TECNOLÓGICO
DE MONTERREY

Campus Ciudad de México
Escuela de Graduados de Ingeniería y Arquitectura

Master of Sciences in Engineering
Control and Automation Systems

Intelligent Speed Control of Induction Motors using Direct Torque Control

Author:
BEng(Electronics) Fernando David Ramírez Figueroa

Thesis Advisor:
Prof. PhD. Alfredo Victor Mantilla Caeiros

June 2011

Supported by:
Cátedra de Investigación en Diseño de Aplicaciones Mecatrónicas

Abstract

This thesis present a novel adaptive fuzzy controller for the regulation of speed on induction machines with direct torque control. A state-of-the-art study is presented on induction machine drives involving the use of intelligent control techniques. PID control with decoupled gains is used to regulate the speed of the machine. Genetic algorithm and genetic programming techniques are used for several offline optimizations and later FCM is introduced for online optimization on the limits of fuzzy membership functions. Finally simulations on Simulink and LabVIEW are presented, results and conclusions are discussed.

List of Figures

List of Tables

Table of Contents

1. Introduction

Induction machines are among, if not the most, widely used of all the electric motors. Because they are so widely used, they are worth understanding. They are simple to build, rugged, and offer a reasonable asynchronous performance: the torque-speed curve is manageable, have a stable operation under load and their efficiency is almost always satisfactory.

In addition to their current economic importance, induction machines may find applications in new designs not similar to the ones currently found. An example could be very high speed motors for gas compressors, with squirrel cage or with solid iron rotors. The following -Table 1- shows the use of induction motors in different applications.

Table 1. Induction Motor Applications [46].

Constant Torque	Variable Torque
Steel mills machine	Centrifugal pumps
Paper machine: Winder, Tension reels, Mill stands	Centrifugal fans
Cement mills, rubber mills	Compressors
Mixers and Crushers	
Conveyors	
Cranes	
Elevators	

An induction machine is a type of asynchronous machine powered by alternating current where power is supplied to the rotating device by means of electromagnetic induction. An electric motor converts electrical to mechanical power in its rotating part called –rotor. There are several ways to supply power to rotors, in a direct current machine the power is supplied to the armature directly from the source; while in an induction motor the power is *induced* into the rotating device.

The induction motor could be described by stator and rotor as the main parts; Figure 1 shows a simple figure of an induction machine, as it can be seen, rotor and stator are coaxial. The stator has a polyphase winding slots while rotor can have either a winding or a cage also in slots.

A controller is a device that observes and regulates the operational conditions of a system. Operational conditions or most commonly referred as the output variables, can be affected by changing the input variables. Thus, a motor controller is a device (or a group of devices) which purpose is to regulate or control in some predetermined behavior the performance of an electric motor [62].

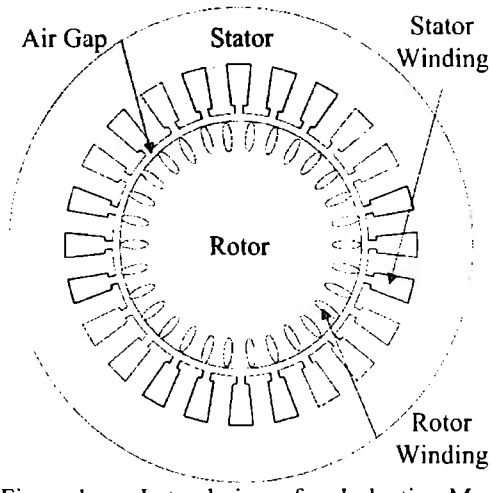


Figure 1. Lateral view of an Induction Machine.

Drives with high performance that control instantaneous electromagnetic torque for induction motors and other alternating current motors, have been in use for several decades. Thanks to the pioneers on vector control like Blaschke and others [1], [28] and [38] vector control based drives have become a standard on industry.

Direct torque control was developed in the 1980's by Depenbrok [13], Takahashi and Noguchi [64] and the most significant industrial contribution to direct torque controlled drives has been made by Siemens however at present only ABB has a commercially available direct torque controlled drive [67].

Artificial intelligence -AI- based control techniques have been very popular since the decade of the 1990's given that most of these techniques do not need any model to be designed; however very few industrial applications use them, the ability of research to impact products does not depends on finding the best solution, but on finding the right problem and solve it in a commercial way [10].

As researchers, we should recognize capabilities by new techniques and theories and then point their applications to specific market segments where there are critical needs for those capabilities [10]. Despite the fact that research has been a very populated brand of research for more than 60 years, the incredible complexity of human behavior makes applications as *Hal 9000 in: 2001 A Space Odyssey* [36] still far from reality [30], [31].

AI is here, just not as we have expected; todays AI does not tries to recreate the brain. Instead it uses all kinds of computer sciences techniques, sensors and sophisticated algorithm to master specific tasks. Google [23] search uses AI to interpret human queries; automated robots in warehouses deliver items to humans [39]. AI is not just hype, as many real working applications use it [32], [39]; though we still have to develop systems that span the full spectrum of intelligent behavior.

Intelligent control methods can be classified in two main categories [68]: fully based intelligent control regulators and classical or advanced regulators supervised by artificial intelligence based functions. For the first category the fuzzy controllers developed by Mamdani and Takagi-Sugeno are the most popular.

Other approaches consist in the use of Artificial Neural Networks to replace classical regulators. Neuro-Fuzzy control systems combine the strengths of both systems and overcome the weaknesses of them when used alone. Since the response of these controllers cannot be considered as optimal, Genetic algorithms can be used to supervise and optimize the response of them [21], [54], [56] and [68].

Finally, modern control techniques are often based on the elimination or the reduction of number of sensors in applications. In this way, hardware sensors are replaced by software based on parameter identification, estimation, observation and/or signal injection [6].

1.1. Vector Control Systems

Although there are several kinds of induction motor control, vector control -VC- and direct torque control -DTC- are the most widely used. VC drives where introduced in Germany by Blaschke [4], Hasse [28] and Leohhard [38] in 1972 which have achieved a high level of maturity and have become increasingly popular on a wide range of applications. DTC was introduced and in Japan by Takahashi [64], [65] and in Germany by Depenbrock [13] in 1984 and 1985 respectively, however only one industrial company makes drives with this technology.

Other types of induction machine control are: Individual Channel Design, a novel framework that allows analysis and synthesis of multivariable control systems by applying classical techniques based on Bode/Nyquist plots. Such controlling system generates the appropriate reference voltage signals for the inverter via a space-vector which in turn, provides the voltage signals for the terminals of an induction motor [40].

Limitations of field oriented controls such as DTC and VC at low speed operations in sensorless applications are due to parameter variations, integral drift and noise. A wide range of nonlinear methods for feedback control, state estimation and parameter identification has emerged. Among them, sliding-mode control [59] gained wide acceptance due to the use of straightforward fixed nonlinear feedback control functions, which operate effectively over a specified magnitude range of system parameter variations and disturbances.

The essential property of the sliding-mode control is that the discontinuous feedback control switches on one or more manifolds in the state space. Ideally, the switching of control occurs at infinitely high frequency to eliminate the deviations from sliding manifolds. In practice, the switching frequency is not infinitely high due to the finite switching time of power electronics and combined with the effects of un-modeled dynamics causes undesired chattering of the control [58].

1.1.1. Induction Machine Control

Induction machines -IM- are considered the workhorse of the industry because they are simple and rugged. However the control structure of an induction motor is complicated given that the stator field is revolving. Further complications arise given that the rotor currents and flux cannot be directly monitored on the squirrel-cage induction machine.

Torque is similarly produced in direct current (d.c.) and alternating current (a.c.) machines. However this was not discovered before 1970, which is why vector controlling techniques where not early proposed. It was first implied that in order to monitor instantaneous electromagnetic torque -EMT- of an induction machine, it was necessary to monitor rotor currents and position. Later, by using space-vector theory it was proven that EMT of an IM can also be expressed as the product of flux-producing and torque-producing currents as in d.c. machines [67].

If flux oriented reference frame is employed, stator current components are transformed to a new rotating frame, which rotates together with a selected flux-linkage space vector. There are three main possibilities for the selection of the flux-linkage vector, which can either be: stator-flux-linkage, rotor-flux-linkage and magnetizing-flux-linkage vector. Hence, the terminology for the control name is derived: stator-flux, rotor-flux and magnetizing-flux-oriented control.

Consequently the instantaneous EMT can be expressed with the following equation: $t_e = c_1 \left|\bar{\psi}_f\right| i_{sy}^f$ where for linear magnetic conditions, c_1 is a constant, $\left|\bar{\psi}_f\right|$ is the modulus of the flux-linkage space vector that depends on the frame of reference, i_{sy}^f is the torque producing stator current that also depends on the flux oriented reference frame. We can see that it is similar to the equation of a separately exited d.c. machine which is $t_e = c i_f i_a = c_1 \psi_f i_a$.

1.1.2. Basics of DTC Drives

In a DTC drive flux linkage and electromagnetic torque are controlled directly and independently by selecting the optimum mode for the inverter. This selection is made to maintain the flux and EMT errors within their respective hysteresis bands. It is also made to obtain a fast torque response, low inverter switching frequency and low harmonics. The optimal selection table for the inverter can be obtained by making physical considerations of the stator flux space vector, the available switching vectors and the required torque and flux references.

Figure 2 shows the diagram of a DTC for induction motors using stator flux, other forms are possible (rotor or magnetizing flux) but this is one of the most common. The IM obtains energy from a voltage source inverter -VSI- the stator flux and EMT are restricted with their corresponding hysteresis bands, two and three levels respectively. The outputs of the comparators are used by the inverter switching table, which also uses information of the sector in which the stator flux space vector is located.

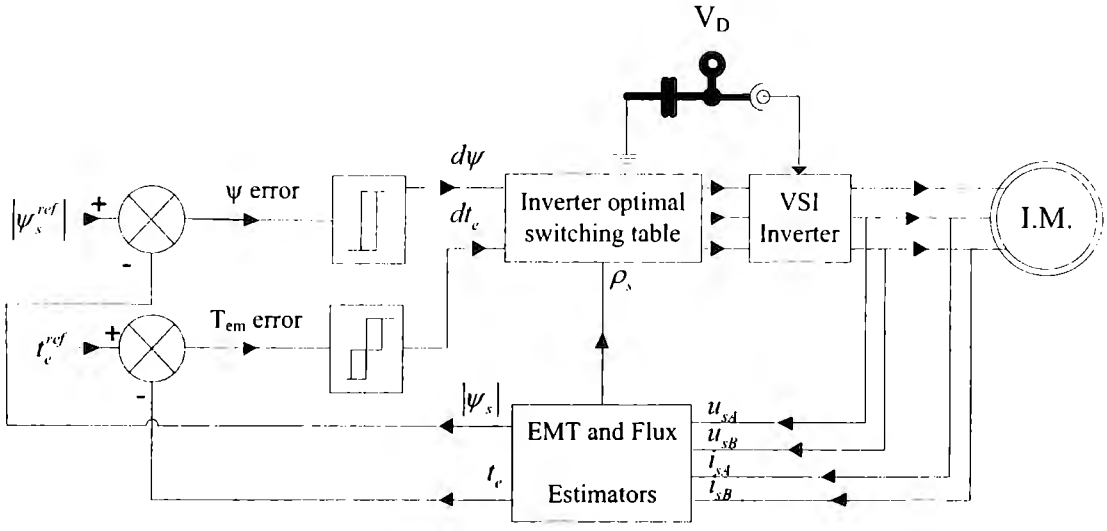


Figure 2. Diagram of stator based DTC for induction machine drive.

Estimators for EMT and flux are required; the latter can be obtained similarly as in vector control: by integrating the terminal voltages. Although at low frequencies large errors are obtained due to variations in stator resistance, noise and drift. This is one of the reasons to use other types of estimation systems like observers.

Observers tolerate parameter variations and their accuracy can be increased using online parameter estimators. It is also possible to use a joint state and parameter observers to calculate values like stator flux, stator resistance and rotor speed among others. For stator flux it is not necessary to monitor stator voltages since they can be reconstructed by reading d.c. link voltage and knowing the state of the inverter switching table.

Table 2. Advantages and Disadvantages of Direct Torque Control.

Advantages	**Disadvantages**
No need of additional PWM technique	Zero and low speed operation problems
No need of coordinate transformations	Possible problems during changes in torque
No need of voltage decoupling circuits	Required flux estimation
The number of controllers is reduced	Required EMT estimation
Precise flux vector position is not needed	Variable switching frequency
Stator flux and electromagnetic torque are the controlled variables	Variable switching frequency on the inverter
Pulse Width Modulation is unnecessary	Short control cycle $(<25\mu s)$ is needed for its digital implementation
Dynamic performance is excellent	High harmonic content in phase currents that increments with high speeds and partial loads
Torque is linear	
High performance torque control at low stator frequencies	

Predictors are also used to estimate the vector on the inverter. Advantages and disadvantages of DTC can be seen in Table 2. The main features of DTC are listed below:

1. Flux and EMT can be controlled independently and directly.
2. Stator currents and voltages are controlled indirectly.
3. The forms of stator fluxes and stator currents have an approximate sinusoidal form.
4. Has excellent torque dynamic response and reduced oscillations.
5. The inverter switching frequency depends on the flux and torque hysteresis bands.

1.1.3. Sensorless Drives

Reducing hardware complexity, costs, increasing mechanical robustness, reliability and increased noise immunity can be achieved by eliminating sensors in a.c. drives. Furthermore, additional advantages can be obtained like elimination of electromechanical sensors, which increment system maintenance requirements and inertia. In small sized motors it is impossible to use EMT sensors, also in low powers drives the cost of the sensor could be equal or higher than the cost of the drive and motor.

Drives that operate in hostile environments or high speed applications cannot use speed sensors. Estimation of speed and position can be achieved by using software based estimators (based on stator currents and voltages measured) due to the fact that real time processors continue to decrease in cost an size. In summary, Table 3 contains the main objectives of a sensorless drive and Table 4 contains the main techniques of sensorless control for induction machines.

It can be foreseen that sensorless drives with DTC and intelligent control will soon emerge in the form of integrated drives. The development of micro and power electronics, in conjunction with digital signal processing and intelligent control are topics with great research potential, that will introduce revolutionary changes in the drive industry.

Table 3. Advantages and Disadvantages of Sensorelss Drives.

Reduced hardware complexity and costs
Overall ruggedness, increased mechanical robustness
Can operate in hostile environments
High reliability
Decreased or null maintenance requirements
Immunity to noise
Not affected by machine inertia

Table 4. Main techniques of sensorless control for IM.

Estimator using third harmonic voltages
Estimator using saliency effects
Model reference adaptive systems (MRAS)
Observers (Leunberger, Kalman)
Estimators using Intelligent Control (ANN, Fuzzy Logic, Neuro Fuzzy Systems)

1.2. State of the Art

This work proposes a direct torque control scheme for induction machines, however it is not the first and only one proposed. DTC was originally proposed by Depenbrock [13], Takahashi and Noguchi [64] in 1984. Through the pass of time, numerous contributions

have been made and several variations have emerged. In order to contrast the relevance of this thesis, recent applications and related research in the fields are discussed below. A diagram of the classification of induction motor control methods is shown in Figure 3 (based on the original in a paper by Buja [6]).

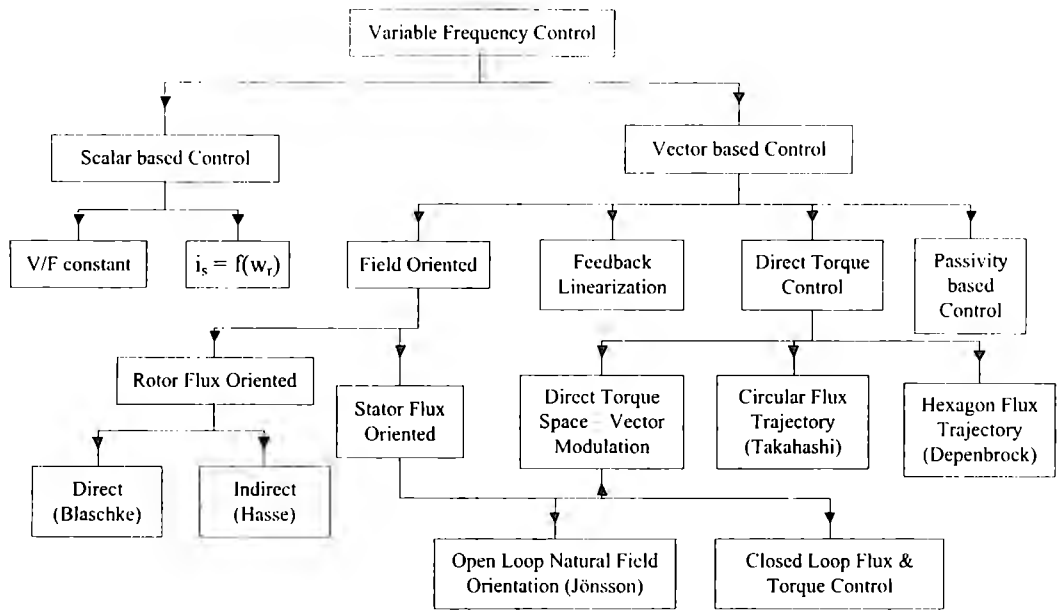


Figure 3. Classification of Induction Motor Control Methods.

1.2.1. Conventional DTC Contributions

Recent contributions to the field include a combination of DTC and space vector modulation –SVM, which produces better quality in steady state performance and in a wide speed range. It also reduces torque ripple at low speeds, which is an inconvenience of conventional DTC.

This is the case of the research conducted by Zhifeng Zhang et al. [74] which propose a sensorless DTC-SVM scheme for IM, with a full order speed adaptor stator flux observer and a speed-adaptive law. The observer gain matrix is based on state feedback H_∞ control theory [60]. They guarantee the stability and robustness of the observer even at very low speeds, but so far only simulations results have been presented.

Zaid S. A. and his colleagues at the University of Cairo conduct research on an improved DTC algorithm for IM drives [70]. They consider two main drawbacks of conventional DTC: the first one is the variation of switching frequency according to the amplitude of hysteresis bands and motor operating speed; the second one is that the selection of voltage vectors is not optimized for fast torque response. Their study proposes a new DTC scheme which enables fast torque response and constant switching frequency, with simulation and hardware implementation results.

Efficiency optimization on IM drives is a major subject of research based on their extensive use in industry. Masood Hajian and his team at the Isfahan University of

Technology on Iran, conduct research on an adaptive nonlinear sensorless DTC drive with energy optimization [42]. They present a novel model based on minimization loss approach, combined with an improved search based method for efficiency optimization and backstepping DTC control for IM drives. They claim that the proposed controller has fast tracking capabilities of rotor flux and electromagnetic torque.

On predictive direct torque control for a.c. machines professor Pacas and his team at the University of Siegen, Germany conducts research since several years ago. The PDTC algorithm [16], [45], [48] calculates the switching times of an active space phasor one step in advance, such that in steady state the torque time area under and over the torque reference has the same size.

This way switching frequency can be kept constant and sampling frequency does not have to be as high as in the digital implementation of conventional DTC. Their work also includes sensorless techniques [45]. As the one for synchronous reluctance machines with rotor saliency, that produces reluctance torque casing motor inductances dependent of the shaft position and providing a way to sense rotor position.

Their work also includes elimination of the rotor sensor at low and zero speed based on a test voltage signal injection technique, which allows calculating rotor position signals at very low speeds, including zero speed [45]. Their work also includes fault tolerant control systems that detect broken bars in the rotor and other types of failures. These are detected by the appearance of harmonics in some machine variables, as well as an increment of the losses that can lead to complete damaging of the machine [16], [66].

Predictive DTC is a very extensive line of research at this time, and there are many contributions on the field as the one of Jef Beerten at K. U. Leuven in Belgium. He proposes a predictive algorithm [3] that can be easily implemented as an extension of the basic DTC. It does not requires additional motor parameters, only previously calculated values of electromagnetic torque and flux and reduces torque and flux ripples.

More recently predictive DTC is including model predictive control (MPC) approaches to improve the performance on drives. MPC is inherently computational demanding, as an underlying optimization problem needs to be solved. Tobias Geyer and others at the University of Auckland in New Zeland propose a MPDTC scheme [22], which claims that the major benefits are superior performance in terms of switching frequency.

This control scheme reduces the switching frequency by an average of 25% reaching up to 50% compared to ABB industrial drive ACS6000, while better respecting torque and flux hysteresis bounds [22]. Some of the key features of the control system are: extrapolation, computations performed online, admissible switching sequence and the use of nonlinear model prediction.

1.2.2. Intelligent Control with DTC Contributions

As it can be seen in section 1.2.1 vast research in the field of DTC is been done, including predictive models, nonlinear control approaches and sensorless control, among many others. But what about the contributions that include intelligent control techniques? Below, some of the most novel proposals and solutions are discussed.

A literature survey in [68] indicates that applications of intelligent control to electric drives mainly discuss speed or position controllers based in fuzzy logic or neural networks. Although these are important applications, there are many more possibilities for a much wider range of applications based on intelligent control, as shown in Table 5.

There have been numerous contributions to DTC control aided by intelligent control; such as the direct torque neuro fuzzy control for induction motor drives [25], proposed by Grabowski at the University of Warsaw in 1998. They propose two discrete DTC schemes based on neuro-fuzzy control, the first a neuro-fuzzy switching time calculator (STCC) and the second an incremental neuro-fuzzy controller with space vector modulator. The first method reduces sampling time with the same results as classical DTC; the second reduces sampling time and the switching frequency is constant.

Other applications include the use of a neural network for speed estimation of an induction motor using DTC [52] by Pedro Ponce in 2001. A small structured neural network is used as speed estimator of an induction machine using DTC, which leads to short off-line training and processing times. It has the advantage of not increasing the complexity of the whole control system.

Table 5. Applications of Intelligent Control on Variable Speed Drives (a.c. and d.c).

Replace classical controllers (speed, position and others) for intelligent controllers
New and combined control structures based on conventional and intelligent control like: intelligent control universal drives, intelligent control electronic motors
Firing signal generation schemes and new switching vector schemes
Compensation of nonlinear effects in discontinuous operation of the converter
Parameter estimation (improved observers, harmonics, efficiency among many others)
Self-commissioning systems
Virtual sensors based on intelligent control techniques
Condition monitoring and diagnosis
Machine design
Models, including steady state and transient states
Self-repairing and self-constructing controllers
Fault detection and localization

Later in 2004 Cheng-Zhi proposed a DTC scheme optimized by fuzzy neural networks [9] making emphasis at low speeds, where the problems are more pronounced. The fuzzy neural network is optimized by a genetic algorithm to avoid local minima. They claim that it can infer stator voltage vectors reasonably good, its control is easy and dynamic response of torque and rotating speed is fast; it also improves the low speed performances of direct toque system.

In more recent times Rajaji and others at the University of Sathyabama at the India presented an ANFIS model of an induction motor drive used for modeling, estimation and control [56]. They estimate up to eleven parameters of a non-linear model and can be used for on-line compute of the model on drives or as an estimator. There are other records of similar works that are based on fuzzy observers for the stator resistance, such as [33].

At the University of Shenyang Ligong at China Hongkui Li is working on a Fuzzy DTC scheme [29] that optimizes the stator flux reference, which also uses a fuzzy logic controller to select voltage vectors. The inputs to the fuzzy controller are error of EMT and error on stator flux and the output is the corresponding voltage vector.

Reduction of torque ripple is achieved in the whole speed range, simulations are presented. His work and of his team also includes other development of flux observers for sensorless DTC schemes that uses a dynamic neural network to design the observer gain matrix, which can be adjusted online [70].

At the University of Science and Technology in Qindao China, Meie Sui and his team are working on a sensorless direct space vector modulation DTC for induction motors, based on fuzzy control and model reference adaptive control [44]. They propose an adaptive law based on fuzzy logic which adjusts the gain on the model and estimates speed. They claim the results are superior to classical estimators.

As of industrial applications, currently Hitachi has a drive in the market, the -J300- which incorporates fuzzy logic. Hitachi claims it to be the first one using it. The main features of the J300 are sensorless vector control, efficiency and auto-tuning, among others. According to Hitachi, the drive intelligent inverter takes into account the characteristics of motor and the system and ensures accurate torque control through an entire range of frequency.

The J300 also includes an acceleration and deceleration fuzzy control. There is also a drive by Yaskawa, the VS-616G5, a general purpose inverter which contains flux vector control for induction machines. According to Yaskawa, it contains a magnetic flux observer that is based on intelligent neuro-control.

1.3. Motivations

In the past, direct current motors were extensively used in applications where variable speed was necessary. D.C. motors can be controlled very easily by varying field and armature currents and controlling their flux and torque. However d.c. motors require frequent maintenance because of the existence of brushes and commutator, they also have limited capabilities for high speed and high voltage operation conditions.

The disadvantages presented on d.c. machines can be easily overcome by alternating current machines, which have simple and rugged structures. They also eliminate the need of maintenance, are robust and immune to heavy overload and are of smaller dimensions when compared to d.c. motors. Variable speed a.c. drives can be used to replace d.c. motors, because of their commutator limits, or when needed in hazardous environments. It is

expected that with the rapid development of micro and power electronics, torque control for different kinds of a.c. machines will become a commonly used technique.

Direct Torque Control -DTC- has an excellent dynamic behavior; compared to Vector Control, the DTC algorithm is less complex and is more tolerant to parameters variations. For these reasons DTC, has become attractive to industrial companies that develop alternating current drives. Conventional DTC has a simple structure and is easily implementable; drives that use DTC have exceptional dynamic performance [1].

With the most recent contributions to digital processing and microelectronics, induction motor drives have reached the status of modern technology. Every time more complex and robust algorithms are developed and can be implemented thanks to the power processing capabilities of new technology.

Intelligent control techniques emulate characteristics of biological systems and novel capabilities. These control techniques can provide products with the important competitive edge that companies seek. Although several applications are described in literature, few become laboratory prototypes and only a handful becomes final products. Thus, the ability of research work to impact products is not centered on finding the best solution, but in finding the correct problem and solving it in a commercial way [10].

The application of intelligent based control methods has had the greatest impact on the aerospace industry, but it is most likely that this tendency will change in the near future. In many cases, the application of intelligent control to drive systems leads to significantly reduced development times. It also helps avoid classical control theory limitations such as: complex mathematical models, relying on simplifications and assumptions, may contain parameters difficult to measure and that may change significantly during the operation; all of these can make impossible to determine a mathematical model.

Even further, classical control theories suffer limitations of form, due to the nature of the controlled system (linearity, time-variance, etc.) These problems can be easily solved by intelligent control. IC techniques can be used even when analytical models are not known. They are also less sensitive to parameter variations making them more robust than classical control systems. Finally Table 6 presents the key advantages of intelligent controllers and estimators [68].

Thanks to the widespread of digital signal processors, the development of more advanced drives has become possible. Given that the greatest advancements of variable-speed drives have taken place in parallel with the advances of semiconductor devices. Variable speed drives are the most important commercial application of modern power electronics. The revenue generated by electronic industry in 2010 in the world was of $1200 billion euros and it is expected to grow 1298 billion euros [14]. Table 7 contains the market of fuzzy logic applications.

Table 6. Key advantages of artificial intelligence controllers and estimators.

Mathematical model of the plant not required for their design
When properly tuned, they can lead to improved performance
Can be designed based on linguistic information from human experts
Can be designed on the basis of response data in the absence of expert knowledge
Can generalize information, becoming independent of particular characteristics of the systems
Can adapt to new information as it becomes available
Can provide solutions to control problems impossible to conventional control
Have good noise rejection capabilities
Are inexpensive to implement
Are easy to extend and modify

Table 7. Market of fuzzy logic applications.

1989	*World*	5 USD million	*USA, Canada*	1 USD million		
1993	*World*	20 USD million	*USA, Canada*	10 USD million		
1995	*Europe*	1 USD billion	*USA*	800 USD million	*Japan*	2 USD billion
2000	*Europe*	7 USD billion	*USA*	3 USD billion	*Japan*	8 USD billion

1.4. Objectives

The purpose of this work is to research, develop and implement a direct torque control for three phase induction motors, control rotor speed and improve the control system using intelligent control techniques. As it has presented in section 1.2, most of the research and applications has been done on artificial neural networks and fuzzy logic in the specific case of motor control. The purpose of this work is to apply artificial intelligence techniques not previously done to motor control, like evolutionary algorithms and clustering techniques.

1.4.1. Specific Objectives

More specific objectives are:

- Develop conventional Direct Torque Control.
- Create speed controllers based on intelligent control techniques.
- Enhance the response of the speed controllers.
- Tests and results on simulations and experimental prototype.

1.5. Outline of the Chapters

This thesis is organized as follows: Chapter 1 is a brief description on the state of the art and the objectives of this work. Chapter 2 describes the theoretical foundations about vector control and more specific information about DTC and other conventional control techniques. Chapter 3 describes the theory of intelligent control used for the development of this work. Chapter 4 is the description of the proposed and developed system, the desired controlling schemes, proposed algorithms and the implementation process. Chapter 5 describes tests and results; finally, Chapter 6 presents conclusions and future work.

2. Theoretical Foundations

The theoretical foundations used in this thesis are presented in this chapter. The principles of various forms of Direct Torque Control systems can be understood by comparing the production of electromagnetic torque in d.c. and a.c. machines; for these purposes space phasors must be understood first.

With the help of them it is possible to formulate space-phasor model of a.c. machines and in our case, induction machines. Later, the model of the Induction Machine and its equations are explained, the theory of conventional Direct Torque Control is presented and finally conventional PID control theory is developed in this chapter.

2.1. The Space Phasor

In order to correctly understand and design torque control systems, it is necessary a model with dynamic response of the machine to be controlled. The model must preferably incorporate all important dynamic effects occurring during steady state and transient operation. It should also be valid for arbitrary time variations of voltages and currents supplied by the converter that feeds the machine.

Such models can be obtained by the utilization of space phasor theory, which is closely related to two axis theory of electrical machines. The space phasor has some advantages such as being simple and compact; leading to very clear physical pictures of the modeled machines.

With the aid of space phasor theory, a three phase system in a given arbitrary frame of reference being described by phase quantities (currents, voltages, flux linkages) can be replaced by a resulting space phasor of their respective quantity. It is also possible to introduce the space phasor by using the two axis theory; originally introduced by Park.

2.1.1. Equations of Transformation of Two-axis Theory

In late 1920s [49] Park introduced a new approach to machine analysis where he formulated a change of variables. This approach replaced variables associated with the stator winding of the synchronous machine for virtual windings rotating on par with the rotor. As a result, he transformed stator variables to a frame of reference fixed to the rotor. These transformations eliminate time varying inductances from the voltage equations of the machine which occur due to: electric circuits in relative motion and with varying magnetic reluctances.

2.1.1.1. Transformation to Stationary Frame of Reference

For simplification purposes a change of variables that transform the three-phase variables of stationary circuit elements to an arbitrary frame of reference can be expressed as follows:

$$f_{sDQ0} = T_s f_{sABC} \qquad \text{eqn (1)}$$

$$\text{where } f_{sDQ0}^{T} = \begin{bmatrix} f_{sD} & f_{sQ} & f_{s0} \end{bmatrix} \qquad \text{eqn (2)}$$

$$f_{sABC}^{T} = \begin{bmatrix} f_{sA} & f_{sB} & f_{sC} \end{bmatrix} \qquad \text{eqn (3)}$$

$$T_s = \frac{2}{3}\begin{bmatrix} \cos\theta & \cos(\theta - 2\pi/3) & \cos(\theta + 2\pi/3) \\ \sin\theta & \sin(\theta - 2\pi/3) & \sin(\theta + 2\pi/3) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \qquad \text{eqn (4)}$$

$$\omega = \frac{d\theta}{dt} \qquad \text{eqn (5)}$$

And for the inverse transformations:

$$T_s^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 1/2 \\ \cos(\theta - 2\pi/3) & \sin(\theta - 2\pi/3) & 1/2 \\ \cos(\theta + 2\pi/3) & \sin(\theta + 2\pi/3) & 1/2 \end{bmatrix} \qquad \text{eqn (6)}$$

$$\theta = \int \omega\, dt \qquad \text{eqn (7)}$$

In the above equations f represent any of the variables in the machine (voltage, current, flux...), the superscript T denotes transposition of the matrix and the subscript s denotes everything associated to the stationary frame of reference. The angular displacement θ must be continuous however; the angular velocity ω associated with the change of variables is unspecified.

The frame of reference may rotate at any angular speed or remain stationary. It is convenient to visualize the transformation equations as trigonometric relationships between variables. As shown in Figure 4, where f_{sD} and f_{sQ} are orthogonal rotating at angular velocity ω and f_{sA}, f_{sB} and f_{sC} are stationary each displaced by 120°. Finally the total instantaneous power expressed in 3 axes is denoted by eqn (8) and in 2 axes by eqn (9).

$$P_{sABC} = v_{sA} i_{sA} + v_{sB} i_{sB} + v_{sC} i_{sC} \qquad \text{eqn (8)}$$

$$P_{sDQ0} = \frac{3}{2}\left(v_{sD} i_{sD} + v_{sQ} i_{sQ} + 2 v_{s0} i_{s0}\right) \qquad \text{eqn (9)}$$

Where: The 3/2 factor is used because of to the constant used in eqn (4). As it can be seen the total power is independent of the frame of reference which is being evaluated.

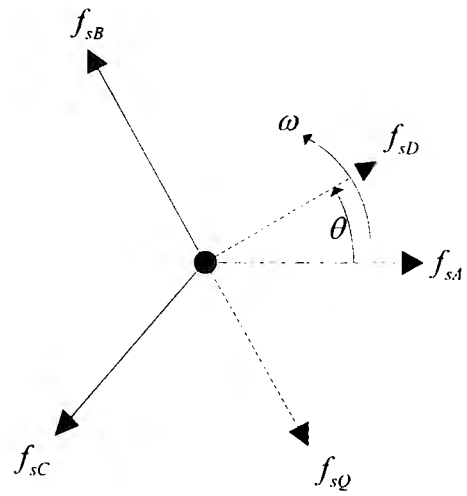


Figure 4. Transformation of Stationary Frame of Reference.

2.1.1.2. Most Common Reference Frames

With the Park transformation, the frame of reference can be arbitrarily set, yet the most commonly used in the analysis of electric machines and power systems are: the *arbitrary*, *stationary*, *rotor* and *synchronous,* more information can be seen in Table 8.

Table 8. Common Frames of Reference.

Name	Speed	Variables	Transform	Characteristics
Arbitrary	ω	f_{sDQ0}	T_s	Variables are referred to an arbitrary frame of reference.
Stationary	0	f^s_{sDQ0}	T^s_s	Variables are referred to the stationary frame of reference.
Rotor	ω_r	f^r_{sDQ0}	T^r_s	Variables are referred to the rotor frame of reference.
Synchronous	ω_e	f^e_{sDQ0}	T^e_s	Variables are referred to the synchronously rotating frame of reference.

2.1.1.3. Transformation Between Reference Frames

Sometimes it is useful to transform from one frame of reference to another, without involving the main variables of transformation. Let x denote the reference frame from which the variables are being transformed, and y denote the reference frame to which the variables are being transformed. Several trigonometric identities are used to reach eqn (13), referred as *vector rotator*:

$$f^y_{sDQ0} = T^y_s f_{sABC} \qquad \text{eqn (10)}$$

$$T^y_s = {}^xT^y T^x_s \qquad \text{eqn (11)}$$

$${}^xT^y = T^y_s \left(T^x_s\right)^{-1} \qquad \text{eqn (12)}$$

$$ {}^{x}T^{y} = \begin{bmatrix} \cos(\theta_y - \theta_x) & -\sin(\theta_y - \theta_x) & 0 \\ \sin(\theta_y - \theta_x) & \cos(\theta_y - \theta_x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{eqn (13)} $$

2.1.2. Space Phasors of Stator

After the space phasor theory being introduced, several space phasors will be introduced, which are necessary for the development and understanding of the alternating current machines with a smooth air gap.

2.1.2.1. Space Phasor of Stator M.M.F.s and Stator Currents

If i_{sA}, i_{sB} and i_{sC} are instantaneous current values in stator phases *A*, *B* and *C*, the stator current space phasor, $\bar{i}_s$ is defined by:

$$ \bar{i}_s = \frac{2}{3}\left[i_{sA} + i_{sB} + i_{sC}\right] = \frac{2}{3}\left[i_{sA} + a i_{sB} + a^2 i_{sC}\right] \qquad \text{eqn (14)} $$

$$ a = e^{j2\pi/3} \qquad \text{eqn (15)} $$

Where: eqn (14) is the complex space phasor of the three phase stator currents in the complex plane, in a stationary reference frame fixed to the stator. eqn (15) shows the spatial operator. The real axis of the stator is denoted by *sD* corresponding to Direct-axis terminology of the stator.

The stator current space vector expressed in the stationary reference frame fixed to the stator can be seen in eqn (16). Depending on place of edition of the book, sometimes the notation is $i_{s\alpha}$ and $i_{s\beta}$ instead of i_{sD} and i_{sQ}. In symmetrical three-phase machines the direct and quadrature axis stator currents i_{sD} and i_{sQ} are related to the actual three phase stator currents:

$$ \bar{i}_s = i_{sD} + j i_{sQ} \qquad \text{eqn (16)} $$

$$ i_{sD} = c\left[i_{sA} - 1/2 i_{sB} - 1/2 i_{sC}\right] \qquad \text{eqn (17)} $$

$$ i_{sQ} = c\sqrt{3}/2\left(i_{sB} - i_{sC}\right) \qquad \text{eqn (18)} $$

Where: c is a constant. For the non-power-invariant form of the transformation, the value of $c = 2/3$, and for the power-invariant form $c = \sqrt{2/3}$. Similarly, the instantaneous zero-sequence current component of the stator current is defined by eqn (19); where for non-power-invariant $k = 1/3$ and for power-invariant $k = 1/\sqrt{3}$.

$$ i_{s0} = k\left[i_{sA} + i_{sB} + i_{sC}\right] \qquad \text{eqn (19)} $$

2.1.2.2. Space Phasor of Rotor M.M.F.s and Rotor Currents

The rotor current phasor is denoted by eqn (20) expressed in the rotor reference frame (real axis denoted by $r\alpha$ and imaginary axis by $r\beta$). The speed of the reference frame is ω_r as in eqn (5). The definition is similar to the stator space phasor in the stationary reference frame.

$$\bar{i}_r = \frac{2}{3}\left[i_{rA} + i_{rB} + i_{rC}\right] = \frac{2}{3}\left[i_{rA} + a i_{rB} + a^2 i_{rC}\right] = \left|\bar{i}_r\right| e^{-j\alpha_r} \qquad \text{eqn (20)}$$

Thus, it is possible to generalize, and it follows the definitions of the space-phasor quantities in their own reference frames. Let $i_{r\alpha}$ and $i_{r\beta}$ be the instantaneous values of the three-phase rotor currents (direct and quadrature axis components) similar to eqn (16) and eqn (17) represented by eqn (21) and eqn (22):

$$i_{r\alpha} = c\left[i_{rA} - 1/2 i_{rB} - 1/2 i_{rC}\right] \qquad \text{eqn (21)}$$

$$i_{r\beta} = c\sqrt{3}/2\left(i_{rB} - i_{rC}\right) \qquad \text{eqn (22)}$$

Where: $c = 2/3$ for the non-power invariant form of the transformation. Then the space phasor of the rotor currents in the rotor reference frame is given by **Error! Reference source not found.**.

$$\bar{i}_r = i_{r\alpha} + j i_{r\beta} \qquad \text{eqn (23)}$$

2.1.3. Flux Linkage Space Phasors

Now the space phasors for flux linkages of stator and rotor will be presented in different frames of reference. It is also possible to define stator flux linkage in terms of the instantaneous values of the flux linkages ψ_s of the three stator windings; it will be defined for several reference frames.

2.1.3.1. Flux Linkage Space Phasor in Stationary Reference Frame

For the stationary reference frame fixed to the stator, the flux linkage can be expressed as in eqn (24) with the instantaneous values of phase variable flux linkages in eqn (25), eqn (26) and eqn (27).

$$\bar{\psi}_s = 2/3\left(\bar{\psi}_{sA} + a\bar{\psi}_{sB} + a^2\bar{\psi}_{sC}\right) \qquad \text{eqn (24)}$$

$$\bar{\psi}_{sA} = \bar{L}_s i_{sA} + \bar{M}_s i_{sB} + \bar{M}_s i_{sC} + \bar{M}_{sr}\cos\theta_r i_{rA} + \bar{M}_{sr}\cos\left(\theta_r + 2\pi/3\right) i_{rB} + \bar{M}_{sr}\cos\left(\theta_r + 4\pi/3\right) i_{rC} \qquad \text{eqn (25)}$$

$$\bar{\psi}_{sB} = \bar{L}_s i_{sB} + \bar{M}_s i_{sA} + \bar{M}_s i_{sC} + \bar{M}_{sr}\cos\theta_r i_{rB} + \bar{M}_{sr}\cos\left(\theta_r + 2\pi/3\right) i_{rC} + \bar{M}_{sr}\cos\left(\theta_r + 4\pi/3\right) i_{rA} \qquad \text{eqn (26)}$$

$$\bar{\psi}_{sC} = \bar{L}_s i_{sC} + \bar{M}_s i_{sB} + \bar{M}_s i_{sA} + \bar{M}_{sr}\cos\theta_r i_{rC} + \bar{M}_{sr}\cos(\theta_r + 2\pi/3) i_{rA} + \bar{M}_{sr}\cos(\theta_r + 4\pi/3) i_{rB} \quad \text{eqn (27)}$$

Where: $\bar{L}_s$ is the self-inductance of the stator phase winding, $\bar{M}_s$ is the mutual inductance between stator windings and $\bar{M}_{sr}$ is the maximum value of stator-rotor mutual inductance. By substituting in eqn (25), eqn (26) and eqn (27) in eqn (24) yields to the following:

$$\bar{\psi}_s = L_s \bar{i}_s + L_m \bar{i}_r' = L_s \bar{i}_s + L_m \bar{i}_r e^{j0r} \quad \text{eqn (28)}$$

Where: $L_s = \bar{L}_s - \bar{M}_s$ is the total three-phase stator inductance and L_m is the three phase magnetizing inductance: $L_m = (3/2)\bar{M}_{sr}$.

Stator flux linkage space phasor gives the modulus and phase angle of the peak of the sinusoidal stator flux distribution on the air-gap. There are two space phasor components: the first $(L_s \bar{i}_s)$ is the self-flux linkage space phasor of stator phases caused by stator currents. The second $(L_m \bar{i}_r')$ is a mutual flux linkage space phasor expressed in stationary reference frame and is caused by rotor currents.

It is important to remember that eqn (28) is general even for nonlinear conditions. We can also represent the stator flux in terms of the ψ_{sD} and ψ_{sQ} components, as shown in eqn (29); which by consequence gives the definition of eqn (30) and eqn (31).

$$\bar{\psi}_s = \bar{\psi}_{sD} + j\bar{\psi}_{sQ} \quad \text{eqn (29)}$$

$$\bar{\psi}_{sD} = L_s i_{sD} + L_m i_{rD} \quad \text{eqn (30)}$$

$$\bar{\psi}_{sQ} = L_s i_{sQ} + L_m i_{rQ} \quad \text{eqn (31)}$$

2.1.3.2. Rotor Flux Space Phasor In Rotating Reference Frame

The space phasor of the rotor flux expressed in its natural reference frame (fixed to the rotor), and rotating speed ω_r is defined by eqn (32), where ψ_{rA}, ψ_{rA} and ψ_{rA} are the instantaneous values of the rotor flux in the rotor phases. In terms of instantaneous values of stator and rotor currents can be expressed as shown in eqn (33), eqn (34) and eqn (35).

$$\bar{\psi}_r = 2/3\left(\bar{\psi}_{rA} + a\bar{\psi}_{rB} + a^2\bar{\psi}_{rC}\right) \quad \text{eqn (32)}$$

$$\bar{\psi}_{rA} = \bar{L}_r i_{rA} + \bar{M}_r i_{rB} + \bar{M}_r i_{rC} + \bar{M}_{sr}\cos\theta_r i_{sA} + \bar{M}_{sr}\cos(\theta_r + 2\pi/3) i_{sC} + \bar{M}_{sr}\cos(\theta_r + 4\pi/3) i_{sB} \quad \text{eqn (33)}$$

$$\bar{\psi}_{rB} = \bar{L}_r i_{rB} + \bar{M}_r i_{rA} + \bar{M}_r i_{rC} + \bar{M}_{sr}\cos\theta_r i_{sB} + \bar{M}_{sr}\cos(\theta_r + 2\pi/3) i_{sA} + \bar{M}_{sr}\cos(\theta_r + 4\pi/3) i_{sC} \quad \text{eqn (34)}$$

$$\bar{\psi}_{rC} = \bar{L}_r i_{rC} + \bar{M}_r i_{rA} + \bar{M}_r i_{rB} + \bar{M}_{sr}\cos\theta_r i_{sC} + \bar{M}_{sr}\cos(\theta_r + 2\pi/3) i_{sB} + \bar{M}_{sr}\cos(\theta_r + 4\pi/3) i_{sA} \quad \text{eqn (35)}$$

Where: $\bar{L}_r$ is the rotor self-inductance of the rotor winding and $\bar{M}_r$ is the mutual inductance between two rotor phases. If in eqn (33), eqn (34) and eqn (35) are substituted in eqn (32) a lot of simplification is obtained as can be seen in eqn (36).

$$\bar{\psi}_r = L_r \bar{i}_r + L_m \bar{i}_s' \qquad \text{eqn (36)}$$

Where: $L_r = \bar{L}_r - \bar{M}_r$ is the rotor inductance (three phase) and $\bar{i}_s'$ is the space phasor of the stator currents referenced to the rotor.

The two terms presented in eqn (36) contain the space phasor $\left(L_r \bar{i}\right)$ which is the rotor self-flux space phasor (exclusively given by rotor currents) and; the space phasor $\left(L_m \bar{i}_s'\right)$ which is the mutual flux space phasor in the stator reference frame produced by stator currents. Defining the rotor flux in the two axis components $\psi_{r\alpha\beta}$, as in eqn (37) and defining the rotor flux components as eqn (38) and eqn (39).

$$\bar{\psi}_r = \psi_{r\alpha} + j\psi_{r\beta} \qquad \text{eqn (37)}$$

$$\bar{\psi}_{r\alpha} = L_r i_{r\alpha} + L_m i_{sD} \qquad \text{eqn (38)}$$

$$\bar{\psi}_{r\beta} = L_r i_{r\beta} + L_m i_{sQ} \qquad \text{eqn (39)}$$

2.1.4. Space Phasors of Voltages

Space phasors for stator and rotor voltages can be defined similarly to current and flux phasors. The stator voltage phasor in the stationary reference frame is defined in eqn (40); while the rotor voltage space phasor in the reference frame in the rotor is in eqn (41).

$$\bar{u}_s = 2/3\left[\bar{u}_{sA} + a\bar{u}_{sB} + a^2\bar{u}_{sC}\right] = u_{sD} + ju_{sQ} \qquad \text{eqn (40)}$$

$$\bar{u}_r = 2/3\left[\bar{u}_{rA} + a\bar{u}_{rB} + a^2\bar{u}_{rC}\right] = u_{r\alpha} + ju_{r\beta} \qquad \text{eqn (41)}$$

Where: all the $\bar{u}_s$ and $\bar{u}_r$ are the instantaneous values of stator and rotor voltages and $\bar{u}_{sDQ}$ and $\bar{u}_{r\alpha\beta}$ are the corresponding direct-quadrature axis components. The relationship between three phase and quadrature phase voltages is shown in eqn (42) in matrix form.

$$\begin{bmatrix} u_{sD} \\ u_{sQ} \\ u_{s0} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}\begin{bmatrix} u_{sA} \\ u_{sB} \\ u_{sC} \end{bmatrix} \qquad \text{eqn (42)}$$

2.1.5. Electromagnetic Torque Generation in A.C. Machines

In general, in a symmetrical smooth air gap a.c. machine the generation of electromagnetic instantaneous torque can be expressed by eqn (43) the cross product of flux linkage and current phasors, which similar to the d.c. machine equation. It can also be written as in eqn (44) using Euler forms of the vectors.

$$T_{em} = c\bar{\psi}_s \times \bar{i}_r' \qquad \text{eqn (43)}$$

$$T_{em} = c\left|\bar{\psi}_s\right|\left|\bar{i}_r\right|\sin\gamma \qquad \text{eqn (44)}$$

$$T_{em} = c\psi_s i_r \qquad \text{eqn (45)}$$

Where: $\left|\bar{\psi}_s\right|$ and $\left|\bar{i}_r\right|$ are the magnitude of the stator flux and rotor current space phasors. γ is the torque angle, when $\gamma = 90°$ the maximum torque is obtained as in eqn (45).

To derive the Torque equation of the induction machine the energy and work will be analyzed; energy stored in a coupling field may be written as in eqn (46). Energy stored in leakage inductances does not form part of the energy stored in the coupling field; magnetical linearity is assumed in the machine so the energy field W_f is equal to the coenergy W_c.

The change of mechanical energy to a rotational system is described in eqn (47), here T_{em} is the positive electromagnetic torque for motor action and θ_{rm} is the angular displacement of the rotor. Flux linkages, currents, W_f and W_c, are expressed as function of electrical angular displacement θ_r as seen in eqn (48).

$$W_f = \frac{1}{2} i^2_{sABC}\left(L_s - L_{ls}\right) + i_{sABC} L'_{sr} i'_{rABC} + \frac{1}{2} i^2_{rABC}\left(L_r - L_{lr}\right) \qquad \text{eqn (46)}$$

$$dW_m = -T_{em} d\theta_{rm} \qquad \text{eqn (47)}$$

$$\theta_r = \frac{P}{2}\theta_{rm} \qquad \text{eqn (48)}$$

Where: P is the number of poles in the machine. Because of the equality $W_f = W_c$, electromagnetic torque may be evaluated as in eqn (49). When eqn (46) is substituted into eqn (49) it yields the electromagnetic torque in eqn (50) $\left[Newton \cdot meters\right]$. Finally torque and rotor speed can be related by eqn (51).

$$T_{em}\left(i_j, \theta_r\right) = \frac{P}{2} \cdot \frac{\partial W_c\left(i_j, \theta_r\right)}{\partial \theta_r} \qquad \text{eqn (49)}$$

$$T_{em} = \frac{P}{2} i^2_{sABC} ABC \frac{\partial}{\partial \theta_r} L'_{sr} i'_{rABC} \qquad \text{eqn (50)}$$

$$T_{em} - T_L = J\frac{2}{P} d\omega_r \qquad \text{eqn (51)}$$

Where: J is the inertia of the rotor and T_L is the load torque on the shaft of the IM .

2.2. The Induction Machine: Modeling and Control

A symmetrical induction machine (Figure 5) expressed in terms of machine variables is presented. A wye (Y) connected, symmetrical induction machine is considered, with identical and sinusoidal distributed windings 120° apart where N_s stands for the number of turns and r_s is the stator resistance. Rotor windings are also identical and sinusoidal distributed and 120° apart, N_r stands for the number of turns and r_r is the stator resistance. Thus the voltage equations are expressed in eqn (52).

$$V_{srABC} = I_{srABC} R_{srABC} + \frac{\partial \psi_{srABC}}{\partial t} \qquad \text{eqn (52)}$$

This equation is valid for stator and rotor, where V_{srABC} denotes the voltage vector, I_{srABC} is the current vector and R_{srABC} is the resistance. For magnetically linear systems flux linkages can be expressed with eqn (53) and eqn (54).

$$\psi_s^s = L_s I_s^s + L_m I_r^s \qquad \text{eqn (53)}$$

$$\psi_r^r = L_r I_r^r + L_m I_s^r \qquad \text{eqn (54)}$$

Where: L_s (eqn (55)) and L_r (eqn (56)) are the self-inductance of stator and rotor phase windings respectively and L_m is the mutual inductance. L_{ls} and L_{ms} are correspondingly the leakage and magnetizing inductances of the stator windings.

L_{lr} and L_{mr} are for the same rotor windings; finally L_{sr} is the amplitude of mutual inductances between stator and rotor windings, with the relationships in eqn (58). If other variables like voltages or currents are referred to the stator, eqn (59) is used.

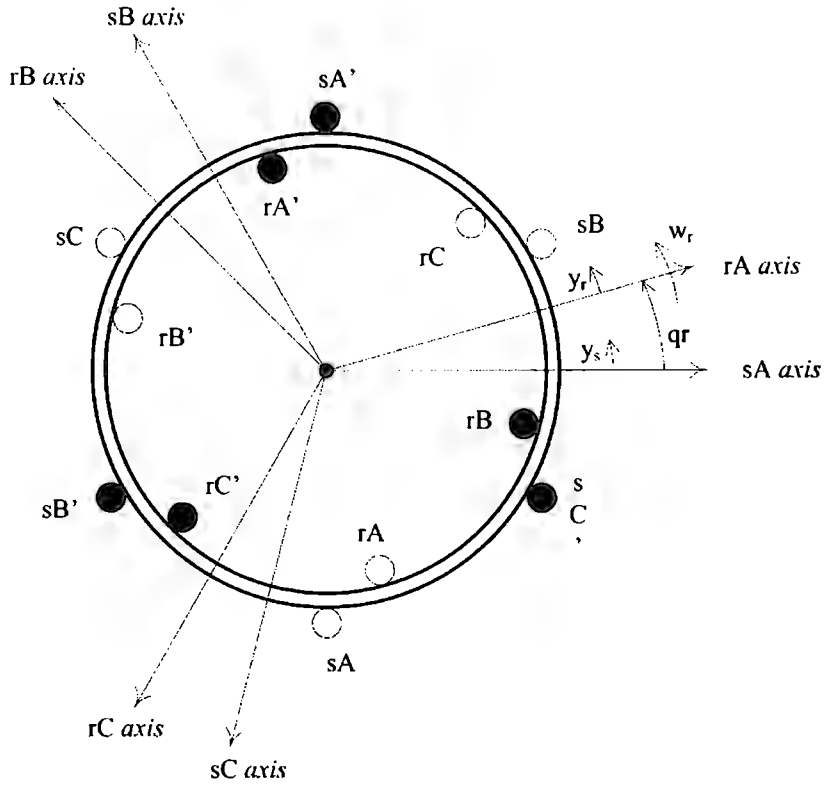
$$L_s = \begin{bmatrix} L_{ls}+L_{ms} & -1/2\,L_{ms} & -1/2\,L_{ms} \\ -1/2\,L_{ms} & L_{ls}+L_{ms} & -1/2\,L_{ms} \\ -1/2\,L_{ms} & -1/2\,L_{ms} & L_{ls}+L_{ms} \end{bmatrix} \qquad \text{eqn (55)}$$

$$L_r = \begin{bmatrix} L_{lr}+L_{mr} & -1/2\,L_{mr} & -1/2\,L_{mr} \\ -1/2\,L_{mr} & L_{lr}+L_{mr} & -1/2\,L_{mr} \\ -1/2\,L_{mr} & -1/2\,L_{mr} & L_{lr}+L_{mr} \end{bmatrix} \qquad \text{eqn (56)}$$

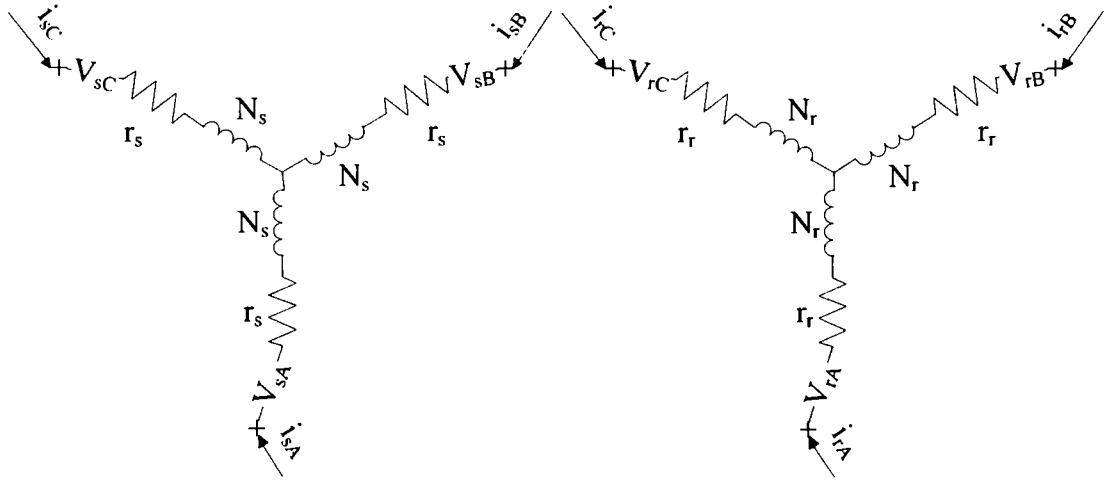
$$L_{sr} = |L_{sr}| \begin{bmatrix} \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r \end{bmatrix} \qquad \text{eqn (57)}$$

$$L_{ms} = \frac{N_s}{N_r} L_{sr} \qquad \text{eqn (58)}$$

$$A'_r = \frac{N_r}{N_s} A_r \qquad \text{eqn (59)}$$



a) Expressed in terms of machine variables



b) Circuits for the stator and rotor

Figure 5. Symmetrical Induction Machine, 2-poles and 3-phases.

2.2.1. Mathematical Model in Stationary Reference Frame

As shown in Section 2.1.1, we can write the model variables in an arbitrary reference frame; here they will be written in the two axis stationary reference frame. It is a vector model in D-Q reference frame for a symmetrical squirrel cage induction machine. It does not include the magnetic circuit model and it is considered to be balanced.

$$v_{sD} = r_s i_{sD} + \frac{d\psi_{sD}}{dt} \quad \text{eqn (60)} \qquad v_{sQ} = r_s i_{sQ} + \frac{d\psi_{sQ}}{dt} \quad \text{eqn (61)}$$

$$v_{rD} = 0 = r_r i_{rD} + \frac{d\psi_{rD}}{dt} - \omega_r \psi_{rQ} \quad \text{eqn (62)} \qquad v_{rQ} = 0 = r_r i_{rQ} + \frac{d\psi_{rQ}}{dt} + \omega_r \psi_{rD} \quad \text{eqn (63)}$$

$$\psi_{sD} = L_s i_{sD} + L_m i_{rD} \quad \text{eqn (64)} \qquad \psi_{sQ} = L_s i_{sQ} + L_m i_{rQ} \quad \text{eqn (65)}$$

$$\psi_{rD} = L_r i_{rD} + L_m i_{sD} \quad \text{eqn (66)} \qquad \psi_{rQ} = L_r i_{rQ} + L_m i_{sQ} \quad \text{eqn (67)}$$

$$T_{em} = \frac{3P}{2J}\left(\psi_{sD} i_{sD} - \psi_{sQ} i_{sQ}\right) \quad \text{eqn (68)} \qquad \omega_s = \omega_{slip} + \omega_r \quad \text{eqn (69)}$$

Here stator voltages are eqn (60) and eqn (61), rotor voltages are eqn (62) and eqn (63), which are equal to zero. Flux linkages for stator and rotor are eqn (64), eqn (65), eqn (66) and eqn (67). Electromagnetic torque is described by eqn (68). Finally stator, rotor and slip rotational speeds are related by eqn (69). A block diagram of the model is shown in Figure 6, here F_{sDQ} are the internal equations that govern the model previously described.

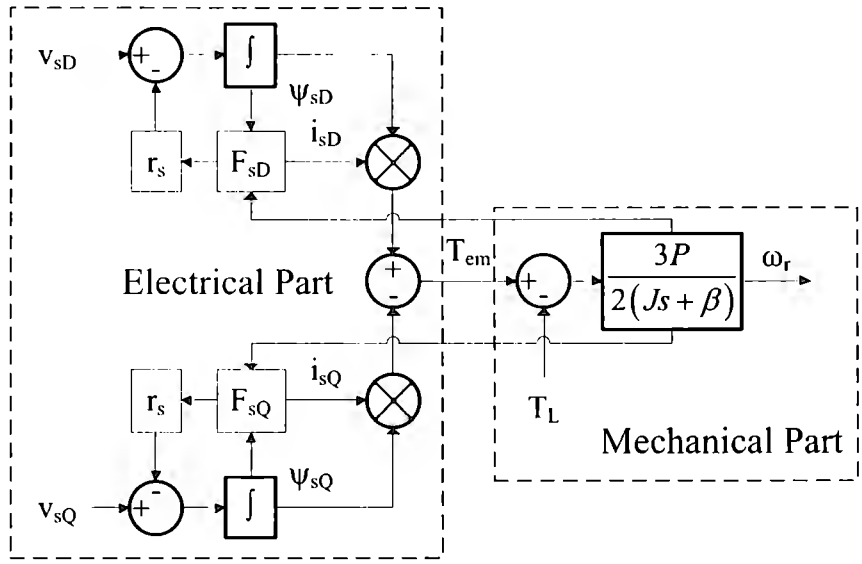


Figure 6. Block diagram of induction motor model in stator reference frame.

2.2.2. Direct Torque Control for Induction Machines

Direct Torque Control comes up with the work of Takahashi [64] and Depenbrock [13] in the middle of 1980's. It must be emphasized that it was meant to be an analog implementation. A diagram of DTC is shown in Figure 7 it is made in two basic stages.

The first stage has two hysteresis bands, one for the electromagnetic torque and the other for the stator flux. A voltage vector that fulfills torque and stator flux requirements must be selected. This voltage vector is selected by using hysteresis bands for stator flux with two possible states and electromagnetic torque with three possible states. The resulting state is the comparison between the limits of the bands and the real instantaneous values, as shown in Figure 8.

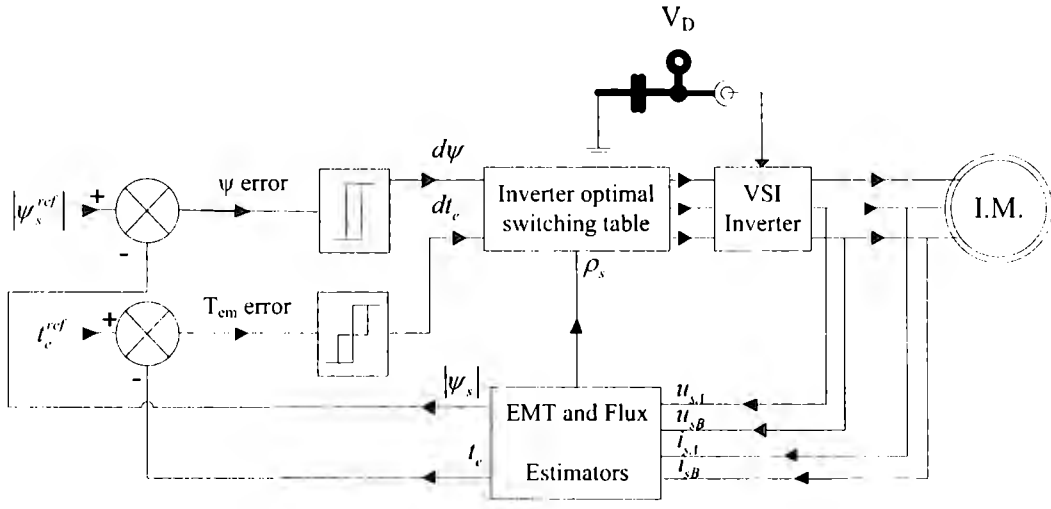


Figure 7. Simplified DTC Scheme for IM Drive.

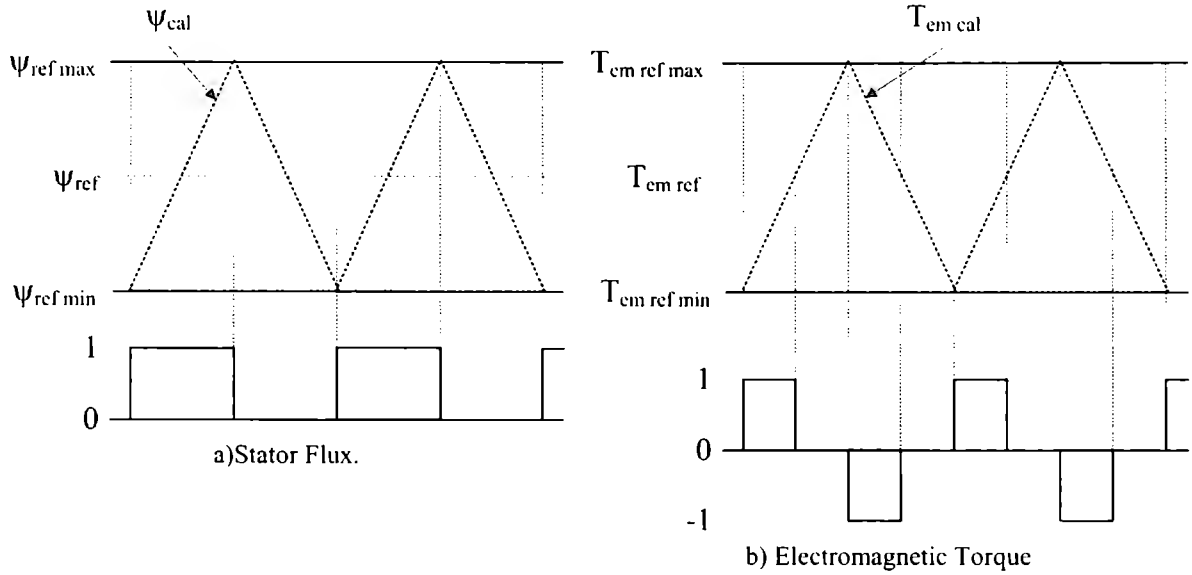


Figure 8. Hysteresis bands.

Values of T_{em} and ψ_s are taken from the stationary DQ model and then compared with their respective hysteresis bands, thus, generating the necessary data to make the

decision on what voltage vector will keep them inside their hysteresis bands. The hysteresis bands are hard limits for the comparison of the real value of the element being compared. If this element is outside the limits of the hysteresis bands, action must be taken to maintain it inside the limits, as shown in Table 9.

Table 9. Hysteresis bands for ψ_s and T_{em}

For ψ_s as in Figure 8.a:	For T_{em} as in Figure 8.b:
- $\psi_{cal} < \psi_{ref\,min} \rightarrow 1$ and $\psi_s \uparrow$	- $T_{em\,cal} < T_{em\,ref\,min} \rightarrow 1$ and $T_{em} \uparrow$
- $\psi_{cal} > \psi_{ref\,max} \rightarrow 0$ and $\psi_s \downarrow$	- $T_{em\,cal} = T_{em\,ref} \rightarrow 0$ and T_{em} remains equal.
	- $T_{em\,cal} > T_{em\,ref\,max} \rightarrow -1$ and $T_{em} \downarrow$

The second stage on the DTC execution is to calculate the adequate voltage vector. The position of the voltage vector must be taken into consideration, although not precisely but just *the sector* in which it resides (Figure 10.b). This way the voltage vector (which is normally implemented by using fixed tables) can be selected and sent to the voltage source inverter. This control strategy does not require of a coordinate transformation thus, simplifies the control algorithm.

The motor decoupling is made by a simple on-off control law, which uses selection tables using the relative voltage vectors and then switches the semiconductors in the voltage source inverter. In DTC, stator flux and electromagnetic torque can be measured or estimated, if they are estimated the signals in the terminals of the motor are used. Flux can be estimated as shown in eqn (70), which can be written in an incremental discrete form as in eqn (71), depreciating the losses in the stator resistance.

$$\psi_s = \int v_s - r_s i_s \qquad \text{eqn (70)}$$

$$\Delta\psi_s = \Delta t\left(v_s\right) \qquad \text{eqn (71)}$$

. It can be seen from eqn (71) that a change in the stator voltage is proportional in magnitude and sense to the stator flux vector. It is for this reason that DTC uses voltage source inverters -VSI- for the control of the stator flux.

2.2.2.1. Voltage Source Inverters in DTC

It is possible to control stator flux and electromagnetic torque through the control of the voltage source inverter, as it is shown in Figure 9, it has eight possible states shown in Figure 10. The switching state is defined with the switching functions $\left(S_U, S_V, S_W\right)$ when its value is *one* -1- the switch is set to positive voltage and when it is set to *zero* -0- it is a negative voltage.

Each switching state generates a voltage space phasor, six of them are active voltage space phasors -AV- $\left(\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4, \bar{v}_5, \bar{v}_6\right)$ and the other two are zero voltage space phasors

-ZV- named $(\vec{v}_0, \vec{v}_7)$. Each inverter vector voltage can be described by eqn (72); in Table 10 the states of the inverter and the voltages in DQ reference frame are shown.

Using Table 10 the position of every voltage phasor in DQ frame can be determined, this way it can be determined which phasors will increment the stator flux and electromagnetic torque. This information can be seen in Table 11, here an arrow up means an increment in flux or torque, an arrow down means a decrement and a horizontal line means a null effect.

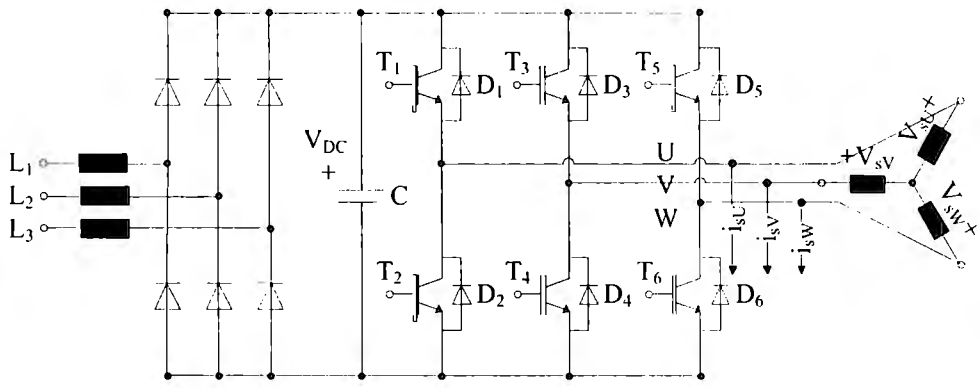
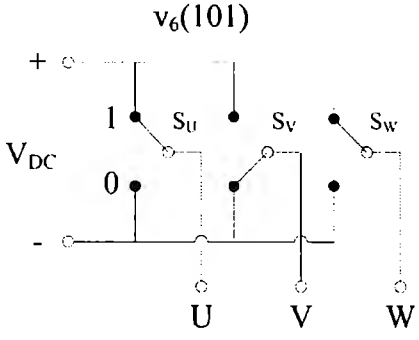
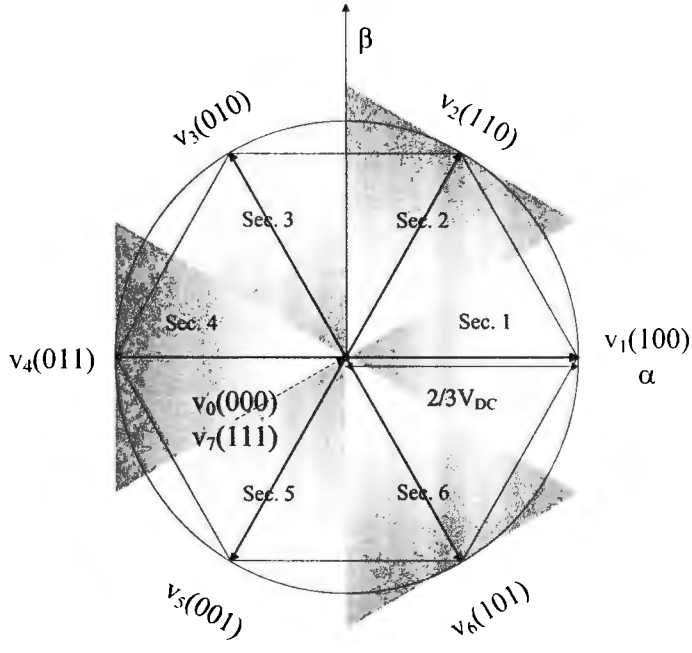


Figure 9. Three-phase power converter connected to the stator of the Induction Machine



a) Simplified scheme of the VSI in switching state $v_6(101)$.



b) Output voltages generated by the switching states of the VSI.

Figure 10. VSI switching modes.

$$v_{sn} = \begin{cases} \frac{2}{3} V_{DC} e^{j(n-1)\frac{\pi}{3}} & \text{for } n = 1,\ldots,6 \\ 0 & \text{for } n = 0,7 \end{cases} \qquad \text{eqn (72)}$$

Table 10. Inverter states and voltage relationship of V_{DC} in DQ reference frame

VSI State	Branch A	Branch B	Branch C	Voltage D	Voltage Q
V_0	0	0	0	0	0
V_1	1	0	0	2/3	$1/\sqrt{3}$
V_2	1	1	0	1/3	$-1/\sqrt{3}$
V_3	0	1	0	1/3	0
V_4	0	1	1	−2/3	$-1/\sqrt{3}$
V_5	0	0	1	−1/3	$-1/\sqrt{3}$
V_6	1	0	1	1/3	0
V_7	1	1	1	0	0

Table 11. Inverter states and voltage relationship of V_{DC} in DQ reference frame

	V_1	V_2	V_3	V_4	V_5	V_6	V_0 & V_7
$\lvert\psi_s\rvert$	↑↑	↑	↓	↓↓	↓	↑	– – –
T	↓	↑	↑	↓	↓↓	↓↓	↓

2.3. PID: Proportional Integral Derivative Control

Despite of decades of research on developing new control methods, PID controllers are the most widely used on industry, due to its simplicity and popularity. Even though, PID controllers do not have a satisfactory response on nonlinear systems, such as the induction machine.

PID is implemented in many different forms, as a standalone controller, as part of bigger control scheme or a hierarchical distributed process control system. The PID control scheme can be approached in many different ways, it can be viewed as a device which operates with very few rules, and it can also be analytically approached [2].

2.3.1.1. Feedback Principle

The feedback idea is very simple and yet very powerful, its application has resulted in major advances in control, and communications, among other branches of engineering. Assume for a moment that a process variable increases every time a manipulated variable is decreased. This type of feedback is known as negative feedback, shown in Figure 11, because the manipulated variable moves in opposite direction of the process variable.

The reason feedback systems are interesting is because feedback makes the process variable close to the set point, regardless of disturbances and variations of the process characteristics.

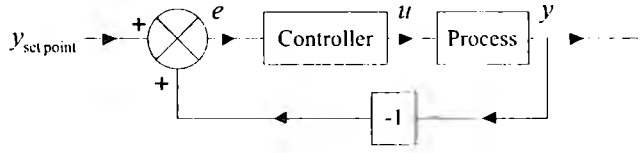


Figure 11. Process with feedback controller

2.3.1.2. On-Off Control

Feedback can be arranged in many different ways. The simplest way to describe a feedback is using on-off control, a law that implies the maximum corrective action is always used. As shown in eqn (73) and previously shown in Figure 11, where e is the control error described in eqn (74).

$$u = \begin{cases} u_{\max} & \text{if } e > 0 \\ u_{\min} & \text{if } e < 0 \end{cases} \qquad \text{eqn (73)}$$

$$e = y_{\text{set point}} - y \qquad \text{eqn (74)}$$

This type of feedback is called on-off control, is simple and there are no parameters to choose. It will often succeed in keeping the process variable closer to the set point, but it will oscillate.

2.3.2. PID Control Theory

A PID controller is the result of the sum of three terms, proportional, integral and derivative errors, as shown in eqn (75). Where: e is the control error described previously in eqn (74), K is the proportional gain, T_i is the integral time and T_d is the derivative time.

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau + T_d\frac{de(t)}{dt}\right] \qquad \text{eqn (75)}$$

2.3.2.1. Proportional Action

The oscillation effect caused by on-off control is caused because the system over reacts and because any kind of change, even the most subtle, the manipulated variable will change over the entire range. This can be avoided using proportional control, because the response of the controller is proportional to the control error. The response is linear and saturated with positive and negative limits.

The saturation limits must be set, and the linear range can be specified by giving the slope of the characteristic or giving the range where the response is linear. The range is normally centered in the set point, a proportional controller acts like an on-off controller for large errors, and it produces static or steady state error, not taking the process variable to the set point [2].

2.3.2.2. Integral Action

One of the main drawbacks of proportional control is that it generates a steady state error, meaning that the controlled variable will never reach the set point. This can be corrected with an integral action, which main function is to make sure that the process output *agrees* with the set point in steady state. The integral action works as follows: with a

small positive error the control signal will increase and with a negative error the control signal will decrease [2].

2.3.2.3. Derivative Action

The derivative action will improve the stability of the loop. Because of the process dynamics, it will take some time before a change in the control variable be noticeable to the process, thus the control will be late in correcting the error [2]. The action of the derivative action may be interpreted as if the control is made proportional to the predicted process output, made by extrapolating the error by the tangent to the error curve as in Figure 12.

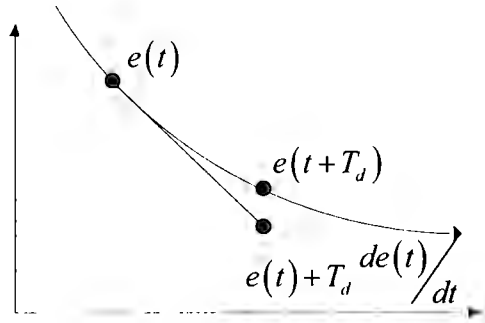


Figure 12. Interpretation of the derivative action as a predictive control.

2.3.2.4. Improved PID Control

There are several variations of the PID control law described in eqn (75) which will substantially increase the performance of the controller. The non-interactive form of the PID algorithm which is a more general expression of the PID [2], is shown in Figure 13 and transfer function in eqn (76), which is used in this thesis. The reason to call it non-interactive [2] is because the integral time does not influence the derivative part and vice versa.

$$G(s) = K\left[1 + \frac{1}{sT_i} + sT_d\right] \qquad \text{eqn (76)}$$

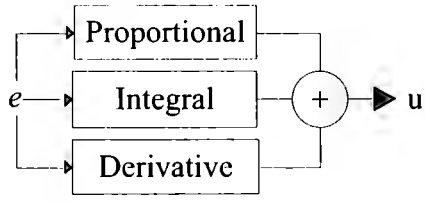


Figure 13. Non-interacting [2] form of the PID control.

3. Intelligent Control

Intelligent Control -IC- techniques emulate characteristics of biological systems, offering the possibilities to create control techniques with novel capabilities. Today's competitive environment is an excellent place for these control techniques because they can provide competitive edge capabilities sought by every company out there. Even though many applications have been described in the literature, only a minority becomes real products [10], [39].

Intelligent Control is the synergy of *Artificial Intelligence* -AI- and *Control*. AI can be defined as the ability of a computer to perform tasks usually related to intelligent beings. *Control* is the branch of Mathematics and Engineering which deals with design, identification and analysis of systems headed for controlling them to make them behave in a desired way [54].

Artificial intelligence has made significant advances on intelligence regarding *expertise, planning, interaction, vision, coordination, regulation, and reaction.* However there is still a gap in the *common sense* region. Creating systems able to make sensible decisions about unfamiliar everyday situations in non-specialized domains is still a very difficult mission [30].

After the massive earthquake in Japan on March 11 of 2011 several reactors on the Fukushima nuclear energy plant were damaged. We would have expected robots could be sent to repair the damaged reactors [26]. However several problems have to be considered and tackled by researchers before robots can act in a situation like this. The damaged reactor represents very challenging environments for the robots having collapsed structures and rough terrain.

Obstacles as simples for humans as closed doors [27] can represent major problems to robots. Despite Japan being the nation with the worlds most advanced humanoid robots, they are still research projects; they can walk, dance, climb stairs but not perform complex tasks. And even though two PackBots from iRobot were sent to inspect the reactors [26], [32] they are tele-controlled robots. So, there is still a gap in AI and intelligent control of real world applications that must be filled, because we have underestimated the complexity of human intelligence.

Intelligent control is designed to create control methods providing a degree of intelligence and autonomy in their control decisions giving them the ability to refine their performance. This makes intelligent control the fastest growing area in control systems on the last years. The main tools of intelligent control used in this thesis are presented in the following sections of this chapter.

3.1. Fuzzy Logic

Uncertainty and ambiguity makes the world a complex place, as human's, evolution gave us intellect and the capacity of thinking making able to address complex, ambiguous and uncertain problems. All of this is possible because we can use reason with incomplete and vague information.

Professor Lofti A. Zadeh introduced the seminal paper on fuzzy set theory in 1965 [72], with constant novel developments on the field all around the world. Japan has been the leader on implemented applications of the fuzzy theory. On western cultures the interpretation given to fuzzy theory is not good given that we are married to the *crisp* idea of yes or no, and that the concept of fuzziness gets a negative connotation.

3.1.1. Fuzzy Logic Concepts

As said by professor Zadeh in his article on fuzzy logic: *To begin with, fuzzy logic is not fuzzy. Fuzzy logic is precise. Basically, fuzzy logic is a precise logic of imprecision* [sic] [73]. A more precise and formal description would be that fuzzy logic is a reasoning system in which objects are classes with unsharp limits. Thus, *unsharpness* of class boundaries may be equated to *fuzziness*.

Fuzzy logic may be seen as a generalization of the classical, bivalent logic; where no degrees of truth are permissible. In multivalued logic, degrees of truth are permitted but fuzzy logic goes even further. In fuzzy logic everything is allowed to be a *matter of degree* or, to have a *membership degree*. Moreover, *degrees* are allowed to be fuzzy.

The basic idea in fuzzy logic is the concept of a fuzzy set. A fuzzy set, F, in a universe U, $U = \{u\}$ is a class of elements belonging to U with unsharp limits; which limits are precised through association with a membership function. The membership function relates with each element u, *of* U, is the degree of membership in F. Degrees of membership are numbers in the unit interval and more generally, elements of a lattice [73].

Two basic concepts of fuzzy logic are graduation and granulation. *Graduated* objects are classes with unsharp limits, known as fuzzy sets. *Granulated* partitions a set (fuzzy or crisp) converted into granules, which is typically a fuzzy set. Granulation can be seen as a generalization of quantization. As an example granulation applied to the temperature variable results in fuzzy sets labeled cold, warm and hot.

Granulation brings about the notion of a linguistic variable: a variable whose values are fuzzy sets with linguistic labels. Most of the fuzzy logic applications use the notion of a linguistic variable. We then proceed by asking a question q in the form: What is the value of variable Q? The answer to this question is obtained from an information set I; a collection of propositions mostly from world knowledge.

To obtain the answer to q questions two steps are involved. In the first step the propositions in I are to be defined. In the second step these propositions are computed,

giving us an answer to q. Here fuzzy logic is used to define and compute propositions p; being it represented as a generalized constraint.

A generalized constraint is an expression of the form: X *is* R where X is the constrained variable, R is the constraining relation and r is an identical variable whose values define the way in which R constrains X. In natural language constraints are mostly *possibilistic*, but *probabilistic* and *veristic* can also be found [73].

On the second step the computation is carried out as follows: assuming that I is a collection of generalized constraints $X_i, i = 1,...,n$. The answer to q, $Ans(q/I)$ is expressed as a function of X_i:

$$Ans(q/I) = f(X_1,...,X_n)$$

The values of X_i are not known but we know R_i, the constraints in the values of X_i. So we can apply the *Extension Principle* of fuzzy logic to complete the constrained value of $Ans(q/I)$, the answer that we are looking for. Fuzziness is prevalent in the reality; although in science models are widely based in classical logic. An extensively unrecognized fact is that classical logic being intolerant to imprecision is not the right logic to deal with the fuzziness of reality; what it is best suited for this purpose is fuzzy logic [73].

3.1.2. Fuzzy Logic Controllers

A fuzzy logic systems is one type of approximate reasoning system, it can be considered a type of intelligent non-linear approximator. A non-linear function can be estimated by using a finite set of fuzzy rules. A fuzzy function estimation with a finite number of rules can approximate any continuous function on a close domain with any grade of precision.

Rules define each bounded region; the fuzzy approximator covers the curve with rule regions and adds average regions that overlap. The number of required rules grows exponentially with the number of inputs and the degree of precision required or desired. The most general form of the rule is the following:

IF antecedents THEN consequences.

With this in mind, it will be easier to explain a fuzzy logic approximator or controller. At least three main stages must be performed in fuzzy logic in order to obtain a result. On the first stage the variables have an uncertainty metalinguistic level. Thus, the universe of discourse of each variable is classified in fuzzy sets related to a label for example: cold, warm, hot.

The normal, *crisp* variables go through the process of *fuzzification* (Figure 14) which converts the crisp values to fuzzy values with a membership degree in the *[0, 1]* range. The purpose is to determine in which degree the crisp value fits in a fuzzy set, characterized by membership functions.

In the second stage, linguistic rules used to infer are proposed (normally proposed by an expert). These will set a structure to make the system behave in an adequate manner according to the reference model. The set of inference rules defines a consequence by assigning a membership degree to a fuzzy set characterizing the outputs. There are many possibilities on a fuzzy inference system that can be used with different operators, some of the most widely used are described in Table 12.

Table 12. Fuzzy Inference Operators

Name	Description
AND	T-norm operator for intersections, minimum, algebraic product. It is used to obtain the firing strengths of an individual rule with antecedents joined by *AND*.
OR	T-conorm operator for unions, maximum, algebraic sum. It is used to obtain the firing strengths of an individual rule with antecedents joined by *OR*.
Implication	A T-norm operator for minimums, algebraic products, bounded products, among many others. It is used to obtain qualified consequent membership functions based on given firing strengths
Aggregate	Usually a T-conorm operator for maximums, algebraic products, bounded sums, among many more. Used for aggregating qualified consequents membership functions into an overall output of the M.F.

The third and final stage is a process to determine the optimum output values, it begins once the consequences are obtained and is known as *defuzzifiction*. This process converts fuzzy membership values, which come from the consequences of the inference rule, to crisp values again. To do so, previously established membership functions for each one of the outputs are used to obtain a measurable value.

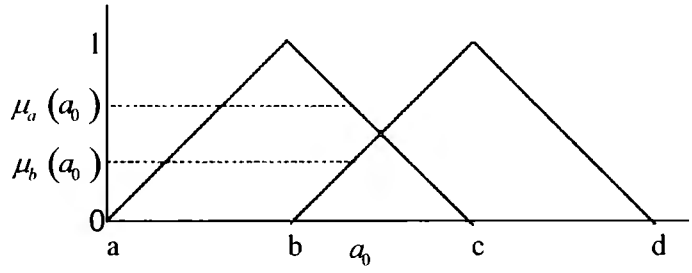


Figure 14. Fuzzification Process of a_0

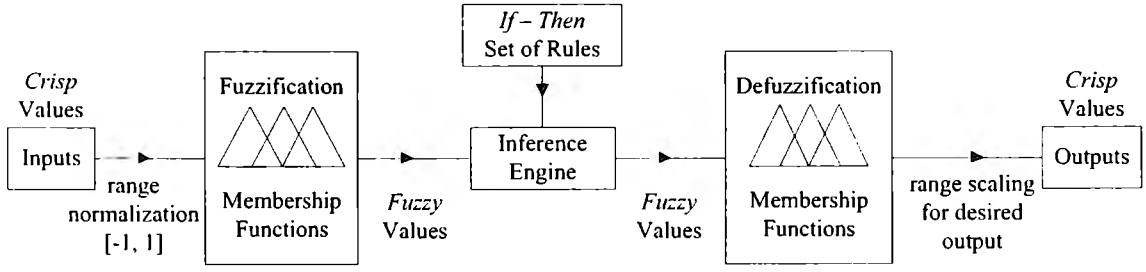


Figure 15. Fuzzy Logic Controller

Even though these are the three main parts of fuzzy systems, not all the existing versions of fuzzy system make use of them, the principal form of fuzzy systems are *Mamdami*-, *Sugeno*- and *Tsukamoto*-type. Both Sugeno and Tsukamoto avoid the use of

defuzzifiers and the Mamdami can be considered special cases of both of them, they will be further explained in the following sections.

An advantage of fuzzy systems is that they does not require of a model of the plant or real time identification of parameters. It will essentially apply linguistic control to automatic control. A list with the following applications has motivated the development of fuzzy logic in control systems [55] are shown in Table 13.

Table 13. Applications that have motivated the development of Fuzzy logic in control systems [55]

Autonomous	Simplified	Robust
Adaptable	Understandable	
No model plant required	Implementable	

3.1.2.1. Mamdami Fuzzy Controller

The Mamdani is one type of fuzzy controller proposed in 1974 by E. H. Mamdani [41] to demonstrate that fuzzy if/then rules could regulate the model of a steam engine. This controller has the same diagram of Figure 15 and has three main parts: fuzzification, rule evaluation and defuzzification. It works as the previously described controller.

Several defuzzification techniques exist, in this thesis we use *Center of Sums Method,* for Mamdami controllers. In this method the output from each contributing rule is taken, and then we add all of them. It can be described by equation eqn (77) and graphically described in Figure 16.

$$u^* = \frac{\sum_{i=1}^{N} u_i \cdot \sum_{i=1}^{N} \mu_{A_k}(u_i)}{\sum_{i=1}^{N} \sum_{k=1}^{n} \mu_{A_k}(u_i)} \qquad \text{eqn (77)}$$

At the end the fuzzy controller will deliver crisp values, consequences of the linguistic rules previously created. With that the system will understand orders (inputs) and perform pertinent actions (outputs); all of these process is illustrated in Figure 15.

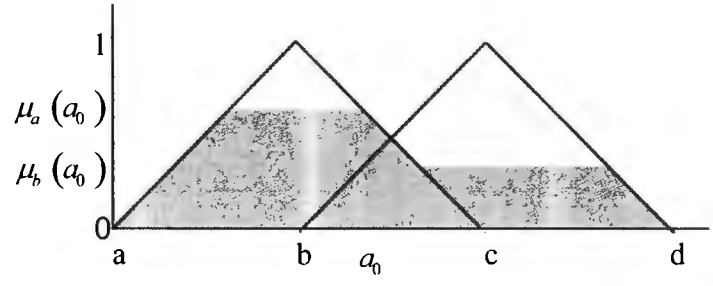


Figure 16. Membership Function Defuzzification by Center of Sums.

3.1.2.2. Takagi-Sugeno Fuzzy Controller

The Takagi-Sugeno is another kind of fuzzy controller [63] with no separate defuzzification block. In this controller the rule take the form that the consequence is a function of the input variables:

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ z = f(x, y).$$

Where: x and y are input variables, A and B fuzzy sets, $f(x, y)$ is a function in the consequent and z is the crisp output. To defuzzify, a crisp function (usually a polynomial) is weighted by the values of the fired rules. The product of the sum of the minimum of the antecedents of every rule fired is divided by the value of the polynomial evaluated in that fired rule, all this divided by the sum of the minimum of the antecedents of every rule fired -eqn (78).

$$u^* = \frac{\sum_{i=1}^{r}[\min \mu(a_0) \cdot Y(a_0)]}{\sum_{i=1}^{r} \min \mu(a_0)} \qquad \text{eqn (78)}$$

Where: $Y(a_0)$ can have any form, but it usually is a polynomial of n order that depends on the crisp inputs: $Y(a_0,\ldots,a_n) = p_0 + p_1 \cdot a_1 + \ldots + p_n \cdot a_n$. These polynomials can be calculated using regressions to adjust them to a desired form or function.

When the function f is constant (zero order shown in Figure 17) the special Sugeno system is obtained which can be seen as a special Mamdani type with singleton consequences. It can be also obtained a special case of the Tsukamoto controller, where the consequent of each rule is specified by a membership function of a step crossing at a constant.

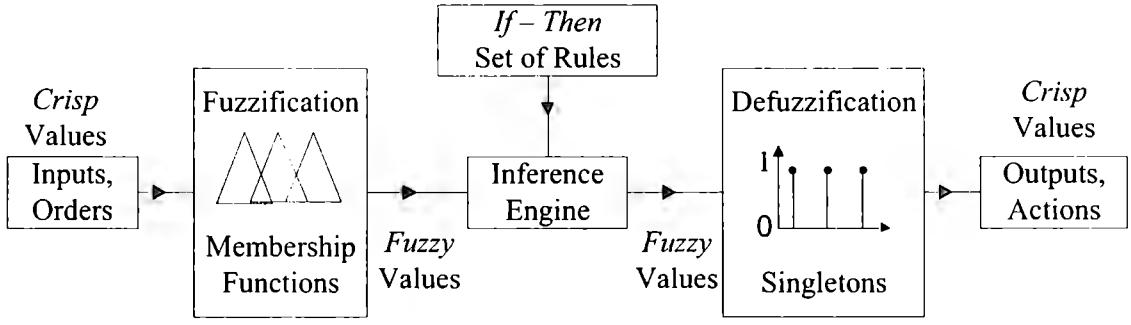


Figure 17. Takagi-Sugeno zero order fuzzy system.

3.1.2.3. Tsukamoto Fuzzy Controller

This controller is similar in the the two first blocks of the other fuzzy controllers. However, the consequent of each rule is represented by a monotonic fuzzy membership function. This causes that the inferred output of each rule to be crisp. Similar to the case of the Sugeno controller the output is obtained as weighted average of individual rule outputs. A monotonic function is a function that only increases or never decreases and vice versa, which means that it always preserves its order.

3.2. Artificial Neural Networks

Artificial Neural Networks -ANNs- are universal function approximators, capable of narrowly approximate complex functions and systems, including non-linear systems. Our brain is like a machine for information processing. All the signals obtained from the senses running through the nervous system is the processed information.

It is estimated that the brain consists of around 100 to 500 billion neurons [54]. These neurons are arranged in networks, depending on the objective, neurons can be hierarchically organized or layered. Nervous cells or neurons are organized as computer networks; they communicate through a region called *synapse*. Neurons are mainly composed by the axon and dendrites (Figure 18).

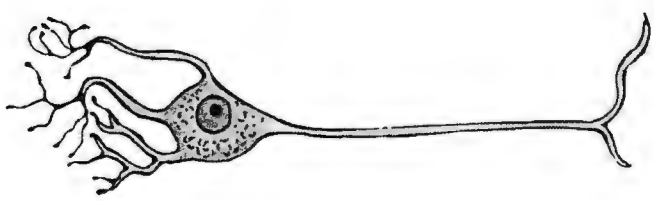


Figure 18. Schematic of a Neuron

For simplified purposes, the way a neuron work can be seen where none or many electrical impulses are received through dendrites coming from other neuron. These electrical impulses are added and a resultant potential is obtained. Certain level must be obtained so the axon of this neuron generates an electrical impulse on its axon, otherwise a potential is not generated.

In ANNs the artificial neurons are connected via weights, which give positive or negative value to the incoming signals. After that, the signals are summed and processed by evaluating the result in a function called *activation function*. The result can be then transmitted to other neurons; the process can be seen in Figure 19.

Activation functions produce the output of the neuron taking into account the summed inputs. There are many activation functions; however the most used are *Step, Linear, Sigmoidal* and *Hyperbolic Tangent*, as shown in Figure 20.

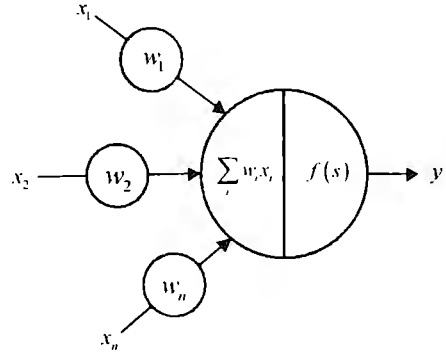
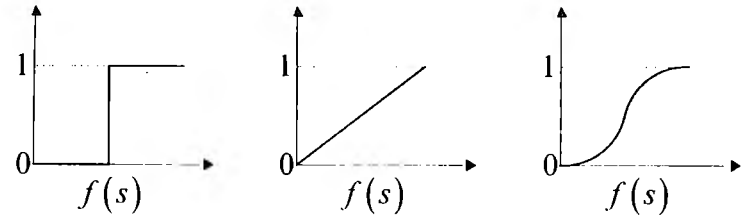


Figure 19. Schematic of an Artificial Neuron



a) Linear b)Step c)Hyperbolic Tangent and Sigmoidal

Figure 20. Activation Functions of Artificial Neurons approximated forms

Neurons connect among them and create neural networks. The main concept behind ANNs is to simulate the behavior of the human brain in order to define an artificial computation and solve problems. ANN may have an advantage in speed over the human brain, given that modern processors operate at very high rates; however these processors do not work in parallel whereas the brain does. Table 14 shows some of the most important properties of ANNs.

Table 14. Properties of Artificial Neural Networks

Generalize information
Recognize patterns
Classify data
Learn from experience
Massive parallel processor
Approximate and predict functions
Universal approximators

Artificial neural networks can generalize data, but to do this they must be trained first. Training is the process where ANNs find the weights of each neuron to represent any given function. There are several training processes, one of the most important is the Backpropagation algorithm used in feed forward networks; this process derives from the fact that error wants to be minimized.

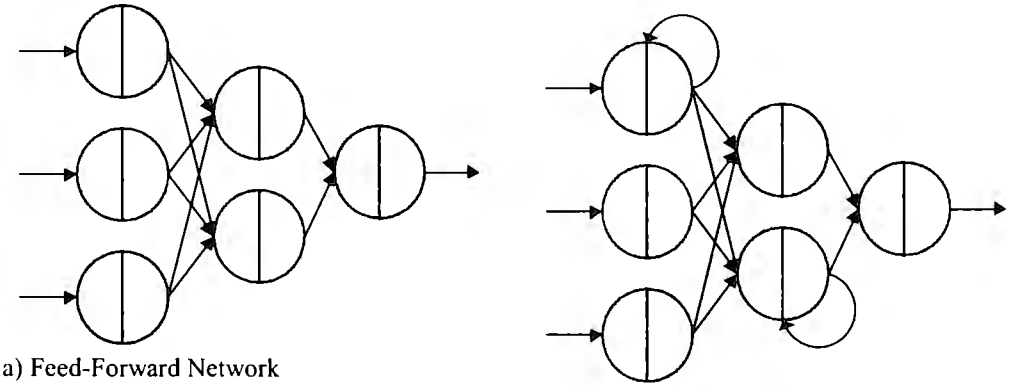
To train a network some data samples are needed, with these data during the training process, the ANN measures the error between the desired output and the actual real output. In the case of backpropagation, the error is retro-propagated to see what weights have to be changed so the difference is minimized and optimize the weights, so the network gets *trained.*

3.2.1. Artificial Neural Networks Classifications

Just as in biological neurons, ANNs be classified depending on their different structures (Figure 21). Depending on the application, and they can also be classified by the learning procedure (Figure 22), as shown in Table 15.

Table 15. Classification of Artificial Neural Networks [54].

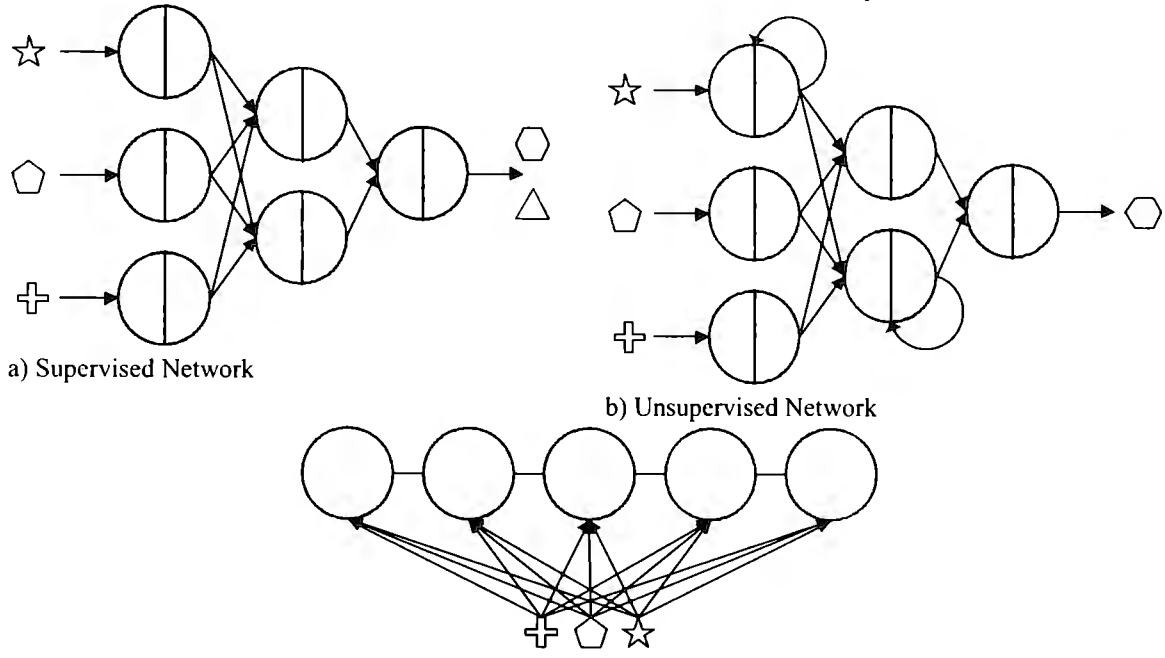
Name	Description
By Structure shown in	
Feed-Forward Networks	There models use the input signals that flow in the direction of the output signals. The outputs are consequences of input signals and weights involved.
Feedback Networks	Similar to the feed forward networks with some neurons having loop signals so the output signals are used in the computation process. The outputs are the result of a non-transient response
By Learning Process	
Supervised Networks	When the data that wants to be trained is known then we can train a network by imposing inputs and outputs and the weights can be found.
Unsupervised Networks	When the information is unknown, this model is used to find patterns in the input space in order to train it.
Competitive and Self-Organizing Networks	No information is needed to train these networks also; however neurons fight for a dedicated response by specific inputs.



a) Feed-Forward Network

b)Feedback Network

Figure 21. Classification of Artificial Neural Networks by Structure.



a) Supervised Network

b) Unsupervised Network

c) Competitive and SOM Networks.

Figure 22. Classification of Artificial Neural Networks by Learning Procedure.

3.2.1.1. Trigonometric Artificial Neural Networks: T-ANN

Considering that conventional ANNs take time during the training process, they cannot be used in dynamic and real-time applications because of this. Sometimes the training algorithm will not converge and the ANN will not be able to assimilate the desired information. Taking this into account, trigonometric series can be used in ANNs.

Trigonometric Fourier series consist in the sum of functions, multiplied by a coefficient added with a constant; the topology of a network based on these series can be seen in Figure 23. The advantage of using a trigonometric series in ANN's is that the weights of the network can be calculated using analytical methods. The error of the solution decreases as the number of neurons increases, which corresponds to the addition of harmonics in the series [54].

A detailed explanation of this topology can be found in [54]. It has been proven by Joseph Fourier in his series [18] that they can model any periodic signal. For any given function $f(x)$ it is said to be periodic if $f(x)= f(x+T)$ where T is the fundamental period of the signal. Knowing this, the function can be modeled using the Fourier series shown in equations eqn (79), eqn (80), eqn (81) and eqn (82).

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos(nx) + b_n \sin(nx)\right) = \sum_{n=1}^{\infty} A_k(x) \qquad \text{eqn (79)}$$

$$a_0 = \frac{1}{T}\int_0^T f(x)dx \qquad \text{eqn (80)}$$

$$a_n = \frac{1}{T}\int_0^T f(x)\cos(n\omega x)dx \qquad \text{eqn (81)}$$

$$b_n = \frac{1}{T}\int_0^T f(x)\sin(n\omega x)dx \qquad \text{eqn (82)}$$

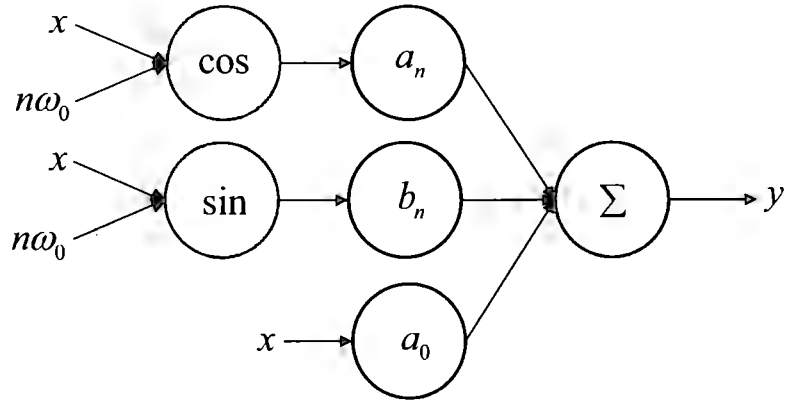


Figure 23. Trigonometric Artificial Neural Network.

3.3. Neuro-Fuzzy Systems

With the use of fuzzy systems the vague form of human reasoning can be represented by mathematical systems. Fuzzy logic gives us the possibility to understand the information contained in the system. Most of the knowledge of fuzzy systems comes from human experts; which sometimes can be disadvantageous.

Artificial neural networks learn from experience, can generalize and represent almost any kind of system. However, the information contained in the networks is difficult to understand. By combining ANNs and Fuzzy systems Neuro-Fuzzy systems are breed, combining the strengths of both systems the weaknesses of each individual system are complemented by the other's strengths.

Neuro-Fuzzy systems can acquire knowledge automatically using the strengths of ANNs, and this information is easy to understand by humans. The information can be presented to the system, trained and no need of a human expert is required.

Other uses of Neuro-Fuzzy systems is clustering, usually employed to initialize unknown parameters such as the number of membership functions or rules. They are also used to update the parameters, creating dynamic Neuro-Fuzzy systems. The main features and types of Neuro-Fuzzy systems are highlighted in Table 16.

Table 16. Main features and types of Neuro-Fuzzy systems

Most useful characteristics of the two systems are combined
The combination should perform in a way that the resulting system should be more efficient
Neuralized fuzzy systems are fuzzy systems mapped to a neural network
Fuzzified neural systems are those where fuzzy concepts are introduced into ANNs

3.3.1. Neuro-Fuzzy Controller with T-ANN's and FCM

This Neuro-Fuzzy controller originally proposed in [50] is a controller based on Takagi-Sugeno inference method [63] but instead of using a polynomial function for the defuzzification process, T-ANN's are used (presented in 3.2.1.1). The schematic diagram of the controller is shown in Figure 24. This controller has the ability to shape the form of the membership functions depending on the data used, thus it is said that it can learn online.

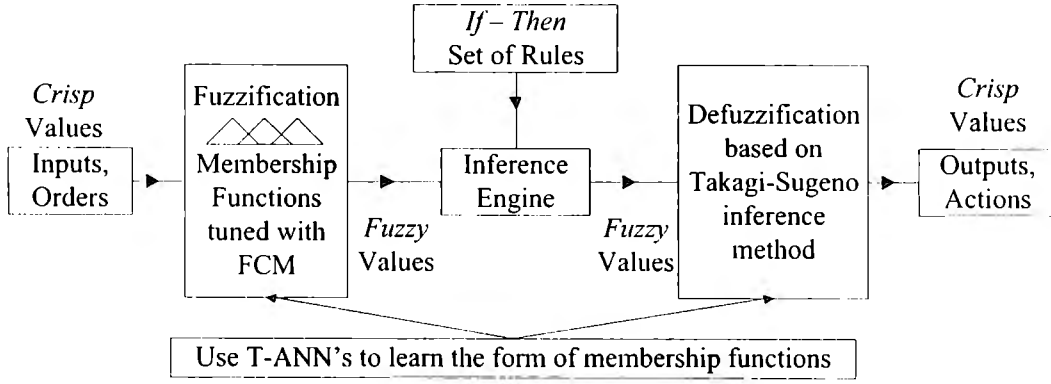


Figure 24. Neuro-Fuzzy Controller with T-ANN's and FCM [50].

3.3.1.1. Fuzzy Cluster Means

Fuzzy Cluster Means -FCM- is a clustering method which splits a set of elements in smaller sets or clusters. While traditional clustering methods assume that each element can belong to only one class, in practice this is not true as clusters normally overlap and some elements can belong to several clusters.

Clustering methods split a set of N elements $X=\{x_1,...,x_n\}$ into c groups or *clusters* $c=\{\mu^1,...,\mu^n\}$. Fuzzy set theory provides a natural way to describe clustering methods in a more real form using FCM. The FCM algorithm is described as follows: Fuzzy partition matrices M, for c classes and N data points are defined by three conditions $M=\{U\in V_{cN}|1,2,3\}$:

1. $\forall\ 1\le i\le c\quad \mu_{ik}\in[0,1],\quad 1\le k\le N$
2. $\sum_{k=1}^{c}\mu_{ik}=1\quad\forall\quad 1\le k\le N$
3. $\forall\quad 1\le i\le c\quad 0<\sum_{K=1}^{c}\mu_{ik}<N$

Textually these conditions means: the first is that the membership value for any element must be in the range of fuzzy set theory [0, 1]. The second implies that the sum of the membership degree on any element on c cluster must be one, so no element has more or less membership degree. Finally the third states that any cluster must be empty.

The FCM algorithm (Table 17) will maximize the distance between the centers of the clusters and minimize the distances of the elements of a same cluster. The FCM criteria function is shown in eqn (83). d_{ik} is the inner product norm (distance) defined by eqn (84). A is a positive definite matrix and m the weighted exponent: $m\in[1,\infty)$.

By assigning values to m and c and defining the working sets, (U,V) can be a global minimum of $J_m(U,V)$ if eqn (85) and eqn (86) are fulfilled [54]. Parameter m determines the fuzziness of the clusters, the larger m the fuzziness of the clusters. If $m = 1$ the algorithm becomes the crisp k-means version, and $m = \infty$ the algorithm is as fuzzy as possible; but usually $m = 2$ [54].

$$J_m=(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^m d_{ik}^2 \qquad \text{eqn (83)}$$

$$d_{ik}^2=\|x_k-v_i\|_A^2 \qquad \text{eqn (84)}$$

$$\forall\ 1\le i\le c\quad 1\le k\le N\quad u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_k-v_i\|}{\|x_k-v_j\|}\right)^{2/(m-1)}} \qquad \text{eqn (85)}$$

$$\forall \quad 1 \le i \le c \quad v_j = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m} \qquad \text{eqn (86)}$$

Table 17. FCM Algorithm [54]

1. Fix c and m, set $p=0$ and initialize $U^{(0)}$
2. Calculate fuzzy centers for each cluster $V^{(p)}$ using eqn (86)
3. Update fuzzy partition matrix, $U^{(p)}$, using eqn (85)
4. If $\left\|U^{(p)} - U^{(p-1)}\right\| < \in$ then, $j \leftarrow j+1$ and return to step 2.

The FCM algorithm is usually applied to shape the input membership functions, because most of the time the form of the output functions is known. After the algorithm is applied the shapes are trained to T-ANN's. Finally the defuzzification process is calculated with eqn (87). Here: r is the number of rules in the fuzzy system, $\mu_{iinputs}$ is the membership value of each rule and $TANN_{defuzzification}$ is the evaluation of the T-ANN's used for the defuzzification process.

$$Output = \frac{\sum_{i=1}^{r} \min \mu_{iinputs} TANN_{defuzzification}}{\sum_{i=1}^{r} \min \mu_{iinputs}} \qquad \text{eqn (87)}$$

3.4. Genetic Algorithms as Optimization Techniques

Since the 1960's there has been an increasing interest to imitate life to develop algorithms for hard optimization problems. A now common term with related research topics is called *evolutionary computation*. Genetic Algorithms -GA's- as powerful and broadly stochastic search and optimization techniques are perhaps the most widely known type of evolutionary computation methods today [21].

Genetic algorithms share an important characteristic with other optimization techniques. They are primarily global search techniques; they identify the optimum by searching the design space for the solution. A significant drawback of these techniques is that they require empirical tuning, based on the class of problems being resolved. There is also no easy way to determine technique/problem-sensitive parameters to implement an automatic optimization technique [69].

There is a multitude of optimization techniques; among them calculus-based, enumerative and random are the most widely used. Calculus-based and enumerative can arrive to a reasonable good solution in small search spaces. Yet when confronted to search spaces of great size and wide variation their efficiency decrements drastically. They are insufficiently robust for complex problems involving huge search spaces [34]. GA's are

exceptionally useful for handling ill behaved, discontinuous and non-differentiable problems [69].

One of the advantages of GA's is that it uses stochastic operators instead of deterministic rules to search for the solution. Moreover, a GA considers multiple points in the search space simultaneously reducing the chance of converging to local minima, in which other algorithms may end up. Another attractive feature of GA is that it searches for many optimum points in parallel, since the evaluation of each point requires and independent computation [68].

Fuzzy systems and artificial neural networks are powerful intelligent control techniques that can be used to control different processes; however the response of these systems is not guaranteed to be optimum. Perhaps the problem is so complex that finding a solution will require a huge amount of time and computation, thus a quasi-optimal solution in a short period of time is a more real and reasonable approach.

Here is where optimization techniques can be useful, GA's assimilate the way in which evolution works and can be used to perform optimization on other intelligent control techniques such as fuzzy and neuro-fuzzy systems. In the following sections this kind of optimization techniques will be presented.

3.4.1. Genetic Algorithms Description

Genetic Algorithms [24] are simple yet powerful general purpose stochastic optimization techniques. They are inspired by evolution of a population subject to reproduction, crossover and mutation in a selective environment where the fittest survive. GA combine the artificial survival of the fittest with genetic operators assimilated from nature, suitable to optimize (can be minimize or maximize depending on the application and problem) a variety of problems.

In mathematical terms, the goal of a GA is to optimize an objective function. They use the concept of Darwin's law of evolution, natural selection and genetics. One of the main advantage is that they are derivative free, meaning that they do not need functional derivative information to search for a set of parameters that will optimize a function.

As an alternative, they rely on repeated evaluation of the objective function. In particular the optimum solution is obtained by investigation new solutions with three genetic operations: selection, crossover and mutation in a selective environment where the fittest survive. This technique is slower than derivative-based methods. However, the freedom of functional derivatives means that the objective function can be as complex as required [68].

The information processed by a GA is a *population*, the most common representation are binary fixed size strings (although many more representations exist [24]). Each individual in the population is a *chromosome*, representing multiple points in the search space. Each bit, part of an individual is an *allele* or *gene* decoded by function to obtain the objective function value of the individual in the search space.

These functions are optimized by the GA and subsequently assigned a fitness value, proportional to its nearness to the desired solution. The main advantages and disadvantages of GAs are shown in Table 18.

Table 18. Genetic Algorithms advantages and disadvantages

Advantages	Disadvantages
Derivative free technique	Probabilistic
Optimize continuous and discrete problems	Expensive computational resources
Consider many points in search space simultaneously	Prone to premature convergence
Parallel search procedure	Difficult to encode problems
Use Stochastic operators in search procedure	Slow, offline optimization
Do not depend on analytical knowledge	Highly dependent of the correct design of the fitness function for optimum performance
Robust, intuitive operation	

3.4.2. Genetic Algorithms Steps

Several forms of Genetic Algorithms exists, however it can be said that methods considered as GAs have at least the following elements: population, selection, crossover and mutation. A GA can be seen on the diagram in Figure 25, is normally divided in the following steps or stages [54]:

1. *Initialization*: Where the necessary elements such as constants and cycles are initialized and created.
2. *Selection*: Selects individuals in the population by means of a fitness function, the fittest the chromosome the more times it will be selected.
3. *Crossover*: Individuals are selected, and mated to obtain offspring.
4. *Mutation*: The offspring is then mutated, changing their genes.

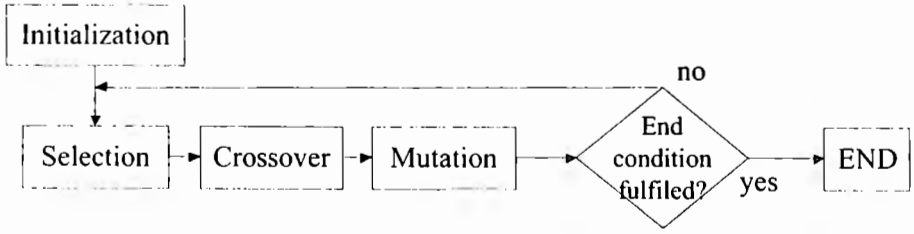


Figure 25. Genetic Algorithm diagram.

3.4.2.1. Initialization

Individuals in the populations are randomly or heuristically generated. A population is a collection of randomly generated individual binary strings. The most used and general elements are described in Table 19.

Table 19. Genetic Algorithms Initialization parameters

Parameter	Description
g	Number of generations of the GA
m	Population size
n	Size of the array of bits representing each individual
PC	Probability of Crossing two individuals
PM	Probability of mutating an individual

3.4.2.2. Selection and Fitness Function

Prior to crossover and mutation operations, a careful selection of the individuals must be performed. There are several selection methods like rank based, tournaments, and probabilistic procedures. Rank based selection mechanisms rank the order of the fitness of the individuals. These types of selection methods are better for situations where it is easier to assign subjective scores rather than specifying an exact objective function.

Tournament selection implies that two or more individuals compete for selection, it is rank based. The tournament may be done with or without reinsertion of competing individuals in the population. Roulette-wheel is also very popular, where the fittest individuals have more probability to be selected.

In order to perform selection it is necessary to measure the performance of the individuals. By selection we aim to maximize the performance of the population, here is where the *Fitness Function* is used. The search must be concentrated on the regions of the search space where the best individuals are located and found. For selection purposes a performance value is associated to every individual, representing the *fitness* of the function.

A *fitness function* is usually used to measure the explicit performance of each individual, which take the form of bit strings being different points in the search space. The population is processed by the GA and is mainly driven by the response obtained to the fitness function. This function can be mathematical, a problem, or more generally, a certain task where the population is evaluated [54].

3.4.2.3. Crossover

The crossover operation mates individuals by combining chromosomes segments. There are many variants of the crossover operation; one of the simplest is the one point crossover operation widely used in the binary encoded frameworks. As depicted in Figure 26, two parents are selected then a crossing point is randomly selected.

After that, the parents are cut at the selected point and their tails are exchanged to generate the offspring. The crossover operation depends on a crossing probability which is compared to a random number. The role of crossover operation is to ensure the exploration of the search space by generating new individuals and act as impetus to the search [54].

3.4.2.4. Mutation

Classically, a mutation is denoted by an altered segment of DNA. Spontaneous mutation is normally not adaptive, and normally does not provide a selective advantage.

Changes may destroy the genome and lead to lethality. This kind of mutation is called silent or natural. However, neutral mutation may play a good role in evolution.

Mutation is considered the second most important operator on genetic algorithms of binary type. The effect of this operator is to change a single bit (gene) in a chromosome (individual). If it not were for this operator other individuals could not be generated through other mechanisms. The operation, a random bit is changed on an individual (as shown in Figure 27) it is executed if a random number is lower than the mutation probability.

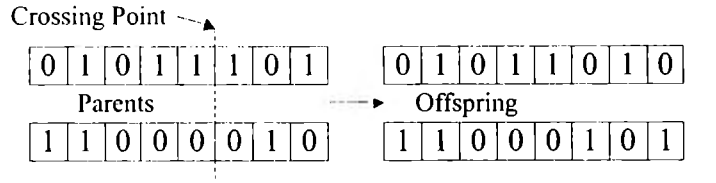


Figure 26. One point Crossing operation for Genetic Algorithms.

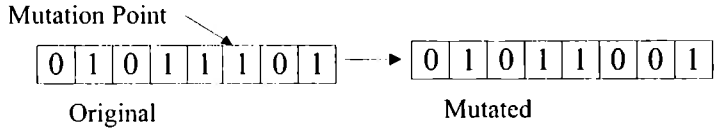


Figure 27. Mutation operation for Genetic Algorithms.

This completes the basic stages of a Genetic Algorithm, as shown in Figure 25, the execution is stopped until the completion of a condition is fulfilled, such as the number of generations, or because it is stopped manually.

3.5. Genetic Programming as Optimization Technique

Genetic Programing -GP- deals with the problem of autonomous programing, evolving the structure of computer programs; this technique provides a way for searching the best or *fittest* program to solve a problem. GP among many other techniques like GA are grouped under the term evolutionary computation, because they share the base of simulating evolution [54].

GP is a technique for auto-generating programs; it works very similarly to genetic algorithms by genetically breeding a population of programs and using Darwin's natural selection and biological inspired operators. The computer programs are represented as trees; their branches are then evolved to create the best solution.

Jhon Koza proposed a variation of genetic algorithms in 1992 [20] that automated the process generation of genetic programming. While in conventional GA's the length of the chromosome is fixed and can restrict the search space, in GP the tree representation eliminates the problem of fixed size chromosomes making the search in a more organized form. More complex optimizations can be done with GP than with GA due to their more organized nature [5].

3.5.1. Genetic Programing Algorithm

It can be seen that Genetic Programming is very similar to Genetic Algorithms because it contains biologically inspired operators. However, the differences cover some of the weaknesses of original GA's as it will be further described. The stages of the algorithm are [35]:

1. Generate an initial population of random functions (these are the computer programs) that will supposedly solve the problem.
2. Execute each program in the population and assign a fitness value to it.
3. Create new population programs which can be:
   i. Copy the best existing program.
   ii. Create new programs using crossover functions.
   iii. Create new programs by mutation.
4. The best computer program, or the best solution found is designated as the result of the algorithm [54].

To understand the way genetic programming algorithm is executed, a very basic example is presented. We we want to create a function that will match the output of x^2 in the [1, 2] range. The program can contain the following operators: {+,-,*,%} that is, add, subtract, multiply and divide operations.

If we apply genetic programming to solve the example previously described, we will have to create an initial population, it can be seen in Figure 28. Then the fitness of each individual must be evaluated, which is carried out as previously explained in section 3.4.2.2. Selection is made by the tournament method because it selects the best individuals. Later, a new population must be generated using crossover and mutation operators.

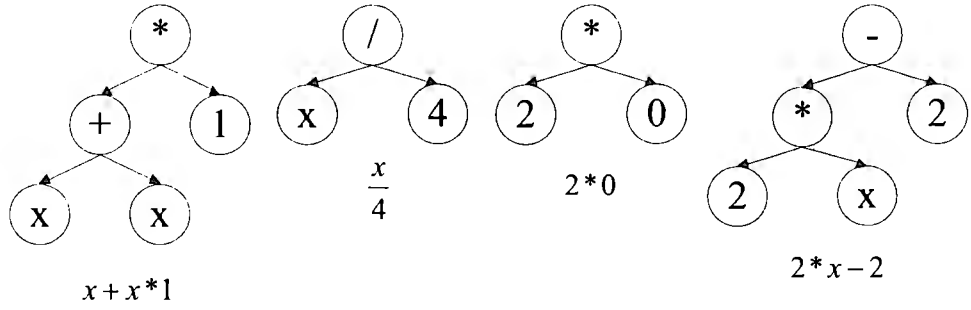


Figure 28. Initial population of random individuals in a GP problem at generation zero.

3.5.1.1. Genetic Programming Crossover

The way crossover operator work depend on the representation of the individuals, since we are representing them by trees, we must use a crossover operator that will maintain coherence in their forms. It is also compared to a crossover probability, which is most of the time big to insure new individuals.

Individuals can grow or diminish their form by interchanging bigger or smaller branches, eliminating the problem of fixed sized chromosomes. A simple crossover

operation is that two individuals (trees) interchange their branches, in a way such that syntactic correctness is maintained [54], as can be seen in Figure 29.

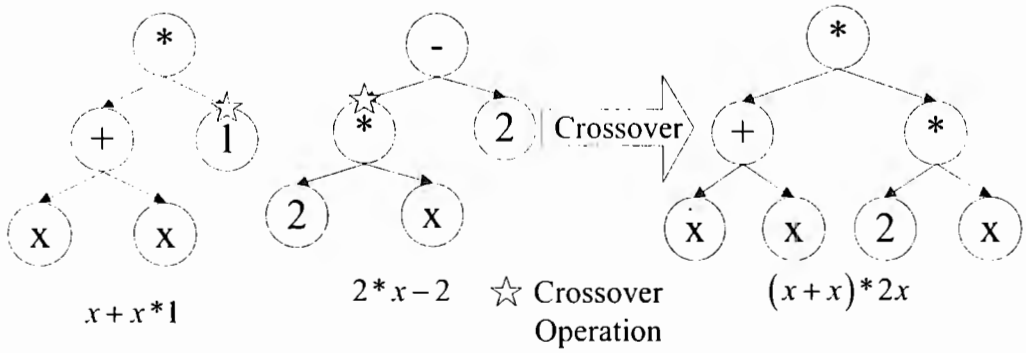


Figure 29. GP crossover operation.

Continuing with the example previously proposed, the result of a crossover operation of the initial population would be that of the Figure 29 where the branches of the individuals are interchanged.

3.5.1.2. Genetic Programming Mutation

There are several forms of mutation in genetic programming. It is also compared to a mutation probability which is regularly very low otherwise it will cause chaos in the population. When the mutation operation is to be performed, one of the branches of the individual is mutated generating a new form in the branch and a new individual. This operation is illustrated in Figure 30, which shows a mutated individual of the example.

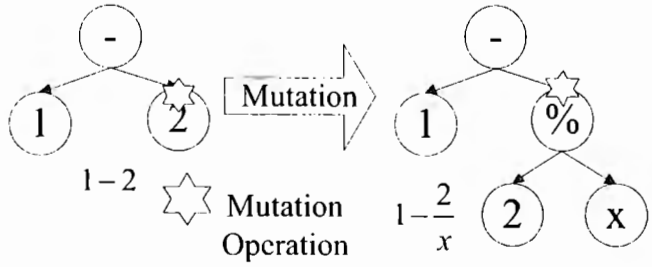


Figure 30. GP mutation operation.

After the individuals are crossed and mutated, they have to be evaluated again to measure their fitness. The loop of crossover and mutation must continue until a certain condition is fulfilled, like a number of generations or because it is manually stopped, something similar to the genetic algorithm.

4. Intelligent Speed Control of IM Drives

In this chapter the control system for the induction machine drives is described and implemented as shown in Figure 31, which shows a simplified DTC scheme for IM with closed loop speed controller.

The closed speed loop is necessary for an optimal control of the motor speed. In the diagram, the speed controller is highlighted in a block with the broadest line. This speed control will regulate the electromagnetic torque reference of the DTC loop thus, indirectly regulating the speed.

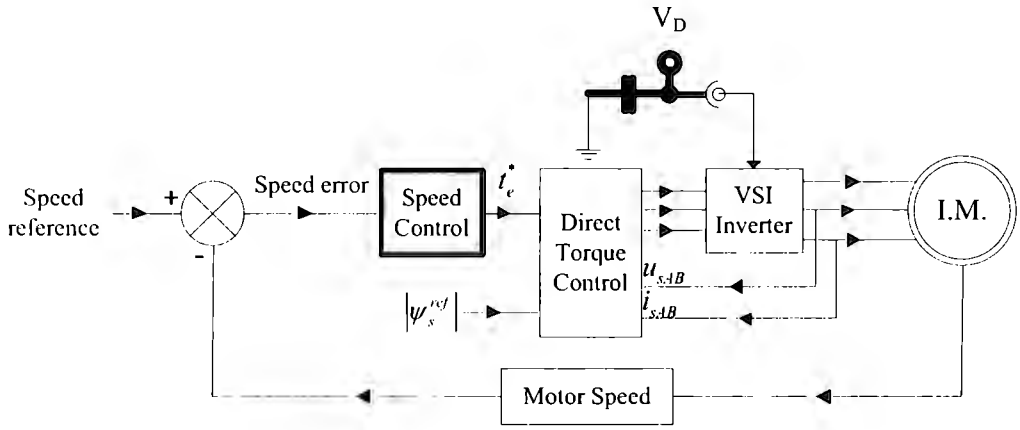


Figure 31. Simplified DTC for IM Drive with closed loop speed controller.

Several controllers were developed and implemented (shown in Table 20) they are used in the speed control block shown in Figure 31. A conventional fuzzy Proportional Integral Derivative -PID- controller is the basis to be compared against the other intelligent control techniques all further explained in the following sections.

Table 20. Control techniques used for speed control of the IM

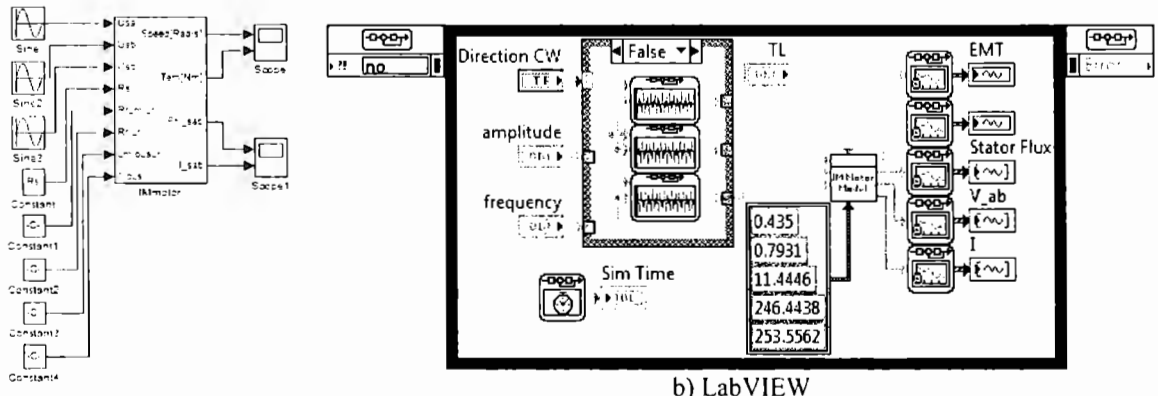
Fuzzy PID Controller
Fuzzy PID Genetically Enhanced Controller
Neuro-Fuzzy PID Controller
Adaptable Fuzzy PID Controller

The implementation of the intelligent controllers was made in LabVIEW, with the aid of the Intelligent Control Toolkit for LabVIEW [51].

4.1. Conventional DTC for IM Drives

The conventional direct torque control for induction machines (theory presented in section 2.2.2) was implemented in Simulink [61] and LabVIEW [37] both graphical simulation systems for non-linear systems, were differential equations can be written in the

form of blocks. The program of the motor model is based on the diagram shown in Figure 6 and equations 60 to 69 described in section 2.2.1. The program for the Motor model in Simulink is shown in Figure 32.a and in LabVIEW program is shown in Figure 32.b.



a) Simulink b) LabVIEW

Figure 32. IM Motor Model simulation programs.

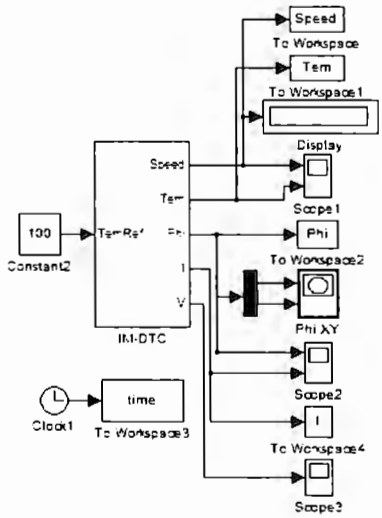


Figure 33. DTC Simulink Simulation.

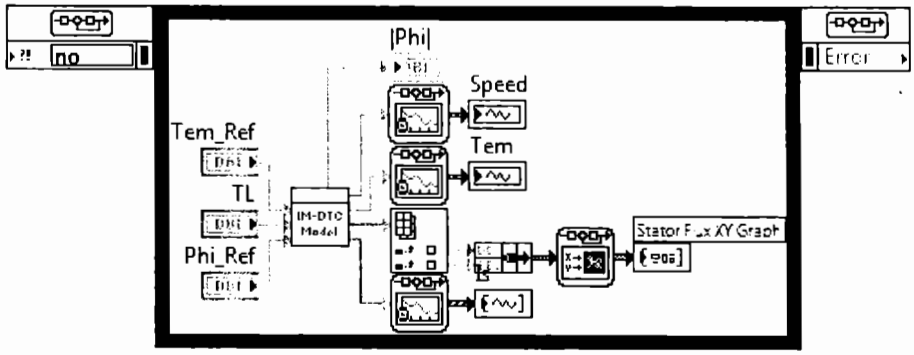


Figure 34. LabVIEW DTC Simulation.

The program of the DTC made in Simulink is shown in Figure 33. Next the implementation of conventional DTC in LabVIEW was done; program is shown in Figure 34. The tests and results of the simulations made with the programs presented in this section are presented in section 5.1.

4.2. Fuzzy PID Controller

This controller (diagram shown in Figure 35) is based on the non-interactive improved PID version, as previously specified in section 2.3. The advantage of designing the controller this way results in a smaller controller, easily transferable to an embedded system. The control law is implemented with $\dot{u} = k_i e + k_p \dot{e} + k_d \ddot{e}$ and the correction sent to the plant with eqn (88).

$$u = \dot{u} + \Delta u \qquad \text{eqn (88)}$$

Another reason to implement it this way is that a difference gives more information because they represent a reason of change. Digital systems saturate integrators and differences can be easily computed with subtractions. Each of the gains is based in a Mamdani fuzzy system, the diagram can be seen in Figure 36 and the theory is explained in section 3.1.2.

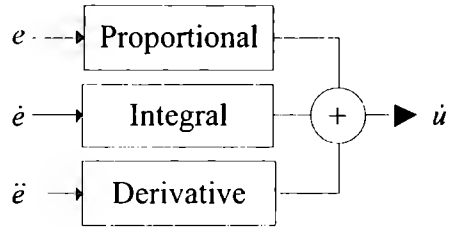


Figure 35. Fuzzy PID Controller

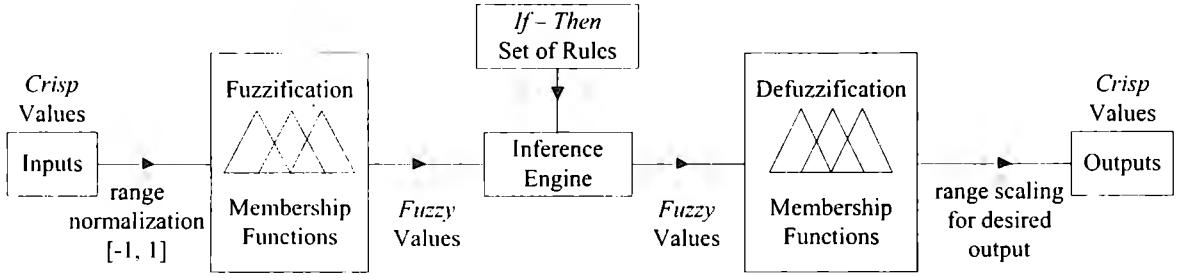


Figure 36. Fuzzy PID Controller component gains, P, I, D.

The inputs of the controller are normalized to the [-2,2] region. The fuzzy outputs of the controller are also in this range but scaled back to the desired range when defuzzified. Each of the fuzzy systems that represent a gain has three triangular membership functions for inputs and outputs. For each gain -proportional, integral and derivative- a system with three rules was created, generating a total of nine rules. The rules are shown in Table 21, the form of the rule is:

IF input is membership_function THEN output is membership_function.

Table 21. Inputs, outputs and fuzzy rules for Fuzzy PID

Inputs	**Outputs** for the three different Gains		
	Positive	**Integral**	**Derivative**
Negative	Positive	Negative	Positive
Zero	Zero	Zero	Zero
Positive	Negative	Positive	Negative

Once the response of each gain is calculated, the output is summed as shown in Figure 35 and the control law sent to the plant is calculated with eqn (88).

4.2.1.1. Implementation of Fuzzy PID Controller

The inputs are normalized by dividing them by 377, which is the top speed that the rotor can attain in radians per second. The output is scaled back with a constant of 200. The values for these constants are set empirically based upon knowledge from the user; the simulation program is shown in Figure 37.

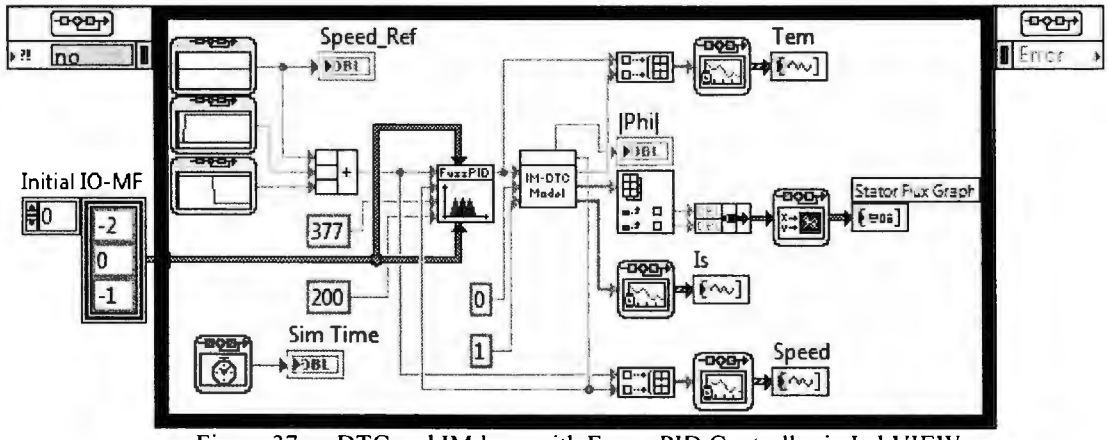
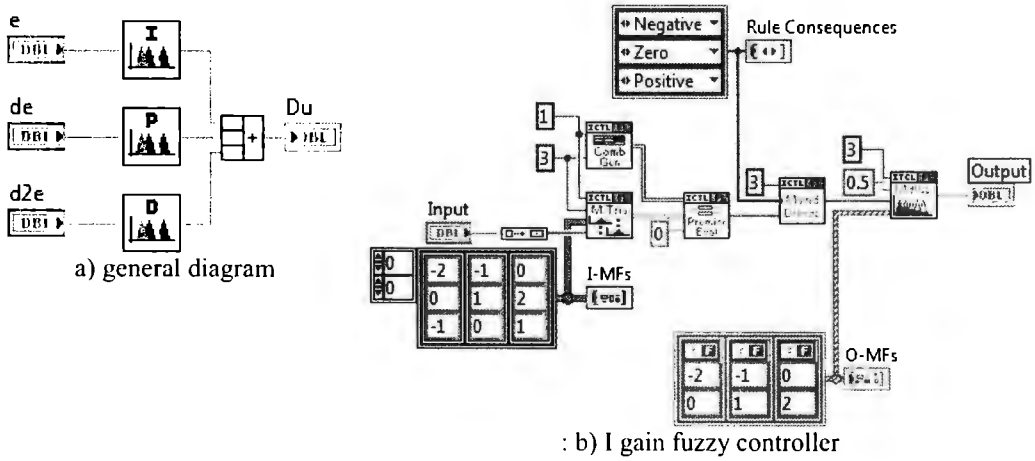


Figure 37. DTC and IM loop with Fuzzy PID Controller in LabVIEW.

The block diagram of the controller can be seen in Figure 38.a. In Figure 38.b the block diagram for a gain is presented. All of the gains have the same structure only the defuzzification membership functions rules change according to the information presented in Table 21. Results for these controllers are shown in section 5.2.1.



a) general diagram

: b) I gain fuzzy controller

Figure 38. Fuzzy PID Controller in LabVIEW.

4.3. Fuzzy PID Genetically Enhanced Controller

The previous fuzzy PID was genetically enhanced in different parts of the system, as it can be seen in Figure 39. Specifically the enhanced parts were: the input and output normalization constants, the limits of the membership functions for inputs and outputs and the consequences of the fuzzy rules.

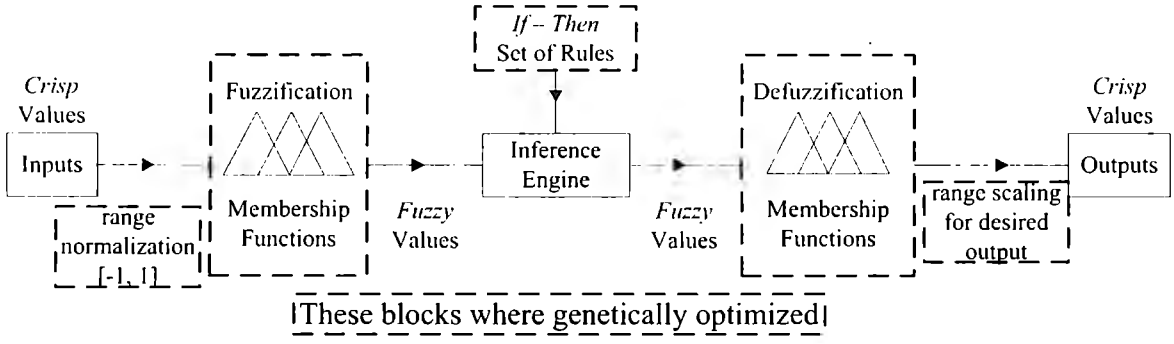


Figure 39. Fuzzy PID Controller Genetically Enhanced.

Enhancements were performed using first genetic algorithms and later genetic programming. With GA's, optimizations of the fuzzy controller are made first to the normalizing constants and second to the limits of all the membership functions. Using GP, the optimizations were first made to the normalizing constants, using a more detailed and complex approach that will be explained in future sections and later optimization to the consequences of the set of rules.

When an optimization process was performed, the other elements on the controller remained constant. After optimizations were successfully accomplished the new enhanced elements were used in the controller and considered in future optimizations processes. The optimizations obtained by the genetic algorithms and genetic programming on the scaling factors are also used in the controllers proposed on the next sections.

The fitness is measured with the mean squared error calculated with eqn (89); which is the reference speed minus the real speed squared, all of these samples summed and divided by the number of samples. The number of samples is the inverse of the step simulation time multiplied by the total simulation time eqn (90).

$$Fitness = \frac{\sum\left(Speed_{ref} - Speed_{real}\right)^2}{\#Samples} \qquad \text{eqn (89)}$$

$$\#Samples = \left(StepSimTime\right)^{-1} \cdot TotalSimTime \qquad \text{eqn (90)}$$

4.3.1.1. PID Genetically Enhancements: Scaling Factors

Normalization factors are very important since they dictate what portion of the decision table is used. They change membership functions uniformly over input/output domains, changing the controllers gain over the whole domain uniformly.

The scaling factors (Figure 40) were codified in different forms: with 8 bit unsigned integers, later with 16 bit unsigned integers to make the range of search wider. These scaling factors move the effect of each gain over the entire range.

Figure 40. Scaling Ranges for Inputs and Outputs.

4.3.1.2. PID Genetically Enhancements: Rule Consequences

The consequences of the rules in the fuzzy controllers are optimized by genetic algorithms and genetic programing methods. The originally proposed consequences for the rules are shown in Table 21; the fuzzy clusters used in this consequences -*Negative*, *Zero* and *Positive*- can be changed.

Each rule consequence is assigned a number: 0.- *Negative*, 1.- *Zero* and 2.- *Positive*; then this numbers are coded using 2 bit unsigned integers. As it can be seen in Table 22 the consequences may change to increase the performance of the controller.

Table 22. Enhancement example of Fuzzy Rule

Inputs	**Outputs** for the three different Gains		
	Positive	**Integral**	**Derivative**
Negative	Positive	Negative	*Negative*
Zero	Zero	*Positive*	Zero
Positive	Negative	Positive	*Positive*

4.3.1.3. PID Genetically Enhancements: Membership Functions

The original membership functions are shown in Figure 41 (on the left side). The *a* limit is where the triangle begins, the *c* limit is where the triangle value is 1 and the *b* limit is where the triangle ends.

The original membership functions are shown at the left of Figure 41 and in Table 23. A constant is added to the limits of these functions: a 8 bit signed integer which is previously scaled down from a range of [-127, 128] to a smaller range like [-0.5, 0.5] or [-1, 1]. In this way the original membership functions limits and shapes can be changed by the optimization technique.

Table 23. Original Membership Functions

Function	Limits		
	a	**c**	**b**
Negative	-2	-1	0
Zero	-1	0	1
Positive	0	1	2

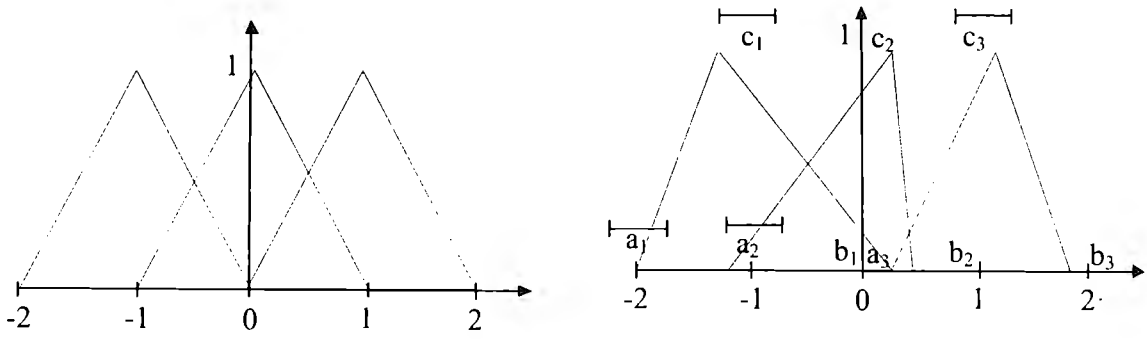


Figure 41. Membership Function codification in Genetic Algorithm.

4.3.2. Implementation of Fuzzy PID Genetically Enhanced Controller

The fuzzy PID is genetically enhanced in order to optimize its performance, as previously explained, the first optimizations were made using genetic algorithms and later using genetic programming; in this section the implementation process is explained; results for this controller are presented in sections 5.2.2 and 5.2.3.

4.3.2.1. Enhancements using Genetic Algorithms

The optimization was executed in different stages; eqn (91) and eqn (92) show the form of the individuals. In eqn (91), the numbers can be 8 or 16 bit unsigned integers. These individuals represent the input and output scaling factors. During this optimization stage the membership functions do not change.

$$\{input_{scale}, output_{scale}\} \qquad \text{eqn (91)}$$

$$\{a_1, c_1, b_1, a_2, c_2, b_2, a_3, c_3, b_3\} \qquad \text{eqn (92)}$$

Later the forms of the membership functions were enhanced, as shown in eqn (92). The numbers correspond to the constants that are added to the original limits of the membership functions and are signed integers. The values of the functions are varied in different ranges (Table 24). Using 8 bit or 16 bit signed integer numbers and divided by a scaling constant so the range is in the membership functions area; this has been explained more carefully in section 4.3.1.3.

During this optimization stage the already optimized scaling factors were used, remaining constant during the optimization process. Finally Figure 42 shows the block diagram of the program made in LabVIEW.

Table 24. Membership Functions Ranges.

Range	Scaling Factor
[-0.5, 0.5]	256
[-0.67, 0.67]	192
[-1, 1]	128

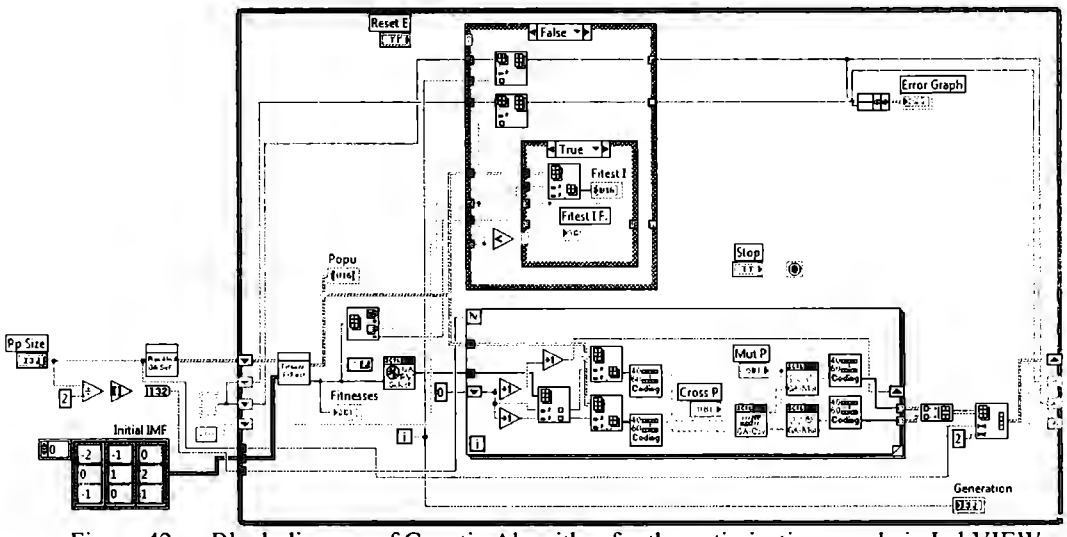


Figure 42. Block diagram of Genetic Algorithm for the optimizations made in LabVIEW.

4.3.2.2. Enhancements using Genetic Programming

With genetic programming the optimization process for scaling factors was made more specific, using a constant for each gain in the PID controller. The selection method used was tournament, along with different sizes for the number representation. The individual represented by a tree is shown in Figure 43. It is basically a tree with four branches; each one represents a scaling factor for the three inputs and one output.

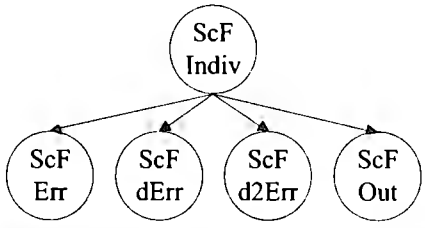


Figure 43. Scaling Factor tree form.

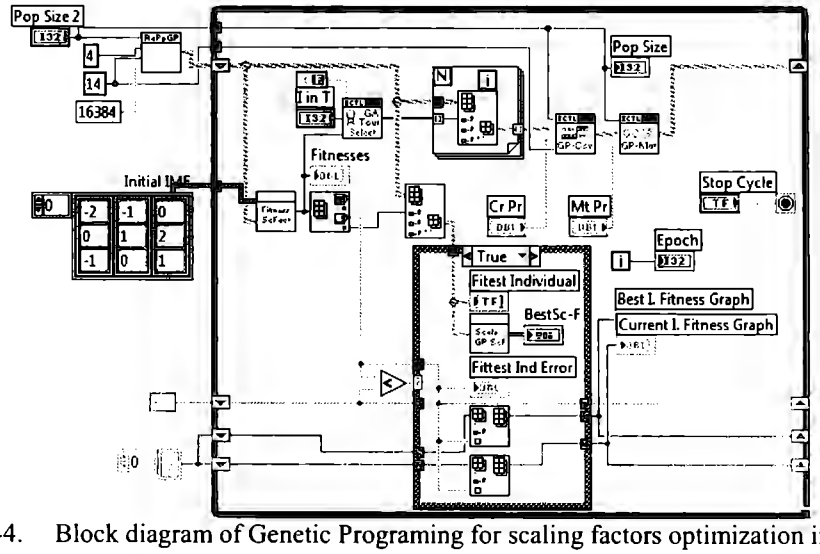


Figure 44. Block diagram of Genetic Programing for scaling factors optimization in LabVIEW.

These inputs and outputs are then scaled back with their own optimized normalizing constant. The block diagram for the program is presented in Figure 44. The optimization of the consequences of fuzzy rules was performed after the optimization of scaling factors. The block diagram is similar to Figure 44, just with a different fitness function to evaluate rules performance. The individual represented by a tree is shown in Figure 45.

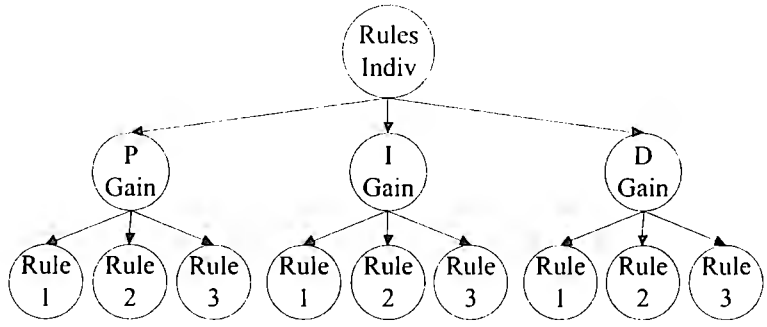


Figure 45. Rule optimization consequences tree form.

4.4. Neuro-Fuzzy PID Controller

The same fuzzy controller shown in Figure 35 is programed with the Neuro-Fuzzy system scheme. The optimizations obtained by the genetic algorithms and genetic programming on the scaling factors are also used in this controller. With the FCM algorithm, this controller is able to adjust the form of its membership functions online.

The theory of this controller is described in section 3.3.1 the diagram of the controller is shown in Figure 46. The original neuro-fuzzy system is based on the Takagi-Sugeno inference system. For this application the fuzzy system was based on the Mamdani inference system. The input and output membership functions are tuned with the FCM algorithm and can be of any form, since T-ANN's can be adjusted to any kind of shape as shown in Figure 47.

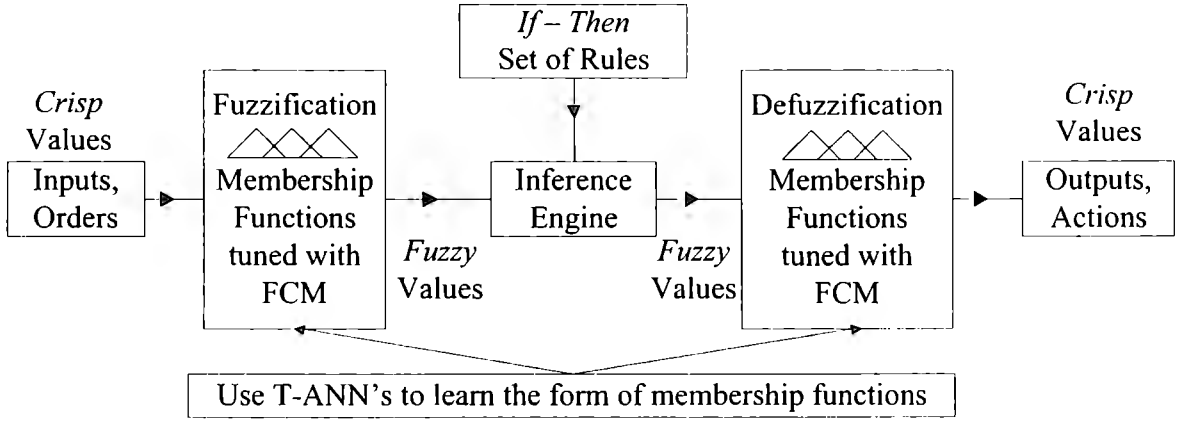


Figure 46. Neuro-Fuzzy PID Controller

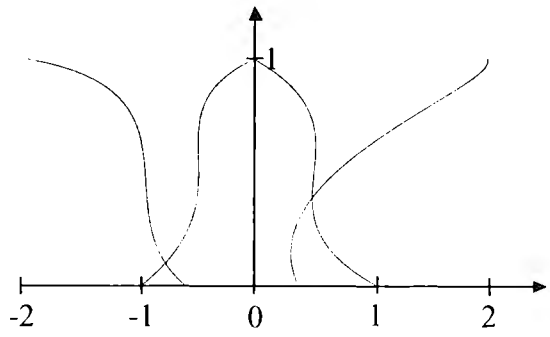


Figure 47. Fuzzy membership functions tuned with FCM and learnt with TANN's

4.4.1. Adjusting Membership Function for Response Optimization

The form of the membership functions can be updated and optimized by the use of the FCM algorithm. The form if the input membership functions are maintained during the execution of the controller, so the action of the controller is not changed and erroneous response is obtained.

If the form of the input membership functions is changed, an error with a crisp value of zero may have a membership value on the fuzzy zero set. By altering the form of the membership functions it may end up having a membership value on the positive set, even though it is explicitly known to be negative.

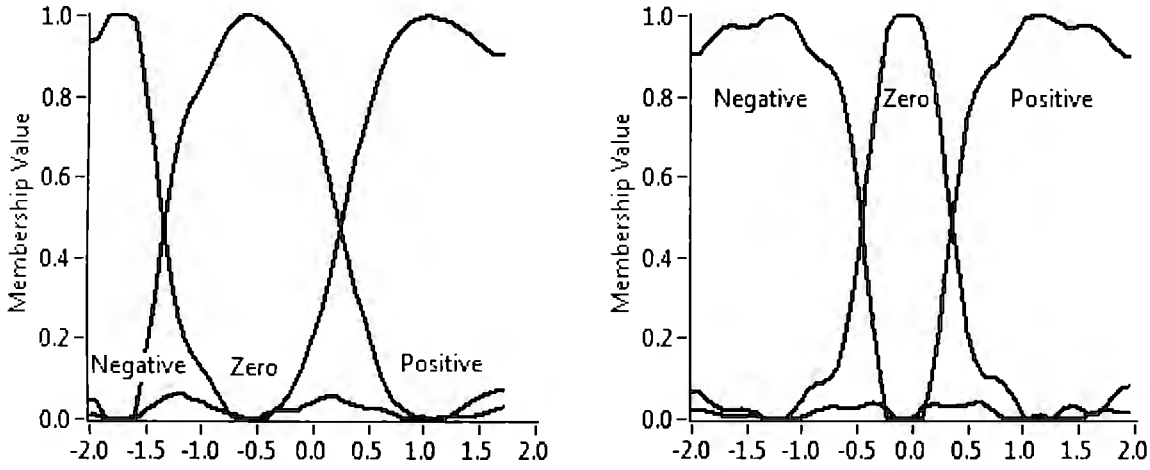
However, adjusting the form of the output membership functions will have a positive impact, since it will take action on the crisp output response of the controller. The response can be then optimized depending on the input response, and for that a simple algorithm is proposed and explained in Table 25.

Table 25. Algorithm for the adaption of membership functions for output optimization response

1. Generate a set of data in the range of action of the membership functions
2. Eliminate data from the initial set around certain point
3. Adjust the form of the membership functions using FCM and T-ANN's

The limits of the membership functions are updated as follows, according to the algorithm in Table 25. Because the range of operation of the membership functions is known: [-2,2], a set of data points in that range is generated. Next, elements around a certain point in the range are eliminated and the set is feed to the FCM algorithm, which will return the form of the fuzzy sets.

The algorithm can be seen applied in Figure 48 around the zero crisp value, the result is that the *Zero* fuzzy set is smaller while the *Negative* and *Positive* fuzzy sets are larger. This way output fuzzy membership functions are FCM tuned and the response of the controller can be optimized during the execution of the controller.



a) Membership functions applying the algorithm around -1.5 crisp value

b) Membership functions applying the algorithm around 0 crisp value.

Figure 48. Fuzzy membership functions tuned with algorithm shown in Table 25.

4.4.1.1. Implementation of the Neuro-Fuzzy Controller

The program of the Neuro Fuzzy controller is in Figure 49, based on the theory previously explained. In this program two main loops can be identified, the left one shows the simulation of the controller and de DTC-IM model. The right loop executes the FCM algorithm and tunes the form of the membership functions according to the information presented in section 4.4.1. Tests and results for this controller are in section 5.2.4.

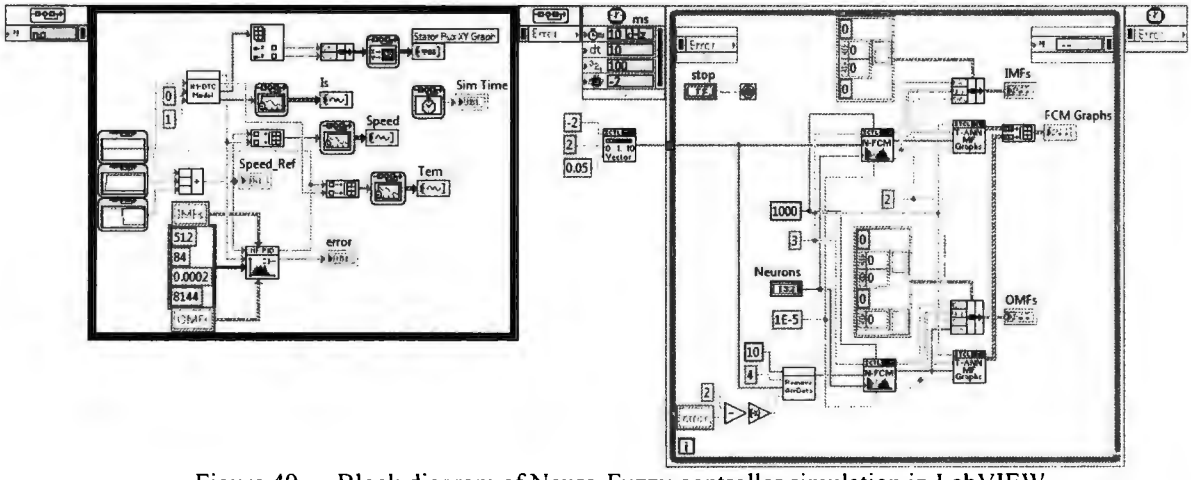


Figure 49. Block diagram of Neuro-Fuzzy controller simulation in LabVIEW

4.5. Adaptable Fuzzy PID Controller

This controller was proposed after the evaluation of the Neuro-Fuzzy controller, which was found to have some disadvantages for the control of speed in induction machines.

Although the scheme of the controller was originally proposed as a Neuro-Fuzzy system (Figure 24) it was found that for this application the use of the T-ANN's would only increase of computational time without any additional increase in performance. Instead, the

FCM algorithm is used and with this information the limits of the triangular membership functions are adjusted, the diagram of the controller is shown in Figure 50.

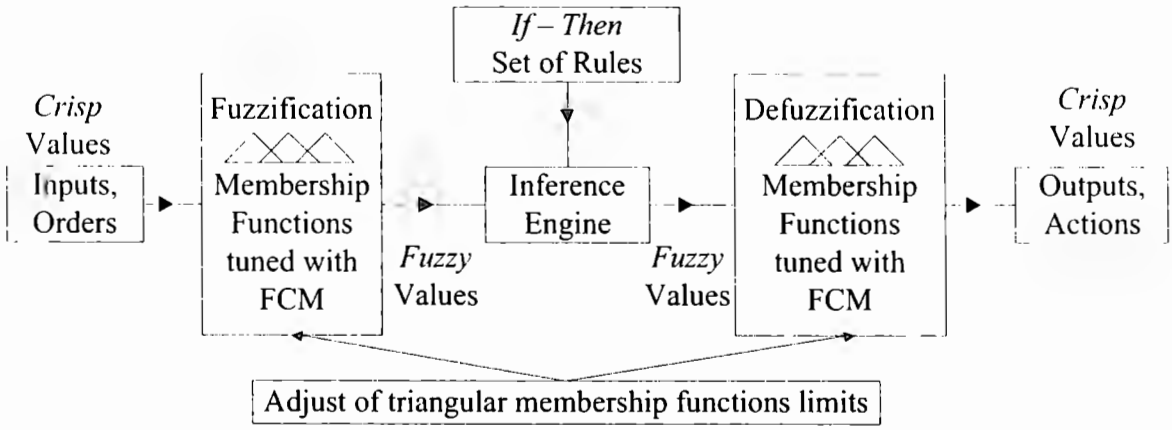


Figure 50. Adaptable Fuzzy PID Controller

Membership functions are updated following the methods explained in section 4.4.1 with the difference that their forms are not the same as the output of the FCM, but triangular shaped. The centers of the FMC clusters are used as the maximum value in the triangular membership functions.

Later, the points where the functions begin and end are detected, to set the limits of the triangular functions; this can be better seen in Figure 51. This way the T-ANN's are eliminated and the triangular membership functions are shaped similar to the fuzzy clusters obtained by the FCM algorithm.

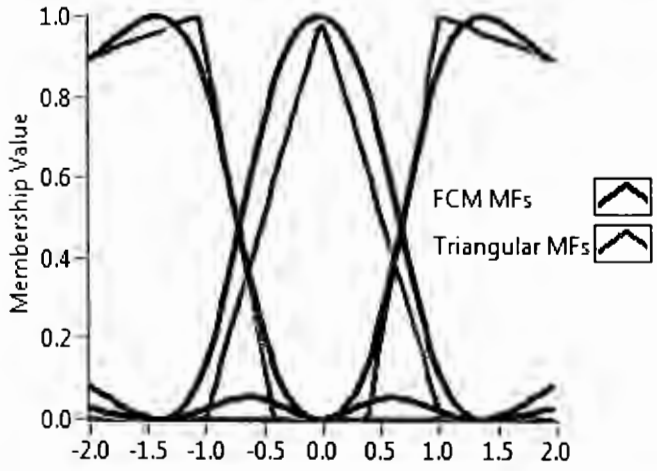


Figure 51. Membership functions for the Adaptable Fuzzy PID Controller

4.5.1.1. Implementation of the Adaptable Fuzzy Controller

By using adaptable triangular membership functions the control scheme was successful in regulating speed. The FCM system is still used but instead of using the fuzzy clusters obtained by the algorithm, only their limits are identified and triangular membership functions are used, program is shown in Figure 52.

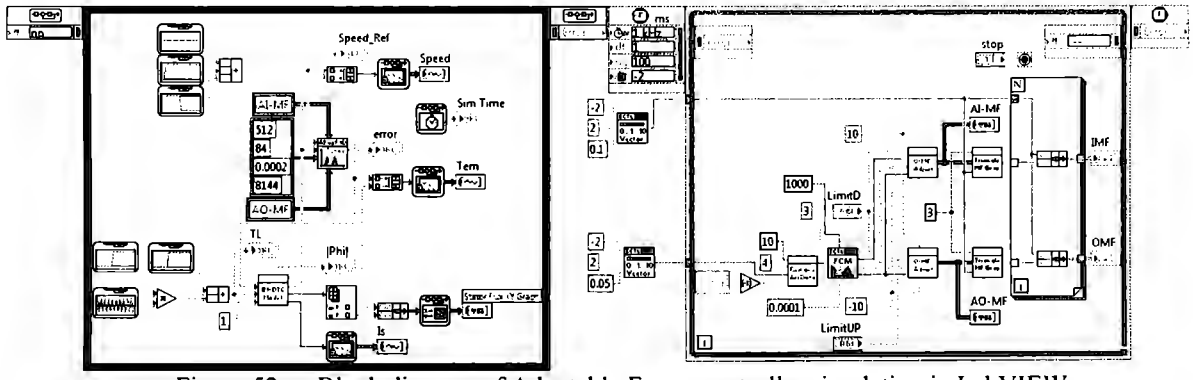


Figure 52. Block diagram of Adaptable Fuzzy controller simulation in LabVIEW

5. Simulations and Results

In this chapter the results obtained from simulations are reported. The main simulation environments are Simulink [61] and LabVIEW [37]. Simulations were first carried out in Simulink and later in LabVIEW [37]. Parameters consulted in [53] for the induction motor model used in the simulations are shown in Table 26.

Table 26. Parameters for motor simulations.

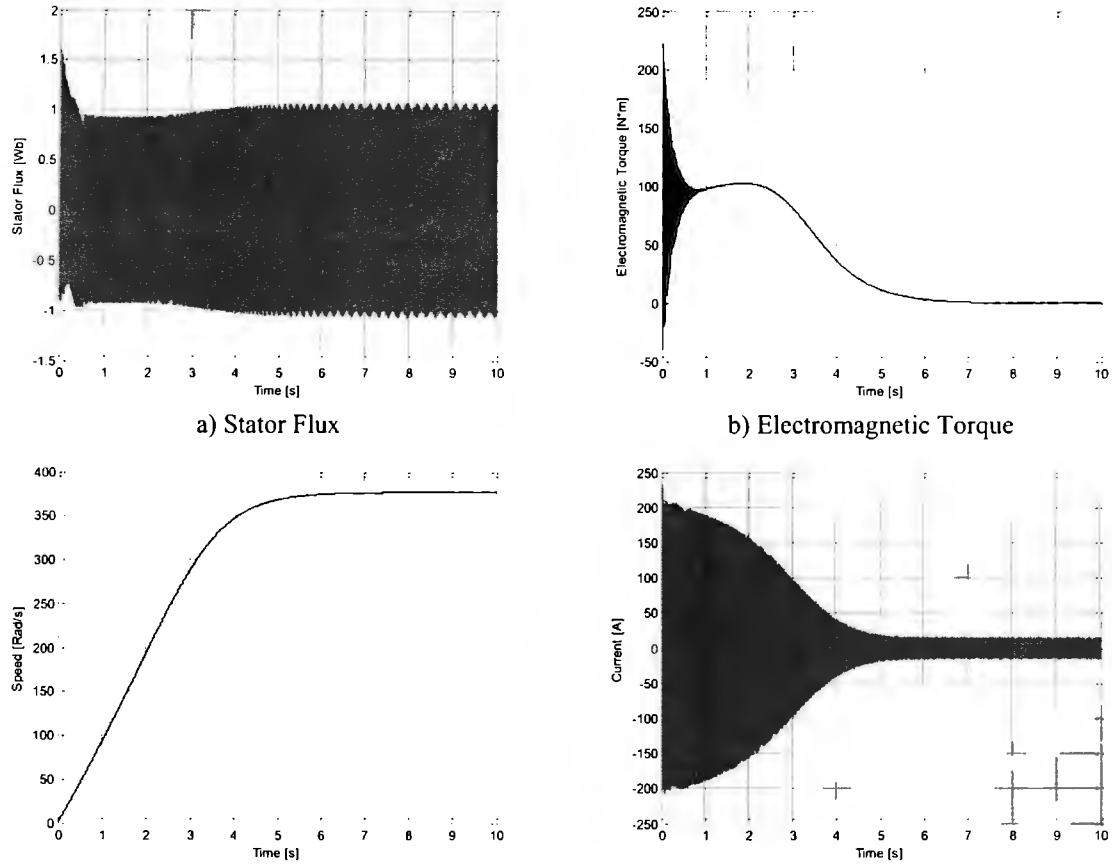
Parameter	Value
R_s	0.435 Ω
R_r	0.816 Ω
L_m	0.06931 H
L_s	$0.002 + L_m = 0.07131\ H$
L_r	$0.002 + L_m = 0.07131\ H$
σ	$1 - L_m^2 / L_s L_r = 0.05531$
T_m	1 Nm
T_l	0 Nm
f	60 Hz
Poles	2

5.1. Conventional DTC and Motor Model Simulations Results

In this section results are presented for the simulations for the model of a symmetrical squirrel cage induction machine in the two axis stationary reference frame. It is a vector model in D-Q reference frame presented in section 2.2.1. Conventional DTC, used for the regulation of torque and flux, was also programed and evaluated.

5.1.1. Simulink Motor Simulation

The model was the tested with three sine signals with an amplitude of 400 V, frequency of 60 Hz and 120 electrical degrees apart to simulate a three phase voltage source. In Figure 53 it can be seen the Stator Flux with a value closer to one. Electromagnetic torque start around 100 N*m and reach to zero. Speed go from zero to almost 400 rad/s or 3600 rpm and Stator currents go from peak of 200 [A] to 15 [A].



a) Stator Flux

b) Electromagnetic Torque

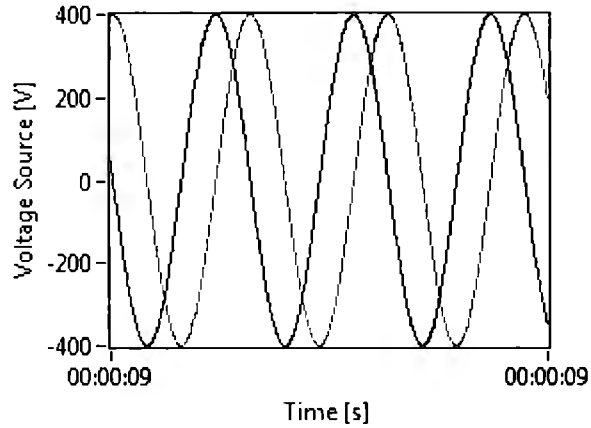
c) Speed

d) Stator Currents

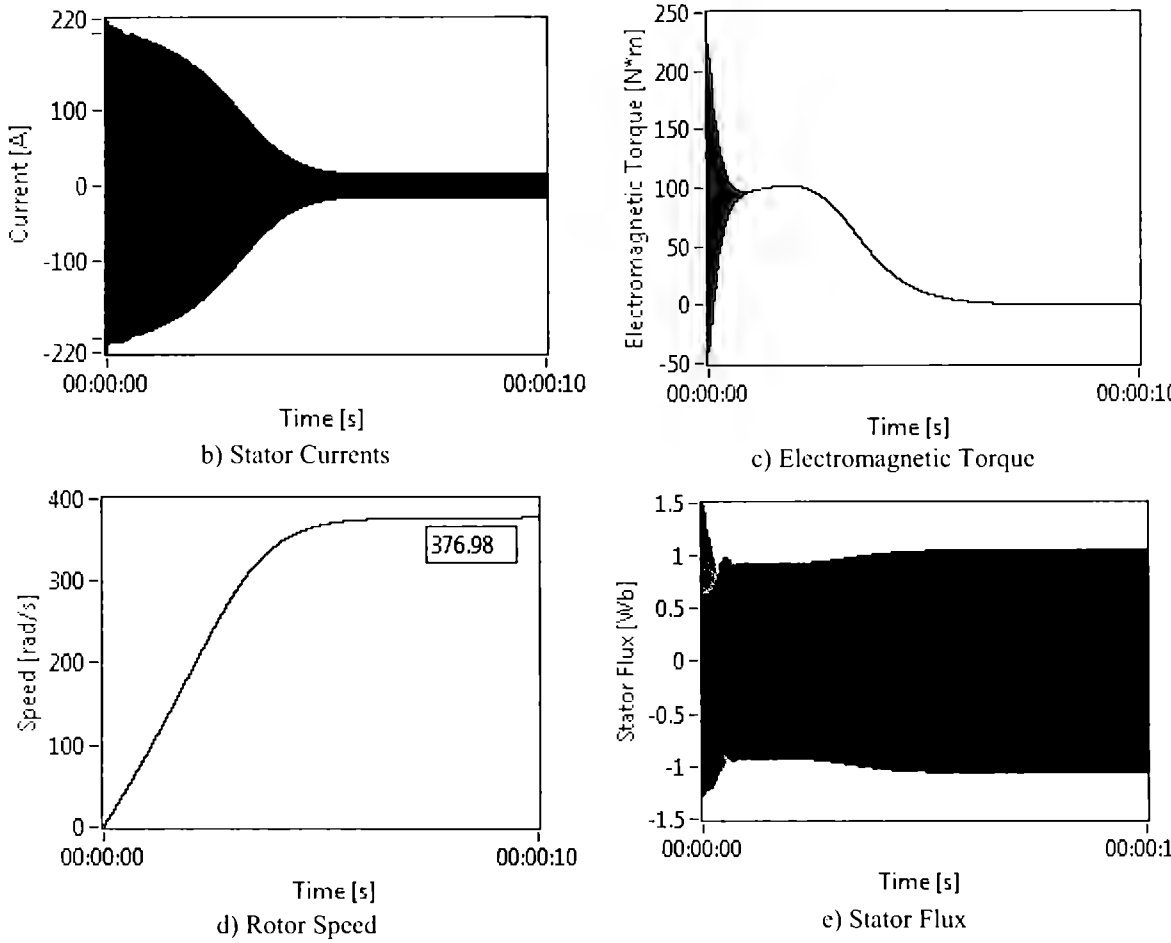
Figure 53. Simulink Motor Simulation.

5.1.2. LabVIEW Motor Simulation

The model programed was also tested with three sine signals of 60 Hz and 120 electrical degrees apart to simulate a three phase voltage source. In Figure 54 it can be seen the voltage in the stator in two axis reference. Stator flux with a value closer to one, Electromagnetic torque start around 100 N*m and reach to zero. Speed go from zero to almost 400 rad/s or 3600 rpm and Stator currents go from peak of 200 [A] to 15 [A].



a) Voltage Source



b) Stator Currents

c) Electromagnetic Torque

d) Rotor Speed

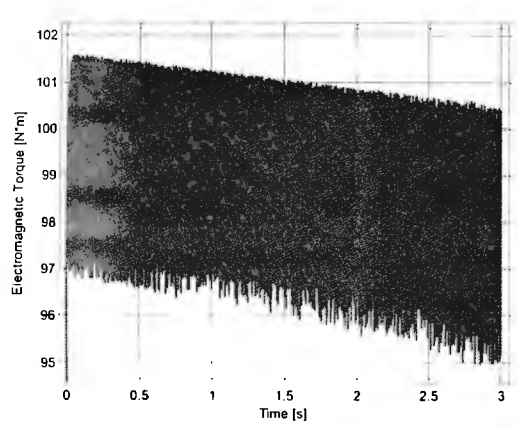
e) Stator Flux

Figure 54. LabVIEW Motor Model Simulation Results.

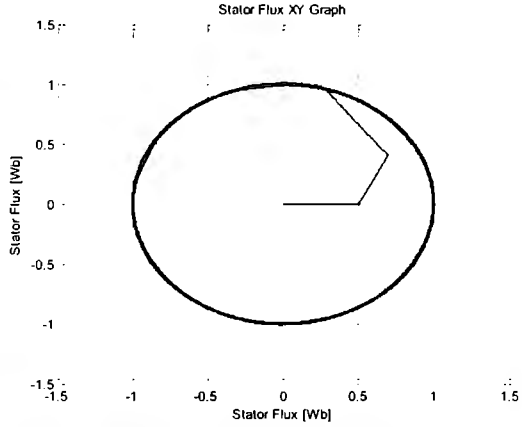
5.1.3. DTC Simulink Simulation

Simulations of the DTC in Simulink were carried out with the reference of the EMT was set to 100 *N*m* and Flux to 1 *Wb*. As it can be seen in the results in Figure 55 the DTC is able to control the EMT and Flux in the desired ranges (Figure 55.a and Figure 55.b), and as in almost every vector control technique, the plot of the flux in the complex plane looks like a circle.

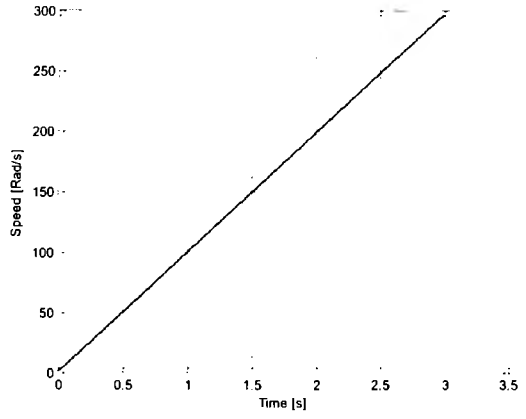
It can also be seen that speed (Figure 55.c), increases constantly from zero to 300 rad/s and the stator currents (Figure 55.d) remain constant, with an amplitude of approximately 125 [A] peak and the with a frequency inversely proportional to time.



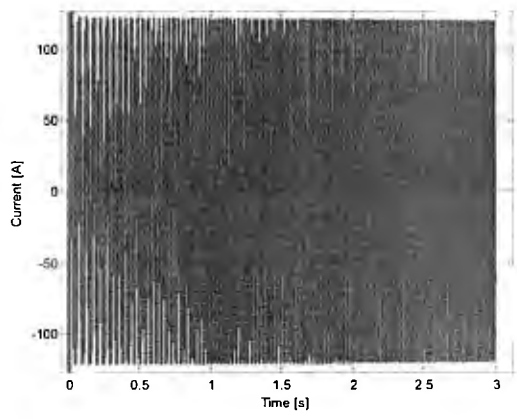
a) Electromagnetic Torque



b) Stator Flux in xy form



c) Speed

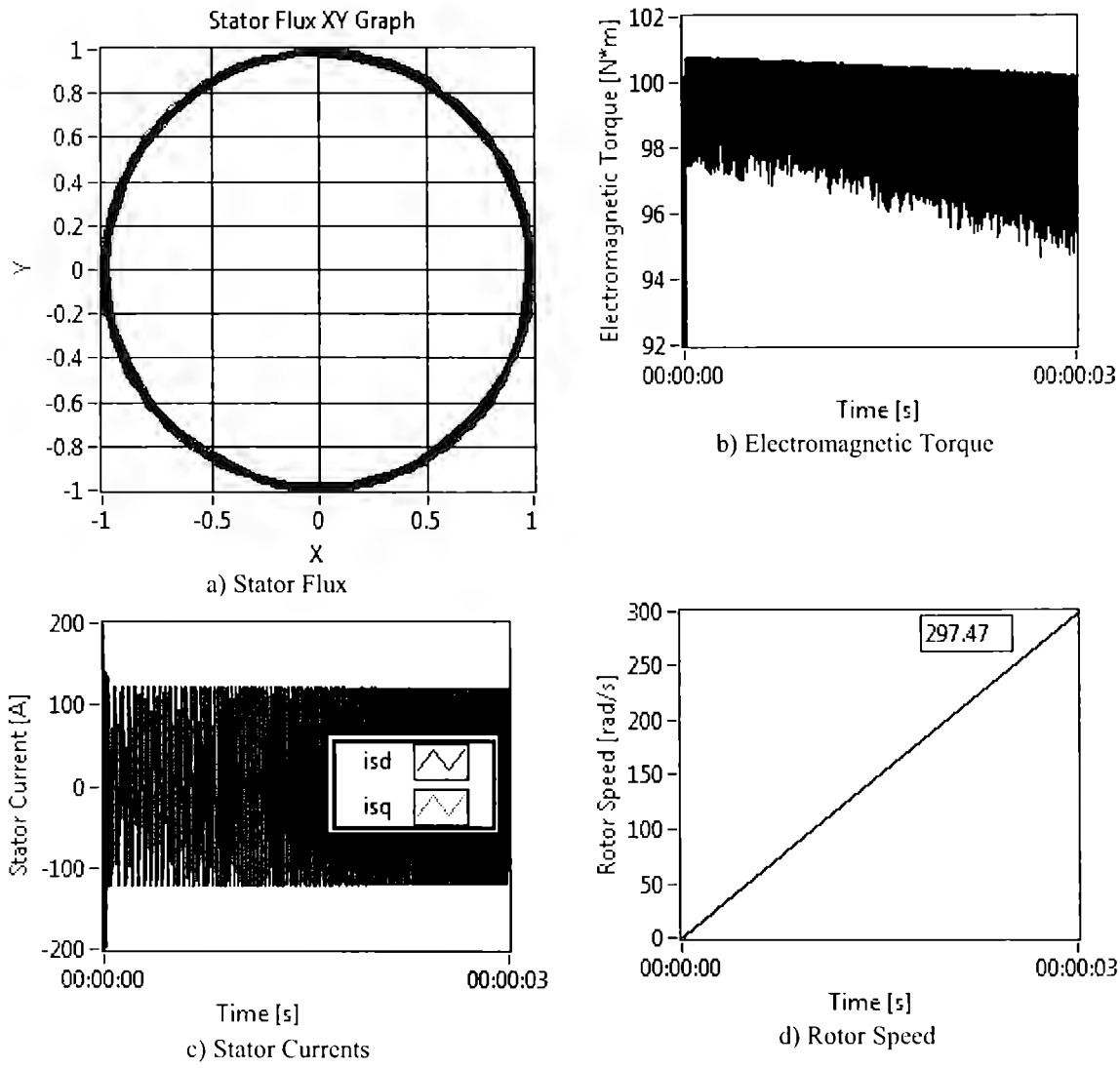


d) Stator Currents

Figure 55. DTC Simulink Simulation Results.

5.1.4. LabVIEW DTC Simulation

The following step was the simulation conventional DTC in LabVIEW. The references of the DTC where set to 100 N*m in EMT and 1 Wb for stator flux. As it can be seen in figure Figure 56 the form of the stator flux in the complex plane is also circular as any other vector controls. EMT is maintained around the reference with some slight decrement, the stator current has peak amplitude of approximately 125 A. And the frequency increases with time and finally the rotor speed increases to almost 300 rad/s with a constant slope.



a) Stator Flux

b) Electromagnetic Torque

c) Stator Currents

d) Rotor Speed

Figure 56. LabVIEW DTC Simulation Results. a) Voltage Source.

5.2. Simulations and Results of Intelligent Speed Controllers

In this section the results of the intelligent speed controllers previously presented in chapter 4 are presented. The simulations had different speed references as can be seen in Table 27, for a total of 10 seconds of simulation time. The solver is Euler with a fixed step time of 5E-5 sec.

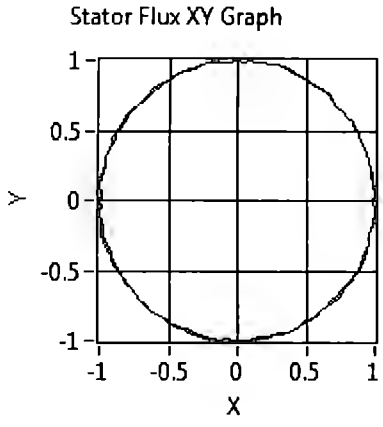
Table 27. Speed references test.

Time Range [s]	Speed [rad/s]
[0, 1]	50
[1, 5]	300
[5, 10]	100

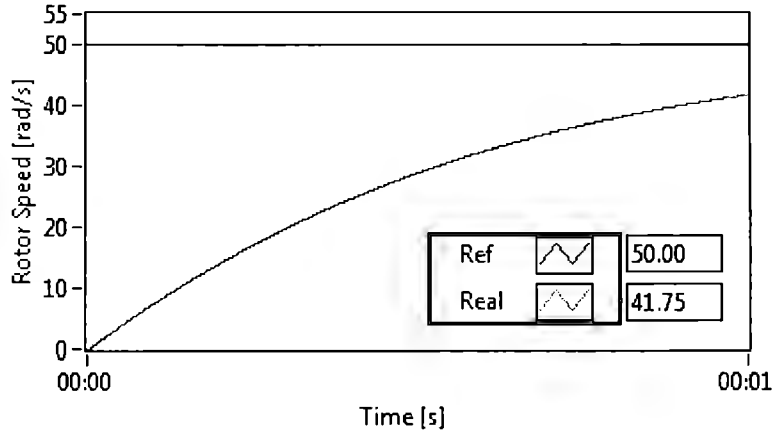
5.2.1. Simulation of Fuzzy PID Controller

As previously explained in section 4.2.1.1 the input normalization constant is 377 and the output constant is 200. In Figure 57.a the circular form of the stator flux is presented which is maintained during the whole simulation.

Figure 57.b shows the first section of the speed reference and response, which is 50 rad/s for 1 second; it can be seen that the controller is not able to take the motor to that speed reference. The next step reference lasts four seconds the response can be seen in Figure 58.a, the controller is able to take the reference to almost the reference it reaches 298.75 rad/s just at the end of the period.



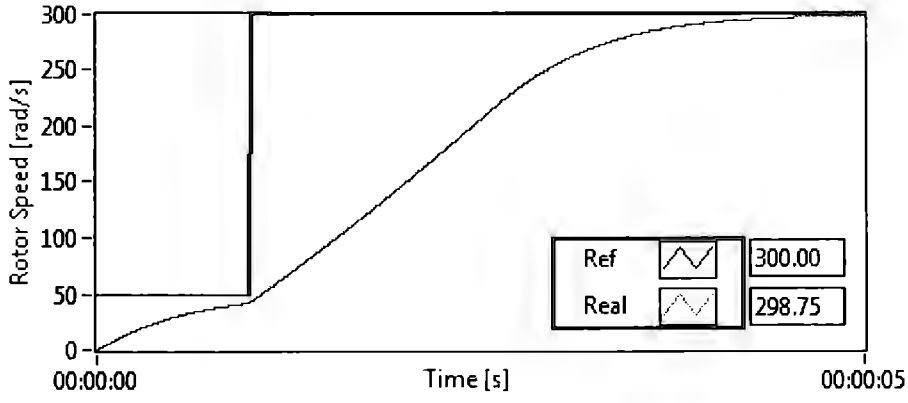
a) Stator Flux XY Graph



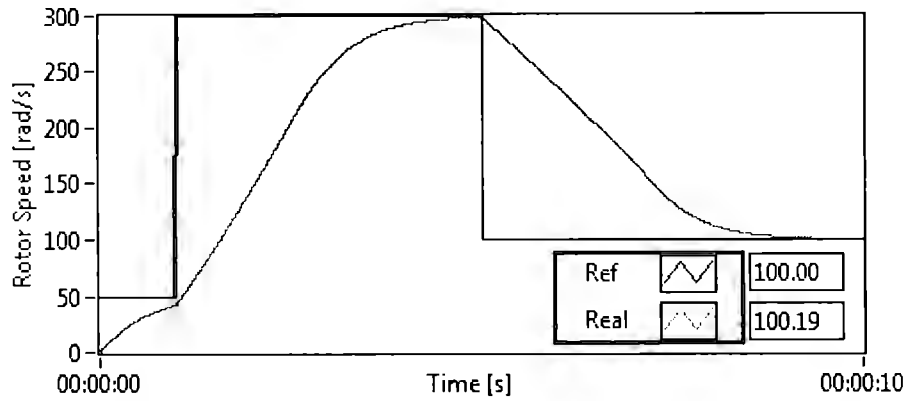
b) Speed Reference to 50 rad/s

Figure 57. Results for Fuzzy PID.

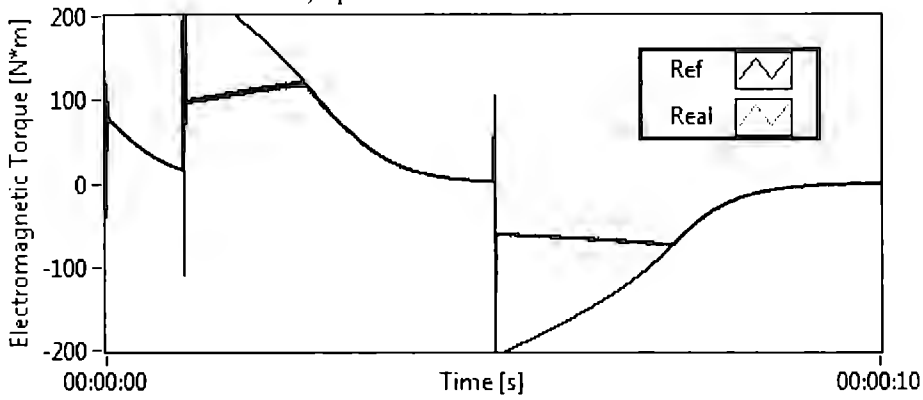
In Figure 58.b the controller lowers the speed of the machine from 300 to 100 rad/s, the controller is able to lower the speed to 100.19 rad/s. The overall response is smooth as it can be seen from the output of the EMT if the machine in Figure 58.c. When the speed of the machine is closer to the reference the response of the controller is even smoother, which causes the motor to not reach the desired output. Finally, it can be seen that the controller needs to be tuned to obtain a better performance.



a) Speed Reference to 300 rad/s



b) Speed Reference to 100 rad/s



c) Electromagnetic Torque response of the controller and the motor.

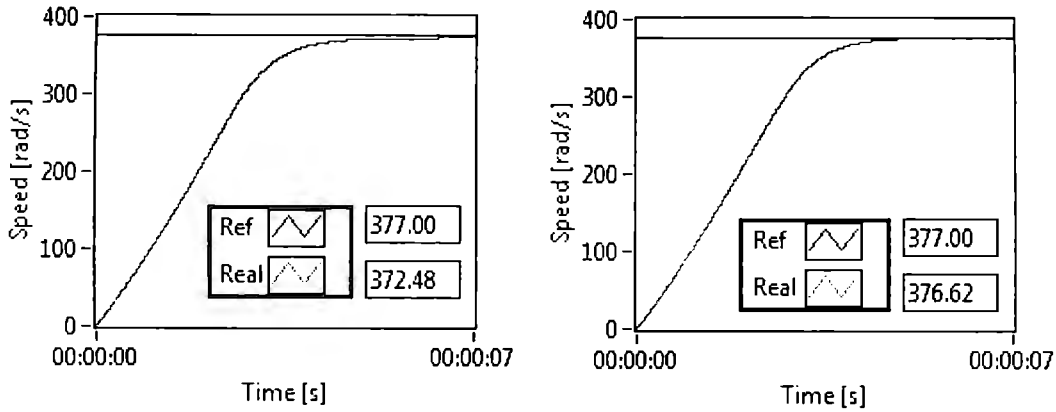
Figure 58. Results for Fuzzy PID.

5.2.2. Simulation of Fuzzy PID Enhanced by Genetic Algorithms

As presented in section 4.3 the optimization process using genetic algorithms was divided in two stages, first for the optimization constants and later for the limits of the membership functions; here the results will be presented in the same order. The detailed implementation of these programs is shown in section 4.3.2.1.

The fitness function used (more details in section 4.3) is a simulation of the induction motor model with the DTC control scheme and the fuzzy PID loop controlling the speed. The reference of the speed is 377 rad/s which converted to rpm is 3600 rpm's, the max speed of the machine. The simulation is executed for 7 seconds, with a step size of 0.2 milliseconds and solved with the Euler method; all of this is considered to decrease the simulation time.

As it can be seen in Figure 59, the rotor speed differs by some numbers but it remains accurate despite the broad step size. This is not considered a factor given that a smaller step time will mean a clear simulation, but the genetic algorithm will always optimize the controller and attempt to minimize the fitness value.



a) Step simulation time of 0.2 milliseconds b) Step time of 10 microseconds

Figure 59. Fitness function simulation, Reference and Real speed values.

On the first optimization stage the genetic algorithm was ran with a population of 20 individuals, a Crossing probability of 0.9 and a Mutation probability of 0.05, tournament selection with half the population -10 individuals- was used.

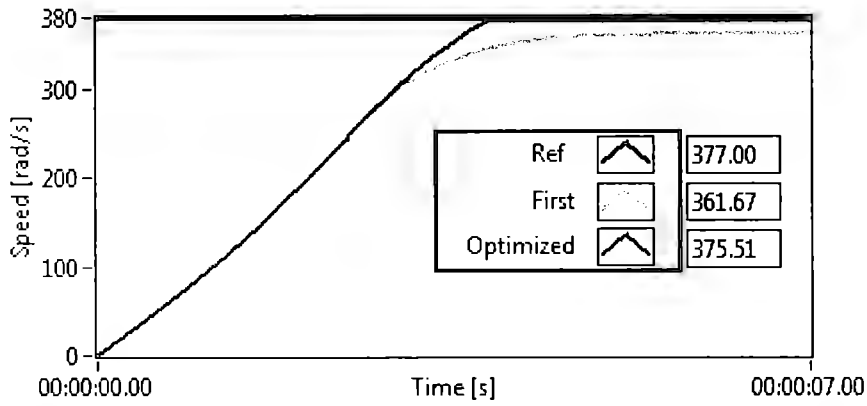
On Table 28 the results are shown for the different ranges of the scaling factors. The first values are the user defined factors presented in section 4.2.1.1, which after running the fitness function results in a value of 24674. After that, the GA was executed several times with different search ranges. It can be seen that the input scaling factor value is in 190's range, while the output scaling factor will always increase to the limit of the search range.

Table 28. Genetic algorithm results for input and output scaling factors optimization

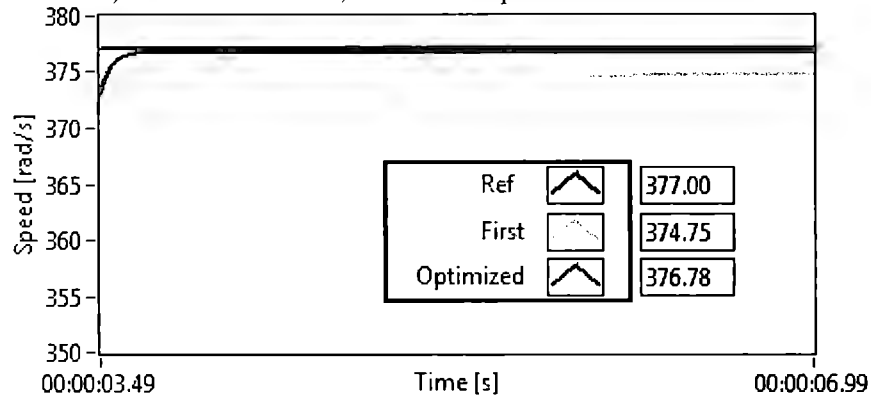
Search Range	Generations	Input Factor	Output Factor	Fitness
User Defined	---	377	200	24674
[0, 256]	21	189	256	24469
[0, 512]	32	192	512	24459
[0, 1024]	56	189	1024	24457.2

As it can be seen in Figure 60, the response from the controller with the enhanced values is better than the response of the values originally proposed. In Figure 60 images *Ref* means the speed reference of 377 rad/s, *First* is the response of the machine with the controller and the original scaling factors and *Optimized* is the response of the controller with the enhanced scaling factors obtained by the GA.

From the Figure 60.a and Figure 60.b it can be seen that the response is clearly optimized, the response of the controller with the original values is smoother this causes that the induction machine not to get to the desired speed reference. However, the response of the optimized controller will take the induction machine to the desired speed ranges without any problem.



a) 7 second simulation, simulation step time of 0.4 milliseconds



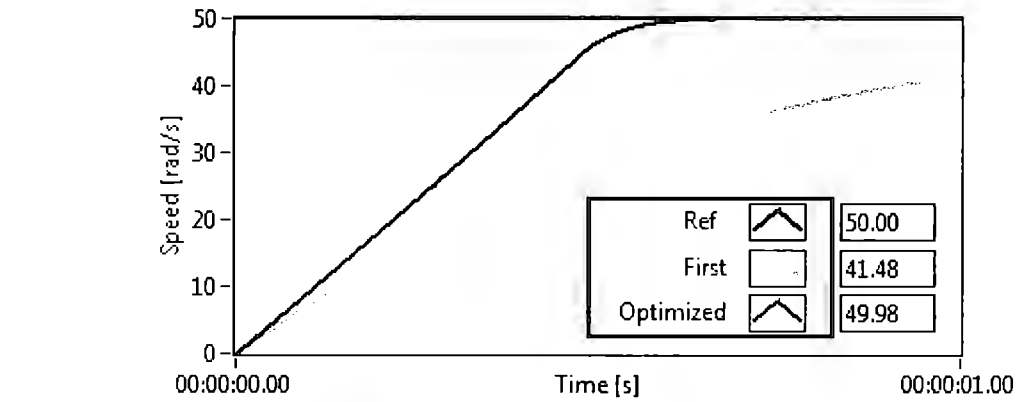
b) 3.5 seconds of the simulation, with step time of 50 microseconds

Figure 60. Fitness response of original and enhanced scaling factors values.

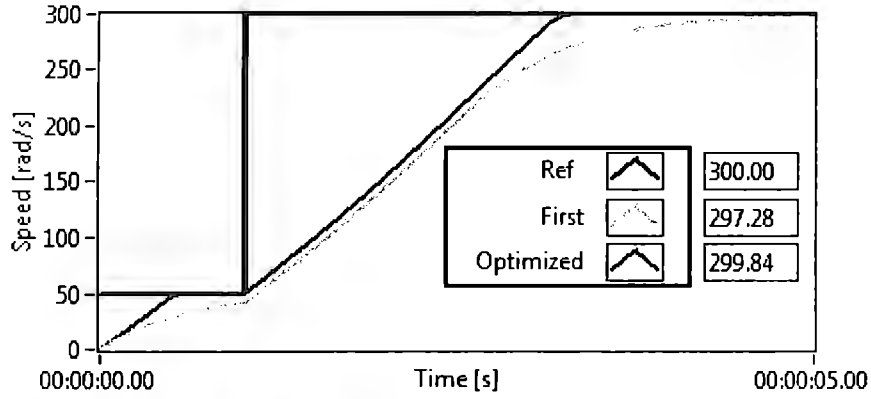
To compare with the same benchmark of Table 27, the response of the original fuzzy controller and the enhanced fuzzy controller were simulated. Figure 61 shows the response of the controllers in the different stages of the test. From Figure 61.a we can see that the original fuzzy controller is unable to take the machine to the desired reference, but the optimized fuzzy controller has no problem to take it to the reference and maintain it there.

Figure 61.b shows that both controllers are able to take the machine to the speed reference. The response of the original controller is smother and almost at the end of the speed range takes the motor to the desired reference while the enhanced controller has a faster response and can take the motor to the desired speed much more faster.

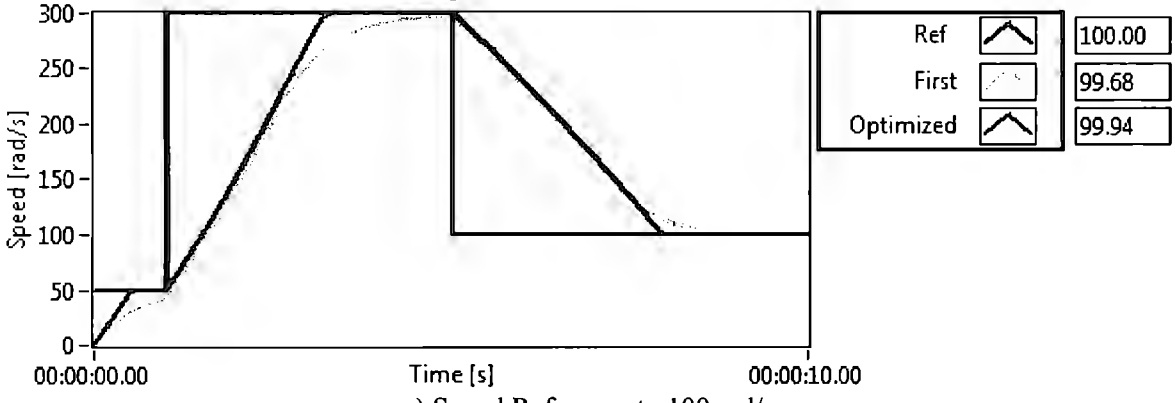
Finally Figure 61.c shows the response of the controller during the whole evaluation period; it can be seen that the Genetic Algorithm was able successfully improve the response of the controller, making the response of the machine faster.



a) Speed Reference to 50 rad/s



b) Speed Reference to 300 rad/s



c) Speed Reference to 100 rad/s

Figure 61. Results for Fuzzy and Enhanced Scaling Factors Fuzzy PID Controllers.

After the optimization of the scaling constants, they were set to their optimum values and the GA was executed to optimize the form of the membership functions. The GA was executed for several times, results are shown in Table 29. On the first run, after 142 generations with a scaling factor of 256 no improvements were found.

The fitness of the individual is 24457.2 as previously found with all its values in zero which shows that the search process becomes difficult. After later runs individuals with better response were found. The block diagram for this program is similar to the one in Figure 42, only the information used on the fitness function will change.

Table 29. Genctic algorithm results for membership function optimization.

Generations	Scaling Factor	Fitness
142	256	24457.2
168	256	24456.8
307	192	22999.7
200	128	23521.3

In Figure 62 the forms of the enhanced membership functions of the candidate with the best fitness (22999.77). It can be seen that the forms are unusual and something that would be hard to propose; the limits for each of the optimized candidates are shown in Table 30.

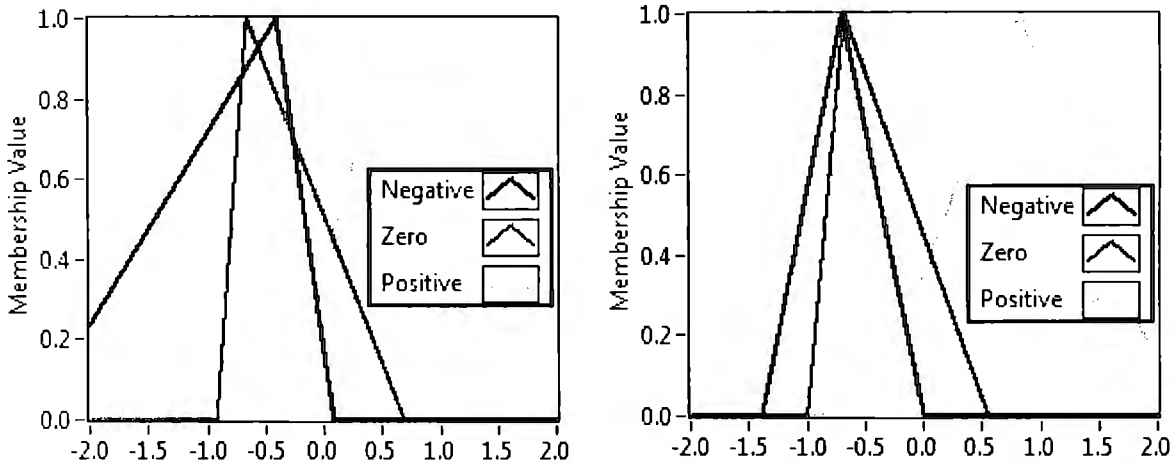


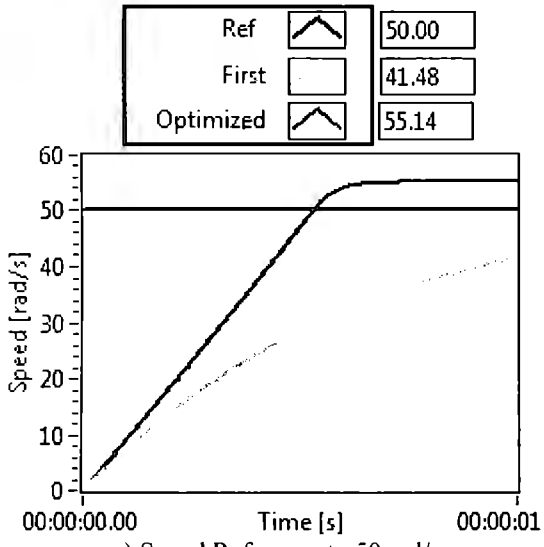
Figure 62. Enhanced forms of Membership Functions for individual with fitness 22999.7.

Table 30. Limits for enhanced membership functions with Genetic Algorithms

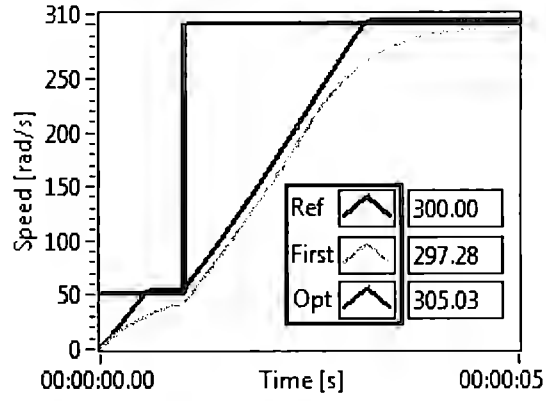
Fitness		Input MFs				Output MFs		
		Left	*Center*	*Right*		*Left*	*Center*	*Right*
22999.77	***Negative***	-2.5	-0.4	0.089	***Negative***	-1.4	-0.69	0.01
	Zero	-0.91	-0.66	0.69	***Zero***	-0.99	-0.67	0.56
	Positive	-0.31	0.57	1.6	***Positive***	-0.44	0.84	2.2
23521.3	***Negative***	-2	-0.062	0.0078	***Negative***	-1.9	-2	0.086
	Zero	-0.99	0.56	1.1	***Zero***	-0.91	-0.57	1.4
	Positive	0.078	0.98	1.8	***Positive***	0.44	0.85	1.2
24456.8	***Negative***	-2.3	-1	0	***Negative***	-1.7	-1.1	-0.00783
	Zero	-1	-0.42	1.2	***Zero***	-0.99	0.0039	1
	Positive	0.18	0.93	1.5	***Positive***	0.016	0.7	2

Finally it can be seen in Figure 63 the response of the best candidate (fitness 2299.77) compared to the response of the first fuzzy controller. It is clear that the response of the controller is better, however a small error can be found once the controller has taken the machine to the desired reference. In Figure 63.a were the reference is relatively small this error is more than the 10% -5.14- of the reference.

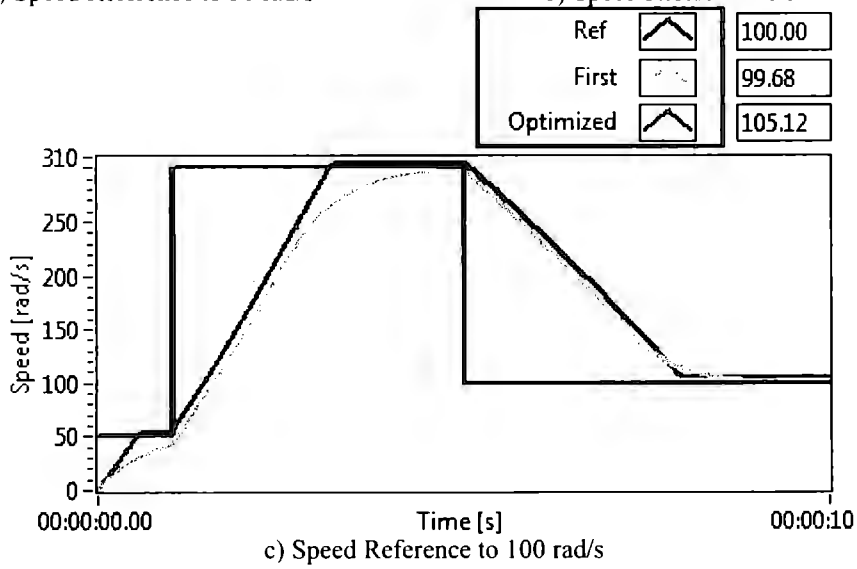
As the reference is changed and the value is increased it can be seen in Figure 62.b and Figure 62.c that approximately the same amount is about 5 more units of the desired speed reference where the machine will settle in steady state response. The importance of this steady-state error will depend on the final application.



a) Speed Reference to 50 rad/s



b) Speed Reference to 300 rad/s



c) Speed Reference to 100 rad/s

Figure 63. Results for Fuzzy and Enhanced MFs Fuzzy PID Controllers.

The steady-state error found on the response of the optimized forms of the membership functions shown in Figure 62; is given to the fact that the forms of these optimized membership functions is not symmetric. This symmetry was not considered on the design of the genetic algorithm, but it can be seen that the response is maintained in the best possible form as is desired, with a decrease in the fitness value.

5.2.3. Simulations of Fuzzy PID enhanced by Genetic Programing

Optimizations made using Genetic Programming are also two, the first one for the normalizing constants and the second for the consequences of the fuzzy rules, the detailed implementation process is explained in section 4.3.2.2.

The fitness function for these tests had a period of time of 10 seconds, with a step time of 0.0002 second and was solved with the Euler method. Results of the GP are shown in Table 31. Several individuals were found with different combinations of bit size lengths and other parameters. The best individual found had a fitness value of 17119.8; Figure 64 shows the results of this individual for the test presented in section 5.2.1 and Table 27.

Table 31. Genetic Programming results for input and output scaling factors optimization

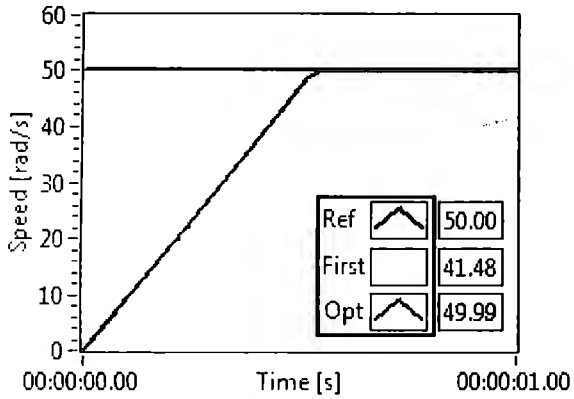
# Bits	Pop Size	I in T	C Pr	M Pr	Epochs	Fitness	ScF Err	ScF dErr	ScF d2Err	ScF Out
10	12	3	0.9	0.02	147	21399.71	266	0.0022	424	988
10	12	3	0.9	0.035	352	20655.92	585	0.0020	0.0020	988
10	16	4	0.9	0.025	60	21399.7	195	0.0025	214	834
13	24	6	0.9	0.01	35	19261.2	7182	0.0004	0.0003	5168
13	***14***	***4***	***0.9***	***0.015***	***82***	***17119.8***	***512***	***84***	***0.0002***	***8144***
13	14	4	0.9	0.015	194	19247.7	6114	0.0003	0.0005	1822
16	24	6	0.9	0.01	206	19250.93	55378	3.3E-5	5.7E-4	54277

Where:

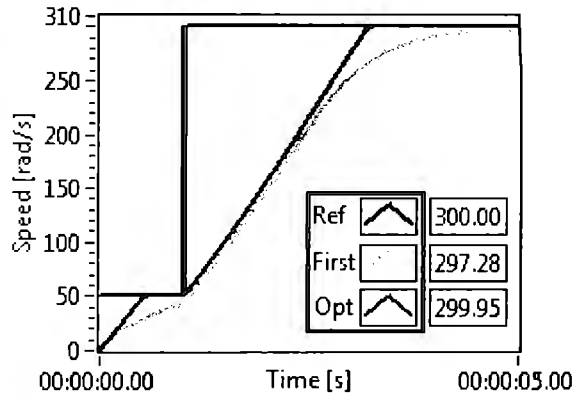
# Bits	Number of bits used to represent numbers in the search space
Pop Size	Size of the population during the execution of the algorithm.
I in T	Number of individuals in the tournament selection method
C Pr	Crossover probability of crossing two branches of the same kind in an individual
M Pr	Mutation probability of mutating a bit in a branch of the individual
Epochs	Number of generations executed
Fitness	Fitness of the best individual found during that execution of the algorithm
ScF Err	Scaling Factor for the input error in the PID controller
ScF dErr	Scaling Factor for the input of the first derivate of the error in the PID controller
ScF d2Err	Scaling Factor for the input of the second derivate of the error in the PID controller
ScF Out	Scaling Factor for the output in the PID controller

Figure 64 shows that the controller takes the machine to the desired reference much faster and accurately that the originally proposed controller. On the subfigure Figure 64.a the controller is able to take the reference in a small period of time, something the original controller is not; these response is consistent in all the optimized controllers.

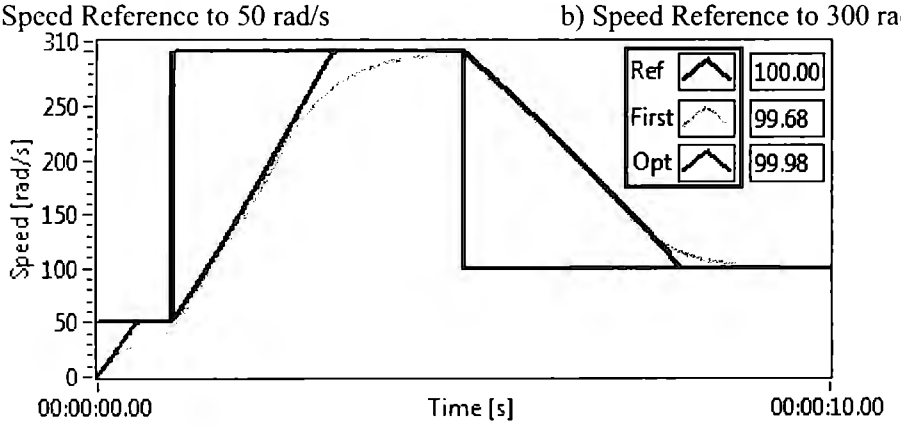
Figure 64.b again shows that the response of the optimized controller is much better. Finally Figure 64.c shows that both controllers are able to take the motor to the desired speed reference without any bigger difference. This results show that GP is able to successfully optimize the overall response of the controller and that it can deal with more complex problems than original GA's.



a) Speed Reference to 50 rad/s



b) Speed Reference to 300 rad/s



c) Speed Reference to 100 rad/s

Figure 64. Results for enhanced Scaling Factors on Fuzzy PID Controllers.

The optimization of the consequences of fuzzy rules was executed after the optimization of scaling factors. A different fitness function was designed: two separated simulations of 10 seconds each with reference of 377 and -377 rad/s are used, the result fitness is summed. Why not use a single simulation of 20 seconds? Because the error is minimized but a desired response is not obtained.

Table 32. Genetic Programming results fuzzy rule optimization

Pop Size	T in T	C Pr	M Pr	Epochs	Fitness	Rules Changed
20	6	0.9	0.1	19	17119.8	None
10	5	0.9	0.1	25	17119.8	*Derivative Gain:* Positive, Positive, Zero
20	8	0.9	0.05	10	17119.8	*Positive Gain:* Positive, Zero Positive *Derivative Gain:* Positive, Zero, Positive.
20	8	0.9	0.05	11	17119.8	*Positive Gain:* Negative, Negative, Negative *Derivative Gain:* Positive, Positive, Zero.
10	5	0.9	0.01	85	17119.8	*Positive Gain:* Zero, Zero, Positive

The results of this optimization are shown in Table 32, in the *Rule Changed* column are the changed consequences from the originally proposed in Table 21. As it can be seen from the results, the rule consequences from the Positive and Derivative Gain are changed, without a difference on the fitness.

Positive and Derivative gains are related to the first error and second error derivative. This is happening because these gains have almost no contribution to the action of the controller; thus allowing the genetic programming algorithm to change the consequences of the rules without altering the response of the controller. These rules can also be eliminated without losing great performance of the controller.

5.2.3.1. Simulations on Enhanced Controllers with Load

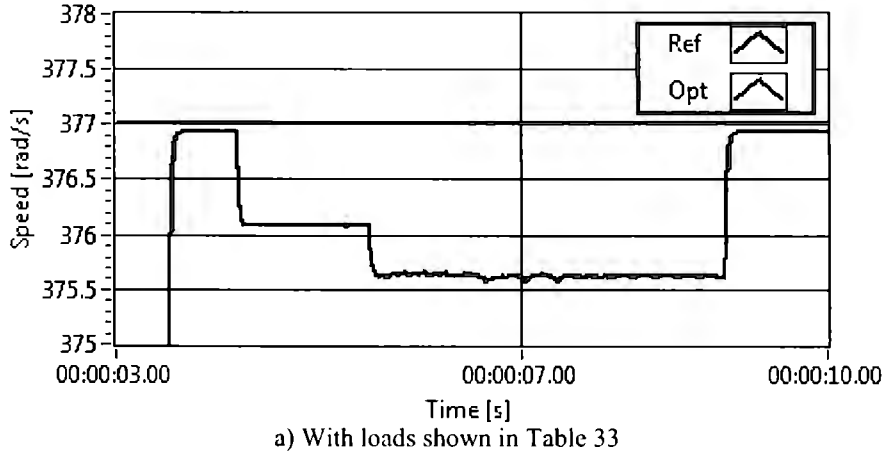
Although the response of the controller has been optimized it is interesting to analyze what would happen if load is introduced to the controller. For that reason a simple test is shown in Table 33; after the controller has reached a desired speed, different loads are introduced to the machine.

Table 33. Load Test.

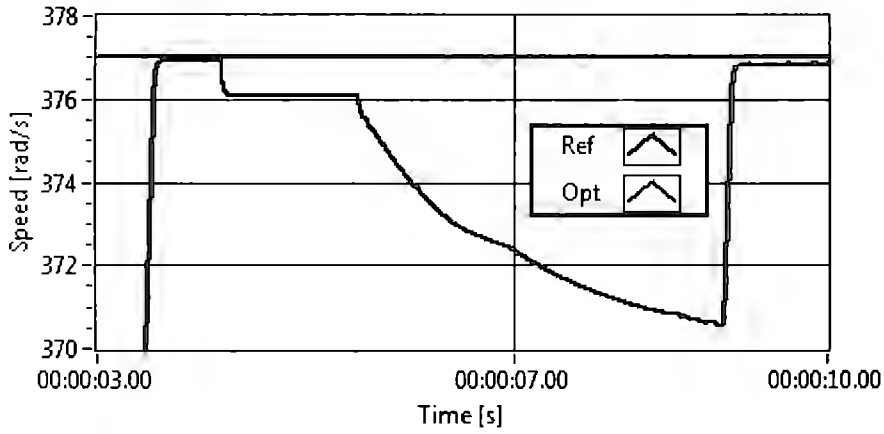
Time Range [s]	Torque Load [Nm]
[4.2, 5.5]	50
[5.5, 9]	75

From the result shown in Figure 65.a it can be seen that once the first amount of load is introduced to the machine the speed reference drops by almost 1 rad/s in difference, later once the load is increased the speed reference decreases even more. The decrease in the speed reference is not that big however this could be a factor depending on the application.

If the second load is increased to 80 Nm, it can be seen in Figure 65.b that the controller fails to maintain the desired reference speed and speed reference beings to decrease, until the load is retired from the motor and the speed reference can be controlled again.



a) With loads shown in Table 33



b) With second load increased to 80 Nm

Figure 65. Result from enhanced fuzzy controller with load.

Results shown in Figure 65 show that although optimization has been made to the controllers, there are certain things out of the possibilities to be optimized by genetic algorithms. This kind of optimization needs to be done during the execution of the control system, so the controller can adapt to the current conditions.

Because it was found that the shapes of fuzzy clusters can be optimized online, genetic programming was not carried out for their optimization.

5.2.4. Simulations of Neuro-Fuzzy Controller

Implementation of this controller is shown in section 4.4.1.1. The first test for the neuro-fuzzy controller was to create fuzzy clusters with the FCM algorithm by generating a set of points in the range of [-2,2]. Then this data set was feed to the FCM program and the fuzzy sets were generated. This information was used as the input and output membership functions in conjunction with the scaling factors found in the previous execution of the genetic programming algorithm.

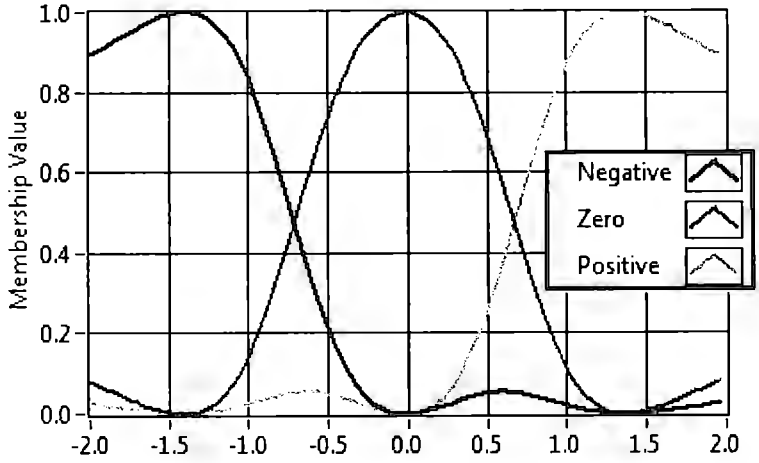


Figure 66. Membership functions after the execution of FCM algorithm

After the FCM algorithm is executed the form of the membership functions is shown in Figure 66. The centers of the fuzzy clusters found by the algorithm are in [-1.4, 0, 1.4]; these clusters are approximated by T-ANN's with 15 neurons each. After the execution of the FCM algorithm the membership functions are used in the controller, using the speed references of Table 27.

After the membership functions are tuned the speed test was executed, showing very bad results -Figure 67. It can be seen that the speed reference is totally out of reference. This response is attributed to the rounded form of the membership functions and that the negative and positive membership functions have very small membership value around zero.

However, irregular boundaries -not commonly proposed by humans- can be used to control the machine as shown in Figure 62, were the GA found unusual limits for the membership functions which would effectively control the speed of the machine with a small steady state error. But in the case of the FCM algorithm, its execution rules lead to symmetric fuzzy clusters.

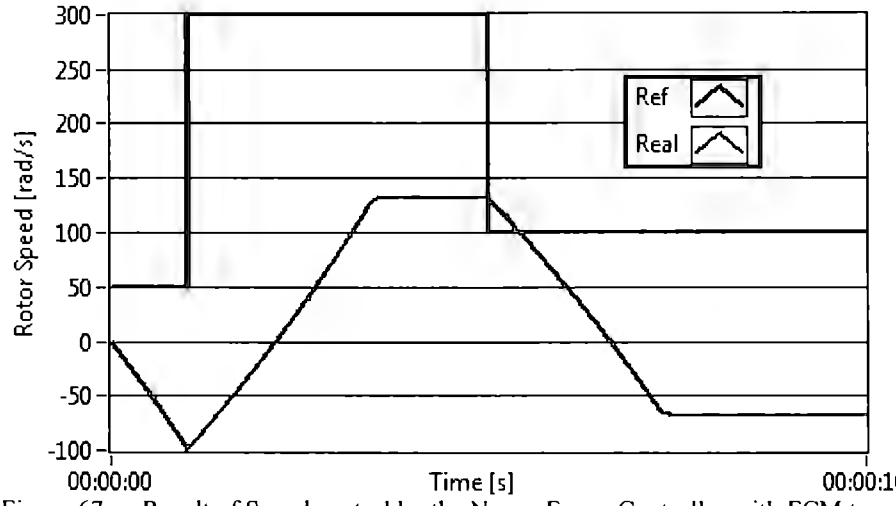


Figure 67. Result of Speed control by the Neuro-Fuzzy Controller with FCM tuned MFs.

To probe that the response seen in Figure 67 is wrong because of the form of the membership functions tuned with FCM, the limits of the fuzzy clusters obtained by the FCM algorithm were detected. With those limits triangular membership functions were generated and trained to the Neuro-Fuzzy controller membership functions, as seen in Figure 68; for this operation 20 neurons were used. This new membership functions are used in the test.

Once the triangular membership functions are adjusted to the limits of the clusters found by the FCM algorithm (Figure 68) the response of the controller is optimum and there is no malfunction. This is shown in Figure 69, this response is similar to the previous fuzzy controllers.

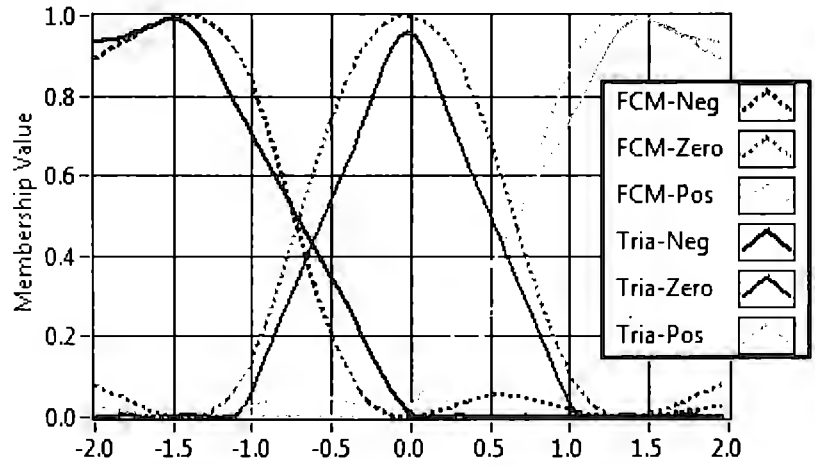


Figure 68. Triangular shaped membership functions extracted from FCM clusters.

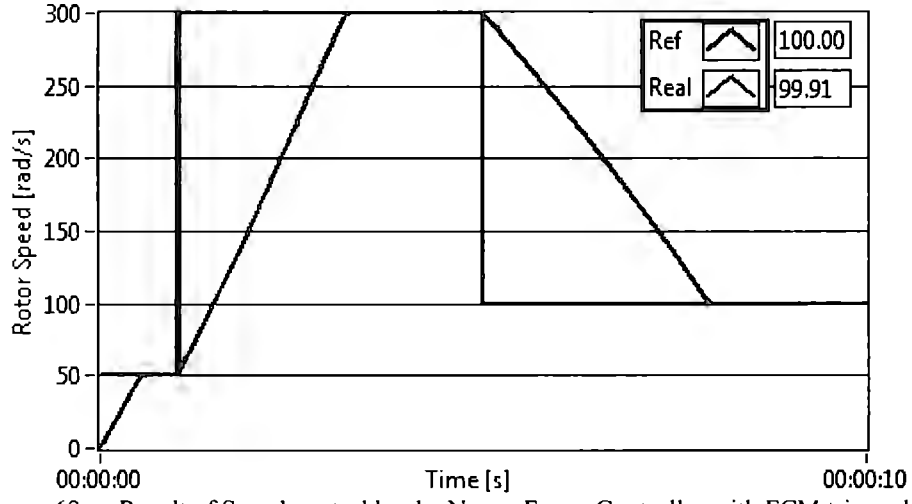


Figure 69. Result of Speed control by the Neuro-Fuzzy Controller with FCM triangular MFs.

The next step is to prove the response of the controller by modifying the form of the membership functions as explained in section 4.4.1. As it is previously explained, information around a certain point is eliminated from the data set feed to the FCM algorithm so the shapes of the resulting fuzzy clusters are altered.

The negated error was the element chosen to eliminate the data around it and create the new clusters. If the response of the controller is analyzed, we will realize that depending on the size of the error the response will be directly proportional. If the error is positive, the overall response of the controller needs to be positive to compensate and vice versa.

Thus, by eliminating elements around the negated value of the error the size of the membership functions in which the controller is acting will be bigger, increasing the response of the controller. Once the controller takes the motor near the speed reference the size of the zero membership function will decrease; if perturbations such as loads are introduced to the machine will be compensated better by the control system.

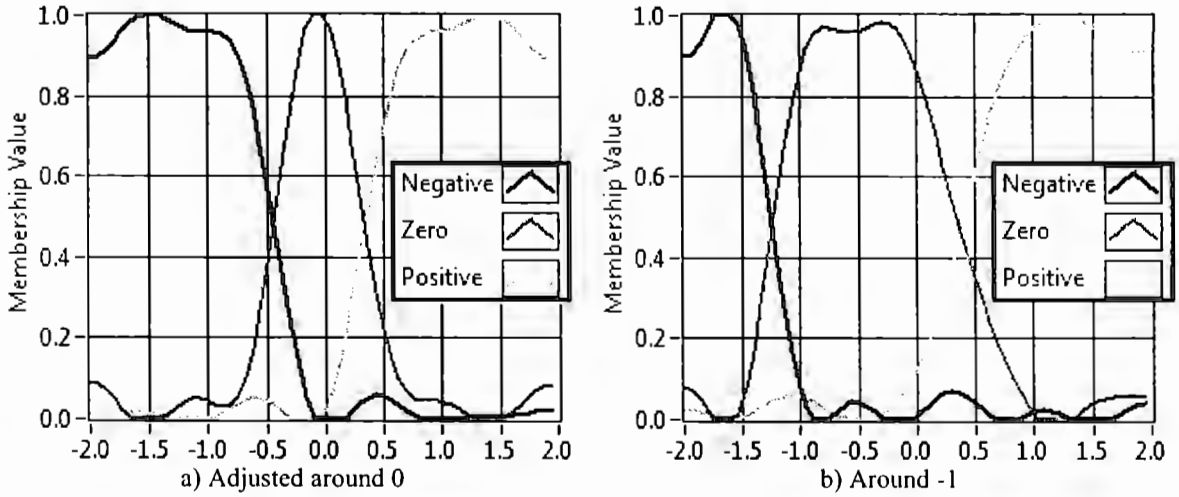


Figure 70. Membership functions adjusted with FCM to optimize response of the controller using 10 neurons for the T-ANN's.

As shown in Figure 70 membership functions are adjusted on the fly by the FCM algorithm; on the images 10 neurons were used to adjust the shapes of the MF's but in the speed control test 15 were used, to increase accuracy.

The result of the test is shown in Figure 71 again it shows that the controller fails to regulate the speed of the motor, showing an enormous difference between the reference and the real value of the speed. The result shown in Figure 71 is again attributed to the irregular shapes of the fuzzy clusters obtained from the FCM algorithm, not being suitable for this application.

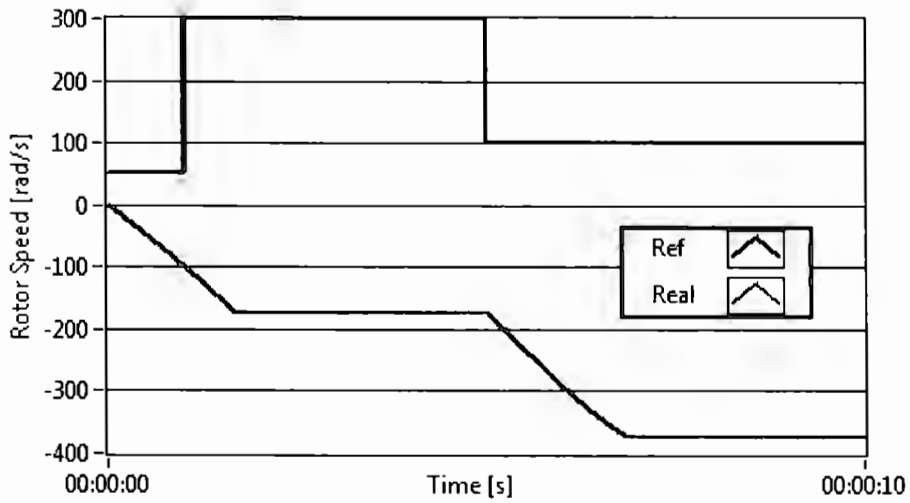


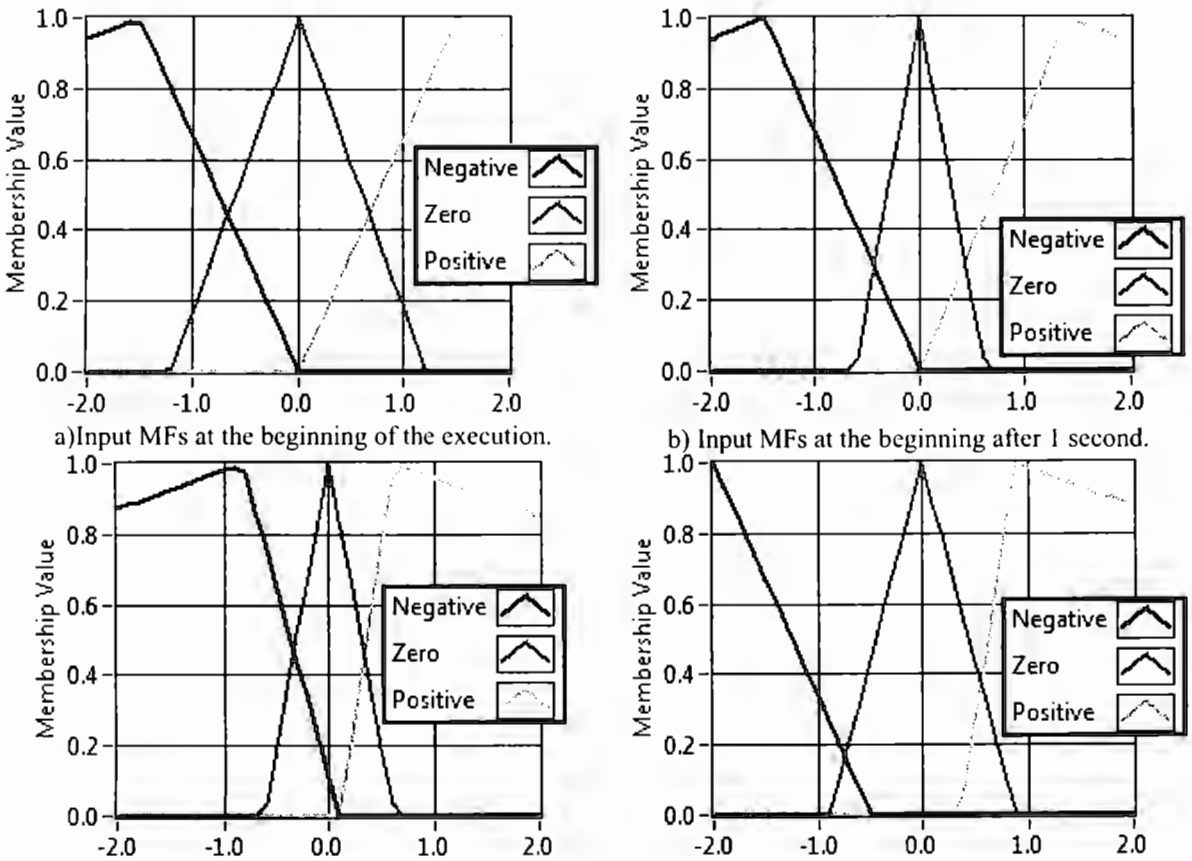
Figure 71. Result of Speed control by the Neuro-Fuzzy Controller with FCM tuned MFs.

5.2.5. Simulation of Adaptable Fuzzy Controller

The implementation process of this controller is shown in section 4.5.1.1 being the result of an adaption of the neuro-fuzzy controller not able to correctly regulate the speed on the IM drive.

The adjusted membership functions are shown in Figure 72. The first two subfigures Figure 72.a and Figure 72.b are the shapes of the triangular membership functions for the inputs. The shapes of the input MF's are optimized with the information of the fuzzy centers coming from FCM, the beginning and ending limits of the zero cluster.

The limits of the output MF's are also optimized with the centers coming from FCM, the ending limit of negative cluster, beginning and ending of zero cluster, and beginning of the positive cluster as shown in subfigures Figure 72.c and Figure 72.d.



a)Input MFs at the beginning of the execution.

b) Input MFs at the beginning after 1 second.

c) Output MFs at the beginning of the execution.

d) Output MFs after 1 second.

Figure 72. Triangular membership functions adjusted with FCM algorithm.

The results for the test presented in Table 27 are shown in Figure 73; as it can be seen the response of the controller is optimal being able to set the speed of the motor to the desired reference and update the limits of its membership functions on the fly.

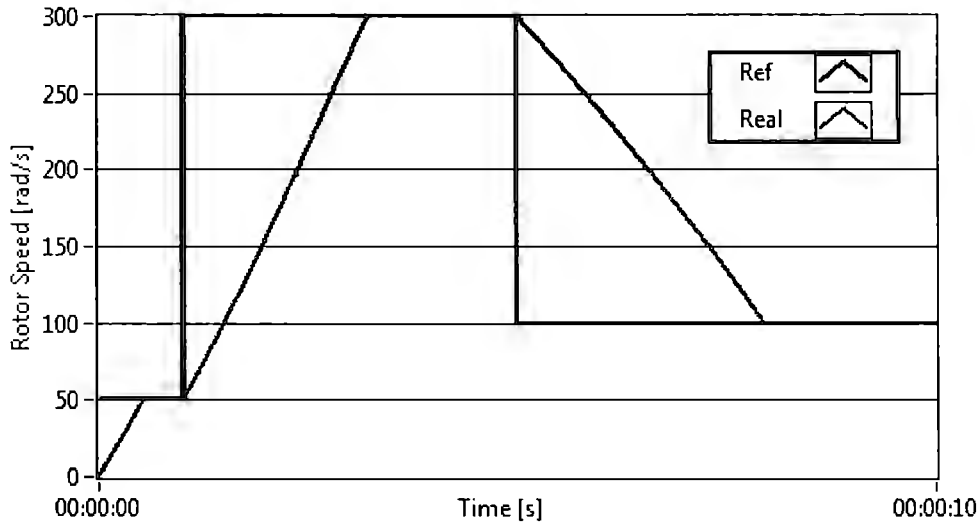


Figure 73. Speed test for Adjustable Fuzzy Controller.

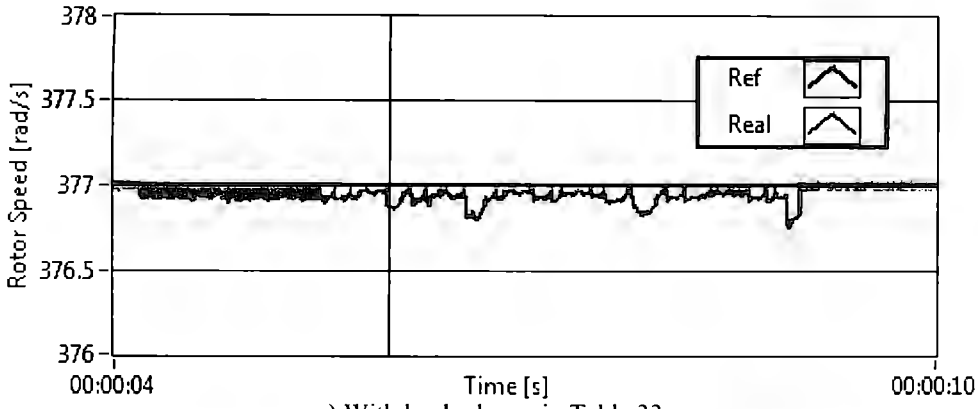
5.2.5.1. Simulations of Adaptable Fuzzy Controller with Load

It has been proven that the response of the controller is optimal without loading the motor; however it is necessary to validate that the controller is also able to control the speed when the motor is loaded. The test shown in Table 33 was applied to the adaptable fuzzy controller.

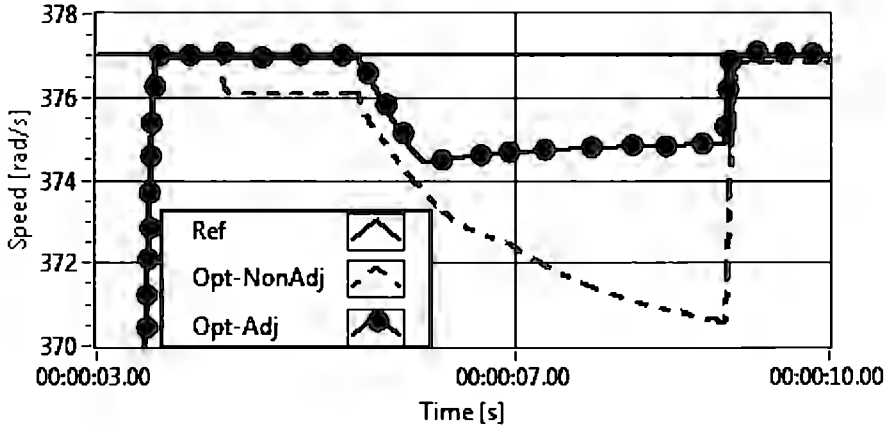
From the results shown in Figure 74.a it can be seen that once the load perturbation is introduced the controller is able to regulate and maintain the speed, some ripple and noise is introduced but the error is minimum. Later when the load is increased to 80Nm, Figure 74.b shows that the response of the controller is much better than the non-adaptable version of the fuzzy controller.

As can be appreciated in Figure 74.b, when the first load of 50Nm is introduced, the online optimized controller response does not changes; while the response of the normal fuzzy controller drops a bit but remains constants. Later when the 80Nm load is introduced the response of both controllers drops; however after some moments the online optimized controller is able to compensate and the response begins to be corrected.

The normal controller is not able to compensate and the response drops until the load is withdrawn. After the load is retired both controllers are able to compensate and correct the response. A small steady state error can be seen in the normal fuzzy controller, while the optimized controller does not show a steady state error.



a) With loads shown in Table 33.



b) With second load increased to 80 Nm

Figure 74. Result from adjustable fuzzy controller with load.

A second load test was designed, the speed reference is set to 100 rad/s after 2.2 seconds a static load of 50Nm is introduced together with a dynamic sinusoidal load of 30 Nm peak to peak summed to the 50Nm static load. The simulation is then run for 10 seconds; results are shown in Figure 75.

The response of the fuzzy controller with fixed MFs *-normal-* again is compared with the response of the controller with adjustable MFs *-optimized.* As it can be seen from Figure 75 both responses are identical until the load is introduced. The response of the *normal* controller drops a little and shows some ripple while the response of the *optimized* controller remains without any visual perturbations.

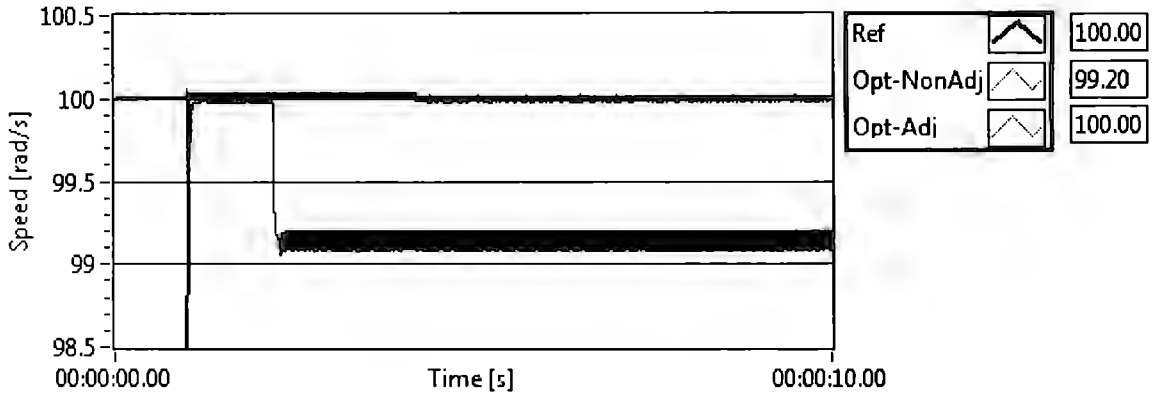


Figure 75. Result from adjustable fuzzy controller with dynamic load.

6. Conclusions and Future Work

This work presents a novel approach on intelligent control for the speed control on induction machines. In this chapter conclusions are presented and discussed, finally future work for a possible direction of the thesis is presented.

As it was presented in the state-of-the-art research presented in chapters 1 and 2, there is a vast area were research can be done in the area of intelligent control applied to electric drives. Most of these applications require of great computational capabilities, which is getting easier to get at lower costs.

6.1. Conclusions

As the objectives of this thesis were proposed in section 1.4, the simulated implementation of direct torque control was carried out successfully. The successful implementation of the programs in Simulink validated the simulation, given that Simulink is a proven environment for simulations.

Once the basic program was proven to work in Simulink it was transferred to LabVIEW to compare the results and validate that the simulation worked correctly in this environment. LabVIEW is just starting to provide tools for control and complex differential equation simulations, but provides a faster development environment and prototype design.

Several intelligent controllers were proposed, validated and analyzed. Optimization techniques were applied to these controllers; also modifications for online execution were proposed and included to further improve their performance.

The design of all the controllers was easy, given that fuzzy systems are easily understood by humans. The proposed PID decoupled topology was validated, and can be easily transferable to an embedded system. Of four controller proposed in chapter 4, three of them were able to control the speed of the machine effectively.

The Neuro-Fuzzy controller was unable to correctly control the speed of the machine. This is attributed to the arbitrary shapes of the membership functions; and to the fact that speed is not directly controlled on the machine but through the control of torque. Yet the use of this controller gave us the chance to propose a similar topology with a better response and simpler design.

The optimizations based on genetic algorithms and genetic programming proved to effectively optimize the fuzzy controllers. Genetic programing can deal with more complex forms of optimizations and yet find results faster than a similar but simple approach using genetic algorithms. These optimizations have to be carried offline due to the nature of evolutionary algorithms and their delayed response due to large search spaces.

However, thanks to the advancements in processing power and multiple cores computing this is something that was not possible to execute few years ago in reasonable time lapses. The simulation of a differential equation model is computationally expensive; nonetheless it is now possible to execute small time lapses executions in short periods of time.

The controllers were systematically optimized with the aid of these algorithms, and some information was found to be irrelevant being discarded on the different stages of the optimization. Such is the case of the positive and derivative gains being eliminated by reducing its interaction, first during the Scaling factors optimization and next in the Rule consequences optimization. After applying these optimization techniques the contribution of these gains can be effectively eliminated.

The online optimization proposed proved to correctly compensate unknown uncertainties introduced to the system and further improve the response of the controllers. Furthermore, this optimization can be executed online without introducing to many computational load, as it is executed with a 1 millisecond period.

6.2. Future Work

This work is intended to be published in one or two congresses, a journal or transaction magazine and by a specialized editorial. It is necessary to validate these algorithms in a prototype. During the development of this work a version of the DTC program that can be executed on the Real Time Machine of LabVIEW was developed and validated. This program can be executed on hardware with a real time operating system like the Compact RIO [11] from National Instruments.

Given that the speed controllers were developed in LabVIEW, they can also be executed in this system. So the complete control system proposed here can be executed on National Instruments hardware and software. Nevertheless this work is not limited to that brand and can also be programed on a small embedded system like a DSP or a FPGA card.

As it was analyzed and proved in this thesis, the use of genetic programing to optimize fuzzy systems can provide very good results. A more complex task like the development and evolution of a complete fuzzy controller using could be an interesting next step.

Eventually computational power will allow us to execute these optimization systems on small embedded systems that will be able to automatically adjust. It would also be interesting to program the genetic optimization system on an FPGA card, executing the differential equation model of the machine on a DSP card or something similar. It could eventually lead to the development of robust intelligent sensorless drives.

References

[1] ABB Industry Oy: *Direct Torque Control the World's Most Advanced AC Drive Technology*. Technical Guide No. 1.

[2] Astrom K and Hagglund T. 1995. *PID Controllers: Theory, Design and Tuning*. Instrument Sociery of America, USA.

[3] Beerten J. et al. 2010. Predictive Direct Torque Control for Flux and Torque Ripple Reduction. *IEEE Transactions on Industrial Electronics* Vol. 57, No. 1: 404 – 12.

[4] Blaschke, F. 1972. *The principle of field-orientation as applied to the new Transvektor closed-loop control system for rotating machines*. Siemens Review 34: 271-20.

[5] Bourmistrova, A. & Khantis, S. 2007. *Control system design optimisation via genetic programming*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007. Singapore: 1993 - 2000.

[6] Buja G. S. and Kazmierkowski M. P. 2004. Direct torque control of PWM inverter-fed AC motors—A survey. *IEEE Transactions on Industrial Electronics*, Aug. 2004, Vol. 51, No. 4: 744–57.

[7] Casillas Jorge, Cordon O., and Herrera F. 2000. *Learning Fuzzy Rules Using Ant Colony Optimization Algorithms*. Proceedings of the 2nd International Workshop on Ant Algorithms, Brussels, Belgium: 13-21.

[8] Capolino Gerard-Andre. *Recent Advances and Applications of Power Electronics and Motor Drives - Advanced and Intelligent Control Techniques*. Paper presented at 34th Annual Conference of IEEE Industrial Electronics 2008, Florida, USA, November 10 – 13, 2008.

References

[9] Cheng-Zhi Cao, et al. 2004. *Optimization Design Of Fuzzy Neural Network Controller In Direct Torque Control System.* Proceedings of the Third International Conference on Machine Learning and Cybemetics, Shanghai, 26-29 August 2004, Vol. 1: 378 – 82.

[10] Chiu Stephen. 1997. Developing commercial applications of intelligent control. *IEEE Control Systems Magazine*, Vol. 17 Issue 2: 94 – 100.

[11] Compact RIO, Rugged and Reconfigurable Control and Monitoring System by National Instrumens. Accesed on March 24 of 2010. http://www.ni.com/compactrio

[12] Cordón O. and Herrera F. 2000. A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems*, 8(3).

[13] Depenbrock, M. 1985. *Direkte Selbtregelung (DSR) für hochdynamische Drehfeldantribe mit Stromrichterschaltung.* ETZ A 7: 211-18.

[14] Desicions Etudes Council. Last modified on April, 2010. World Electronic Industries 2008 – 2013 Executive summary. Accessed on August 19 of 2010 http://www.decision.eu/doc/brochures/exec_wei_current.pdf

[15] Dorigo M, and Gambardella. L M. April 1997. Ant Colony System: A Cooperative Learning Approach To The Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* Vol. 1 No. 1: 53-66.

[16] Dufoo S., Pacas M. 2010. *Predictive Direct Torque Control of an Induction Machine with Unsymmetrical Rotor.* Paper presented at the IEEE International Conference on Industrial Technology (ICIT): 1851 – 6.

[17] Flach E, Hoffmann, Mutschler P. 1997. *Direct mean torque control of an induction motor.* Paper presented at the European Conference on Power Electronics and applications, (EPE' 97) Trondheim, Norway, vol.3: 672 - 677.

[18] Fourier Joseph. 2003. *The Analytical Theory of Heat.* USA: Dover Publications.

References

[19] Jang, J-S. R. 1993. ANFIS: Adaptive Network Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics* Vol. 23 No. 3: 665 – 85.

[20] Ghanea-Hercock, Robert. 2003. *Applied evolutionary algorithms in Java.* Springer, New York, USA.

[21] Gen Mitsou & Runwei Cheng. 2000. *Genetic Algorithms & Engineering Optimization.* John Wiley & Sons, Inc. USA

[22] Geyer T. et al. 2009. Model Predictive Direct Torque Control—Part I: Concept, Algorithm, and Analysis. *IEEE Transactions on Industrial Electronics* Vol. 56 No. 6: 1894 – 1905.

[23] Google. Google's mission is to organize the world's information and make it universally accessible and useful. http://www.google.com Accessed on Mayo 30 of 2011.

[24] Goldbergh, D. E. 1989. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, USA.

[25] Grabowski, P.Z.; Blaabjerg, F. 1998. *Direct Torque Neuro-Fuzzy Control Of Induction Motor Drive.* Paper presented at the 23rd International Conference on Industrial Electronics, Control and Instrumentation, 1997. IECON 97 Vol. 2: 557 – 62.

[26] Guizzo Erico. *Robots Enter Fukushima Reactors, Detect High Radiation.* Automaton IEEE Spectrum magazine blog. http://spectrum.ieee.org/automaton/robotics/industrial-robots/robots-enter-fukushima-reactors-detect-high-radiation Accesed on May 30 of 2011.

[27] Guizzo Erico. *Can Japan Send In Robots To Fix Troubled Nuclear?* Automaton IEEE Spectrum magazine blog. http://spectrum.ieee.org/automaton/robotics/industrial-robots/japan-robots-to-fix-troubled-nuclear-reactors Accesed on May 30 of 2011.

References

[28] Hasse, K. 1972. *Drehzahkregelverfahren für schnelle Umkehrantriebe mit stromrichtergespeisten Asynchron-Kurzschlisslaufermotoren.* Regelunstechnik 20: 60-6.

[29] Honkui Li. 2010. *Fuzzy DTC for Induction Motor with Optimized Command Stator Flux.* Paper presented at the 8th World Congress on Intelligent Control and Automation July 6-9 2010, Jinan, China: 4958 – 61.

[30] Hopgood A. A. 2005. The state of artificial intelligence. *Advances in Computers* Elsevier ,Vol. 65: 1-75.

[31] Hopgood A. A. 2003. Artificial intelligence: hype or reality? *Computer, IEEE Computer Society Magazine.* Vol. 36, Issue: 5: 24.

[32] iRobot, designs and builds robots that make a difference. http://www.irobot.com Accessed on May 30 of 2011.

[33] Jinhua, Liu; Xuezhi, Hu. 2010. *The Research of Induction Motor DTC Systems Based on Fuzzy Observer.* Paper presented at the 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA), Vol. 3: 915 – 8.

[34] Kamepalli, H.B. 2001. The optimal basics for GAs. *IEEE Potentials Magazine*, Vol. 20, No. 2: 25

[35] Koza John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, USA, 1992.

[36] Kubrick Stanley. 1968. *2001: A Space Odyssey.* MGM studios, USA.

[37] LabVIEW, graphical programming language by National Instruments. Accessed on November 26 of 2010. http://www.ni.com/labview

[38] Leonhard, W. 1985, 1996. *Control of electrical drives.* Springer-Verlag, Berlin.

References

[39] Levy Steven. The AI Revolution Is On. 2010. *Wired Magazine.* January 2011 edition. http://www.wired.com/magazine/2010/12/ff_ai_essay_airevolution/ Accessed on May 30 of 2011.

[40] Liceaga-Castro, E.; Ugalde-Loo, C.E.; Liceaga-Castro, J.; Ponce, P. 2005. *An Efficient Controller for SV-PWM VSI Based on the Multivariable Structure Function.* Paper presented at the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05: : 4754 – 9.

[41] Mamdani E. H. 1974. *Application of Fuzzy Algorithms for Control of Simple Dynamic Plant.* Proceedings of fhe Institution of Electrical Engineers, Control & Sciences. Vol. 121, No. 12: 1585 – 8.

[42] Masood Hajian. 2010. Adaptive Nonlinear Direct Torque Control of Sensorless IM Drives With Efficiency Optimization. *IEEE Transactions on Industrial Electronics* Vol. 57 No. 3: 975 – 985.

[43] MATLAB, The language of technical computing, by MathWorks. Accessed on November 26 of 2010. http://www.mathworks.com/products/matlab

[44] Meie Sui; Kairu Zhang; Junqing Yang. 2008. *An Improved Sensorless DSVM-DTC of Induction Motor Based MRAFC.* Paper presented at the 7th World Congress on Intelligent Control and Automation, 2008. WCICA 2008: 775 – 80.

[45] Morales-Caporal R. and Pacas M. 2008. Encoderless Predictive Direct Torque Control for Synchronous Reluctance Machines at Very Low and Zero Speed. *IEEE Transactions on Industrial Electronics* Vol. 55 No. 12: 4408—4416.

[46] Ney E. T. Merheb, General Electric Last modified on June of 1999. Application of Induction Motors with Variable Frequency Drives. Accessed on August 29, 2010. http://www.geindustrial.com.br/download/artigos/asdi.pdf

[47] Ogata Katsuhiko. 2002. *Modern Control Engineering.* Prentice-Hall Inc., USA.

References

[48] Pacas M. and Weber J. 2005. Predictive direct torque control for the PM synchronous machine. *IEEE Transactions on Industrial Electronics* Vol. 52 No. 5: 1350 – 56.

[49] Park R. H. July 1929. Two-Reaction Theory of Synchronous Machines – Generalized Method of Analysys. Part I. *AIEE Transactions*, Vol.48: 716 – 27.

[50] Ponce Pedro et al. 2007. *Neuro-Fuzzy Controller Using LabVIEW.* Paper presented at the 10th ISC conference IASTED, Cambridge, MA, USA, November 19 – 21, 2007.

[51] Ponce Pedro et al. 2008. *Intelligent Control Toolkit for LabVIEW.* ITESM Research Project, Mexico.

[52] Ponce C. Pedro; Rivas, J.J.R. 2001. *A Small Neural Network Structure Application In Speed Estimation Of An Induction Motor Using Direct Torque Control.* Paper presented at the IEEE 32nd Annual Specialists Conference on Power Electronic PESC. 2001 Vol. 2: 823 – 7.

[53] Ponce C. Pedro & Sampe Javier. 2008. *Maquinas Electricas y Tecnicas Modernas de Control.* Alfaomega Grupo Editor.

[54] Ponce-Cruz Pedro & Fernando D: Ramirez-Figueroa. 2009. *Intelligent Control Systems with LabVIEW.* Springer-Verlag, London, UK.

[55] Ponce C. Pedro. 2010. *Inteligencia Artificial con Aplicaciones a la Ingenieria.* Alfaomega Grupo Editor.

[56] Rao S. Singiresu. 1996. *Engineering Optimization.* John Wiley & Sons, Inc. USA.

[57] Rajaji, L.; Kumar, C. 2008. *Adaptive Neuro-Fuzzy Inference Systems into Squirrel Cage Induction Motor Drive: Modeling, Control And Estimation.* Paper presented at International Conference on Electrical and Computer Engineering, 2008, ICECE 2008: 162 – 9.

[58] Rodic, M.; Jezernik, K. 2002. Speed-Sensorless Sliding-Mode Torque Control Of An Induction Motor. *IEEE Transactions on Industrial Electronics*, Vol. 49: 87 – 95.

References

[59] Sabanovic, Asif ; Izosimov, Dmitrij B. 1981. Application of Sliding Modes to Induction Motor Control. *IEEE Transactions on Industry Applications*, Vol. IA-17 , Issue 1: 41 – 9.

[60] Simon Dan. 2006. *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches.* John Wiley & Sons Inc., USA.

[61] Simulink, Simulation and model based design, by MathWorks. Accesed on November 26 of 2010. http://www.mathworks.com/products/simulink

[62] Siskind, Chales S. 1963. *Electrical Control Systems in Industry.* McGraw-Hill. New York, USA.

[63] Takagi T. and Sugeno M. 1985. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics* Vol. 15: 116-132.

[64] Takahashi, I. and Noguchi, T. 1984. *Quick torque response control of an induction motor using a new concept.* IEEE J. Tech. Meeting on Rotating Machines, paper RM84-76: 61-70.

[65] Takahashi, I. and Noguchi, T. 1985. *A new quick response and high efficiency control strategy of an induction motor.* IEEE IAS Annual Meeting: 496-502.

[66] Serna E., Pacas M. 2006. *Detection of Rotor Faults in Field Oriented Controlled Induction Machines.* Conference Record of the 2006 IEEE on Industry Applications Conference 41st IAS Annual Meeting Vol. 5: 2326 – 32.

[67] Vas Peter. 2003. *Sensorless Vector and Direct Torque Control.* Oxford University Press.

[68] Vas Peter. 1999. *Artificial-Intelligence-Based Electrical Machines and Drives.* Oxford University Press.

[69] Venkataraman P. 2002. *Applied Optimization with MATLAB ® Programming.* John Wiley & Sons, Inc., USA

[70] Xiaoguang Qu, et al. 2010. *DTC with Adaptive Stator Flux Observer and Stator Resistance Estimator for Induction Motors.* Paper presented at the 8th World Congress on Intelligent Control and Automation July 6-9 2010, Jinan, China: 2460 – 3.

[71] Zaid S. A. 2009. Implementation of a new fast direct torque control algorithm for induction motor drives. *IET Electric Power Applications* Vol. 4 Issue 5: 305–313.

[72] Zadeh Lotfi A. 1965. *Fuzzy Sets.* Information and Control. Vol. 8: 338 – 353.

[73] Zadeh Lotfi A. 2010. *A Summary and Update of "Fuzzy Logic."* Paper presented at the IEEE International Conference on Granular Computing (GrC). August 14-16 2010, San Jose, CA, USA: 42 – 4.

[74] Zhifeng Zhang. 2010. Novel Direct Torque Control Based on Space Vector Modulation With Adaptive Stator Flux Observer for Induction Motors. *IEEE Transactions on Magnetics* Vol. 46 No. 8: 3313 – 36