon_d$ and $\varepsilon_q$ where:

$$V_d = V_{d1} - \varepsilon_d \tag{2-4}$$

$$V_q = V_{q1} + \varepsilon_q \tag{2-5}$$

With:

$$V_{d1} = R_s i_d + \ L_d \frac{di_d}{dt} \tag{2-6}$$

$$V_{q1} = R_s i_q + \ L_q \frac{di_q}{dt} \tag{2-7}$$

$$\varepsilon_d = \omega L_q i_q \tag{2-8}$$

$$\varepsilon_q = \omega L_d i_d + \omega \varphi_f \tag{2-9}$$

The application of these mathematical term's grants both $i_d$ and $i_q$ independence of $\varepsilon_d$ and $\varepsilon_q$, respectively.

In laplace domain, $i_d$ and $i_q$ can now be written as in (2-10), and (2-11) where $s$ is laplace operator:

$$i_d = \frac{V_{d1}}{R_s + \ sL_d} \tag{2-10}$$

$$i_q = \frac{V_{q1}}{R_s + \ sL_q} \tag{2-11}$$

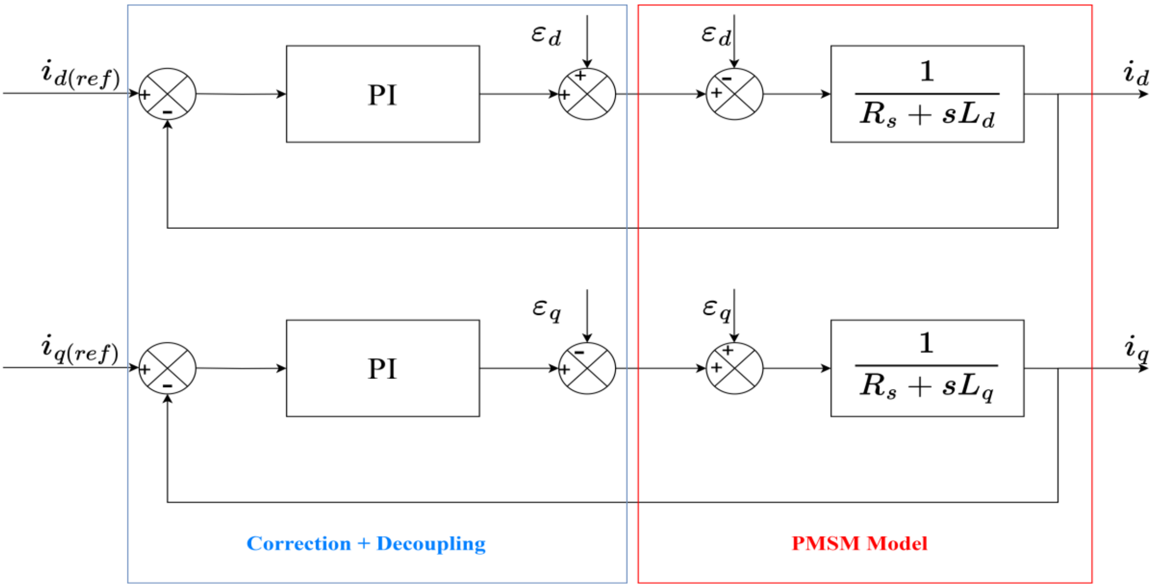


**Figure 2.2 Decoupling via FFC**

### 2.1.2 Zero $i_d$ current benefits

In FOC of PMSM, a negligible $i_d$ current maximizes the motor overall torque production efficiency, reduces current losses while maintaining the same nominal power, and decouples the flux and torque control which leads to a linearized electromagnetic torque that only relies on quadratic axis $i_q$ current. Moreover, in salient PMSM, a negative $i_d$ could irreversibly demagnetize permanent magnets and weaken the rotor flux linkage, that’s why reducing the latter ($i_d$) to a value very close to zero mitigates the risk of demagnetization.

### 2.1.3 PI design and structure

The role of a controller is maintaining an output value of the controlled variable equal to the desired setpoint reference value even at the presence of external and internal disturbances. PI is one of oldest and most common used controllers [61], and it consists of two parts, the proportional part, and the integral part. The proportional part is used to accelerate the dynamic response of the controller, whereas the integral gain is used to reduce the steady-state-error (SSE) of setpoint reference tracking [62]. Additionally, PI controller in the parallel form presents a sum of the proportional and the integral part as can be seen in Figure 2.3 and equations (2-12), and (2-13):

$$U(s) = E(s)k_p + E(s)k_i\frac{1}{S} \qquad (2\text{-}12)$$

Where:

$$E(s) = R(s) - Y(s) \qquad (2\text{-}13)$$

And:

$k_p$: Proportional gain.

$k_i$: Integral gain.

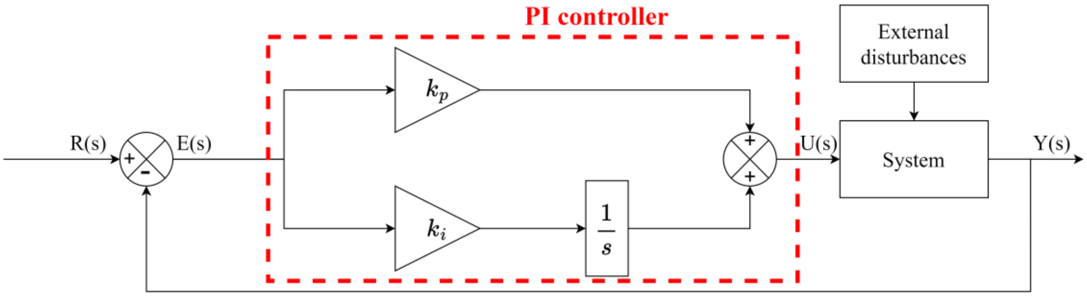


**Figure 2.3 diagram of system control via PI controller**

The output transfer function can be written as

$$U(s) = E(s)[k_p + k_i \frac{1}{S}] \tag{2-14}$$

PI controller equation can also be described by the following equation:

$$PI \rightarrow \frac{U(s)}{E(s)} = \frac{1 + sT_1}{sT_2} \tag{2-15}$$

With:

$$k_p = \frac{T_1}{T_2}; k_i = \frac{1}{T_2} \tag{2-16}$$

### 2.1.4 Classical tuning of PI controllers

### 2.1.4.1 $i_q$ current controller

From equations (2-11) and (2-15), we can construct a closed loop diagram for the controled of $i_q$ current as can be seen in Figure 2.4:

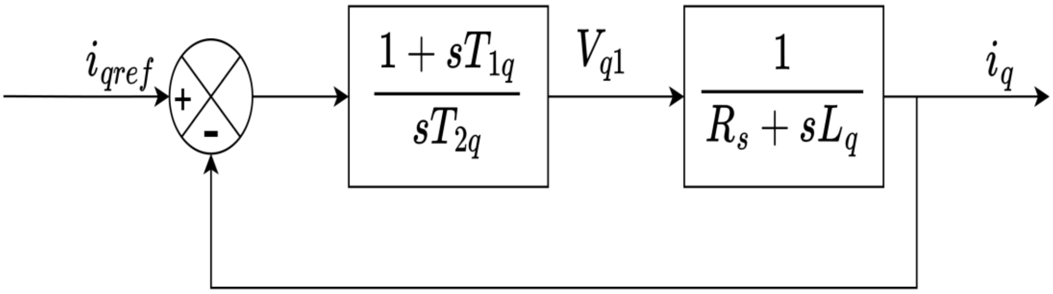


**Figure 2.4 Closed loop control of $i_q$**

Open loop transfer function ($T_{OL}$) of Figure 2.4 can be written as:

$$T_{OL} = \frac{1 + sT_{1q}}{sT_{2q}(R_s + sL_q)} = \frac{1 + sT_{1q}}{sT_{2q}R_s(1 + s\frac{L_q}{R_s})} \tag{2-17}$$

Using compensation of poles technique [21], we get:

$$1 + sT_{1q} = 1 + \ s\frac{L_q}{R_s} \tag{2-18}$$

Which meets the condition:

$$\tau_q = T_{1q} = \frac{L_q}{R_s} \tag{2-19}$$

Where $\tau_q$ is the electrical time constant of $q$ axis.

After replacing (2-19) in (2-17), we get:

$$T_{OL} = \frac{1}{sR_sT_{2q}} \tag{2-20}$$

Closed loop transfer function ($T_{CL}$) of (2-20) can be written as:

$$T_{CL} = \frac{TOL}{1 + TOL} \tag{2-21}$$

$$T_{CL} = \frac{1}{1 + sR_sT_{2q}} \tag{2-22}$$

Equation (2-22) can be written as ( $\frac{1}{1+s\tau_q}$), with identification we find that:

$$\tau_q = R_sT_{2q} \rightarrow T_{2q} = \frac{\tau_q}{R_s} \tag{2-23}$$

We impose a response time $T_s$ where:

$$T_s = 3\tau_q \tag{2-24}$$

Then:

$$T_{2q} = \frac{T_s}{3R_s} \tag{2-25}$$

After replacing (2-23) in (2-16) we get:

$$k_{pq} = \frac{R_s\tau_q}{\tau_q} = R_s;\ k_{iq} = \frac{R_s}{\tau_q} = \frac{R_s^2}{L_q} \tag{2-26}$$

### 2.1.4.2 $i_d$ current controller

Similar to $i_q$ current controller, $i_d$ current controller can be obtained used the closed loop depicted in Figure 2.5:

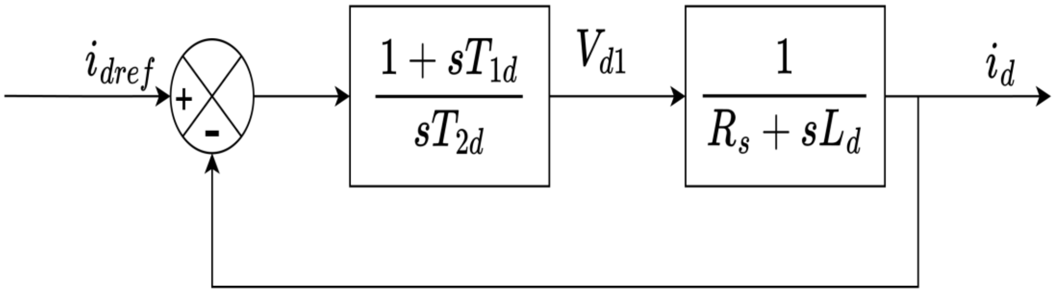


**Figure 2.5 Closed loop control of $i_d$**

$T_{OL}$of $i_d$ current control can be written as:

$$T_{OL}=\frac{1+sT_{1d}}{sT_{2d}(R_s+sL_d)}=\frac{1+sT_{1d}}{sT_{2d}R_s(1+s\frac{L_d}{R_s})} \tag{2-27}$$

$$T_{1d}=\frac{L_d}{R_s} \tag{2-28}$$

After replacing (2-28) in (2-27) we can write the $TCL$ as:

$$T_{cL}=\frac{1}{1+sR_sT_{2d}}=CLTF=\frac{1}{1+s\tau_d} \tag{2-29}$$

Where $\tau_d=R_sT_{2d}\rightarrow T_{2d}=\frac{\tau_d}{R_s}$.

Overall, $k_{pd}$ and $k_{id}$ can calculating using

$$k_{pd}=\frac{R_s\tau_d}{\tau_d}=R_s;\ k_{id}=\frac{R_s}{\tau_d}=\frac{R_s^2}{L_d} \tag{2-30}$$

### 2.1.4.3 Mechanical speed controller

The mechanical speed $\omega_r$ control consists of two subsystems:

- Subsystem of the regulation of $i_q$ and the electromagnetic torque part.
- Subsystem of the motor mechanical part.

Closed loop control of $\omega_r$ can be seen in Figure 2.6:

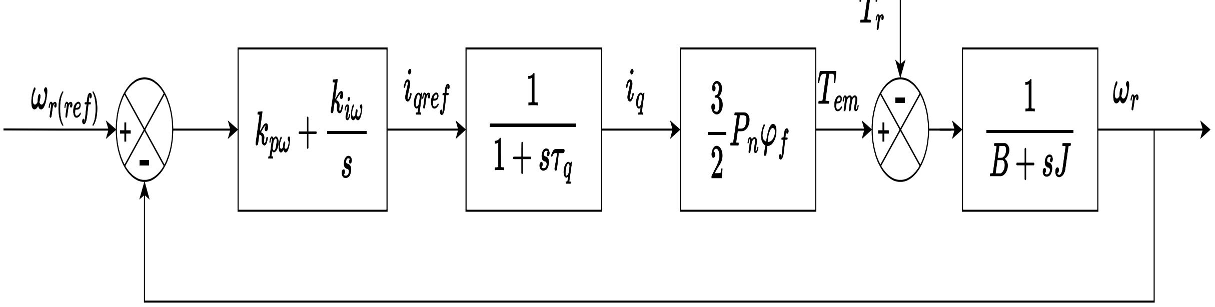


**Figure 2.6 Closed loop control of $\omega_r$**

The diagram in Figure 2.6 can be simplified and depicted as in Figure 2.7.

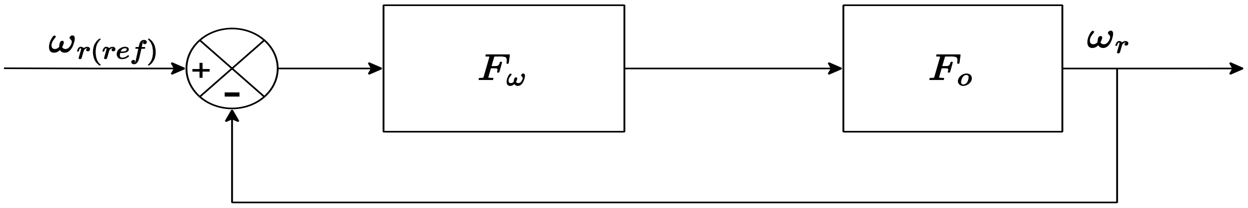


**Figure 2.7 Simplified Closed loop control of $\omega_r$.**

Where $F_\omega$ is the speed control function that can be written as:

$$F_\omega(s) = k_{p\omega} + k_{i\omega}\frac{1}{S} = \frac{1 + sT_{1\omega}}{sT_{2\omega}} \qquad (2\text{-}31)$$

Where:

$$k_{p\omega} = \frac{T_{1\omega}}{T_{2\omega}};\ k_{i\omega} = \frac{1}{T_{2\omega}} \qquad (2\text{-}32)$$

Suppose $T_r = 0$, $F_o$ can be expressed as:

$$F_o(s) = \frac{P_n\varphi_f}{B(1+s\tau_q)(1+s\tau_m)} \qquad (2\text{-}33)$$

Where $\tau_m$ is the motor mechanical constant and it can be obtained using (2-34):

$$\tau_m = \frac{J}{B} \qquad (2\text{-}34)$$

Additionally, $T_{cL}$of speed regulation is:

$$T_{cL} = \frac{F_\omega F_o}{1 + F_\omega F_o} \qquad (2\text{-}35)$$

After replacing (2-31) and (2-33) in (2-35) we get:

$$T_{cL} = \frac{P_n\varphi_f(k_{p\omega}s + k_{i\omega})}{J\tau_q s^3 + (J + B\tau_q)s^2 + (P_n\varphi_f k_{p\omega} + B)s + P_n\varphi_f k_{i\omega}} \qquad (2\text{-}36)$$

In case we neglect $(J\tau_q)$ and $(B\tau_q)$, the polynomial of (2-36) can be written as:

$$p(s) = JS^2 + \ (P_n\varphi_f k_{p\omega} + B)s + P_n\varphi_f k_{i\omega} \qquad (2\text{-}37)$$

This $T_{cL}$ possess the dynamics of a $2^{nd}$ order equation, thus, we can use the canonical form identification of a $2^{nd}$ order equation written as:

$$\frac{1}{\gamma_0^2}s^2 + \left(\frac{2\xi}{\gamma_0}\right)s + 1 = 0 \qquad (2\text{-}38)$$

Here $\gamma_0$ is the system own pulsation, and $\xi$ is the damping factor. Furthermore, with $\xi$ set to 0.7, and by comparing (2-37) to (2-38) we can easily obtain $\gamma_0$, $k_{p\omega}$, and $k_{i\omega}$ using poles adjustment method [20, 21], where:

$$\gamma_0 = \sqrt{\frac{1}{J}} \tag{2-39}$$

$$k_{p\omega} = \frac{2J\xi\gamma_0 - B}{P_n\varphi_f} \tag{2-40}$$

$$k_{i\omega} = \frac{\gamma_0^2 J}{P_n\varphi_f} \tag{2-41}$$

### 2.1.5 Metaheuristics algorithms and proposed hybrid algorithm

#### 2.1.5.1 Definition of metaheuristic algorithms and how they work

Metaheuristic algorithms (MHA) are basically a modern strategy often employed to find approximate solutions for highly sophisticated mathematical problems and nonlinear systems [63]. In the field of electrical engineering, these algorithms can be applied in various ways. In [64], deep analysis of the latest studies of the usage of MHA for the regulation of reservoirs water supply was presented. In [65], whale optimization algorithm which is a very popular MHA was used to find the maximum power point of a photovoltaic system. Additionally, in [66], proportional-integral-derivative (PID) controller was tuned using MHA and classical methods. The overall results were analyzed and comprehensively compared to see if MHA are superior to classical tuning techniques.

MHA does not guarantee the detection of the best possible solution. However, they are surly capable of providing efficient and reasonable solutions with little information and very limited computational power within a relatively small amount of time compared to exact algorithms such as linear programming.

The initial step of any MHA is the generating of a set $X_s$ of search agents where the total number of search agents in this set is $N_s$, and the length of each $X_s$ parameters matrix is $nPar$ with:

$$X_s = X_{i \ldots\ldots\ldots\ldots N_s} \tag{2-42}$$

$$X_i = [x_i^1, x_i^2 \ldots\ldots\ldots\ldots x_i^{nPar}] \tag{2-43}$$

Further, each search agent $X_i$ gets assigned with a number or a matrix of a random parameters of a possible solution from a preassigned bounded interval denoted by $B_s$. Furthermore, $B_s$ upper and lower bounds can be obtained based on past experience or trial and error method. The smaller is $B_s$, the easier for MHA to obtain a decent solution faster. Moreover, after assigning a possible solution for all search agents from $X_s$, the algorithm will classify and calculate their fitnesses where $F_s = F_{i \ldots\ldots\ldots\ldots N_s}$, and then assign a global optimum $X_{Gbest}$ which is basically the search agent with the best fitness among all other search agents from $X_s$ at the moment. Finally, MHA will initialize an iterative process where in each iteration:

- Search agents will explore and exploit $B_s$ by updating their solutions parameters using mathematical equations based on the used MHA.
- Search agents fitnesses are calculated and the global optimum is rechecked during and after each iteration and then updated when possible.

MHA will not stop until a certain criterion is met, or when it reaches the assigned maximum number of iterations.

### 2.1.5.2 Proposed Symbiotic-particle-sewing metaheuristic algorithm

Symbiotic-particle-sewing (SPS) is a hybrid MHA that combines key features of three algorithms namely:

1. Sewing trainee based optimization (STBO) [67].
2. Symbiotic organism search (SOS) algorithm [68].
3. Particle swarm optimization (PSO) algorithm [69].

What makes SPS special is its balanced exploring and exploiting approach, which enhances the algorithm capability of quickly finding a global optimum without the risk of falling into a local optimum. Additionally, SPS iterative part contains three phases. First two phases are inspired by the initial phase of both STBO and SOS algorithms, these two phases main goal is to explore as much area from $B_s$ as possible. This intense exploration, however, is partially guided, and doesn’t fully rely on random possibilities. As a result, SPS can efficiently identify the area containing the global optimum in a very short amount of time. The iterative three phases of SPS can be defined as follows:

**Phase I:** This phase is based on STBO first phase, it employs the equations presented in (2-44), and (2-45) to execute a partially guided exploration aimed to locate the area that has the highest probability of containing the global optimum of the entire search space $B_s$. This phase updates the parameters of the current search agent $X_i$ by addressing it as a trainee, then a set of search agents denoted by $S$ is generated, $S$ includes all the search agents from $X_s$ that has a better fitness than $X_i$. Furthermore, SPS randomly chooses a search agent from $S$, this search agent is then named $X_j$ and used to help update the parameters and enhance the fitness of $X_i$ where:

$$X_i^P = X_i + r_i \times (X_j - I_j \times X_i) \qquad (2\text{-}44)$$

Where $r_i$ is a random number from [0;1], $I_j$ value is chosen randomly to be either 1 or 2. Here $X_i^P$ is the potential new parameters matrix of $X_i$ which can be determined after comparing the fitnesses $F_i^P$ and $F_i$ of $X_i^P$ and $X_i$ respectively:

$$X_i = \begin{cases} X_i^P, & if\ F_i^P < F_i \\ X_i, otherwise \end{cases} \quad (2\text{-}45)$$

**Phase II:** the second phase of SPS is an imitation of the third phase of SOS called parasitism phase. In this phase, SPS tries to update a single parameter from the entire matrix of solution parameters of each search agent from $X_s$. This can be done by randomly choosing a dimension denoted by $x_i^j$ from the parameter's matrix of $X_i$. The parameter of $x_i^j$ is then replaced by a random value from $B_s$, which generates a new protentional candidate $X_i^P$ that could enhance $X_i$ fitness using (2-45).

**Phase III:** the final phase of SPS is a complete replica of PSO algorithm, it employs the entire iterative process of PSO to guarantee a quick and efficient convergence after the initial phases are successful at locating a global optimum area. What makes PSO exploitation special is its utilization of search agents' memory by adding a new variable called the personal best denoted by $X_i^{Pbest}$ which is the solution parameters of each search agent best achieved fitness so far. This allows for a better guided convergence, and overall faster and better exploitation process. Moreover, this phase updates solutions parameters using (2-46), (2-47), and (2-48):

$$V_i^P = r_1 \times V_i \times \omega + c_1 \times r_2 \times \left( X_i^{Pbest} - X_i \right) + c_2 \times r_3 \times ( X_{Gbest} - X_i) \quad (2\text{-}46)$$

$$\omega = \omega Max - t \times (\frac{\omega Max - \omega Min}{Maxiter}) \quad (2\text{-}47)$$

$$X_i^P = X_i + V_i^P \quad (2\text{-}48)$$

$V_i^P$ and $V_i$ are the new and previous search agent $X_i$ velocities respectively. Maxiter is the maximum allowed iterations. $\omega Max$ and $\omega Min$ are 0.9 and 0.4 respectively. Finally, $c_1$ and $c_2$ are both equal to 2.

**REMARK:** both $X_{Gbest}$ and $X_i^{Pbest}$ are updated after every single phase or iteration.

### 2.1.6 Metaheuristics algorithms-based tuning of PI controllers

Regardless of the controlled system, MHA can be used to tune its controllers. It has been proved repeatedly that the tuning of various controllers' parameters using MHA is more reliable than classical tuning techniques such as trial and error method. In this section, SPS algorithm was used to tune the gains of PI controllers for FOC of PMSM.

Initially, SPS parameters such as $X_s$, $B_s$, $N_s$, Maxiter…., and $nPar$ were established. Then, a special objective function (OBJ) was built to evaluate the fitness of each search agent parameters. The goal of the algorithm was to minimize the selected OBJ as much as possible by updating solutions until the maximum number of iterations is reached.

The controllers were tuned to maintain a small SSE in the presence of external disturbances, and respond as fast as possible with minimal overshoot before fully settling down. The expression of OBJ can be seen in (2-49) where absolute error of the mechanical speed was multiplied by 2 and didn't include the time component forcing the algorithm to prioritize enhancing the transient response and SSE of the speed tracking performance more than minimizing the SSE of the induced $dq$ axis currents:

$$OBJ = \int 2 \times \left|(\omega_{r(ref)} - \omega_r)\right| + \int t \times \left|(i_{q(ref)} - i_q)\right| + \int t \times \left|(i_{d(ref)} - i_d)\right| \qquad (2\text{-}49)$$

After the tuning was complete, optimal parameters of SPS based PI (SPS-PI) were obtained and can be seen in Table 2 along with classically tuned PI (CT-PI) controllers:

**Table 2 SPS-PI and CL-PI parameters**

| | $k_{p\omega}$ | $k_{i\omega}$ | $k_{p(iq)}$ | $k_{i(iq)}$ | $k_{p(id)}$ | $k_{i(id)}$ |
|---|---|---|---|---|---|---|
| SPS-PI | 3.63 | 0.0629 | 266.71 | 5.574 | 73.87 | 51.25 |
| CT-PI | 0.125 | 2.15 | 1.4 | 338 | 1.4 | 297 |

## 2.1.7 Operating limits of PMSM

AC machines operate with maximum efficiency when the operating limits of voltages and currents aren't exceeded. If overcurrent is present, dielectric of windings is the most vulnerable its effects. Dielectric class describes the limit of heat at which windings can operate safely. The cooling system also plays a significant role in setting safe operating limits of heat and current. operating limits range is tighter during steady-state operation.

### 2.1.7.1 Current limits

Usually, steady state current must not exceed the nominal rated current. However, during transient state, it is permissible for the current as high as several time of nominal current. during operation, phase current $i_s$ must be observed or measured all the time. It's worth noting that this current can be presented in terms of $dq$ axis currents as in (2-50).

$$i_s = \sqrt{i_d^2 - i_q^2} \le i_{sn} \qquad (2\text{-}50)$$

Where $i_{sn}$ is the rated safe limit phase current of the motor. In practice, control system must contain currents limiters for $dq$ axis currents. These limiters can be saturation functions that allow the passage of current based on (2-51).

$$i_s = \begin{cases} i_s, & i_s < i_{sn} \\ i_{sn}, & i_s \ge i_{sn} \end{cases} \qquad (2\text{-}51)$$

### 2.1.7.2 Voltage limits

Just like the commended currents, $dq$ axis commended voltages must also not exceed certain limits. Theoretically, the limit can be set by equation (2-52).

$$v_s = \sqrt{v_d^2 - v_q^2} \leq v_{sn} \tag{2-52}$$

For the sake of simplicity, during steady-state, all time derivates are considered to be zero including the terms $L_d \frac{di_d}{dt}$ and $L_q \frac{di_q}{dt}$. Additionally, small voltage drops are also ignored which would mean the terms $R_s i_d$ and $R_s i_q$ can be neglected and the final steady state expressions of $dq$ axis voltages are reduced to (2-53), and (2-54).

$$V_d = \omega L_q i_q \tag{2-53}$$

$$V_q = \omega L_d i_d + \omega \varphi_f \tag{2-54}$$

### 2.1.7.3 Torque limits

in all the presented simulations, the targeted electromagnetic torque $T_e$ was limited to 25 N.m so it doesn’t far exceed the motor nominal torque during transient state response.

### 2.1.8 Numerical comparison between SPS-PI and CT-PI

This section presents a simulation of FOC of PMSM that presents a comparison between SPS-PI and CL-PI controllers. The parameters from the previous section were assigned to both sets of controllers to see how does MHA based PI compares to classically tuned PI when used for FOC of PMSM. For a brief comparison, both set of controllers were subjective to an identical conditions of speed set point reference and external disturbances in the form of a 5 $N.m$ constant load charge. After running a duration of 2 seconds simulation, which included a 100 $rad/s$ speed set point reference at the start, a speed reversal after 0.5 sec, another speed reversal back to positive 100 $rad/s$ after second and then an appliance of a 5 $N.m$ load charge at 1.5 sec. Mechanical speed, electromagnetic torque, and phase current were obtained and can be seen in Figure 2.8, Figure 2.9, and Figure 2.10.

In Figure 2.9, and Figure 2.10, it appears that CL-PI induced phase current and $T_e$ are more stable with way less overshoot than SPS-PI produced $T_e$ and phase current. However, the latter provided a smaller settling time with much better oscillations’ suppression during transient state. In Figure 2.8.(a), when it comes to the produced mechanical speed, SPS-PI seems to be far better than CL-PI due to its superior response, settling time, and overshoot rejection. Additionally, at start-up, CL-PI regulated mechanical speed overshoot exceeded 15% of the change of speed set point reference (SSR), conversely to SPS-PI, which didn’t surpass 1% maximum overshoot and responded way faster than CT-PI. Moreover, right after the reversal of speed SSR, CT-PI managed to track the targeted speed. However, at the cost of an overshoot that exceeded 20% of the change in the desired SSR. SPS-PI once again responded faster than CT-PI with way less overshoot. Finally, after adding a load charge, SPS-PI managed to respond faster and negate the effect of the load charge better than CT-PI. However, as we can see in Figure 2.8 zoom in, SPS-PI SSE was higher than CT-PI SSE where it exceeded 1% of the targeted speed. Nevertheless, general evaluation of the entire Figure 2.8 affirms SPS-PI superiority over CL-PI.

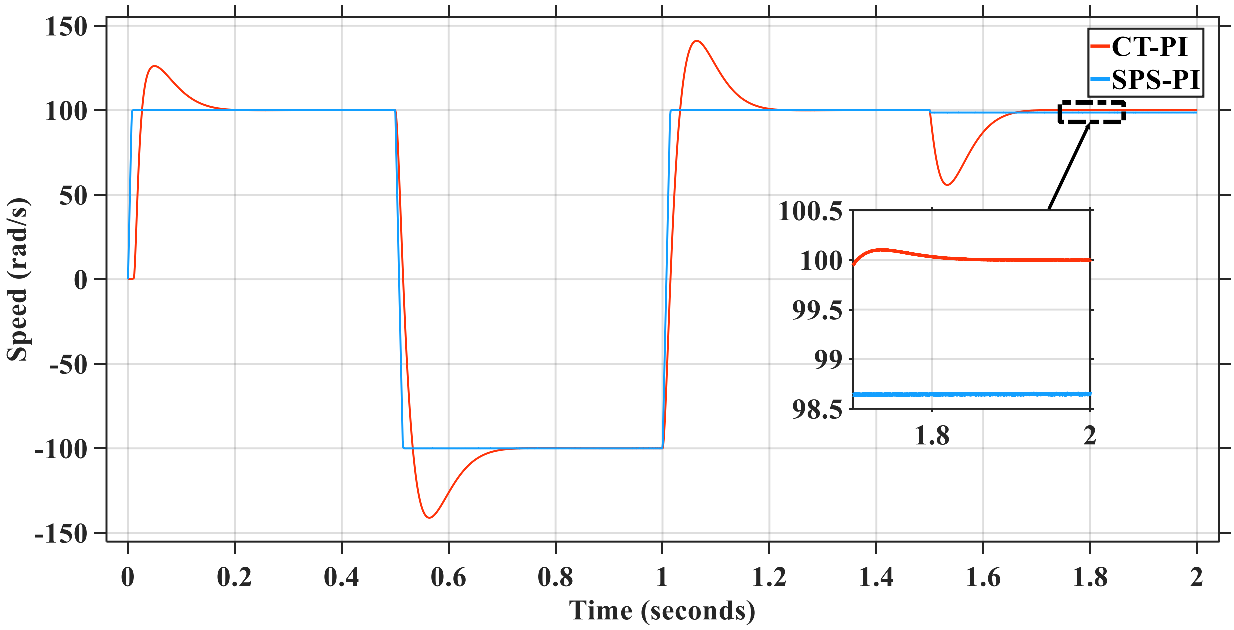


**Figure 2.8 CL-PI vs SPS-PI mechanical speed tracking performance.**

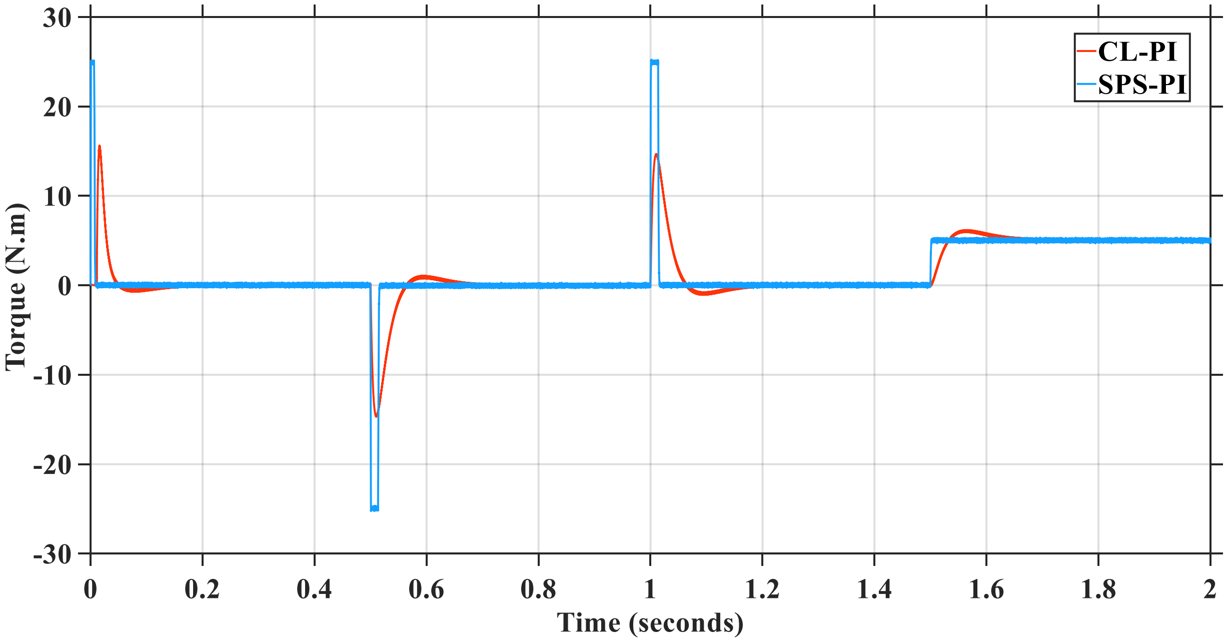


**Figure 2.9 CL-PI vs SPS-PI produced electreomagnetic torque.**

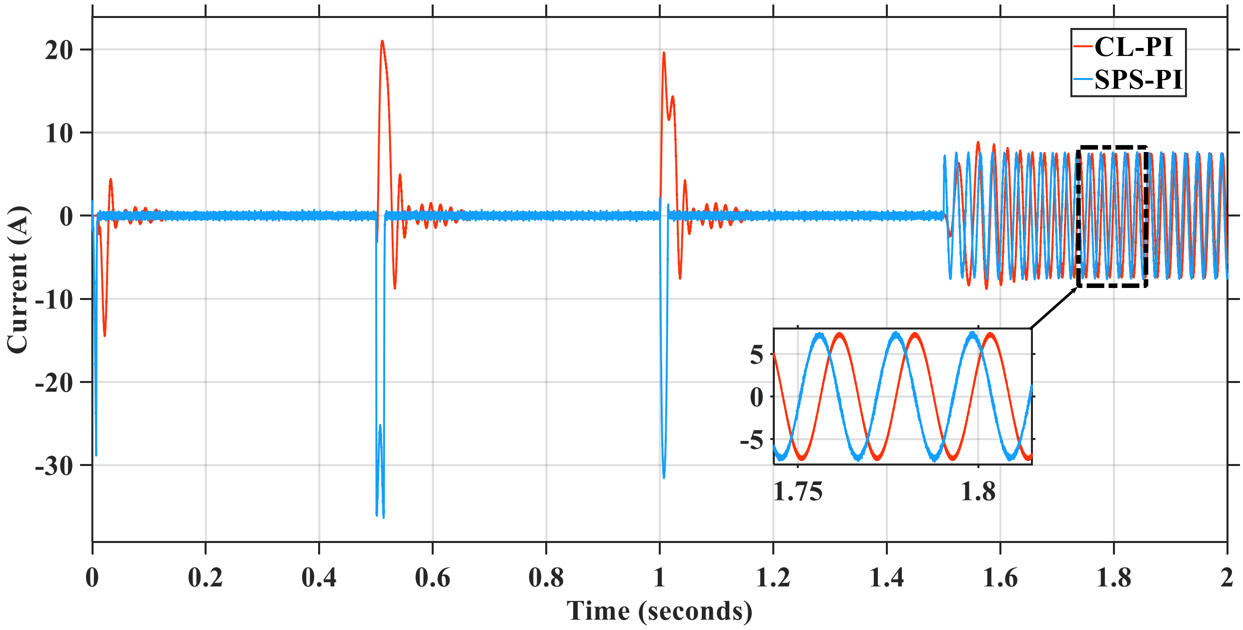


**Figure 2.10 CL-PI vs SPS-PI induced phase currents.**

## 2.2 CONCLUSION

This chapter presented a detailed introduction to FOC of PMSM and how to perform it using classical PI controllers and FFC decoupling technique. Moreover, classical tuning techniques that are used to tune PI controllers for FOC of PMSM were presented. Then, the general functioning concept of MHA was showcased, and a newly proposed hybrid algorithm denoted by SPS was explained and used to tune the gains of PI controllers. Finally, a brief MATLAB/SIMULINK simulation of FOC of PMSM was performed to compare CT-PI to SPS-PI controllers. This simulation resulted in a clear win for SPS-PI. However, despite its clear success, there was still a considerable amount of overshoot and SSE when using SPS-PI. Fortunately, this SSE and overshoot can be reduced using a fuzzy logic controller (FLC) which will be presented in the next section.

# Chapter 3: SPS based fuzzy logic controller

Despite the great success that conventional controllers, such as PI controller have achieved in the industry, they're still far away from matching the adaptivity and potential that the fuzzy logic controller (FLC) can provide when dealing with systems that has a mathematically complex models, or involves an imprecise knowledge of internal parameters and a considerable amount of nonlinearity when controlling it [70-72]. Thus, this chapter main goal is to explain the basics of fuzzy logic control and implement it for an improved speed control of PMSM. After presenting a comprehensive definition of how a FLC works, a conventional interval Type-1 FLC with symbiotic-particle-sewing (SPS) tuned scaling factors denoted by (FC1) was used for speed control of PMSM, where the $dq$ axis currents were controlled using conventional PI controllers. Moreover, in order to further enhance the performance of FC1, its membership functions (MFS) were also tuned using SPS algorithm resulting in a controller with optimized MFS denoted by (F1MF) which seemed to far exceed the performance of the conventional FC1.

## 3.1 FUZZY LOGIC CONTROL BASIC DEFINITIONS

Fuzzy logic is built based on artificial intelligence (AI) reasoning algorithms. These algorithms objective is to replicate how human brains makes precise decisions, statements, and conclusions based on a set of conditions and facts [73]. Take for example if a car gas tank level is at 30%, calling it medium does not provide a clear notion of its level for the driver to decide when to refill it. However, this notion can be made precise and efficient by adding a quantitative percentage to it for example by calling it 52% medium, 32% low, and 0% high, the driver can then use his experience to make a decision if or when he decides to refill the tank. Fuzzy logic uses these quantitative percentages in the form of degrees of memberships that ranges from 0 to 1 as in Figure 3.2. These degrees of memberships can be obtained and analyzed using a set of rules, fuzzy sets, and processing algorithm which analyses a set of quantitative inputs to deduce an output.

The key feature of fuzzy logic is its fuzzy sets, where contrariwise to classical sets, fuzzy sets elements memberships are not restricted to 0 or 1, but instead, it can be any fractional or integer number from the interval [0,1] [74]. Let's consider a group of elements or a universe of discourse denoted by $X$ and contains all the possible percentages of the current level of gas compared to the full capacity of a car tank, and keep in mind that the level of gas percentage is the studied fuzzy set linguistic variable. In the case of classical sets, the level of gas percentage can either be considered low, medium, or high as can be seen in Figure 3.1. However, in the case of fuzzy sets, the gas level can belong to multiple sets but with different values of membership as can be seen in Figure 3.2. If we consider the car gas level percentage at a certain moment to be $x =$

30% where $x \in X$, and after a vertical projection of $x$ on the fuzzy set MFS which are (Low, Medium, and High), we can get a variety of degrees of membership where [75]:

$$\mu_{Medium}(x) = 0.52 \tag{3-1}$$

$$\mu_{Low}(x) = 0.32 \tag{3-2}$$

$$\mu_{High}(x) = 0 \tag{3-3}$$

**REMARK:** Previously, it was necessary that the sum of $x$ degrees of membership with all output membership functions equals one, which means the fuzzy sets are normalized. However, recent research papers affirms that it's not a necessity. In fact, unnormalized fuzzy sets proved to be much more reliable and way more efficient.

In most cases, fuzzy logic reasoning can be adjusted and optimized based on experience and trial and error technique, which is rarely reliable in the field of electrical engineering. However, fuzzy logic high tolerance and big margin of error makes these classical optimizations practical in most machine applications.

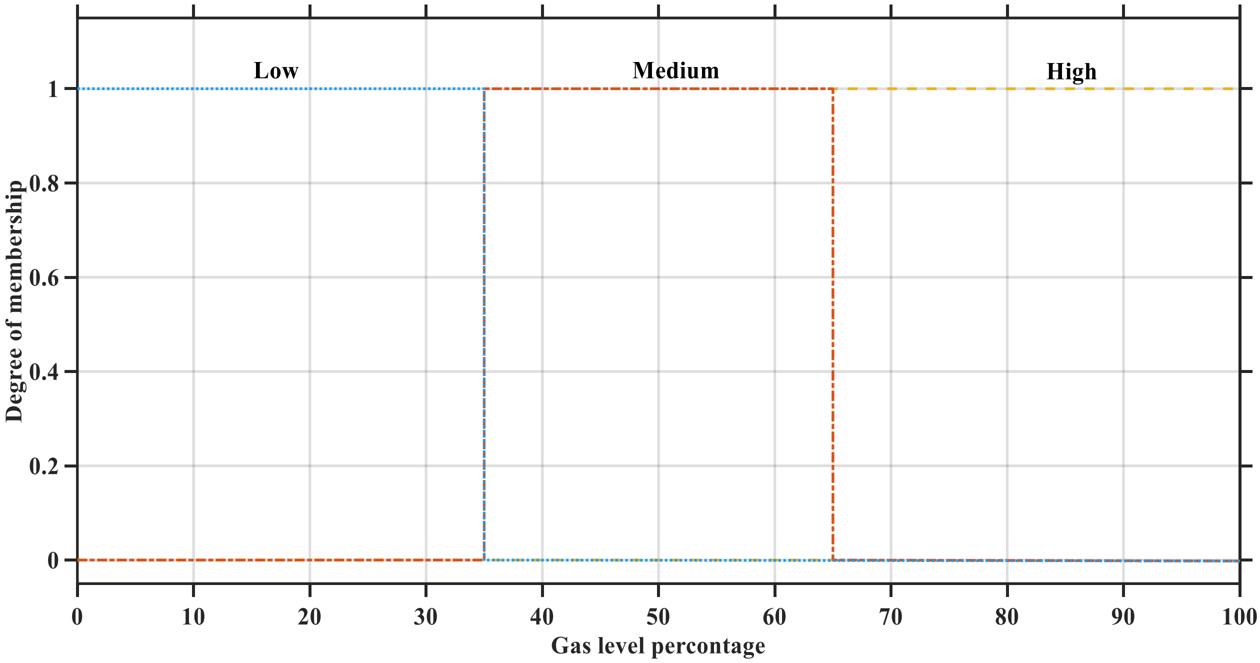


**Figure 3.1 Classical sets**

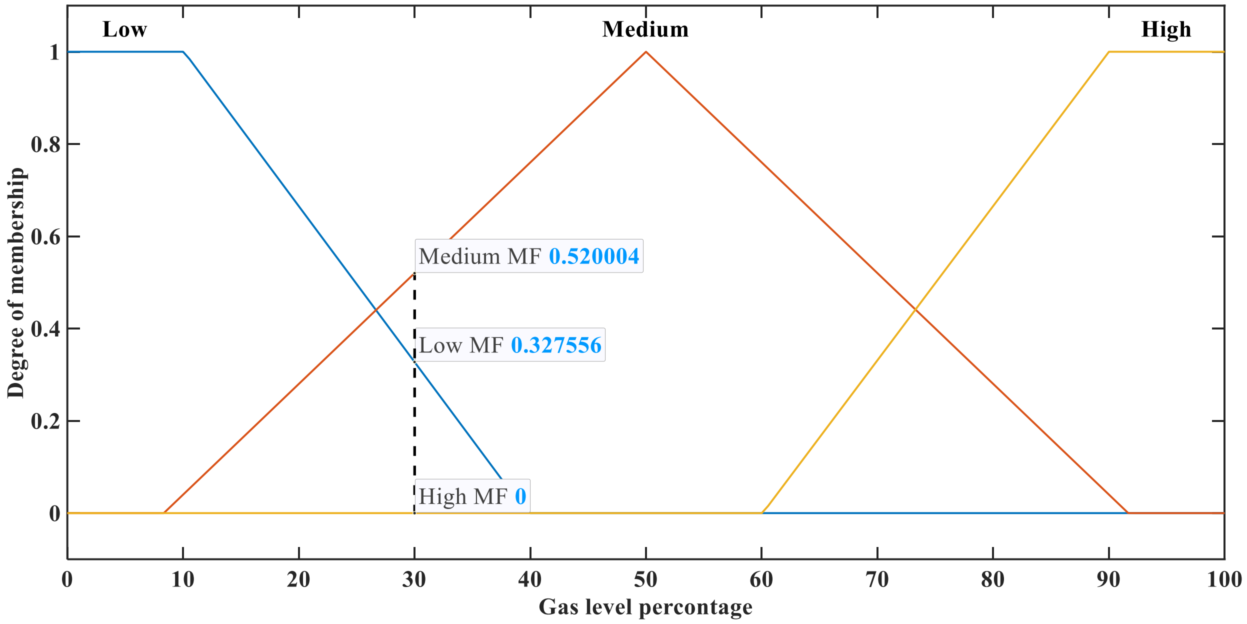


**Figure 3.2 Fuzzy sets and membership functions**

### 3.1.1 Fuzzy sets membership functions

A fuzzy input has combination of multiple fuzzy sets which also called MFS, these MFS can be parameterized to take a variety of shapes with various dimensions depending on the design and implementation of FLC [76]. In this particular research, we use MFS of triangular shape which can be represented by three parameters $(a, b, c)$, and trapezoidal shape which can be parametrized by four values $(a, b, c, d)$. Additionally, we can obtain the degree of membership of $x$ with these MFS using (3-4) for triangular shape, and (3-5) for trapezoidal shape:

$$\mu_{\text{triangular}}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases} \tag{3-4}$$

Where $a < b < c$.

$$\mu_{\text{trapezoidal}}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0, & d \le x \end{cases} \tag{3-5}$$

Where $a < b < c < d$.

The shape of triangular and trapezoidal MFS can be seen in Figure 3.3.

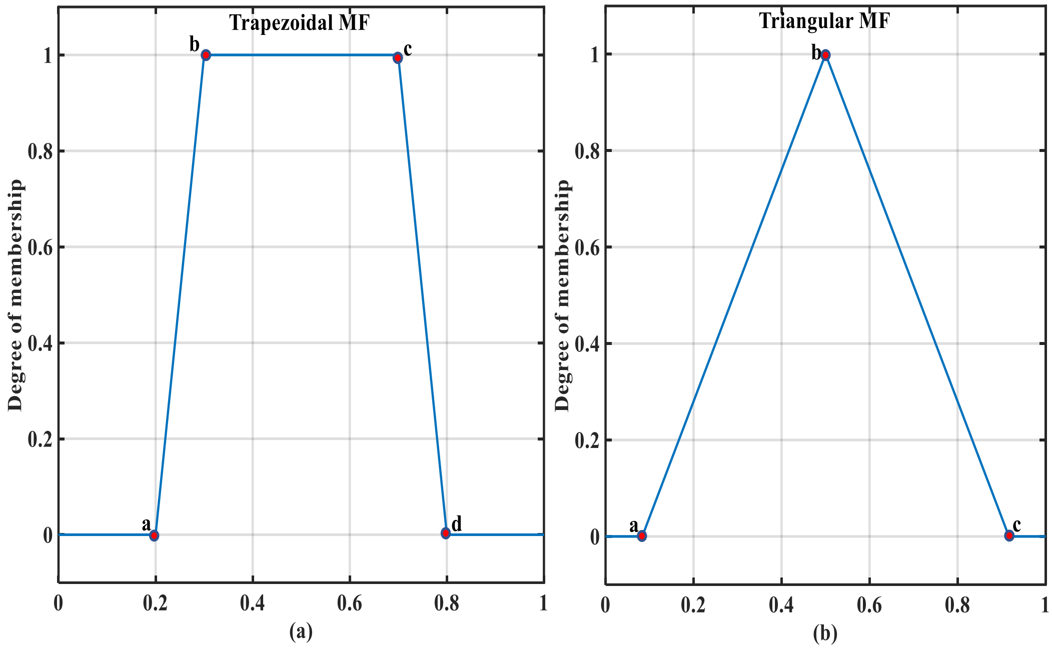


**Figure 3.3 trapezoidal and triangular MFS shapes**

### 3.1.2 Fuzzy rules and linguistic variables

Linguistic variables and values are concepts that were first proposed by professor ‘’L.Zadeh’’ in 1973 [77]. Linguistic variables were used to describe the inputs and outputs of FLC in linguistic notions or nouns. As we’ve seen in the previous section, the studied fuzzy sets was used to analyses the percentage of the level of gas which is a linguistic variable. The MFS of this fuzzy sets are linguistic values used to describe the linguistic variable numerical value in an adjective form. After the value of the linguistic variable is described by a linguistic value, an inference system that uses if-then rules is implemented to map an output degree of membership from the inputs degrees of membership. Fuzzy if-Then rules can be written as in (3-6) [78]:

$$if\ x\ is\ A\ then\ y\ is\ B \qquad (3\text{-}6)$$

Here $A$ and $B$ are fuzzy variables or MFS of input and output fuzzy sets respectively, the part ``$if\ x\ is\ A$`` is called the antecedent, whereas the part `` $then\ y\ is\ B$`` is called the consequent. Usually, in FLC we deal with multiple inputs and outputs, each input or output has its own fuzzy set as can be seen in (3-7). In the antecedent part, inputs linguistic values and variables are combined using fuzzy operators namely ‘’And/or’’ operators [79, 80].

$$if\ u_1\ is\ A_1\ and\ u_2\ is\ A_2\ \ldots..\ u_M\ is\ A_M\ then\ y_1\ is\ B_1 \ldots y_Z\ is\ B_Z \qquad (3\text{-}7)$$

Where $M$ is the number of inputs, and Z is the number of outputs. Each operator handles inputs differently. ‘’And’’ operator is the intersection $(t - norm)$ between inputs degrees of membership [81], whereas ‘’Or’’ operator is their union $(s - norm)$ [82]. Let’s consider a fuzzy rule $R^1$ with two inputs $u_1$ and $u_2$, and one output $y_1$. Where $u_1 \in X$, and $y_1 \in Y$. $u_1$ is characterised by the MF linguistic value $A_1$ and $u_2$ is characterized by the MF linguistic value $A_2$ where:

$$R^1 = if\ u_1\ is\ A_1\ and\ u_2\ is\ A_2\ then\ y_1\ is\ B_1 \tag{3-8}$$

The firing level $f^1$ of the rule $R^1$ can be calculated using $t-norm$ operator as in (3-9), and Figure 3.4:

$$f^1 = \mu_{A_1 \cap A_2}(u) = \min\left[\mu_{A_1}(u_1), \mu_{A_2}(u_2)\right] \tag{3-9}$$

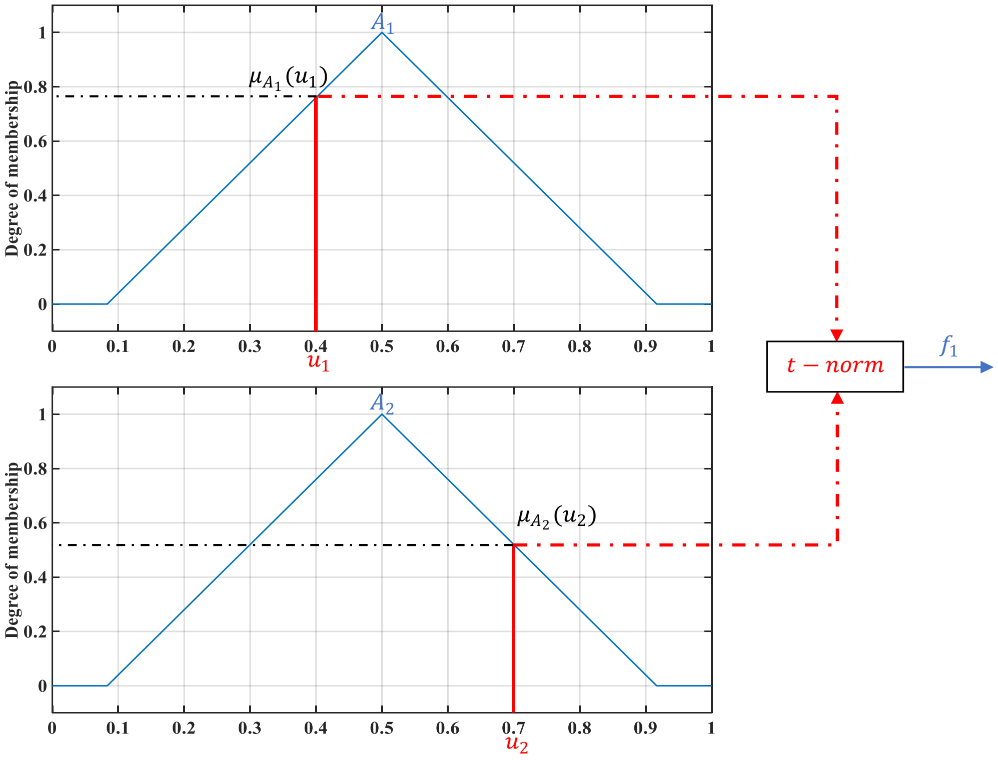


**Figure 3.4 Rule firing using $t-norm$**

When multiple rules with the same output linguistic value are valid at the same time, appliance of ''Max'' operator is needed [83], which usually scales all the identical output linguistic values degree of membership into a singular output with the highest degree of membership among them. For example, if we have three rules $R^1$, $R^2$, and $R^3$ with the same output MF linguistic value $B_1$, and with three firing levels higher than zero where $f^1$=0.35, $f^2 = 0.12$, and $f^3$=0.88. $B_1$ degree of membership can be determined using (3-10):

$$\mu_{B_1}(y) = \mu_{f^1 \cup f^2 \cup f^3}(y) = \max[f^1, f^2, f^3] = f^3 \tag{3-10}$$

### 3.1.3 Fuzzy Mamdani inference system

Overall, a FLC receives crip inputs and produces crisp outputs after going through multiple steps as can be seen in  Figure 3.5.

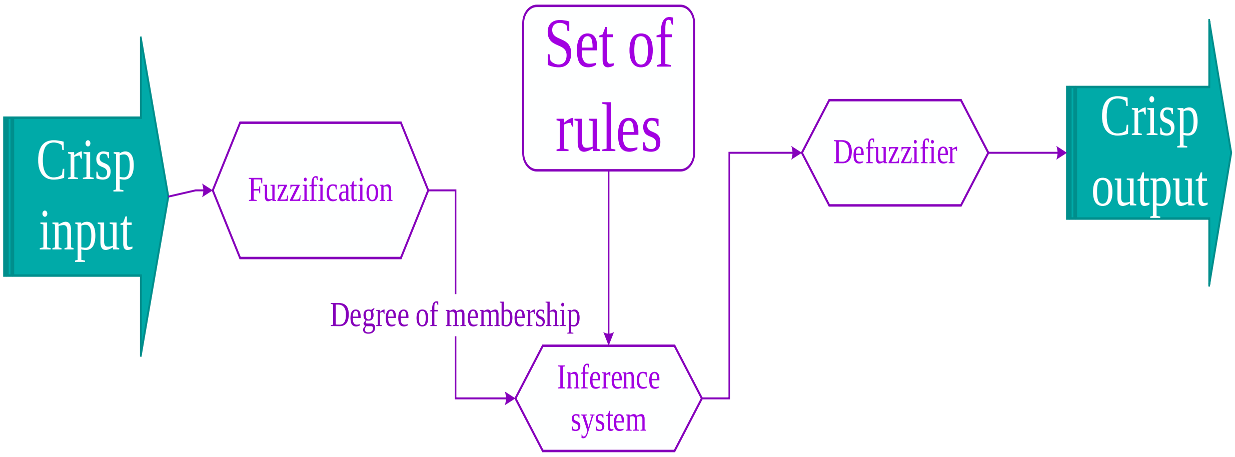


**Figure 3.5 Fuzzy logic controller steps**

Inference system is the engine that uses inputs degrees of membership and fuzzy set of rules to map out an output degree of membership that is later defuzzified and converted into a crisp output. This research uses one of the most common fuzzy inference systems which is Mamdani inference system [84]. This particular inference system is well suited for complex applications that are very well known allowing the rules set to be derived from human's experience [85].

### 3.1.4 Aggregation and defuzzification

After processing all the valid rules and output linguistic values with a firing level higher than zero, we can use them to map out an aggregated area or surface that we can defuzzify into a crisp output [86]. Let us consider two valid output linguistic values $B_1$, and $B_2$ with firing levels $f_1$, and $f_2$ respectively as we can see in Figure 3.6

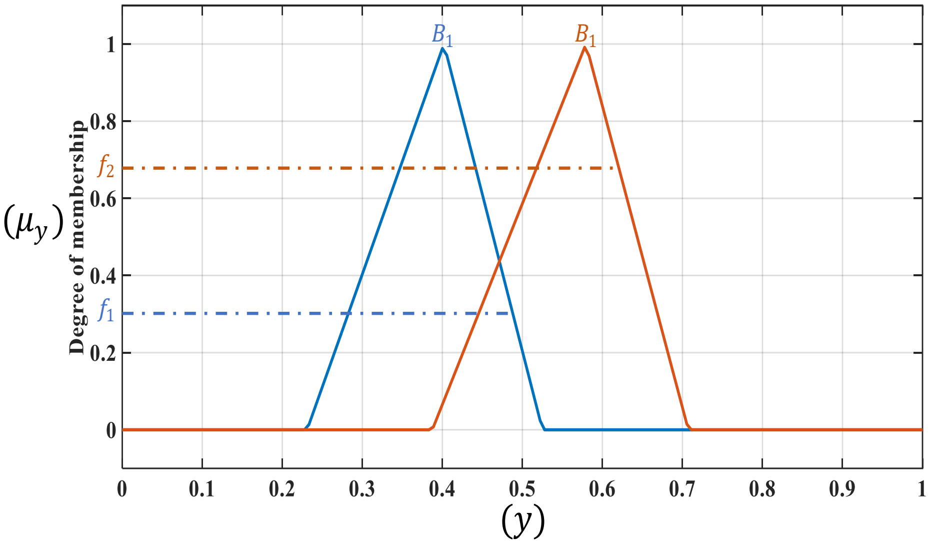


**Figure 3.6 Output MFS firing levels**

If we use Mamdani minimum implication operator [87, 88], we can obtain an aggregated area of $B_1$ and $B_2$ output MFS highlighted by grey color as in Figure 3.7.

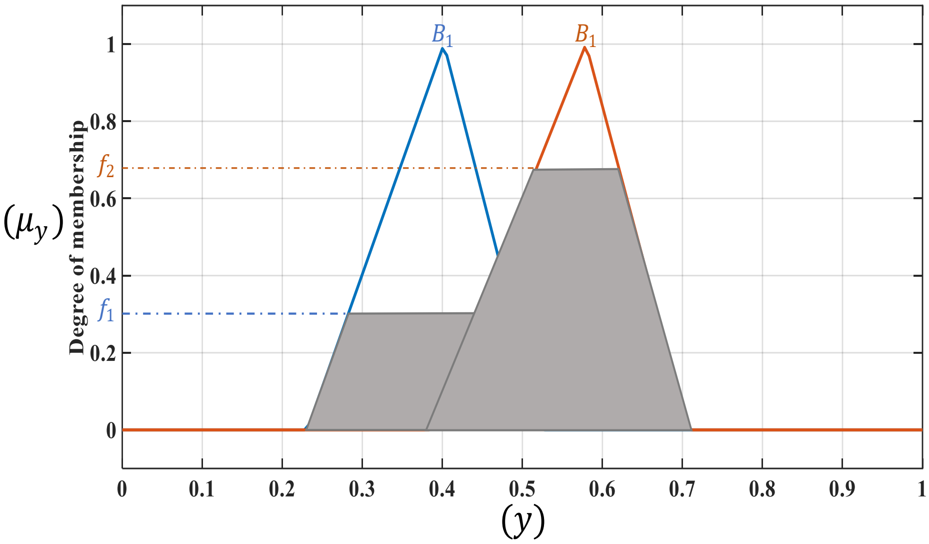


**Figure 3.7 Aggregation using Mamdani minimum operator**

After aggregation is complete, the obtained surface can then be defuzzified to produce a crisp output using defuzzification techniques [89, 90].

There is a wide range of defuzzification techniques, the most common one is centroid of area (COA) method due to its balanced simplicity and impact [91, 92]. COA uses the coordinates of several points of the aggregated area to find the center of it which is the FLC crisp output. This can be done as in Figure 3.8 using equation (3-11):

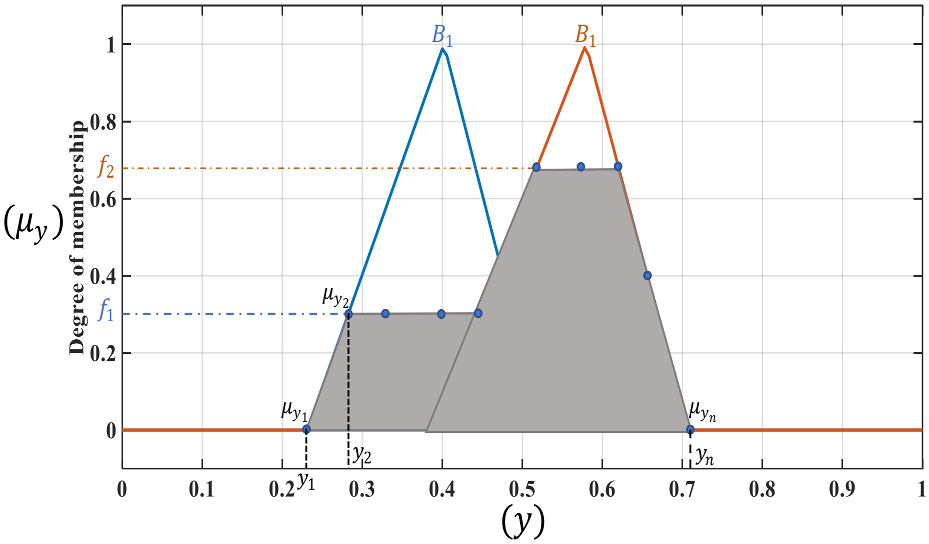


**Figure 3.8 Centroid of the area defuzzification method**

$$u_0 = \frac{\int_{y=y_1}^{y=y_n} \mu_y(y) y \, dy}{\int_{y=y_1}^{y=y_n} \mu_y(y) \, dy} \tag{3-11}$$

$Y$ is the output fuzzy set universe of discourse with $\forall y \in Y$. $n$ is number of the sampled values of the aggregated surface. $u_0$ is the FLC crisp output.

## 3.2 DESIGN OF AN SPS BASED FLC WITH OPTIMIZED MFS FOR SPEED CONTROL OF PMSM

In most cases, when a FLC is designed, its MFS are built based on experience and trial and error technique. However, recent studies indicates that a FLC can perform better when having its MFS tuned by metaheuristic algorithms (MHA). The fuzzy rule set can also be optimized by MHA. However, this does not offer a decent lift in performance than the implementation of user experience and trial and error methods.

The designed FLC has two inputs which are the speed tracking error and its derivative, and one output that is multiplied by a scaling factor $K_s$ to obtain the desired electromagnetic torque $T_{e(ref)}$. Speed tracking error was normalized using Saturation 1 to never surpass the interval $[-120, 120]$, whereas Saturation 2 was tuned with the rest of FLC parameters

Additionally, $K_s$ and Saturation 2 parameters along with MFS of the proposed F1MF were all tuned simultaneously using SPS algorithm. Whereas the $dq$ currents controllers were pre-tuned from an individual tuning process of a standard FLC with unoptimized MFS (FC1). This individual tuning was performed for the sake of comparing FC1 to F1MF and see how does the tuning of MFS impact the overall performance of the speed controller. Moreover, F1MF inputs were saturated. However, its output was not.

The structure of the F1MF in MATLAB/SIMULINK can be seen in Figure 3.9, whereas its fuzzy rule set can be seen in Table 3

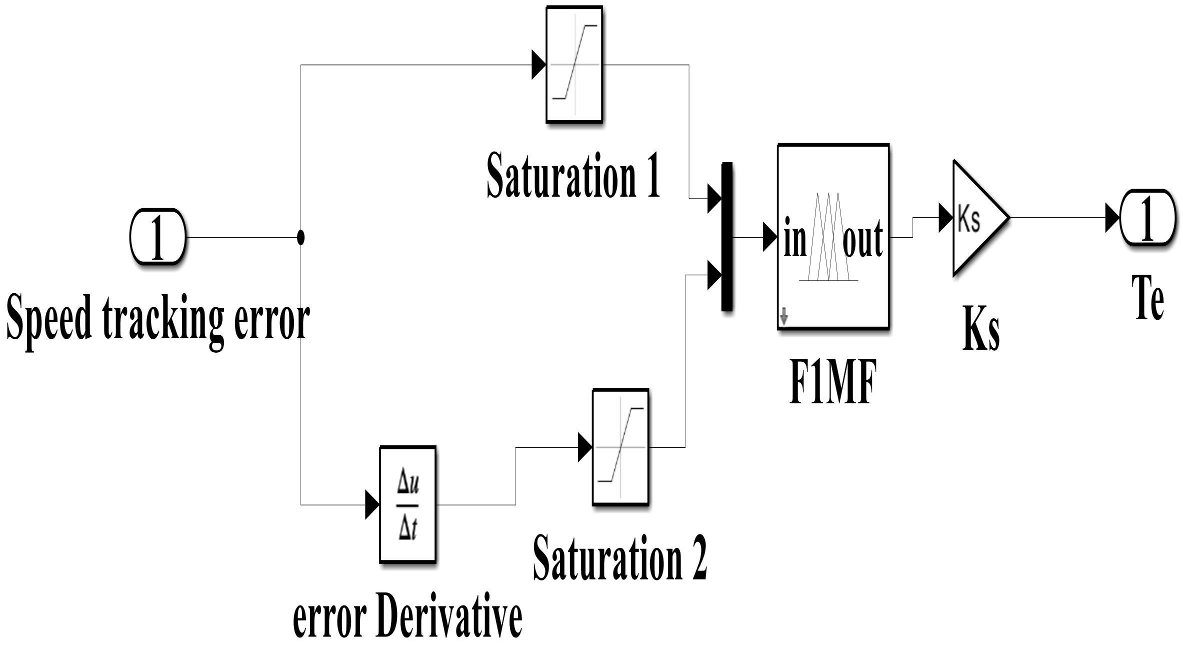


**Figure 3.9 F1MF MATLAB/SIMULIKN depiction**

**Table 3 fuzzy rule set**

| | | $\boldsymbol{Speed\ error}\ (\boldsymbol{e_{\omega r}})$ | | | | |
|---|---|---|---|---|---|---|
| | | $NB$ | $NS$ | $ZO$ | $PS$ | $PB$ |
| $\boldsymbol{Derivative\ of\ error}$<br>$(\boldsymbol{de_{\omega r}})$ | $NB$ | $NB$ | $NB$ | $NB$ | $NS$ | $ZO$ |
| | $NS$ | $NB$ | $NB$ | $NS$ | $ZO$ | $PS$ |
| | $ZO$ | $NB$ | $NS$ | $ZO$ | $PS$ | $PB$ |
| | $PS$ | $NS$ | $ZO$ | $PS$ | $PB$ | $PB$ |
| | $PB$ | $ZO$ | $PS$ | $PB$ | $PB$ | $PB$ |

Here (NB, NS, ZO, PS, and PB) are respectively acronyms of $(negative\ big, negative\ small, zero, positive\ small, positive\ big)$. Additionally, NS, ZO, and PS are triangular MFS. Whereas, PB and NB are trapezoidal MFS.

**REMARK:** during the tuning of F1MF, phase 2 of SPS was set to only update the output scaling factor of F1MF since it's the hardest parameter to tune.

F1MF was tuned using the same objective function (OBJ) in (2-49) that was used to tune SPS-PI. This tuning progress can be seen in Figure 3.10 along with F1MF tuning progress when using PSO, SOS, or STBO. As we can see in the zoom-in highlighted in Figure 3.10, SPS was the most successful algorithm followed by STBO, SOS, and then PSO respectively.

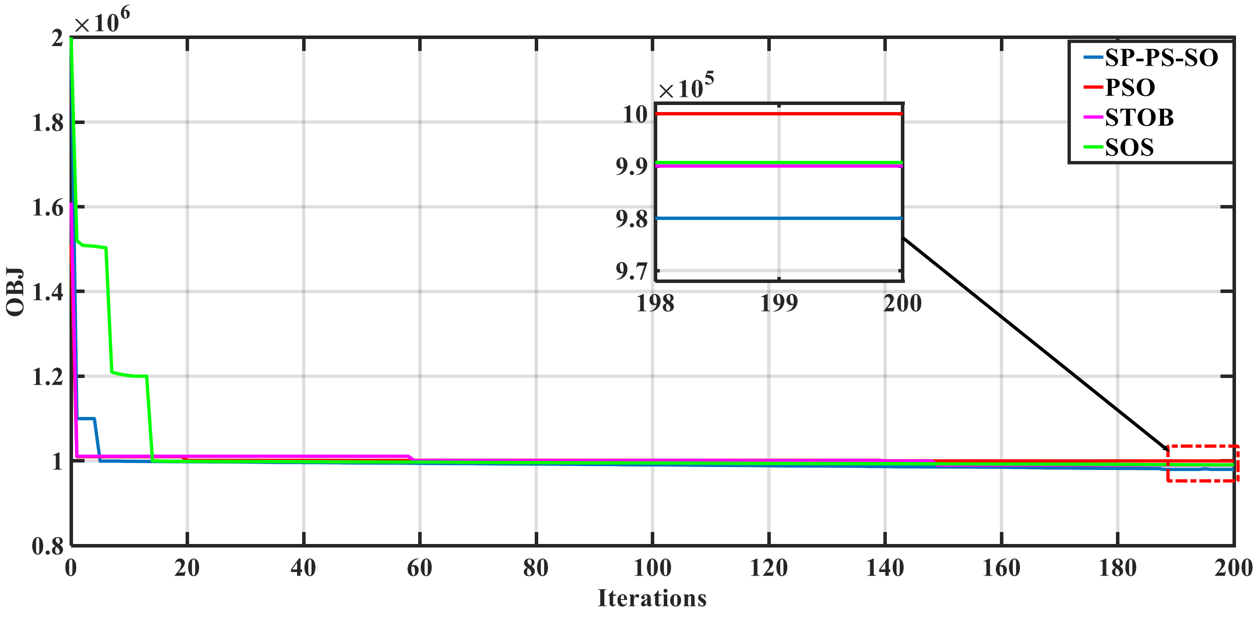


**Figure 3.10 Tuning progress of F1MF when using a variety of MHA**

F1MF optimized inputs and output fuzzy sets can be seen in Figure 3.11, Figure 3.12, and Figure 3.13 respectively.

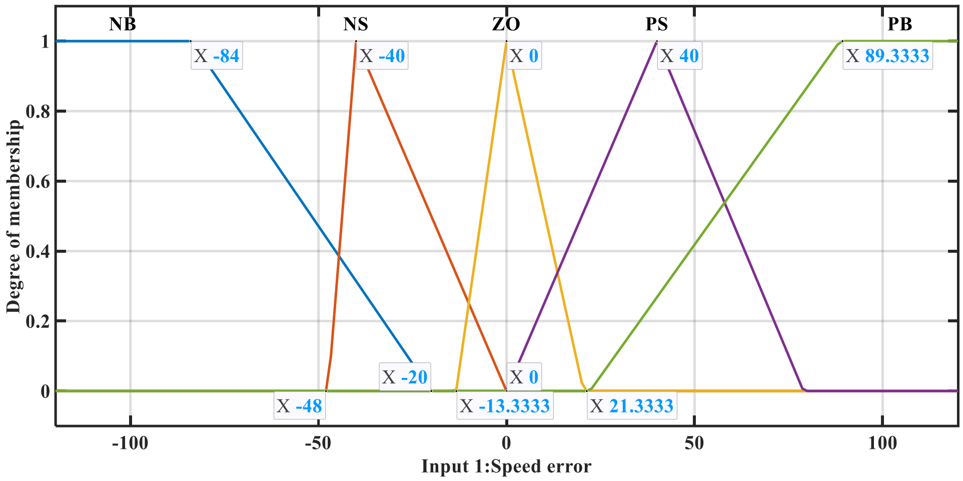


**Figure 3.11 Input 1 of F1MF**

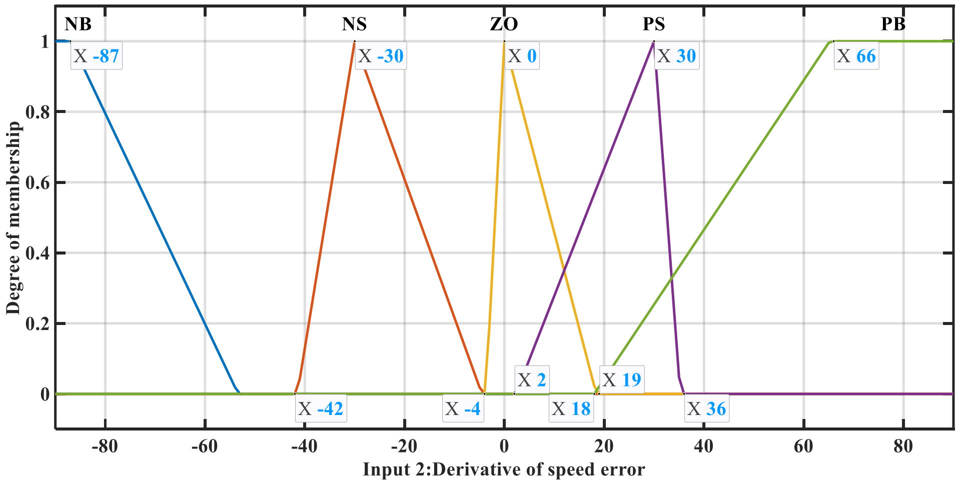


**Figure 3.12 Input 2 of F1MF**

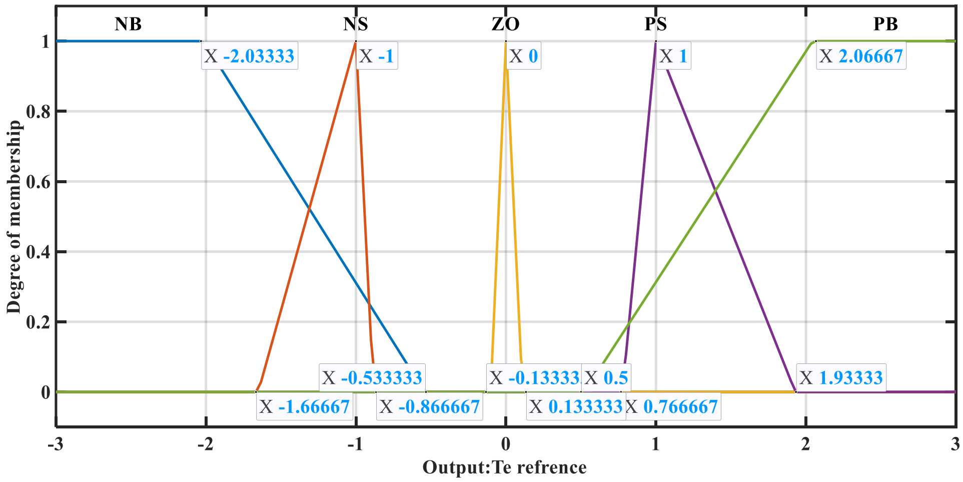


**Figure 3.13 Output of F1MF**

## 3.3 SIMULATED COMPARISON BETWEEN SPS-FLCMF, SPS-FLC, AND SPS-PI

This section presents a series of tests that compares the performance of F1MF, FC1, and SPS-PI when used for speed control of PMSM. The parameters of the fuzzy controllers can be seen in Table 4. SPS-PI parameters were already presented in Table 2.

**Table 4 comparison simulation controllers' parameters**

| | $Ks$ | $Kp_{iq}$ | $Ki_{iq}$ | $Kp_{id}$ | $Ki_{id}$ |
|---|---|---|---|---|---|
| $FC1$ | 84.3 | 61.96 | 143 | 79.5 | 54.46 |
| $F1MF$ | 23.66 | 61.96 | 143 | 79.5 | 54.46 |

### 3.3.1 Test I: Variable speed reference with an introduction of external disturbances

This test investigates the speed tracking performance and the produced currents and electromagnetic torque of the analyzed three controllers. A variety of speed setpoint references (SSR) changes were applied throughout 2 seconds of simulation. A load charge of 5 N.m was applied at 1.5s.

The performance of speed tracking of Test I can be seen in Figure 3.14. The changes of SSR are highlighted and zoomed-in in Figure 3.15Figure 3.14 where we can clearly see that F1MF is the most successful at suppressing the overshoot of SSR tracking in addition to its fast response and small settling time. FC1 outperformed SPS-PI in the aspect of handling and minimizing the overshoot resulted from the sudden changes in speed reference. However, it still couldn't match F1MF level of efficiency especially in terms of steady-state-error (SSE) reduction. At 1.5s, a load charge was added as we can see in Figure 3.15. This load caused a significant steady-state-error (SSE) in the speed tracking performance of FC1 and SPS-PI. F1MF however, managed to maintain a smaller SSE than both.

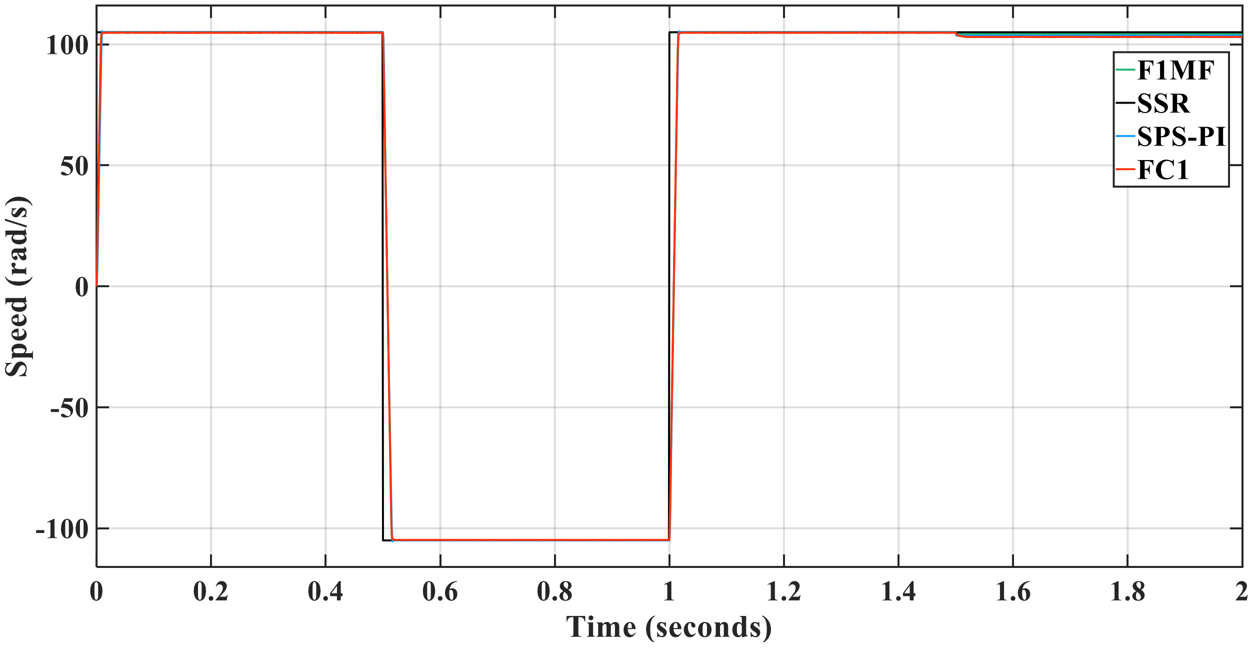


**Figure 3.14 Speed tracking performance in Test I**

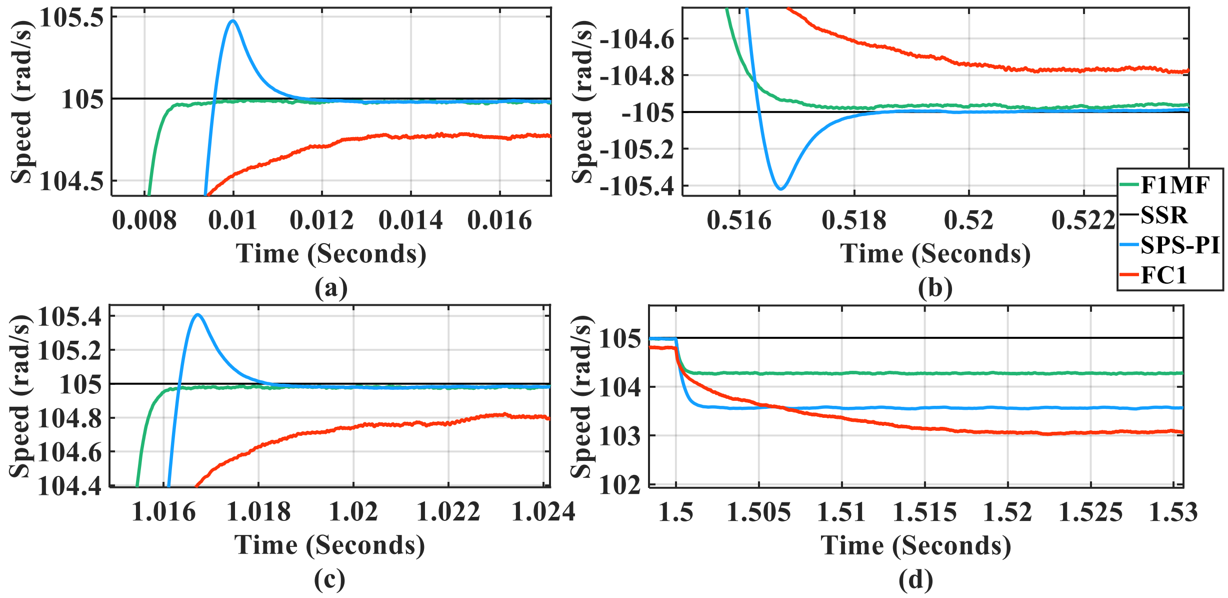


**Figure 3.15 Zoom-in of Figure 3.14.**

The produced electromagnetic torque $T_e$ of each controller can be seen in Figure 3.16. and upon a thorough analysis of the latter, we can confidently say that in the case of F1MF and FC1, the spikes of $T_e$ changes were less and more suppressed than that of SPS-PI. However, the latter torque ripples were significantly less than FC1 produced torque ripples, and slightly less than F1MF which showcases how can the speed controller affect the whole operation of FOC of PMSM. And also proves the importance of optimizing the MFS of FLC.

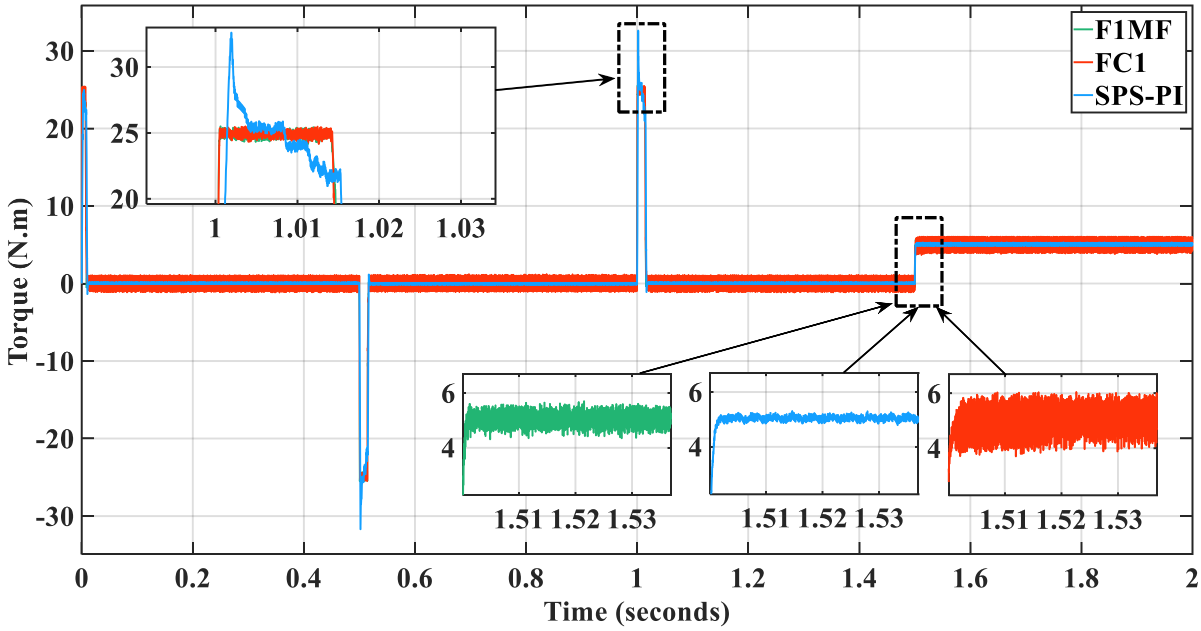


**Figure 3.16 Produced $T_e$ in Test I.**

Induced phase current $i_a$ can be seen in Figure 3.17. At steady state, during the presence of 5 N.m load charge. SPS-PI current experienced fewer total harmonics distortion (THD) presence than both F1MF and FC1 where it only reached 3.34%. F1MF THD was around 6.08%, and finally

FC1 with 10.11%. Nevertheless, F1MF speed tracking SSE was less than SPS-PI which makes it the preferable choice despite experienced higher THD than the letter.

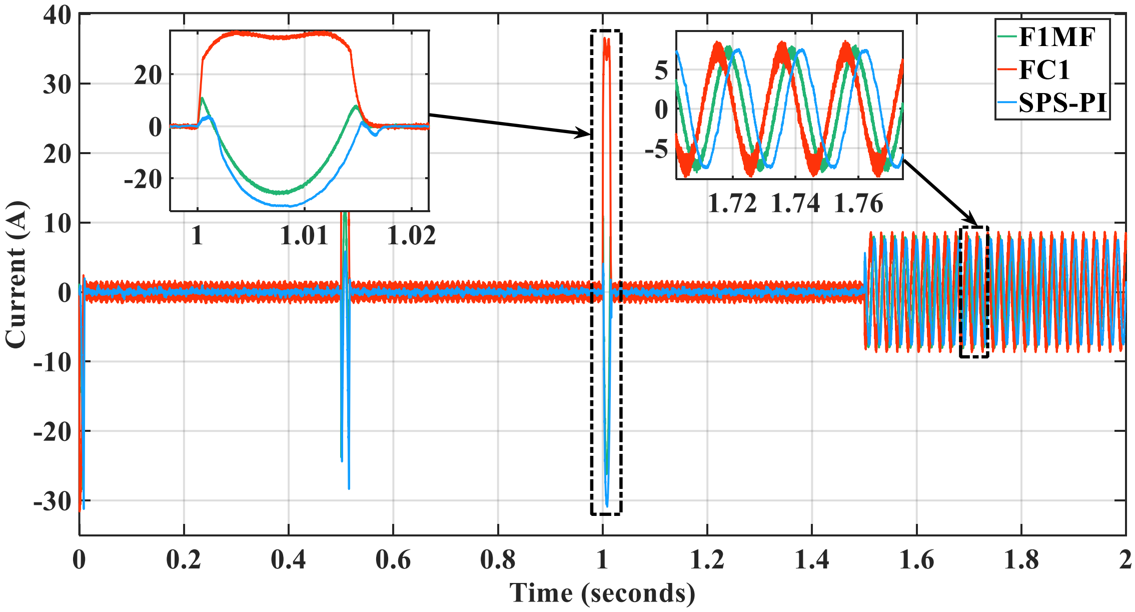


**Figure 3.17 Produced phase current $i_a$ in Test I.**

### 3.3.2 Test II: Fixed speed reference with different motor parameters

Conventional PI controller is known for its small bandwidth especially in the speed control outer loop in FOC of PMSM. Which makes controller retuning inevitable in most cases. However, a well-designed FLC does not need to be retuned and can perform well without the need for further adjustments. Therefore, the goal of this test is to investigate how F1MF, FC1, and SPS-PI can handle a significant change in PMSM crucial parameters which are the electrical parameters known as $dq$ axis inductances, and the phase resistance $R$.

This test applies a significant increase in the values of the electrical parameters throughout a 1 second of simulation. The targeted speed is fixed to 100 $rad/sec$, and the load charge is also a constant 0.5 $N.m$. The values of $L_d$, $L_q$, and $R$ are constant $0.013H$, $0.022H$, and $3.5\Omega$ respectively.

The speed tracking curves of Test II can be seen in Figure 3.18. It appears from these results that F1MF is the only controller capable of handling the crucial variation in the motor internal parameters by delivering the targeted speed with no significant SSE, whereas both FC1 and SPS-PI malfunctioned and failed to produce any reliable mechanical speed.

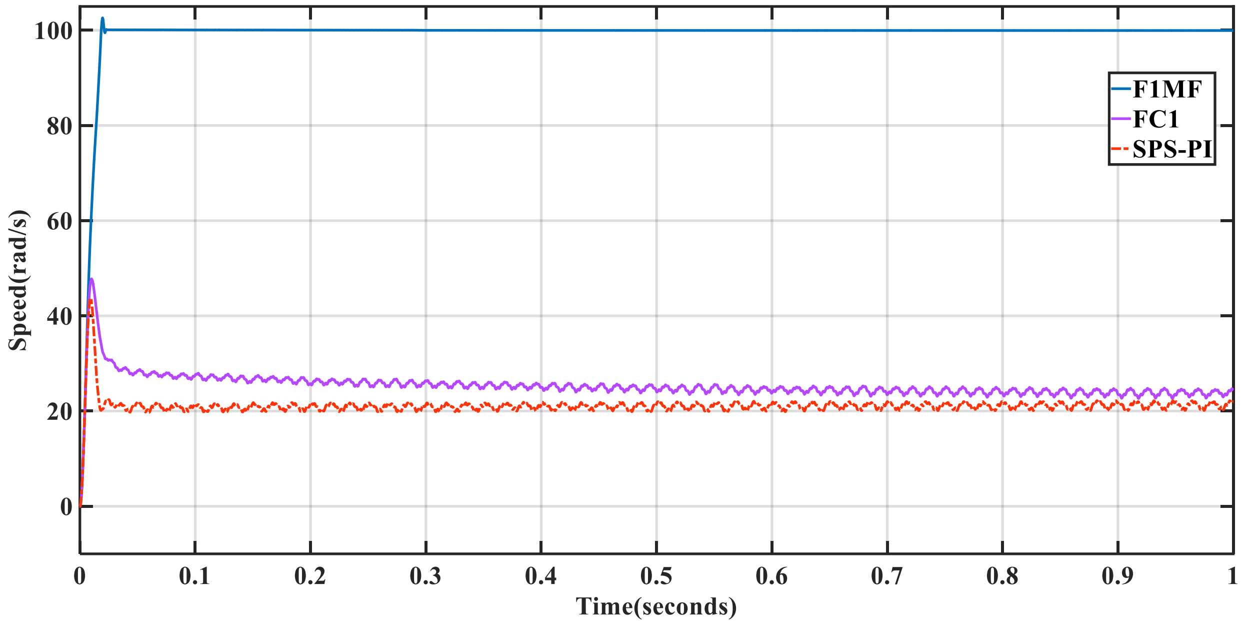


**Figure 3.18 Speed tracking performance in Test II**

In Figure 3.19, results of the induced $i_d$ current affirms that F1MF speed controller is way more rigid than both FC1 and SPS-PI controllers, since it's the only one that managed to regulate and maintain a functioning decoupling of $dq$ voltages, by successfully handling the introduced parametric changes, which led to an easier and more optimal regulation of $i_d$.

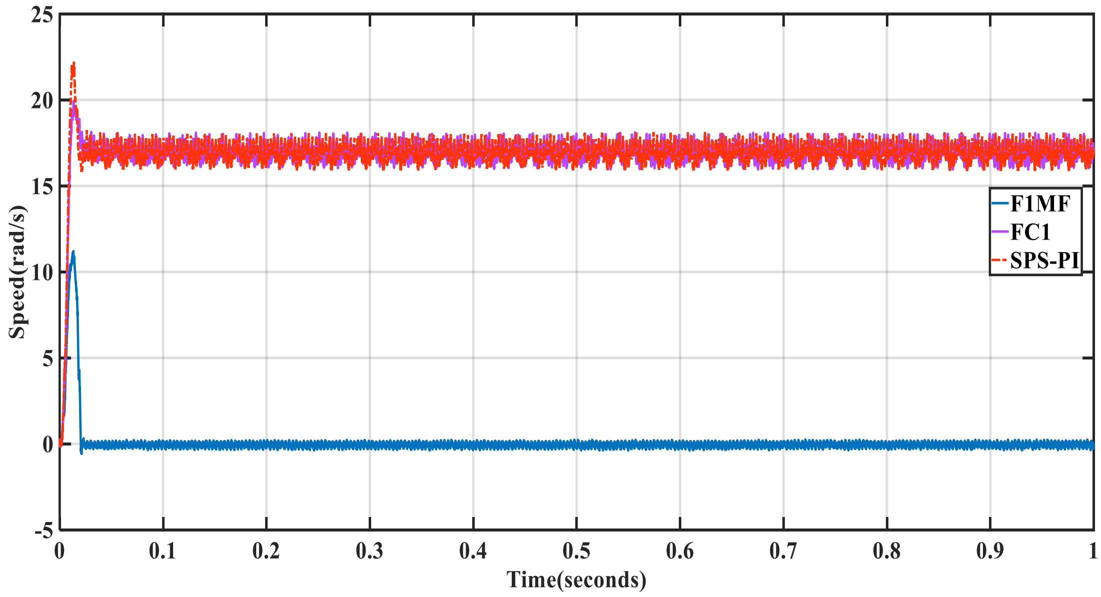


**Figure 3.19 Produced Id in Test II**

The illustrations of the produced $T_e$ in Figure 3.20 further confirms the superior robustness and higher tolerance of F1MF compared to the conventional FC1, and the classical SPS-PI. F1MF was the only controller capable of producing a reliable torque that matches the load charge and doesn't oscillate or significantly surpass the targeted $T_{e(ref)}$.

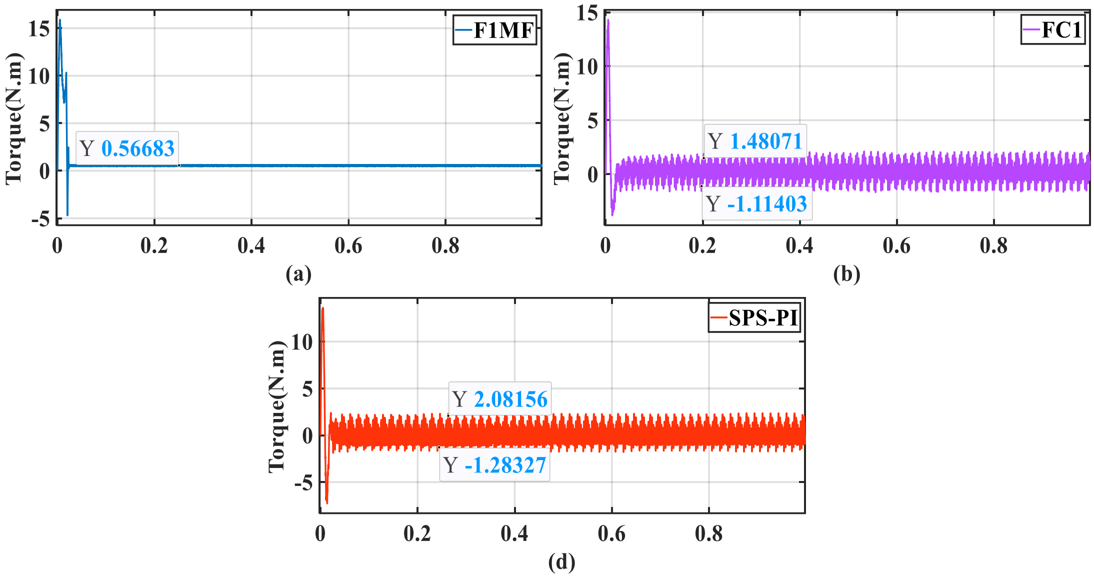


**Figure 3.20 Produced $T_e$ in Test II**

## 3.4 CONCLUSION

This whole chapter was dedicated to explaining the detailed concepts of fuzzy logic, as well as the implementation of FLC for speed control of PMSM. Moreover, SPS was used to tune the parameters of a standard FLC with unoptimized MFS, and a novel FLC with tunable MFS denoted by F1MF. Furthermore, two tests of FOC of PMSM were run in MATLAB/SIMULINK environment to compare the performance of the proposed F1MF to the conventional FC1 and SPS-PI controllers. Final results affirmed the significance of the increase in speed tracking performance after tuning the MFS of FLC. F1MF overshoot suppression was unmatched, however, despite the relatively small value of added external disturbances, it still failed to achieve a satisfactory minimization of speed tracking SSE. This problematic, SSE however, can still be dealt with using an extended interval Type-2 FLC which we will be showcased in the next chapter.

# Chapter 4: Type-2 FLC with an adaptive fractional derivative

Although the standard type-1 FLC with optimized membership functions (MFS) performs significantly better than the classical SPS-PI and the conventional FC1 with unoptimized MFS, it still falls short at reducing speed set-point tracking steady-state-error (SSE) when controlling systems that are exposed to high values of internal and external disturbance. One way to address this is using an advanced variant of the latter known as interval Type-2 FLC (FLC2) [93]. FLC2 is known for its superior tolerance to all types of disturbances. However, its transient response sometimes fails to match the required benchmarks. Thus, in this chapter, the concept of Type-2 FLC is explained, and a newly proposed FLC2 with optimized MFS (F2MF) is paired with an also newly proposed adaptive fractional derivative low pass filter combination (ALD). The constructed controller is denoted by (F2MF-ALD). This novel combination assures a successful SSE minimization, and a precise and very fast transient response at the same time. This was confirmed by comparing F2MF-ALD to all its simpler variants when utilized for speed control of PMSM. At the end, a processor in the loop (PIL) co-simulation was performed to demonstrate the applicability of F2MF-ADL.

## 4.1 TYPE-2 FLC

FLC2 is an advanced FLC known for its higher tolerance and capability of performing better than the conventional Type-1 FLC [94]. FLC2 superior tolerance comes from its extended form of fuzzy sets that allows it to adapt and function better than FLC in the presence of high disturbances whether it being interior or exterior [95]. Additionally, FLC2 whole process contains an extra step called ‘’Type reduction’’ (TR). This step reduces the output fuzzy set type to the conventional Type-1 fuzzy set using type reduction algorithms. TR is fundamental since defuzzification cannot be performed on Type-2 fuzzy sets.

A Type-2 fuzzy set can be seen in Figure 4.1. What’s special about this type of fuzzy sets is that each linguistic value has an extra MF called lower bound MF (LMF). Upper bound MF (UMF) is basically the classic MF of Type-1 FLC. In this case, any input $x$ from the universe of discourse $X$ receives two degrees of membership being LMF $x$ corresponding degree of membership ($\mu_L$), and UMF $x$ corresponding degree of membership ($\mu_U$). Both $\mu_L$ and $\mu_U$ range from 0 to 1 depending on their MFS shape and parameterization. The grey area between UMF and LMF is called the footprint of uncertainty (FUC) and it allows FLC2 to adapt to noises and parameters deviations in the controlled system model. The height of UMF is always 1, whereas the height of

LMF is a tunable parameter denoted by HLB and it ranges from 0 to 1. The lower HLB is the bigger the size of FUC. FLC2 extended steps can be seen in Figure 4.2.

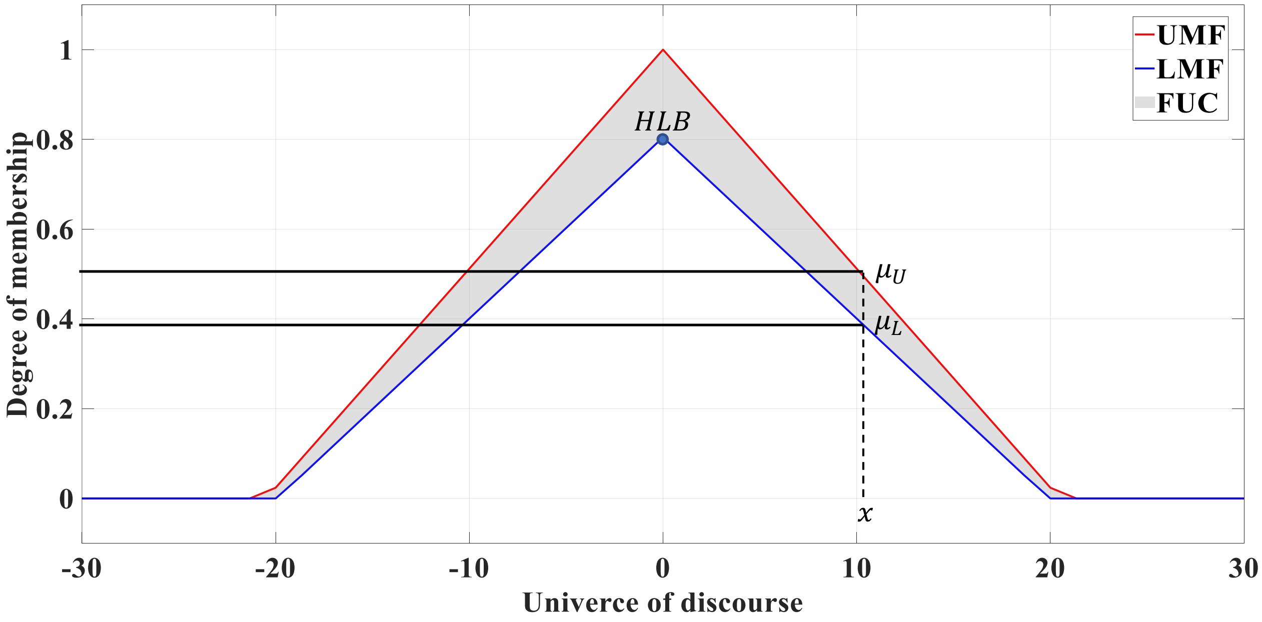


**Figure 4.1 Interval Type-2 fuzzy set**

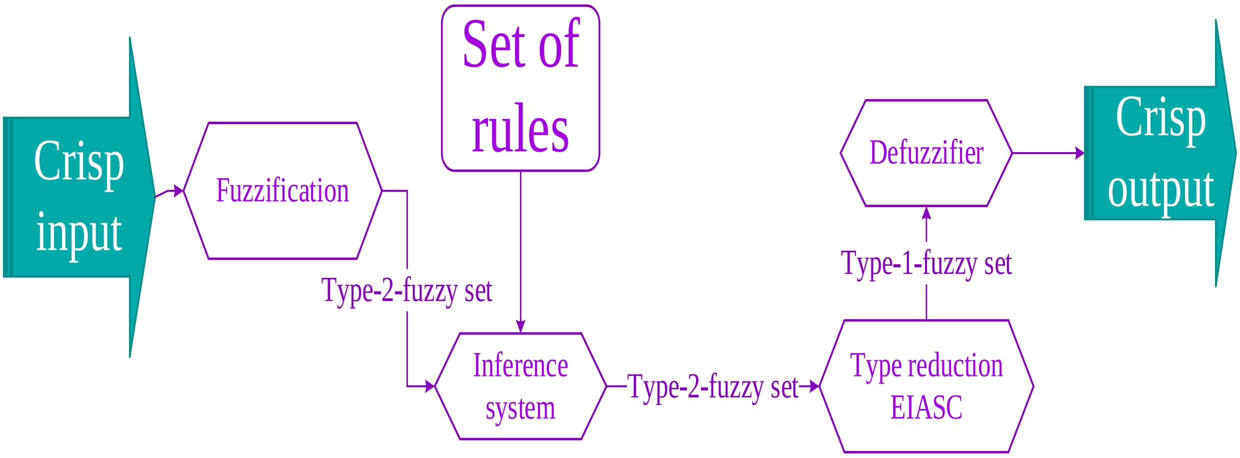


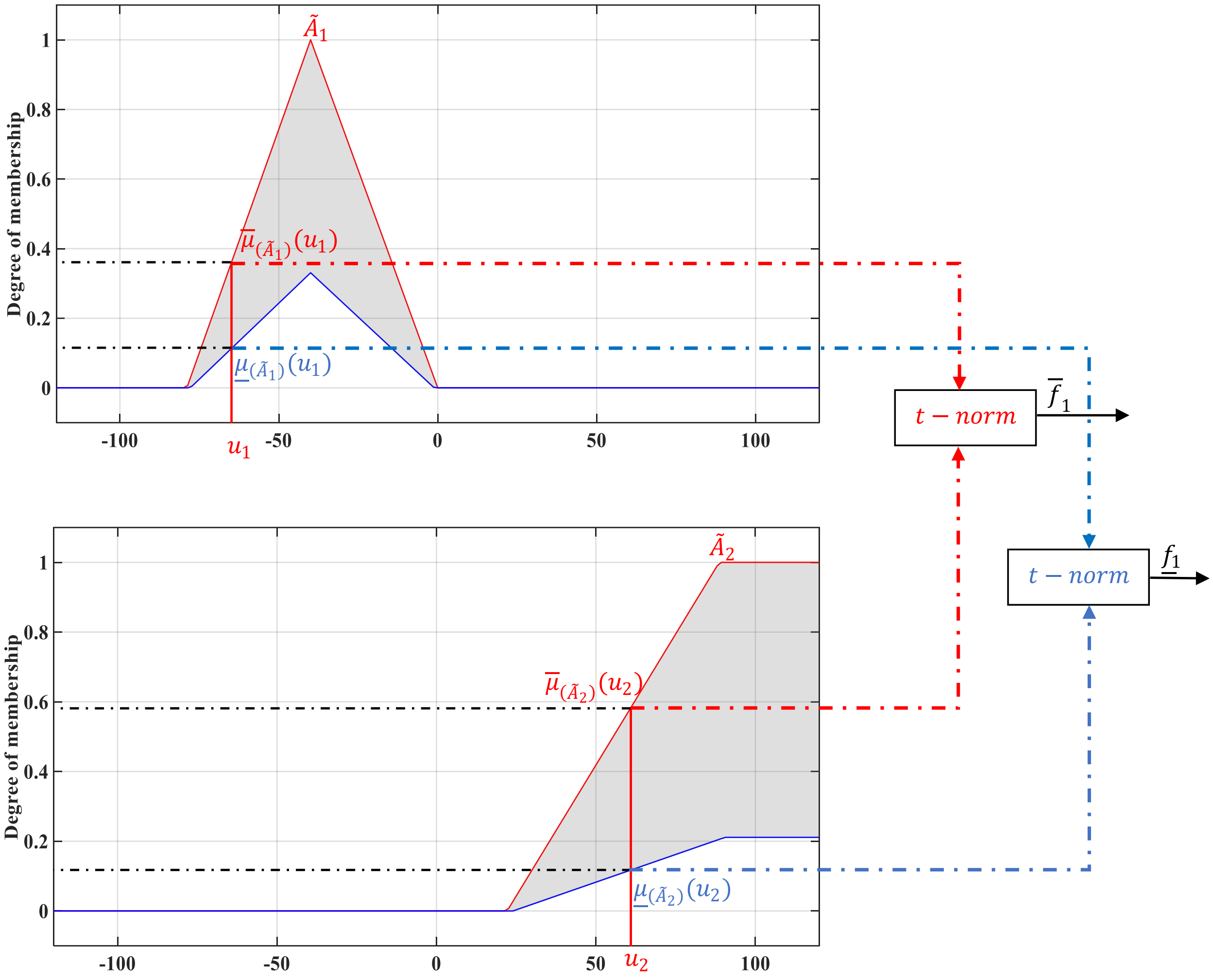
**Figure 4.2 Interval Type-2 fuzzy logic controller steps**

### 4.1.1 FLC2 inference engine

The most notable difference between FLC1 and FLC2 steps is the processes of the inference engine. With double the MFS of FLC1, FLC2 MAMDANI inference engine produces double the number of rules firing levels [96]. A similar example to what was presented in the previous chapter, consider a FLC2 with two inputs and one output, if we have a valid rule $\tilde{R}^1$ with two inputs $u_1$ and $u_2$, we can fuzzify these inputs and then determine the upper and lower firing levels of $\tilde{R}^1$ from equation (4-1) [75]:

$$\tilde{R}^1 = if\ u_1\ is\ \tilde{A}_1\ and\ u_2\ is\ \tilde{A}_2\ then\ y_1\ is\ \tilde{B}_1 \qquad (4\text{-}1)$$

Where $\tilde{A}_1$, $\tilde{A}_2$, and $\tilde{B}_1$ are MFS of interval Type-2 fuzzy sets. $\overline{f}_1$ and $\underline{f}_1$ are upper and lower firing levels of $\tilde{R}^1$ and they can be obtained using the intersection or the minimum operator denoted by $t - norm$. The full rule implication and firing level determination can be carried on as in Figure 4.3.



$\tilde{A}_1$
$\overline{\mu}_{(\tilde{A}_1)}(u_1)$
$\underline{\mu}_{(\tilde{A}_1)}(u_1)$
$u_1$
Degree of membership
1
0.8
0.6
0.4
0.2
0
-100
-50
0
50
100
$t - norm$
$\overline{f}_1$
$t - norm$
$\underline{f}_1$
$\tilde{A}_2$
$\overline{\mu}_{(\tilde{A}_2)}(u_2)$
$\underline{\mu}_{(\tilde{A}_2)}(u_2)$
$u_2$

**Figure 4.3 Rule firing using $t - norm$ in FLC2**

### 4.1.2 FLC2 aggregation process

FLC2 aggregation is somewhat similar to FLC1 aggregation process. After the fuzzification of inputs and the determination of firing levels of all valid rules is complete, we can map out an output aggregated area with upper and lower bounds determined by the upper and lower bounds of the valid rules firing levels. For example, consider a FLC2 with two valid rules $\tilde{R}^1$ and $\tilde{R}^2$, with the firing levels [ $\overline{f}_1, \underline{f}_1$ ] and [ $\overline{f}_2, \underline{f}_2$ ] respectively. The active outputs MFS of $\tilde{R}^1$ and $\tilde{R}^2$ are $\tilde{B}_1$ and $\tilde{B}_2$ respectively. Based on each rule firing levels, the aggregated area of their outputs MFS can be mapped out as in Figure 4.4, and Figure 4.5 [97].

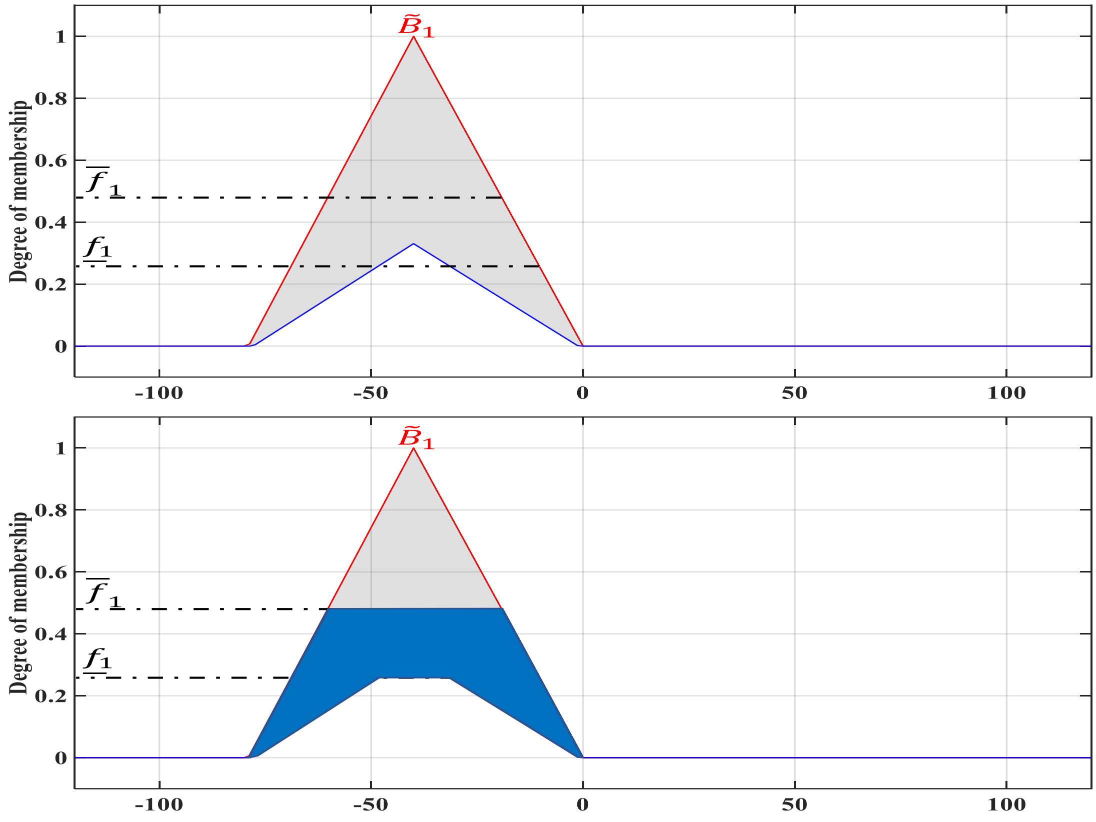


**Figure 4.4 Aggregation of $\tilde{B}_1$**

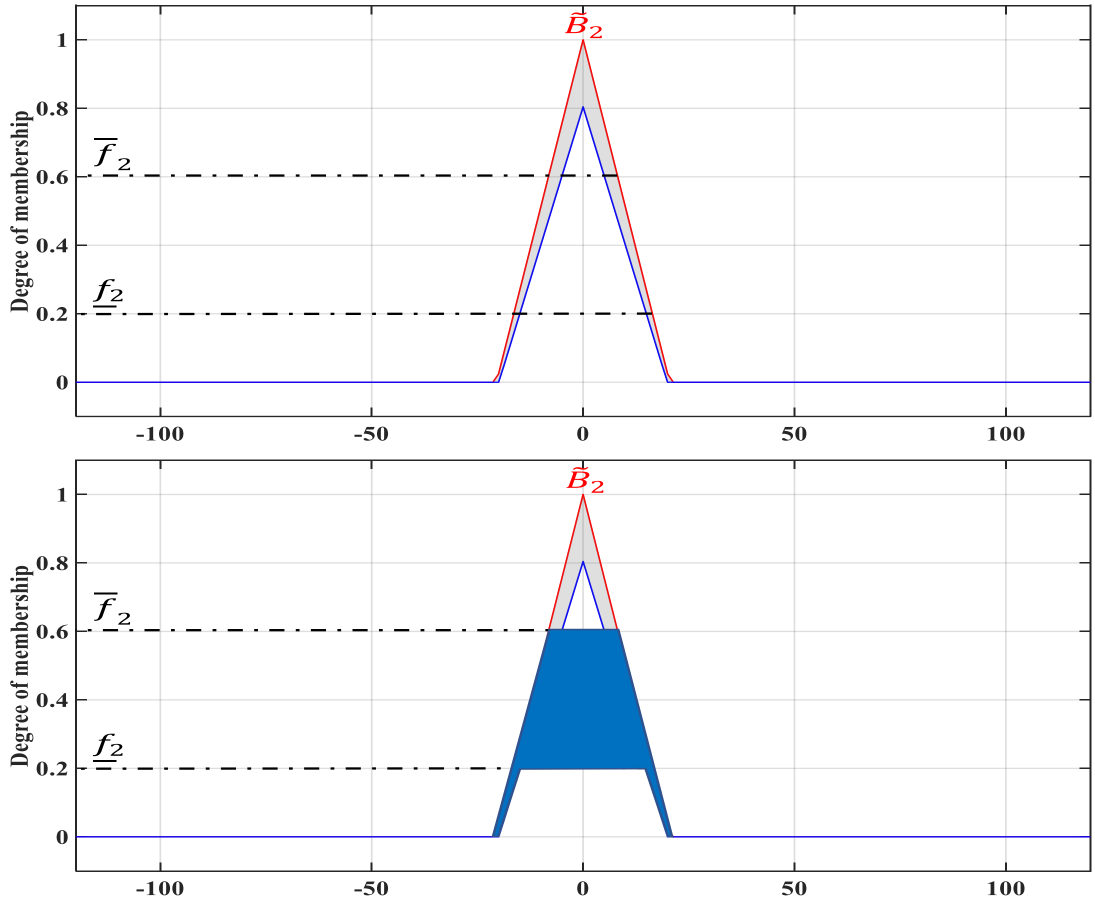


**Figure 4.5 Aggregation of $\tilde{B}_2$**

Finally, an output aggregated area can be obtained from the two outputs $\tilde{B}_1$ and $\tilde{B}_2$ as we can see in Figure 4.6.

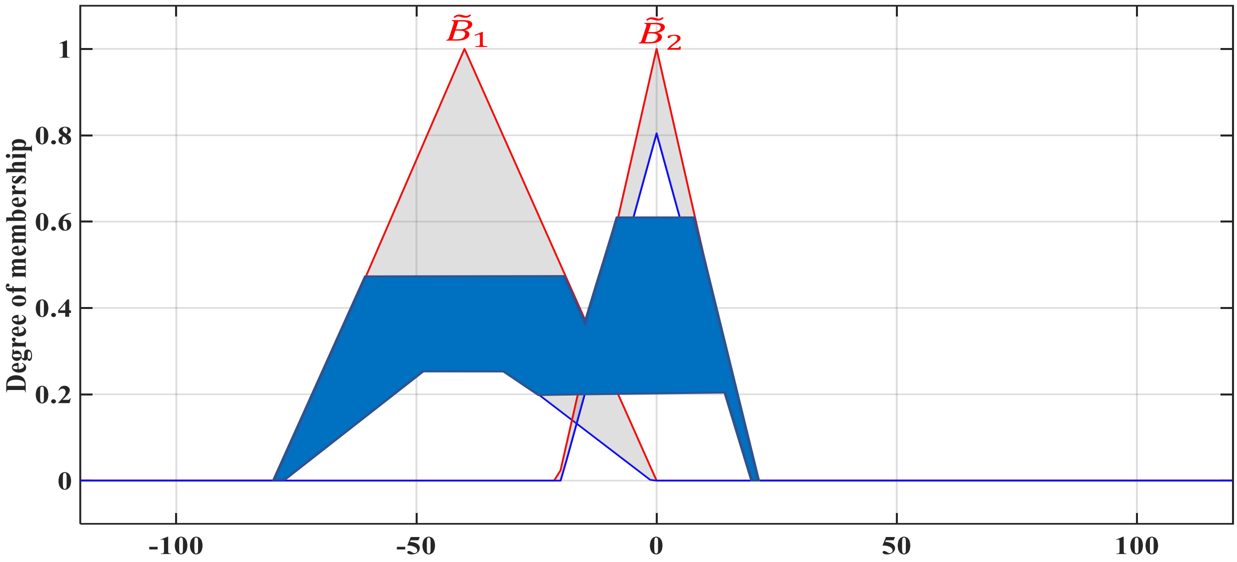


**Figure 4.6 Output aggregated area**

### 4.1.3 FLC2 output Type-reduction

Type reduction algorithm (TRA) is an extension of defuzzification process. Its aim is to reduce the type of the aggregated area so that a crisp output can be obtained from it. There are a wide range of TR algorithms, the most implemented one is Karnik and Mendel (KM) [96, 98]. KM design is based on famous FLC1 defuzzification algorithms such as COA and height and modified height algorithm. COA is the most common since it doesn't require an extensive amount of calculations and performs better than height and modified height algorithms in the presence of high number of rules [96]. Similar to COA defuzzification algorithm, COA based KM starts by sampling $K$ discrete points from $Y$ universe of discourse of the aggregated interval Type-2 fuzzy sets as we can see in Figure 4.7.

With $K \in Y$, each point from $[1, K]$ has an upper and lower bound degree of membership. It is also assumed that $y_1 < y_2 < y_3 \dots \dots . < y_K$. Moreover, before TRA is carried on. We need to obtain two Type-1 fuzzy sets with centroids that provides the best approximation of interval Type-2 fuzzy set centroid. In order to do that, the sampled upper and lower bounds points are used to obtain optimal switching points $[L, R]$, as illustrated in Figure 4.7.

Both $L$ and $R$ satisfy $L \in [1, K]$ and $R \in [1, K]$ respectively. The centroid of the aggregated area is an interval bound denoted by $[\Upsilon_l, \Upsilon_r]$. Additionally, $L$ is $\Upsilon_l$ discrete switching point which changes the course from the upper bound to the lower bound of the aggregated area, whereas $R$ is $\Upsilon_r$ switching point that changes the course from the lower bound to the upper bound. More details can be seen in Figure 4.8 and Figure 4.9.

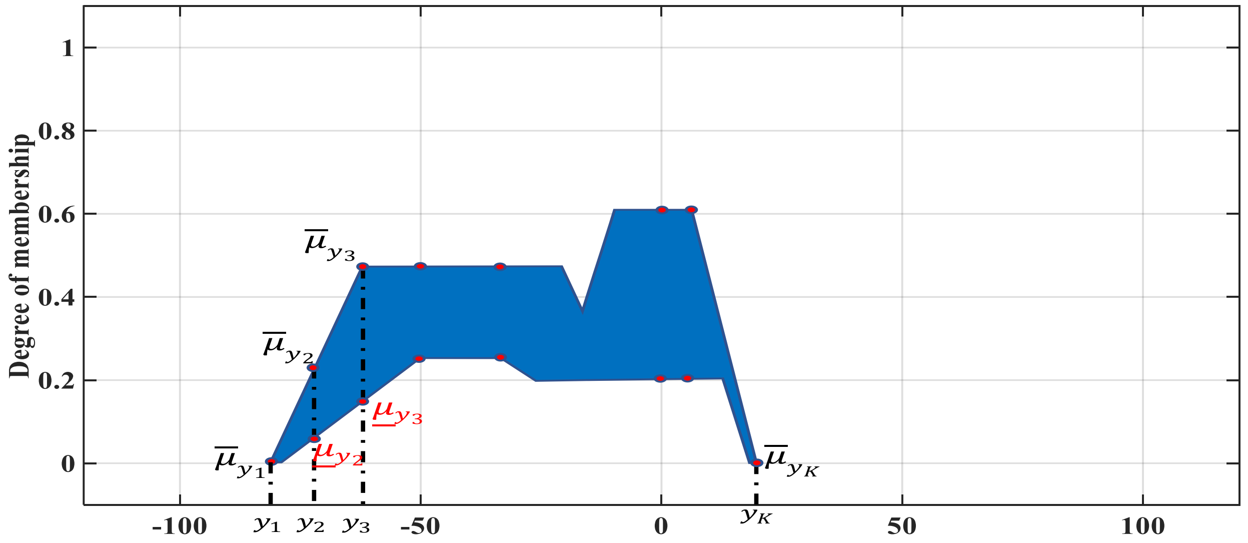


**Figure 4.7 Sampling of the aggregated Type-2 fuzzy set**

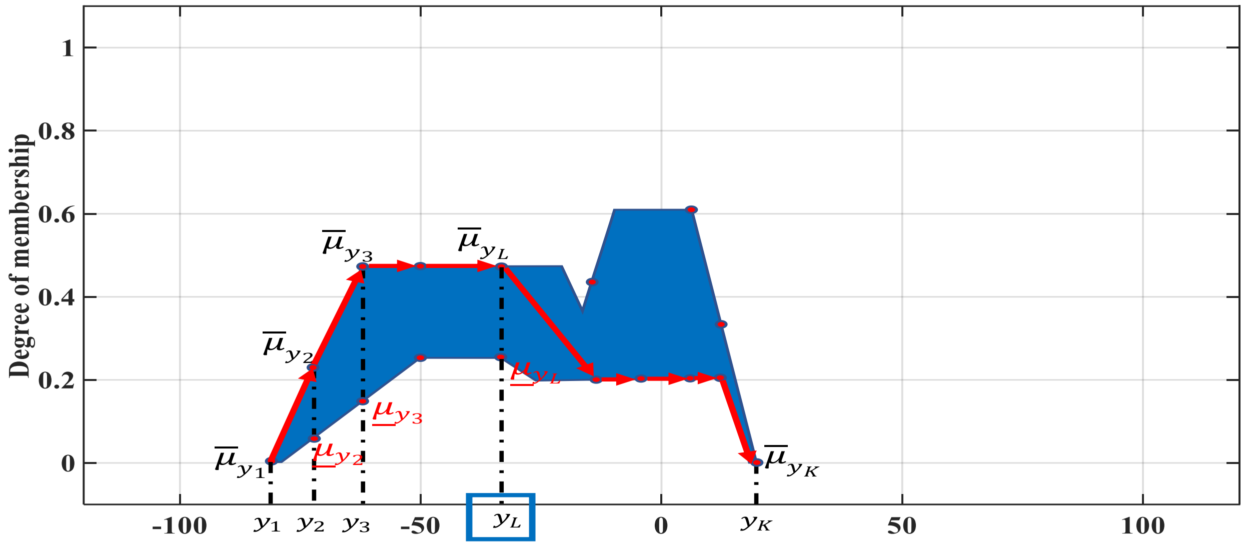


**Figure 4.8 $\Upsilon_l$ representation**

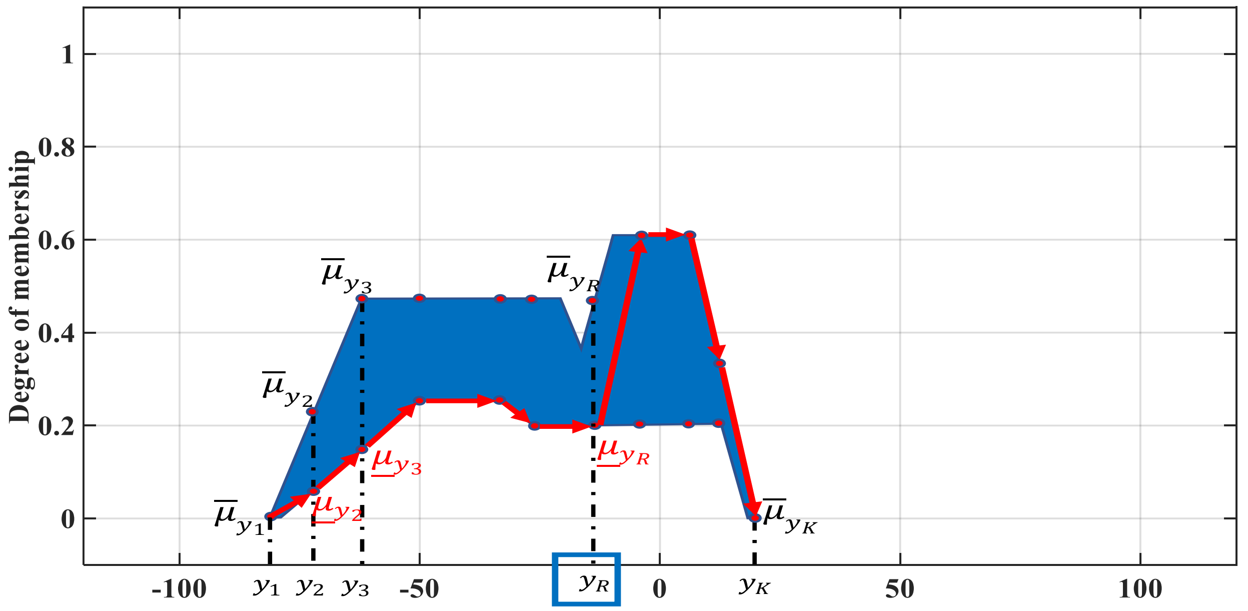


**Figure 4.9 $\Upsilon_r$ representation**

The easiest way to find $L$ and $R$ is through iterative approach TRA algorithms which starts by calculating candidate samples using (4-2) and (4-3) until the criteria of (4-4) and (4-5) are met. Consider $k$ as an integer number from the interval [1, K-1] we get [75]:

$$\Upsilon_l = \underbrace{min}_{k\in[1,K-1]} \Upsilon_l(k) = \Upsilon(L) = \frac{\sum_{i=1}^{L} y_i\overline{\mu}_i + \sum_{i=L+1}^{K} y_i\underline{\mu}_i}{\sum_{i=1}^{L} \overline{\mu}_i + \sum_{i=L+1}^{K} \underline{\mu}_i} \tag{4-2}$$

$$\Upsilon_r = \underbrace{max}_{k\in[1,K-1]} \Upsilon_r(k) = \Upsilon(R) = \frac{\sum_{i=1}^{R} y_i\underline{\mu}_i + \sum_{i=R+1}^{K} y_i\overline{\mu}_i}{\sum_{i=1}^{R} \underline{\mu}_i + \sum_{i=R+1}^{K} \overline{\mu}_i} \tag{4-3}$$

$$\Upsilon(L) \leq \Upsilon(l) < \Upsilon(L+1) \tag{4-4}$$

$$\Upsilon(R) \leq \Upsilon(r) < \Upsilon(R+1) \tag{4-5}$$

A variety of iterative TRA, such as KM, iterative algorithm with stop condition (IASC) [99], and enhanced iterative algorithm with stop condition (EIASC) can be utilized to find $L$ and $R$ [100]. Nevertheless, regardless of which algorithm is used, the performance of FLC2 won't be affected as long as the TR process is based on COA method.

Furthermore, when using KM TR algorithm, optimal switching points $L$ and $R$ can be obtained through the following steps [99]:

- For finding $\boldsymbol{L}$ and computing $\Upsilon_{\boldsymbol{l}}$:

1. **Step 1**: For $i = K$. Calculate $\sigma_i$ where:

$$\sigma_i = \frac{\overline{\mu}_i + \underline{\mu}_i}{2} \tag{4-6}$$

2. **Step 2**: compute $y$ where:

$$\Upsilon = \frac{\sum_{i=1}^{K} y_i\sigma_i}{\sum_{i=1}^{K} \sigma_i} \tag{4-7}$$

3. **Step 3**: Find $l \in (1 \leq l < K-1)$ such that:

$$\Upsilon_l \leq \Upsilon < \Upsilon_{l+1} \tag{4-8}$$

4. **Step 4**: Set the following value for $\sigma_i$:

$$\sigma_i = \begin{cases} \overline{\mu}_i, & i \leq l \\ \underline{\mu}_i, & i > l \end{cases} \tag{4-9}$$

And compute $y'$ where:

$$\Upsilon' = \frac{\sum_{i=1}^{K} y_i\sigma_i}{\sum_{i=1}^{K} \sigma_i} \tag{4-10}$$

5. **Step 5**: If $\Upsilon' = \Upsilon$ stop and set $L = l$ and $\Upsilon_l = \Upsilon$; otherwise, set $\Upsilon = \Upsilon_l$ and go back to **Step 3**.

- For finding $\boldsymbol{R}$ and computing $\boldsymbol{\Upsilon_r}$:

1. **Step 1**: For $i = K$. Calculate $\sigma_i$ where:

$$\sigma_i = \frac{\overline{\mu}_i + \underline{\mu}_i}{2} \tag{4-11}$$

2. **Step 2**: compute $\Upsilon$ where:

$$\Upsilon = \frac{\sum_{i=1}^{K} y_i \sigma_i}{\sum_{i=1}^{K} \sigma_i} \tag{4-12}$$

3. **Step 3**: Find $r \in (1 \le l < K - 1)$ such that:

$$\Upsilon_r \le \Upsilon < \Upsilon_{r+1} \tag{4-13}$$

4. **Step 4**: Set the following value for $\sigma_i$:

$$\sigma_i = \begin{cases} \underline{\mu}_i, & i \le r \\ \overline{\mu}_i, & i > r \end{cases} \tag{4-14}$$

And compute $y'$ where:

$$\Upsilon' = \frac{\sum_{i=1}^{K} y_i \sigma_i}{\sum_{i=1}^{K} \sigma_i} \tag{4-15}$$

5. **Step 5**: If $\Upsilon' = \Upsilon$ stop and set $R = r$ and $\Upsilon_r = \Upsilon$; otherwise, set $\Upsilon = \Upsilon_r$ and go back to **Step 3**.

### 4.1.4 FLC2 defuzzification

Once a TR algorithm is executed, and the switching points $L$ and $R$ are established, we still need to obtain a crisp output from the fuzzy set centroids $[\Upsilon_l, \Upsilon_r]$. Fortunately, defuzzification in the case of FLC2 is a straight forward process since the computations of COA are done during TRA phase. FLC2 crisp output $u_0$ can be calculated using (4-16) where [97]:

$$u_0 = \frac{\Upsilon_l + \Upsilon_r}{2} \tag{4-16}$$

## 4.2 FRAVTIONAL DERIVATION WITH A NOVEL ADAPTIVE LOW PASS FILTER

An integer derivative operator is often used to suppress overshoot in the transient response of nonlinear systems [101]. A fractional derivative is an extension of the conventional integer derivative with the ability to perform derivation with a non-integer order [102]. This non-integer order can be adjusted and tuned same way classical controllers are tuned using MHA. Nevertheless, this extended derivation flexibility allows the fractional derivative to perform better. However, it does not solve or reduce the problem of derivative kick back effect (DKE), which causes sudden spikes in the actual values of the variables of the controlled system. This crucial issue can be solved by adding a low pass filter (LF) to the fractional derivative operator. The combination of the fractional derivative and the LF is denoted by (LD) and can be modeled in the form of a simplified closed loop system that uses a fractional integrator instead of a fractional derivative as can be seen in Figure 4.10 [103]. Nonetheless, the output is still exactly the same of LD but with simpler computations. The transfer function of LD is presented in (4-17), where $E(s)$ is the input, and $U(s)$ is the output.

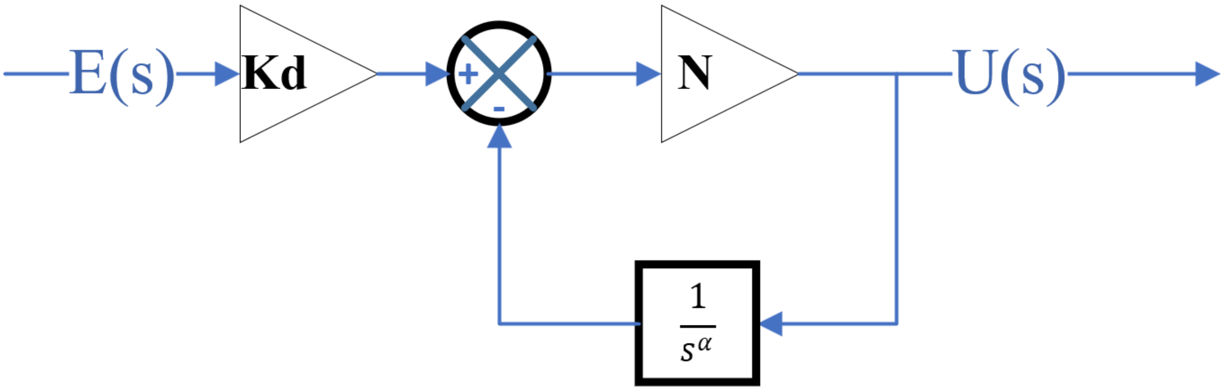


**Figure 4.10 Fractional derivative with a low pass filter**

$$LD = \frac{K_d \times N}{1 + N \times \frac{1}{s^{\alpha}}} \tag{4-17}$$

The output of the fractional integrator cannot be calculated, but it can be approximated using fractional integration approximation algorithms. In this research, we use refined oustaloop filter approximation algorithm  where $s^{\alpha}$ is approximated using (4-18) [104]:

$$s^{\alpha} \approx \left(\frac{d\omega_h}{b}\right)^{\alpha} \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha}\right) G_p \tag{4-18}$$

Here $G_p$ can be calculated using the following equation:

$$G_p(s) = \prod_{k=-N_f}^{N_f} \frac{s + \omega_k'}{s + \omega_k} \tag{4-19}$$

Where:

$$\omega_k = \left(\frac{b\omega_h}{d}\right)^{\frac{\alpha+2k}{2N_f+1}} \tag{4-20}$$

And:

$$\omega_k' = \left(\frac{b\omega_b}{d}\right)^{\frac{\alpha-2k}{2N_f+1}} \tag{4-21}$$

$b$ and $d$ values were based on [105] with $b = 10$ and $d = 9$.

As for $N_f$, $\omega_b$, and $\omega_h$, the following values were assigned via trial-and-error technique due their insignificant sensitivity and ease of calibration: $N_f = 50$, $\omega_b = 0.01$, $\omega_h = 10^5$.

It has been proven repeatedly that a fractional integrator is more flexible and can deliver a significantly superior output compared to the conventional non-fractional integrator if tuned properly, that's why it is not necessary to showcase a comparison between the two in this research. Nevertheless, LD performance is remarkable. However, it can be improved further more with an adaptive gain that adjusts the input of its fractional integrator in the feedback loop. This gain is denoted by $K_a$, and it helps adjust and reset the input and output of the fractional integrator. The overall fractional derivative and the adaptive LD is denoted by ALD. Moreover, ALD transfer function can be seen in (4-22), and its model is depicted in Figure 4.11.

$$ALD = \frac{Kd \times N}{1 + N \times K_a \times \frac{1}{s^{\alpha}}} \tag{4-22}$$

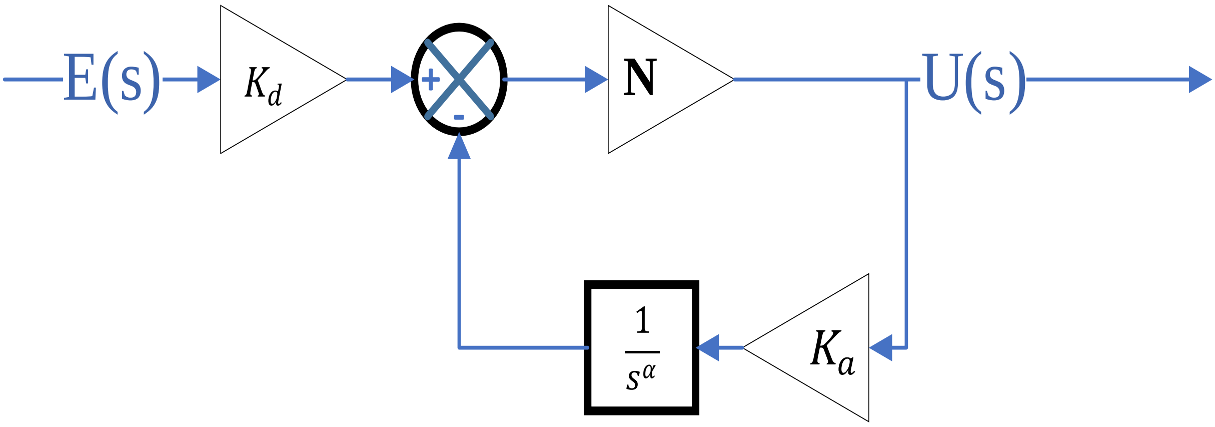


**Figure 4.11 Fractional derivative with an adaptive low pass filter**

$K_a$ gain switches its value based on the state and magnitude of the controlled input. In the case of speed control of PMSM, after completion of the tuning of F2MF-ALD, $K_a$ was adjusted manually via trial-and-error technique where:

If $\omega_{r(ref)} > 0$:

$$K_a = \begin{cases} 1, & if\ (\omega_{r(ref)} - \omega_r) > 10 \\ 5, & if\ (\omega_{r(ref)} - \omega_r) \leq 10 \end{cases} \tag{4-23}$$

Else if $\omega_{r(ref)} < 0$ :

$$K_a = \begin{cases} 1, & if\ \left|(\omega_{r(ref)} - \omega_r)\right| > 10 \\ 5, & if\ \left|(\omega_{r(ref)} - \omega_r)\right| \leq 10 \end{cases} \tag{4-24}$$

Finally, $\alpha$, $N$, and $K_a$were tuned along with the parameters of F2MF-ALD using symbiotic-particle-sewing (SPS) algorithm.

## 4.3 F2MF-ALD DESIGN AND TUNING

The structure of F2MF-ALD is simple, yet very effective. The coordinated cooperation between the two parts of this controller delivers an extraordinary uplift in performance compared to other similar fuzzy controllers in terms of transient response and disturbances rejection credentials. F2MF-ALD fuzzy type-2 based part ensures a solid disturbances rejection and a consistent SSE minimization, whereas the fractional derivative with the adaptive low pass filter part provides a resilient suppression of transient response overshoot. Figure 4.12 depicts the structure of F2MF-ALD in MATLAB/SIMULINK.

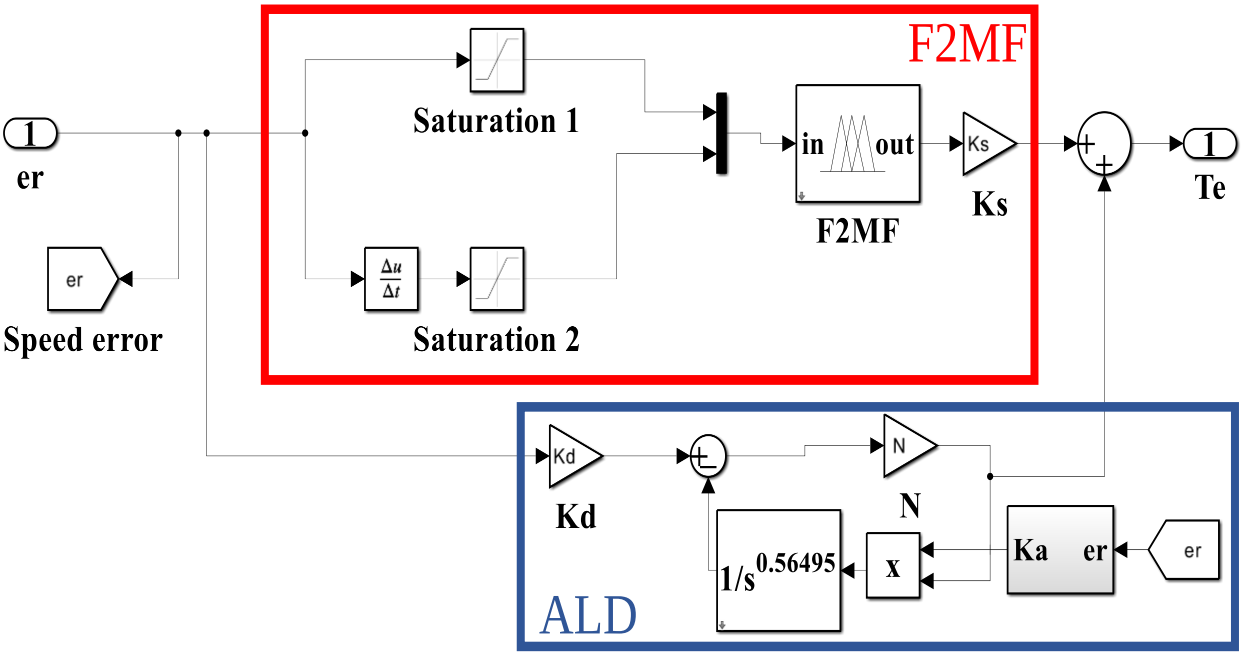


**Figure 4.12 F2MF-ALD model in MATLAB/SIMULINK**

Evidently, F2MF-ALD possesses the best features of all its simpler variants combinations which are: FLC2 with tuned MFS (F2MF), F2MF with LD (F2MF-LD), SPS-FLCMF which will be denoted by F1MF from here on, LD and FLC1 with tuned MFS (F1MF-LD), and F1MF with ALD (F1MF-ALD). Additionally, all these controllers were tuned using SPS algorithm.

F2MF-ALD, F2MF-LD, and F2MF tuning process was much more complex than the tuning process of F1MF-ALD, F1MF-LD, and F1MF. F2MF-ALD being the hardest to tune but not by much. Moreover, three stages of tuning were required to obtained optimal parameters for F2MF-ALD, these stages go as follows:

1. **Stage 1:** In this stage, a conventional Type-1 FLC with symmetric and optimized MFS is tuned along with PI current controller using SPS algorithm.

2. **Stage 2:** The Type-1 FLC from Stage one is converted into a Type-2 FLC by adding LMF to all its fuzzy sets, and then adding LD operator which is later tuned simultaneously with the added LMF and all the rest of the fuzzy controller scaling parameters using SPS. At the end of this stage, we get a fully optimized F2MF-LD.
3. **Stage 3:** this is the least complicated stage by far since it only adds the adaptive gain $K_a$ to LD turning it into ALD. $K_a$ doesn't require an extensive tuning process and thus so it is tuned via trial-and-error technique. At the end of this stage, we get the desired F2MF-ALD with optimized parameters.

The tuning process included a speed set point reference of nominal speed, a speed reversal, and then a 5 N.m external disturbance appliance. The performance of speed tracking of each controller can be quantified using the integral absolute error (IAE), and integral time-weighted absolute error (ITAE) functions as in [106] where:

$$IAE_{Speed\ error} = \int \left|(\omega_{r(ref)} - \omega_r)\right| \quad (4\text{-}25)$$

$$ITAE_{Speed\ error} = \int t \left|(\omega_{r(ref)} - \omega_r)\right| \quad (4\text{-}26)$$

ITE and ITAE of speed tracking performance of each controller can be seen in Table 5:

**Table 5 Ranking index and values of ITE and ITAE of speed tracking performance**

| | | F2MF-ALD | F1MF-ALD | F2MF-LD | F1MF-LD | F2MF | F1MF |
|---|---|---|---|---|---|---|---|
| IAE | Value | **1.02E+05** | 1.10E+05 | 1.02E+05 | 1.08E+05 | 1.09E+05 | 1.41E+05 |
| | Rank | **1** | 5 | 2 | 3 | 4 | 6 |
| ITAE | Value | **9.25E+04** | 1.00E+05 | 9.26E+04 | 1.00E+05 | 9.95E+04 | 1.39E+05 |
| | Rank | **1** | 5 | 2 | 4 | 3 | 6 |

**REMARK:** ‘’Saturation 2’’ interval parameters of all fuzzy controllers after tuning completion were assigned a very small values, hence forcing their fuzzy controllers to only interact with $ZO$ membership function of Input 2, hence making the rest of MFS of this fuzzy set irrelevant. While this might seem impractical, test results prove otherwise.

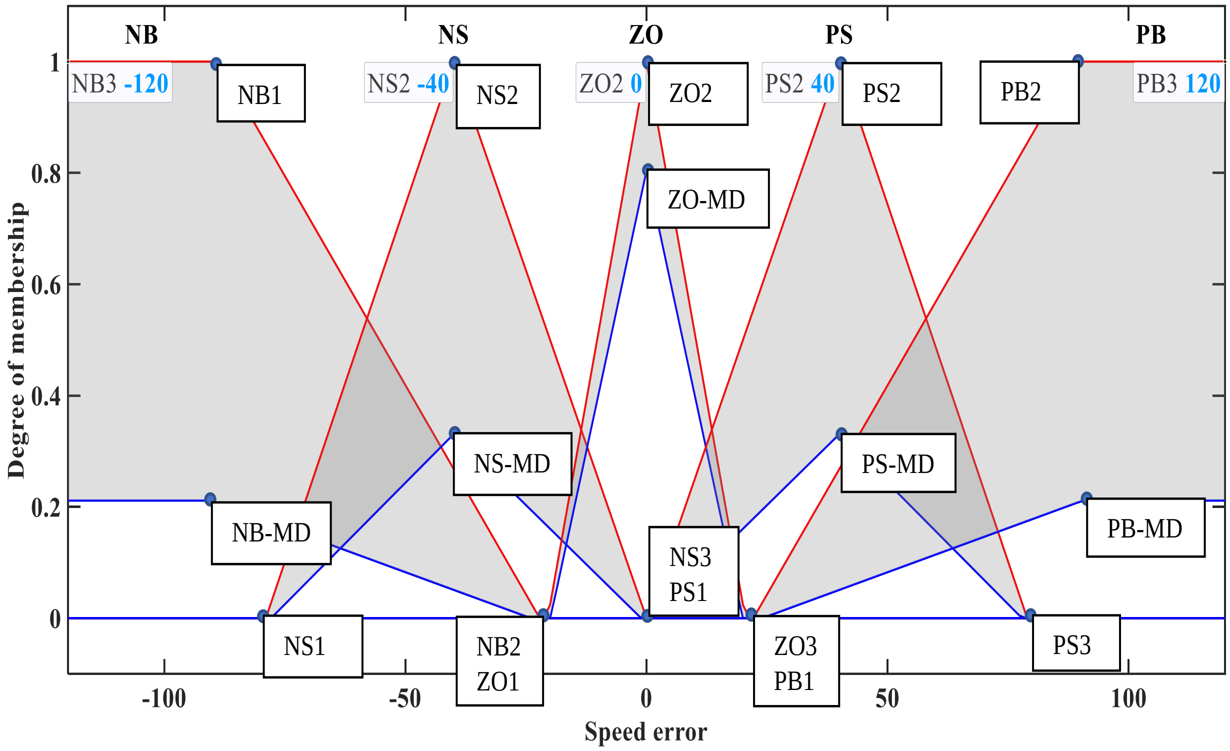


**Figure 4.13 Input 1 of F2MF-ALD**

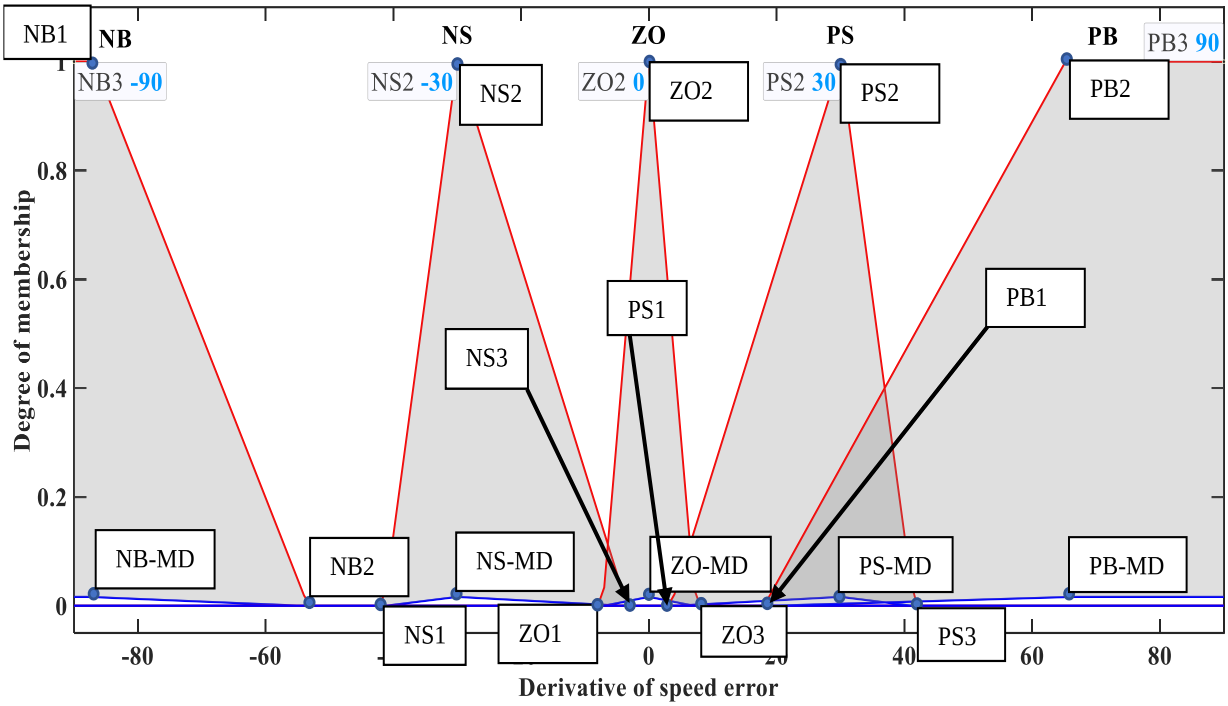


**Figure 4.14 Input 2 of F2MF-ALD**

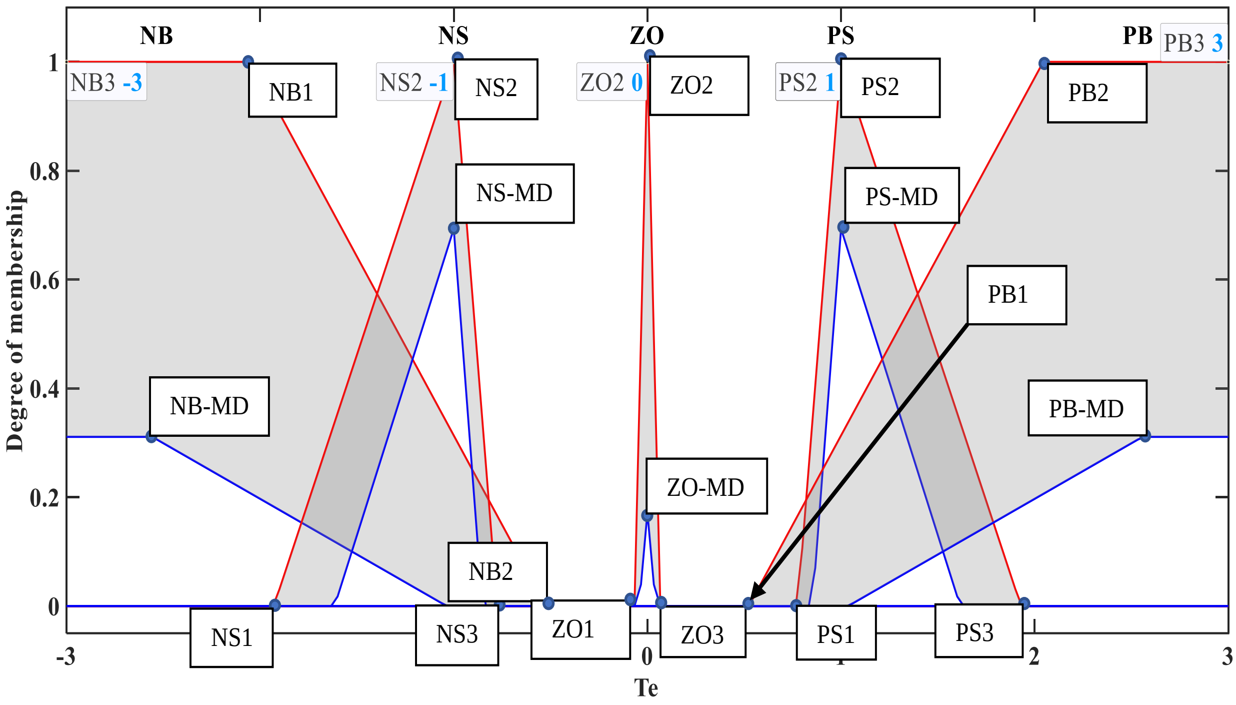


**Figure 4.15 Output of F2MF-ALD**

## 4.4 COMPARATIVE STUDY BETWEEN F2MF-ALD AND ITS VARIANTS

### 4.4.1 Test I: Speed set-point reference and load charge variations

In this test, a wide range of speed SSR variations were introduced to test the performance of F2MF-ALD against F2MF, F2MF-LD, F1MF, F1MF-LD, and F1MF-ALD during speed control of PMSM.

Throughout Test I, 8 different changes were introduced at 8 different instances where:

1. **Instance 1:** At 0 second, SSR increased from 0 to 100 $rad/s$.
2. **Instance 2**: At 0.5 second, SSR decreased from 100 to 50 $\frac{rad}{s}$.
3. **Instance 3:** At 1.5 second, speed reference was decreased and reversed from 50 to $-100\frac{rad}{s}$.

The outcome of this speed tracking performance of this test can be seen in Figure 4.16 with the instances SSR variations being highlighted and zoomed-in in Figure 4.17.

For a clear and thorough evaluation of speed tracking results, instances of speed reference changes will be analyzed based on SSE, settling time (ST), and overshoot (OS) magnitude of each controller.

Upon through analysis of the curves of speed tracking at **Instance 1, Instance 2, Instance 3,** Figure 4.17. It is evident that F2MF-ALD is the most accurate and successful at reducing the overshoot and transient response with minimal overshoot in the actual value of the produced mechanical speed. F2MF-LD performed similarly, but with slightly slower ST and some oscillations after applying a SSR of 50 rad/s. Furthermore, F1MF was accurate enough to match F2MF-ALD, whereas F1MF-LD, F2MF, and F1MF-ALD suffered a high SSE after the introduction of -100 rad/s SSR.

Additionally, results of OS, ST, and SSE during this test can be seen in Table 6 where it is evident that F2MF-ALD possesses the best features and potential which is affirmed by its superior SSR tracking benchmarks.

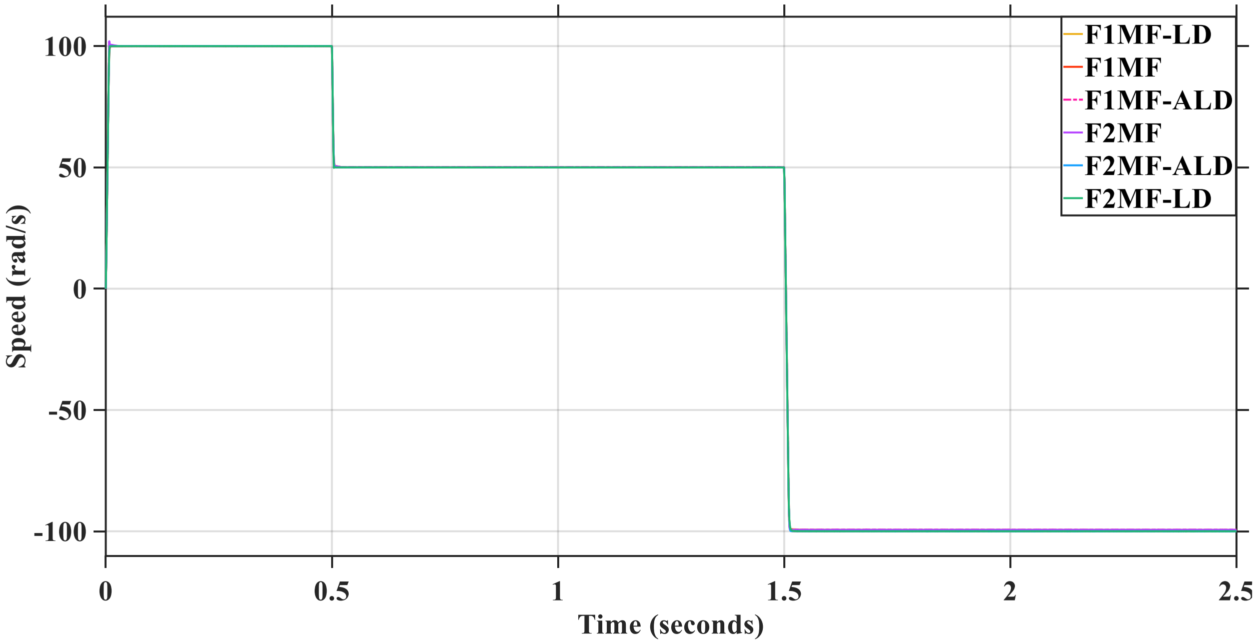


**Figure 4.16 Speed tracking performance of F2MF-ALD and its variants during Test I.**

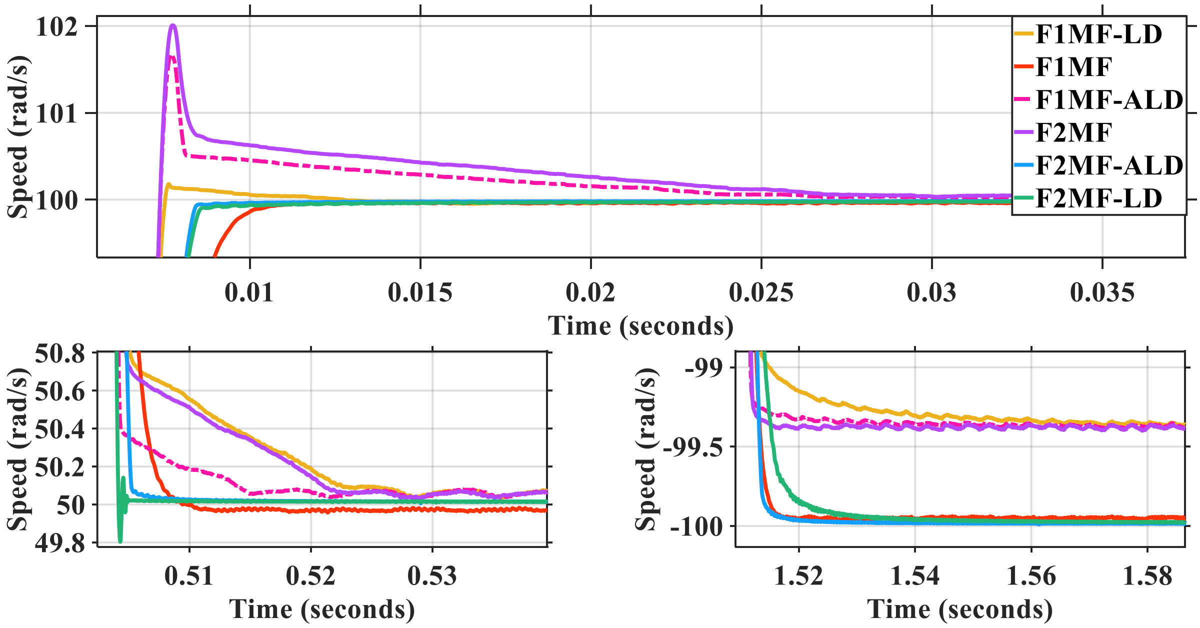


**Figure 4.17 Zoom-ins of all instances from Figure 4.16.**

**Table 6 Benchmarks of speed tracking SSE, OS, and ST during Test I.**

| | | F2MF-ALD | | F1MF-ALD | | F2MF-LD | | F1MF-LD | | F2MF | | F1MF | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Speed set point(rad/s) | ST (seconds) OS (rad/s) | Value | # | Value | # | Value | # | Value | # | Value | # | Value | # |
| **100** | ST | **0.011** | **1** | 0.0313 | 5 | 0.0132 | 2 | 0.0163 | 4 | 0.0318 | 6 | 0.0154 | 3 |
| | OS | **0** | **1** | 1.7 | 3 | **0** | **1** | 0.2 | 2 | 2 | 4 | **0** | **1** |
| | SSE | **0.01** | **1** | 0.03 | 2 | **0.01** | **1** | **0.01** | **1** | 0.03 | 4 | 0.04 | 4 |
| **50** | ST | 0.0080 | 2 | 0.0211 | 4 | **0.0057** | **1** | 0.029 | 6 | 0.0243 | 5 | 0.0132 | 3 |
| | OS | **0** | **1** | **0** | **1** | 0.2 | 2 | **0** | **1** | **0** | **1** | **0** | **1** |
| | SSE | **0.01** | **1** | 0.07 | 3 | **0.01** | **1** | 0.07 | 3 | 0.07 | 3 | 0.03 | 2 |
| **-100** | ST | 0.0206 | 2 | **0.0168** | **1** | 0.0285 | 4 | 0.029 | 5 | **0.0168** | **1** | 0.0216 | 3 |
| | OS | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** |
| | SSE | **0.01** | **1** | 0.62 | 5 | 0.02 | 2 | 0.64 | 6 | 0.6 | 4 | 0.04 | 3 |
| Sum of ranks (#) | | **11** | | 25 | | 15 | | 30 | | 29 | | 21 | |

### 4.4.2 Test II: appliance of load charges

This section aims to scrutinize and investigate how does F2MF-ALD fair against its simpler variants thorough an inclusive test that includes two different values of constant external disturbances throughout 2 seconds of speed control of PMSM simulation. External disturbances were introduced in the form of load charges increasing at certain instances of the simulation.

Overall, the whole test analysis was divided into 3 instances, last two instances include a load charge amplification. All instances were described as follows:

1. **Instance 1:** At 0 second, speed reference increased from $0$ to $105\ rad/s$.
2. **Instance 2:** At 1 second, a load charge of $5\ N.m$ was applied.
3. **Instance 3:** At 1.5 second, the applied load charge was increased from $5$ to $10\ N.m$.

The results of speed tracking with all the tested controllers can be seen in Figure 4.18. During steady-state after applying a load charge of 5 N.m. F2MF-ALD and F2MF-LD presented the best SSE minimization along with the least oscillations amplitude. Followed by F2MF with slightly higher SSE by with the highest oscillations amplitude. F1MF-ALD and F1MF-LD performed similarly. Whereas F1MF delivered the highest SSE among all tested controllers. After increasing the load charge to 10 N.m. F2MF-ALD and F2MF-LD fell back behind F2MF in terms of SSE magnitude. F1MF-ALD and F1MF-LD maintained similar performance compared to prior the increase in the load charge. Whereas in the case of F1MF, the SSE was increased massively compared to what it was during the appliance of only 5 N.m load charge.

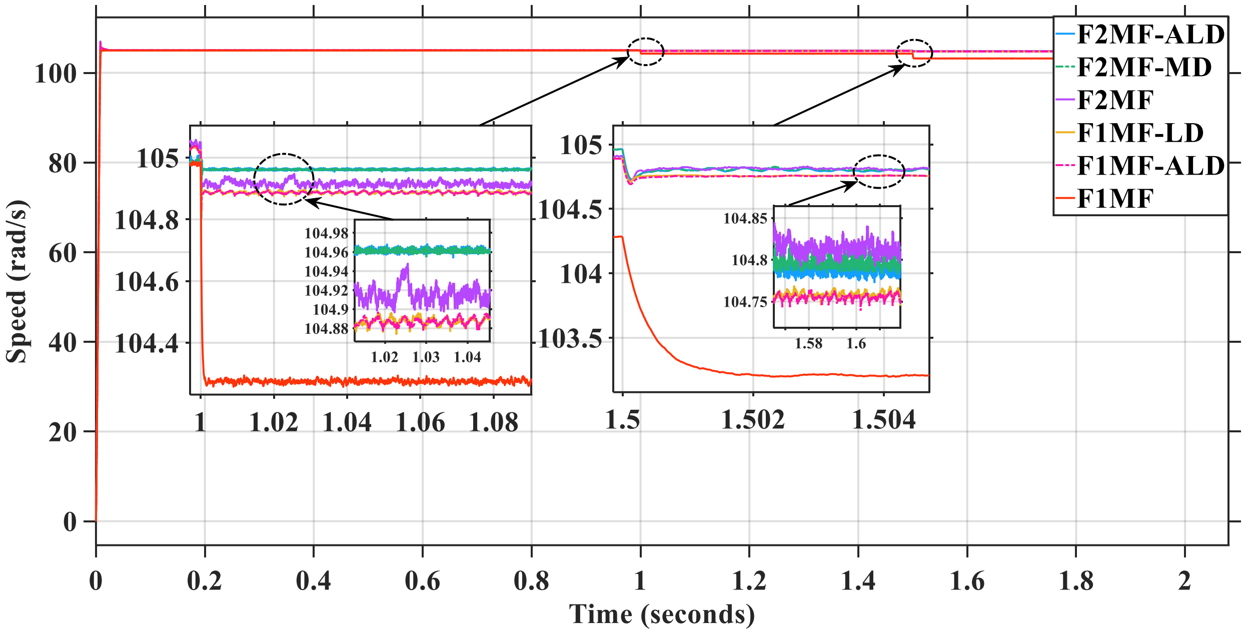


**Figure 4.18 Speed tracking performance of F2MF-ALD and its variants during Test II.**

Summary of results of SSE and total harmonics distortion (THD) during Test II can be seen in Table 7 where in regards to overall performance and ranking index, F2MF-ALD and F2MF-LD were the overall best with superior SSE minimization and THD suppression, followed by F1MF-LD and F1MF-ALD. F2MF managed to provide a smaller SSE than F1MF but with higher THD results.

**Table 7 results of SSE and THD during Test II**

| | | F2MF-ALD | | F1MF-ALD | | F2MF-LD | | F1MF-LD | | F2MF | | F1MF | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Load charge (N.m) | SSE (rad/s) THD | Value | # | Value | # | Value | # | Value | # | Value | # | Value | # |
| **5** | SSE | **0.04** | **1** | 0.12 | 3 | **0.04** | **1** | 0.012 | 3 | 0.09 | 2 | 0.073 | 4 |
| | THD | 3.19 | 2 | 4.06 | 4 | **3.15** | **1** | 3.98 | 3 | 8.9 | 6 | 6.84 | 5 |
| **10** | SSE | 0.21 | 2 | 0.23 | 3 | 0.21 | 2 | 0.23 | 3 | **0.2** | **1** | 1.8 | 4 |
| | THD | 2.37 | 3 | 2.09 | 2 | 2.39 | 4 | **2.04** | **1** | 4.45 | 6 | 3.41 | 5 |
| Sum of ranks (#) | | **8** | | 12 | | **8** | | 10 | | 15 | | 18 | |

$i_a$ phase currents curves during the transition from 5 N.m load charge to 10 N.m load charges. Obviously, due to the smaller value of THD, F2MF-ALD and F2MF-LD phase currents appears to have the smoothest sinusoidal shape compared to the rest of controllers especially F1MF and F2MF. F1MF-LD delivered the best THD minimization during the appliance of 10 N.m load charge. F1MF-ALD had a higher THD than F2MF-ALD and F2MF-LD in the presence of 5 N.m. however, it managed to outperform the latter after the last increase in load charge.

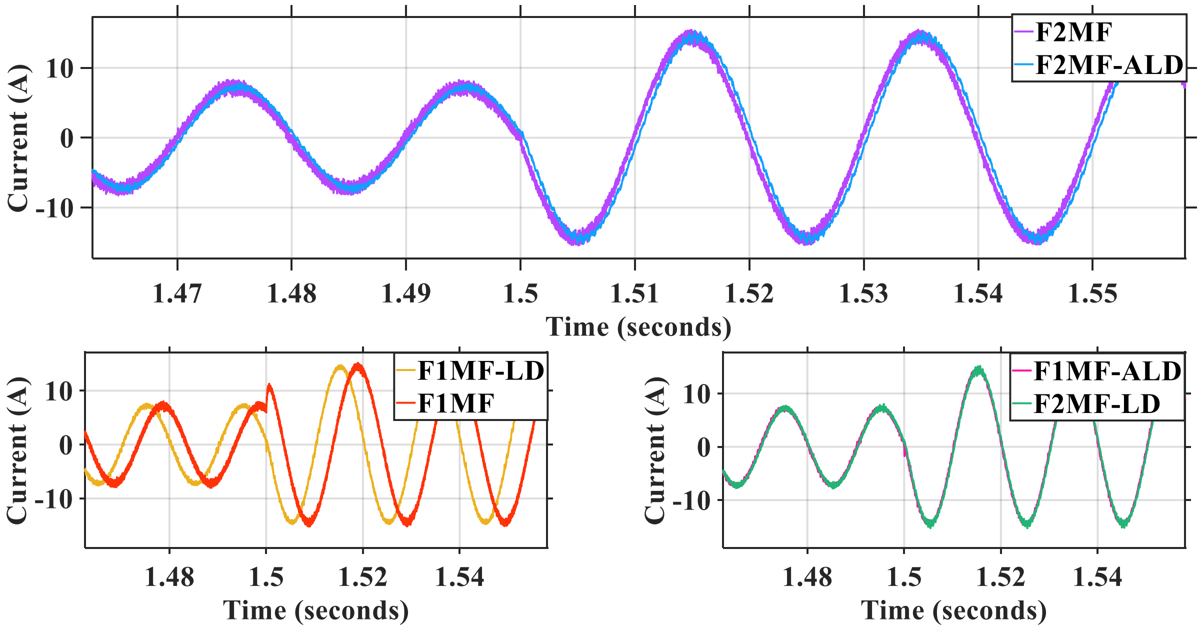


**Figure 4.19 $i_a$ phase currents during Test II.**

## 4.5 PROCESSOR IN THE LOOP (CO-SIMULATION) OF F2MF-ALD

MATLAB/SIMULIN environment also known as software-in-the loop (SIL) is a great tool to analyze and test out written algorithms code during the development of complex non-linear systems such as FOC of PMSM. However, its ideal properties and lack of disturbances and real time delays caused by environmental non-linearities such as variations of motor internal parameters and semi-conductors switching frequencies decrease due to temperature, humidity, and materials wearing off because of intense use pose a substantial doubt regarding its results. Nevertheless, it's still an essential process, and an easy way to have an overall idea of how a system would function in real time, and one of the few techniques to tune controllers' parameters without risking to damage physical hardware. In order to test non-linear systems with better validity than SIL we can use processor-in-the loop (PIL) co-simulation. Where instead of running both the plant and the control part on a simulated environment, PIL compiles and runs the developed algorithms of the control part on a target microprocessor such as TMS320F28379D while still running the code plant part on MATLAB/SIMULINK environment. If PIL simulation is successful and the results are similar enough to SIL simulation results, then the integrity of the developed controllers is validated, which means it's much more likely to succeed at controlling the target hardware of the plant in real time with no issues on the control part.

The control part of FOC of PMSM is basically the speed and currents controllers, and FFC and SVPWM algorithms, whereas the plant part is the voltages inverter, and the PMSM as can be seen in Figure 4.20.

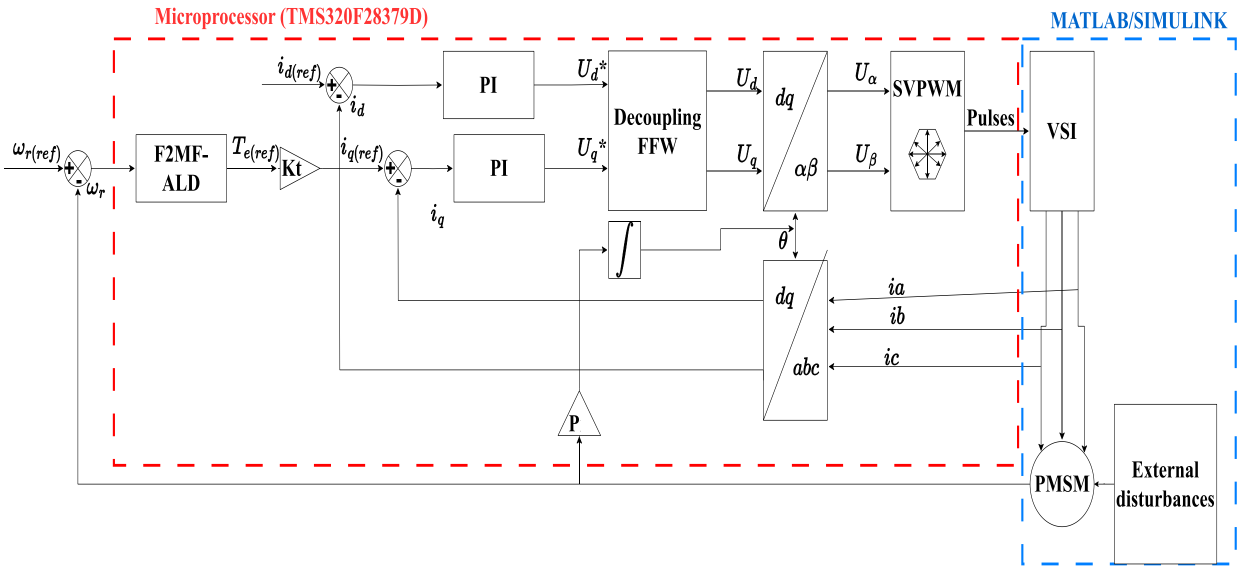


**Figure 4.20 PIL of FOC of PMSM using F2MF-ALD**

In order to execute PIL simulation of FOC of PMSM, the control part algorithms are compiled into code and then loaded onto the target hardware processor. In this research, the used microprocessor is TMS320F28379D due to its limitless capabilities, high computational power, low price, and its remarkable compatibility with MATLAB/SIMULINK algorithms. TMS320F28379D capability of controlling real time hardware using complicated controllers such as FLC has been proven repeatedly [107-111].

In order to validate the applicability of F2MF-ALD for speed control of PMSM, its SIL simulation was compared to its PIL co-simulation. The comparison included a few speed reference changes along with a 10 $N.m$ load charge applied after 1.5 seconds. The results of this comparison produced mechanical speed, electromagnetic torque, and the induced phase current can be seen in Figure 4.22.

In Figure 4.22, the results of the produced mechanical speed showed a higher noise and oscillations in the case of PIL compared to SIL. However, this noise was significantly reduced after the load charge was applied. In Figure 4.23.(a), and Figure 4.23.(b), the curves of the induced current and the produced electromagnetic torque also showed a high noise and oscillations in the case of PIL. However, after the appliance of the load charge, both PIL and SIL performed with more than enough similarity to prove the applicability of speed control of PMSM using F2MF-ALD.

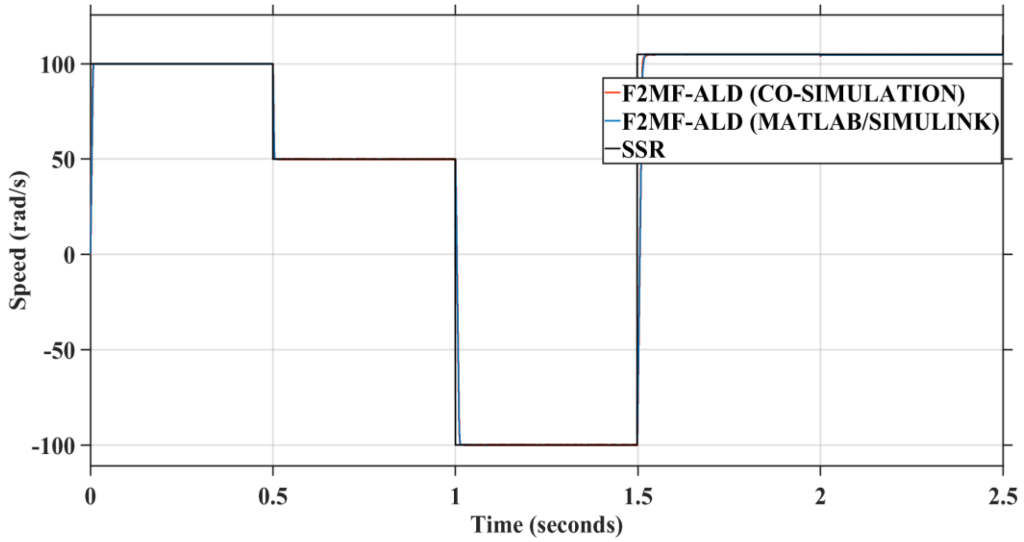


**Figure 4.21 SIL and PIL F2MF-ALD controlled mechanical speed comparison.**

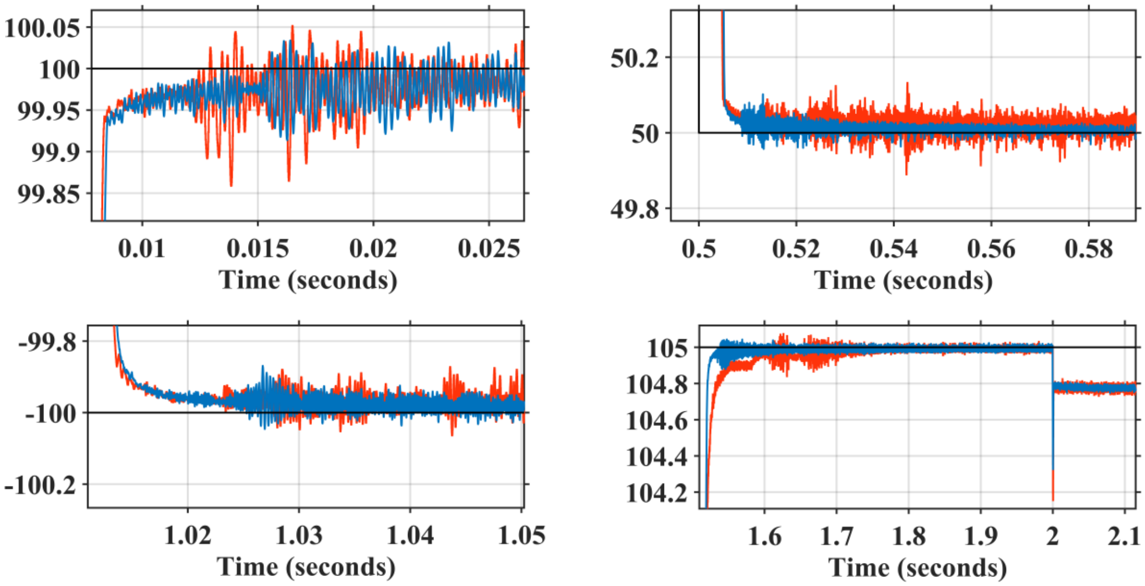


**Figure 4.22 Zoom-in of Figure 4.21**

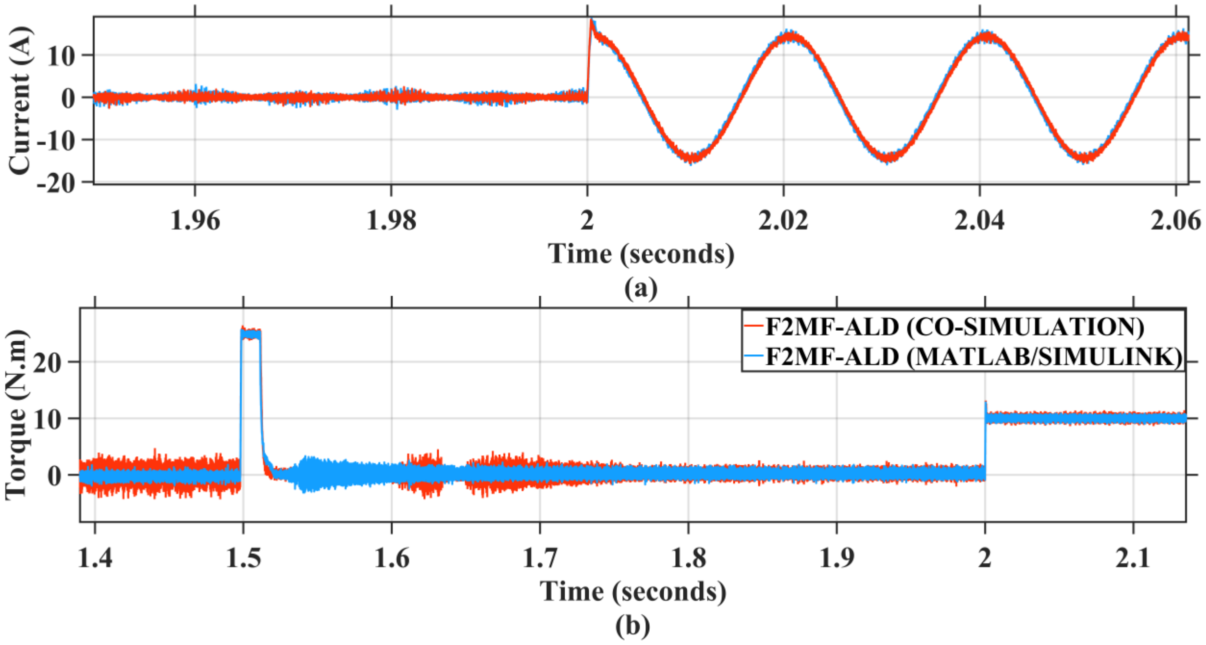


**Figure 4.23 a) Induced phase current during PIL test. b) Produced $T_e$ during PIL.**

## 4.6 CONCLUSION

To deal with the issue of SSE of speed tracking of the proposed F1MF when high external disturbances are applied, this chapter proposes the employment of interval Type-2 fuzzy logic along with fractional derivative operator combined with a new adaptive LF (ALF). First, Type-2 fuzzy logic concepts along with the fractional integration of feedback loop of the combination of the fractional derivative and ALF denoted by ALD were explained in detail. FLC2 part denoted by F2MF was built based on MAMDANI inference engine, and the fractional integration of ALD was producing an output using refined oustaloop filter approximation algorithm. The parameters of the proposed F2MF-ALD were tuned using SPS algorithm. Furthermore, a simulated comparison was carried on MATLAB/SIMULINK environment to find out how does F2MD-ALD fare against its simpler variants. Upon a thorough analysis of this comparison, F2MF-ALD superiority was confirmed where it managed to deliver a satisfying transient response along with a remarkable OS and SSE minimization in the presence of high external disturbances which is not an easy feat despite the present advancement of fuzzy logic controllers. Finally, PIL co-simulation of speed control of PMSM using F2MF-ALD was accomplished, which confirms the integrity and practical usability of the proposed controller.

# General conclusion

In order to understand the mechanics and the concept of PMSM vector control, Chapter 1 of this thesis presents a comprehensive overview of PMSM types, main structural parts, and foundational mathematical model equations, along with the detailed definition, working steps, and properties of SVPWM and VSI and how to integrate them into FOC of PMSM using PARK transformation and its variants.

Chapter 2 presents a newly proposed hybrid algorithm denoted by SPS for the tuning of PI controllers. The results of this tuning approach were better than classical tuning techniques such as poles imposition. However, it was still uncapable of providing the best-balanced performance possible in a system such as FOC of PMSM.

Chapter 3 changes the approach of speed control of PMSM presented in Chapter 2 by replacing the classical SPS-PI speed controller by an AI inspired FLC controller with MHA based MFS. After thoroughly explaining the main steps of MAMDANI based FLC, SPS algorithm was used to tune the parameters of the designed F1MF scaling factors and MFS. Additionally, when compared to SPS-PI speed controller presented in Chapter 2 and FLC with unoptimized MFS (FC1), F1MF reigned supreme with better speed tracking OS rejection and ST. However, it was unsuccessful in efficiently reducing the speed tracking SSE caused by the application of high load charges.

Chapter 4 introduces F2MF-ALD which is a combination of an FC2 and a modified design of a fractional derivative low pass filter combination. F2MF-ALD was tested intensively in both PIL co-simulation and MATLAB/SIMULINK environments where it was proven to be able to preserve F1MF fast and resilient response with a significantly improved disturbance rejection and SSE reduction capabilities compared to the rest of the tested controllers such as the classical PI and the conventional F2MF. Moreover, for future work, the focus will be on performing stability analysis for F2MF-ALD with experimental testing and validation. As well as more refinement and upgrades for better dynamic performance including a superior THD minimization.

# References

[1] R. Saidur, "A review on electrical motors energy use and energy savings," *Renewable and sustainable energy reviews,* vol. 14, no. 3, pp. 877-898, 2010.

[2] F. Petruzella, *Electric motors and control systems*. McGraw-Hill, Inc., 2009.

[3] A. B. Yildiz, "Electrical equivalent circuit based modeling and analysis of direct current motors," *International Journal of Electrical Power & Energy Systems,* vol. 43, no. 1, pp. 1043-1047, 2012.

[4] S. Singer and J. Appelbaum, "Starting characteristics of direct current motors powered by solar cells," *IEEE Transactions on Energy Conversion,* vol. 8, no. 1, pp. 47-53, 1993.

[5] S. Yamamura, *AC motors for high performance applications: analysis and control*. CRC Press, 1986.

[6] H. E. Gelani, F. Dastgeer, M. Nasir, S. Khan, and J. M. Guerrero, "AC vs. DC distribution efficiency: Are we on the right path?," *Energies,* vol. 14, no. 13, p. 4039, 2021.

[7] A. M. Trzynadlowski, *Control of induction motors*. Elsevier, 2000.

[8] K. M. Rahman and S. Hiti, "Identification of machine parameters of a synchronous motor," *IEEE Transactions on Industry Applications,* vol. 41, no. 2, pp. 557-565, 2005.

[9] T. A. Bigelow, *Electric circuits, systems, and motors*. Springer, 2020.

[10] A. G. M. A. Aziz, A. Y. Abdelaziz, Z. M. Ali, and A. A. Z. Diab, "A comprehensive examination of vector-controlled induction motor drive techniques," *Energies,* vol. 16, no. 6, p. 2854, 2023.

[11] W. Cai, X. Wu, M. Zhou, Y. Liang, and Y. Wang, "Review and development of electric motor systems and electric powertrains for new energy vehicles," *Automotive Innovation,* vol. 4, pp. 3-22, 2021.

[12] S. S. Rauth and B. Samanta, "Comparative analysis of IM/BLDC/PMSM drives for electric vehicle traction applications using ANN-Based FOC," in *2020 IEEE 17th India Council International Conference (INDICON),* 2020: IEEE, pp. 1-8.

[13] P. Thampi, "A review on controlling techniques for permanent magnet synchronous motor (PMSM) and current state of the art in the research area," *Communications on Applied Electronics,* vol. 7, no. 25, pp. 8-17, 2019.

[14] D. Ocen, "Direct torque control of a permanent magnet synchronous motor," ed, 2005.

[15] A. Guezi, A. Bendaikha, and A. Dendouga, "Direct torque control based on second order sliding mode controller for three-level inverter-fed permanent magnet synchronous motor: comparative study," *Electrical Engineering & Electromechanics,* no. 5, pp. 10-13, 2022.

[16] V. M. Bida, D. V. Samokhvalov, and F. S. Al-Mahturi, "PMSM vector control techniques—A survey," in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus),* 2018: IEEE, pp. 577-581.

[17] A. Dendouga, "Conventional and second order sliding mode control of permanent magnet synchronous motor fed by Direct Matrix Converter: Comparative Study," *Energies,* vol. 13, no. 19, p. 5093, 2020.

[18] T. Nouaoui, A. Dendouga, and A. Bendaikha, "Fractional Order PID tuned using symbiotic organism search algorithm for field oriented control of the permanent magnet synchronous motor," in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS),* 2023: IEEE, pp. 1-5.

[19] T. Nouaoui, A. Dendouga, and A. Bendaikha, "Design and Tuning of Fuzzy Controller using Human Inspired Algorithm for Speed Control of PMSM," in *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM),* 2023, vol. 1: IEEE, pp. 1-6.

[20] A. Djerioui, A. Houari, M. Ait-Ahmed, M.-F. Benkhoris, A. Chouder, and M. Machmoum, "Grey Wolf based control for speed ripple reduction at low speed operation of PMSM drives," *ISA transactions,* vol. 74, pp. 111-119, 2018.

[21] N. Bounasla, "Commande par mode de glissement d'ordre supérieur de la machine synchrone à aimants permanents," 2018.

[22] A. A. Abd Samat, M. Zainal, L. Ismail, W. S. Saidon, and A. I. Tajudin, "Current PI-gain determination for permanent magnet synchronous motor by using particle swarm optimization," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 6, no. 2, pp. 412-421, 2017.

[23] A. Mohammed, W. S. Saidon, M. A. A. Razak, and A. A. A. Samat, "Optimization of PI Parameters for Speed Controller of a Permanent Magnet Synchronous Motor by using Particle Swarm Optimization Technique," *International Journal of Simulation--Systems, Science & Technology,* vol. 17, no. 41, 2016.

[24] F. Mohd Zaihidee, S. Mekhilef, and M. Mubin, "Robust speed control of PMSM using sliding mode control (SMC)—A review," *Energies,* vol. 12, no. 9, p. 1669, 2019.

[25] P. Q. Khanh and H. P. H. Anh, "Hybrid optimal fuzzy Jaya technique for advanced PMSM driving control," *Electrical Engineering,* vol. 105, no. 6, pp. 3629-3646, 2023.

[26] T. Nouaoui, A. Dendouga, and A. Bendaikha, "Speed control of PMSM using a fuzzy logic controller with deformed MFS tuned by a novel hybrid meta-heuristic algorithm," *Electrical Engineering,* pp. 1-13, 2024.

[27] H. Açıkgöz, O. F. Kececıoglu, A. Ganı, and M. Sekkelı, "Speed control of direct torque controlled induction motor by using PI, anti-windup PI and fuzzy logic controller," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 2, no. 3, pp. 58-63, 2014.

[28] S. Ünsal and I. Aliskan, "Investigation of performance of fuzzy logic controllers optimized with the hybrid genetic-gravitational search algorithm for PMSM speed control," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije,* vol. 63, no. 2, pp. 313-327, 2022.

[29] L. Zhang, J. Ma, Q. Wu, Z. He, T. Qin, and C. Chen, "Research on PMSM speed performance based on fractional order adaptive fuzzy backstepping control," *Energies,* vol. 16, no. 19, p. 6922, 2023.

[30] H. Ahn, H. Park, C. Kim, and H. Lee, "A Review of State-of-the-art Techniques for PMSM Parameter Identification," *Journal of Electrical Engineering & Technology,* vol. 15, pp. 1177-1187, 2020.

[31] H. Chaoui, M. Khayamy, and A. A. Aljarboua, "Adaptive interval type-2 fuzzy logic control for PMSM drives with a modified reference frame," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 5, pp. 3786-3797, 2017.

[32] M. Nouh, B. A. Zalam, and A. Sayed, "Hybrid FOT2F-FOPD controller for permanent magnet synchronization motor based on ILA optimization with SRF-PLL," *Scientific Reports,* vol. 14, no. 1, p. 13095, 2024.

[33] A. Herizi, R. Rouabhi, and A. Zemmit, "Comparative study of the performance of a sliding, sliding-fuzzy type 1 and a sliding-fuzzy type 2 control of a permanent magnet synchronous machine," *Przeglad Elektrotechniczny,* vol. 98, no. 11, 2022.

[34] Y. Qiao and K. Wang, "Fuzzy sliding mode speed control strategy of permanent magnet motor under variable load condition," *International Journal of Dynamics and Control,* vol. 12, no. 5, pp. 1616-1625, 2024.

[35] D. Wu, Y. Liao, C. Hu, S. Yu, and Q. Tian, "An enhanced fuzzy control strategy for low-level thrusters in marine dynamic positioning systems based on chaotic random distribution harmony search," *International Journal of Fuzzy Systems,* vol. 23, pp. 1823-1839, 2021.

[36] Y. Yang, Q. He, C. Fu, S. Liao, and P. Tan, "Efficiency improvement of permanent magnet synchronous motor for electric vehicles," *Energy,* vol. 213, p. 118859, 2020.

[37] Q. Huang, Q. Huang, H. Guo, and J. Cao, "Design and research of permanent magnet synchronous motor controller for electric vehicle," *Energy Science & Engineering,* vol. 11, no. 1, pp. 112-126, 2023.

[38] Y. Song *et al.*, "Barrier Lyapunov function-based adaptive prescribed performance control of the PMSM used in robots with full-state and input constraints," *Journal of Vibration and Control,* vol. 29, no. 5-6, pp. 1400-1416, 2023.

[39] A. P. Thurlbeck and Y. Cao, "Fully integrated fault-tolerance for PMSMs in aviation applications," in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020: IEEE, pp. 4916-4922.

[40] W.-S. Jung, H.-K. Lee, Y.-K. Lee, S.-M. Kim, J.-I. Lee, and J.-Y. Choi, "Analysis and Comparison of Permanent Magnet Synchronous Motors According to Rotor Type under the Same Design Specifications," *Energies*, vol. 16, no. 3, p. 1306, 2023.

[41] D.-C. Pang, Z.-J. Shi, P.-X. Xie, H.-C. Huang, and G.-T. Bui, "Investigation of an inset micro permanent magnet synchronous motor using soft magnetic composite material," *Energies*, vol. 13, no. 17, p. 4445, 2020.

[42] B. Qu, Q. Yang, Y. Li, M. A. Sotelo, S. Ma, and Z. Li, "A novel surface inset permanent magnet synchronous motor for electric vehicles," *Symmetry*, vol. 12, no. 1, p. 179, 2020.

[43] K.-Y. Yoon and S.-W. Baek, "Performance improvement of concentrated-flux type ipm pmsm motor with flared-shape magnet arrangement," *Applied Sciences*, vol. 10, no. 17, p. 6061, 2020.

[44] A. K. Sahu, R. Z. Haddad, D. Al-Ani, and B. Bilgin, "Thermomechanical Rotor Fatigue of an Interior Permanent Magnet Synchronous Motor," *Machines*, vol. 12, no. 3, p. 158, 2024.

[45] J.-W. Jung, B.-H. Lee, K.-S. Kim, and S.-I. Kim, "Interior permanent magnet synchronous motor design for eddy current loss reduction in permanent magnets to prevent irreversible demagnetization," *Energies*, vol. 13, no. 19, p. 5082, 2020.

[46] H. Elsherbiny, L. Szamel, M. K. Ahmed, and M. A. Elwany, "High accuracy modeling of permanent magnet synchronous motors using finite element analysis," *Mathematics*, vol. 10, no. 20, p. 3880, 2022.

[47] Z. Chen, F. Xiao, Y. Chen, W. Tang, and Z. Chen, "An Improved Model Predictive Direct Speed Control with Synchronous Prediction and Weight Factor Optimization for PMSM Application," *Journal of Electrical Engineering & Technology*, vol. 18, no. 6, pp. 4257-4268, 2023.

[48] S. O. P. MAGNET, "Mathematical modeling and simulation of permanent magnet synchronous motor," *International journal of advanced research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 8, 2013.

[49] H. Swathi Hatwar, A. Shetty, and K. Suryanarayana, "Mathematical Model Approach to Study and Analyze FOC-Based PMSM Control," in *International Conference on Advances in Renewable Energy and Electric Vehicles*, 2022: Springer, pp. 25-45.

[50] A. S. Mokhtar, M. B. I. Reaz, and A. M. ALI, "EFFICIENT FPGA-BASED INVERSE PARK TRANSFORMATION OF PMSM MOTOR USING CORDIC ALGORITHM," *Journal of Theoretical and Applied Information Technology*, vol. 65, no. 1, 2014.

[51] Z. Wang, J. Chen, M. Cheng, and Y. Zheng, "Fault-tolerant control of paralleled-voltage-source-inverter-fed PMSM drives," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 4749-4760, 2015.

[52] F. Wang, W. Shen, D. Boroyevich, S. Ragon, V. Stefanovic, and M. Arpilliere, "Voltage source inverter," *IEEE Industry Applications Magazine*, vol. 15, no. 2, pp. 24-33, 2009.

[53] K. V. Kumar, P. A. Michael, J. P. John, and S. S. Kumar, "Simulation and comparison of SPWM and SVPWM control for three phase inverter," *ARPN journal of engineering and applied sciences*, vol. 5, no. 7, pp. 61-74, 2010.

[54] N. V. Nho and M.-J. Youn, "Comprehensive study on space-vector-PWM and carrier-based-PWM correlation in multilevel invertors," *IEE Proceedings-Electric Power Applications*, vol. 153, no. 1, pp. 149-158, 2006.

[55] O. C. Kivanc and S. B. Ozturk, "MATLAB function based approach to FOC of PMSM drive," in *2015 IEEE European Modelling Symposium (EMS)*, 2015: IEEE, pp. 96-102.

[56] M. NICOLA, C.-I. NICOLA, and D. Marian, "Sensorless Control of PMSM Based on FOC Strategy and Fractional Order PI Controller," in *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*, 2020: IEEE, pp. 302-307.

[57] O. Dieterle, T. Greiner, and P. Heidrich, "Feedforward compensation of torque ripples in dual three-phase PMSM fed from separate DC links with different voltage levels," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 9036-9045, 2020.

[58] V. G. BT, "Comparison between direct and indirect field oriented control of induction motor," *Int. J. Eng. Trends Technol*, vol. 43, no. 6, pp. 364-369, 2017.

[59] X. Sun, N. Xu, M. Yao, F. Cai, and M. Wu, "Efficient feedback linearization control for an IPMSM of EVs based on improved firefly algorithm," *ISA transactions,* vol. 134, pp. 431-441, 2023.

[60] Q. Zhang and C. Zhang, "Speed Control of PMSM Based on Fuzzy Active Disturbance Rejection Control under Small Disturbances," *Applied Sciences,* vol. 13, no. 19, p. 10775, 2023.

[61] C.-H. Liu and Y.-Y. Hsu, "Design of a self-tuning PI controller for a STATCOM using particle swarm optimization," *IEEE Transactions on industrial Electronics,* vol. 57, no. 2, pp. 702-715, 2009.

[62] E. Aguilar-Garnica, J. P. García-Sandoval, and V. González-Álvarez, "PI controller design for a class of distributed parameter systems," *Chemical engineering science,* vol. 66, no. 18, pp. 4009-4019, 2011.

[63] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications,* pp. 185-231, 2018.

[64] M. A. A. Almubaidin, A. N. Ahmed, L. B. M. Sidek, and A. Elshafie, "Using metaheuristics algorithms (MHAs) to optimize water supply operation in reservoirs: a review," *Archives of Computational Methods in Engineering,* vol. 29, no. 6, pp. 3677-3711, 2022.

[65] D. Zabia, H. Afghoul, and O. Kraa, "Maximum power point tracking of a photovoltaic system under partial shading condition using whale optimization algorithm," in *Artificial Intelligence and Heuristics for Smart Energy Efficiency in Smart Cities: Case Study: Tipasa, Algeria*, 2022: Springer, pp. 121-132.

[66] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems," *Heliyon,* vol. 8, no. 5, 2022.

[67] M. Dehghani, E. Trojovská, and T. Zuščák, "A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training," *Scientific reports,* vol. 12, no. 1, p. 17387, 2022.

[68] A. E. Ezugwu and D. Prayogo, "Symbiotic organisms search algorithm: theory, recent advances and applications," *Expert Systems with Applications,* vol. 119, pp. 184-209, 2019.

[69] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft computing,* vol. 22, no. 2, pp. 387-408, 2018.

[70] R. S. Rebeiro and M. N. Uddin, "Performance analysis of an FLC-based online adaptation of both hysteresis and PI controllers for IPMSM drive," *IEEE Transactions on Industry Applications,* vol. 48, no. 1, pp. 12-19, 2011.

[71] S. Waley, C. Mao, and N. K. Bachache, "Biogeography based optimization for tuning FLC controller of PMSM," in *Advances in Swarm and Computational Intelligence: 6th International Conference, ICSI 2015, held in conjunction with the Second BRICS Congress, CCI 2015, Beijing, China, June 25-28, 2015, Proceedings, Part I 6,* 2015: Springer, pp. 395-402.

[72] M. Chakraborty, "Comparative analysis of speed control of PMSM using PI-controller and fuzzy controller," *International Journal of Scientific & Engineering Research,* vol. 4, no. 7, pp. 103-108, 2013.

[73] H. R. Berenji, "Fuzzy logic controllers," in *An introduction to fuzzy logic applications in intelligent systems*: Springer, 1992, pp. 69-96.

[74] A. Ansari and J. Islamia, "The basics of fuzzy logic: A tutorial review," *COMPUTER EDUCATION-STAFFORD-COMPUTER EDUCATION GROUP-,* vol. 88, pp. 5-8, 1998.

[75] R. Antão, *Type-2 fuzzy logic: uncertain systems' modeling and control*. Springer, 2017.

[76] P. Guillemin, "Fuzzy logic applied to motor control," *IEEE transactions on industry applications,* vol. 32, no. 1, pp. 51-56, 1996.

[77] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on systems, Man, and Cybernetics,* no. 1, pp. 28-44, 1973.

[78] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Information Sciences,* vol. 181, no. 20, pp. 4340-4360, 2011.

[79] D. Dubois and H. Prade, "What are fuzzy rules and how to use them," *Fuzzy sets and systems,* vol. 84, no. 2, pp. 169-185, 1996.

[80] V. P. Popchev, Ivan, "Fuzzy logic operators in decision-making," *Cybernetics & Systems,* vol. 30, no. 8, pp. 725-745, 1999.

[81] M. M. Gupta and J. Qi, "Theory of T-norms and fuzzy inference methods," *Fuzzy sets and systems,* vol. 40, no. 3, pp. 431-450, 1991.

[82] W.-B. Zhu, B. Shuai, and S.-H. Zhang, "The linguistic interval-valued intuitionistic fuzzy aggregation operators based on extended Hamacher t-norm and s-norm and their application," *Symmetry,* vol. 12, no. 4, p. 668, 2020.

[83] R. Boukezzoula, S. Galichet, and L. Foulloy, "MIN and MAX operators for fuzzy intervals and their potential use in aggregation operators," *IEEE Transactions on Fuzzy Systems,* vol. 15, no. 6, pp. 1135-1144, 2007.

[84] S. Uppalapati and D. Kaur, "Design and implementation of a Mamdani fuzzy inference system on an FPGA," in *NAFIPS 2009-2009 Annual Meeting of the North American Fuzzy Information Processing Society,* 2009: IEEE, pp. 1-6.

[85] M. Salman and N. I. Seno, "A comparison of Mamdani and Sugeno inference systems for a satellite image classification," *Anbar Journal for Engineering Sciences,* pp. 296-306, 2010.

[86] R. R. Yager and D. P. Filev, "Generalizing the modeling of fuzzy logic controllers by parameterized aggregation operators," *Fuzzy Sets and Systems,* vol. 70, no. 2-3, pp. 303-313, 1995.

[87] I. Iancu, "A Mamdani type fuzzy logic controller," *Fuzzy logic-controls, concepts, theories and applications,* vol. 15, no. 2, pp. 325-350, 2012.

[88] S. Izquierdo and L. R. Izquierdo, "Mamdani fuzzy systems for modelling and simulation: A critical assessment," *Available at SSRN 2900827,* 2017.

[89] J. J. Saade and H. B. Diab, "Defuzzification methods and new techniques for fuzzy controllers," 2004.

[90] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy sets and systems,* vol. 108, no. 2, pp. 159-178, 1999.

[91] S. Akkurt, G. Tayfur, and S. Can, "Fuzzy logic model for the prediction of cement compressive strength," *Cement and concrete research,* vol. 34, no. 8, pp. 1429-1433, 2004.

[92] W. W. Tan and T. W. Chua, "Uncertain rule-based fuzzy logic systems: introduction and new directions (Mendel, JM; 2001)[book review]," *IEEE Computational intelligence magazine,* vol. 2, no. 1, pp. 72-73, 2007.

[93] D. W. W. W. Tan, "A simplified type-2 fuzzy logic controller for real-time control," *ISA transactions,* vol. 45, no. 4, pp. 503-516, 2006.

[94] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information sciences,* vol. 8, no. 3, pp. 199-249, 1975.

[95] H. Hassani and J. Zarei, "Interval Type-2 fuzzy logic controller design for the speed control of DC motors," *Systems Science & Control Engineering,* vol. 3, no. 1, pp. 266-273, 2015.

[96] J. M. Mendel, "Uncertain rule-based fuzzy systems," *Introduction and new directions,* vol. 684, 2017.

[97] B. Yao, H. Hagras, D. Alghazzawi, and M. J. Alhaddad, "A big bang–big crunch type-2 fuzzy logic system for machine-vision-based event detection and summarization in real-world ambient-assisted living," *IEEE Transactions on Fuzzy Systems,* vol. 24, no. 6, pp. 1307-1319, 2016.

[98] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *information SCiences,* vol. 132, no. 1-4, pp. 195-220, 2001.

[99] C. Chen, D. Wu, J. M. Garibaldi, R. I. John, J. Twycross, and J. M. Mendel, "A comprehensive study of the efficiency of type-reduction algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 6, pp. 1556-1566, 2020.

[100] !!! INVALID CITATION !!! .

[101] B. Kristiansson and B. Lennartson, "Robust tuning of PI and PID controllers: using derivative action despite sensor noise," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 55-69, 2006.

[102] B. T. Krishna and M. Janarthanan, "Realization of fractional order lowpass filter using different approximation techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3552-3561, 2023.

[103] Y. Luo and Y. Chen, "Fractional-order [proportional derivative] controller for robust motion control: Tuning procedure and validation," in *2009 American Control Conference*, 2009: IEEE, pp. 1412-1417.

[104] A. Halim, "Evaluation of Approximation And Reduction Method for Fractional Order Transfer Function," in *2021 17th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering*, 2021: IEEE, pp. 5-10.

[105] A. Tepljakov and A. Tepljakov, "FOMCON: fractional-order modeling and control toolbox," *Fractional-order modeling and control of dynamic systems*, pp. 107-129, 2017.

[106] F. G. Martins, "Tuning PID controllers using the ITAE criterion," *International Journal of Engineering Education*, vol. 21, no. 5, p. 867, 2005.

[107] T. A. Fagundes, G. H. F. Fuzato, P. G. B. Ferreira, M. Biczkowski, and R. Q. Machado, "Fuzzy controller for energy management and soc equalization in dc microgrids powered by fuel cell and energy storage units," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 3, no. 1, pp. 90-100, 2021.

[108] C. Boutahiri, A. Nouaiti, A. Bouazi, and A. M. Hsaini, "Experimental Test of a Three-Phase Inverter Using a Launchpad TMS320F28379D Card," in *International Conference on Electronic Engineering and Renewable Energy Systems*, 2022: Springer, pp. 451-459.

[109] S. Muthukumari, S. Kanagalakshmi, and S. K. TK, "Development of a novel model matching decentralized controller design algorithm and its experimental validation through load frequency controller implementation in restructured power system using TMS320F28379D controlCARD," *ISA transactions*, vol. 148, pp. 285-306, 2024.

[110] X. Zhang, T. Zhao, and X. Gui, "Force Ripple Suppression of Permanent Magnet Linear Synchronous Motor Based on Fuzzy Adaptive Kalman Filter," in *2021 13th International Symposium on Linear Drives for Industry Applications (LDIA)*, 2021: IEEE, pp. 1-5.

[111] F. Issi and O. Kaplan, "An Experimental and Simulation Study of Load Matching in Class-E Inverters using Adaptive Neuro-Fuzzy Controller," *Electric Power Components and Systems*, vol. 51, no. 11, pp. 1040-1050, 2023.

# Appendix A

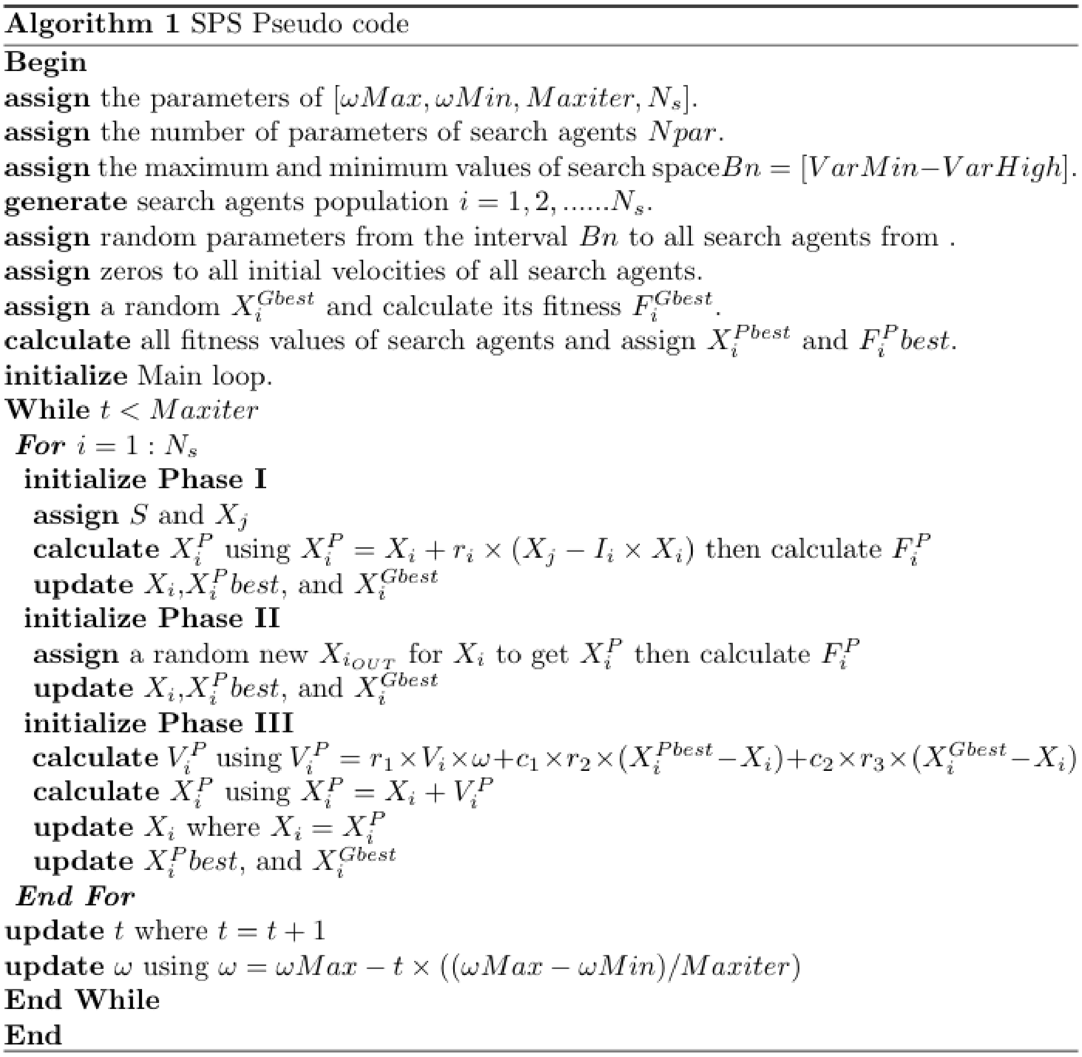
## PMSM parameters

**Table 8 PMSM parameters**

| $Parameter$ | $Value$ |
|---|---|
| $Nominal\ power(Kw)$ | 1.5 |
| $P$ | 3 |
| $B.(Nm/\ rad/s\ )$ | 0.00038 |
| $Lq.(H)$ | 0.0058 |
| $Ld.(H)$ | 0.0066 |
| $Flux.(permanant\ magnet\ flux\ linkage).(Wb)$ | 0.1546 |
| $R.(ohm)$ | 1.4 |
| $J.(kg-m\text{^}2)$ | 0.00176 |

# Appendix B

## SPS algorithm



**Algorithm 1** SPS Pseudo code

```
Begin
assign the parameters of $[\omega Max, \omega Min, Maxiter, N_s]$.
assign the number of parameters of search agents $Npar$.
assign the maximum and minimum values of search space$Bn = [VarMin{-}VarHigh]$.
generate search agents population $i = 1, 2, ......N_s$.
assign random parameters from the interval $Bn$ to all search agents from .
assign zeros to all initial velocities of all search agents.
assign a random $X_i^{Gbest}$ and calculate its fitness $F_i^{Gbest}$.
calculate all fitness values of search agents and assign $X_i^{Pbest}$ and $F_i^P best$.
initialize Main loop.
While $t < Maxiter$
 For $i = 1 : N_s$
  initialize Phase I
   assign $S$ and $X_j$
   calculate $X_i^P$ using $X_i^P = X_i + r_i \times (X_j - I_i \times X_i)$ then calculate $F_i^P$
   update $X_i$,$X_i^P best$, and $X_i^{Gbest}$
  initialize Phase II
   assign a random new $X_{i_{OUT}}$ for $X_i$ to get $X_i^P$ then calculate $F_i^P$
   update $X_i$,$X_i^P best$, and $X_i^{Gbest}$
  initialize Phase III
   calculate $V_i^P$ using $V_i^P = r_1 \times V_i \times \omega + c_1 \times r_2 \times (X_i^{Pbest} - X_i) + c_2 \times r_3 \times (X_i^{Gbest} - X_i)$
   calculate $X_i^P$ using $X_i^P = X_i + V_i^P$
   update $X_i$ where $X_i = X_i^P$
   update $X_i^P best$, and $X_i^{Gbest}$
 End For
update $t$ where $t = t + 1$
update $\omega$ using $\omega = \omega Max - t \times ((\omega Max - \omega Min)/Maxiter)$
End While
End
```