FUZZY LOGIC CONTROL OF PWM INVERTER-FED SINUSOIDAL PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVES

ZULKIFILIE IBRAHIM

Thesis submitted in partial fulfilment of the requirements of
Liverpool John Moores University for the degree of
Doctor of Philosophy

July 1999

ABSTRACT

A conventional PI speed controller has been used in motion control applications for a long time. Numerous works reported in the recent past have shown that a fuzzy logic controller has the potential to replace the conventional PI controller. Fuzzy logic (FL) control apparently offers a possibility of obtaining an improvement in the quality of the speed response, compared to PI control. The major obstacles to wider fuzzy logic control applications at present are the lack of simple procedures for the design and its relatively high computational requirements.

The research focuses on investigation and evaluation of the performance of a sinusoidal permanent magnet synchronous motor (SPMSM) drive, controlled by PI and constant parameter FL speed controllers. The SPMSM is controlled using the principle of rotor flux orientation. Current control is performed in the stationary reference frame, using hysteresis current controllers. The drive is simulated using Simulink.

A PI and a constant parameter FL speed controller are at first designed. A detailed investigation of the impact of various FL controller parameters (scaling factors, membership functions, rules) on the attainable speed response is performed at this stage. Next, an in-depth comparative analysis of the drive performance, obtainable with PI and FL speed control, is made using the simulation approach. The overshoot / undershoot, settling time and rise time of the speed response, and integral speed error criterion are used to asses the controller performance. The simulation results have proved that the robustness to the inertia variation and load rejection properties are better with FL speed control. Both controllers have demonstrated good performance for speed commands close to the design case, while the response deteriorated for small speed commands.

In order to further improve the operation of the drive with FL speed control, a number of adaptive FL control schemes are developed. The tuning strategy is at first established by investigating the influence of the rule base, membership functions and scaling factors on the drive behaviour. It is concluded that adaptive tuning of the scaling factors offers the easiest and the best way of improving the performance of the FL speed controller. Based on this finding, four types of adaptive FL speed controllers are developed. Two particular adaptive schemes are aimed at on-line adaptation and provide compensation of either variable speed reference, or both the variable speed reference and the variation of the drive inertia.

ACKNOWLEDGEMENT

The author would like to express gratitude to his academic supervisor, **Dr. Emil Levi**, for his help, encouragement, and guidance throughout the period of the work, and in the presentation of this thesis.

Appreciation also goes to **Professor Dave Williams**, the second academic supervisor, for giving invaluable assistance and guidance.

Thanks go to the **members of the Power Research Group** for friendly support on many occasions, in particular **Dr. M.Sokola, Dr. Y.W.Liao** and **Mr. M.Wang.**

To **Lawrence Hughes, Mike Harrington** and **Geoff Newton**, for all their generous help in purchasing components, photocopying, etc. will be remembered forever.

The author is also extremely grateful to the **Standards and Industrial Research Institute of Malaysia** (SIRIM Berhad) for the three year award of a research studentship.

Last but not least, thanks to the parents, **Allahyarham Hj. Ibrahim L. Maidin** and **Aminah Kassim**, the wife **Siti Morni Baron** for their support, patience and sympathy for all the time taken throughout the period of this work, and also to the daughter **Sakinah**, **Asma' Huda** and the son **Muhammad Hanzallah,** who provided the cheerful moments whenever the time was tough during writing of this thesis.

CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

A REVIEW OF LITERATURE ON HIGH PERFORMANCE SPMSM DRIVES

CHAPTER 3

MODELING AND SIMULATION OF SINUSOIDAL PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVES

CHAPTER 4

DESIGN OF FUZZY LOGIC SPEED CONTROLLER

CHAPTER 5

A COMPARATIVE STUDY OF SPMSM DRIVES CONTROLLED BY PI AND FL SPEED CONTROLLERS

CHAPTER 6

INVESTIGATION OF SELF-TUNING FUZZY LOGIC SPEED CONTROLLER

CHAPTER 7

CONCLUSION

CHAPTER 8

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

i) Principal symbols:

ce	change of speed error
e	error signal
e_{ds}	d-axis decoupling voltage
e_{qs}	q-axis decoupling voltage
f	frequency
G_e	standard form of error scaling factor
G_{ce}	standard form of change of error scaling factor
G_u	standard form of output scaling factor
G_e^{UoD}	error scaling factor defined in the universe of discourse
G_{ce}^{UoD}	change of error scaling factor defined in the universe of discourse
G_u^{UoD}	output scaling factor defined in the universe of discourse
i	current
i_{qs}	*q*-axis current
i_{ds}	*d*-axis current
J	total inertia
j	imaginary unit
K_p	proportional gain
K_i	integral gain
K_d	differential gain
K_c	critical gain
L	inductance
M	mutual stator - to - rotor inductance
P	number of pole pairs
θ	instantaneous rotor position angle
R	resistance

S	switching function
s	Laplace operator
T_e	electromagnetic torque
T_L	load torque
T_s	sampling time
T_i	integral time constant
T_d	derivative time constant
u	control signal
v	voltage
ψ	flux lingkage
ω	rotor electrical angular speed
V_{dc}	DC link voltage
Δi	hysteresis band
Δi_{qs}^*	incremental torque current command
β	torque angle
α	spatial position of the stator current space vector

ii) Indices:

a, b, c	stator phase variables and phases of the original three-phase system
ds, qs	stator d-q axis variables
s	stator quantities and variables
r	rotor quantities and variables
m	magnetising (air-gap) quantities
f	excitation winding
n	rated value

iii) Superscripts:

*	commanded value

iv) List of abbreviations

AC	alternative current
AOR	adaptive and optimised current controller
ASIC	application specific integrated circuit
AFOC	air-gap flux oriented control
ANN	artificial neural network
AND	fuzzy connective (linguistic)
BLDC	brushless DC
CPFL	constant parameter fuzzy logic controller
CoG	centre of gravity
CSI	current source inverter
DTC	direct torque control
DC	direct current
DSP	digital signal processor
EMF	electromagnetic force
FL	fuzzy logic
FLC	fuzzy logic controller
FMRLC	fuzzy model reference learning control
IFPIC	improved fuzzy-tuned PI controller
IAE	integral of the absolute of the error
ITAE	integral of time multiplied by absolute error
IM	induction motor
MoM	mean-of-maxima defuzzification
Max-Min	fuzzy inference (conjunction and aggregation operation)
MRAC	model reference adaptive control
MF	membership function
PWM	pulse width modulation
PI	proportional plus integral (controller)
PD	proportional plus derivative (controller)
PID	proportional, integral plus derivative (controller)
UoD	universe of discourse
ODE	ordinary differential equation
OR	fuzzy connective (linguistic)

RFOC	rotor flux oriented control
RDC	resolver-to-digital converter
RLS	recursive least squares
SFOC	stator flux oriented control
ST	self-tuning
SLMC	sliding mode control
SW	ideal switches
SPMSM	sinusoidal permanent magnet synchronous machine
VSI	voltage source inverter
VSC	variable structure controller

CHAPTER 1

INTRODUCTION

1.1 HISTORY AND DEVELOPMENT OF HIGH PERFORMANCE AC DRIVES

Rapid development of industrial automation requires continuing improvement of different types of electrical drives. Manufacturing lines typically involve variable-speed motor drives to power conveyor belts, industrial robots, and other types of processing operations. High reliability, good control characteristics, low maintenance requirements, low investment and low running costs are among the important features that are required from a modern drive [Leonhard, 1986], [Vas, 1999].

For simple drives, industry has always relied mainly on squirrel-cage induction machines. A major disadvantage of this type of machine was in the past its inability to be controlled in an efficient manner. Direct current (DC) machines and wound-rotor induction machines were therefore used in drives where variable-speed operation was essential. The advent of thyristors in the late Fifties enabled efficient and controlled voltage rectification, making variable-speed drives with DC motor simple to control. However, serious limitations of DC motors, such as sparking, extensive maintenance and poor overloading capability, brought attention back to the induction motor [Novotny and Lipo, 1996].

Development of thyristor-based inverters, that could provide variable-frequency alternating current (AC) voltage, took place during the Sixties. As a consequence, so-called `scalar' control techniques emerged, where the frequency and magnitude of supply voltages are controlled by an inverter. In this way an AC motor was able to operate in a controlled variable-speed mode, yielding steady-state characteristics similar to those of a DC motor. Systems with scalar control have been replacing DC drives in numerous applications where accuracy and transient response are not crucial requirements (pumps, fans, compressors, etc.) ever since.

Nevertheless, DC machines were still unbeatable in the area of high-performance (servo) drives, where very fast and accurate torque and/or speed responses are required. Furthermore, position servo control was the application area in which DC machines were the only choice. Superior dynamic control characteristics of DC machines are a consequence of inherently decoupled control of flux and torque, that is realised utilising quite simple control strategy and equipment.

Basic principles of vector control (field orientation), introduced in the early Seventies [Blaschke, 1972], showed that decoupled control of flux and torque was theoretically possible in three-phase AC machines as well. Since there are three flux vectors in an induction machine, three methods of vector control can be distinguished: the stator-flux-oriented control (SFOC), the air-gap-flux-oriented control (AFOC) and the rotor-flux-oriented control (RFOC) [Vas, 1990]. RFOC is the most popular method because of its relatively simple control system structure. All vector controllers, regardless of the orientation system, require accurate information about the instantaneous spatial position of the selected flux vector. This information can be obtained in two ways: directly (by appropriate feedback devices) and indirectly (by feed-forward type of estimation). This leads to additional sub-division of vector control schemes into the feedback and the feed-forward schemes.

Vector control principles are directly applicable to all types of synchronous motors as well [Vas, 1990]. Due to inherent differences that exist between induction and synchronous machines in their operating principles, control system required for decoupled flux and torque control is considerably simpler for a synchronous motor. Synchronous motors, particularly the `brushless' motors with novel high-energy permanent magnets, have therefore found many applications, typically in the areas of machine tools and robotics. Development of vector control theory has even forced manufacturers to build specially designed machines, aimed for vector-controlled drives only. Researchers are still putting a lot of effort in that direction [Slemon, 1994].

Power electronics is an area that has significantly contributed to the evolution of variable-speed drives. Different semiconductor devices such as metal-oxide-semiconductor field-effect transistor (MOSFET), insulated-gate bipolar transistor (IGBT), MOS-controlled thyristor (MCT), emitter switched thyristor (EST), etc., were invented and their characteristics have been constantly improved [Baliga, 1995]. Developments in power electronics have had enormous impact on vector controlled drives. Fast switching devices, coupled with sophisticated pulse-width modulation (PWM) techniques, have enabled excellent current control, so that current feeding became a reality [Holtz, 1992]. This in turn has enabled a wider use of the concept of current-fed machines, with a vector control system that is much simpler than the one utilised for voltage-fed machines.

Developments in VLSI technology have rapidly enhanced the performance of the microprocessors and other hardware, and reduced cost. Advanced digital signal processor (DSPs), application specific integrated circuit (ASICs), and parallel processor (transputers) nowadays provide enough computing power even for the most demanding applications. Advanced control theory techniques, such as model reference adaptive control (MRAC) and Kalman filters, as well as artificial intelligence (AI) techniques, such as fuzzy logic (FL) and neural networks (ANN) can be used in implementation of vector controlled drives [Vas, 1998].

Fuzzy logic (FL) theory has been recently introduced to the area of drive control. FL speed control can provide a very fast response time with good disturbance rejection [Vas and Drury, 1995]. Another possibility is to implement a standard PI speed controller whose gains are varied on-line by

FL [Vas *et al*, 1994b]. Artificial neural networks (ANNs) are very suitable for controlling highly non-linear processes. Fuzzy-neural approach has been successfully implemented in DC and AC drives [Vas and Stronach, 1996a, b], [Stronach *et al*, 1997]. It is expected that application of the artificial intelligence techniques in electric drives will usher a new era in motion control in the coming decades [Bose, 1994], [Vas, 1999].

It should be noted that vector control has had some competitors. Several alternative techniques for control of high-performance electric drives have been proposed. The most popular alternative control method, direct torque control (DTC), was introduced by Takahashi and Naguchi [1986] and Depenbrock [1988]. The main feature of DTC is the absence of co-ordinate transformation and current controllers. DTC has the same problem as vector control in that they both require flux and torque estimates. However, the overall complexity of the control system is substantially reduced, compared with vector controlled drives.

Vector control theory and practice has experienced tremendous development and growth in less than three decades. The outcome is that vector control has grown out from being just an interesting research area, useful for building prototypes only. The applications are numerous and they are found in a number of industrial processes, covering a wide power range. However, there are still numerous problems to be solved and a lot of research effort is still invested in achieving better drive performance.

1.2 AN OVERVIEW OF FUZZY LOGIC CONTROL

In general, fuzzy logic control is a method of control where linguistic rather than numerical rules are used in order to control complex or non-linear industrial process. Fundamentals of fuzzy control systems with engineering applications are given in detail in [Passino and Yurkovich, 1998]. Fuzzy logic is very useful in applications that are difficult to model accurately or are mathematically complex [Thomas *et al*, 1995]. In the field of process control, fuzzy logic is simply a special case of rule based expert systems used by control engineers to provide a useful controller. It is believed that the main merits of fuzzy control are its robustness in the presence of load disturbance, non-linear behaviour and good handling of process characteristic changes (e.g. process gain and delay). A general practical guide to design of fuzzy logic controllers is discussed in detail in many text books [Pedrycz, 1993], [Passino and Yurkovich, 1998]. Fuzzy control provides methods to construct controller algorithms (if-then rules) in a user-friendly way and provides the ability to capture the non-linear control behaviour of humans which has proven to be appropriate for many complex tasks. In fact, fuzzy logic offers a convenient tool for applying human experience and knowledge to automatic control by combining them with well understood control theories. This has been proven by numerous real-world applications, where even a highly complex chemical process, requiring manual control 24 hours a day, has been automated with less than 50 fuzzy rules [Altrock, 1997]. When the fuzzy logic controller adjusts set-points of the underlying single-loop controller, an operator can always identify the rules responsible for the action taken. It makes continuous improvement of the fuzzy logic controller easy.

The mathematical concept of fuzzy logic and fuzzy sets was first introduced in 1965 by Zadeh [Zadeh, 1965]. However, it was not until the mid-seventies that the idea of using fuzzy logic for industrial applications became viable. The first successful applications of fuzzy set theory to control of steam engine and boiler combination were reported by Mamdani [1974] and by King and Mamdani [1977]. Since that time researchers around the world are implementing this concept in various applications. Control engineering system is currently the most popular application of fuzzy logic. The initial applications of fuzzy logic control were followed by temperature and liquid level control in the industrial process plants, such as warm water plant [Kickert and Lemke, 1976], sludge waste water plant, [Tong *et al*, 1980], and control of pH [Paszkiewic and Lin, 1987]. A constant parameter fuzzy logic controller (CPFLC) is widely used in many applications because of its simplicity, easiness of implementation and good performance, especially with regard to steady state error. This has been proved for example in temperature control of a chemical plant, where response without overshoot or oscillatory behaviour (that results with traditional PID control) has been obtained [Keuer, 1997]. On the other hand, advanced fuzzy logic controllers, such as self-tuning, self-organising and adaptive controllers can be used to maintain the controller performance when plant's parameters vary and load disturbances are present. Summarising, many successful industrial applications of fuzzy logic control have been reported in recent past. The application of fuzzy logic control is receiving nowadays large momentum, mostly in Japan, especially for consumer products and home appliances. According to a survey by IEEE Spectrum [Self, 1990], many of consumer products, such as washing machines, microwave ovens, video camcorders, air-conditioners and elevators, utilise already fuzzy logic control.

Very often however, implementation is not done and the good performance is proved and tested only by simulations. The lack of mathematical tools for fuzzy logic controller design, which means that trial-and-error method has to be used, limits somewhat fuzzy control applicability at present. Implementation and tuning of a fuzzy logic controller requires relatively long processing time, approximately 2.5 times longer when compared to a typical full PID controller [Keuer, 1997], and large memory space. A faster processing time can be obtained if the implementation of fuzzy logic controller is based on off-line look-up table. However, the resolution of the fuzzy logic controller is then reduced and optimum values of output are pre-computed for every input value and stored in a RAM. In this case the processing speed is very high but there is an exponential growth of the required memory if the number of input and output variables, or the resolution, increase [Birou *et al*, 1996].

In motion control applications, the fuzzy logic control is still an emerging technology [Bose, 1989]. Until now there is not a standard method and optimal solution for the application of fuzzy logic controllers. Different approaches are used for design, tuning and optimisation of the controller algorithm, so that consistent performance is obtained [Li and Gatland, 1995], [John and Herschel, 1995]. A specific design of fuzzy logic controller is always required for a specific application. A fuzzy algorithm implemented in a general purpose microcontroller can only be used to control systems that have relatively slow dynamics (i.e., frequency response in the range of 0.1 - 1kHz) [Costa *et al*, 1995]. Although fuzzy logic control can bring with it advantages of robustness, the tuning and learning procedure can be complex [Vivek and Kwong, 1995], [Ghwanmeh, 1996], because capability of adaptation and automated learning, inherent to neural networks, does not exist.

Fuzzy logic applications have also been reported in a variety of other engineering areas such as pattern recognition, forecasting, reliability engineering, signal processing, monitoring and diagnosis. As application areas widen or specific applications are required, challenges of making fuzzy logic controllers widely available on various hardware and software platforms and of finding an adequate way to formulate and incorporate operator knowledge, become greater and greater.

A critical analysis of fuzzy logic control, given by Elkan [1994], states that fuzzy controllers are characterised by the following properties:

- fuzzy controllers use typically fewer than 100 rules, often even less than 20 rules;
- the knowledge within a fuzzy controller is usually shallow, both statically and dynamically;
- the knowledge within a fuzzy controller typically provides correlation between controller inputs and outputs;
- the numerical parameters of a fuzzy controller are tuned in a learning process;
- fuzzy controllers use fuzzy logic operators.

When the nature of fuzzy logic control is considered, it can be stated that the main areas in which fuzzy control can be applied are the following:

1. Processes which can be adequately controlled by humans and the process, for which controller is to be designed, has sensors to provide similar information to the one used by humans to control the process.

2. Processes which are at present controlled by (basically) linear control algorithms and need further development, resulting in non-linear control algorithms which are known by operators or process engineers. Mamdani [1994] states that "fuzzy logic is successful because it replaces the classical PID controller. When tuned, the parameters of PID controller affect the shape of the entire control surface. Because fuzzy logic control is a rule-based controller, the shape of the control surface can be individually manipulated for the different regions of the state space, thus limiting possible effects to the neighbouring regions only."

1.3 APPLICATION OF FUZZY LOGIC IN HIGH PERFORMANCE CONTROL OF VARIABLE SPEED ELECTRIC DRIVES

Industrial electric drives often require high dynamic performance over a wide range of speeds. This requires a control system with fast torque response such as direct torque control or vector control. In many applications, an electric drive is used solely for speed control. Response of the drive depends on the proper functioning of the speed controller. Output of the speed controller is torque command, which is the input for the torque control loop, if torque control loop exists. To achieve a high quality of performance from a servo drive requires that the drive responds to velocity errors quickly, by developing torque in the shortest possible time [Vas, 1990]. The speed controller is one of the most important parts of a drive system, because it significantly influences the overall performance of the drive [Bose, 1989]. Due to their simple structure and virtually no maintenance, sinusoidal permanent magnet synchronous machines have become attractive for vector control. The research undertaken in this project focuses on development of fuzzy logic speed controller for vector controlled variable speed drives with sinusoidal permanent magnet synchronous machines (SPMSM). Its results are however relevant to other types of high performance drives, such as vector controlled induction motor, separately excited DC motor, vector controlled synchronous reluctance motor, etc.

Selection of a control algorithm, followed by modelling and simulation of a drive, are the most important steps in designing a speed controller for a high performance drive. Speed control algorithms can be based on classical methods of control, such as PI or PID, or on artificial intelligence methods, such as fuzzy logic and neural networks. A standard approach for speed control in industrial drives is to use a PI controller. Recent developments in fuzzy logic have brought into focus a possibility of replacing a PI speed controller with a fuzzy logic (FL) equivalent [Bose, 1994; 1997a, 1997b]. Fuzzy controllers tend to produce better results than PI controllers in terms of response time and settling time [Heber *et al*, 1997]. Advanced control algorithms, such as self-tuning and self-learning, can also be applied in conjunction with fuzzy logic control algorithms. Tuning of the drive velocity loop can have a major impact on the ability of the drive to respond adequately to speed set-point changes or to load torque disturbances. It is essential therefore that the speed loop is well tuned, otherwise the overall dynamic capability or bandwidth of the drive will be limited in performance. The classical speed controller can be difficult to tune in practise. Even if speed controller is designed to yield a critically damped response, its performance can degrade as the load characteristic changes. If PID controllers are tuned aggressively, oscillatory behaviour results. Tuning the PID controller conservatively leads to poor transient response [Keuer, 1997].

Variable speed control of high performance drives is most frequently realised by means of vector control. Alternatively, DTC can be applied. In both cases torque estimator is used to track the rapidly changing torque demand and so compensate for the effect of the load disturbance. In DTC scheme for example, the errors between the estimated flux and torque and their reference values are used at every control instant to determine the appropriate switching signals to keep the speed error as small as possible [Vas, 1998; Bird *et al*, 1997]. Simple, constant parameter, speed controller cannot provide the same response to speed reference variation over the entire speed control region. This is a consequence of the non-linear nature of an electric machine. An improvement in speed response is possible if some kind of adaptive speed controller is used. A gain scheduling method is one possibility: speed controller's parameters are continuously changed as functions of the speed error. In many cases FL controller is used to adjust gains of the PI speed controller. Model reference adaptive scheme is another method that can be used to improve speed response but the performance will in this case degrade if the model is not accurate enough, especially during load disturbances.

Realisation of vector controlled high performance AC drives is based on cascaded speed and current control loops. Current control loops are the inner loops, while speed control loop is the outer loop. Feedback signals are obtained from appropriate sensors and/or estimators. Many successful applications of fuzzy logic control in AC drives have been reported recently [Heber *et al*, 1997], [Vas *et al*, 1994a, 1994b]. Fuzzy logic control has been used in DC and vector controlled AC motor drives [Vas *et al*, 1994a, 1994b], [Vas *et al*, 1997]. In 1994, Hitachi launched the first commercial vector drive with fuzzy logic control [Young, 1995], the Hitachi J300 drive.

1.4 AIMS OF THE RESEARCH

1.4.1 The scope and approach

The research focuses on design and investigation of the performance of the speed controller in the outer control loop of a vector controlled permanent magnet synchronous machine. A constant parameter fuzzy logic (CPFL) speed controller is designed for the outer control loop of the high performance variable-speed SPMSM drive. The controller's parameters are tuned manually until satisfactory performance is obtained. Previous experience has shown that the performance of the drive depends much more on the controller design than on the control structure. The controller can not maintain the same quality of the drive performance over a wide range of operating conditions and it cannot compensate for load disturbance or variation of motor inertia. This is a consequence of the fact that different operating conditions require different sets of controller parameters. A self-tuning or adaptive control scheme can be used to improve the drive performance. A novel self-tuning mechanism is therefore developed, in order to maintain good controller performance under all operating conditions.

The work undertaken in this project consists of three stages. Modelling of the three phase sinusoidal permanent magnet synchronous motor using differential equations in the reference frame fixed to the rotor, and principles of vector control are encompassed by the first stage. A constant parameter fuzzy controller and PI controller are then designed for the speed control loop based on nominal plant parameters. The second stage consists of investigation of the behaviour of the drive with both PI and fuzzy logic speed control and is based on numerical simulations, using Simulink and Fuzzy Logic Toolbox. A detailed comparative analysis of speed response to step speed reference application, load application and robustness to the motor inertia change is carried out. The third stage involves the development of a self-tuning FLC as a speed controller for the vector controlled permanent magnet synchronous machine drive.

1.4.2 Research objectives

Generally, the requirements for a high-performance motor drive are:

(i) Fast tracking of step change of speed reference, usually without overshoot.

(ii) The maximum speed change and the restoration time due to step load torque change must be kept as small as possible.

(iii) The steady state speed error has to be regulated to zero under all operating conditions.

(iv) The performance should be insensitive to system parameter changes, such as, for example total inertia.

These requirements may be met to a lesser or larger extent by using different control schemes in conjunction with a permanent magnet synchronous machine. This research project investigates vector control of a SPMSM, using both conventional PI speed control and fuzzy logic speed control.

The major research objectives of the project can be summarised as follows:

(i) Development of a fuzzy logic speed controller for vector controlled SPMSM with hysteresis current control.

(ii) Comparison of performance of the control scheme utilising FL speed controller with the scheme that relies on standard PI speed controller.

(iii) Investigation and development of self-tuning FL speed controller for a SPMSM.

The underlying objective of the whole project is to try to develop a novel speed control scheme, based on FL controller, that will provide superior performance when compared to the existing solutions.

1.4.3 Originality of the research

The originality of the research consists of:

1. A detailed comparative analysis of fuzzy logic and PI speed control in high performance AC drives. A detailed comparison between the operation of the drive with speed control by PI and FL technique is

made, for the cases of application of large step speed command other than rated, application of step load torque, operation with inertia other than rated and small step-wise speed reference change. The comparison is made on the basis of numerous results obtained by simulation of transients. Existing comparisons [Boussak and Bauer, 1996], [Ficcara *et al*, 1996] are usually done for a single operating point and/or operating regime.

2. Development of a self-tuning fuzzy logic speed controller with self-adjustment of the input and output scaling factors by means of another FLC. This adaptive speed controller aims at enabling an ac drive to maintain its good performance over a wide range of operating conditions.

The controller investigated in this research is a fuzzy PI-like speed controller where the size of control effort depends upon the size of the error and the rate of change of error. The controller output is integrated to obtain integral action. Although fuzzy logic can bring with it the advantage of robustness, the tuning procedure can be complex. The reason for this is that there are many parameters to be adjusted in a fuzzy controller. These include definition of the fuzzy set properties (shapes, size etc.) of the membership functions and scaling factors. The objective of development of a self-tuning FL controller is to establish a simple and easy-to-tune structure for a FL speed controller. This can be achieved by fixing some of the parameters and adjusting only scaling factors. Such an adaptive FL speed controller is developed and a comparative analysis is made between CPFL controller and self-tuning FL controller. Response to application of large and small step speed command other than rated, application of step load torque, operation with inertia other than rated and small step-wise reference speed change are studied by simulation.

1.5 THESIS OUTLINE

A brief overview of fuzzy logic control and its applications in process control, motion control, robotics and consumer products has been given in this chapter. High performance SPMSM drives and the concept of vector control are reviewed in **Chapter 2**. This includes a detailed literature survey of fuzzy logic speed control in high performance AC drives. The advantages of a vector controlled AC motor with respect to a DC motor in high performance drives are also discussed. Several methods of vector control of the sinusoidal permanent magnet synchronous motor drive are described in this chapter. These include rotor flux oriented control of a current-fed SPMSM and rotor flux oriented control of a voltage-fed SPMSM. Classification of standard current control techniques is included. Several current control strategies such as hysteresis and ramp-comparison are described.

The mathematical modelling of a permanent magnet synchronous machine, speed controller and inverter for adjustable speed drives is presented in **Chapter 3**. Modelling of hysteresis current controller is also described in this chapter. Some simulation results based on conventional PI speed control are given.

Basic operations with fuzzy sets and the construction of a fuzzy rule-based controller is given in **Chapter 4.** The chapter continues with presentation of the development of constant parameter FL speed controller. Simulation of vector controlled permanent magnet synchronous machines using FL speed controller is carried out next. The controller parameters and their effect on the controller's performance are investigated.

In **Chapter 5,** a detailed evaluation of the drive performance is presented by comparing the simulation results obtained with both PI speed controller and constant parameter FL speed controller. In particular, speed response for various reference speed settings, load rejection properties and robustness to total inertia changes are compared.

A new scheme for the self-tuning of input and output scaling factors of a FLC is described in **Chapter 6**. The tuning mechanism is based on heuristic fuzzy rule base. In this Chapter the influence of the scaling factors on the step speed response is investigated as well. Two possible approaches to adaptation are employed and four types of self-tuning mechanisms are developed and investigated using simulation procedure. The comparison is also made between the adaptive controllers and the off-line optimised CPFL speed controller, with regard to parameters used in the controller design, overshoot in the speed response, settling and rise time, and integral speed error criteria. Transients considered are variation of the step speed command with constant and variable total inertia. The overall performance of three types of adaptive self-tuning FLCs is finally discussed and conclusions regarding their applicability for different cases are drawn.

Finally, the discussion and conclusions of the research are given in **Chapter 7**. Some recommendations for further work are also included.

Chapter 8 lists the referenced literature. The thesis concludes with Appendices. **Appendix A** contains data of the permanent magnet synchronous motor, used in the simulations. **Appendix B** presents simulation block diagrams, formed in Simulink software. Flow charts used for development of two of self-tuning methods of Chapter 6 are included in **Appendix C**. Finally, **Appendix D** contains publications from this thesis.

CHAPTER 2

A REVIEW OF LITERATURE ON HIGH PERFORMANCE SPMSM DRIVES

2.1 INTRODUCTION

Recent developments in motion control technology are driven by ever increasing requirements in industrial applications for high performance, better reliability, and lower cost, these being enabled by advances in power electronics, control theory, and microprocessor technology [Henneberger, 1993], [Bose, 1994], [Le-Huy, 1994], [Vas and Drury, 1995], [Costa *et al*, 1995], [Vas and Stronach, 1996a], [Vas *et al*, 1996a, 1996b]. For a long time, the control of electrical drives has been dominated by analogue technology. During the last two decades, with the developments of microprocessor and peripheral circuits, digital technology has gradually replaced analogue technology in conventional control applications and allowed the implementation of advanced control algorithms that were previously unattainable. A modern motion control system, such as a manipulator arm or a positioning system, consists in general of one or several electrical drives. The control problem of such a complex system can be divided into two levels: high-level control and low level control. High level control involves the co-ordination of all the actuators to produce the desired motion from one point to another, following a planned trajectory. On the other hand, low-level motion control deals with the control of each motor drive on its own, for example induction motor drive control [Lorenz *et al*, 1994] and permanent magnet AC motor drive control [Jahns, 1994]. This research focuses on low-level motion control which involves the control of velocity and torque of an electrical drive.

Basically, a digitally controlled electrical drive consists of three main components: the electric motor, the power converter, and the digital control system [Novotny and Lipo, 1996]. The mechanical load is driven, directly or through a reducing gear, by the electrical motor, which is fed by the power converter. The requirements imposed on high precision AC motor drives are: high quality of dynamics, low torque ripple, low acoustic noise, minimum harmonic content of motor current and insensitivity to parameter changes [Vas, 1990], [Vas, 1992].

At present, numerous systems are available for such purposes including DC motor drives [Vas, 1992], [Vas, 1990], [Novotny and Lipo, 1996], variable-reluctance stepper motor drives [Miller, 1993], [Vas, 1998], [Nasar *et al*, 1993], permanent magnet synchronous motor drives [Vas, 1990], [Vas, 1998], [Nasar *et al*, 1993], [Boldea and Nasar, 1992] and induction motor drives [Vas, 1990], [Vas, 1992], [Vas, 1998], [Novotny and Lipo, 1996]. DC motors have been commonly used for high performance electric drives in the past. As the flux and torque control of a DC motor are inherently physically decoupled, a DC motor drive system can have very good dynamic behaviour. However, the advantages of DC motors can be offset by their large size, weight, high cost, and dedicated maintenance when compared with AC motors in general, including the permanent magnet synchronous motor [Vas, 1990], [Henneberger, 1993], [Novotny and Lipo, 1996], [Nasar *et al*, 1993], [Pillay and Krishnan, 1989], [Pillay and Krishnan, 1988].

Permanent magnet synchronous machines can be categorised by the shapes of their respective back-EMF wave-forms [Vas, 1990], [Nasar *et al*, 1993], [Henneberger, 1993], [Pillay and Krishnan, 1988] as

a) the brushless DC (BLDCM) motors fed by trapezoidal currents and having trapezoidal flux distribution

b) sinusoidal permanent magnet synchronous motors (SPMSM) having approximately sinusoidal air-gap flux-density distribution and fed by sinusoidal stator currents.

Under idealised conditions, each of these two types of permanent magnet synchronous motor drives is capable of delivering perfectly smooth instantaneous torque waveforms. Both the machine back-EMF and current excitation waveforms have to be perfectly either sinusoidal or trapezoidal for ideal smooth torque generation. Trapezoidal machines are advantageous when simple control and a minimum number of sensors are required. However, they have higher torque ripple and more limited high-speed range compared to SPMSM or induction machines. Sinusoidal back-EMF waveforms require that the machine's stator windings are sinusoidally distributed around the air-gap and that the radial magnetic flux density amplitude, generated by rotor magnets, varies sinusoidally along the air-gap. Sinusoidal phase currents are typically developed using a current-regulated inverter that requires individual phase current sensors and a high-resolution rotor position sensor to maintain accurate synchronisation of the excitation waveforms with the rotor angular position at every time instant. Any source of non-ideal properties, which causes either the phase current or the back-EMF waveforms to diverge from their purely sinusoidal shape, will typically give rise to undesired pulsating torque components. This project deals exclusively with sinusoidal permanent magnet synchronous machines.

Sinusoidal permanent magnet synchronous machines are showing increasing popularity in recent years for many industrial drive applications, ranging from general purpose drives to high-performance machine tool servos. This is because of their attractive characteristics in such key categories as power density, torque-to-inertia ratio and high efficiency [Vas, 1990], [Henneberger, 1993], [Pillay and Krishnan, 1990]. The rotor excitation flux required for operation of the machine is provided by the permanent magnets, and is essentially constant, which is in contrast to a synchronous machine with a rotor field winding, where the excitation flux is provided by the rotor field winding. Since the copper and iron losses are then concentrated in the stator, cooling of the machine is more easily achieved [Vas, 1990], [Vas, 1992], [Vas, 1998].

Rotors for SPMSMs can be designed using either surface-mounted or interior magnet configurations [Boldea and Nasar, 1992], [Vas, 1990]. Surface permanent magnet machine magnets are mounted on the rotor surface, while in the interior permanent magnet machines magnets are mounted inside the rotor. When the magnets are inside the rotor, a mechanically robust construction is obtained which can be used for high-speed applications since the magnets are physically contained and protected. However, the machine cannot be considered to have a uniform air gap. Thus, there is a reluctance torque from the saliency effects (L_q is greater than L_d) which significantly alters the total torque production mechanism of the machine. The resulting complex torque expression dictates that the control scheme will also be relatively complex in comparison to the machine with surface mounted magnets. In the case when the magnets are mounted on the rotor surface, the machine can be considered to have a large effective air gap, which makes the effects of saliency and effects of armature reaction negligible (thus the direct-axis magnetising inductance, L_d is equal to the quadrature-axis magnetising inductance, L_q) [Vas, 1990], [Vas, 1998]. Consequently the torque contains only the fundamental component. A surface mounted permanent magnet synchronous machine is less robust than the interior SPMSM and is usually used for operation in the base speed region only (e.g. up to the rated speed).

In this Chapter a literature review of the present technology, related to control of SPMSM drives, is presented. The comparison is made, based on performance, complexity of the control structure and practical realisation requirements, with other types of drives. Existing research shows that the vector control of SPMSM drives is at present focused on:

a) improvement of the two-axis motor model
b) enhancement of speed and torque response
c) improvement in robustness
d) minimisation of motor current harmonic content.

The major problems associated with implementation of a vector controlled SPMSM are:

a) variation of motor electrical parameters
b) presence of the position/speed sensor
c) variation of mechanical parameters.

2.2 VECTOR CONTROL OF A SURFACE MOUNTED PERMANENT MAGNET SYNCHRONOUS MACHINE

At present, existing control techniques of AC drives can be classified in two groups: one is the scalar open-loop voltage to frequency (volt/hertz) control for ordinary applications [Bose, 1989], and the other is vector (field-oriented) control for high-performance applications. A number of solutions have been proposed to implement vector control [Blaschke, 1972], [Gabriel *et al*, 1980], [Leonhard, 1986], [Bose, 1989], [Vas, 1990], [Vas, 1992], [Boldea and Nasar, 1992], [Vas and Drury, 1994], [Vas, 1998]. In vector control of the SPMSM drive the quadrature axis stator current is used to control the torque, and thus indirectly speed and position, of the motor up to the base speed. For the operation above base speed it is required to weaken the flux, which can be done using the stator *d*-axis current component [Boldea and Nasar, 1992]. The efficiency of the drive decreases because of the increase in copper loss due to the increase in the stator current.

The conventional linear controllers such as PI or PID have been used as speed controllers in many applications, in both AC and DC motor drives [Nonaka and Kawaguchi, 1991], [Sen, 1990], [Pilay and Krishnan, 1990], [Liu and Liu, 1990], [Vas, 1990], [Boldea and Nasar, 1992], [Chang *et al*, 1994], [Vas, 1998] including sensorless control of AC motor drives [Rajashekara *et al*, 1996]. However, these controllers are sensitive to plant parameter variations and load disturbance. The performance varies with operating conditions, and it is also difficult to tune the controller gain both on-line and off-line. Adaptive and optimal control techniques, considered in the past for DC drives only, are now being extended for AC drives. One reason is the recent availability of high speed and powerful DSPs. This makes possible implementation of, for example, adaptive fuzzy, adaptive fuzzy-neural or artificial neural networks in DC or AC drives [Vas and Stronach, 1996b], [Stronach *et al*, 1997], [Beierke *et al*, 1997]. Artificial-intelligence-based speed estimators [Stronach and Vas, 1998] for sensorless drives [Vas, 1998], [Vas *et al*, 1998] hold substantial promise for the future. Another reason is that with vector control the machine has a simple control structure that corresponds to the case of a separately excited DC machine. Therefore, the DC drive control algorithms can be applied directly. Adaptive control methods, such as self-tuning regulator, model reference adaptive control (MRAC) and sliding mode control give robust drive performance. Self-tuning adaptive control techniques have been applied in DC drives [Stronach *et al*, 1994], [Brickwedde, 1985]. In this method, the controller parameters are tuned to adapt to the plant parameter variations. Model reference adaptive control (MRAC) has also been applied in electric drive systems (Subsection 2.4.2.1). In MRAC, the output response is forced to track the response of a reference model (idealised model with fixed plant parameters) irrespective of plant parameter variations. The controller parameters are adjusted to give a desired closed-loop performance. The variable structure control using sliding mode was recently introduced into the field of controlled electric drive systems as well. With sliding mode control (SLMC), the control system can be designed to provide parameter-insensitive features, prescribed error dynamics, and simplicity in implementation. Applications of SLMC in AC servo drives have been reported (Subsection 2.2.2.3). Finally, expert systems, fuzzy logic and neural networks are emerging technologies that are characterised with a large potential impact on advanced machine control techniques (Section 2.4).

2.2.1 Control of sinusoidal permanent magnet synchronous machines

The design criteria for synchronous servomotors, to be used in machine-tool drives, manipulators and industrial robots, differ from those of conventional synchronous machines. High power/weight ratio (greatest possible power/motor mass ratio), large torque/inertia ratio (to enable fast acceleration), smooth torque operation (small torque ripple) even at very low speed, high speed operation, high transient torque capability (quick acceleration and deceleration), high efficiency and compact design are required [Slemon, 1994], [Jahns, 1994], [Vas, 1990], [Vas, 1992], [Nasar *et al*, 1993], [Henneberger, 1993], [Pillay and Krishnan, 1989]. These requirements can be adequately met by the permanent-magnet synchronous machine employing vector control.

In recent years, vector (field-oriented) control techniques have been employed in order to enable conversion of an AC machine into an equivalent separately excited DC machine. Thus field oriented control enables decoupled (independent) control of flux and torque in an AC machine by means of two independently controlled (fictitious) stator currents, as in a separately excited DC machine. To obtain true vector control, stator current components must be placed into a pre-defined position with respect to one of the flux space vectors. Basically there are two possible types of vector control of an SPMSM, namely: rotor flux oriented control and stator flux oriented control. Rotor flux oriented control is by far the most common method applied in practical realisations [Vas, 1990], [Vas, 1998], [Pillay and Krishnan, 1989], [Ho and Sen, 1995]. The sole reason is that, compared to the other orientation possibility, rotor flux oriented control asks for the simplest control system structure. A SPMSM is converted into its equivalent separately excited DC machine in the simplest way by selecting a reference frame fixed to the rotor flux. The difficulty is the fact that stator *d-q* axis current components, needed for this decoupled flux and torque control, do not exist in the actual machine; instead, the machine is fed with three phase *a, b, c* currents. The problem is overcome by including co-ordinate transformation as an interface between the control system and the machine. Since the flux produced by the permanent magnets can be assumed to be constant, the electromagnetic torque can be varied by changing the quadrature-axis stator current expressed in the rotor reference frame. Thus a constant torque is obtained if the stator quadrature current component is constant. Maximum torque per ampere of stator current is obtained if the stator current space vector is at all times placed in quadrature with the rotor flux space vector. Quick torque response is obtained if the quadrature-axis stator current is changed quickly, e.g by the application of a current-controlled pulse-width-modulated inverter.

Precise rotor position information is required in any variable speed drive comprising SPMSM. It can be obtained directly from the rotor shaft by monitoring the rotor angle or rotor speed. In practice, rotor speed can be measured by using an analogue tachometer and the rotor angle can be monitored by, say, a resolver. Analogue tachometers have about 0.1% accuracy and at low-speed operation of a servo drive, very high-speed resolution is necessary to obtain accurate result [Vas, 1998]. This problem can be solved by using the same digital encoder for sensing the rotor position and rotor speed [Vas, 1990], [Henneberger, 1993], [Pillay and Krishnan, 1990]. It should be noted that at high speeds optical encoders can have limited accuracy and are susceptible to temperature variation, since the monitoring device has to be mounted within the motor enclosure. Resolvers are inherently accurate, but they must be combined with high-resolution digital circuits to achieve and maintain high accuracy over a wide speed range.

2.2.1.1 Rotor flux oriented control of current-fed SPMSM

In this case it is assumed that the machine is fed from an ideal current source, hence the phase currents from the controller can be regarded as being directly impressed into the machine stator windings. The control structure for the current fed machine is simpler than that of the voltage fed machine [Vas, 1990], [Vas, 1992], [Vas, 1998], [Novotny and Lipo, 1996], [Boldea and Nasar, 1992], [Sokola *et al*, 1992]. The concept of current-fed machine is commonly used for practical realisation. The stator *q*-axis current command is obtained from the speed controller while the stator *d*-axis current command is set to zero. These two components are then converted from the rotational reference frame into three phase current commands in the stationary reference frame, by means of co-ordinate transformation. The angle for co-ordinate transformation is the one supplied directly from the position sensor. Current control is performed in the stationary reference frame.

2.2.1.2 Rotor flux oriented control of a voltage-fed SPMSM

If the machine is regarded as being fed from a voltage source, the control system becomes much more complex [Boldea and Nasar, 1992], [Vas, 1992], [Sokola *et al*, 1992]. Current control now takes place in the rotating reference frame and the outputs of the control system are voltage references rather than current references. The inner control loop controls the stator current *d-q* axis components, while the outer control loop performs speed control. The reference value for the direct axis stator current in the rotor flux reference frame is again set to zero. The quadrature axis stator current reference is obtained as an output of the speed controller whose input is the error signal between the commanded rotor speed and the actual rotor speed. The error signals between the commanded current values and the actual current values serve as the inputs to the respective direct and quadrature axis current controller. The outputs from the current controllers are the direct and quadrature axis stator voltage reference values.

However, these reference values are not decoupled and therefore decoupling voltages are needed. The voltage reference values are then transformed to the stationary reference frame by utilisation of the co-ordinate transformation [Vas, 1990], [Trzynadlowski, 1994]. The resultant voltages are the phase voltage reference values in the stationary reference frame, which are fed to the inverter control circuit. It should be noted that in the case of a voltage-fed machine, measured stator currents have to be transformed into the rotor reference frame using co-ordinate transformation.

2.2.2 Alternative methods of vector control of a SPMSM

2.2.2.1 Improved *d-q* model of a SPMSM

One possibility of improving the behaviour of a vector controlled SPMSM consists in utilisation of a modified machine model in the development of the control system. The idea of a modified model is that it accounts for one or more phenomena that are neglected in the standard, constant parameter model. A constant parameter motor model is widely used in the implementation of vector control for AC drives. The mathematical model of the SPMSM is derived using simplifications and it neglects numerous phenomena in the complex physical structure of the machine, such as variation of stator resistance, iron loss and magnet flux change with temperature [Sebastian *et al*, 1986], [Pillay and Krishnan, 1989]. The resulting type of controller is called a parameter dependent controller because its performance would be degraded when the motor parameters change in wide range of operating conditions [Stumberger, 1996].

A method to improve the performance of vector controlled SPMSM drives using improved (modified) models has been proposed [Senjyu *et al*, 1996]. In this method the compensation strategy is based on re-tuning the controller by using the accurate information produced from the improved motor model, with additional consideration of the iron loss. The influence of iron loss should not be neglected and it depends on the operating frequency. A few assumptions have been made in order to develop voltage equations that account for the stator iron loss: harmonic components of current and flux are neglected, there is no saturation in the magnetic circuit, the rotor iron loss and stray load loss are neglected, and the stator iron loss is produced in the equivalent eddy current circuit in the stator. Using the above assumptions, the equivalent eddy current winding model in phase reference frame, which has a winding resistance and a winding inductance, is set up. By transforming the three-phase winding axes into *d-q* axes, the *d-q* winding model, which includes the equivalent eddy current windings on the stator, is derived. The control voltages to the SPMSM are composed of both the decoupling voltages and the torque control voltage. The decoupling voltages are calculated using measured actual currents, actual rotor speed and the equivalent iron-loss resistance.

2.2.2.2 Speed and position sensorless SPMSM drives

Conventionally, the speed of an electrical machine can be measured by DC tachogenerators, while rotor position can be measured by either using resolvers, or absolute or optical encoders. One of the most active areas of development during recent years involving AC machine drives has been the rapid evolution of new control techniques for elimination of the rotor position sensor, conventionally used in high performance drives. The issue of sensor elimination in SPMSM is challenging because accurate position information is essential for proper operation of the drive. As a result, sophisticated observer estimation techniques are generally required to extract position information from stator phase current and voltage measurement.

Sensorless vector drives have become more important for industry because of their advantageous features [Vas, 1998]: reduced total hardware complexity and cost, increase in mechanical robustness and reliability of the drive, and increased noise immunity. To improve the robustness of the system and reduce cost it is necessary to minimise the number of sensors. Furthermore, an electromechanical sensor increases the system inertia, which is undesirable in high-performance drives. It also increases the maintenance requirements. In very small motors it is impossible to use electromechanical sensors. On the other hand, speed and position estimation can be performed by using software-based state-estimation techniques where stator voltage and current measurements are performed [Vas, 1998].

A detailed discussion regarding the main techniques of sensorless control of AC drives is given in the recent text book [Vas, 1998]. These include the conventional estimators such as Kalman and Luenberger observers [Vas, 1998], [Sanz *et al*, 1996], or make use of special characteristics of the motor such as saliency (geometrical, saturation) effects [Vas, 1998], [Sathiakumar *et al*, 1994].

Speed estimators using artificial intelligence (neural networks, fuzzy logic, and fuzzy-neural networks) are also discussed [Vas, 1998]. Such an approach has many advantages over the conventional techniques (e.g. it does not rely on the many simplifying assumptions related to the conventional mathematical models, it can be robust to parameter variations, it offers the possibility of reduced computational time, etc.) [Vas, 1998].

A review of the various schemes and methodologies used for speed sensorless operation of induction motors and position sensorless operation of permanent magnet, synchronous reluctance, and switched reluctance motors is given in [Rajashekara *et al*, 1996]. At zero speed most methods for rotor position estimation fail. The magnets do not induce any voltage and therefore no information on the magnetisation is accessible. In general, to solve the problems which occur at low frequencies, a number of special techniques have been proposed by various investigators: for SPMSMs [Schroedl, 1994], [Ertugrul and Acarnley, 1994], for induction machines [Ohtani *et al*, 1992], [Jansen *et al*, 1994] for brushless DC motors [Ogasawara and Akagi, 1991] and for synchronous reluctance motors [Matsuo and Lipo, 1995].

The improved sensorless control of a SPMSM drive, with parameter variation compensation, has been introduced recently [Kim and Şul, 1995], [Wijenayake, 1995]. In this method the rotor position, speed and rotor magnet temperature of a SPMSM are estimated in an on-line parameter identification scheme by measuring stator currents, voltages and temperature. Machine parameter variations due to saturation and temperature (change in stator resistance, magnet flux variation and core losses) are taken into consideration at every speed, including zero speed: therefore, the estimation accuracy is considerably increased. Inclusion of the core loss resistor and modification of the *d-q* equivalent circuit also enable the use of this equivalent circuit in efficiency optimisation algorithms, where the objective is to find the optimum operating point at which the total losses are minimised.

2.2.2.3 Sliding mode control method

Recently, sliding mode control has been investigated in vector controlled induction motor drives [Kim *et al*, 1996], [Yoon *et al*, 1996], [Hikita,1988], in permanent magnet synchronous motor drives [Namuduri and Sen, 1987], [Wei,1993], [Ghribi and Le-Huy, 1994], [Sepe, 1991], [Chiricozzi *et al*, 1996] and in DC motor drives [He *et al*, 1997], [Silva *et al*, 1996]. Sliding mode control is also used in power converter applications [Raviraj and Sen, 1995]. Sliding modes are an important feature of the variable structure control system: the controller is switched between two distinct control structures. In general, the design of a sliding-mode controller can be divided into two phases: the hitting phase and the sliding phase. Before the system reaches the switching surface (hitting phase), control is directed towards a switching surface, and when all the states of the controlled system are constrained to lie within a switching hyperplane the sliding mode occurs (sliding phase). Once the state of the controlled system enters the sliding mode, the dynamics of the system are detected by the choice of sliding hyperplane and are independent of uncertainties and external disturbance. Like MRAC, the combination of sliding mode and variable structure control system gives robust performance against plant parameter variations and load disturbance effects. The ultimate goal of the controller is to maintain invariant velocity control of the motor in the presence of varying mechanical parameters such as rotor inertia and load torque.

Variable structure control (VSC) is characterised with the chattering problem, that causes vibration and wearing of the mechanical parts of the actuator. Theoretically, VSC is based on the assumption of the infinite switching action on the sliding surface. It is, however, difficult to meet this assumption in practise because of the limitations of the sampling and switching frequency. Thus, ideal sliding motion having infinite switching action cannot be realised and this non-ideal sliding motion is a factor of producing an undesirable steady-state error in practical applications, [Chern and Wu, 1993], [Hashimoto *et al*, 1988].

Many investigations have been undertaken to overcome these problems. One possibility is the implementation of an adaptive approach. The gain of the speed controller is re-tuned based on the gain self-tuning method [Chiricozzi *et al* ,1996], or the gain is tuned based on the estimation of the mechanical parameters [Sepe, 1991]. In the gain self-tuning method there are two major loops in the controller structure, the fast loop comprises the motor, inverter and current controller. The slow outer loop consists of parameter estimator and the re-tuning algorithm for the speed controller. The inner loop is used to regulate motor current so that the dynamic performance of the motor is maintained. The outer control loop adjusts the gains in the speed controller to effect invariant velocity control in the presence of changing mechanical parameters. This is accomplished in two steps. Firstly, the mechanical parameters are estimated using input and output measurements of the mechanical dynamics of the motor. These measurements are provided by a speed controller and state filter, respectively. Secondly, new gains for the speed controller are computed using estimates of the mechanical parameters. The gains in this multi-loop controller are chosen so that the closed-loop electrical dynamics of the motor are much faster than both the open-loop and closed-loop mechanical dynamics. The required controller re-tuning algorithm is based on pre-specified, closed-loop pole locations for the mechanical dynamics of the motor. Because the parameters of the motor change much more slowly than the state, the outer control loop, which is responsible for adaptation, is implemented at a slower rate than the inner control loop. However, since it is desirable to directly estimate the parameters of the mechanical dynamics, the input and output measurements, fed into the parameter estimator, are sampled at the same rate as the mechanical dynamics. In this method the adaptation approach is implemented with the on-line parameter estimation so that a long execution time is required and high sampling rate is not possible.

In general, in sliding-mode control, the upper boundary of uncertainties, which include parameter variations and external load disturbance, must be available. However, the boundary of uncertainties is difficult to obtain in advance for practical applications. A fuzzy sliding-mode controller is investigated to resolve this difficulty, in which a simple fuzzy inference mechanism is used [Lin and Chiu, 1998]. In [Chiricozzi *et al*, 1996] the fuzzy inference mechanism is used to compute the strip area which corresponds to the load variations. In [Lu and Chen, 1994] the best features of self-organising fuzzy control and sliding mode control are combined to achieve rapid and accurate tracking control of a class of non-linear systems. The fuzzy rule base is used to approximate the equivalent control through self organising, and the variable structure control effort is used to compensate for the approximation error and to provide exponential convergence of the sliding variable. In [Wu and Liu, 1996] the sliding modes are used to determine the best values for parameters in the fuzzy control rules to improve the robustness of the fuzzy control. Lin and Chiu [1998] used the fuzzy controller to adjust the sliding surface of the sliding-mode controller. In [Ting *et al*, 1996] a fuzzy control scheme is constructed from the concept of sliding mode. If the original control rules are inappropriate, an adaptive mechanism modifies these rules.

2.3 DIRECT TORQUE CONTROL

In general, the conventional direct torque control of an AC machine supplied by a voltage-source inverter (VSI) involves the direct control of the flux-linkage space vector (e.g. stator flux-linkage space vector) and electromagnetic torque by applying optimum voltage switching vectors of the inverter which supplies the motor [Vas, 1998]. In 1995, the first commercial direct torque controlled induction motor drive was introduced by ABB [Schofield, 1995]. A detailed discussion of numerous DTC schemes for synchronous motors and induction motors is available in [Vas, 1998]. The main advantages of the DTC, when compared with vector control are [Vas, 1998]:

i) absence of coordinate transformation (which is required in all of the vector-controlled drive implementations);
ii) absence of a separate voltage modulation block (required in vector controlled drives);
iii) absence of voltage decoupling circuits (required in voltage-fed vector controlled drives);
iv) absence of current controllers;

v) only the sector where the flux-linkage space vector is located, and not the actual flux-linkage space-vector position, has to be determined;

vi) minimal torque response time.

However, the main disadvantages of a conventional DTC scheme are [Vas, 1998]:

i) possible problems during starting and low speed operation and during changes in torque command;

ii) requirement for flux and torque estimators (same problem exists for vector drives);

iii) variable switching frequency;

iv) high torque ripple.

2.4 FUZZY LOGIC CONTROL OF HIGH PERFORMANCE DRIVES

Fuzzy logic can be applied in the closed-loop control of a drive system, and it can provide fast and robust control [Vas, 1998], [Vas and Stronach, 1996a], [Sousa and Bose, 1995]. The drive system may be based on a DC or AC machine. Since vector-controlled AC drives and DC drives have identical dynamic models, the same fuzzy control principle is valid in either case.

Previous research in fuzzy logic control (FLC) has shown improvement in controlling induction motor drives [Vas *et al*, 1997] and permanent magnet motor drives [Cerruto *et al*, 1995]. The robustness of the standard fuzzy logic controller is limited to a certain extent because variations of the motor parameters and load disturbance take place in a wide range of operating conditions. Adaptive control is thus required to make fuzzy controllers able to cope with operating condition changes. In this section, several adaptive control methods will be discussed.

2.4.1 Constant parameter FL controller (CPFLC)

A constant parameter FL controller has already been successfully implemented in high performance vector controlled drives [Baghli *et al*, 1997a], [Eminoglu and Atlas, 1996]. A fuzzy logic speed controller is used in induction motor drives [Fodor *et al*, 1996], [Zhen and Xu, 1996], [Afonso *et al*, 1997], [Bebic and Jeftenic, 1998], [Heber *et al*, 1997], [Baghli *et al*, 1997a], DC motor drives [Eminoglu and Atlas, 1996], [Monti and Scaglia, 1997], brushless DC motor drives [Donescu *et al*, 1996] and switched reluctance motor drives [Abut *et al*, 1997]. Fuzzy logic control is usually applied in the speed control loop where *error* and *change of error* are the input signals, while torque current command is the output of the controller. However, recently a greater number of inputs of a FL speed controller for a brushless DC motor drives [Bodin *et al*, 1997] has been proposed. A FL speed controller is designed with four inputs and two output variables for DC motor drives [da Silva *et al*, 1997]: speed error, change of speed error, amature current and load disturbance are the inputs, while the amature voltage is the output of the controller. In position control applications [Silva *et al*, 1996], [Senjyu *et al*, 1996], a FL controller is employed to regulate the rotor position and its output is speed reference.

In [Feng and Chen, 1996], a load observer is implemented in conjunction with a FL speed controller to obtain satisfactory control performance. In this case, the observer is used to compare the calculated and measured speed producing an output of torque current. In the DTC scheme [Bird *et al*, 1997], a FL speed controller is used to improve the dynamic speed response with an ability to quickly recover from speed drops due to load disturbances. In this application, the CPFL controller has two inputs that are speed error and change of speed error. The output of the CPFL controller is a torque current command which needs to be increased or decreased according to the speed error and change in speed error. In general, a fuzzy logic based controller is established by interpreting rules that are based on experience and formed in a decision table relating the input and output of the controlled system [Fodor *et al*, 1996] and [Liaw and Cheng, 1995].

2.4.2 Adaptive FL controller

In practical applications the optimal setting of the constant controller parameters varies under different working conditions. When the parameters are fixed, ideal performance cannot be achieved in all the cases. A solution to this problem is addressed by adaptive controllers, whose aim is to maintain consistent performance of a system by adjusting the controller parameters to the varying conditions. The principle of this control method has been given in detail in many books [Davies, 1970], [Aström and Wittenmark, 1990].

Adaptive control theory has been studied for several decades. Different adaptation schemes, such as self-tuning (S-T) [Harris, 1981], [Gawthrop, 1987] and model reference adaptive control (MRAC) are applicable in high performance drives [Vas, 1998]. The design of a classical adaptive control system is based on mathematical modelling and the implementation of such a system is usually complex due to the computationally intensive algorithms. On the other hand, the adaptation of a FL control system can be achieved by using the same approaches as in the classical adaptive systems. The difference will be in the adaptation mechanism which can be implemented using fuzzy reasoning. Thus, the design of the adaptation algorithm can be made simpler since no mathematical manipulation is required.

Adaptive controllers in high performance drives can be classified into two types: indirect and direct controllers [Bose, 1997a, 1997b], [Dash *et al*, 1997]. An indirect adaptive artificial-intelligence-based (AIB) controller contains a real-time identification model and also requires inputs and output of the plant at every sampling instant. However in direct AIB controllers, the plant monitor is represented by a performance table which is used to determine the current response using such quantities like the error and the change of the error over two consecutive sampling periods [Vas *et al*, 1996b]. In this case, the adaptation can be change of scaling factors, change of the rules and/or change of the membership functions.

2.4.2.1 Model reference adaptive controller (MRAC)

This type of adaptation (Fig. 2.1) has been applied in induction motor drives [Le-Huy, 1995], [Cerruto *et al*, 1995] and PMSM drives [Silva and Le-Huy, 1997], [Zhen and Xu, 1996]. The system response and the reference model output are compared. The error between the model output and the actual speed, and its change, are calculated in every sampling period.

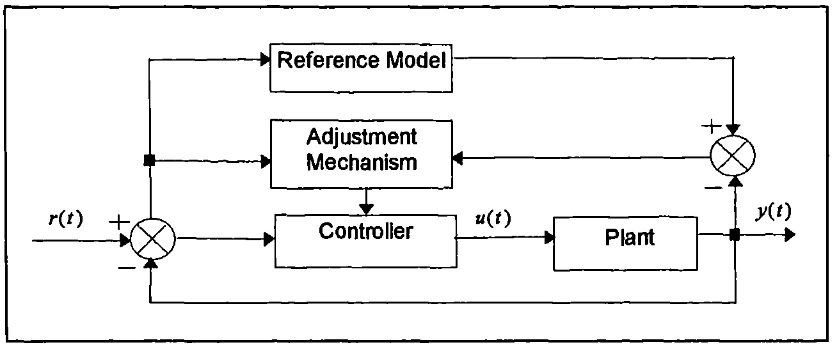


Figure 2.1: Model reference adaptive control (MRAC)

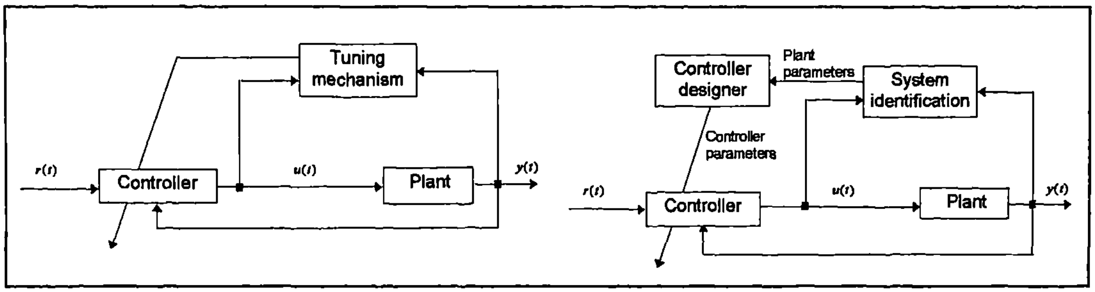
In the adaptation scheme, the adaptation mechanism produces an auxiliary control signal to compensate for the deviation of the performance due to changes in operating or load conditions. This adaptation approach does not provide learning capability but its implementation is much simpler since the knowledge base is not modified [Ta-Cao and Huy, 1996], [Le-Huy, 1995]. However, MRAC speed control systems do not achieve consistently satisfactory performance over a wide range of speed demand, especially at low speed and there is no defined rule to guide designers to choose the adaptation gains. In the paper [Cheung *et al*, 1996], a MRAC with adaptation of PI controller gains is proposed to control the speed of a DC motor drive and its superiority over conventional MRAC is demonstrated by simulation results.

MRAC, based on neuro-fuzzy speed controller, has been proposed [Fischle and Schroder, 1997] to dissipate the torsion vibrations between motor and load. This is because the behaviour of many real drive systems is substantially influenced by non-linearity, especially coulomb friction and backlash. Furthermore, the structure and parameters of these non-linearities may not be exactly known. The method is at first examined by a simulation example and then verified experimentally with a laboratory electric drive system. However, the learning times are still too long for this application and need relatively high speed processing. The adaptive speed controller and rotor resistance estimator, based on a fuzzy logic approach for high-performance indirect vector controlled induction motor drive, have been proposed [Ta-Cao and Huy, 1996]. The simulation results have shown that speed and flux response are improved for varying inertia, load torque and stator resistance variations. A robust fuzzy logic speed controller [Goureau and Ibalden, 1996] is based on fuzzy model reference learning control which employed a CPFL controller and FL adapter. The input of the second fuzzy controller is the error between the output of the system and the signal given by the model reference of the system. To provide the learning capability the controller parameter adaptation must be included in the structure of the adaptation mechanism and as a result fuzzy model reference learning control (FMRLC) for controlling induction motor has been proposed [Liaw and Cheng, 1995]. It consists of two fuzzy logic controllers with fixed shape membership functions, fixed width but with flexible centres of all the membership functions at zero to represent the fact that the fuzzy controller initially doesn't know how to control the machine. These centres are shifted by the output of the dynamic fuzzy logic adapter and suitable rules are fired to ensure that the actual motor response follows the reference speed. The control method is based on the fixed number of 25 rules for any operating condition. The processing time of the fuzzy logic algorithm is mostly affected by the number of rules.

2.4.2.2 Self-tuning FL controller

The basic idea of a self-tuning adjustment mechanism was conceived originally as a means of handling the initial tuning of a controller. Thus, after the initial (self) tuning, the adjustment mechanism is not required and can be disabled. If not disabled, it can provide continuous adaptation to changes in the system [Wellstead and Zarrop, 1991].

The self-tuning control can be classified as either explicit or implicit. In the first approach (Fig. 2.2a) the "tuning mechanism" observes the signals from the control system and tunes the parameters of the controller to maintain performance. A simple example of explicit or direct adaptive control is the gain scheduling control of an inertia varying speed control system [Bose, 1997a, 1997b], provided that inertia parameter can be identified on real-time basis.



a) b)

Figure 2.2: Self-tuning control; a) direct self-tuning; b) indirect self-tuning

In the second approach or indirect self-tuning (Fig. 2.2b) the system is identified using measured input and output data and then the parameters of the main controller are modified on the basis of the system identification. Digital self-tuning algorithms for speed and current (armature and field) controllers have been implemented in a DC drive [Stronach *et al*, 1998]. The controller is also implemented in direct torque control of induction motor drive with a single self-tuning speed controller and also in vector control of induction motor drive incorporating four self-tuning controllers: speed controller, torque producing current controller, magnetising current controller and flux producing current controller. The design of the controllers is based on a pole-placement algorithm with plant model parameter data being derived from a real-time estimator. The RLS form of estimator [Wellstead and Zarrop, 1991] is chosen in preference to Kalman or Luenberger types because of lower computational demands and simpler design. The simulation and experimental results show that the controllers can be implemented from an initial open loop configuration thereby eliminating the need for any detailed a *priori* design [Stronach and Vas, 1995]. The controller gains are initially zero so that the drive starts from an open loop control configuration. In [Stronach *et al*, 1994] a discrete transfer function-based pole placement algorithm is used to determine the controller gains. Once again, a single RLS estimator is used to produce the required parameter values for twin-loop, current and speed loop model simultaneously. This eliminates the need for any a priori detailed controller design. An improved fuzzy-tuned PI controller (IFPIC, Fig. 2.3) has been implemented in an electrical drive system [Vas *et al*, 1994b]. A fuzzy self-tuning mechanism is used to tune the gains of the PI speed controller. The effects of the sampling interval, width of fuzzy sets and number of rules are also investigated. It is noted that it is possible to obtain a good response by using only 16 rules rather than using all 72 fuzzy rules.

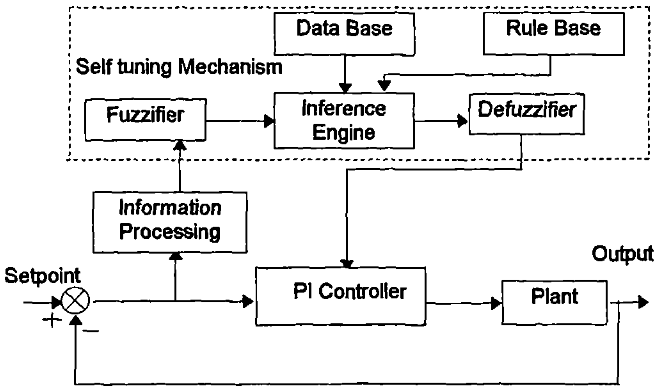


Figure 2.3: Improved fuzzy-tuned PI controller (IFPIC)

In [Wellstead and Zarrop, 1991], the gain scheduling is classified as another alternative to the self-tuning control method. A fuzzy gain self-tuning method based on the cycle information for a PI speed controller has been implemented for a PMSM drive system [Chiricozzi *et al*, 1996]. A sliding trajectory is used to define the strip area between the optimal and the effective response. Based on the fuzzy inference mechanism, PI gain parameters are tuned in each cycle with the computed strip area as a fuzzy input variable, so that the effective response can be improved. The optimal response is obtained based on known mechanical and control system time constants. Another type of fuzzy gain scheduled PI speed controller is proposed in [Panda *et al*, 1997], where the proportional gain and integral gain of the PI speed controller are defined by the membership functions. At the beginning of the transition period, the controller needs a large control signal either to accelerate or decelerate the motor as quickly as possible. In order to produce a large signal, the PI controller should have a large proportional gain and the integral gain should be small to avoid overshoots. When the motor reaches the reference speed, a small control signal is required to maintain the motor speed at the desired value. That is, the PI controller should have a small proportional gain, and a large integral gain to overcome the steady state error. Based on this principle, proportional and integral gains are varied between the maximum and minimum values to get satisfactory control performance.

For most industrial processes, the degree of non-linearity varies with respect to the operating regions. As the controller works in different regions, different controller gains are usually required to obtain optimal performance. For example, aggressive control action is demanded in the transient period to shorten the rise time while mild control action should be issued in the steady-state to avoid too long a settling time. It is difficult to achieve good responses in both the transient and steady state operation with fixed controller parameters [Liaw and Cheng,1995]. In this paper, the fuzzy logic controller is employed during transients, while the PI controller is used in steady-states. In another publication [Parasiliti *et al*, 1996] the adjustable input and output fuzzy logic scaling factors are implemented to compensate and maintain the controller performance at different stages of operation. This results in different control actions due to the firing of different control rules and naturally affects the output signal. The adaptation technique is based on the analysis of the general system dynamic response in terms of required control inputs in different operating conditions. The adaptation method is based on a look-up table to reduce processing time. In most cases the on-line computation of input and output scaling factors, instead of the application of a look-up table, will take a longer processing time and will be more complicated for implementation.

2.4.2.3 Fuzzy neural network

In conventional fuzzy logic controllers no formal procedures exist for the direct incorporation of the expert knowledge during the development of the controller. The structure of the FL controller (number of rules, the rules themselves, number and shape of membership function, etc.) is arrived at through a time consuming tuning process which is essentially manual in nature [Fodor *et al*, 1996], [Zhen and Xu, 1996], [Afonso *et al*, 1997], [Bebic and Jeftenic, 1998], [Heber *et al*, 1997], [Baghli *et al*, 1997b], [Eminoglu and Atlas, 1996], [Monti and Scaglia, 1997], [Donescu *et al*, 1996], [Abut *et al*, 1997]. The ability to automatically 'learn' characteristics and structure that may be obscured to the human observer is, however, inherent in neural networks. A fuzzy logic-type controller having a neural network structure offers the advantages of both the ability of fuzzy logic to use expert human knowledge and the learning ability of the neural networks, and overcomes their disadvantages, the lack of a formal learning procedure for the fuzzy controller [Vas and Stronach, 1996b].

In [Beierke *et al*, 1997] the fuzzy-neural network is used to represent the non-linearity of the input-output relationship of the magnetising current approximator. Therefore, no explicit mathematical model is required. The approximator is used to maintain a constant magnetising current in all modes of operation (for all speed and load values) using slip frequency as the input to the approximator. If a mathematical model is used, this requires a value for the rotor time constant. The rotor time constant varies with temperature and degree of saturation. Experimental results [Beierke *et al*, 1997] have proved that the drive using the fuzzy-neural function approximator can be successfully implemented. Furthermore, in [Beierke *et al*, 1997] the speed controller is implemented based on a fuzzy-neural scheme, while the other three controllers (torque-producing current controller, magnetising current controller and flux producing current controller) are based on a PI controller. The good experimental results confirm the validity of the scheme. This technique offers some advantages. Fuzzy rules and the membership functions of the speed controller are established based on an automatic and systematic design procedure. The non-linear approximator of input-output mapping, using a fuzzy-neural speed controller, has been implemented in DC drives by Vas and Stronach [1996a].

In adaptive fuzzy-neural speed control of a DC drive [Vas and Stronach, 1996b], the number of rules and membership functions (their numbers and shapes as well) are determined automatically on-line. This technique is simple to implement, provides robust solution and leads to minimum topological configuration as it contains only four rules. This also implies rapid self-learning and quick convergence. The experimental results [Stronach and Vas, 1995] confirm that good performance is maintained even when minimum configuration of fuzzy-neural controller is used rather than twin-loop PI controllers [Stronach and Vas, 1995]. The clustering-based approach with a five layer network [Stronach and Vas, 1995] is used to initialise the number of membership functions, the centre and width of each fuzzy set membership function. Further fine tuning using a back-propagation-type algorithm is performed [Vas and Stronach, 1996b]. Furthermore, the fuzzy rules are directly generated from numerical data. The input-output data˙ space is subdivided into a pre-specified number of regions to each of which a triangular membership function is assigned. Each input-output data set thus determines a fuzzy rule, according to which membership function each data item has a minimum degree of belongingness. Therefore, conflicting rules are eliminated based on the 'winning' output node principle [Vas and Stronach, 1996a, 1996b], [Stronach *et al*, 1997]. A detailed discussion of the structure and development of a speed fuzzy-neural controller for electromechanical drives is given by Stronach *et al*, [1997]. It is noted that this technique offers a structure which enables "automated" design, requiring a minimum of human interventions for the tuning while maintaining high dynamic performance. Practical results that illustrate satisfactory drive performance for a variety of DC and AC drives incorporating adaptive fuzzy-neural controllers are reported in [Vas and Stronach, 1996a, 1996b].

Neuro-fuzzy control for field oriented control of induction motors is implemented in [Bagli *et al*, 1997b]. The output surface of the fuzzy controller is approached by a neural network. The neural network structure used for this controller is presented, as well as is the learning process necessary to prepare the neural network controller to act like a fuzzy one. Experimental results on position control and speed control show the effectiveness of the neural network controller used.

2.4.3 Other possible concepts of control using fuzzy logic

Previous research for an induction machine drive [Liaw and Cheng, 1995] shows that the fuzzy controller can give a fast transient response but leads to non-zero steady state error. On the other hand a conventional PI controller has good stability and zero steady-state error but the transient response is slower than with the fuzzy controller. The transient response can be much improved by employing the fuzzy controller during the transient periods, while the PI controller is used to yield better accuracy in the steady state. The changes of controller modes are controlled by a switching mechanism.

The quality of the controller design is dependent on how accurately the model describes the actual system behaviour. Though this is not an issue for a system where a model can be derived straight from the laws of physics, for most systems the inaccuracy of the model can cause problems in the controller performance. To overcome this drawback, techniques such as adaptive and robust control can be developed [Goureau and Ibaliden, 1996]. Most conventional control design methods are based on some form of explicit plant models (e.g. a transfer function model to design a PI controller or a state space model to design Kalman filter, etc.). Fuzzy logic control on the other hand has mostly been based on implicit plant models. The reason for this is that most fuzzy controllers are designed using the engineering intuition of the designer and/or expert advice, both of which draw on an implicit model of the plant behaviour derived from common sense and experience. This method has been investigated in [Le-Huy *et al*, 1995] to find the best fuzzy controller parameters such as membership functions, scaling factors and dominant rules for controlling a BLDC machine.

Self-organising control based on a fuzzy-neural approach has been successfully implemented in DC and AC drives [Stronach *et al*, 1997]. The explicit model is not required and overall development time is reduced compared to the conventional approach. Minimal configuration is obtained in terms of controller structure and number of fuzzy rules, while the initial and fine-tuning of width and centres of membership functions is performed on-line based, on a back-propagation algorithm. The self-configuring of the high-performance electromechanical drives, using a DSP, is also investigated in [Stronach *et al*, 1998], [Vas and Stronach, 1996a].

To control the plant more precisely and effectively, the on-line rule tuning and two sets of scaling factors for rise time and steady state have been proposed [Ghwanmeh, 1996]. This combined method modifies and generates fuzzy rules according to the operating point either in the transient or in the steady state. This method is self-organizing (Fig. 2.4) and has been implemented for on-line process control [Ghwanmeh, 1996]. The adaptation mechanism will generate the best rules according to the operating condition. This approach has the capability to reduce processing time because less rules will be fired compared to the fixed rule fuzzy logic controller. The modification of fuzzy rules is dependent on the two sets of off-line computed look-up-tables. Processing time can be reduced because a minimum number of rules will be used in the transient state and more rules are generated to obtain zero steady state error.

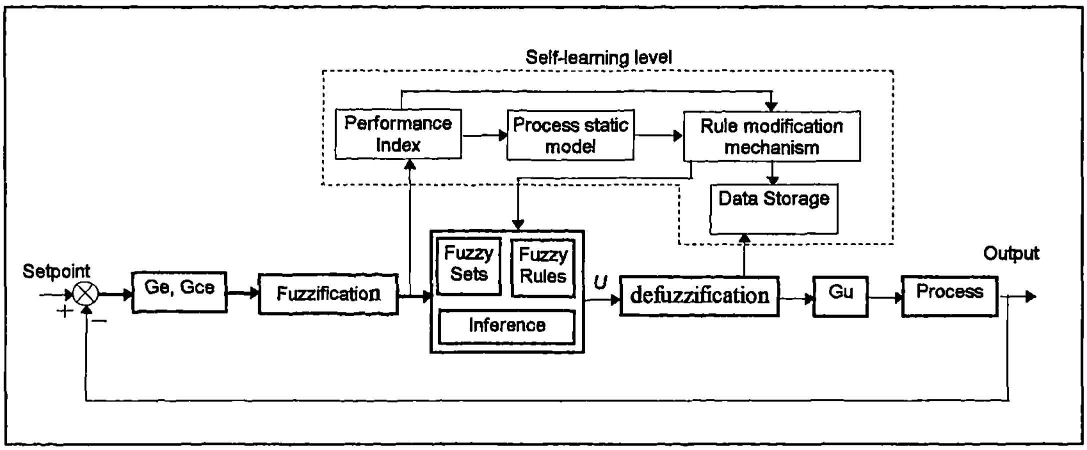


Figure 2.4: Self-organising fuzzy logic controller

2.5 CURRENT CONTROL METHODS FOR PWM VOLTAGE SOURCE INVERTERS

Inverter application areas and topologies are still expanding as a result of improvements in semiconductor technology which offers higher voltage and current ratings as well as better switching characteristics [Baliga, 1995]. The main features of a modern power electronic inverter such as high efficiency, low weight and small dimensions, fast operation and high power densities can be achieved through the use of the so called switch mode operation, in which power semiconductor devices are controlled in ON/OFF fashion (no operation in active region). This leads to different types of pulse width modulation (PWM), which is the processing technique applied in inverter applications. PWM is a high speed process involving switching speeds up to a few kHz for motor control application. In variable speed motor drives the desirable voltage and frequency output are obtained by converting the fixed DC source voltage into AC voltages. One important advantage of the PWM technology is that the harmonics are moved to the high frequency area and these harmonics can be relatively easily filtered. The low-harmonic content of the output current is almost zero, so that smooth low speed operation is achieved, free from torque pulsation and cogging.

Depending on the type of DC source supplying the inverter, inverters can be classified as voltage source inverters (VSI) or current source inverters (CSI). VSIs can be either voltage or current controlled. In a voltage-controlled inverter, feed-forward voltage control is employed, since the inverter voltage is dependent only on the supply voltage and the states of the inverter switches. Current-controlled VSIs require sensors of the output currents which provide the necessary control feedback. The theory of VSIs and CSIs is widely available [Trzynadlowski, 1994], [Vas, 1998], [Novotny and Lipo, 1996]. The implementation of the PWM VSIs for high performance AC drives is discussed in [Vas, 1998]. Controlled current operation of a PWM VSI inverter offers substantial advantages in improving the motor dynamics and is most suitable for high-performance applications [Brod and Novotny, 1985], [Schauder and Caddy, 1982]. VSI has a faster current response compared with CSI and can be applied in a PWM mode more easily due to the low impedance [Ide *et al*, 1995]. If the SPMSM is fed from a current controlled PWM-VSI with fast current control loops, than the machine can be considered as current fed (i.e. fed with impressed stator current). The current in each motor phase follows a reference waveform, derived from the control algorithm and position signal provided by a position sensor mounted on the motor shaft, so that synchronism is ensured for any speed.

Recently, new emerging technologies such as neural networks and fuzzy logic methods have been increasingly applied for current control of PWM inverters in variable speed drive applications.

2.5.1 Standard current control techniques

In general there are two methods of current control of PWM voltage source inverters (VSI). The first group encompasses all the methods where current control is executed in stationary reference frame and the SPMSM can in this case be considered as current-fed. The second group encompasses methods of current control in rotational reference frame and the SPMSM has to be regarded as voltage-fed. In this case, the control system of SPMSM is more complicated as it has to include a so-called decoupling network. A high dynamic current control for AC motor drives with excellent transient response can be realised. The voltage source inverter topology is commonly used in SPMSM drives and the addition of closed-loop current control yields a configuration often referred to as a current-regulated voltage source inverter. This closed-loop system behaves like a very fast current source.

In vector control of permanent magnet synchronous machines correct positioning of stator current space vector with respect to selected flux linkage space vector is required [Holtz, 1992]. Hence it is of utmost importance to provide satisfactory current control of the PWM inverter, which will provide good tracking between actual and reference currents. As already mentioned, all the available current control techniques may be classified into two broad groups [Vas, 1992], [Trzynadlowski ,1996], [Sokola *et al*, 1992]:

- *Current control in stationary reference frame*

  The actual phase currents are measured and compared with phase current references. The errors between reference and actual currents are then processed in a convenient way and firing signals for inverter switches are created so as to reduce the errors. The machine is considered as current fed which leads to a very simple structure for the controller. Current control loops are normally implemented using analogue means.

- *Current control in rotational reference frame*

  Here current control loops are closed in a rotational reference frame, which may be rotor flux or stator flux oriented, depending on the selected type of field orientation. In other words, stator *d-q* axis current components are controlled rather than phase currents. The measured phase currents have to be transformed into the rotational reference frame. The resulting *d-q* axis currents are compared with *d-q* axis current reference. The errors are processed and used further to build *d-q* axis voltage references. When the current control is performed in a rotational reference frame, the machine is considered to be voltage-fed and the overall controller structure is more complicated than in the previous case.

Several current control strategies have been proposed in recent years [Trzynadlowski ,1996], [Malesani and Tomasin 1993], [Sokola *et al*, 1992], [Tripathi and Sen, 1992], [Huy, 1989]. The most common strategies of current control in a stationary reference frame are hysteresis and ramp comparison methods. Each scheme has its own advantages and drawbacks with regard to accuracy and dynamic response over the entire speed range. The advantages of hysteresis current control are in its simplicity and provision of fast response and good accuracy, because it acts quickly [Sokola *et al*, 1992]. However, the fixed band hysteresis controller has some drawbacks: it generates a random PWM voltage and the switching frequency varies during the fundamental period. increase in the switching frequency results in irregular inverter operation and increases the switching losses. The current ripple is relatively large and theoretically can reach double the value of the hysteresis band [Tripathi and Sen, 1992]. The ramp comparison controller has the advantage of limiting the maximum inverter switching frequency to the frequency of the triangular carrier waveform and produces well defined harmonics [Sokola *et al*, 1992]. On the other hand, this controller produces a magnitude and phase error in the phase currents. Also, multiple crossings of the ramp may become a problem when the rate of change of the current error exceeds that of the ramp.

Current control in a rotational reference frame can be achieved using either PI controllers or using predictive current control. In the predictive control scheme the switching instants of the inverter power switches are determined by calculating the required voltage to force the motor currents to follow the references. This control scheme provides constant switching frequency and lower current ripples. Until now a large variety of current control strategies, different in concept and performance, have been developed and described. The main objective of many investigations is to obtain quick current response in the transient state and low harmonic content in the steady state. Their implementation in the AC drive systems depends on the machine type, the power level, and the semiconductor devices used in the power converter. Apart from these basic current control strategies, a number of advanced methods have been proposed recently. These are summarised in the next sub-section.

2.5.2 Advanced current control methods

In recent years, many advanced current control methods have been proposed, based on self-tuning and adaptive control. Most of them have not yet been applied in practice. This is because these techniques are complex to implement and require extensive knowledge of the system parameters. Among the existing current control methods, hysteresis current control is still investigated intensively by many researchers [Enjeti *et al*, 1992], [Sonaglioni, 1995], [Malesani and Tomasin, 1993]. Fixed switching frequency, reduced current ripple and lower complexity are the main factors that can improve the fixed band hysteresis current control scheme. Hysteresis and predictive current control schemes, when employed in conjunction with adaptive and optimal control, can give better performance in terms of robustness and easiness of implementation. On the other hand, a real time adaptive controller requires a long execution time so that a pre-programmed PWM switching pattern is frequently used [Enjeti *et al*, 1992]. Recently, knowledge based current control techniques have been introduced for the future development of the high performance AC drives. Expert systems, fuzzy logic and neural networks are the emerging technologies that have a large potential impact on advanced current control techniques.

A self-adjusting hysteresis band [Sonaglioni, 1995] is proposed to overcome the disadvantages of the current controller with fixed hysteresis bands. This is an improved hysteresis current control technique for voltage source inverters, which ensures good control of the phase output voltage. This allows to approach the optimal condition of 'pulse centering' which characterises the classic three-phase voltage vector control, resulting in a minimisation of the current ripple. Due to its simplicity, self-adjusting capability and robust performance, the method is suitable for an integrated implementation, by means of hybrid circuits or ASICs. An improved digital method was proposed in [Malesani and Tomasin, 1993]. It suggested that the proper hysteresis band should be calculated in order to obtain zero current error and fast transient recovery. Although this method guarantees fixed switching frequency and zero current error, it is not so easy to implement, and it requires an appreciable amount of fast computations. This method has been improved by proposing a predictive digital hysteresis current controller [Malesani and Tomasin, 1993], which uses digital calculation to predict the proper hysteresis band in order to commutate at fixed frequency. The method does not require any A/D conversion, only timing measurements, and can be easily implemented in a single micro-processor for a whole three-phase control system. A sinusoidal band hysteresis current controller is another method [Tripathi and Sen, 1992] to reduce current ripple and improve harmonic content. The hysteresis band varies sinusoidally over a fundamental period. Thus the ripple can be varied with the current magnitude thereby reducing the current ripple content. However, the switching frequency is higher with the sinusoidal band. A non-optimal and adaptive hysteresis-band current control is proposed to improve fixed hysteresis-band current control. In this method, the band is modulated as a function of the system parameters to maintain the modulation frequency nearly constant [Bose, 1990]. An adaptive and optimised current controller (AOR) [Ackva and Reinold, 1992] is proposed based on multilevel hysteresis comparators and a corresponding switching logic. In steady state, the controller applies a voltage vector to the machine that is directly adjacent to the machine voltage (in complex plane) thus leading to low switching frequencies and low harmonic currents. Due to the ingenious structure of the multilevel comparators, the controller adapts automatically to the machine voltage. Compared to other current control schemes the AOR features a very low switching frequency. In the transient state the AOR selects exactly those voltage vectors that force the current as quickly as possible to its new reference value. The dynamic performance is as good as for the three-phase bang-bang controller and the hardware implementation is very simple.

In the neural network method [Kazmierkowki *et al*, 1995], initial training is required to teach the network about the plant behaviour. The inputs to the neural network are three phase current errors and the outputs are inverter switching patterns. Current error can be randomly generated at the neural network input and the back-propagation method is used to update the weights so as to decrease the current errors. Since the hysteresis current control for the VSI is known, the neural network must learn the dynamic behaviour of the hysteresis current control. It can be trained on-line or off-line. This method has been proposed in [Kazmierkowki *et al*, 1995] to drive an induction motor.

2.6 SUMMARY

A survey of literature, related to high performance SPMSM drives, is presented in this chapter. Advantages and disadvantages of the AC and DC motor drives are reviewed. A comparison is made between permanent magnet synchronous machines and brushless DC motors in terms of the shapes of back-emf wave-form, flux-density distribution, control structure and smooth torque generation. A comparison is also made for the surface-mounted and interior magnet configuration of the permanent magnet synchronous machines. AC drives control can be classified in two groups: one is the scalar voltage to frequency control and the other is vector control for high-performance applications.

The general requirements to realise high performance variable speed SPMSM drives, such as co-ordinate transformation as an interface between the control system and the machine, orientation of the current with respect to flux space vector, current sensors and position sensor to provide rotor position information, are explained. Rotor flux oriented control of current fed SPMSM is chosen for further work because it requires the simplest control structure. The discussion also covered several alternative methods of vector control of a SPMSM, e.g. use of improved *d-q* model of the motor, speed and position sensorless drives and sliding mode control. A short explanation regarding the advantages and disadvantages of direct torque control (DTC) of an AC machine is given as well.

The successfulness of the constant parameter FL speed controller for high performance drives is discussed. It is noted that fuzzy logic is not only used for speed control but it can also be applied as a load observer and estimator. This chapter also reviews the requirement for the adaptive FL controller in order to compensate the variation of parameters, load and inertia under different working conditions. Due to advances in power electronics, control theory, and microprocessor technology, three main types of adaptation techniques, based on fuzzy logic control, are possible to implement. These are model reference adaptive control (MRAC), self-tuning and fuzzy-neural network. Furthermore, self-organising and self-configuring controllers are becoming more important in high-performance drives.

The advantages and disadvantages of the conventional current control strategies, e.g. hysteresis and ramp-comparison control, are discussed. In general, all the available current control techniques are classified into two broad groups: one is current control in stationary reference frame and the other one is current control in rotational reference frame. The chapter also discussed advanced current control schemes such as self-adjusting hysteresis band current control, optimised current control and neural-network based current control.

CHAPTER 3

MODELING AND SIMULATION OF SINUSOIDAL PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVES

3.1 INTRODUCTION

AC machines are widely used in variable speed drives with applications ranging from computer peripherals, robotics and machine tools to railway traction, ship propulsion and rolling mills. Scalar speed control methods such as constant volts/hertz control, slip frequency control and voltage control are only able to control the steady-state behaviour of the machine. Since instantaneous torque depends on instantaneous values of currents, the torque developed by the machine exactly corresponds to the torque command in steady-state only, while the dynamic response is generally sluggish and slow. Vector theory of electrical machines overcomes this problem by transforming the original models in three-phase domain into models in two-phase rotating domain. Appropriate transformations are applied to all variables and parameters of the original model, resulting in models that are much simpler than the original model and thus easier to analyse. The mathematical models, that describe the steady state and dynamic behaviour of AC machines in terms of rotating *d-q* axes are well-known and available in many books [Vas,1990], [Trzynadlowski, 1994].

An electric machine is used to do a certain amount of mechanical work by driving a load. In practice, variations in the magnitude and duration of the load take place and a requirement for variable speed operation is often present. Operation of a drive is specified in terms of transient and steady state torque, speed and/or position response requirements. The electrical machine in the drive system should match the load requirements, without exceeding the voltage and current limitations imposed by the machine rating and the associated power electronics. When a new drive system is to be designed it is very useful to initially model the drive system by using appropriate mathematical expressions. The model can then be used to perform a numerical experiment or simulation. Simulation is thus an inexpensive and safe way to experiment with the system. However, the accuracy of the simulation results depends entirely on the accuracy of the model of the drive system.

The aim of this chapter is to describe the modelling of a permanent magnet synchronous motor drive and to evaluate the entire drive behaviour so that the performance can be predicted even before the drive is built. If the characteristics of the entire drive can be expressed mathematically, then the performance can be analysed with a computer program to solve the ordinary differential equations (ODE). In the past, analysis of the AC motor drive behaviour was done using transfer functions [Lipo, 1974]. The transfer function method, in general, enables analysis of a machine behaviour at constant speed of operation, under sinusoidal feeding conditions. Its applicability in modern AC drives is therefore limited. The advent of computers has made transfer function approach redundant. Analysis of a drive behaviour is nowadays always performed using appropriate software packages or custom-designed programmes on digital computers. Different drive components are at first modelled, often with varying degree of accuracy, and an overall drive model is then built. The entire drive model is further implemented in software and the drive performance is analysed by simulation. Several of the external signals that influence the system would also have to be modelled in order to understand and simulate their effects on the system (load disturbance). The process of modelling and simulation forms an important stage in the system design and is often used to determine suitable values for a certain set of parameters, in order to yield the desired performance of the system.

The system under consideration consists of three main parts: the motor and mechanical load, the inverter, and the controller, as shown in Fig. 3.1. In high performance AC drives the d-q model approach is used to convert the highly non-linear three phase AC motor model into an equivalent two phase model. Vector control [Vas,1990] is normally used in AC machines to convert them into an equivalent separately excited DC machine which has highly desirable control characteristics. For high performance servo drives, hysteresis or ramp comparison current controllers are often used to ensure that the actual currents flowing into the motor are as close as possible to the reference. The inverter is therefore operated as a current-controlled power supply. In this evaluation, the entire drive system is at first modelled and then simulated. The modelling includes the non-linear *d-q* axis equations of the motor, speed controller, the current controlled inverter and vector control circuit. The inverter is assumed to be ideal, and hence it is lossless. The effects of the magnitude of the hysteresis window, variation of DC link voltage, computation time and sampling time on speed response of the drive are then investigated by simulation.

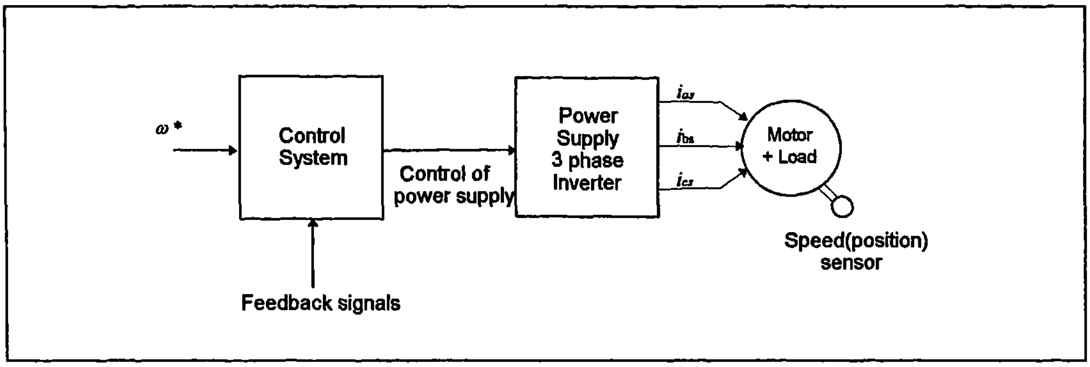


Figure 3.1: A basic configuration of drive system.

3.2 MATHEMATICAL MODEL OF SINUSOIDAL PERMANENT MAGNET SYNCHRONOUS MACHINES

Recent research [Pillay and Krishnan, 1988], [Pillay and Krishnan, 1989] has discussed, in considerable depth procedures for mathematical modelling and simulation of sinusoidal permanent magnet synchronous machines.

The model of the SPMSM, with surface-mounted permanent magnets, is similar to the model of a smooth-air-gap synchronous machine with excitation winding and without damper windings [Vas, 1992]. For the purposes of generality, the mathematical model of sinusoidal permanent magnet synchronous machines is based on a three phase synchronous machine with excitation winding. This is because the SPMSM has a sinusoidal back emf and requires sinusoidal stator currents to produce a constant torque [Pillay, 1989], [Pillay and Krishnan, 1989], [Vas, 1992]. The general theory of electric machines is fully applicable for the derivation of a convenient SPMSM mathematical model [Vas, 1992]. In this section, the equivalent circuit and mathematical model of a synchronous machine are derived at first, based on the original *a-b-c* reference frame. The machine is assumed to have three identical phase windings on the stator and a single winding on the rotor. That is, all the three stator windings, designated as *as, bs and cs,* are symmetrically displaced by 120 electrical degrees from each other (Fig. 3.2). As a SPMSM with surface mounted magnets corresponds to a smooth air-gap synchronous machine with excitation winding, the rotor and stator structures in Fig. 3.2 are both cylindrical. The field winding on the rotor is designated by *rf* and indicates a rotating winding. The three-phase domain

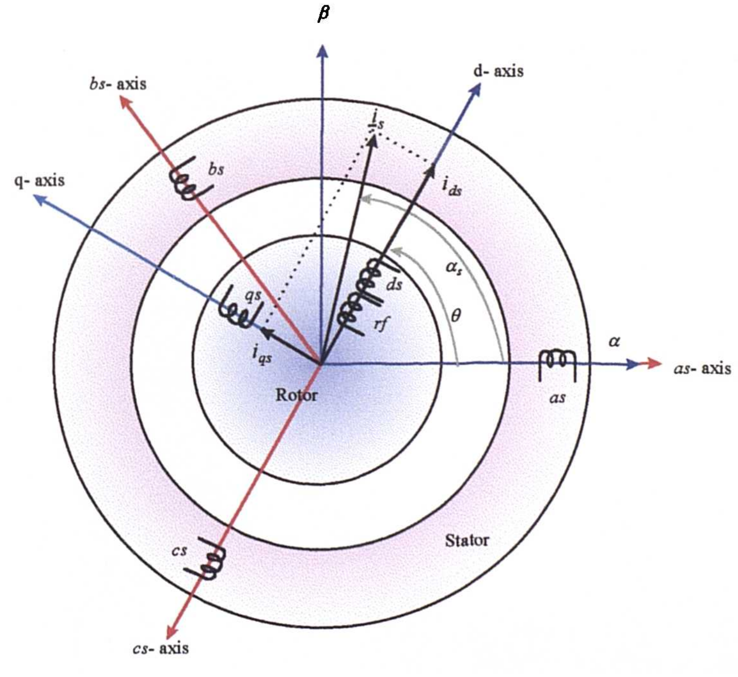


Figure 3.2: Stator and rotor winding magnetic axes, common rotor-fixed reference frame and current space vector.

model is firstly specified and then the corresponding *d-q* axis model is developed. The final stage is the modification of the model of a synchronous machine with a field winding to represent a SPMSM.

3.2.1 Phase-variable voltage equations in the original reference frame

The phase-variable form of the three-phase stator and a single-phase rotor voltage equations is at first formulated in their natural reference frames. Thus the stator voltage equations are formulated in the stationary reference frame fixed to the stator (equation (3.1)) and the rotor voltage equation is formulated in the rotating reference frame fixed to the rotor (equation (3.2)). Equation (3.1) is a matrix equation, while equation (3.2) is a single first-order differential equation. Thus:

$$[v_{abcs}] = [R_s][i_{abcs}] + \frac{d[\psi_{abcs}]}{dt} \tag{3.1}$$

$$[v_{rf}] = [R_{rf}][i_{rf}] + \frac{d[\psi_{rf}]}{dt} \tag{3.2}$$

where column matrices v_{abcs}, i_{abcs} and ψ_{abcs} are defined by

$$[v_{abcs}] = \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix}; \qquad [i_{abcs}] = \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}; \qquad [\psi_{abcs}] = \begin{bmatrix} \psi_{as} \\ \psi_{bs} \\ \psi_{cs} \end{bmatrix} \tag{3.3}$$

and for rotor

$$[v_{rf}] = v_{rf}; \qquad [i_{rf}] = i_{rf}; \qquad [\psi_{rf}] = \psi_{rf} \tag{3.4}$$

Magnetic coupling exists between all of the stator windings and the rotor winding. The total flux linking the stator windings is the sum of the contributions from the stator and the rotor circuit. The stator flux linkages are therefore given with the following matrix equation:

$$\left[\psi_{abcs}\right]=\left[\psi_{abcs(s)}\right]+\left[\psi_{abcs(r)}\right] \tag{3.5}$$

Similarly, the total flux linking the rotor winding is sum of the contributions from the rotor and the stator circuits:

$$\left[\psi_{rf}\right]=\left[\psi_{rf(s)}\right]+\left[\psi_{rf(r)}\right] \tag{3.6}$$

where

$$\left[\psi_{abcs(s)}\right]=\begin{bmatrix} L_{as} & L_{abs} & L_{acs} \\ L_{abs} & L_{bs} & L_{bcs} \\ L_{acs} & L_{bcs} & L_{cs} \end{bmatrix}\left[i_{abcs}\right] \tag{3.7}$$

$$\left[\psi_{abcs(r)}\right]=\begin{bmatrix} L_{as,rf} \\ L_{bs,rf} \\ L_{cs,rf} \end{bmatrix}\left[i_{rf}\right] \tag{3.8}$$

$$\left[\psi_{rf(s)}\right]=\begin{bmatrix} L_{rf,as} & L_{rf,bs} & L_{rf,cs} \end{bmatrix}\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \tag{3.9}$$

$$\left[\psi_{rf(r)}\right]=\left[L_{rf}\right]\left[i_{rf}\right]=L_{rf}\,i_{rf} \tag{3.10}$$

Equations (3.1) to (3.10) can be expressed in the matrix form as follows:

$$\begin{bmatrix} v_{abcs} \\ v_{rf} \end{bmatrix}=\begin{bmatrix} [R_s] & [0] \\ [0] & [R_s] \end{bmatrix}\begin{bmatrix} [i_{abcs}] \\ [i_{rf}] \end{bmatrix}+\frac{d}{dt}\begin{bmatrix} [L_{abcs(s)}] & [L_{abcs(r)}] \\ [L_{rf(s)}] & [L_{rf(r)}] \end{bmatrix}\begin{bmatrix} [i_{abcs}] \\ [i_{rf}] \end{bmatrix} \tag{3.11}$$

where $R_s = diag\left(R_s \quad R_s \quad R_s\right)$, since the stator windings are identical. As the air gap of the machine is constant, all the machine inductances, except for those between stator and rotor windings, are constant. The self inductances and mutual inductance of a smooth air-gap synchronous machine can be represented as

$$\begin{aligned} &L_{as} = L_{bs} = L_{cs} = const \\ &L_{abs} = L_{acs} = L_{bcs} = const \\ &L_{rf} = const \end{aligned} \qquad (3.12)$$

$$\begin{aligned} &L_{as,rf} = L_{af} = L_{af}\cos\theta \\ &L_{bs,rf} = L_{af} = L_{af}\cos(\theta - \frac{2\pi}{3}) \\ &L_{cs,rf} = L_{af} = L_{af}\cos(\theta - \frac{4\pi}{3}) \end{aligned} \qquad (3.13)$$

where the angle θ defines the instantaneous position of the rotor winding's magnetic axes with respect to the magnetic axis of the stationary stator phase *as* axis illustrated in Fig. 3.2.

It is evident from equation (3.13) that mutual inductances between stator and rotor windings are dependent upon the rotor position. Hence the voltage equations which describe the behaviour of the machine will contain time derivatives of time varying inductance. The model, using original *a-b-c* reference frame, has time-dependent inductances and is not convenient for numerical simulation and control purposes [Vas, 1990], [Boldea and Nasar, 1992]. The time-varying coefficients can be eliminated if the stator and rotor equations are referred to a common frame of reference. The *d-q* axes are made to rotate with the angular velocity of the rotor, and such selection is termed the rotor reference frame. In the case of a perfectly smooth air-gap machine, the speed of the common reference frame can be arbitrarily selected [Novotny and Lipo, 1996]. If the machine possesses some degree of saliency on the rotor, the common reference frame has to be selected as attached to the rotor. As this is usually the case with synchronous machines, a model of a synchronous machine is always formed in the rotor reference frame.

The model of the machine is finally completed with the electromagnetic torque equation and the equation of mechanical motion [Vas, 1992]:

$$T_e = -L_{af}\left\{\sin\theta\, i_{as}\, i_{rf} + \sin(\theta - \frac{2\pi}{3})\, i_{bs}\, i_{rf} + \sin(\theta - \frac{4\pi}{3})\, i_{cs}\, i_{rf}\right\} \qquad (3.14)$$

$$T_e - T_L = \frac{J}{P}\frac{d\omega}{dt} \tag{3.15}$$

where P and J the are pole pair number and inertia, ω is the electrical angular speed of rotation, T_L is the load torque and $\theta = \int \omega \, dt$.

3.2.2 Transformation to the d,q reference frame fixed to the rotor

The SPMSM is very similar to the standard wound rotor synchronous machine except that the SPMSM has no damper windings and the excitation is provided by permanent magnets instead of a field winding. The high-energy product permanent magnets have linear magnetisation characteristics and may be replaced by a fictitious field winding carrying constant field current.

As permanent magnets are mounted on the rotor surface, and as permanent magnets are characterised with very low permeability, a SPMSM behaves like a machine with a large effective air-gap. The effects of saliency can therefore be neglected and the machine can be regarded as having a uniform air-gap [Vas, 1992].

The transformation of the synchronous machine equations from the *a-b-c* phase variables (Section 3.2.1) to the *d-q* variables transforms all sinusoidally varying inductances in the *a-b-c* frame into constants in the *d-q* frame [Novotny and Lipo, 1996]. Park's transformation can be used to convert stator winding quantities such as current, voltage and flux linkage to the *d-q* reference frame that is attached to the machine's rotor.

As the common reference frame is selected to be firmly attached to the rotor, there is no need to transform the rotor equation. Only the stator equations need to be transformed from the stationary reference frame to the rotating reference frame. This is accomplished using the following transformation matrix:

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{os} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos(\theta) & cos\left(\theta - \frac{2\pi}{3}\right) & cos\left(\theta + \frac{2\pi}{3}\right) \\ -sin(\theta) & -sin\left(\theta - \frac{2\pi}{3}\right) & -sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} \tag{3.16}$$

The inverse correlation is given by:

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 1 \\ cos\left(\theta - \frac{2\pi}{3}\right) & -sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ cos\left(\theta + \frac{2\pi}{3}\right) & -sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{os} \end{bmatrix} \tag{3.17}$$

where θ is, as previously stated, the instantaneous rotor position. The same transformation applies to stator currents and stator flux linkages.

Application of this transformation yields the model of a smooth airgap synchronous machine, without damper winding and with excitation winding, in the reference frame fixed to the rotor (index "*r*" is omitted further on in symbols of the rotor winding).

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega\psi_{qs} \tag{3.18}$$

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega\psi_{ds} \tag{3.19}$$

$$v_f = R_f i_f + \frac{d\psi_f}{dt} \tag{3.20}$$

$$\psi_{ds} = L_s i_{ds} + M i_f \tag{3.21}$$

$$\psi_{qs} = L_s i_{qs} \tag{3.22}$$

$$\psi_f = L_f i_{fs} + M i_{ds} \tag{3.23}$$

$$T_e = \frac{3}{2} P M i_f i_{qs} \tag{3.24}$$

$$T_e - T_L = \frac{J}{P}\frac{d\omega}{dt} \tag{3.25}$$

Zero sequence components are omitted as the supply is considered to be balanced and zero sequence currents cannot appear in the star connected stator windings without the neutral lead. The *d*-axis of the common reference frame coincides with the magnetic axis of the rotor excitation winding. The model of a SPMSM follows directly from this model as it is only necessary to remove the excitation winding dynamics [Vas, 1990]. The model is valid under the following assumptions which are made in the derivation [Sebastian *et al*, 1986], [Pillay and Krishnan, 1989]:

i. Saturation is neglected although it can be taken into account by parameter changes.
ii. The induced EMF is sinusoidal.
iii. Eddy currents and hysteresis losses are negligible.
iv. There are no field current dynamics.
v. There is no cage winding on the rotor.

Omission of the rotor equations (3.20) and (3.23) and substitution of the term $M i_f$ in (3.21) with the magnet flux ψ_m lead to the model of a SPMSM:

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega\psi_{qs} \tag{3.26}$$

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega\psi_{ds} \tag{3.27}$$

$$\psi_{ds} = L_s i_{ds} + \psi_m \tag{3.28}$$

$$\psi_m = L_s i_{qs} \tag{3.29}$$

$$T_e = \frac{3}{2} P \psi_m i_{qs} \tag{3.30}$$

where the permanent magnet flux is considered to be constant. Elimination of the stator *d-q* axis flux linkages yields the final form of the model.

$$v_{ds} = \left(R_s + L_s s\right) i_{ds} - \omega L_s i_{qs} \tag{3.31}$$

$$v_{qs} = \left(R_s + L_s s\right) i_{qs} + \omega\left(L_s i_{ds} + \psi_m\right) \tag{3.32}$$

$$T_e = \frac{3}{2} P\left(\psi_m i_{qs}\right) \tag{3.33}$$

where *s* represents Laplace operator. The equation of the mechanical motion remains unaltered. The torque is thus determined as the product of the permanent magnet flux and the stator *q*-axis current. The motor model is obtained for the reference frame fixed to the rotor while the permanent magnet flux is fixed to the rotor as well. It should be emphasised that the operation of a SPMSM is entirely dependent on the presence of an appropriate rotor position sensor, which at the same time acts as the rotor flux position sensor. This is a consequence of the fact that a synchronous machine must always run at synchronous speed, so that the rotor angular velocity and rotor flux angular velocity coincide.

3.3 ROTOR FLUX ORIENTED CONTROL OF SPMSM

The model of a SPMSM, derived in the previous sub-section, is given in the reference frame firmly attached to the rotor. Moreover, *d*-axis of the common reference coincides at all times with the axis along which permanent magnet flux acts. Spatial position of the permanent magnet flux therefore coincides at all times with the rotor position and is hence known, as the rotor position is measured. The model therefore describes a SPMSM in the rotor flux (permanent magnet flux) oriented reference frame. If the machine is considered as voltage-fed, the stator voltage equations have to be included in the design of the control system. However, if the machine is assumed to be fed from an ideal current source, the stator voltage equations can be omitted from consideration. This leads to an extremely simple control scheme as only torque equation (3.33) and equation of mechanical motion (3.25) remain. As flux in the machine is provided by magnets, there is no need to supply excitation current to the stator winding. The stator *d*-axis current reference is therefore set to zero for operation in the base speed region. As torque is directly proportional to stator *q*-axis current, the stator *q*-axis current reference is obtained as the output of the speed controller. One has to recall however that the stator *d-q* axis currents are fictitious currents and that they are correlated with the existing phase currents through coordinate transformation (3.17). It is therefore necessary to convert stator *d-q* axis current references into stator phase current references using (3.17), where the transformation angle θ is obtained from the position sensor.

The block diagram of a rotor flux oriented current fed SPMSM is shown in Fig. 3.3. The control structure contains only a speed control loop. The difference between the reference value of the rotor speed and actual rotor speed serves as the input signal to the speed controller, which is usually a PI controller with anti-windup. The output of this controller is the stator *q*-axis reference value.

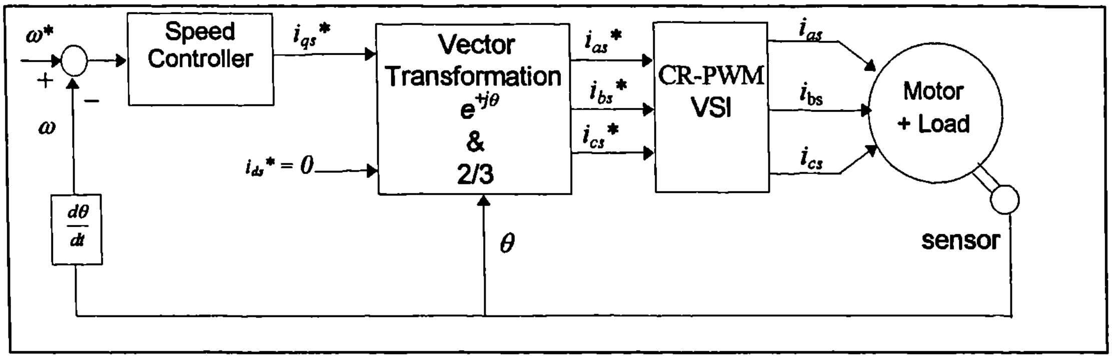


Figure 3.3: Rotor flux oriented control of a current-fed SPMSM

The two-axis *d-q* stator current references are transformed into three-phase stator phase current references. The correlation between *d-q* axis current references and stator phase current references is given from (3.17) with

$$\begin{aligned} i_a^* &= i_{ds}^* \cos(\theta) - i_{qs}^* \sin(\theta) \\ i_b^* &= i_{ds}^* \cos(\theta - \frac{2\pi}{3}) - i_{qs}^* \sin(\theta - \frac{2\pi}{3}) \\ i_c^* &= i_{ds}^* \cos(\theta - \frac{4\pi}{3}) - i_{qs}^* \sin(\theta - \frac{4\pi}{3}) \end{aligned} \tag{3.34}$$

The three-phase reference stator currents are the inputs into the current control loop. The three-phase measured stator currents are compared with the three-phase reference stator currents in the hysteresis current controller. The outputs of the current controllers are switching signals that form the input to the inverter which supplies the SPMSM. The stator currents are impressed by fast current control loops, as indicated in Fig. 3.3. Assuming that the current controlled PWM voltage source inverter behaves as an ideal current source, the actual phase currents supplied to the motor are equal to the phase current references.

The electromagnetic torque of the SPMSM with surface mounted magnets and with symmetrical three-phase stator winding can be given in an alternative form, as

$$T_e = \frac{3}{2} P \psi_m |i_s| \sin(\alpha_s - \theta) \tag{3.35}$$

where i_s and α_s denote magnitude and instantaneous spatial position of the stator current space vector, respectively. Stator current space vector with respect to the stationary phase '*as*' magnetic axis is defined as:

$$\underline{i}_s = i_s e^{j\alpha_s} = \left(i_{ds} + j\, i_{qs}\right) e^{-j\theta} \tag{3.36}$$

It follows from figure 3.2 that the spatial position of the stator current space vector relative to the fixed magnetic axis of the stator winding is α_s. Its spatial position relative to the direct axis of the rotor reference frame is $\alpha_s - \theta$. Electromagnetic torque, T_e, varies with the sine of the angle $\alpha_s - \theta$, the so-called torque angle (β). Since the flux produced by the permanent magnets has been assumed to be constant, the electromagnetic torque can be varied by changing the magnitude and the phase of the stator currents. Thus a constant torque is obtained if the quadrature-axis component of the stator current space vector is kept constant and the maximum torque per Ampere of stator current is obtained if the torque angle is 90^o. This corresponds to i_{ds}= 0. If an operation above the rated speed (i.e., operation in the field-weakening region) is not required, the torque angle is maintained at all times as equal to $\pm 90^o$(sign + applies to the motoring, while sign - applies to braking). This means that the stator *d*-axis current is regulated to zero and that the stator current space vector equals at all times stator *q*-axis current component. The permanent magnet flux and stator current space vector are therefore mutually perpendicular.

High performance variable speed SPMSM drive is realised with rotor flux oriented control using a current controlled voltage source inverter (block CR-PWM VSI in Fig. 3.3). Current control, by means of the hysteresis control scheme, is executed in the stationary reference frame. As mentioned earlier, the control structure of a current fed SPMSM is simpler than the one of the voltage fed machine. If the SPMSM is considered to be fed from a voltage source, current control is performed in the rotor reference frame, as illustrated in Fig. 3.4. This control structure results when stator voltage equations (3.31)-(3.32) are included.

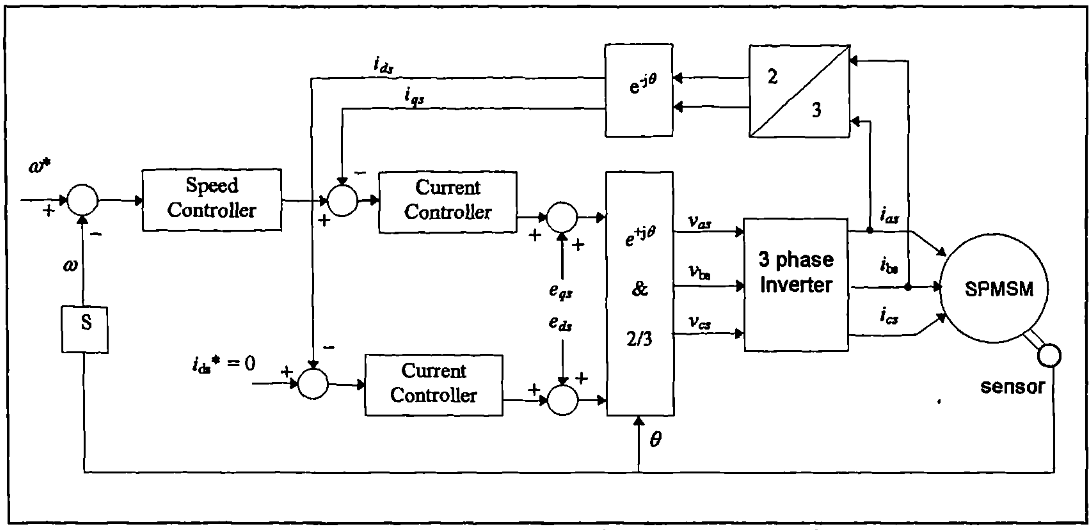


Figure 3.4: Rotor flux oriented voltage-fed SPMSM

In this case stator *d-q* axis voltages and currents are mutually coupled and a decoupling circuit would need to be inserted in the control scheme. Outputs of the decoupling circuit, denoted in Fig. 3.4 as e_{ds} and e_{qs}, are summed with outputs of current controllers in order to create the *d-q* axis voltage references. The control system is more complex because the measured stator currents have to be transformed into the rotor reference frame using co-ordinate transformation and since a decoupling circuit is in general required. Decoupling voltages e_{ds} and e_{qs} follow from (3.31) - (3.32) as $e_{ds} = -\omega L_s i_{qs}$ and $e_{qs} = \omega\left(L_s\, i_{ds} + \psi_m\right)$. As the control system structure of a current-fed SPMSM is considerably simpler then the one of a voltage-fed SPMSM, the current-fed SPMSM is selected for further work.

3.4 SELECTION OF THE SPEED CONTROL METHODS

The fuzzy logic speed control and PI speed control are chosen for the investigation. Fuzzy logic control (FLC) is chosen because of its empirically demonstrated robustness properties as well as the indications that it provides superior performance with respect to PI/PID control [Vas *et al*, 1994a, 1994b], [Heber, 1997], [Basehore, 1993], [Goureau and Ibaliden, 1996], [Sousa and Bose, 1995]. The design of an FLC does not require an accurate model of the plant [Bose, 1994], [Thomas and Armstrong-Hélouvry, 1995], [Self, 1990], [Smith, 1994]. This is greatly appreciated in motor drive applications where the motor and the mechanical load are described by a set of non-linear, differential equations or are partially unknown. An FLC usually demonstrates better results than the conventional PI controller, in terms of response time, overshoot and, particularly, robustness to parameter variations [Vas, 1998], [Vas and Stronach, 1996a], [Tana and Xu, 1994], [Le-Huy *et al*, 1995]. The latter is a worthwhile feature in motor drive applications where the mechanical load is widely varying during operation. PI control is chosen because it is one of the most commonly used methods in industry, and because it has plant perturbation robustness properties which can be mathematically analysed. In order to reduce the large overshoot because of the non-linear elements in the control system, the PI with integrator anti-windup is considered.

The fuzzy and PI speed controller are designed for the speed control loop based on the nominal plant parameters. Each controller is then applied in conjunction with the hysteresis current control scheme and numerical simulations are performed to analyse the response of the drive. The speed controller is configured as either a fuzzy controller or a PI controller. In order to obtain a detailed physical understanding of the behaviour of the fuzzy and PI speed controlled drive, the results from the continuous and discrete-time simulations are studied and compared. PI speed controller design is described in this chapter, while FL speed controller design is detailed in Chapter 4.

In this application, the time-domain specification is defined in terms of the unit-step response. Steady state performance is indicative of the accuracy of the system, while transient performance is indicative of the speed of response and the relative stability. Accuracy is the approximation with which a controlled variable is kept close to the reference value. The difference between the actual and the reference value of the controlled variable can be defined as dynamic or steady-state error.

The main purpose of design is to obtain the desired specification, in terms of speed of response, accuracy and stability. The speed of response depends on the system time constants, that is, the delays produced by the different elements of the system. The response time is the time needed to reach the new steady state. Typical specification includes: overshoot, delay time, rise time and settling time. In high performance drives, oscillations cannot be tolerated and it is desirable that the transient response be sufficiently fast and be sufficiently damped [Vas,1990].

3.5 PI SPEED CONTROLLER WITH INTEGRATOR ANTI-WINDUP

In the case of process control, the tuning techniques for classical. PI speed controller have existed since the 1940's, when Ziegler and Nichols introduced their step response and closed loop cycling methods [Ziegler and Nichols, 1942]. The theories of initial design and automatic tuning procedure have been explained in detail in [Aström and Hägglund ,1988]. Today, in drive applications, the classical PI controller is the main tool which is successfully utilised [Pilay, 1989], [Vas, 1990]. The mathematical model of a PI controller is given by

$$PI = K_p \left[e(t) + \frac{1}{T_i} \int e(t)\, dt \right] \tag{3.37}$$

where K_p is a proportional gain, $K_i = \frac{K_p}{T_i}$ is the integral gain, and $e(t)$ is the speed error.

The PI speed controller is designed based on Ziegler and Nichols method [Vas, 1998]. The controller parameters are then fine-tuned manually to improve the speed response. Any deficiencies in the formula can be supplemented by manual fine tuning based on operator's experience [Aström and Hägglund ,1988].

Many different types of non-linearity are found in practical control systems, as discussed in [Aström and Wittenmark,1990]. Basically, they can be divided into two classes, depending on whether they are inherent in the system or intentionally inserted into the system. Inherent non-linearities are unavoidable and these include saturation, dead zone, hysteresis and backlash. Generally, the presence of such non-linearity in high performance motor control system adversely affects system performance. For small input signals, the output of a saturation element is proportional to the input. For larger input signals, the output will not increase proportionally, and the output is constant at the maximum possible output value. This happens in the SPMSM control where the *q*-axis stator current will saturate at the set current limit, because the current capacity of the inverter is limited. Whenever control saturation happens, integration has to be stopped. A large step speed reference causes output of the speed controlled to saturate at $i_{qs\,max}$, while the integrator keeps integrating the error and the controller output signal keeps increasing. However, the output of the limiter is still at its maximum value ($i_{qs}{}^* = i_{qs\,max}$). When the speed error finally becomes zero and goes negative, the PI controller output still remains very high. A long time is normally required before the PI controller output reduces to desired level. This causes a large overshoot and undesirable oscillatory speed response. This unwanted behaviour is prevented from occurring by using an integrator anti-windup in the PI controller. The topic of anti-windup has been studied over a long period of time by many authors, and the most popular techniques are described in [Aström and Wittenmark, 1990], [Hanus *et al*, 1987], [Kothare *et al*, 1994], [Walgama *et al*, 1992]. A detail comparative study for different anti wind-up strategies is given by [Peng *et al*, 1996]. It is shown that the conditional integration technique (Fig. 3.5) is the most suitable anti-windup strategy for usual applications. On the other hand, the so-called incremental algorithm is a relatively simple method to incorporate in a digital controller. Figure 3.6

shows a typical discrete-time implementation of the incremental algorithm with sampling time T_s. This type of anti wind-up is used to investigate the effects of sampling time on the speed response for digital implementation. However, the response of the conditional integration is much better than the response of the incremental algorithm method. Therefore, the conditional method of PI speed controller anti wind-up is used in the comparative study of high performance SPMSM drives with different types of speed control.

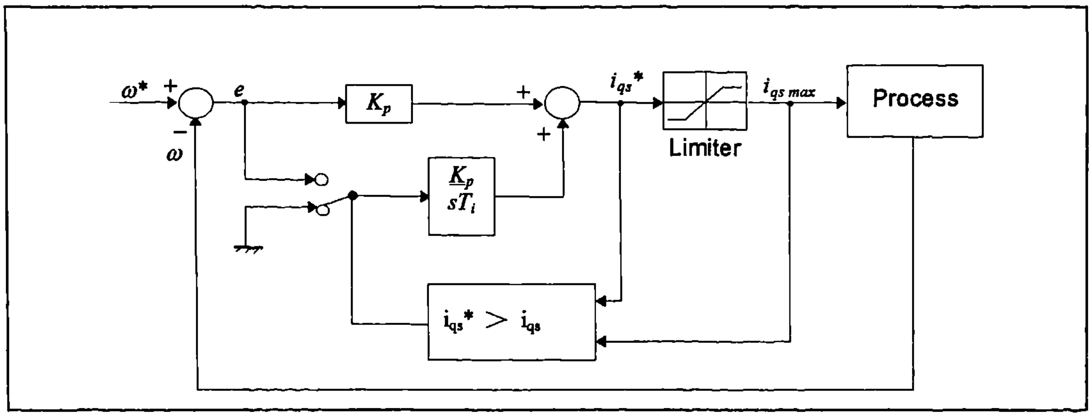


Figure 3.5: PI speed controller with conditional integration method.

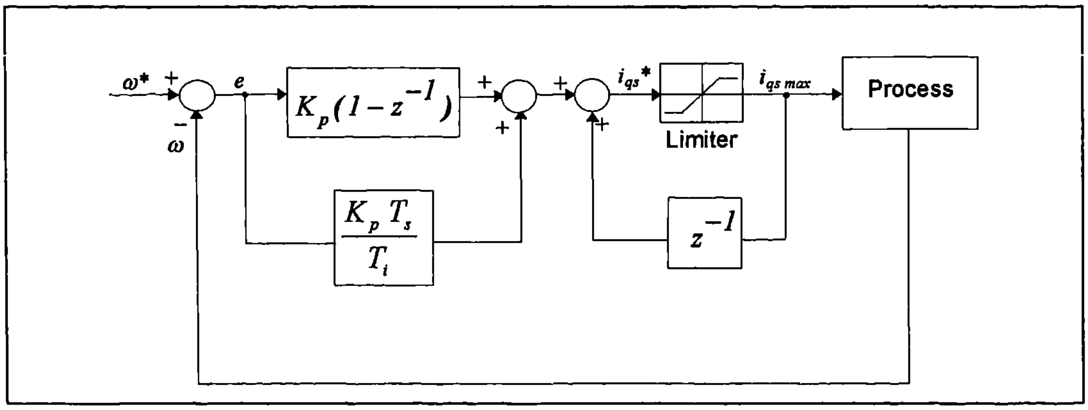


Figure 3.6: Incremental algorithm (discrete-time implementation with sampling time T_s) of PI integrator anti wind-up.

3.6 VOLTAGE SOURCE INVERTER AND HYSTERESIS CURRENT CONTROL

Three-ᵑhase inverters are used to supply voltages and currents of variable frequency and magnitude to the stator. Inverters are composed of semiconductor power switches and the input to the inverter is DC. Depending on the type of DC source supplying the inverter, they can be classified as voltage source inverters (VSI) or current source inverters (CSI). In practice, the DC source is usually a rectifier, typically of the single-phase or three-phase bridge configuration, with the so-called DC link connected between the rectifier and the inverter. The DC link is built from a simple inductive, capacitive, or inductive-capacitive low-pass filter. Since neither the voltage across a capacitor nor the current through an inductor can change instantaneously, a coupling capacitor is used at the DC input terminals of the VSI and an inductive link is employed in a CSI.

VSIs can be either voltage or current-controlled. In a voltage-controlled inverter, the frequency and magnitude of the fundamental of the output voltage are adjusted. Feed-forward voltage control is employed, since the inverter voltage is dependent only on the supply voltage and the states of the inverter switches, and therefore, accurately predictable. On the other hand, the current controlled VSIs require sensors of the output currents which provide the necessary control feedback. Normally, Hall effect current sensors are used to convert three phase stator currents into instantaneous output voltages, that are to be further processed by the controller. An overview of modern PWM voltage-controlled VSIs is given in [Trzynadlowski, 1996]. The principle of voltage control is predominantly used in speed control systems based on the constant Volts/Hertz principle, and is described in detail in [Trzynadlowski, 1994]. Current-controlled VSIs are typically used for high performance drives [Vas, 1998], [Trzynadlowski, 1994].

In general, voltage source inverters are classified into two groups, quasi-square-wave inverters with variable DC link voltage and PWM inverters [Vas,1992]. The output voltages of a square-wave inverter are variable amplitude, quasi square-wave voltages, where the amplitude of the voltages supplying the motor terminals is controlled by changing the firing angle of the rectifier. A detail explanation is given in [Vas, 1990]. For the PWM inverter, both the frequency and the amplitude of the motor voltages are controlled internally by the inverter using pulse width modulation, whereby a series of pulses is generated which have varying widths and constant amplitudes. The advantages of a PWM inverter-fed machine over the square-wave inverter-fed machine are reduced losses, reduced torque oscillations, and it does not require an extra voltage control circuit (e.g. control of rectifier or chopper) [Vas, 1990]. When a current-controlled PWM is used, a conventional PWM voltage-source inverter is equipped with fast current control loops. The basic structure of a three-phase VSI is shown in Fig. 3.7.

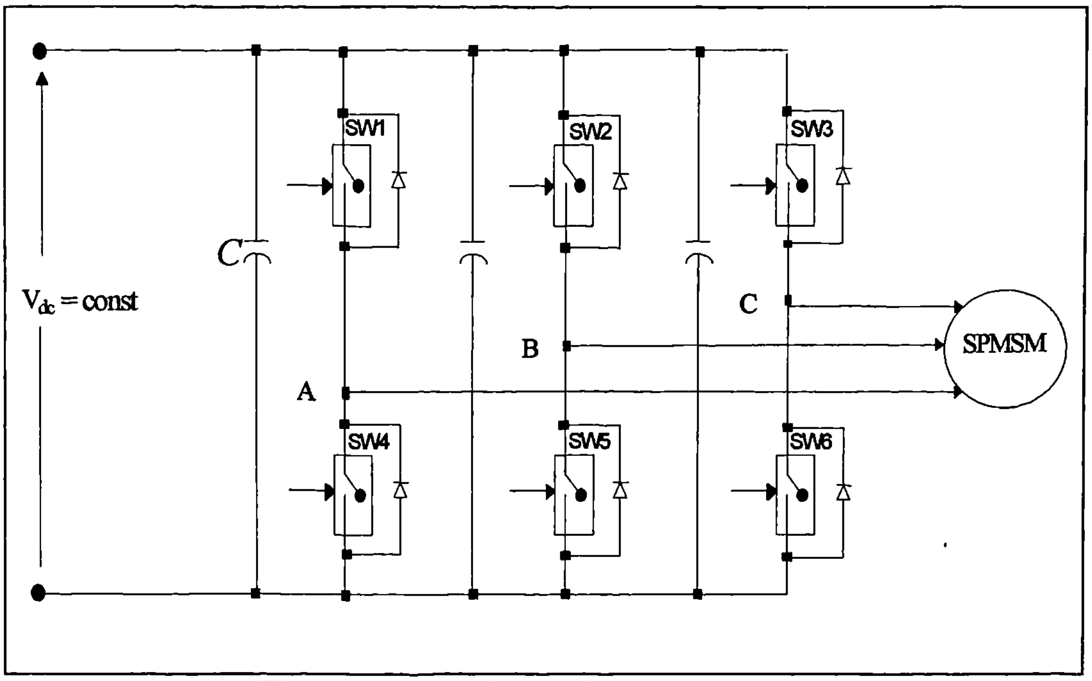


Figure 3.7: Three phase voltage source inverter.

In the simplified model of the VSI, the semiconductor power switches are replaced with ideal switches. The simulation of the inverter is based on the assumption of a constant DC link voltage and uses the concept of switching functions. Let the switching functions of the three inverter legs be denoted as *Sa, Sb, Sc*, respectively. Each switching function takes the value of 1 if the upper switch is on, while it equals -1 if lower switch is on. Leg voltages, v_A, v_B and v_C, are thus referred to the fictitious mid-point of the DC supply. Since only two combinations of states of the switches in each branch are allowed, a switching (logic) variable can be assigned to each phase of the inverter. The switching variables are defined as shown in Table 3.1.

Switching function	Logic states
Sa	-1 if SW1 is OFF and SW4 is ON 1 if SW1 is ON and SW4 is OFF
Sb	-1 if SW2 is OFF and SW5 is ON 1 if SW2 is ON and SW5 is OFF
Sc	-1 if SW3 is OFF and SW6 is ON 1 if SW3 is ON and SW6 is OFF

Table 3.1: Switching functions of VSI.

The instantaneous values of the line-to-line output voltages of the inverter are given with

$$v_{ab} = \frac{V_{dc}}{2}\left(Sa - Sb\right) \tag{3.38}$$

$$v_{bc} = \frac{V_{dc}}{2}\left(Sb - Sc\right) \tag{3.39}$$

$$v_{ca} = \frac{V_{dc}}{2}\left(Sc - Sa\right) \tag{3.40}$$

where V_{dc} is the DC supply voltage of the inverter. In balanced three-phase system, the line-to-neutral voltages can be calculated from the line-to-line voltages as follows:

$$v_a = \frac{1}{3}\left(v_{ab} - v_{ca}\right) \tag{3.41}$$

$$v_b = \frac{1}{3}\left(v_{bc} - v_{ab}\right) \tag{3.42}$$

$$v_c = \frac{1}{3}\left(v_{ca} - v_{bc}\right) \tag{3.43}$$

Hence, after substituting eqs. (3.38) through (3.40) in eqs. (3.41) through (3.43), the line-to-neutral voltages of the load are given by

$$v_a = \frac{1}{3}\frac{V_{dc}}{2}(2Sa - Sb - Sc) \tag{3.44}$$

$$v_b = \frac{1}{3}\frac{V_{dc}}{2}(2Sb - Sc - Sa) \tag{3.45}$$

$$v_c = \frac{1}{3}\frac{V_{dc}}{2}(2Sc - Sa - Sb) \tag{3.46}$$

Equations (3.44) through (3.46) represent the model of the voltage source inverter used in the simulation. These equations show the correlation between the switching functions that are produced by the hysteresis current controllers, DC link voltage (as an input to the three phase inverter) and phase voltages (as an input to the motor). However, the SPMSM model is in the rotating reference frame, while phase voltages are in the stationary reference frame. Therefore, phase voltages have to be transformed into the rotating reference frame based on transformation equation (3.16).

The basic structure of the SPMSM drive with current control in the stationary reference frame and with PI speed controller is shown in Fig. 3.3. Three independent hysteresis current controllers in the three phase *a, b, c* reference frame are applied in this scheme. In high-performance servo drives, hysteresis current controllers are used to ensure that the actual currents flowing into the motor are as close as possible to the current references. The power circuit that feeds the SPMSM is shown in Fig. 3.7. The six switches SW1- SW6 are used to control the three phase currents. The actual values of phase currents that flow into the motor are assumed to be measured using ideal Hall-effect current sensors.

The hysteresis current control scheme is explained with reference to Fig. 3.8, which shows the reference stator current i_a^*, and two other current curves, $i_a^* + \Delta i$ and $i_a^* - \Delta i$. Quantity Δi defines the hysteresis band. The hysteresis property allows the actual values of i_a to exceed or be less than the reference value by Δi. The logic is shown in Table 3.2 for one inverter leg and leg voltage v_A values are included.

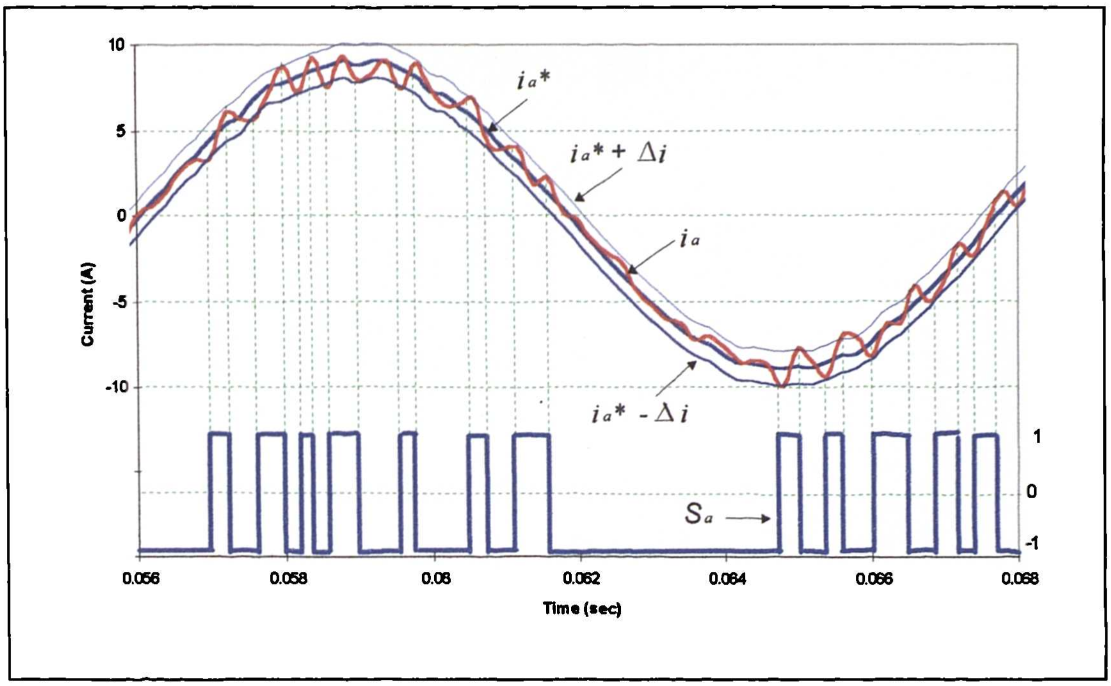


Figure 3.8: Reference stator current and hysteresis bands.

i_a*	i_a	SW1	SW4	v_A
≥ 0	$i_a \leq (i_a* - \Delta i)$	ON	OFF	$+V_{dc/2}$
≥ 0	$i_a \geq (i_a* + \Delta i)$	OFF	ON	$-V_{dc/2}$
< 0	$i_a \geq (i_a* + \Delta i)$	OFF	ON	$-V_{dc/2}$
< 0	$i_a \leq (i_a* - \Delta i)$	ON	OFF	$+V_{dc/2}$

Table 3.2: Principle of the hysteresis controller [Vas, 1992].

Similar logic applies to the other two inverter legs. Note that the complementary switching of the power devices is prohibited. Whenever SW1 is "ON," i_a increases using either the *b* or *c* phase as a return path. As soon as SW1 switches from an "ON" to an "OFF" position, and since the current through the machine winding cannot go to zero instantaneously, the freewheeling diode across its complementary transistor, in this case SW4, begins to conduct the phase "*a*" current. When this occurs, the voltage of leg "*a*" switches from $+V_{dc}/2$ to $-V_{dc}/2$ where the midpoint of the DC supply V_{dc} is taken as the reference. The opposite occurs when SW4 switches from

"ON" to "OFF". A similar procedure exists in the other legs. The reason that this is called a hysteresis controller is that the leg voltage switches to keep the phase current within the hysteresis band (Fig. 3.8). The phase currents are, therefore, approximately sinusoidal in steady state: the smaller the hysteresis band, the more closely do the phase currents represent sine wave. Small hysteresis bands, however, imply a high switching frequency, which is a practical limitation of the power device. Increased switching frequency also implies increased inverter losses. In the hysteresis controller the switching frequency depends on the value of the hysteresis band, and the actual switching frequency demanded from the inverter varies over one cycle of the output frequency. A trial-and-error procedure can be adopted to ensure that the inverter switching frequency is not exceeded.

3.7 SPEED SENSORS FOR HIGH PERFORMANCE DRIVES

For precise motion control, speed controllers usually rely on a sensor to accurately resolve and encode the position of a motor shaft, another rotating shaft, or linear movement of a machine. Besides the actual position value, servo motors also need the actual speed value for velocity control. A closed-loop control of a variable speed drive would employ a speed sensor to minimise the difference between the command and the actual speed. The actual speed can then be determined from position measurement or speed output of the sensor. A number of sensors can be used to provide a dedicated output which is proportional to the rotor speed, e.g. tachometer, encoder and resolver [Pillay, 1989]. The use of a speed sensor depends on the application, e.g. an encoder is normally used for high speed spindle applications since it can operate at speeds up to 20,000 rpm, and a resolver is used to provide absolute feedback. In cases where absolute position information is needed, e.g. for robotic applications, a resolver can be a good choice. On the other hand, rotary encoders can also provide either incremental or absolute output, and in fact, most absolute encoders now provide both output forms. In high performance variable speed SPMSM drives, the resolver is used to estimate the absolute position of the rotor to enable the motor to move from standstill. Because the position information is produced by amplitude modulation of a carrier signal, it usually needs to be converted to a digital format before it can be used by a motor controller. Resolver to digital converters (RDC) interpolate the resolver output signals and provide 10, 12, 14, or 16 bit resolution, depending upon the converter used. The maximum slew, or tracking rate, of an RDC is limited to 1/16th of the resolver reference frequency [Hagl *et al*,1996]. If 400 Hz of a reference is used, a maximum tracking limit of 1500 rpm can be obtained. The maximum speed, on the other hand, is limited by the oscillator frequency to approximately 1 MHz. With a 12-bit converter, this results in a maximum speed of 15,600 rpm. When the tracking rate of the converter is exceeded, the position value cannot be computed fast enough to keep up with the input and consequently the digital output will become totally unpredictable [Pool, 1995].

Many drives use a 2,000 cycle encoder with a 200 kHz frequency response capability [Hagl *et al*, 1996]. This allows the motor to run at 6,000 rpm without missing counts. However, for a digital drive system with 400 μs to 500 μs sampling interval, the one increment per sample speed would be 18.75 rpm. It is obvious that the system could run slower, but control accuracy will degrade from this point [Hagl *et al*, 1996]. Improved low speed operation is usually obtained using interpolation electronics, which use encoder signals that are sinusoidal, like a resolver. A 2048 cycle encoder paired with 512 times interpolation electronics, yields over one million cycles per revolution, where the minimum controllable speed would then be 0.0006 rpm. Hence, encoders are very good at low speed. Resolvers have similar capabilities, but not to the same degree. A 16 bit position resolution used with a 400 μs sample interval would allow a minimum speed control of 0.038 rpm. The selection of the proper resolver or rotary encoder, therefore, depends on the application. If a high control quality and a large speed control range are required, rotary encoders are the only meaningful solution.

In the simulation the speed sensor is assumed to be an ideal speed sensor. Therefore the output of the speed sensor exactly equals the electrical speed of rotation, and is obtained using the mechanical motion equation of the motor model

$$\frac{d\omega}{dt} = \frac{P}{J}\left[T_e - T_L\right] \qquad (3.47)$$

as

$$\omega = \int \frac{P}{J}\left[T_e - T_L\right]dt \qquad (3.48)$$

3.8 SIMULATION STUDY

Figure 3.9 illustrates the control scheme used in the simulation study. The simulations are performed using Simulink. The control algorithms are simulated using the model given in the block diagram form in Fig. 3.9. The data of the motor, that is used in the simulations, are given in Appendix A. Detailed Simulink models are included in Appendix B. The PWM voltage source inverter is controlled by means of three independent hysteresis current controllers. The inverter input DC voltage is set to 220 V, so that sufficient voltage reserve is provided for operation with good current control at rated speed and maximum torque. Stator *q* axis current is limited to 3.39 times rated stator current. The speed PI controller is provided with conditional integration anti-windup (Fig. 3.5). The specific test case examined is the response from rest to step speed command of 180rad/s, followed by the application of the rated load torque, and subsequent step reduction of speed reference to 0.9 of the rated speed.

3.8.1 Initial tuning of PI speed controller

There are many methods of tuning a PI controller. Three frequently used methods of tuning the PI controller are trial-and-error method, Ziegler and Nichols open loop step response (method 1) and Ziegler and Nichols closed-loop marginal-stability condition (method 2). Tuning of the PI controller will involve the setting of the proportional gain and the integral time constant (K_P and T_i).

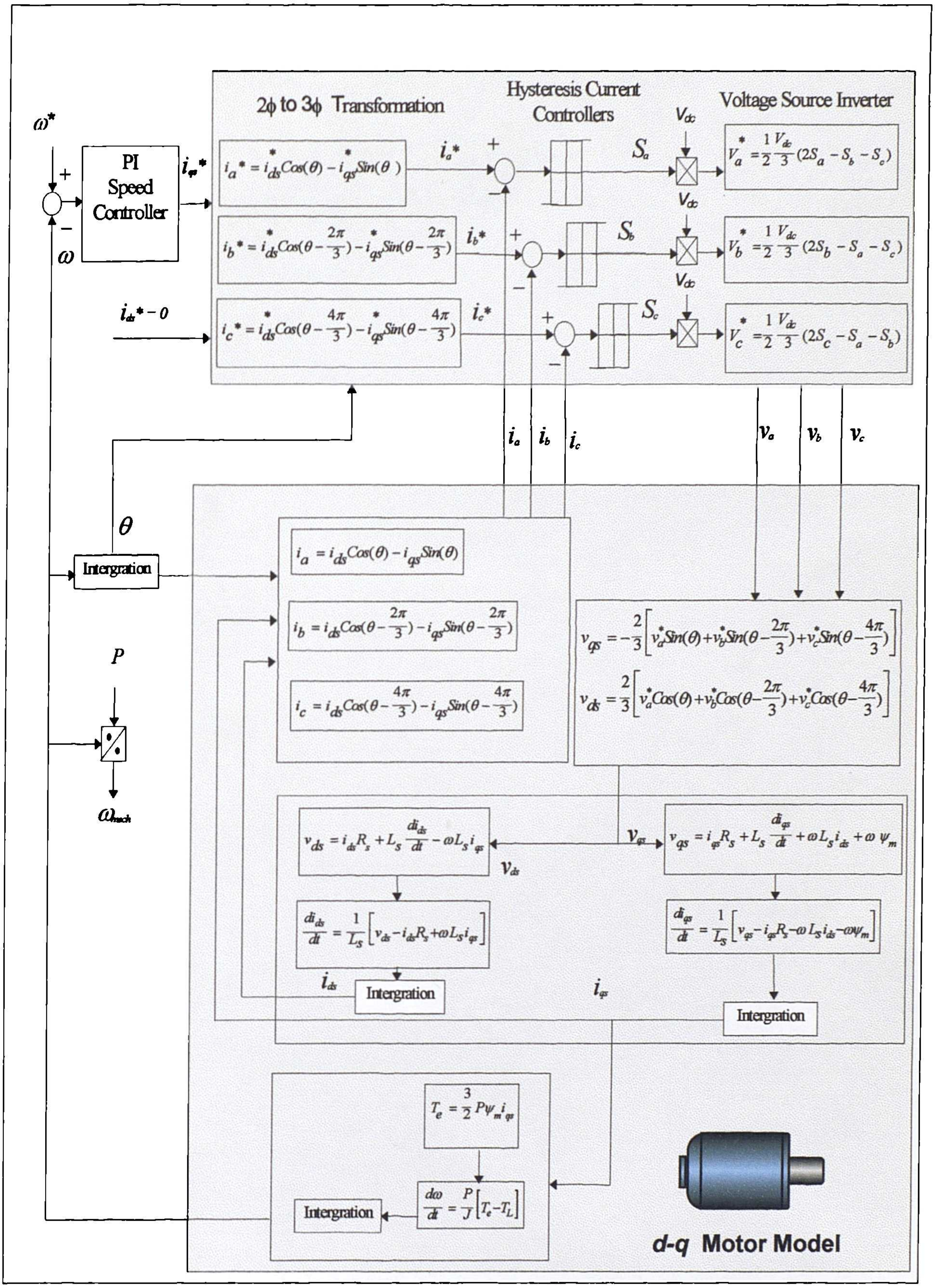

ω*
ω
PI Speed Controller
i_{qs}*
i_{ds}* - 0
2φ to 3φ Transformation
$i_a^* = i_{ds}^* Cos(\theta) - i_{qs}^* Sin(\theta)$
$i_b^* = i_{ds}^* Cos(\theta - \frac{2\pi}{3}) - i_{qs}^* Sin(\theta - \frac{2\pi}{3})$
$i_c^* = i_{ds}^* Cos(\theta - \frac{4\pi}{3}) - i_{qs}^* Sin(\theta - \frac{4\pi}{3})$
i_a*
i_b*
i_c*
Hysteresis Current Controllers
S_a
S_b
S_c
V_{dc}
Voltage Source Inverter
$V_a^* = \frac{1}{2}\frac{V_{dc}}{3}(2S_a - S_b - S_c)$
$V_b^* = \frac{1}{2}\frac{V_{dc}}{3}(2S_b - S_a - S_c)$
$V_c^* = \frac{1}{2}\frac{V_{dc}}{3}(2S_c - S_a - S_b)$
i_a
i_b
i_c
v_a
v_b
v_c
θ
Intergration
P
ω_{mech}
$i_a = i_{ds} Cos(\theta) - i_{qs} Sin(\theta)$
$i_b = i_{ds} Cos(\theta - \frac{2\pi}{3}) - i_{qs} Sin(\theta - \frac{2\pi}{3})$
$i_c = i_{ds} Cos(\theta - \frac{4\pi}{3}) - i_{qs} Sin(\theta - \frac{4\pi}{3})$
$v_{qs} = -\frac{2}{3}\left[v_a^* Sin(\theta) + v_b^* Sin(\theta - \frac{2\pi}{3}) + v_c^* Sin(\theta - \frac{4\pi}{3})\right]$
$v_{ds} = \frac{2}{3}\left[v_a^* Cos(\theta) + v_b^* Cos(\theta - \frac{2\pi}{3}) + v_c^* Cos(\theta - \frac{4\pi}{3})\right]$
$v_{ds} = i_{ds}R_s + L_s\frac{di_{ds}}{dt} - \omega L_s i_{qs}$
v_{ds}
v_{qs}
$v_{qs} = i_{qs}R_s + L_s\frac{di_{qs}}{dt} + \omega L_s i_{ds} + \omega\,\psi_m$
$\frac{di_{ds}}{dt} = \frac{1}{L_s}\left[v_{ds} - i_{ds}R_s + \omega L_s i_{qs}\right]$
$\frac{di_{qs}}{dt} = \frac{1}{L_s}\left[v_{qs} - i_{qs}R_s - \omega L_s i_{ds} - \omega\psi_m\right]$
i_{ds}
Intergration
i_{qs}
Intergration
$T_e = \frac{3}{2}P\psi_m i_{qs}$
$\frac{d\omega}{dt} = \frac{P}{J}\left[T_e - T_L\right]$
Intergration
d-q Motor Model

Figure 3.9: Rotor flux oriented control scheme with PI speed controller and hysteresis current control: Configuration used for simulation.

In the trial-and-error method the value of K_P is selected first in order to achieve the desired transient response while setting T_i to zero. Then T_i is adjusted in order to satisfy the steady-state error requirements. The advantage of this method is that a transfer function of the plant is not required, so that it can be applied when transfer function is not known. The initial tuning of PI speed controller is based on the closed-loop marginal-stability condition of Ziegler and Nichols. Here, the plant is assumed to be in the closed-loop mode with only proportional gain control. The parameters required in the controller in order to obtain optimum response are calculated from the critical gain (K_c) needed to just make the system oscillate with constant amplitude and the period of oscillation (P). The first trial is based on K_c = 27 which produced the time domain response that featured oscillations with increasing amplitude, as shown in Fig. 3.10a. On the other hand, the response oscillates with decreasing amplitude when K_c = 24, as shown in Fig. 3.10b. When K_c is set too low (K_c = 10), the response of the system has an overshoot and then reaches steady-state with a long settling time, as shown in Fig. 3.10c. The final tuning of the proportional gain is done by setting K_c = 25.5. In this case the system response oscillates with constant amplitude, as shown in Fig. 3.10d. The final values of PI parameters are calculated based on parameter setting of Ziegler-Nichols method 2 as K_P= 11.47, K_i = 4922.7 using correlation given in Table 3.3 [Driels, 1995].

Control	K_P	T_i	T_d
P	$0.5K_c$	∞	0
PI	$0.45K_c$	$0.833P$	0
PD	$0.6K_c$	$0.5P$	$0.125P$

Table 3.3: Ziegler-Nichols parameter settings, method 2.

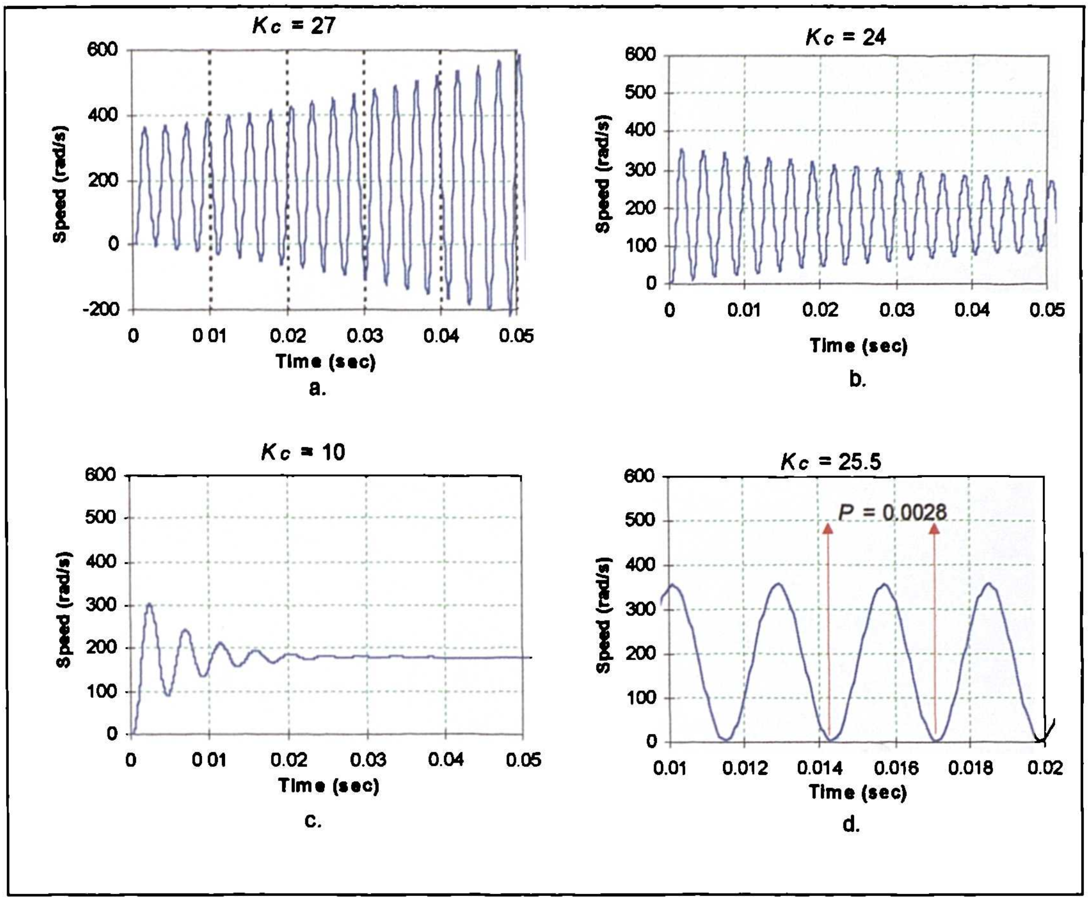


Figure 3.10: System output for different critical gain K_c: a) system response with K_c = 27; b) system response with K_c = 24; c) system response with K_c = 10; d) system response with final tuning K_c = 25.5.

Figure 3.11 (plot 1) shows the speed response produced from the set of PI parameters, which are estimated based on Ziegler and Nichols method 2. It shows that a 0.5% overshoot of speed response with approximately 18.5 ms settling time is obtained with initial PI parameters (K_P = 11.47 and T_i = 2.33ms). To achieve the desired performance, the PI speed controller can be further tuned, based on experience, by adjusting the proportional and integral gain. The proportional gain variation (K_P) will have the effect of changing the rise time, but the steady-state error cannot be eliminated without the integral part (Fig. 3.12a). The integral control (K_I) eliminates the steady-state error, but it affects the transient response. Figure 3.12b shows that the increase in the integral action gain (K_I) increases both the overshoot and the settling time, and eliminates the steady-state error. The influences of each of the two controller parameters, K_P and K_I, on the closed-loop system response are summarised in Table 3.4. The correlation is only approximate, because K_P and K_I are dependent on each other. In fact, changing one of these variables can change the influence of the other on the closed-loop system response.

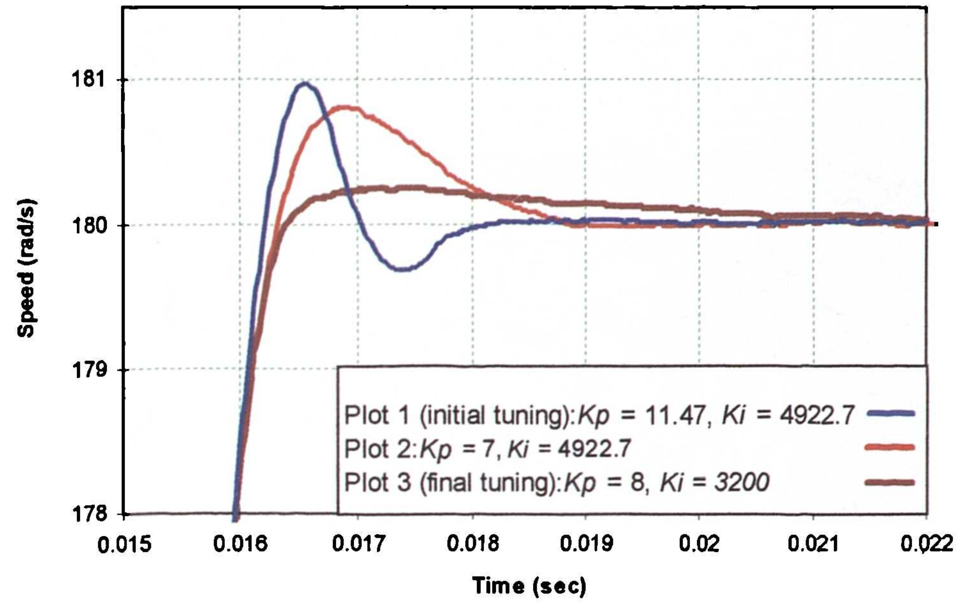


Figure 3.11: Fine-tuning of PI speed controller based on initial design of the PI controller using Ziegler and Nichols method 2.

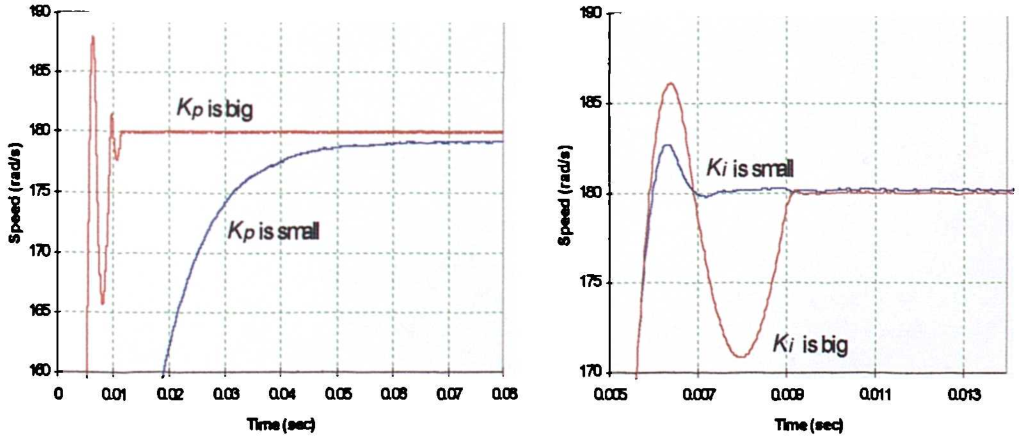


Figure 3.12a: The response with P controller.

Figure 3.12b: The response with PI controller.

Closed-loop response	Rise time	Overshoot	Settling time	Steady-state error
K_P-increase	Decrease	Increase	Small change	Decrease
K_I-increase	Decrease	Increase	Increase	Eliminate

Table 3.4: The correlation of K_P and K_I to rise time, overshoot, settling time and steady state error of speed response.

The speed response obtained from the initial tuning can be further improved with decreasing only the proportional gain, say to K_P = 7, while K_I was maintained at 4922.7 (Fig. 3.11, Plot 2). After several trial and error runs, the gains K_P = 8 and K_I= 3200 provided the response with overshoot less than 0.14% (0.24rad/s) and 21.5 ms settling time (Fig. 3.11, Plot 3).

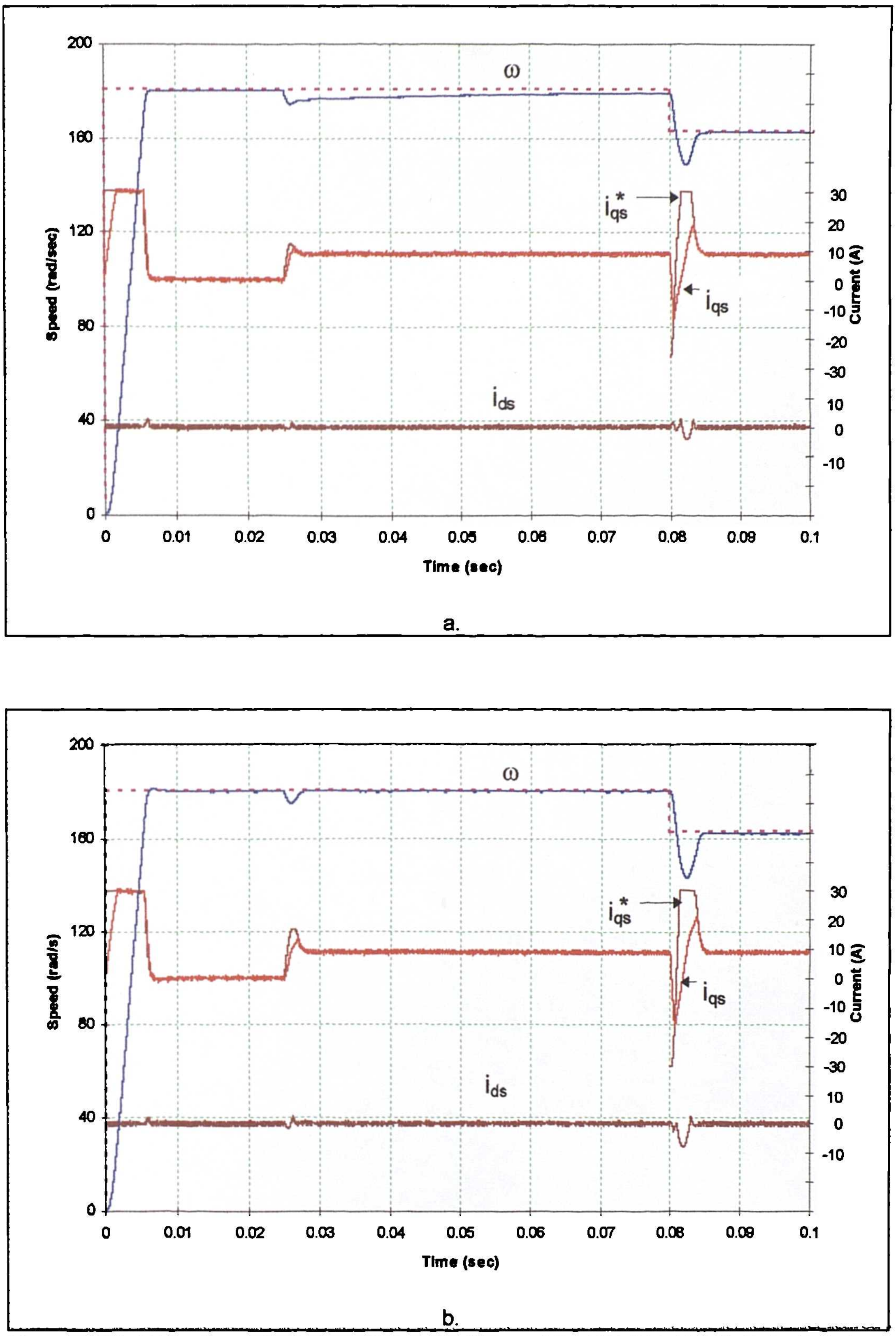
In high performance variable speed drive applications, overshoot of speed response is frequently not allowed. A slower speed response with zero overshoot can be obtained easily if proportional gain and integral gain are reduced. In this case, the designed controller may not satisfy the high performance requirement. However, by proper tuning of PI controller parameters, a short rise time without overshoot and short settling time can be obtained by decreasing the K_P and K_I of the PI speed controller as shown in Fig. 3.11.

Figure 3.13 is a typical sample of the simulation results, obtained using the model of Fig. 3.9. The machine is initially at rest and step speed command, equal to rated speed of 180 rad/s is applied at t = 0. Load torque during acceleration equals zero. Rated load is applied at t = 0.025 s. Finally, at t = 0.08 s, the speed command is reduced by 10% in a step-wise manner. Figure 3.13 illustrates speed response, stator *d-q* axis current references and actual values. After many simulation runs, parameters of PI speed controller, that yield short settling time with zero overshoot speed response (less than 0.1rad/s), are found as K_P = 2.22 and K_I= 111.1 (Fig. 3.13a). These values are smaller compared to the parameters obtained based on fine-tuned Ziegler-Nichols method 2. However, the load rejection is found to be very poor. The controller requires a long time to compensate the speed dip due to the load disturbance. Figure 3.13b shows that the speed response to load application can be improved by an increase in the gains of the PI controller (Fig. 3.13b is valid for K_P = 2.40, K_i = 1255.2). However, the response to the rated speed step command now exhibits an overshoot of 1.3 rad/s. As a conclusion, the same setting of PI controller parameters is not optimal for different operating conditions. Advanced control schemes, such as adaptive control, should be used to adapt controller to various operating condition. This will be further investigated in Chapter 6.

Three specific sets of PI controller gains will be used in further work (Chapter 4 and 5). The first two sets were obtained by manual tuning of the PI controller, using many simulation runs, and will be termed further on as "off-line optimised" PI controller parameters (it is recognised though that, in strictly control terms, optimisation has not been performed). The first set of PI controller gains applies to the zero overshoot design, while the second set yields 1.3 rad/s overshoot for step rated speed command. The third set of parameters is the one obtained using Ziegler-Nichols method 2. Table 3.5 summarises the values of PI controller gains used in subsequent work.

PI speed controller	K_p	K_i
Off-line optimised PI for zero overshoot design	2.22	111
Off-line optimised PI for 1.3 rad/s overshoot design	2.4	1255.2
PI tuned using Ziegler-Nichols method 2	11.47	4922.7

Table 3.5: PI controller gains used in subsequent chapters.



a.

b.

Figure 3.13: Rotor flux oriented control with PI speed controller and hysteresis current controller: a) speed response with K_p = *2.22*, K_i = *111.0*; b) speed response with K_p = *2.40*, K_i = *1255.2*, - - - - speed command.

3.8.2 Speed response for different selections of computation step and sampling time

Fast dynamic response in high performance drives requires a short computation step. It is very easy to realise simulations with small step about 40μs to 10μs. In a practical realisation, this cannot be implemented because of the limitations of the hardware. For real time control, the main problem is the one associated with computation time. In this section the minimum requirement of the computation time is determined through many simulation runs based on Fig. 3.13a (K_P = 2.22, K_I = 111). The comparison of the speed response is made between the speed responses for various computation steps in order to optimise the controller performance. These results can be used to estimate the computational speed requirement of the target processor, that is to be used in a practical realisation. Figures 3.14a and 3.14b show that the computation step should be smaller than 100μs. Otherwise an oscillatory response occurs.

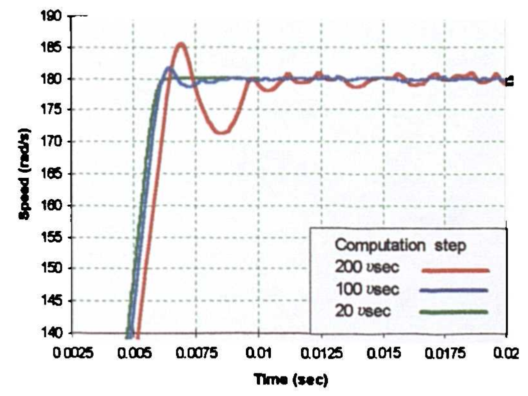


Figure 3.14a: Response to step speed command for different computation steps.

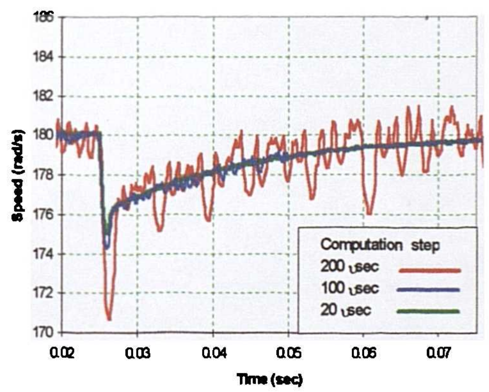


Figure 3.14b: Response to step rated torque application for different computation steps.

A digital speed controller (e.g. PI, PID) may work at a sampling rate of 5 to 10 kHz [Meshkat, 1996]. In discrete simulation, the PI controller can be represented based on discrete transformation. The continuous signal is sampled at the rate of the sampling frequency by an ideal sample and hold mechanism.

The overall performance of the system depends on the sampling frequency. The sampling frequency should be as high as possible to cater for the small time constants involved. The impact of sampling time can be investigated by running the simulation model with different sampling frequencies. The results can then be compared. PI controller with an integrator anti wind-up for discrete-time implementation, as shown in figure 3.6, is used in the discrete simulation study. Figures 3.15a and 3.15b show that the suitable sampling time for practical implementation for high performance SPMSM drives should be smaller than 100 μs. Otherwise the performance of the drive deteriorates, with the speed response becomes oscillatory especially during load disturbance. The simulation results have demonstrated that the desired speed response can be obtained if a computation and sampling time of 20 μs is used. This value would be used for the further simulation studies.

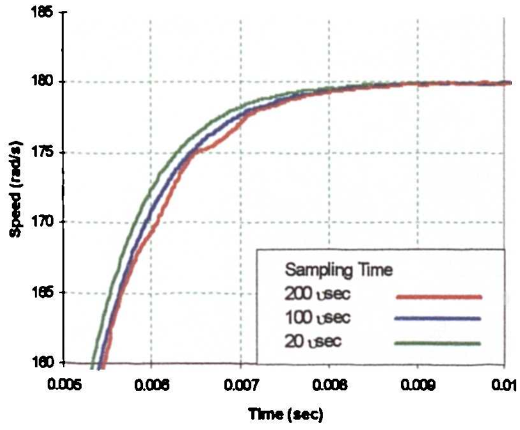


Figure 3.15a: Response to step speed command for different sampling time.

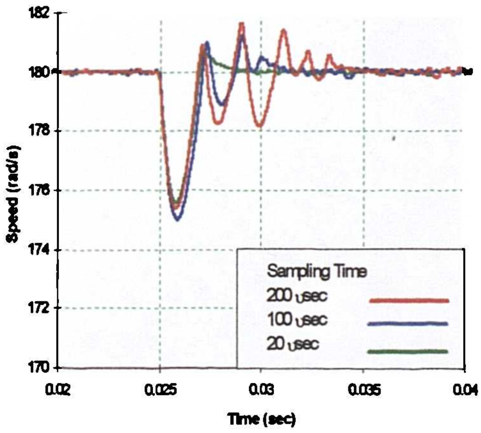


Figure 3.15b: Response to step rated torque application for different sampling time.

3.8.3 Speed response for different sizes of hysteresis band

Switching frequency of the voltage source inverter significantly influences the overall performance of the SPMSM drive. High switching frequency is required to produce precise and good speed or position control. However, the VSI with high switching frequency is expensive. Therefore, a sensible switching frequency of the inverter has to be determined by considering cost, required performance and application. In this section, the influence of switching frequency on speed response of the drive is investigated by many simulation runs. Speed controller parameters are K_p = 2.22 and K_i = 111.

In the case of hysteresis current control, the switching frequency increases nearly inversely proportionally to the decrease in hysteresis band. A small hysteresis band in the hysteresis current control scheme will cause a high switching frequency. Simulation results have shown that a good performance of the drive system can be obtained when the hysteresis band is smaller than +/-0.8 A or 8% of the rated current, Fig. 3.16.

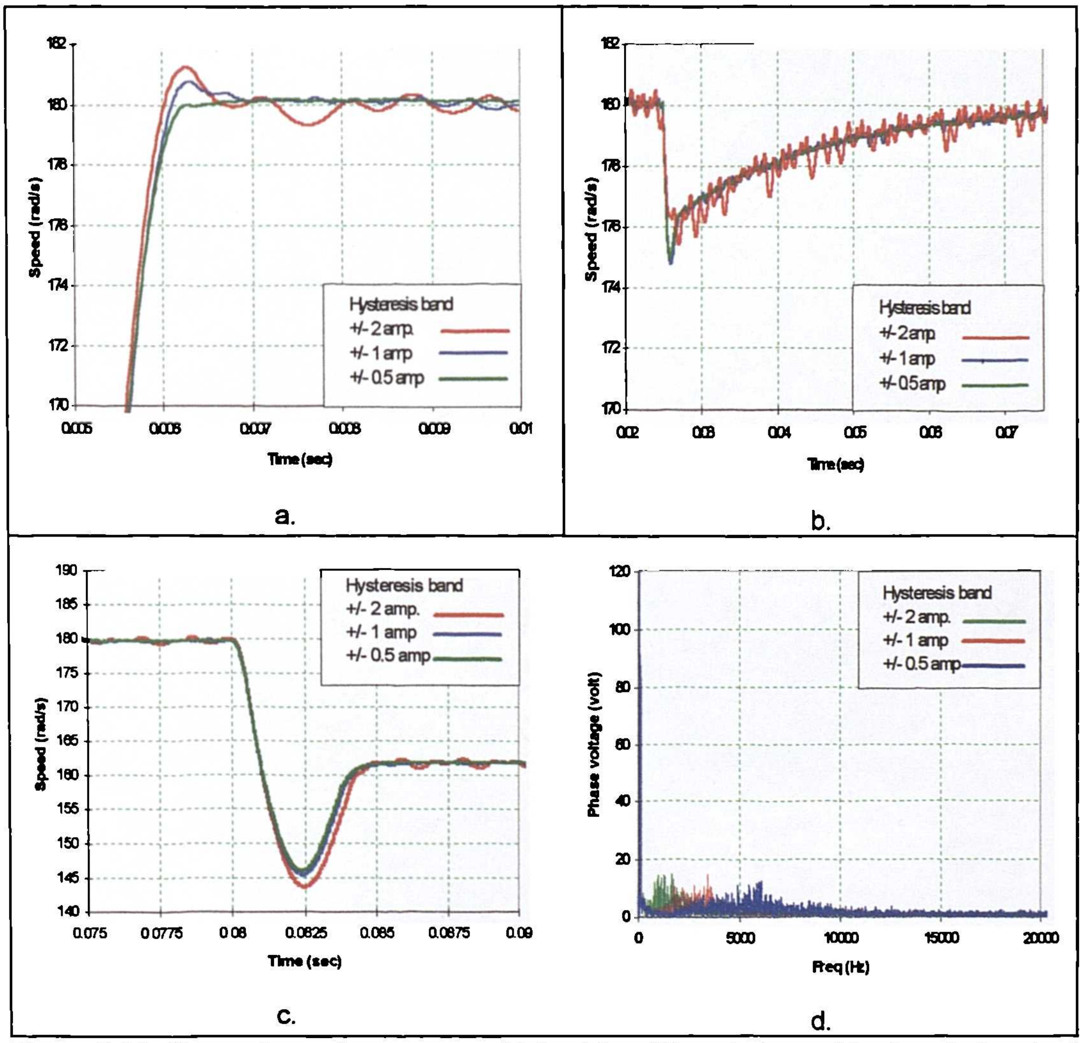


Figure 3.16: Comparison of responses obtained for different sizes of hysteresis band: a) response to rated step speed command; b) response to step rated torque application ; c) response to change of speed command from rated to 0.9 times rated; d) spectrum of the phase voltage of a VSI in the steady state operation at rated speed, rated torque and 220 DC link voltage.

The switching frequency remains below 10 kHz, as indicated by Fig. 3.16d. However, the speed response to speed command and step load application is oscillatory if the hysteresis band is larger than 10% of rated current, as shown in Fig 3.16a,b. The restoration time after the step load torque change is not much different for different sizes of hysteresis band, Fig 3.16b. The undershoot of speed response due to small step change in speed command is found to be large when the hysteresis band is greater than 10% of the rated current. A ±0.5 A hysteresis band of the current controller is selected for the further investigation because it gives good response in terms of overshoot and response time, with a reasonable switching frequency.

3.8.4 Speed response for 10% change of DC link voltage

Correct and constant inverter input DC voltage should be used to provide sufficient voltage reserve for operation with good current control at rated speed and maximum torque. Figure 3.17 (using K_p= 2.22 and K_i= 111) shows that when the DC link voltage is reduced by 10% below the rated value, the speed response overshoots and a longer settling time is needed to achieve steady state (Fig. 3.17a). The maximum speed dip due to step load application is increased compared to the nominal case (Fig. 3.17b). The undershoot in speed response due to step reduction in speed command is larger than with the rated DC link voltage (Fig. 3.17c). On the other hand, the overall performance of the drive is improved when the DC link voltage is increased to 110% of the rated value. The speed undershoot that follows the step decrease of speed command is smaller and the restoration time is shorter than in the nominal case. The maximum speed dip and the restoration time due to the step load torque change are smaller than in the nominal case. However, the settling time of a large step command application is better with nominal case (220 V) compared to +/- 10% changes of the DC link voltage. For further simulation study the DC link voltage is set at 220 V.

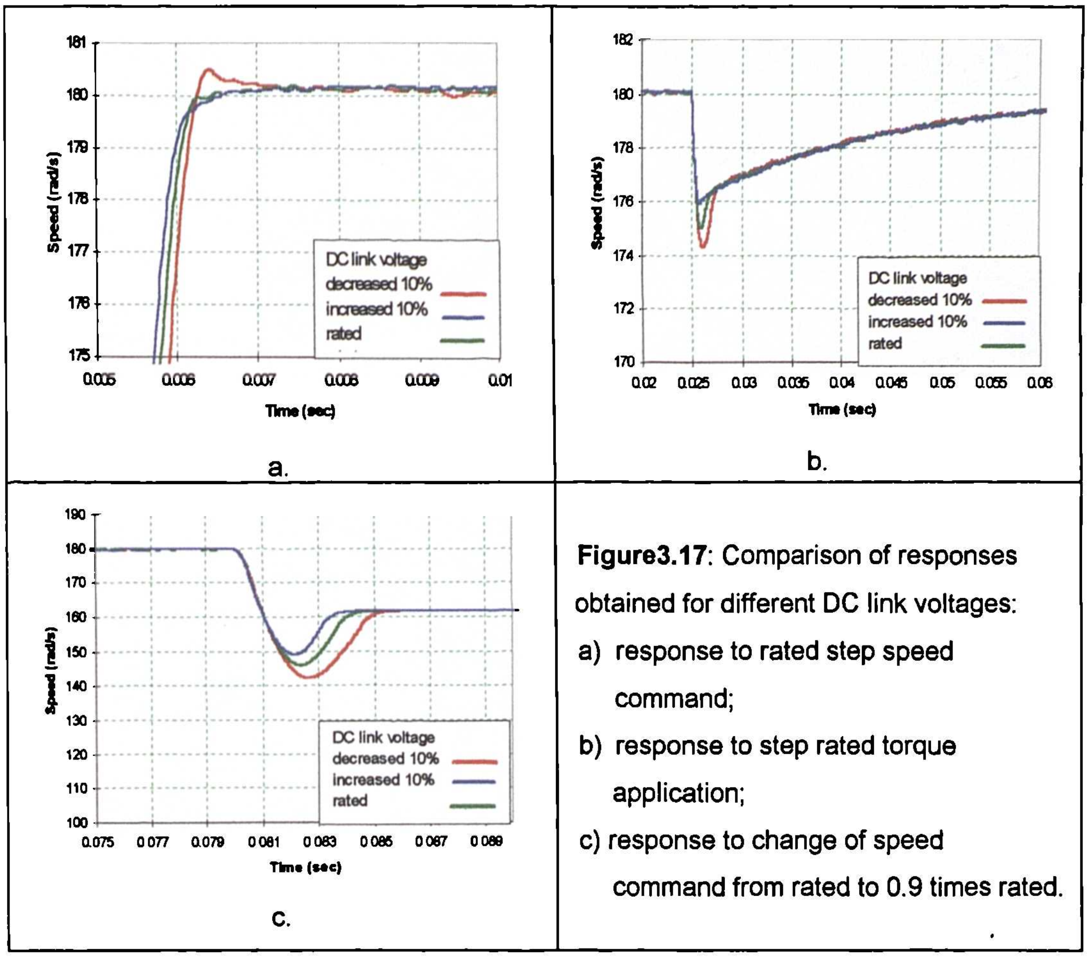


Figure3.17: Comparison of responses obtained for different DC link voltages:

a) response to rated step speed command;

b) response to step rated torque application;

c) response to change of speed command from rated to 0.9 times rated.

3.9 SUMMARY

This chapter presented the modelling and simulation of SPMSM drives. The modelling of the SPMSM is based on a three phase synchronous machine model in rotor flux oriented reference frame. The principle of PWM, based on hysteresis current controller and voltage source inverter, is explained and modelled. The implementation of the speed control incorporating anti-windup is also discussed. Speed sensors are classified based on their capabilities to operate in different speed ranges. The initial tuning of PI controller parameters is achieved based on Ziegler and Nichols method 2. Further fine tuning is done using a trial and error method. The correlation between PI parameters and speed response is investigated. The speed responses for the different sets of PI parameters are also presented. The continuous and discrete time simulation study is carried out and several factors are considered, such as initial PI controller parameter tuning, computation and sampling time, change of hysteresis band and DC link voltage. Three criteria are considered in the evaluation in order to finalise the system parameters used for further simulations. They are the quality of speed response, total simulation time and practical realisation requirements. Quality of the speed response is defined as a small overshoot/undershoot with short settling/restoration time. The total simulation time increases inversely proportionally to decrease in the computation and sampling time. Practical realisation is dependent on the capability of the inverter and target hardware. The influences of these factors are evaluated based on simulation results.

CHAPTER 4

DESIGN OF FUZZY LOGIC SPEED CONTROLLER

4.1 INTRODUCTION

The tuning of a FL controller is an important issue, especially when good performance with simple implementation is required. At present, there are no specific procedures and methods that enable selection and adjustment of the controller parameters for the desired performance. Tuning a FL controller is therefore more difficult than tuning a conventional PI controller. Furthermore, a FL controller has more than two parameters which can be adjusted and that can lead to a different behaviour of the controlled process. The controller behaviour is often very difficult to understand and predict when the controller parameters are changed. In general, some parameters affect the global controller performance (scaling factors and membership functions), while other parameters affect local performance (for example, fuzzy rules). Tuning of scaling factors is very important because a change of scaling factors causes firing of different rules. Usually, research regarding FL controller design either directly applies a set of scaling factors without explanation regarding the method used for their selection or simply gives some rough ideas to guide the choice of the scaling factors to a specific problem (for example, "transient behaviour is improved by reducing the controller input scaling factors, unfortunately this leads to a deterioration in steady state accuracy" or "increasing the controller input scaling factors improves steady state accuracy and vice versa" [Daley and Gill, 1987], [Ghwanmeh, 1996], [Heber *et al*, 1997], [Daley and Gill, 1985]. The adopted methods for scaling factor selection are essentially empirical and based on a trial-and-error approach.

Recently, some researchers have proposed different techniques to overcome this problem. For example, scaling factors can be correlated with gains of the conventional PI controller [Li, 1997], [Tang and Mulholland, 1987]. In [Qin and Borders, 1994] the non-linear gains of the FL controller are classified based on multi-region operating conditions (when the process is operating in the low-gain region, aggressive control action is demanded, while low control action should be used when process is in a high-gain region). Two-dimensional relay approach can be used to determine the FL control action [Ying, 1993]: when the number of input fuzzy sets is increased, the resolution of the global multilevel relay is improved. However, in some cases the methods used to overcome the difficulty of the FL controller tuning become meaningless, because certain parameters have to be re-adjusted until the desired performance is obtained [Tang and Mulholland, 1987], [Ying, 1993], [Palm, 1995], [Li, 1997]. Once again, this will involve a trial-and-error procedure to chose the best parameter values. In most fuzzy controller applications a set of constant scaling factors is used [Lee, 1990a, 1990b], [Birou *et al*, 1996], [Donescu *et al*, 1996], [Monti and Scaglia, 1997], [Heber *et al*, 1997], [Fodor *et al*, 1997], [Zhen and Xu, 1996], [Linkens and Abbod, 1991].

In this chapter, the behaviour and the performance of the FL speed controller are investigated for different methods to calculate initial values for the scaling factors. The main objective is to minimise the difficulty associated with a trial-and-error approach to scaling factor tuning. The initial values for scaling factors are calculated based on the correlation between a PI controller and a FL controller. Since, in linear control, the design and tuning of a PI controller can be accomplished by many systematic methods that yield reasonable performance [Aström and Hàgglund, 1988], this approach can be used to fine tune the FL controller in order to obtain desired performance. Three different FL speed controllers are considered in this Chapter:

A. The "Standard CPFL" speed controller is designed based on the following procedure:

   i) The controller parameters such as fuzzy rules, membership functions, fuzzy inference and defuzzification technique are chosen based on the most common selections reported in existing literature (Section 4.4.1).

   ii) The initial values of scaling factors can be calculated based on the correlation between FL controller and PI controller. In this case, three different sets of PI controller parameters are used to calculate the initial scaling factors (Section 4.3.5.1):

      1) Off-line optimised PI speed controller (Section 3.8.1): zero overshoot design and 1.3 rad/s overshoot design.

      2) Constant parameter PI speed controller tuned using Ziegler-Nichols method 2 (Section 3.8.1).

B. "Off-line optimised CPFL" speed controller where the membership functions, fuzzy rules and scaling factors are tuned based on trial-and-error approach until desired response is obtained (Section 4.4.2). The desired response is defined as a response which has short settling time and zero overshoot.

C. "Design case CPFL" speed controller is based on the controller designed in part B. The only difference is that the scaling factors are calculated based on the "design case" study described in Section 4.3.5.1.1.

4.2 THE ANALOGY OF FUZZY PI AND LINEAR PI CONTROLLER

The correlation between FL controller and PI controller is investigated based on possible structures of fuzzy controllers and relates those structures to non-fuzzy control theory. Such relations will provide a solid framework for analytically solving many important but previously difficult problems in fuzzy control, such as tuning of scaling factors, by utilising well-developed non-fuzzy control techniques. Early work in this area [Ying *et al*, 1990], [Ying, 1993] showed that the simplest possible non-linear fuzzy controller with two members for the input fuzzy sets, "error" and "change of error", was equivalent to a conventional linear PI controller when a linear defuzzification algorithm was used or to a non-linear PI controller when a non-linear defuzzification algorithm was used.

Firstly, the general structure of the FL controller as a PI like- and PD like-controller is established (Sections 4.2.1, 4.2.2). Approaches to implementation of the scaling factors in a FL controller are classified. Next, the relationships between the gains of a PI controller and scaling factors of FL controller are presented in mathematical form, using the conventional PI control law. A comparative study regarding different initial scaling factor calculation methods is finally made.

4.2.1 Framework for PD like fuzzy controller design

The standard conventional continuous-time PD control law is described by

$$u(t) = K_p\left[e(t) + T_d \frac{de(t)}{dt}\right] \qquad (4.1)$$

where K_p and T_d are the controller gain and derivative action time of the controller respectively, and $e(t)$ is the error signal defined by $e(t) = \omega^* - \omega$, with ω^* being the set-point (speed command) and ω the actual rotor speed. Equation 4.1 can be represented in Laplace domain as follows:

$$U(s) = K_p\left[1 + T_d s\right]E(s) \qquad (4.2)$$

This equation can be discretised by using the popular backward form because it uses the current and previous values of the information which is normally available. Let T_s be the sampling period of the continuous-time signals in the digital control system. By applying the standard mapping, s is replaced by $(z-1)/(T_s z)$, and one can convert the continuous-time system to its discrete-time equivalent in the complex z-frequency domain. Under the mapping, we have

$$T_d s \;\Rightarrow\; T_d \frac{(z-1)}{(T_s\, z)} \tag{4.3}$$

so that the correlation between the Laplace transforms of u and e, denoted as $U(s)$ and $E(s)$, is given now in terms of $U(z)$ and $E(z)$, as follows:

$$U(s) = \mathrm{K_p}\left[1 + T_d s\right]E(s) \;\Rightarrow\; U(z) = \mathrm{K_p}\left[1 + \frac{T_d}{T_s}(1 - z^{-1})\right]E(z) \tag{4.4}$$

where $K_d = \frac{T_d}{T_s} K_p$. Consequently, equation (4.4) can be represented in discrete time form as:

$$u(nT_s) = K_p\left[e(nT_s) + \frac{T_d}{T_s}\left[e(nT_s) - e(nT_s - T_s)\right]\right] \tag{4.5}$$

where

$u(nT_s)$ is the output signal

$\Delta e(nT_s) = e(nT_s) - e(nT_s - T_s)$ is the differential input

$e(nT_s)$ is the error

An implementation of the controller like a discrete PD controller is shown in Fig. 4.1a.

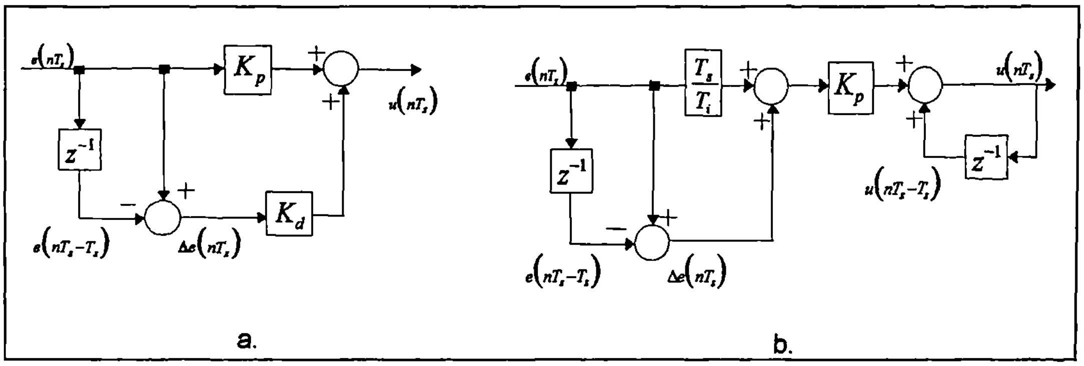


Figure 4.1: Discrete form of: a) PD controller; b) PI controller.

4.2.2 Framework for PI like fuzzy controller design

The standard transfer function of conventional continuous-time PI control law is described by

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int e(t)dt\right] \tag{4.6}$$

where $e(t)$, $u(t)$, K_p and T_i are the input signal, output command, controller proportional gain and integral time constant, respectively. By applying the same standard mapping as in PD controller case, one gets

$$\frac{1}{T_i s} \Rightarrow \frac{T_s z}{T_i(z-1)}\ , \tag{4.7}$$

so that the relation between the Laplace transforms of *u* and *e* transfers into

$$U(s) = K_p\left(1+\frac{1}{T_i s}\right)E(s), \Rightarrow U(z) = K_p\left(1+\frac{T_s}{T_i}\frac{1}{\left(1-z^{-1}\right)}\right)E(z) \tag{4.8}$$

where $K_i = \frac{T_s}{T_i}K_p$. Consequently, equation (4.8) can be represented as follows:

$$(1-z^{-1})U(z) = K_p\left[\left(1-z^{-1}\right)+\frac{T_s}{T_i}\right]E(z) \tag{4.9}$$

$$u(nT_s) - u(nT_s - T_s) = K_p\left[e(nT_s) - e(nT_s - T_s) + \frac{T_s}{T_i}e(nT_s)\right] \tag{4.10}$$

where

$\Delta u(nT_s) = u(nT_s) - u(nT_s - T_s)$ is the incremental control output

$\Delta e(nT_s) = e(nT_s) - e(nT_s - T_s)$ is the differential input

$e(nT_s)$ is the error.

An implementation of the controller like a discrete PI controller is shown in Fig. 4.1b.

4.3 SELECTION OF THE FL CONTROLLER PARAMETERS IN INITIAL DESIGN

Trial-and-error is the most popular technique to design a FL controller. An excellent performance can be obtained if the FL controller's parameters are tuned properly, however in many cases the tuning procedure is very tedious. There are three main parameters which have a major influence on the overall performance of a FL controller. These are the scaling factors, fuzzy rules and membership functions. In this chapter, a FL speed controller is designed with fixed scaling factors, a fixed number of rules and fixed membership functions.

4.3.1 Fuzzy control rules

There are two main types of fuzzy rules. The first type was reported in early applications of fuzzy logic control [Mamdani, 1974], while the second type was proposed by Tagaki and Sugeno [1983]. Mamdani's type of rule has been used in this work because it can provide a natural framework to convert the human knowledge into fuzzy "IF....THEN" rules.

The knowledge about a fuzzy system can be translated into a set of fuzzy linguistic rules which are in the form of:

IF (*conditions are satisfied*) THEN (*consequences can be inferred*)

Each fuzzy rule is a conditional statement in which the antecedent might be in the form of error (e) and the change of error (ce). The consequent may take the form of the output (u), or the incremental change of the output (Δu). For example, the fuzzy control rules using 'e' and 'ce' as inputs and 'u' as an output can be expressed as:

Rule 1: IF E is A_1 AND CE is B_1 THEN U is C_1

Rule 2: IF E is A_2 AND CE is B_2 THEN U is C_2

Rule n: IF E is A_n AND CE is B_n THEN U is C_n

where E, CE and U are linguistic variables representing the physical variables e, ce and u, respectively, while A_n, B_n and C_n are linguistic values (labels) defined over the universe of discourse of E, CE and U.

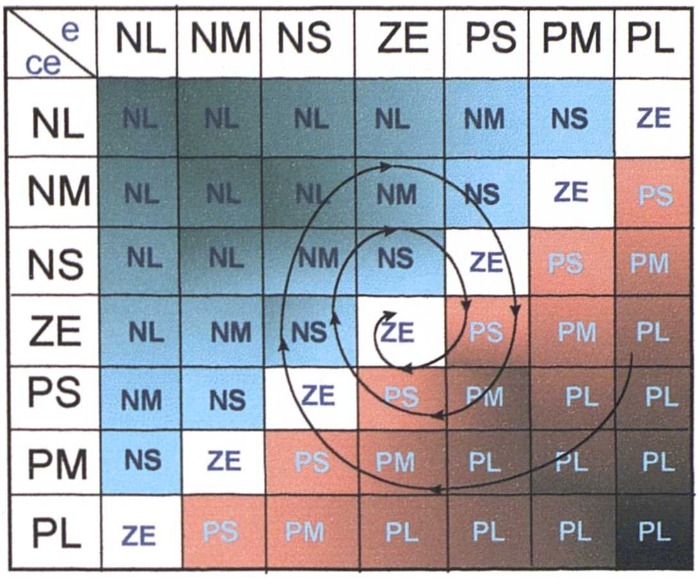
Takagi and Sugeno [1983] proposed four guidelines on how to derive fuzzy rules:

(i) Expert experience and control engineering knowledge
(ii) Operator's control action approach
(iii) Fuzzy model approach
(iv) Learning approach

Vas et al, [1997] discuss five main methods of developing the rule base: method (i) to method (iv) as mentioned above plus artificial neural networks.

In most of the control applications surveyed, the designer's qualitative "feel" is obviously used as a basis for the construction of the control algorithm. However, the best algorithm is frequently obtained by the iterative process of trying the algorithm, observing the results and then modifying the control rules accordingly. The fixed rule base of a FL controller is widely used in many applications [Self, 1990], [Bose, 1994]. It has been shown that a fixed rule based FLC can produce a satisfactory performance compared to the conventional PI or PID speed controller [Self, 1990], [Bose, 1994]. The experimental results show that it is possible to build a real-time navigation system with only 13 fuzzy rules, [Guo and Peters, 1996]. Nevertheless, many researchers have a strong interest in developing a self-learning controller based on a fuzzy rule base or a combination of fuzzy logic and neural networks. The main objective of this technique is to provide self-learning properties using for example neural networks, leading to the reduction of the number of rules generated, as well as to elimination of incompleteness in the rule base and the simplification of the tuning procedure (Stonach *et al,* [1997] in AC drive applications and Ghwanmeh *et al,* [1996] in process control).

The initial rule base of the fuzzy controller considered is principally based on prototype rules [King and Mamdani, 1977], [Zheng, 1992], [Liaw and Cheng, 1995], [Birou *et al,* 1996], [Ta-Cao and Huy, 1996], [Le-Huy *et al,* 1995], [Lee, 1990a], [Bird *et al,* 1997]. In this form, the rule justification is done by referring to a closed system trajectory in a phase plane. The principle of global rule modification in symmetry and monotonicity is also employed. In this work, this will be called a standard fuzzy rule base (Table 4.1).



e ce	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NL	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	NM	NS	ZE	PS	PM
ZE	NL	NM	NS	ZE	PS	PM	PL
PS	NM	NS	ZE	PS	PM	PL	PL
PM	NS	ZE	PS	PM	PL	PL	PL
PL	ZE	PS	PM	PL	PL	PL	PL

Table 4.1: A standard rule table.

4.3.2 Membership functions

A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The function itself can be an arbitrary curve whose shape we can define as a function that suits us from the point of view of simplicity, convenience, speed and efficiency. The Fuzzy Logic Toolbox™ includes 11 built-in membership function types, such as piecewise linear function, the gaussian distribution function, the sigmoid curve, and cubic polynomial curves.

A piecewise linear membership function will be used rather than gaussian or continuous function. Such membership functions are simple to implement and are computationally efficient. Membership functions will be fully overlapping; that is, at any given value of the variable, the total membership sums to one. Another benefit of these choices for membership function is that they allow the interpretation of the system as a simple interpolation between points in the input space, and the system would interpolate smoothly between these points to determine the complete output surface [Higgins and Goodman, 1994].

The membership partitions for error, change of error and change of output are usually symmetric from -1 to 1. The number of membership functions for each variable can vary, depending on the resolution required for that variable. Generally, use of more membership functions offers more degrees of freedom to the functional relationship of the controller, but it requires more effort in the implementation. It is shown in [Ying *et al*, 1990] that a conventional PID controller can be reproduced using a fuzzy logic controller with just two membership functions for each variable. Therefore, the effect of using more than two membership functions is merely adding non-linearity. All the membership functions used here are of the standard triangular type. The selection of a number of fuzzy sets is rather arbitrary, but 5 to 9 sets per universe is quite common in fuzzy control. The limitation of the sampling frequency in the on-line implementation of vector controlled drives can be overcome by minimising the processing time. In the research the triangular shape membership functions have been used, with the number of membership functions limited to seven.

4.3.3 Common fuzzy inference

A fuzzy relation can be used to describe each fuzzy control rule in the rule-base of the controller, expressed as a fuzzy implication between the rule antecedents and consequences. The most common inference methods applied in fuzzy control are Max-Min method and Max-Prod method. There are three fuzzy primitives needed to do inference with the membership functions and rules we have described above. The firing strength of each rule is calculated as a fuzzy AND of its conditions; the weight given to each output membership function is calculated as a fuzzy OR of the firing strengths of each rule which leads to that conclusion; and finally, the crisp final output is calculated as a defuzzification of the weights for each output membership function.

4.3.3.1 Max-min method

The fuzzy controller introduced by Assilian and Mamdani [1975] used what is known as the Max-Min method. A 'Min' operator is chosen for the conjunction in the premise of the rule as well as for the implication function and a 'Max' operator is selected for the aggregation. In this research, the Max-Min method is employed as it appears to offer better control results than other methods [Ghwanmeh *et al*, 1995].

4.3.4 Defuzzification

Defuzzification is needed to translate the fuzzy output of a fuzzy controller to a numerical representation which is normally the crisp control action [Yager and Filev, 1993]. The most commonly used defuzzification methods are the *center-of-area* (*CoA*) and *mean-of-maxima (MoM)* [Mamdani and Assilian, 1975]. In a practical set-up, simplifications of the *CoA* defuzzification method are often used in combination with max aggregation because of the introduction of non-linearities in controller output [Jager, 1995]. The theoretical analysis and experimental results proved that the *CoA* strategy has a good steady-state performance. A FLC based on the *CoA* method generally yields a lower mean square error than that based on other methods [Lee, 1990a]. On the other hand, the disadvantage of *MoM* defuzzification is that the fuzzy controller acts as a multi-level relay [Kickert and Mamdani, 1978]. Another disadvantage of this method is the fact that non-symmetrical fuzzy sets defined for the output can result in undesired behaviour [Jager, 1995]. Some comparative studies of the defuzzification process have been reported by [Kickert and Mamdani, 1978], [Jager, 1995]. The conclusion of these studies is that the *CoA* method provides improvement in control results. Based on these considerations, the *CoA* defuzzification method is chosen in designing a fuzzy logic speed controller.

4.3.5 Evaluation of initial scaling factor values

The common approach to the FLC design is based on the mathematical model of the classical controller, i.e. PI or PD controller. The speed error and change of speed error are normally employed as the input variables of the speed controller. Recently, the relationships between the gains of the classical controllers and the scaling factors of the FL controller have been analysed [Parasiliti *et al*, 1997], [Zheng, 1992], [Ying *et al*, 1990], [Galichet and Foulloy, 1995]. In general, the input scaling factors are utilised to map the physical variables (for example, rotor speed in motor control applications) into fuzzy variables. The scaling factors significantly influence the controller behaviour. In a classical controller, the proportional and integral control action are determined by proportional gain and integral gain, while in a FL controller the correlation is not as clear and is much more complex. In general, scaling factors can be implemented in two different forms:

i) *Standard form of scaling factors*

It is common practice to normalise the input and output fuzzy variables in the range of -1 to 1 of the universe of discourse (*UoD*) [Pasino and Yurkovich, 1998], [Ghwanmeh, 1996]. Therefore, the scaling factors have to be employed outside the universe of discourse (*UoD*) of the FL controller structure. The tuning of the scaling factors can then be done more easily until the desired performance is obtained. This approach to the implementation of scaling factors is termed here as the standard form of scaling factors. For example, if the speed error $e(t)$ is selected as the input of a FLC and it is assumed that the maximum magnitude of speed error is $e(t) =$ 100 rad/s, then the speed error scaling factor G_e is defined as

$$G_e = \frac{\textit{maximum range of error in UoD}}{|e(t)|} \tag{4.11}$$

where the maximum range of error in *UoD* is -1 and +1, so that

$$G_e = \frac{1}{|100|} = 0.01$$

Membership functions for this implementation of speed error scaling factor are shown in Fig. 4.2a.

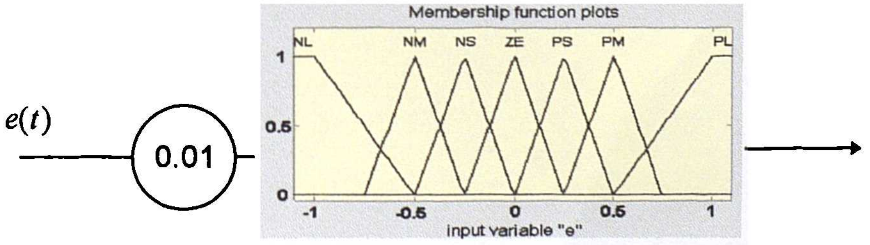


Figure 4.2a: Standard form of error scaling factor implementation.

ii) *Scaling factors in the UoD*

If the scaling factor is utilised within the universe of discourse of the fuzzy variable, the tuning procedure is slightly more difficult because the width and peak value position of membership functions need to be re-calculated when the scaling factors change (Fig. 4.2b) [Passino and Yurkovich, 1998], [Zheng, 1992].

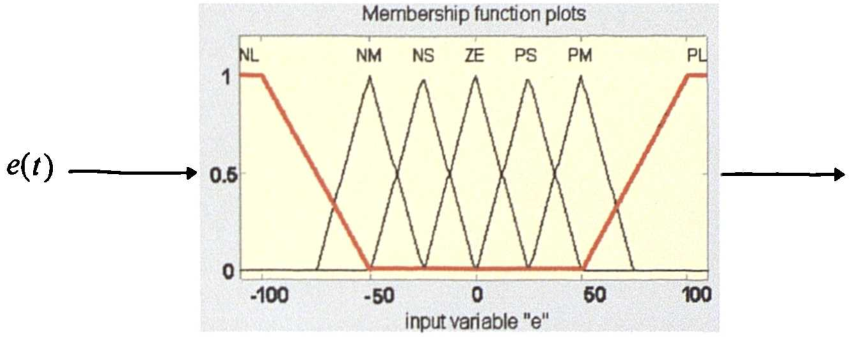


Figure 4.2b: Error scaling factor implementation in the universe of discourse.

In this research, the determination of scaling factors of the FL controller is principally investigated on the basis of previous works by [Ying *et al*, 1990] and [Zheng, 1992]. However, these publications do not emphasise the evaluation of initial scaling factors on the basis of correlation with a linear PI controller. Most of the work attempts to demonstrate the similarities of the step response obtained from FL and non-fuzzy controllers. The evaluation regarding relationships between scaling factors and control performance is more general. Therefore, the possibility to calculate the initial scaling factors based on these methods has to be investigated. The influence of the variation of the scaling factors on control performance needs to be addressed as well. The effectiveness of these methods is studied and compared (Section 4.4.1 and 4.5). The correlation is established by representing the controller parameters in the standard form based on the conventional PI control law. The initial scaling factors can be calculated analytically rather than by trial and error, while the performance of the speed controller obtained using these scaling factors is similar to the one obtained using optimal design (Section 4.5.1).

4.3.5.1 Calculation of initial scaling factors

Many publications [Donescu *et al*, 1996], [Heber *et al*, 1997], [Fodor *et al*, 1997], have shown that scaling factors are the most important parameters because of their significant influence on the overall performance of the controller. The input scaling factors are commonly determined from the maximum input error and maximum input change of error, while the output scaling factor can be determined by trial-and-error. The role of scaling factors is to enable maximum mapping and thus ensure that possible measured values of error and change of error are accurately mapped to the membership functions of fuzzy sets [Barrero *et al*, 1995]. Manual tuning is based on expert knowledge and good performance can be achieved by this technique [Lee, 1993], [Birou and Imecs, 1996], [Donescu *et al*, 1996], [Monti and Scaglia, 1997], [Heber *et al*, 1997], [Fodor *et al*, 1997], [Zhen and Xu, 1996], [Linkens and Abbod, 1991]. The disadvantage of this technique is that scaling factors should be re-tuned manually in order to maintain the required performance especially when the plant parameters change or for a different operating condition (Chapter 6).

The scaling factors are calculated here based on the correlation between scaling factors of the FL controller and gains of the linear PI controller. Gains of the linear PI speed controller which has been designed in Chapter 3 will be used to calculate the initial scaling factors using the methods of sub-section 4.3.5. Three PI speed controller designs will be considered: the first is zero overshoot design ($K_p = 2.22$, $K_i = 111.0$, Section 3.8.1), the second is 1.3 overshoot design ($K_p = 2.40$, $K_i = 1255.20$, Section 3.8.1) and the third is Ziegler-Nichols tuned PI controller (K_p= 11.47, K_i = 4922.7, Section 3.8.1).

The output scaling factor has less significant influence on the drive behaviour. Usually this value was set to unity [Heber *et al*, 1997], [Bolognani *et al*, 1994] or to two [Parasiliti *et al*, 1996], [Ghwanmeh, 1996]. In this research, two values of the output scaling factor, small (G_u=1) and large (G_u= 3), are considered.

4.3.5.1.1 Initial scaling factor calculation using 'Design case'

This is the most common technique used to calculate the initial scaling factors. As mentioned earlier, the purpose of the input scaling factors G_e and G_{ce} is to map the error and change of error variables e and ce into $e \in E$ and $e \in CE$, respectively. In order to utilise the whole space $E \times CE$, it is desirable to map the maximum absolute value of e or ce into the maximum element in E or CE. By noting the property of linear mapping, we have from (4.11):

$$G_e = \frac{maximum\ range\ of\ error\ in\ UoD}{\left|e\right|_{\max}} \tag{4.11a}$$

$$G_{ce} = \frac{maximum\ range\ of\ change\ error\ in\ UoD}{\left|ce\right|_{\max}} \tag{4.11b}$$

where $\left|e\right|_{\max}$ and $\left|ce\right|_{\max}$ are the magnitude of the maximum error and maximum change of error. In standard form, the maximum ranges of error and change of error in *UoD* are normalised into -1 to +1.

The "initial design" scaling factors are calculated using known motor data (Appendix A). Rated speed of the motor is 180 rad/s electrical and operation in forward direction of rotation is required. Thus, assuming that this is the maximum speed, error is 180 for start-up from standstill and the scaling factor for the speed error is obtained as

$$G_e = \frac{1}{180} \approx 0.0056$$

In general, $\left|ce\right|_{\max}$ cannot easily be decided in advance and trial-and-error is normally used to tune the G_{ce}. The scaling factor for the change of error is calculated on the basis of the rated inertia and the maximum torque that the motor is allowed to develop [Ibrahim *et al*, 1998b] as follows:

$$T_{e\max} = \frac{J_n}{P}\left(\frac{\Delta\omega}{T_s}\right) \tag{4.11c}$$

$$\Delta\omega = \frac{(20.7)\,(3)\,(0.00002)}{0.00176}$$
$$= 0.70568 \quad rad/s$$

$$G_{ce} = \frac{1}{change\ of\ error} = \frac{1}{\Delta\omega} = 1.4$$

4.3.5.1.2 Initial scaling factor calculation using 'Method 1'

This method was proposed by Zheng [1992]. The relationship between a PI controller's parameters and scaling factors of the FL controller is established as follows.

For a fuzzy controller

$$\frac{\Delta u(t)}{G_u^{UoD}} = \frac{\Delta e(t)}{G_{ce}^{UoD}} + \frac{e(t)}{G_e^{UoD}} \tag{4.12}$$

$$\Delta u(t) = \frac{G_u^{UoD}\Delta e(t)}{G_{ce}^{UoD}} + \frac{G_u^{UoD}e(t)}{G_e^{UoD}} \tag{4.13}$$

$$\Delta u(t) = \frac{G_u^{UoD}}{G_{ce}^{UoD}}\left(\Delta e(t) + \frac{G_{ce}^{UoD}}{G_e^{UoD}}e(t)\right) \tag{4.14}$$

The discrete form of the PI controller is once more from equation (4.10) given by

$$u(nT_s) - u(nT_s - T_s) = K_p\left[e(nT_s) - e(nT - T_s)\right] + K_ie(nT_s) \tag{4.10}$$

If the sampling time is T_s, then K_p, T_i and K_i can be defined as [Zheng, 1992]:

$$K_p = \frac{G_u^{UoD}}{G_{ce}^{UoD}} \tag{4.15a}$$

$$T_i = \frac{G_e^{UoD}}{G_{ce}^{UoD}}T_s \tag{4.15b}$$

The definition of integral gain in a continuous and discrete time form is given as

$$K_i^{cont} = \frac{K_p}{T_i} \tag{4.15c}$$

$$K_i^{discrete} = T_s \, K_i^{cont} \tag{4.15d}$$

$$K_i^{cont} = \frac{K_p}{T_i} = \frac{G_u}{G_e}\frac{1}{T_s} \tag{4.15e}$$

$$\frac{G_u}{G_e} = T_s \; K_i^{const} = K_i^{discrete} \tag{4.15f}$$

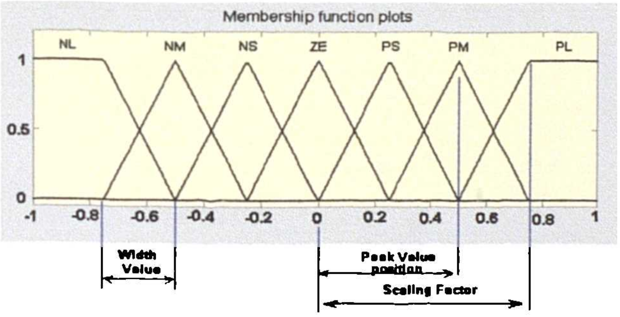
The relationship between scaling factors implemented in the *UoD* and the standard form of scaling factors is given with

$$G_e^{UoD} = \frac{1}{G_e} \tag{4.16a}$$

$$G_{ce}^{UoD} = \frac{1}{G_{ce}} \tag{4.16b}$$

$$G_u^{UoD} = \frac{1}{G_u} \tag{4.16c}$$

Here G_e^{UoD}, G_{ce}^{UoD}, G_u^{UoD} are the scaling factors defined as the maximum peak value in the universe of discourse of the fuzzy variables, as shown in Fig 4.3.



NL = Negative Large
NM = Negative Medium
NS = Negative Small
ZE = Zero
PS = Positive Small
PM = Positive Medium
PL = Positive Large

Figure 4.3: Scaling factor of a membership function.

For the purpose of comparison, equations (4.15) are converted to the standard form of (4.17) by applying equations (4.16):

$$K_p = \frac{G_{ce}}{G_u} \tag{4.17a}$$

$$T_i = \frac{G_{ce}}{G_e} T_s \tag{4.17b}$$

$$K_i = \frac{K_p}{T_i} = K_p \frac{G_e}{G_{ce}} \frac{1}{T_s} \tag{4.17c}$$

The general structure of the FL controller, shown in Fig. 4.4, is based on the standard form of the scaling factors, so that:

$$\left[u(nT_s) - u(nT_s - T_s)\right] G_u = e(nT_s)\, G_e + \left[e(nT_s) - e(nT_s - T_s)\right] G_{ce} \quad (4.18)$$

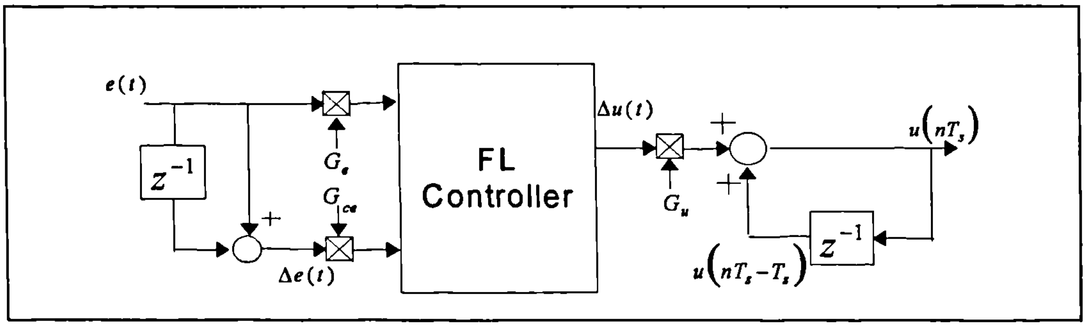


Figure 4.4: FLC with scaling factors.

Figure 4.4 shows that the scaled error and scaled change of error are obtained by multiplying the actual error and change of error with their scaling factors, while the actual change of controller output is obtained by multiplying output scaling factor with the scaled change of controller output.

The calculation of the initial input scaling factors is based on equations (4.17), so that $G_{ce} = K_p\, G_u$, $G_e = K_i\, G_{ce}\left(T_s / K_p\right)$ and sampling time is taken as 20 μs. The calculated input scaling factors are shown in Table 4.2.

Method 1	K_p	K_i	G_u	G_e	G_{ce}
Off-line optimised PI for zero overshoot design	2.22	111	1	0.0022	2.22
	2.22	111	3	0.0066	6.66
Off-line optimised PI for 1.3 rad/s overshoot design	2.4	1255.2	1	0.0251	2.4
	2.4	1255.2	3	0.0753	7.2
PI tuned using Ziegler-Nichols	11.47	4922.7	1	0.0984	11.47
	11.47	4922.7	3	0.2953	34.41

Table 4.2: Different initial scaling factors calculated based on method 1.

The proportional gain K_p in equation 4.17a (method 1), and K_i integral gain in equation 4.17c (method 1) are correlated with G_e, G_{ce} and G_u. The integral gain K_i that results from method 1 can be simplified as $K_i = G_e \; G_u\, T_s$. If G_u and T_s are assumed to be constant, then the integral control action is dependent on the error scaling factor G_e. For example, when G_e is increased, then K_i becomes large, T_i becomes small and the controller will produce strong integral control action. The proportional control gain K_p determined from method 1, can be expressed as $K_p = G_{ce} / G_u$. Hence, the proportional gain K_p is proportional to the change of error scaling factor G_{ce} if G_u is assumed to be a constant.

4.4 DESIGN OF THE CPFL SPEED CONTROLLER

The CPFL speed controller is designed, using the MATLAB Fuzzy Logic Toolbox. It utilises as the input signals speed error and change of speed error at each sampling instant. The control output is the increment of current torque command (Δi_{qs}^*) which is integrated to generate the current command (i_{qs}^*). The error is given by the difference between the speed command (ω^*) and the actual rotor speed (ω), while the change of error is given by the difference between the present error and the previous error. The input variables (*e* and *ce*) are normalised to derive normalised error (e_n) and normalised change of error (ce_n) signals. A fuzzifier is used at the front of the system to convert their crisp data values to fuzzy data. The most widely used fuzzifier is the singleton fuzzifier (used here as well), mainly because of its simplicity and lower computational requirement. The corresponding control rules are evaluated, from membership functions and rule table, composed and finally defuzzified to derive the change of the current torque command (Δi_{qs}^*).

The general procedure for the design of a fuzzy controller was reported by [Dote, 1995]. The CPFL speed controller design in this research is based on the following procedures:

1. Select control variables, such as speed error, change of speed error and current torque command.

2. Determine and select the controller parameters. The most common parameters, reported in literature, are assumed to be the standard

parameters. These include Mamdani fuzzy rules (Section 4.3.1), triangular membership functions (Section 4.3.2), Max-Min fuzzy inference (Section 4.3.3.1) and *CoA* defuzzification technique (Section 4.3.4).

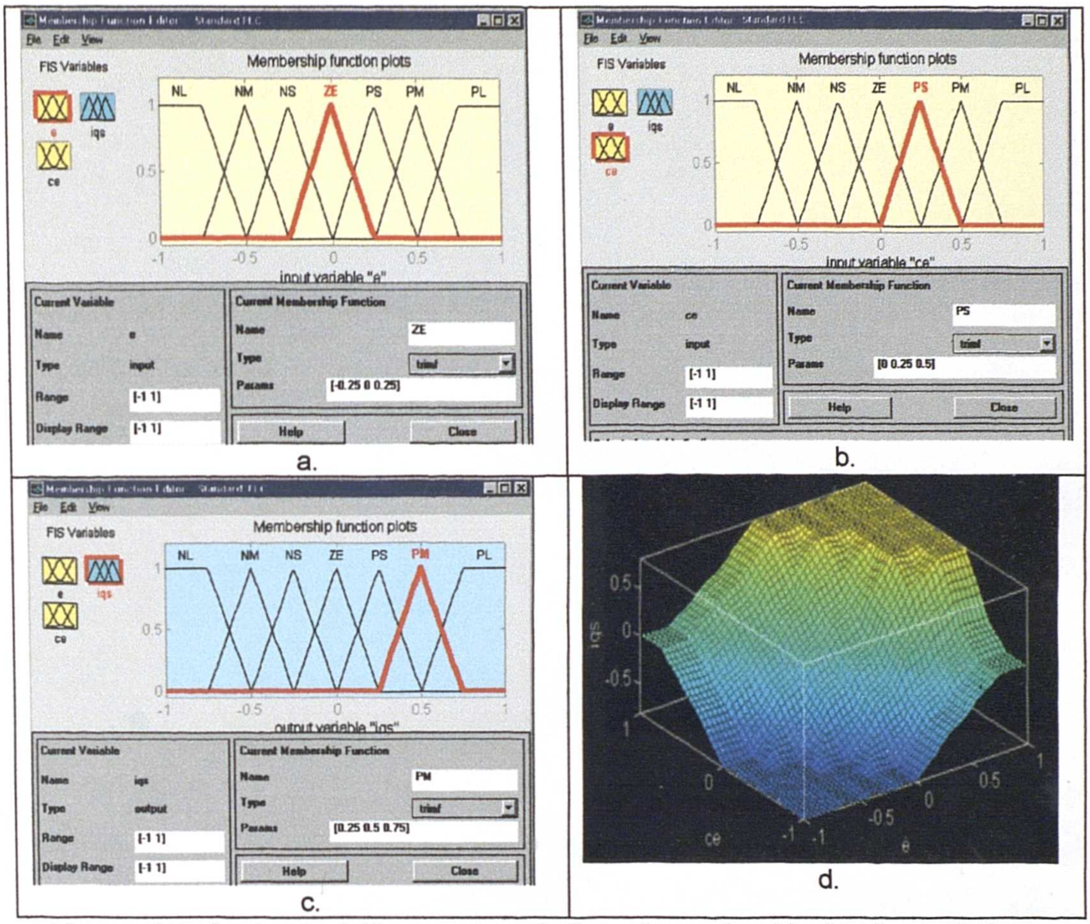
3. Assign labels (membership functions) for input and output fuzzy variables; seven levels of membership function are used, NL, NM, NS, ZE, PS, PM and PL. Therefore, 49 fuzzy rules are produced.
4. Determine fuzzy rules as described in section 4.3.1.
5. Decide upon control performance evaluation methods. At first, two designs are considered: zero overshoot and 1.3 rad/s overshoot. Both designs are tuned to obtain minimum rise-time and settling-time. The overall performance is examined based on integral speed error criterion (Section 4.5.1).
6. Optimisation of performance: tune membership functions, fuzzy rules and scaling factors manually, by many simulations runs.

Step 6 should be repeated until the desired performance is obtained.

4.4.1 Design of the standard CPFL speed controller based on FL-PI correlation approach

The initial design of the standard CPFL controller is shown in Fig. 4.5. Seven triangular membership functions are used to re-present the input and output fuzzy logic controller variables. The controller is designed based on 49 fixed fuzzy rules (Section 4.3.1), fixed membership functions (Section 4.3.2), Max-Min fuzzy inference (Section 4.3.3.1) and *CoA* defuzzification technique (Section 4.3.4). The triangular membership functions are designed to be symmetrical and identical in terms of width and peak position as shown in Fig. 4.5. Different positioning of peak and different width values can affect the rules fired [Zheng, 1992]. In this section, the initial values of the scaling factors are calculated based on method 1 (Section 4.3.5.1.2). The effectiveness of these methods is investigated by many simulation runs. Fig.

4.5d shows the control surface of the standard FL controller. It is evident that the control surface near to the zero control action is approximately linear. Therefore, the correlation technique discussed in Section 4.3.5 theoretically holds true.



a. Current Variable — Name: e; Type: input; Range: [-1 1]; Display Range: [-1 1]. Current Membership Function — Name: ZE; Type: trimf; Params: [-0.25 0 0.25]

b. Current Variable — Name: ce; Type: input; Range: [-1 1]; Display Range: [-1 1]. Current Membership Function — Name: PS; Type: trimf; Params: [0 0.25 0.5]

c. Current Variable — Name: iqs; Type: output; Range: [-1 1]; Display Range: [-1 1]. Current Membership Function — Name: PM; Type: trimf; Params: [0.25 0.5 0.75]

Figure 4.5: A standard FL speed controller: a) membership functions of the error variable; b) membership functions of the change of error; c) output membership functions; d) three dimensional control surface.

The transient speed responses that follow a rated speed command change from 0 rad/s to 180 rad/s under no-load conditions, for different initial scaling factors (Section 4.3.5.1.2), are given in Fig. 4.6. It is obvious that Plot 3 is the best response, with a very short rise time and settling time, without any overshoot. Further investigation is therefore based on these settings. It is important to note that Plot 3 leads to a zero overshoot, although scaling factors for this FL speed controller have been calculated using off-line optimised PI speed controller parameters for 1.3 rad/s overshoot.

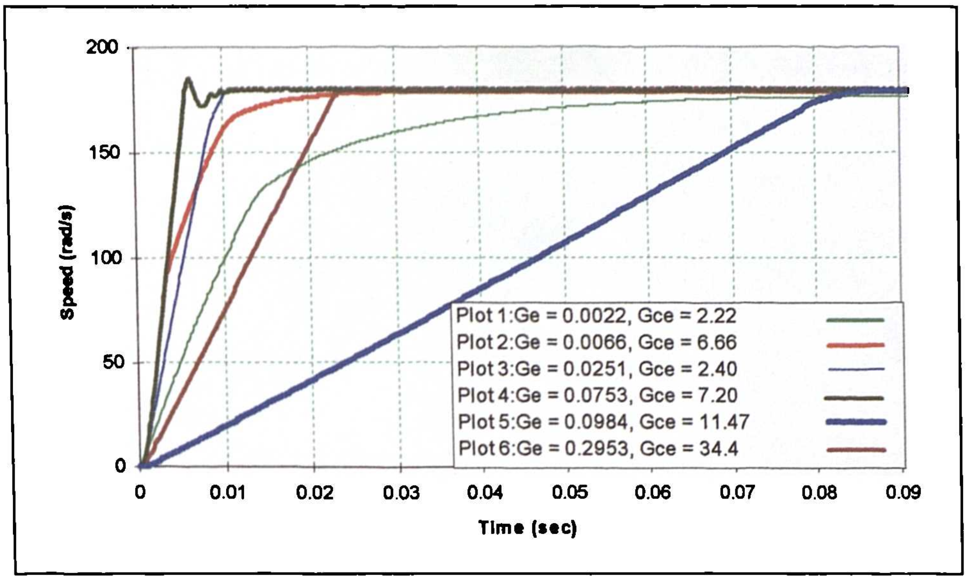


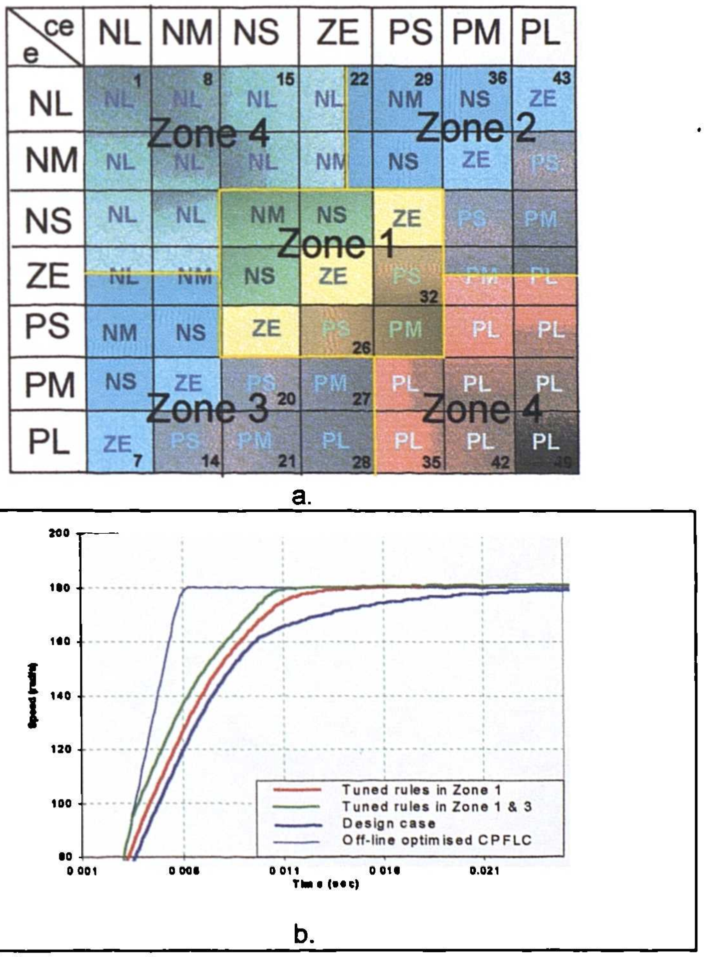
Figure 4.6: Speed response for different initial scaling factors based on method 1.

4.4.2 Off-line optimised CPFL controller

The off-line optimised CPFL speed controller is designed based on the trial and error method. The initial controller parameters are based on 49 fixed fuzzy rules (Section 4.3.1), fixed membership functions (Section 4.3.2), Max-Min fuzzy inference (Section 4.3.3.1), *CoA* defuzzification technique (Section 4.3.4) and scaling factors of Section 4.3.5.1.1. Traces labelled as "Design case" in subsequent figures apply to this initial structure of the FL speed controller, with scaling factors of Section 4.3.5.1.1. Then, the parameters such as rules, the width of membership functions and scaling factors are tuned manually in the order of their significance [Zheng, 1992] until the desired performance is obtained. The procedure of the scaling factor tuning will be given in the Section 4.4.2.3. The tuning of the rules corresponds to the specific zones of a rule table, which affect the performance characteristics of a control system in different ways [Zheng, 1992], [Lee, 1990a, 1990b], as shown in Fig. 4.7.

4.4.2.1 Rule tuning

Rules in specific zones of a rule table will affect the performance characteristics of a control system in different ways. Figure 4.7a is obtained after many simulation runs. Rules in zone 1 and zone 3 are fired when the speed command is positive, while rules in zone 1 and 2 are fired if the speed command is negative. In both cases, 60% to 70% of the total rules are fired in zone 1. In this zone, the controlled variable is close to the steady state. Therefore, the rules in this zone are related to the behaviour of the control system in the vicinity of the steady-state and affect the overshoot and settling time. Most of the rules are fired in zone 2 or zone 3 during the transient period, due to a load rejection or large and small step changes of speed command. This relates to the large signal response, which is usually measured by rise time. The rules in zone 4 are less important for the performance of the control system because of the lower firing frequency [Zheng, 1992]. The settling time of speed response is improved by modifying rules No. 32 and No. 26 in zone 1 from PS into PM as shown in Fig. 4.7b. Further improvement in rise time and settling time is obtained by modifying the degree of control action for rules number 14, 20, 21 and 27 in zone 3 to PL (Fig. 4.7a).



e \ ce	NL	NM	NS	ZE	PS	PM	PL
NL	1 NL	8 NL	15 NL	22 NL	29 NM	36 NS	43 ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	NM	NS	ZE	PS	PM
ZE	NL	NM	NS	ZE	PS 32	PM	PL
PS	NM	NS	ZE	PS 26	PM	PL	PL
PM	NS	ZE	PS 20	PM 27	PL	PL	PL
PL	ZE 7	PS 14	PM 21	PL 28	PL 35	PL 42	PL 49

a.

b.

Figure 4.7: Tuning the rules: a) zones in the rule table; b) response based on tuning of rules in different zones.

4.4.2.2 Tuning of membership functions

The position of the peak value of the membership functions for the speed error can be altered to improve the control performance in terms of rise time and settling time for a large speed command (Fig. 4.8a), load rejection (Fig. 4.8b) and small step change in speed command (Fig. 4.8c). The fuzzy rules, scaling factors and change of error membership functions remain fixed as before. The speed error membership functions, namely ZE, PS, PM, NS and NM are modified to improve the drive behaviour especially in the vicinity of the set point. Tuning the width and moving the peak value position of these membership functions towards the zero error value will cause the speed controller to be more sensitive to a small change in speed error and produce a large control action. The best performance is obtained when the membership functions of speed error are tuned according to Fig. 4.8d.

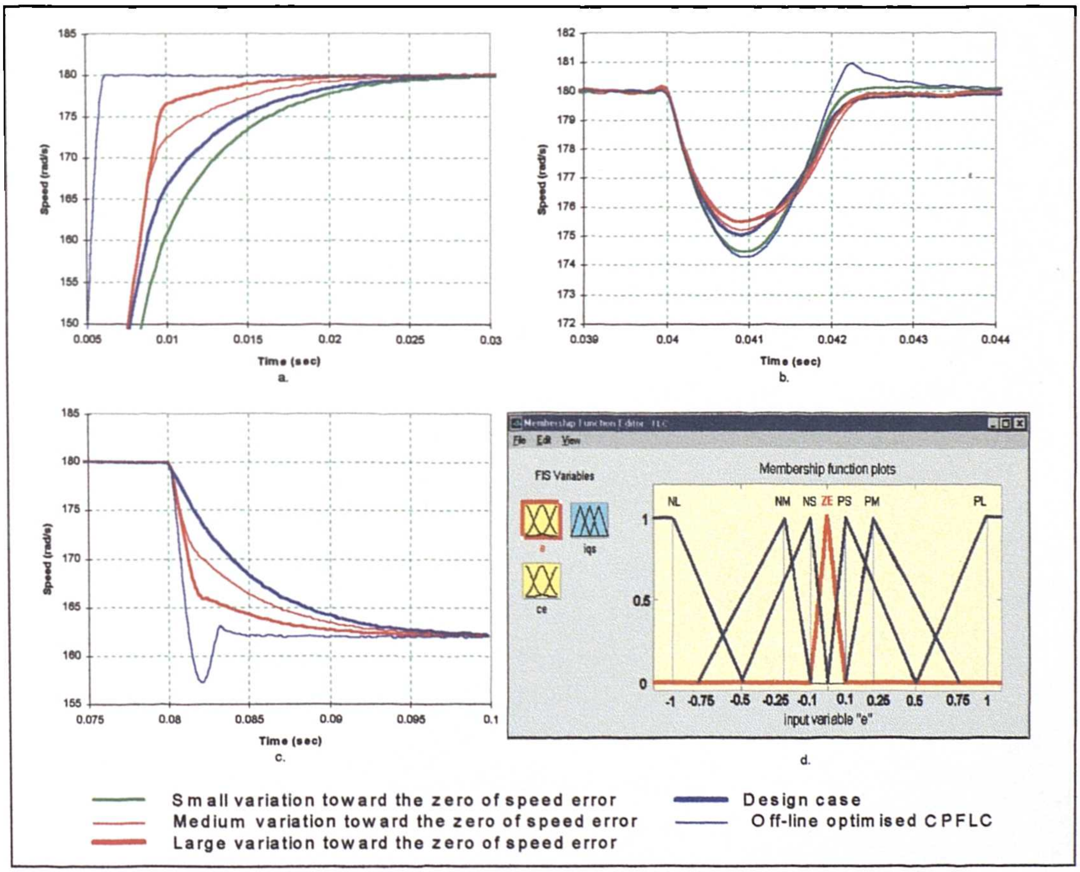


Figure 4.8: Influence of peak value position of error membership functions on control system performance: a) large step speed command; b) load torque applications; c) small change in step speed command; d) membership functions are tuned towards zero value of speed error.

The details of width and peak value position of tuned membership functions of Fig. 4.8d are given in Table 4.3.

Membership functions	Width value		Peak value position	
	Initial design	Tuned	Initial design	Tuned
ZE	±0.25	±0.1	0	0
PS, NS	±0.25	±0.1, ±0.4	±0.25	±0.1
PM, NM	±0.25	±0.15, ±0.5	±0.5	±0.25

Table 4.3: Tuning of peak value positions towards the zero of speed error.

Membership functions	Width value		Peak value position	
	Initial design	Tuned	Initial design	Tuned
ZE	±0.25	±.0.5	0	0
PS, NS	±0.25	±0.5, ±0.25	±0.25	±0.5
PM, NM	±0.25	±0.5, ±0.25	±0.5	±0.75

Table 4.4: Tuning of peak value positions away from the zero of speed error.

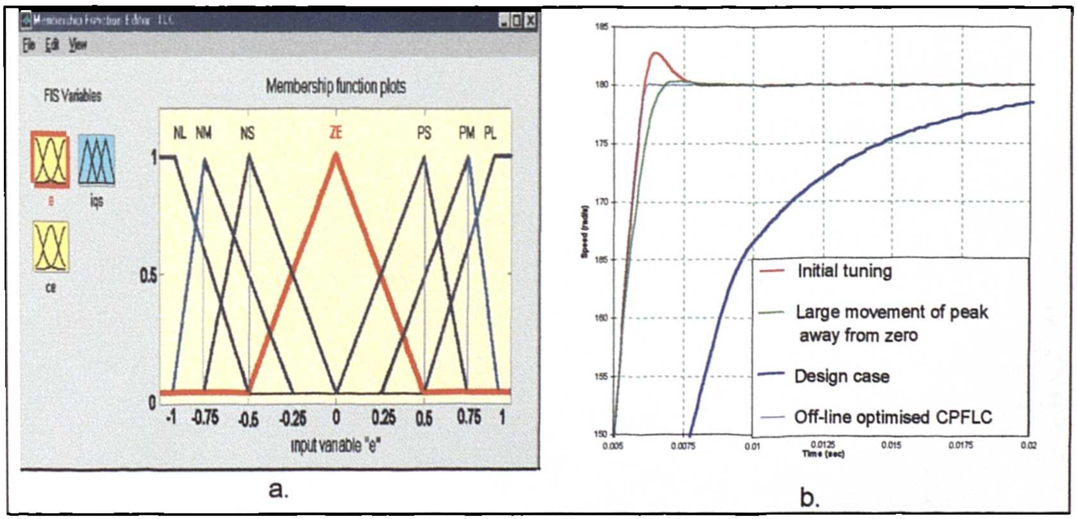


Figure 4.9 : Tuning the width and moving the peak of error membership functions away from zero of error: a) tuned membership functions for speed error variable; b) speed response.

The position of the peak value of membership functions for speed error variable can be utilised to reduce the large overshoot in speed response (Fig. 4.9b, initial tuning curve; in this initial tuning a deliberate overshoot is created , so that impact of width and movement of the peak of membership functions can be studied). In this case, the change of error scaling factor is chosen at first to give a large overshoot in speed response (G_{ce} = 0.9). The tuning strategy is to improve the speed response by tuning the width and position of peak values of membership functions away from zero of error (Fig. 4.9a and Table 4.4). Therefore the FL speed controller will be less sensitive to a small change in speed error and as a result a small control action is produced (Fig. 4.9b).

The same tuning procedure illustrated in Fig. 4.8 is repeated for change of speed error. The membership functions for the change of error are at first tuned towards the zero error value (Fig. 4.10a) based on Table 4.3. The response shown in Fig. 4.11a has the same behaviour as the response illustrated in Fig. 4.9b. In this case, a small control action is produced when the membership functions of speed error are tuned away from zero or toward zero for a change in speed error. However, a large control action is obtained when the membership functions for change of error are tuned away from zero. An overshoot in speed response occurs, as shown in Fig. 4.11b. Once again, this behaviour is similar to the one obtained when the membership functions for speed error are tuned towards the zero of the speed error.

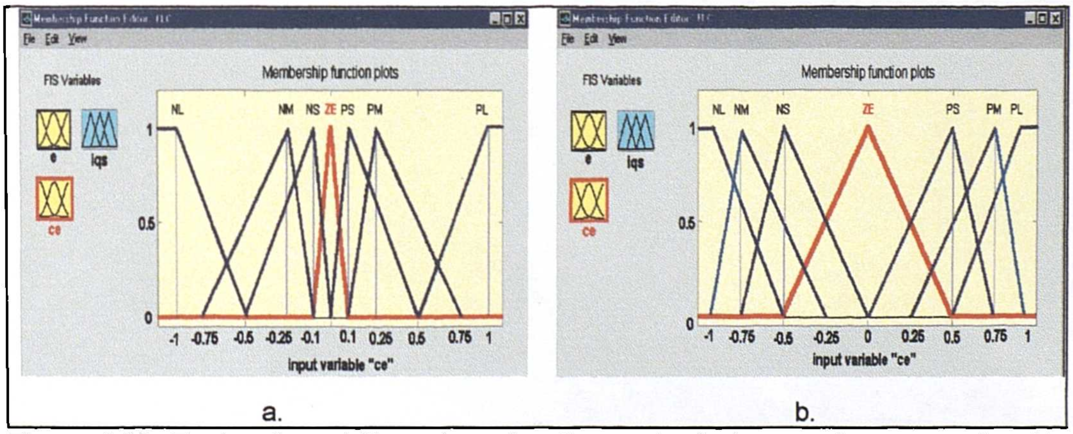


Figure 4.10: Tuning the width and moving the peak of change of error membership functions: a) tuned membership functions towards zero; b) tuned membership functions away from zero.

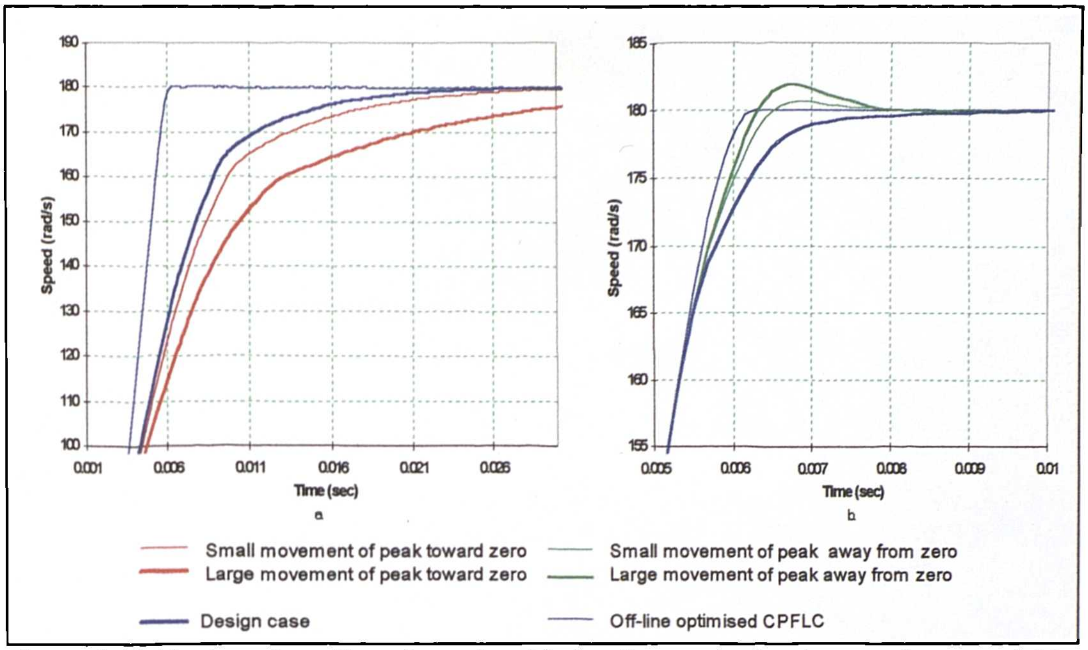


Figure 4.11: Tuning of position of membership function peaks for change of error variable: a) response for membership functions tuned towards zero; b) response for membership functions tuned away from zero.

The output membership functions have less impact than error and change of error membership functions on the control action. This is because the absolute control action is assumed to correspond to the area ' of the membership function and the shape of membership function has less influence. It should be noted that tuning of membership functions for speed error and change of speed error affects the firing of fuzzy rules.

Observations of this section, with regard to the impact of the membership function width and variation of the membership function peak position on the resulting speed response, are utilised next in the design of the final form of the off-line optimised FL speed controller.

4.4.2.3 Tuning of scaling factors and final form of the off-line optimised FL speed controller

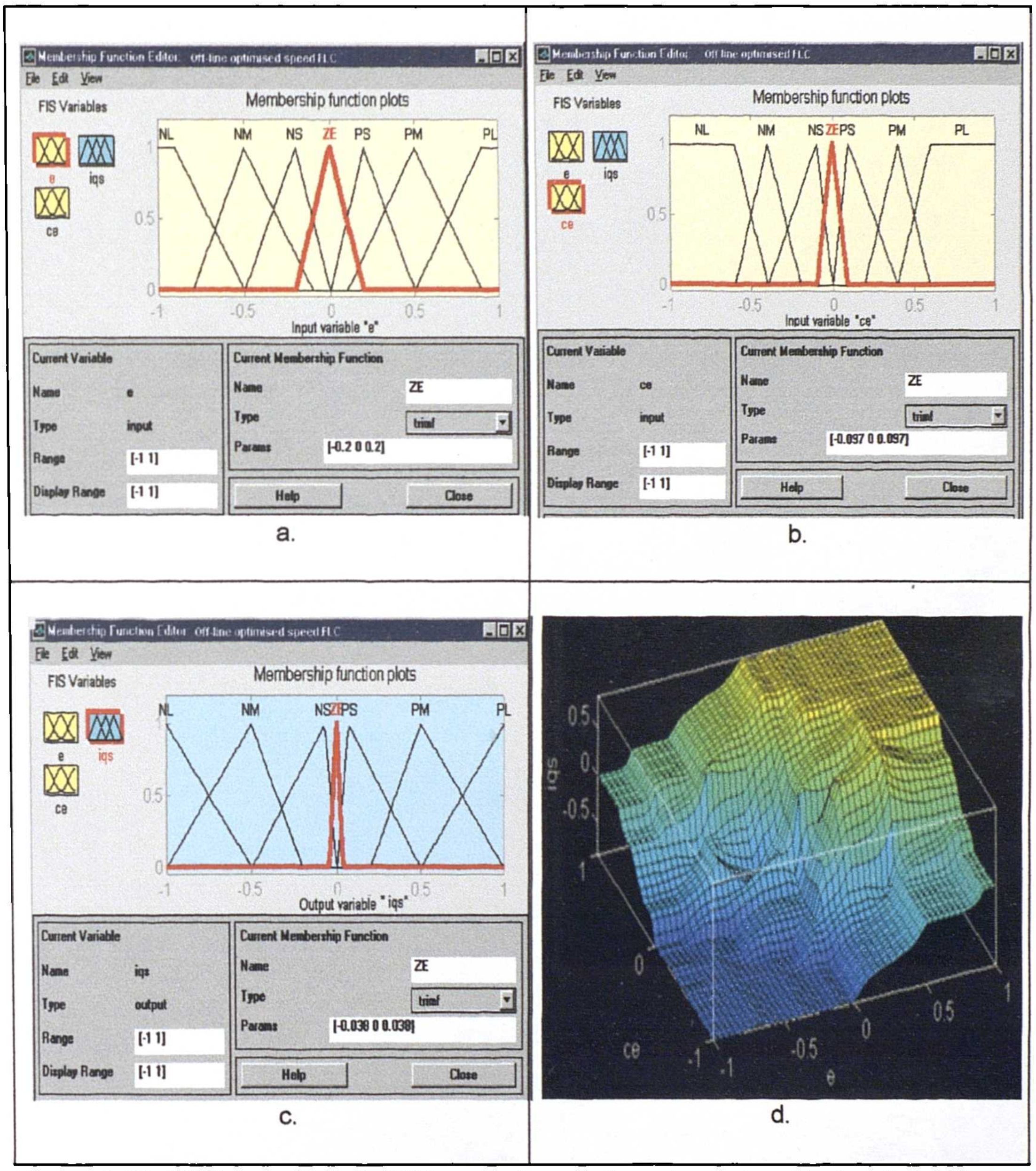
Transients illustrated in Figs. 4.7b, 4.8a, 4.9b and 4.11a, b are obtained after tuning the fuzzy rules and membership functions. The aim of the tuning procedure is to produce a better control behaviour. However, it is clearly demonstrated in these figures that the rise and settling times are far longer than for the response obtained from the off-line optimised FL speed controller. A short rise time, as good as in an optimal response, can be achieved but at the same time the response will have a large overshoot (Fig. 4.11b). The last alternative that can be used to obtain the desired response is tuning of the scaling factors. Scaling factors have a significant influence on the control system performance. Therefore, the method used in this section is to tune the scaling factors, while membership functions and fuzzy rules are re-tuned as well, based on the knowledge gained from the Section 4.4.2.1 and Section 4.4.2.2, until the desired performance is obtained.

After many simulation runs, the final membership functions for the error, change of error and output variables, that give an optimal performance are obtained in the form presented in Figs. 4.12a,b,c, while the final fuzzy rules are mostly based on rules in Table 4.1. It should be noted that these membership functions differ from those shown in Figs. 4.8d and 4.10a. The outputs relating to rules number 7, 14, 20 and 21 are modified to produce a large control action (Fig. 4.7a). The three-dimensional presentation of the optimal control surface is depicted in Fig. 4.12d. It is noted that a non-linear behaviour in the area of the set point can be observed, while the control surface obtained based on standard FL controller, illustrated in Fig. 4.5d, is seen to be more linear. It follows that the non-linear control behaviour of the FL controller can be generated by utilising different sets of rules and membership functions. The optimal speed response shown in Fig. 4.13a is based on G_e = 0.0023, G_{ce} = 0.41, G_u = 3 producing an overshoot of less than 0.1 rad/s (aperiodic speed response), while the transient plotted in Fig 4.13b is based on G_e = 0.0025, G_{ce} = 0.43 and G_u = 2.5 with a 1.3 rad/s overshoot. Both designs apply to the off-line optimised FL speed controller. Further investigations regarding the performance and the behaviour of the FL speed controller are therefore based on the following two FL speed controllers:

i) Off-line optimised FL speed controller for zero overshoot (G_e= 0.0023, G_{ce} = 0.41, G_u = 3).

ii) Off-line optimised FL speed controller for 1.3 rad/s overshoot (G_e = 0.0025, G_{ce} = 0.43, G_u = 2.5).

The difference between these two designs is the use of different scaling factors, while the controller structure is the same.



Current Variable
Name e
Type input
Range [-1 1]
Display Range [-1 1]

Current Membership Function
Name ZE
Type trimf
Params [-0.2 0 0.2]

a.

Current Variable
Name ce
Type input
Range [-1 1]
Display Range [-1 1]

Current Membership Function
Name ZE
Type trimf
Params [-0.097 0 0.097]

b.

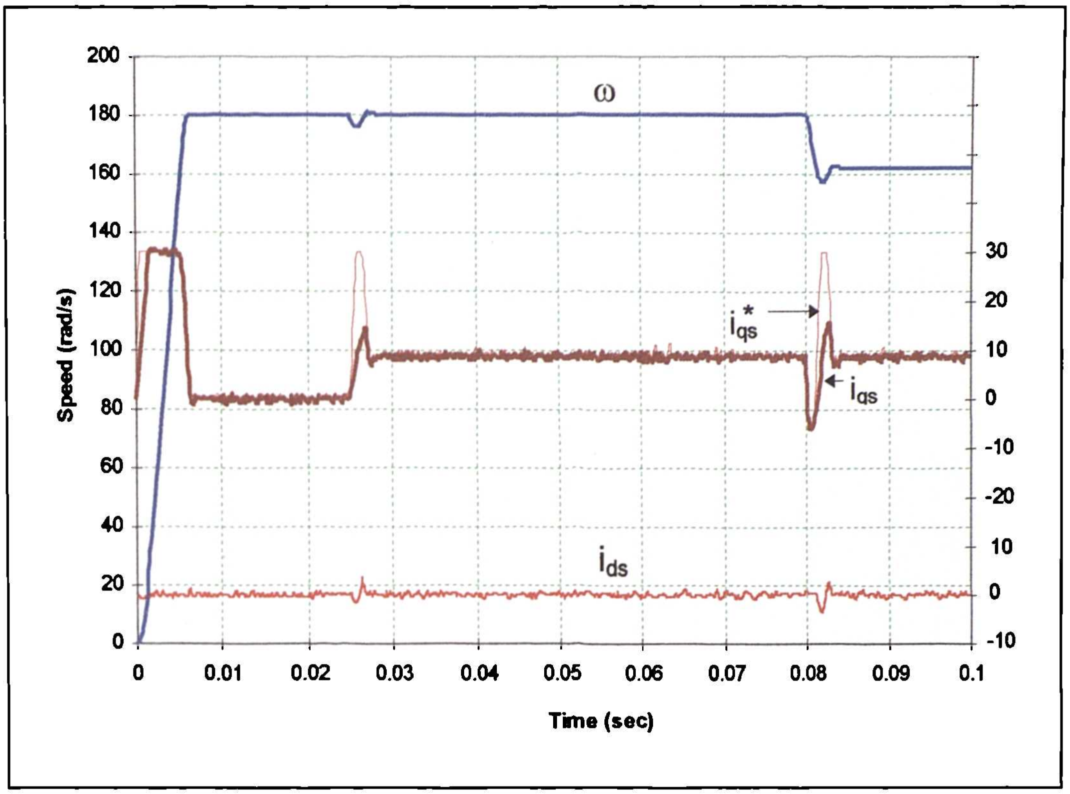
Current Variable
Name iqs
Type output
Range [-1 1]
Display Range [-1 1]

Current Membership Function
Name ZE
Type trimf
Params [-0.038 0 0.038]

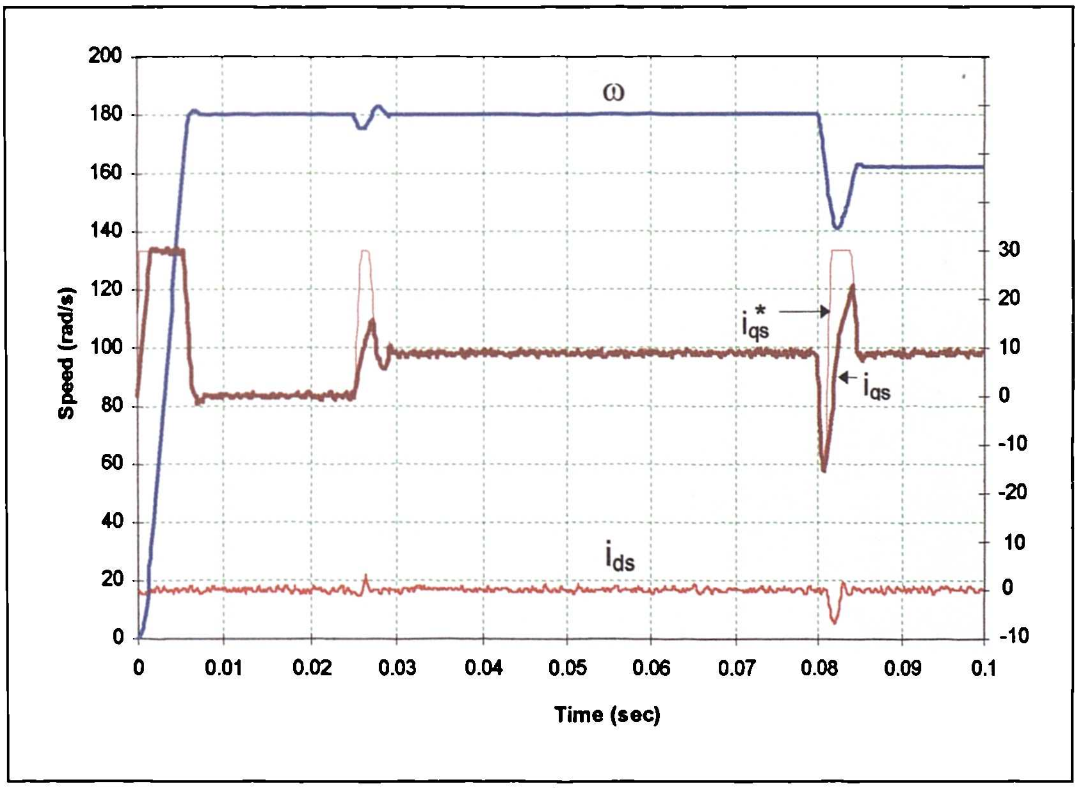
c.

d.

Figure 4.12: Off-line optimised CPFL speed controller: a) membership functions of error variable; b) membership functions of change of error variable; c) membership function of output variable; d) three-dimensional control surface .



a.



b.

Figure 4.13: Rotor flux oriented control with off-line optimised CPFL speed controller and hysteresis current controller: a) speed response with zero overshoot design; b) speed response with 1.3 rad/s overshoot design.

4.5 ANALYSIS OF THE CONTROLLER BEHAVIOUR WITH DIFFERENT INITIAL SCALING FACTORS

4.5.1 Simulation results

The mathematical relationships pertaining to the FL controller and PI linear controller parameters were discussed in Sections 4.2.2 and 4.3.5.1. In this section, the performance of the FL controller is evaluated and compared, for different sets of initial scaling factors. The methods described in Section 4.3.5 are employed to estimate the initial scaling factors. The integral of the absolute value of error (IAE) criterion and integral of time multiplied by absolute error (ITAE) criterion are used to measure the controller performance, as shown in Table 4.5 (these two performance criteria are described in more detail in Section 5.2.2). The simulation results demonstrate that method 1 yields good response in terms of rise time (Fig. 4.6, Plot 3), ITAE and IAE (Table 4.5). The speed response in method 1 is improved in terms of rise time when the output scaling factor is increased from 1 to 3 (Fig. 4.6, Plots 2,4,6). The worst results are obtained with the scaling factors of the 'design case' (Table 4.5 and Fig. 4.14).

Type of speed controller	Scaling factors		ITAE	IAE	Rise time (sec)	Settling time (sec)
	G_e	G_{ce}				
Off-line optimised CPFL controller	0.0023	0.41	0.0032	0.7130	0.0054	0.0062
Design case	0.005	1.4	0.0067	0.9852	0.02	0.038
Method '1', 3rd row of Table 4.2	0.0251	2.40	0.0047	0.6373	0.0105	0.012

Table 4.5: A comparison of the best selected speed responses for different initial scaling factor calculation methods (zero overshoot speed response).

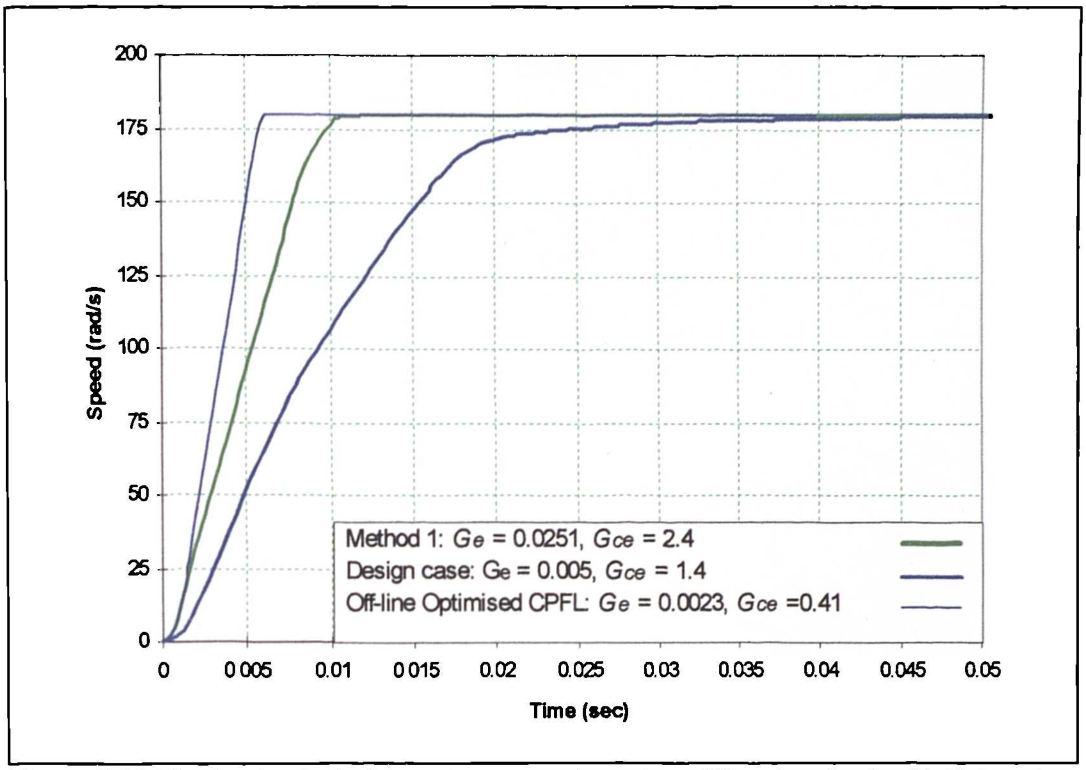


Figure 4.14: The best selected speed response based on different methods of initial scaling factor calculation (zero overshoot speed response).

The calculation of the initial scaling factors using the described techniques can produce good results. For example, for method 1 the ITAE is 0.0047 as compared to 0.0032 for off-line optimised FLC and 0.0067 for the design case. The rise time in method 1 is 0.0105 sec. and 0.03 sec. for design case. The settling time in method 1 is 0.012 sec., compared to 0.0062 sec. for the off-line optimised FL controller and 0.038 sec. for design case.

A further comparison of the transients for off-line optimised FL speed controller and standard FL speed controller (Method 1, Plot 3 of Fig. 4.6) is illustrated in Fig. 4.15. Off-line optimised FL controller has better behaviour for rated load torque application. The undershoot for a small step change in speed is better with the off-line optimised FL speed controller. The only disadvantage of the off-line optimised FL controller is that it requires a long development time in order to obtain optimal performance. Most of the time is used to tune the FL parameters such as width of membership functions, scaling factors and rule base.

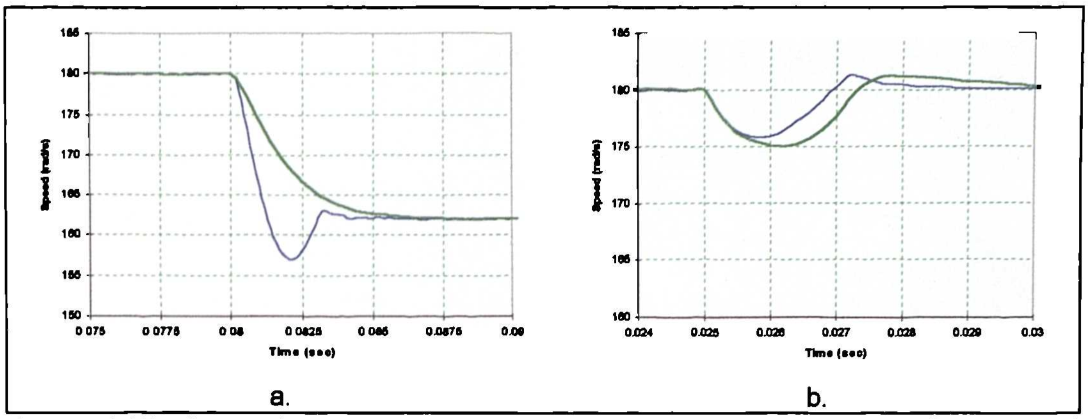


Figure 4.15: Comparison of response obtained with off-line optimised CPFL speed controller and standard CPFL controller where initial scaling factors are calculated based on method 1: a) response to change of speed command from rated to 0.9 times rated; b) response to step rated torque application. (🟦 for Off-line optimised CPFL controller, 🟩 for method 1)

4.5.2 Influence of scaling factors on the dynamic response and speed error

The influence of scaling factors on the dynamic response and speed error near to the set point is investigated by increasing or decreasing the scaling factors by 25%, 50% and 80%. The initial scaling factors used in the study are G_e = 0.0251, G_{ce} = 2.4, and G_u = 1 or 3 (Table 4.2). Figure 4.16 shows the speed response for different values of error scaling factor, while G_{ce} is kept constant. A shorter rise time can be obtained when the error and output scaling factors are increased (Fig. 4.16a,b). On the other hand, large overshoot in speed response occurs if G_e is very large. The behaviour of the FL controller with variations of scaling factors can be related to the PI linear control law as follows.

Strong integral control action is produced when G_e increases, as the integral time constant decreases (equations (4.17)). Therefore, the rise time of the speed response and speed error behaviour are improved (Fig. 4.16a,b). When G_e is decreased the controller produces a smaller integral control action. It is noted that the speed response has smaller or no overshoot (Fig. 4.16c,d). However, the speed error behaviour near to the set point is degraded (Fig. 4.16c,d). In other words, the controller needs a longer time to achieve zero steady state error.

The response for a large step speed command results in a large overshoot when G_e is increased (Fig. 4.16a,b, Plot 3,4). However, this overshoot can be reduced by decreasing G_e (Fig. 4.16c,d, Plot 3,4). Therefore, further investigation regarding the behaviour of standard CPFL speed controller is conducted with a preferred decrease in G_e.

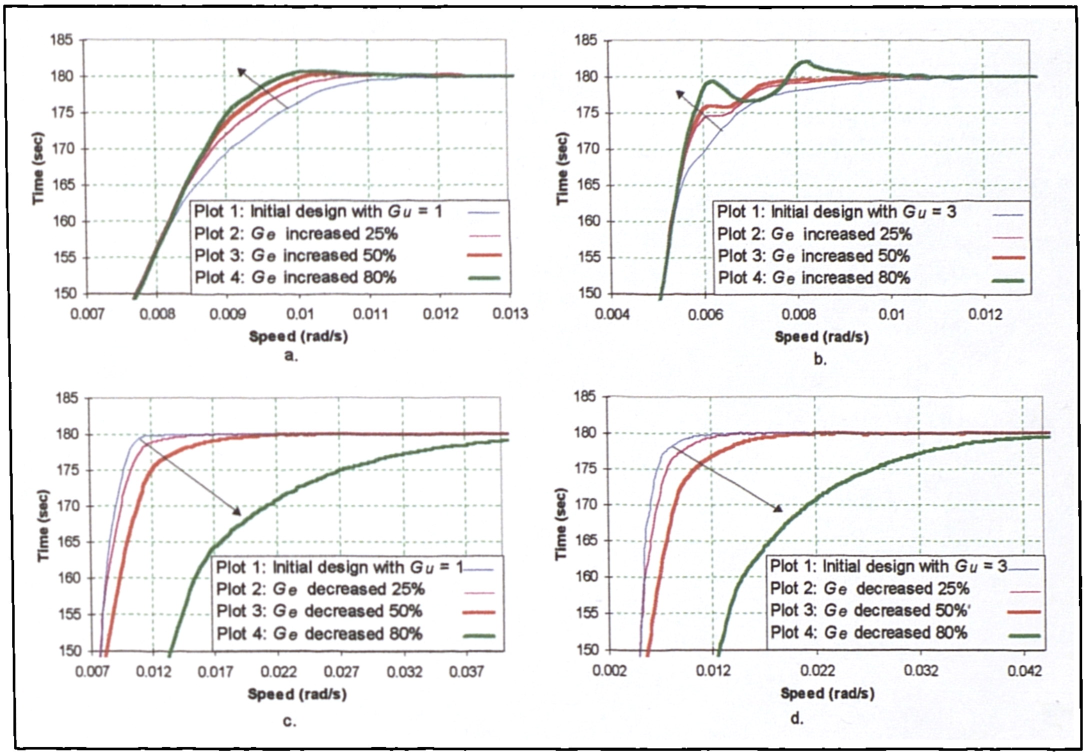


Figure 4.16: Speed response for different values of error scaling factor: a) G_e is increased and G_u = 1; b) G_e is increased and G_u = 3; c) G_e is decreased and G_u = 1; d) G_e is decreased and G_u = 3.

The study illustrated in Fig. 4.16 is repeated for a change of error scaling factor and using the same simulation procedure. In this case G_e and G_u are kept constant. The simulation results reported in Fig. 4.17 indicate that a very low integral control action is produced when G_{ce} is increased too much, leading to an increase in both the rise time and settling time (Fig. 4.17a,b). A very large overshoot in speed response occurs when G_{ce} is decreased, as shown in Fig. 4.17c,d. Decrease in G_{ce} is therefore not considered in the further study.

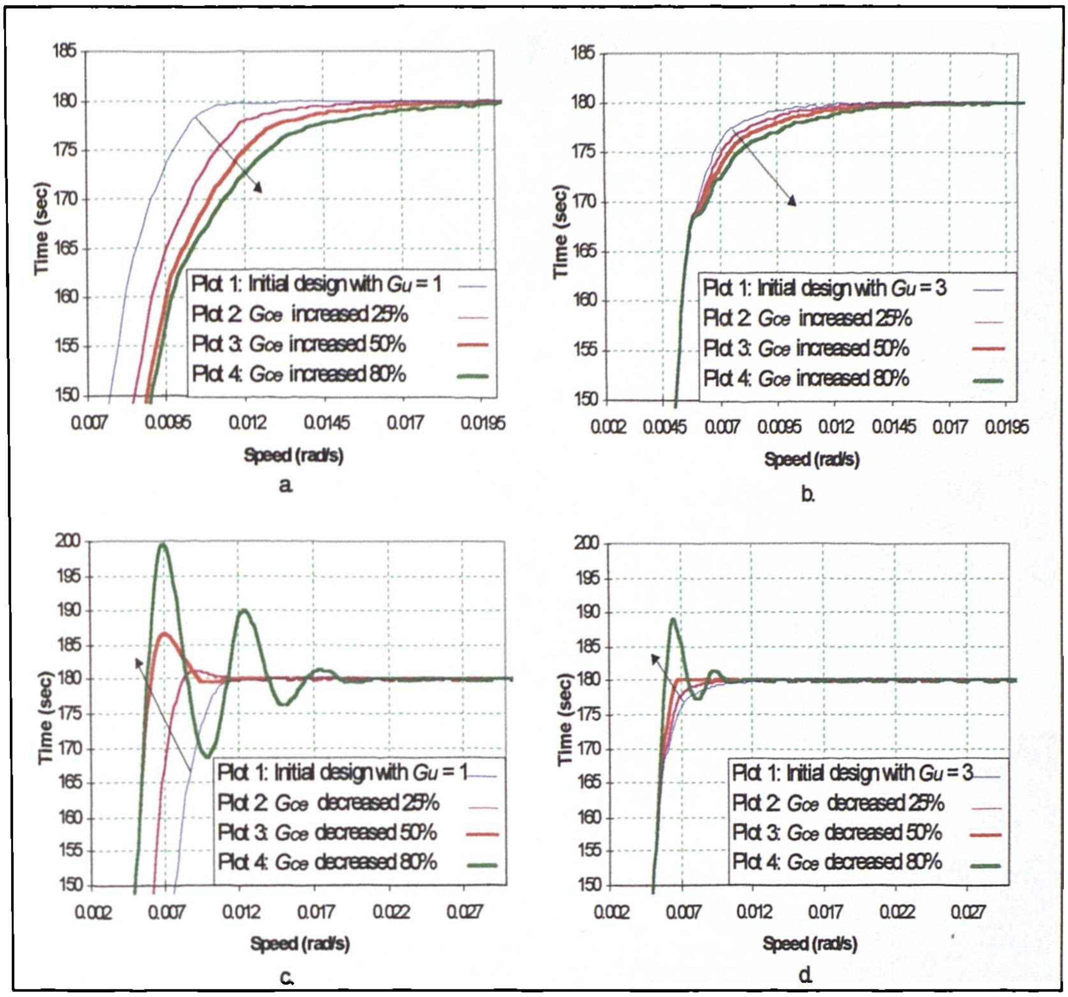


Figure 4.17: Speed response for variation of change of error scaling factor: a) increase in change of error scaling factor with G_u = 1; b) increase in change of error scaling factor with G_u = 3; c) decrease in change of error scaling factor with G_u = 1; d) decrease in change of error scaling factor with G_u =3.

4.5.3 Results of the studies

Simulation results shown in Fig. 4.16 and 4.17 are based on limited selection of scaling factor values. Results of the detailed examination regarding the influence of variation in scaling factors on the dynamic response and speed error behaviour are shown in Figs. 4.18 and 4.19. The speed error near the set point and rise time are measured for every 10% change in input scaling factor. Rated step speed command is applied at *t = 0 s* and speed error is measured at time instant *t = 0.05 s*. In Fig. 4.18b the speed error increases to 0.138% of the rated speed when G_e decreases from 90% to 40% of the original value. The rise time of the speed response also increases by 71%. If the G_e is further decreased from 40% to 10% of the original value, the speed error near to set point increases from 0.138% to 0.76% of the rated speed and the rise time increases up to 814% or 0.057 sec. (Fig. 4.18a,b).

The same case study, done for G_e, is repeated for G_{ce} under the same conditions (G_e and G_u are kept constant). Fig. 4.17 shows that large increase in G_{ce} causes only a small variation in speed error (Fig. 4.18d). However, the rise time is increased by up to 58.3% (0.0124 sec.) when G_u=1 (Fig. 4.19a, Plot 4 and Fig. 4.18c). Short rise time can be obtained if a large output scaling factor is utilised, such as G_u = 3 (Fig. 4.17b and 4.18c). It follows that scaling factors have significant influence on the dynamic response and speed error near to set points (Figs. 4.16, 4.17, 4.18 and 4.19) especially when G_e is decreased or G_{ce} is increased (Fig. 4.19). Figure 4.19 (which is extracted from graphs in Fig. 4.18) shows that the influence in variations of G_e is more dominant than G_{ce} in terms of rise time and accuracy of the speed error near the steady state. Increasing G_{ce} will slow down the integral control action, while at the same time K_p is increased as well. However, the control performance is more sensitive to the variation of G_e especially for an increase of more than 60% from the initial value (Fig. 4.19). Based on this observation, it is concluded that for tuning of the scaling factors, decreasing G_e is preferred. In order to reduce overshoot in the speed response, the error scaling factor is tuned as the first priority. The tuning strategy is as follows: to improve speed response with regard to overshoot, error scaling factor should be reduced but not too much as this will lead to large speed error. In some cases, two sets of error scaling factor can be used, the first scaling factor is applied during transient and the second scaling factor during steady state operating conditions. Change of error scaling factor can be reduced as well to improve rise and settling time. Output scaling factor is chosen based on experience and the value is kept constant.

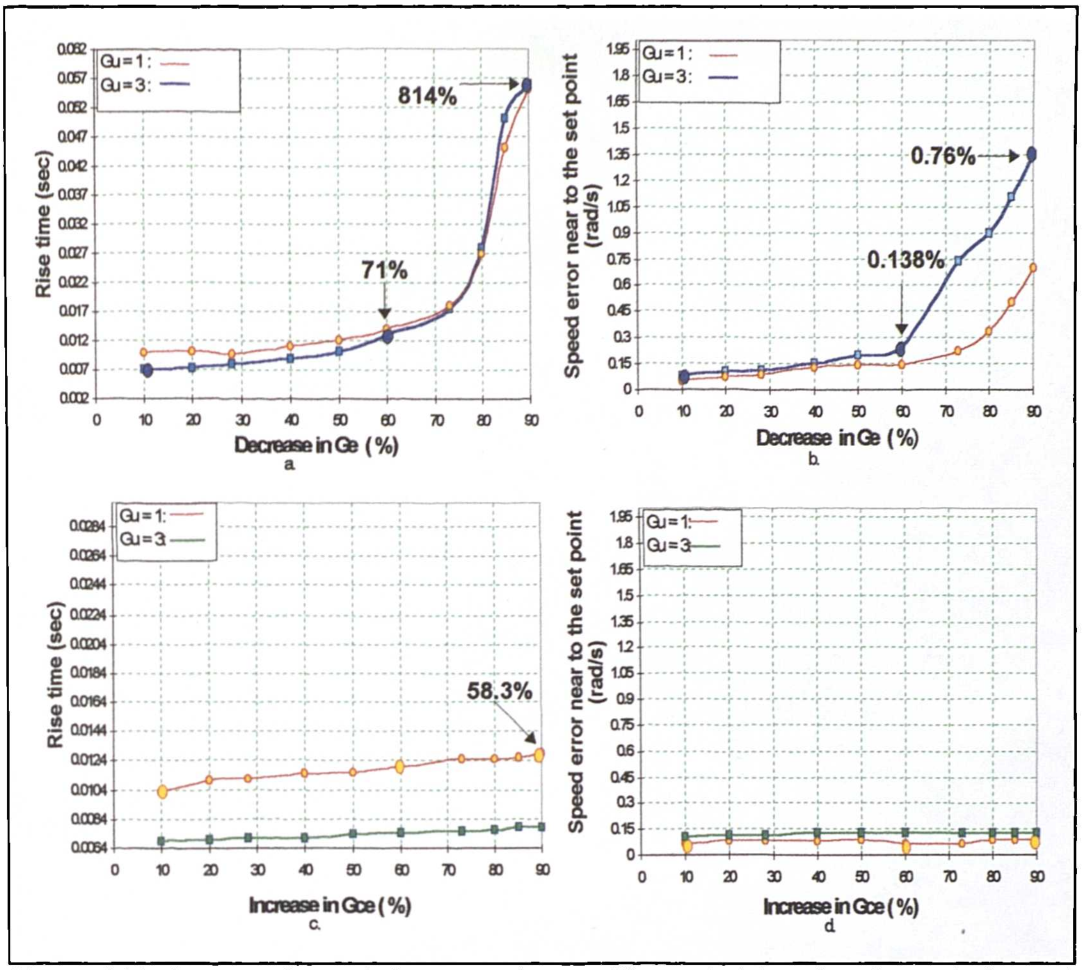


Figure 4.18: A comparison of characteristics for different initial scaling factor designs; a) rise-time for different set of Ge; b) speed error for different set of Ge; c) rise-time for different Gce; d) speed error for different set of Gce.

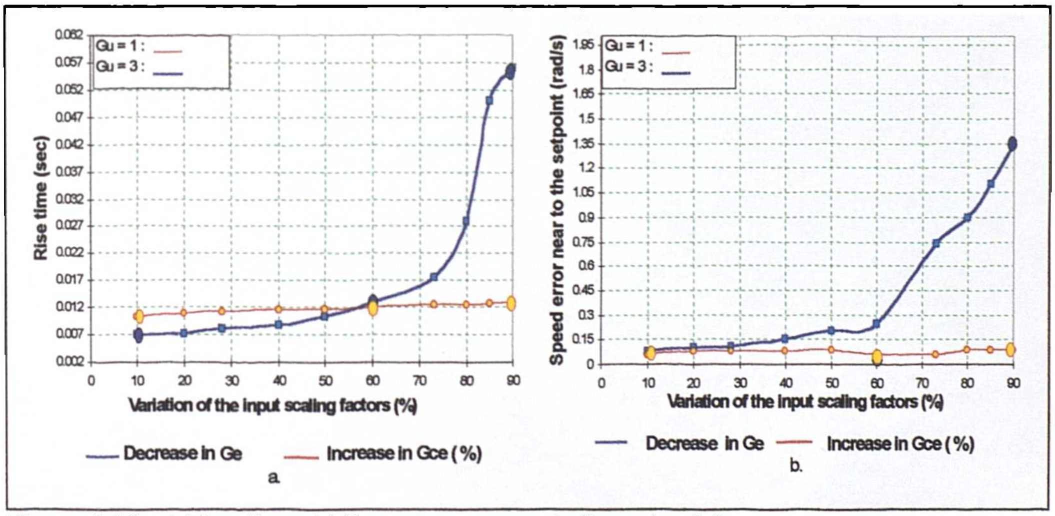


Figure 4.19: a) Rise time; b) Speed error near to the set point.

4.6 SUMMARY

The design procedure of a FL speed controller is firstly established by studying the analogy between a fuzzy controller and a conventional PI controller. The conventional control law is utilised to correlate the relationship between the scaling factors of the FL controller and the gains of the PI controller. The initial values of input scaling factors for method 1 are calculated directly by considering the gains of PI speed controller that was designed in Chapter 3. Three case studies are considered, off-line optimised PI with 1.3 rad/s overshoot design, off-line optimised PI with zero overshoot design and PI with Ziegler & Nichols design. The simulation study is performed and the results are compared for a large step speed command. The final initial scaling factors for further investigation are chosen based on the best response obtained. The initial rule base is generated based on the most commonly used rule base. The triangular symmetrical membership function is chosen for fuzzy variables of the FL speed controller. The standard FL controller is designed based on these initial settings, while the off-line optimised FL controller is designed by tuning the scaling factors, rule base and membership functions until the desired performance is obtained. The behaviour of the drive is investigated further by tuning the peak position and width of the triangular membership functions away from the zero error/change of error or toward the zero of the error/change of error. The performance of the standard CPFL speed controller and off-line optimised CPFL speed controller are evaluated by comparing the simulation results obtained from many simulation runs. Finally, two designs of the off-line optimised CPFL speed controller are considered for further investigation, while further study about the impact of scaling factor variation on the drive behaviour is based on the standard CPFL speed controller. The correlation between rise-time and speed error in the speed response near to the set point, and the scaling factor variations is analysed and evaluated. The conclusions regarding the scaling factor tuning strategy to improve the drive performance are made.

CHAPTER 5

A COMPARATIVE STUDY OF SPMSM DRIVES CONTROLLED BY PI AND FL SPEED CONTROLLERS

5.1 INTRODUCTION

The standard approach for speed control in industrial drives is to use a PI controller. Fuzzy logic control has recently attracted considerable interest in the electric drive area. In particular, the possible replacement of a proportional plus integral (PI) speed controller with a fuzzy logic (FL) controller has been widely considered, with the common conclusion being that performance of a servo drive can be greatly enhanced by the use of a FL speed controller [Bose, 1994], [Bose, 1997a, 1997b], [Sousa and Bose, 1994], [Vas *et al*, 1994b], [Vas *et al*, 1996a], [Feng and Chen, 1996]. Numerous publications are available that describe the development of a FL speed controller and its simulation and experimental verification in conjunction with high performance DC drives [Bose,1994], vector controlled induction motors [Vas *et al*, 1996a], [Ibalidden, and Goureau, 1996], [Boussak and Bauer, 1996], [Fircarra *et al*, 1996], [Fodor *et al*, 1996], [Ta-Cao and Huy, 1996], [Zhen and Hu, 1996], [Heber *et al*, 1995], permanent magnet synchronous machines [Donescu *et al*,1996], [Eminoglu and Atlas, 1996], [Le-Huy, 1995], and switched reluctance motors [Bolognani and Zigliotto, 1996].

The most common approach to comparison of the behaviour of the servo drive controlled by PI and FL speed controller is based on very limited selection of transient responses. Such a situation is met in many publications

[Vas *et al*, 1996a], [Sousa and Bose, 1994], [Goureau and Ibaliden, 1996], [Bossak and Bauer, 1996], [Ficcara *et al*, 1996], [Fodor *et al*, 1996], [Heber *et al*, 1995], [Donescu *et al*, 1996], [da Silva and Acarnley, 1997], [Baghli *et al*, 1997], [Hissel *et al*, 1997]. The main conclusion of these studies seems to be that FL control provides superior performance [Vas *et al*, 1994b], [Sousa and Bose, 1994], [Goureau and Ibaliden, 1996], [Boussak and Bauer, 1996], [Ficcara *et al*, 1996], [Fodor *et al*, 1996], [Heber *et al*, 1995], [Donescu *et al*, 1996], [Ta-Cao and Huy, 1996], [Zhen and Xu, 1996]. However, results from experiments and simulations, that form the basis of such a statement in many papers, often do not seem to provide enough evidence for this conclusion [Vas *et al*, 1994b],], [Sousa and Bose, 1994], [Goureau and Ibaliden, 1996], [Boussak and Bauer, 1996], [Ficcara *et al*, 1996], [Ta-Cao and Huy, 1996], [Zhen and Xu, 1996]. Examples of attempts to provide an in-depth comparison are rather rare [Fodor *et al*, 1996], [Heber *et al*, 1995], [Donescu *et al*, 1996]. Furthermore, numerous difficulties are encountered in FL speed controller applications, such as design difficulties [Bose, 1994], [Donescu *et al*, 1996], [Le-Huy *et al*, 1995], steady-state speed error [Feng and Chen, 1996], [Vas et al, 1994], [Sousa and Bose, 1994], [Goureau and Ibaliden, 1996] and chattering [Feng and Chen, 1996].

The basis for comparison usually constitutes a limited selection of transients, that typically include one speed command setting and application/removal of the load torque at one speed command. Comparison is usually based on the controller design for aperiodic speed response, in which case PI control is known to exhibit sluggish disturbance rejection properties. The study reported in [da Silva and Acarnley, 1997] performs initial design of the controllers in such a way that the same response to a selected step change in speed command is obtained. Such an approach enables a fair comparison to be performed for different operating conditions. Improvement of response obtained by FL control in [da Silva and Acarnley, 1997] appears to be almost non-existent. Similarly, [Fodor *et al*, 1996] reports that there are transients in which PI control will yield better response, while [Donescu *et al*, 1996], shows that, for certain transients, response of PI and FL control will essentially be the same.

This chapter attempts to provide a thorough comparative insight into the behaviour of a servo drive controlled by PI and FL speed controllers. A rotor flux oriented permanent magnet synchronous machine is simulated under varying operating conditions. The goal of the investigation is to attempt to provide a detailed comparison regarding drive operation with constant parameter type PI and FL speed control. It is believed that the general conclusion regarding PI and FL controller performance should be based on the overall performance of the drive rather than on a few operating points. The comparative study is divided into two parts:

1. The first part of the comparison is accomplished to provide a general comparison in terms of rise time, settling time and robustness of the drive. The comparison is based on selected transients in the entire speed range. The objective of this comparison is to show how a selective approach to choice of transients that are used to underpin certain desirable conclusion can be misleading. In addition, a different set of controller parameters can lead to different behaviour of the drive. This has been considered in the comparison by introducing two initial designs of the speed controllers. Results are given in Section 5.4.

2. The second part of the comparison provides a detailed comparison of the behaviour of the drive over the entire speed range. Two initial designs of speed controllers are considered in this investigation. At first, a detailed comparison is made based on a graphical approach. The data are compiled from many simulation runs and the comparison is made by analysing the graphs in terms of overshoot and settling time versus speed, undershoot and restoration time versus speed and undershoot and settling time versus speed. Results are presented in Section 5.5. Alternatively, the presentation of the overall performance of the variable speed drive can be based on one of the integral error criteria (selection of the criterion is

discussed in Section 5.2.2). This method is used to compare the drive behaviour in Section 5.6. Most of the original results of the research, described in this Chapter, are reported in [Ibrahim and Leví, 1998] and [Ibrahim *et al*, 1998a].

5.2 PERFORMANCE INDICES OF HIGH PERFORMANCE VARIABLE SPEED DRIVES

In a high performance drive, used for example in a robot, a maximum transient torque is to be maintained in order to provide rapid acceleration and deceleration in either direction of rotation. Quick speed response enables the robot to move to the next position in a short amount of time. The response time is obviously related to the robot's motion speed. In robotics, stability is generally defined as a measure of the oscillations which occur in the arm during movement from one position to the next. A robot with good stability will exhibit little or no oscillations either during or at the termination of the arm movement. Poor stability would be indicated by a large amount of oscillation. It is generally desirable in control system design to have good stability and fast response time. Unfortunately, these are competing objectives. The performance of a modern AC drive is related to energy conversion indices, quickness of torque, speed and position responses, precision and range, response robustness to parameter de-tuning, inertia and load torque perturbations, reliability, cost, and specific weight. Performance indices can be used to evaluate the performance of a control system. In general, performance indices may be classified into three main categories [Davies, 1970]. The first of these categories defines those indices which shape the transient response characteristic of the system under investigation, such as rise time, percentage overshoot and settling time. The second category includes those performance indices which may also be termed as error criteria. This type defines an optimum approximation of a desired response, where the errors in the approximation are weighted according to the particular performance index. In this case, the optimum approximation occurs when the weighted error function is at a minimum, which simultaneously guarantees an optimum transient response characteristic for the system. The third of these categories includes the more general performance indices, usually based on some economic index or cost. These indices are often functions of the energy involved in obtaining acceptable control as well as the desired results. This category will not be discussed here. On the other hand, in servo motor application, robustness of the drive to load application and motor inertia variation is also considered in the evaluation of the high performance drive. Performance indices related to robustness can be used to indicate the capability of the servo drive to compensate the variation of external parameters and can be regarded as the fourth category.

5.2.1 Quickness of speed controller response

The speed of response depends on the peak transient torque/inertia ratio (Nm/kgm^2) and on torque build-up time (Nm/ms). In a real situation, the total inertia is always larger than the rated motor inertia, while the design of a speed controller is normally based on motor rated parameters. As a result, the speed response will be degraded because the total inertia is different than the one used in the design case. A quick speed response is required in advanced industrial robot applications. The shaft of the motor is normally coupled to a linear guide, rotary actuator or gear to enable the machine to work as required. The speed, of course, determines how quickly the robot can accomplish a given work cycle. It is generally desirable in a production phase to minimise the cycle time of a given task. The weight of the object moved also influences the operational speed. Heavier objects mean greater inertia and momentum, and the robot must be operated more slowly to safely deal with these factors. For example, in a pick and place insertion machine, the robot arm has to be moved at least along 3 to 6 axes to complete one job cycle within a few seconds. One movement requires at least a trapezoidal velocity profile which is composed of acceleration, constant velocity, and deceleration [Klafter, 1989]. Because of acceleration and deceleration problems, a robot is capable of travelling one long distance in less time than a sequence of short distances whose sum is equal to the long distance. The short distances may not permit the robot to ever reach the programmed operating speed [Groover *et al*, 1986]. In addition, overshoot is not permissible in the response to step input. In insertion operations, for instance, overshoot would cause the manipulator hand to travel beyond the required location. In order to improve the acceleration time, the speed controller should be tuned to attain as quickly as possible the *q*-axis current command limit. The relationships between the electrical torque, rotor speed and motor inertia are given in equations (3.25) and (3.30). The acceleration time is defined as the time required to reach the nominal speed from standstill. In vector control the peak transient torque may be built in 2 to 5 ms at standstill, with the flux already present in the machine [Boldea and Nasar, 1992].

5.2.2 Integral speed error criteria

The second performance index that can be used in the comparison is an integral speed error criterion. The precision of speed control can be expressed in terms of the variations within a speed range. The speed control precision may be 2 to 3 per cent at the rated speed in low precision drives. In such a case, the speed may not be measured but estimated from the machine voltages, currents, and known parameters of the machine via real time calculations. In high precision drives, at the maximum speed, the speed control error is 0.1% and goes up to 10% to 20% below 1 rpm [Boldea and Nasar, 1992]. The integral performance indices are, in principle, a function of the difference between the desired system response (i.e. a step speed command) and the actual rotor speed. If the speed command is denoted by $\omega^*(t)$, and the actual rotor speed by $\omega(t)$, then the speed error response $e(t)$ is defined as $e(t)=\omega^*(t)-\omega\,(t)$. The speed error is integrated in a certain way to provide a measure for the overall system behaviour. There are four types of integral equations which can be used to study the speed error of the drive [Davies,1970], [Vas, 1998]:

1. Integral squared error method (*ISE*)

   This is the simplest performance index, which is based on the following integral equation:

$$E_1 = \underset{t\to\infty}{Lim}\frac{1}{2T}\int_{-T}^{T} e^2(t)dt \tag{5.1}$$

For deterministic or transient input signals with $e(t) = 0$ for t < 0 it can be written as

$$E_2 = \int_0^\infty e^2(t)dt \tag{5.2}$$

This equations has one severe limitation however, in that it requires a steady-state error of zero, i.e.

$$\underset{t\to\infty}{Lim}\ e(t) = 0. \tag{5.3}$$

This condition also applies to all the methods that will be discussed next. However, if the condition is not satisfied, the equation loses its meaning, unless the integral is truncated.

2. The second performance index is defined as (*ITSE*) [Davies,1970]

$$E_3 = \int_0^\infty t\, e^2(t)dt. \tag{5.4}$$

This index has the advantage that large initial errors (e.g after step inputs) do not give rise to large contributions in the value of E_3. This characteristic is very desirable especially for servo systems where the initial error is unavoidable due to the inertia of the system.

3. 'Integral of the absolute value of the error' method (*IAE*)

The modulus of the error, used in this criterion, may be quite conveniently handled by computer. It is therefore suitable for simulation studies although it is normally quite difficult to deal with analytically. The integral error criterion is given as:

$$E_4 = \int_0^\infty |e(t)|dt \tag{5.5}$$

4. Integral of time multiplied by absolute error (*ITAE*)

An improvement to the E_4 index can be obtained by weighting the absolute error with time so that

$$E_5 = \int_0^\infty t\left|e(t)\right|dt \tag{5.6}$$

This index reduces the contribution of large initial errors to the value of the integral. Based on equation (5.6), this criterion can be used to characterise the overall drive performance, which is composed of the steady-state performance in terms of accuracy and the transient performance in terms of the quickness of the speed response of the system. ITAE is used for comparative evaluation of the drive performance in Section 5.6. Both IAE and ITAE have already been used in Section 4.5.1.

5.2.3 Control robustness

The third parameter that can be used to measure the performance of an AC drive is robustness. The robustness, within the scope of this research, is related to the sensitivity of the speed response to motor inertia and load torque variations. These parameters have significant influence on the overall performance of the AC drive. The speed controller is designed by assuming that the total inertia (reflected back to the motor shaft) is a constant and does not vary with time. This assumption is valid for a large number of applications. There are, however, many exceptions, such as for example a robot joint. In fact, the reflected inertia of most of the axes of a robot will normally fluctuate significantly while the manipulator is moving. An example of such behaviour can be seen in [Klafter, 1989], where the degree of inertia variation for each of the six joints of a robot arm is approximately 4.5 times under no load conditions and can be as much as 7 times when the manipulator is carrying a 2 kg load. The robustness performance index can be used to evaluate the capability of the speed controller to compensate for the large changes in inertia. All these variations affect the closed-loop response in a fixed structure, fixed parameter controller. Increasing the gain of the controller results in better robustness of the response (lower sensitivity to parameter and load variations), but disturbs the behaviour in the region of the rated torque, speed or position. In many cases, an estimator is used to estimate the inertia and load variation. Therefore the structure of the control system becomes more complex and difficult to implement. There is an advantage if the speed controller has some robustness capability to compensate for the variation in parameters. It is not very economic and practical if a specific controller is designed for a specific type of motor and load or the controller parameters have to be re-tuned to obtain the required response each time when something changes. Therefore, the speed controller robustness should be considered in the design criteria.

5.3 DESCRIPTION OF THE SIMULATION PROCEDURE

The comparison of the simulation results for a SPMSM drive applies to the use of PI and FL speed controller. A detailed description of the design of the speed PI controller and the FL speed controller is given in Chapter 3 and 4, respectively. The basic configuration of a rotor flux oriented SPMSM drive with hysteresis current control is given in Chapter 3 (Figs. 3.3 and 3.9) and the relevant data of the machine are once more those of Appendix A. The PWM voltage source inverter is controlled by means of three independent hysteresis current controllers. The hysteresis band is adjusted to ± 0.5 A (i.e., ± 5.7% of the rated current), the computation time step and the sampling time are chosen as 20 μs and all these settings are kept at these values in all the simulations. The inverter input DC voltage is set to the constant value of 220 V, so that sufficient voltage reserve is provided for operation with good current control at rated speed and maximum torque. Stator q-axis current is limited to ± 30 A in accordance with the maximum allowed stator current rms value.

Both speed controller types are initially tuned to yield essentially an identical speed response to the application of the step rated speed command (180 rad/s) under no load conditions, assuming rated combined inertia of the motor and the load (the inertia J_n value is given in Appendix A). It is believed that only such an approach allows a fair comparison to be made. Two initial controller designs are considered. The criterion for the first one is essentially aperiodic speed response (overshoot less than 0.1 rad/s), while in the second case a small overshoot of 0.72% of the rated speed (1.3 rad/s) is allowed. In both cases the minimum settling time is required, while taking into account the limit on the stator current maximum value. Studies of numerous transients are then undertaken by simulation. The investigations include: response to the large speed command (long operation in limit of stator *q*-axis current command), medium speed command (some operation in the limit of stator *q*-axis current required), and small step speed command (operation in the limit of stator *q*-axis current only short-term or not required), application of the step rated load torque (disturbance rejection), a small 10% step decrease in speed command and operation with inertia other than rated.

The test sequence is, unless otherwise indicated, as follows:

1) Rated step speed command is applied at zero time under no load conditions.
2) Rated load torque is applied at t = 0.025 sec.
3) Speed command is reduced to 0.9 of the previous setting at t = 0.08 sec.

5.4 COMPARATIVE STUDY BASED ON LIMITED SELECTION OF RESPONSES

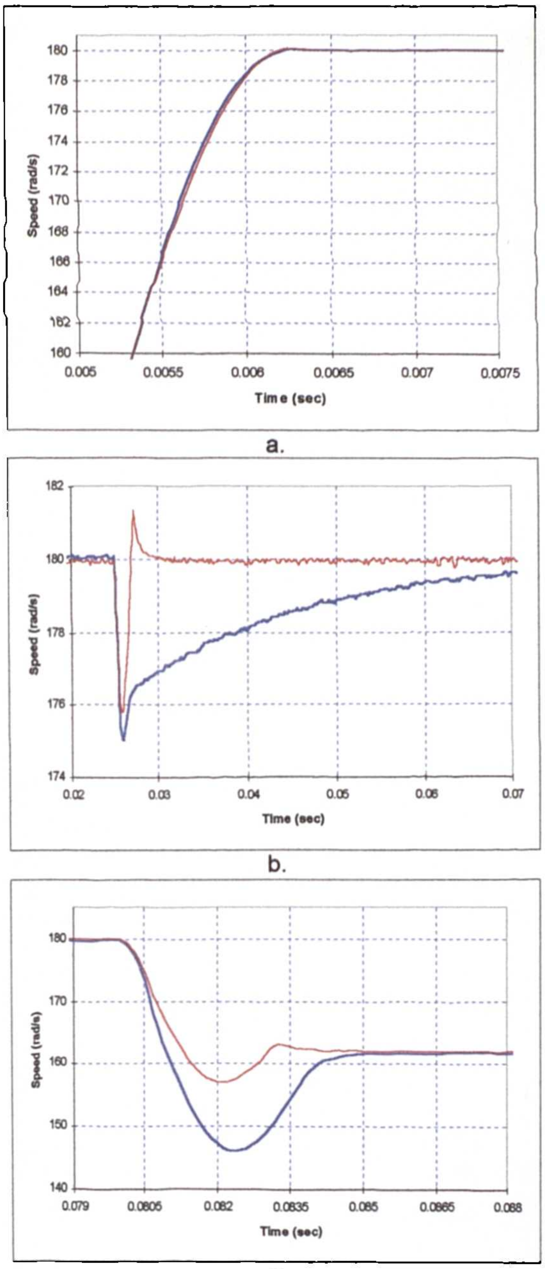
Figures 3.13 and 4.13 show a sample of simulation results for the operation with PI and FL speed controllers, respectively. It is difficult to draw any viable conclusions from traces of these figures regarding comparison of the drive behaviour under PI and FL speed control. Comparison of the drive behaviour under PI and FL speed control is therefore performed by overlapping and zooming speed responses, leading to the results of the type shown in Figs. 5.1 and 5.2. The same approach is utilised further in displaying simulation results that apply to other operating conditions.

PI controller parameters are K_p = 2.22, K_i = 111 for zero overshoot design and K_p= 2.4, K_i = 1235.2 for 1.3 rad/s overshoot design (Section 3.8.1). Scaling factors of the FL speed controller are G_e = 0.0023, G_{ce} = 0.41, G_u= 3 for zero overshoot design and G_e = 0.0025, G_{ce} = 0.43, G_u = 2.5 for 1.3 rad/s overshoot design, while the structure is the one of Section 4.4.2.3. It therefore follows that comparison is based on so-called off-line optimised PI speed controller and the off-line optimised FL speed controller.

5.4.1 'Design' case

Figures 5.1 and 5.2 show zoomed speed responses to the three transients depicted earlier in Figs. 3.13 and 4.13. In all the figures, the bold trace (blue colour) represents the response obtained with PI speed control. Response of the PI control to step speed command of 180 rad/s results in no overshoot, while FL control yields an overshoot of less than 0.1 rad/s as shown in Fig. 5.1. Rise and settling times of the two controllers are effectively the same. The two responses are believed to be close enough, so that a fair comparison is possible for other operating regimes.

A similar study is reported in Fig. 5.2, where the results apply to the speed controller design with 1.3 rad/s overshoot. The responses of both PI and FL controllers give approximately the same rise time as for zero overshoot design, except that the PI control has a slightly longer settling time than the FL controller, Fig. 5.2a. The disturbance rejection comparison in Fig. 5.1b shows that the initial dip in speed with PI control is large. The PI control response is aperiodic and the FL controller restores operation to the rated speed command much more quickly. The load rejection capability (Fig. 5.1b) of the FL controller is therefore far superior. However, in the design with 1.3 rad/s overshoot, the load rejection capability of the PI controller is significantly improved, so that restoration time is almost the same for FL and PI speed control. The dip in speed response is however 1 rad/s larger with PI controller. This is due to the fact that gains of the PI speed controller in 1.3 rad/s overshoot design are slightly bigger than for aperiodic response design. Figures 5.1c and 5.2c show that the nature of response to the speed command change from rated to 0.9 times rated speed is similar with both PI and FL controller. However, the speed undershoot with FL control is significantly smaller for zero overshoot design, although the time needed to achieve the new steady-state is approximately the same for both controllers.

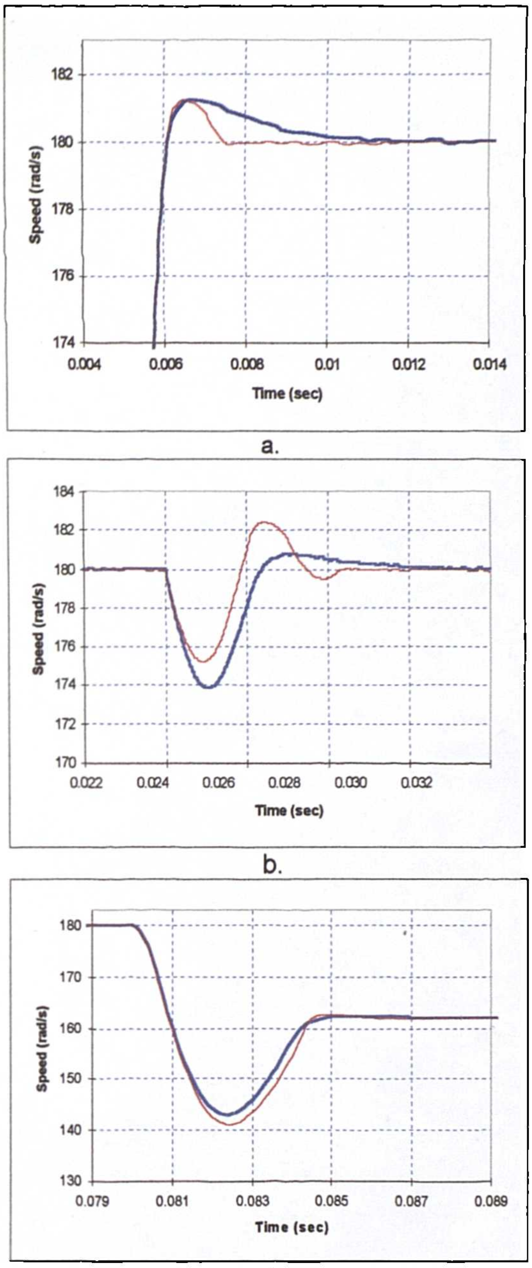


a.

b.

c.

Figure 5.1: Comparison of response obtained with PI and FL speed controllers (zoom extracts from Fig. 3.13a and Fig. 4.13a for zero overshoot design: a) response to rated step speed command; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times rated. (🟦 for PI, 🟥 for FL)



a.

b.

c.

Figure 5.2: Comparison of response obtained with PI and FL speed controllers (zoom extracts from Fig. 3.13b and Fig. 4.13b for 1.3 rad/s overshoot design: a) response to rated step speed command; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times rated.

5.4.2 Initial step speed command other than rated

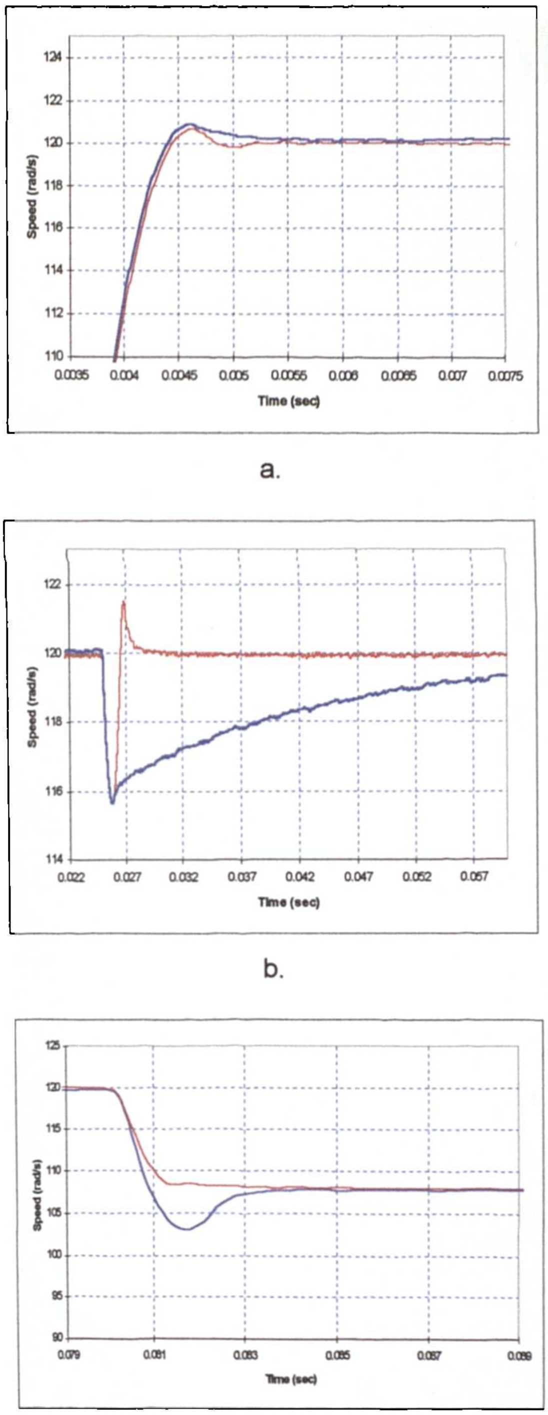
For the purpose of the further investigation, initial step speed commands are classified into three groups. They are: large, medium and small initial step speed command. The large initial step speed command is specified in the range of 180 rad/s to 120 rad/s. The medium initial speed command is defined as 120rad/s to 60rad/s and the small initial speed command is defined from 60rad/s to standstill.

5.4.2.1 Medium initial speed command

The response to step changes in speed command, subsequent rated load torque application and step-wise reduction of speed command to 0.9 times the previous value is investigated next for two arbitrarily selected speed commands, i.e. two thirds of the rated speed (120 rad/s) and one half of the rated speed (90 rad/s). A comparison of the results for these two cases is illustrated in Figs. 5.3, 5.4, 5.5 and 5.6.

Response of both PI and FL controllers to a step speed command application of 120 rad/s results in small overshoot of about 0.8 rad/s, as shown in Fig. 5.3a. Rise and settling times of the two controllers are still the same for both controllers. A similar type of transient is presented in Fig. 5.4 for the 1.3 rad/s overshoot design. The response of both controllers has approximately the same rise time, as for the zero overshoot design, while the PI controller has a longer settling time than the FL controller. The overshoot of the speed response for PI and FL controllers is increased from 1.3 rad/s for rated speed command to 2 rad/s for PI and 1.8 rad/s for FL control.

Transients for rated load torque application are depicted in Fig. 5.3b. The dip in speed response is the same for both PI and FL control. The excellent load rejection capability of the FL controller is still retained as in the design case. Nevertheless, in the design for 1.3 rad/s overshoot, the load rejection capability of the PI controller is significantly improved and is characterised with shorter restoration time, that is of the same order as that of the FL control (Fig. 5.4b). The figure clearly shows that the lack of capability of load rejection of PI control can be improved by introducing a slightly higher gain.



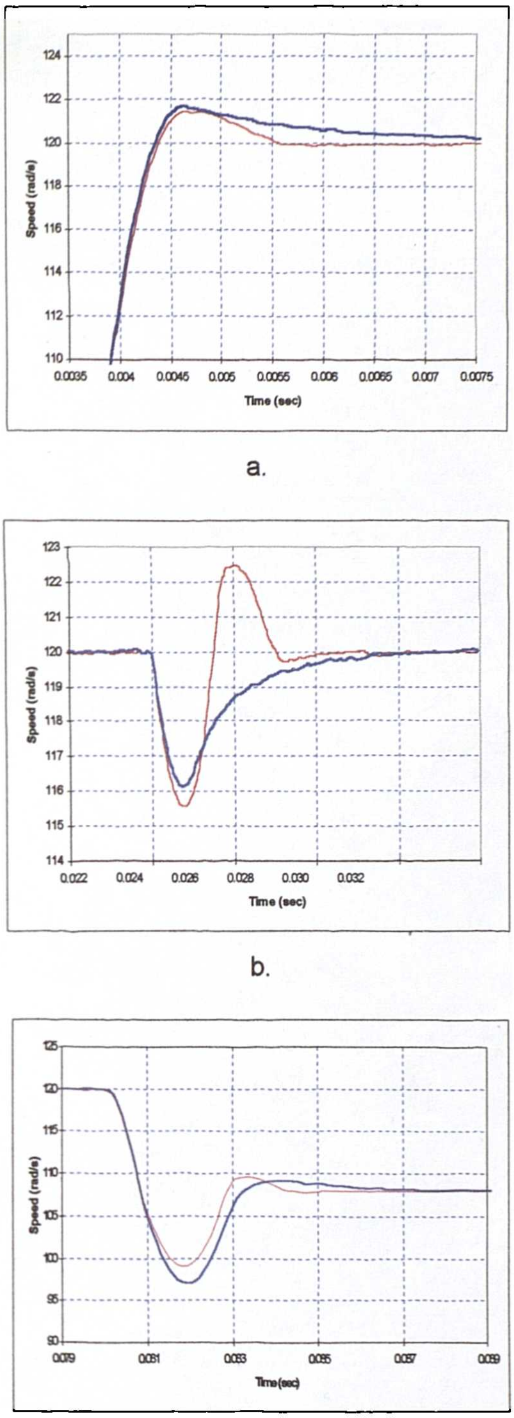
a.

b.

c.

Figure 5.3: Comparison of response obtained with PI and FL speed controllers for zero overshoot design: a) response to step speed command of 120 rad/s; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times 120 rad/s.

(🟦 for PI, 🟥 for FL)



a.

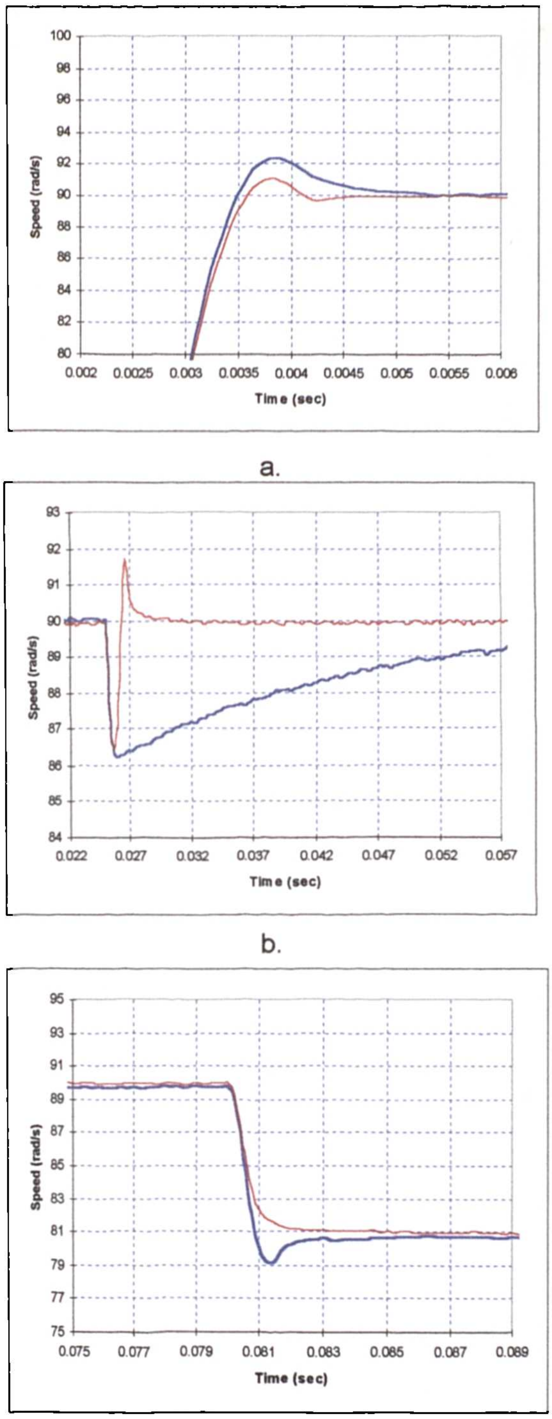
b.

c.

Figure 5.4: Comparison of response obtained with PI and FL speed controllers for 1.3 rad/s overshoot design: a) response to step speed command of 120 rad/s; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times120 rad/s.

In zero overshoot design, the speed command change from 120 rad/s to 0.9 times 120 rad/s leads to better speed response with FL control. PI control has a large undershoot in the speed response of about 5 rad/s (Fig. 5.3c). The settling times for the two control schemes are not much different. For design with 1.3 rad/s overshoot, the undershoot in speed response with FL control is significantly increased and is 8 rad/s. The same applies to the PI control which has 11 rad/s of undershoot in speed response (Fig. 5.4c). The time needed to achieve the new steady-state is shorter with FL control as compared to PI control. There is, however, some deterioration as compared to zero overshoot design. The PI control scheme has shown great improvement in load rejection but at the same time the response to the initial large step speed command and to small step change of speed command is significantly worse in terms of settling time, when compared to zero overshoot design. FL control has therefore demonstrated better capability particularly in load rejection.

When the initial step speed command is 90 rad/s (Figs. 5.5 and 5.6), both control schemes show an increase in overshoot of the speed response. The settling time, for design with 1.3 rad/s overshoot, increases compared to zero overshoot design. Overshoot of PI control has now increased to 3.2 rad/s, while it is 2 rad/s for FL control (Fig. 5.5a, 5.6a). Undershoot in speed response caused by load rejection is approximately the same for both designs. In design with 1.3 rad/s overshoot, the good performance of PI control due to step load change is maintained. A similar control performance of PI control in terms of load rejection as at 120 rad/s step speed command is observed (Fig. 5.6b). In small step down speed command operation for 1.3 rad/s overshoot design, both controllers produce large undershoot in speed but with shorter settling time to achieve the new steady-state (Fig. 5.6c).

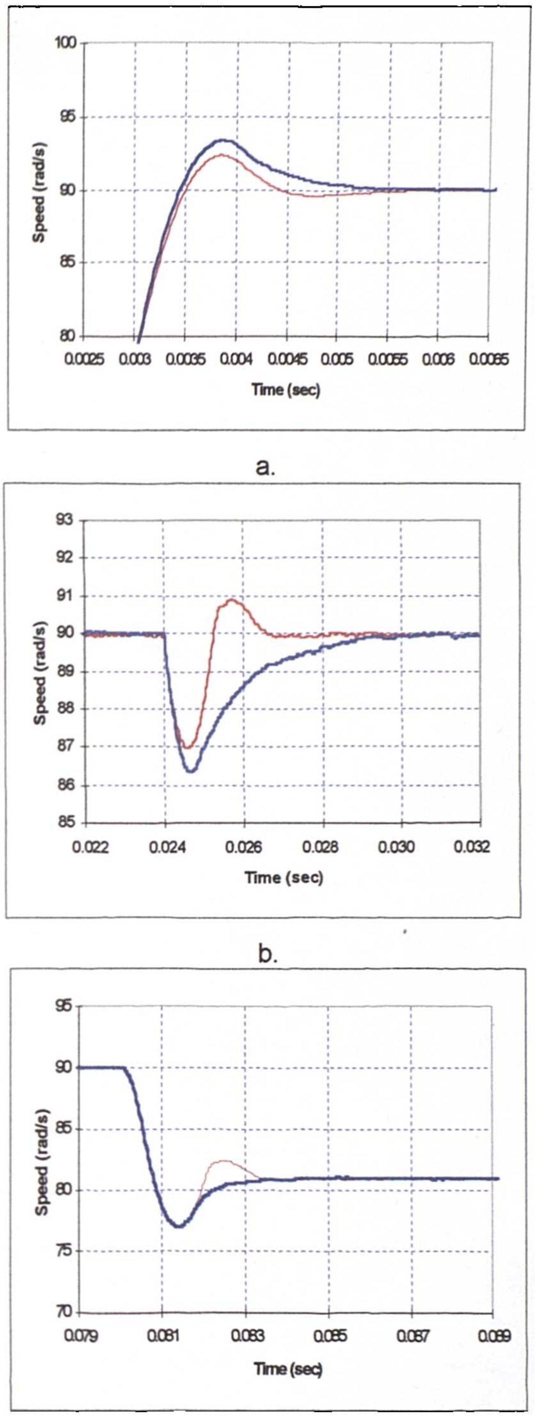


a.

b.

c.

Figure 5.5: Comparison of response obtained with PI and FL speed controllers for zero overshoot design: a) response to step speed command of 90 rad/s; b) response to step rated torque application; c.) and response to change of speed command from rated to 0.9 times 90 rad/s. (🟦 for PI, 🟥 for FL)



a.

b.

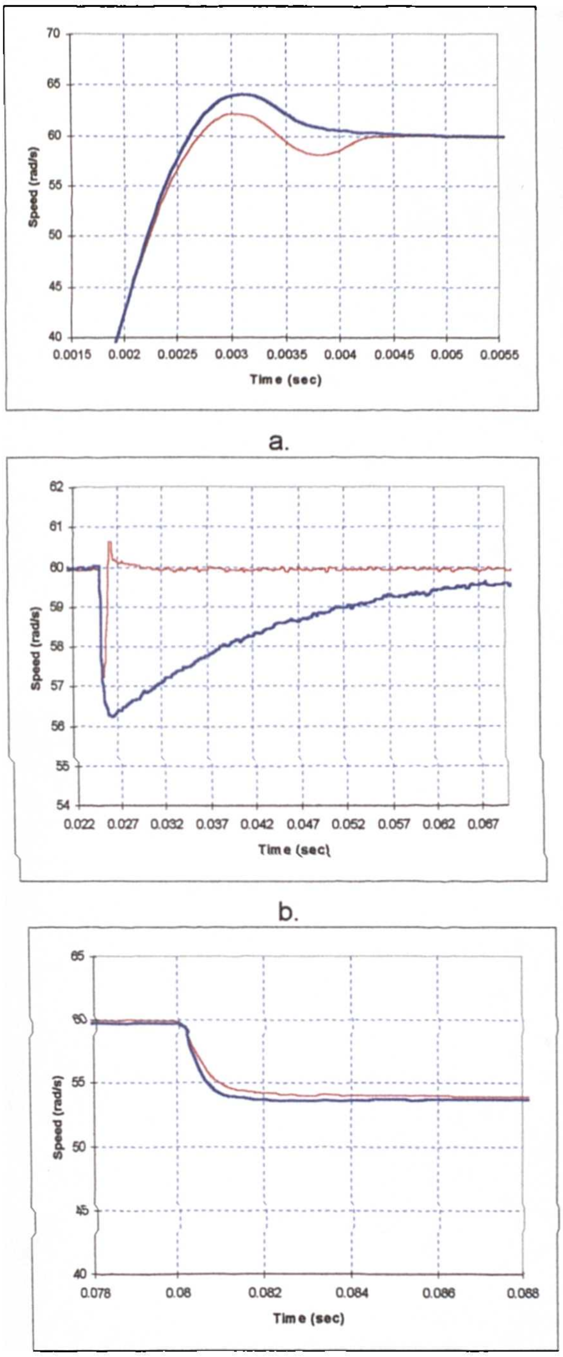
c.

Figure 5.6: Comparison of response obtained with PI and FL speed controllers for 1.3 rad/s overshoot design: a) response to step speed command of 90 rad/s; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times90 rad/s.

5.4.2.2 Small initial speed command

The response to small step change in speed command, subsequent rated load torque application and step-wise reduction of speed command to 0.9 times the previous value, is investigated next for two arbitrarily selected speed commands: one third of the rated speed (60 rad/s) and one sixth of the rated speed (30 rad/s). A comparison of the results for these two cases is illustrated in Fig. 5.7 to Fig. 5.10. When the step speed command is reduced, a large overshoot appears in the speed response of both controllers. For a speed command of 60 rad/s, the PI controller generates a larger overshoot, while the settling time in both cases is again effectively the same (Fig. 5.7a and Fig. 5.8a). However, at a command speed of 30 rad/s, the response of the FL controller is highly oscillatory with a significantly larger overshoot and long settling time (Figs. 5.9a and 5.10a). Thus it appears that the response of the PI controller, for low speed command settings, is far better. The dip in speed due to load application remains similar in both cases (Fig. 5.8b and 5.10b), except for zero overshoot design, where the FL controllers has smaller dip in speed response (Figs. 5.7b and 5.9b). These figures thus confirm the superiority of the FL controller with respect to disturbance rejection in terms of quicker restoration time (Figs. 5.7b, 5.8b, 5.9b and 5.10b).

Response to a small speed command change is now better in zero overshoot design with PI controller (Figs. 5.7c anf 5.9c). Hence, PI control results in better overall response. Indeed, the time needed to achieve a new steady-state is significantly shorter with PI control in Fig. 5.7c with no undershoot. Similar conclusions apply to the case illustrated in Fig. 5.9c, where a small command speed change is initiated at t = 0.12 s. With regard to 1.3 rad/s design (Figs. 5.8c and 5.10c) FL controller leads to a larger undershoot but the settling time is shorter.

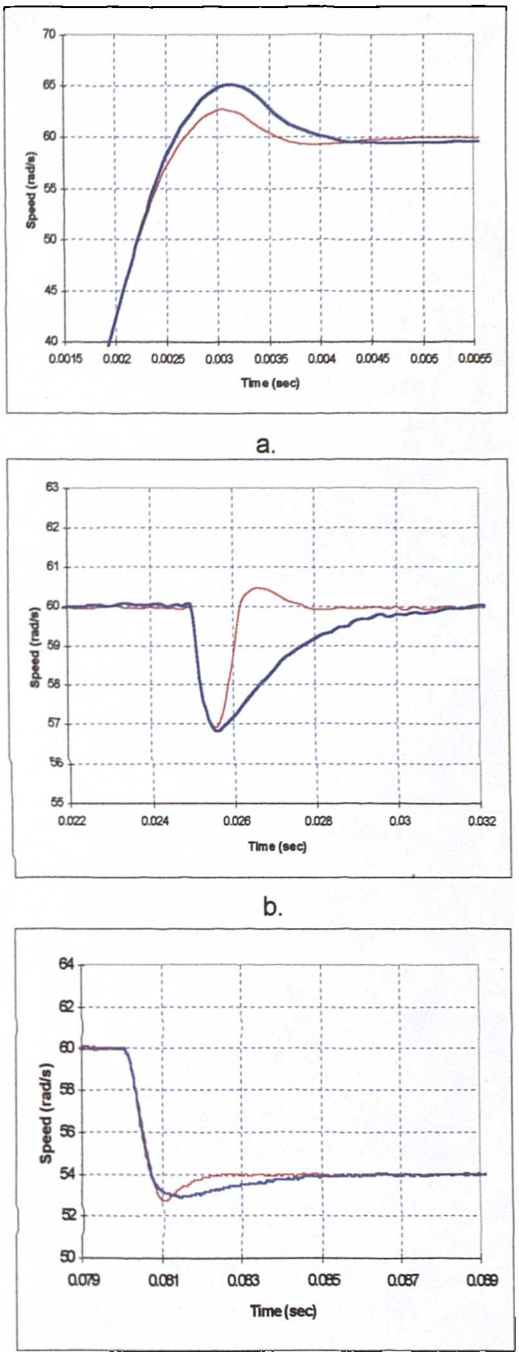


a.

b.

c.

Figure 5.7: Comparison of response obtained with PI and FL speed controllers for zero overshoot design: a) response to step speed command of 60 rad/s; b) response to step rated torque application; c) and response to change of speed command from rated to 0.9 times 60 rad/s. (🟦 for PI, 🟥 for FL)

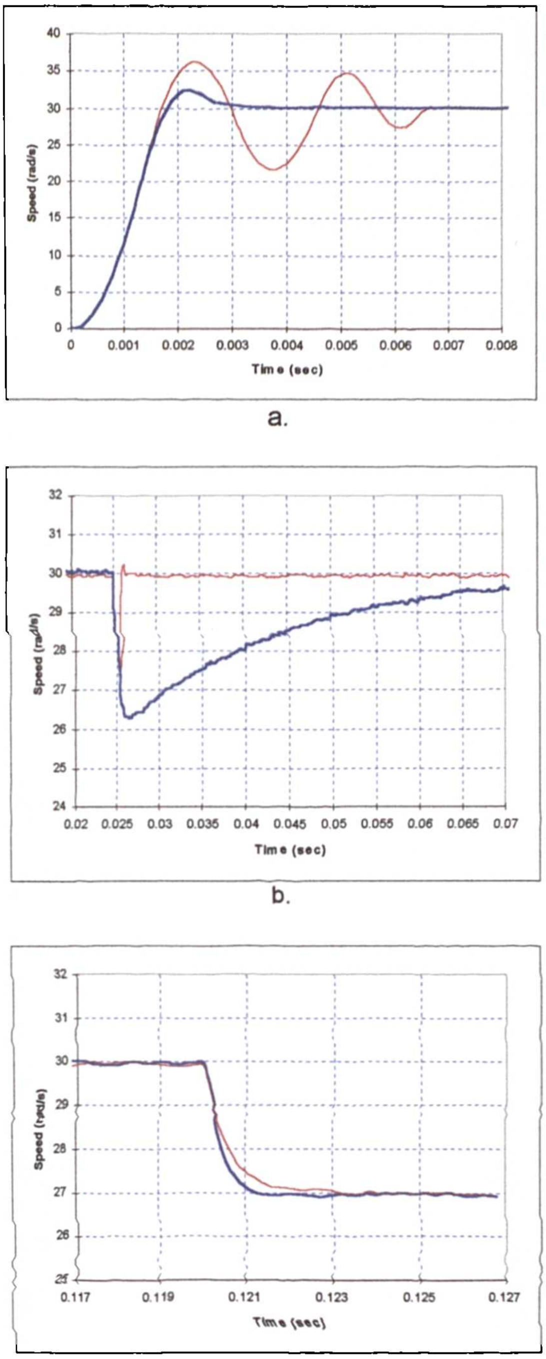


a.

b.

c.

Figure 5.8: Comparison of response obtained with PI and FL speed controllers for 1.3 rad/s overshoot design: a) response to step speed command of 60 rad/s; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times 60 rad/s.

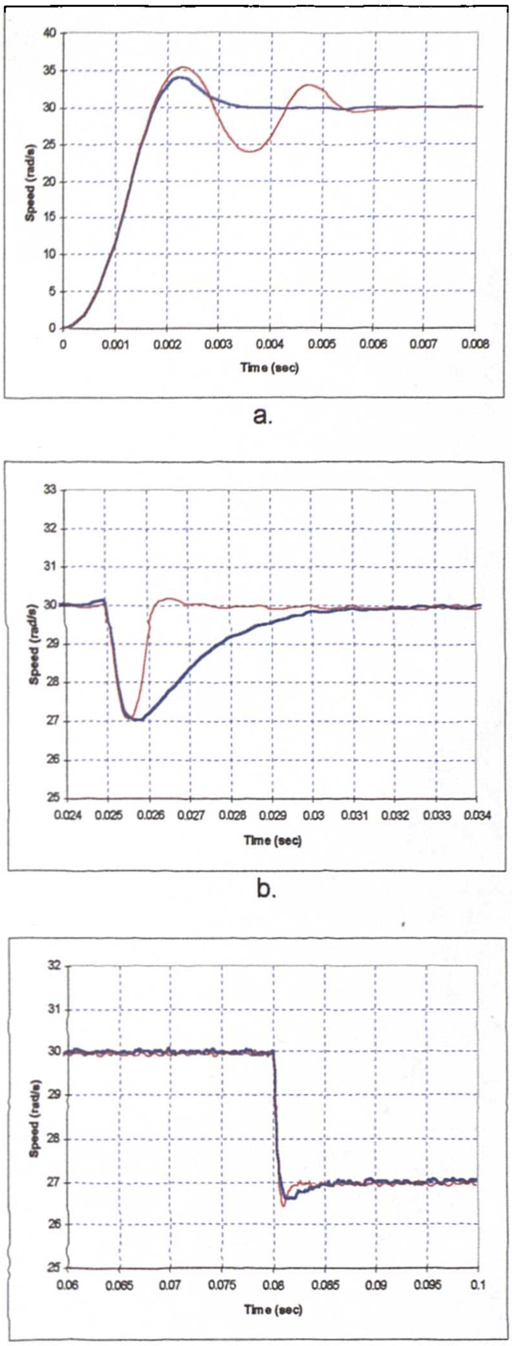


a.

b.

c.

Figure 5.9: Comparison of response obtained with PI and FL speed controllers for zero overshoot design: a) response to step speed command of 30 rad/s; b) response to step rated torque application; c) and response to change of speed command from rated to 0.9 times 30 rad/s. (🟦for PI, 🟥for FL)



a.

b.

c.

Figure 5.10: Comparison of response obtained with PI and FL speed controllers for 1.3 rad/s overshoot design: a) response to step speed command of 30 rad/s; b) response to step rated torque application; c) response to change of speed command from rated to 0.9 times 30 rad/s.

The results reported in Fig. 5.1 through Fig. 5.10 show that, in a certain region the PI control appears to be better than FL control, while in other regions both controllers have the same performance. Indeed, FL control gives superior response to initial speed command application - Figs. 5.3a, 5.5a, 5.7a (zero overshoot design) and Figs. 5.2a, 5.4a, 5.6a, 5.8a (1.3 rad/s overshoot design), load torque application - Figs. 5.1b, 5.3b, 5.5b, 5.7b, 5.9b (zero overshoot design), Figs. 5.2b, 5.6b, 5.8b, 5.10b (1.3 rad/s overshoot design) and small change of speed command - Figs. 5.1c, 5.3c, 5.5c (zero overshoot design) and Figs. 5.4c, 5.8c (1.3 rad/s overshoot design). In the other transients, illustrated in Figs. 5.9a, 5.10a (initial step speed command), load torque application (Fig. 5.4b) and small change in speed command (Figs. 5.2c, 5.6c, 5.7c, 5.9c) PI control yields a better speed response. However, if the conclusion is made on the basis of say, Figs. 5.3, 5.4, 5.5, and 5.8, then the FL control is to be preferred to PI control. Such a conclusion would however be quite erroneous and the previous analysis indicates how misleading improper selection of transients for comparison can be. It is shown that such unclear conclusion can easily be arrived at, if a certain selected transient is utilised to underpin the intention of the comparison. In other words, the transients of small speed command can be used if the intention of the comparison is to show the goodness of the PI control scheme, while the transients of medium and large speed command can be utilised in order to show the superiority of the FL control scheme. In reality, this is not an accurate conclusion to arrive at in describing the overall performance of the SPMSM drive. The overall situation, across the entire speed control range, is much more complex. The detailed investigation of the entire behaviour of the drive for the whole speed range is therefore carried out and reported in the next section.

The results discussed in Sections 5.4.2.1 and 5.4.2.2 are summarised in Tables 5.1 and 5.2.

Zero overshoot design		
i) Transients for large step speed command	PI	FL
180 rad/s 0 rad/s t = 0		
rated torque application t = 0 t = 0.025s	−	+
180 rad/s t = 0.08s 162 rad/s	−	+
ii) Transients for medium step speed command	PI	FL
120 rad/s 0 rad/s t = 0	−	+
rated torque application t = 0 t = 0.025s	−	+
t = 0.08s 120 rad/s 108 rad/s	−	+
90 rad/s 0 rad/s	−	+
rated torque application t = 0 t = 0.025s	−	+
t = 0.08s 90 rad/s 81 rad/s	−	+
iii) Transients for small step speed command	PI	FL
60 rad/s t = 0 0 rad/s	−	+
rated torque application t = 0.025s	−	+
t = 0.08s 60 rad/s 54 rad/s	+	−
30 rad/s 0 rad/s t = 0	+	−
rated torque application t = 0.025s	−	+
t = 0.08s 30 rad/s 27 rad/s	+	−

− worse response

+ better response

□ the same response

Table 5.1: Performance of the PI and FL controllers for zero overshoot design.

1.3 rad/s overshoot design		
i) Transients for large step speed command	PI	FL
180 rad/s 0 rad/s t = 0		
rated torque application t = 0 t = 0.025s	−	+
180 rad/s t = 0.08s 162 rad/s	+	−
ii) Transients for medium step speed command	PI	FL
120 rad/s 0 rad/s t = 0	−	+
rated torque application t = 0 t = 0.025s	+	−
t = 0.08s 120 rad/s 108 rad/s	−	+
90 rad/s 0 rad/s	−	+
rated torque application t = 0 t = 0.025s	−	+
t = 0.08s 90 rad/s 81 rad/s	+	−
iii) Transients for small step speed command	PI	FL
60 rad/s 0 rad/s t = 0	−	+
rated torque application t = 0.025s	−	+
t = 0.08s 60 rad/s 54 rad/s	+	−
30 rad/s 0 rad/s t = 0	+	−
rated torque application t = 0.025s	−	+
t = 0.08s 30 rad/s 27 rad/s	+	−

− worse response   + better response

□ the same response

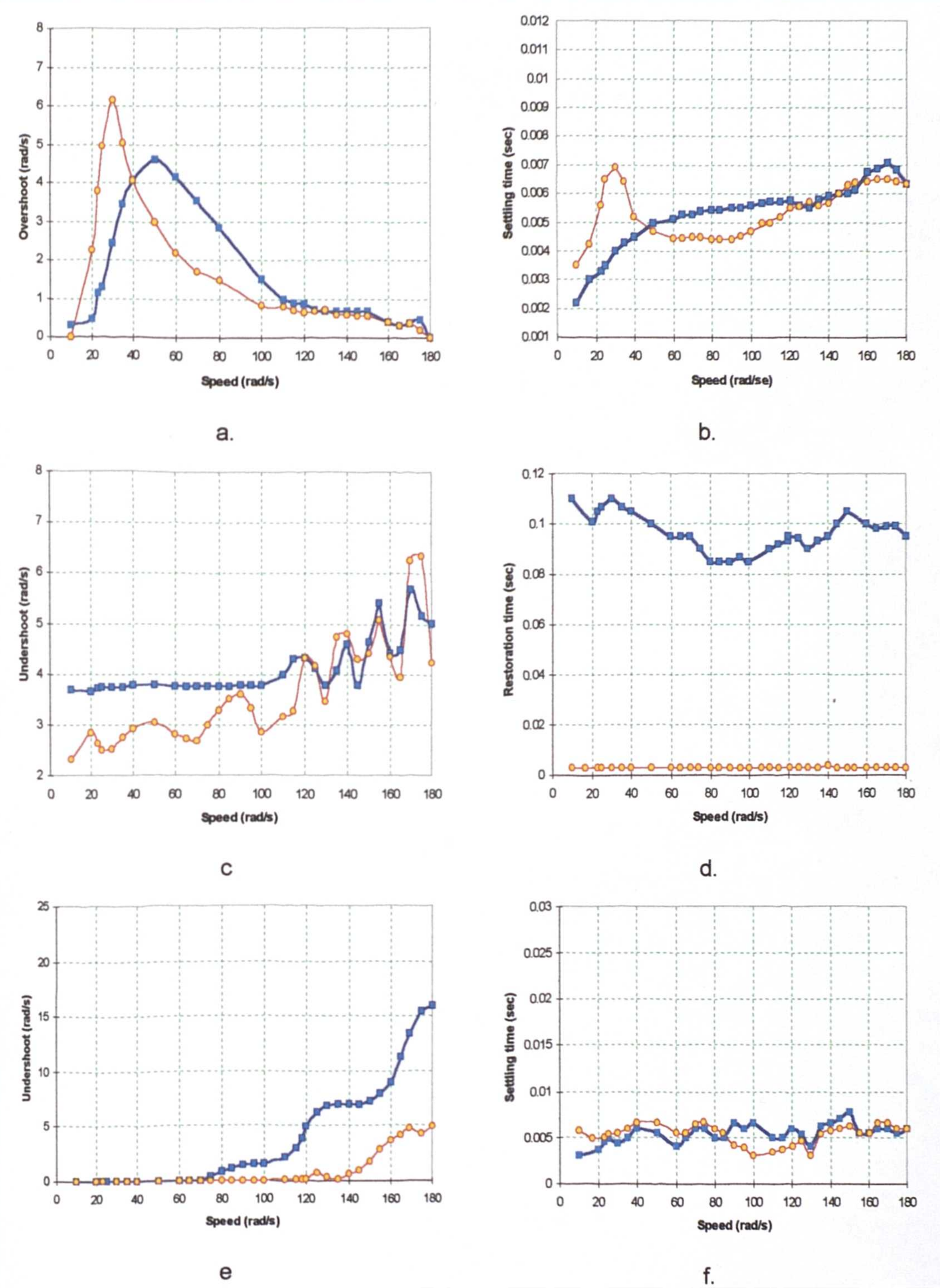
Table 5.2: Performance of the PI and FL controllers for 1.3 rad/s overshoot design.

5.5 A DETAILED COMPARISON OF PERFORMANCE OVER THE ENTIRE SPEED RANGE

The same type of study, reported in Section 5.4.2, is performed over the range of operating speeds from 10 rad/s for zero overshoot design and 20 rad/s for 1.3 rad/s overshoot design up to the rated speed (180 rad/s). The overshoot in speed response for a large speed command change, dip due to the load torque application and the undershoot that follows a small speed command change are measured for PI and FL control, together with the duration of the transient (which is taken as the time needed for the speed error to become smaller than 0.1 rad/s). Results are summarised in Fig. 5.11 for the zero overshoot design and in Fig. 5.12 for the 1.3 rad/s overshoot design. Every attempt was made to provide readings as accurate as possible in compiling data given in Figs. 5.11 and 5.12. This approach of graphical comparison provides a good insight into the overall performance of the servo drive controlled by PI and FL.

The response to the large step speed command change is basically the same with PI and FL control for zero overshoot design, Fig. 5.11a,b for speed commands between 120 and 180 rad/s. The FL control is superior between 50 and 120 rad/s, while the PI control is better below 50 rad/s. Disturbance rejection is better with the FL control in all the cases (Fig. 5.11c,d). The response to small step speed command change (Fig. 5.11e,f) is better with PI control for initial speeds up to 60 rad/s. For speeds from 60 to 180 rad/s the response is better with FL control.

The response to the large step speed command change, with 1.3 rad/s overshoot design (Fig. 5.12a, b), is better with PI control for speeds up to 35 rad/s; above this value, the FL control gives better response, especially in terms of settling time. Disturbance rejection of the PI control is now much better than for zero overshoot design and is similar as with FL control or even better for speeds from 120 to 180 rad/s (Fig. 5.12c,d). The FL control is superior below 120 rad/s in terms of restoration time, while undershoot is very much the same as with PI control. The response to small speed command change, Fig. 5.12e,f, is marginally better with either PI or FL control, depending on the initial speed.



a. b.

c d.

e f.

Figure 5.11: Comparison of PI and FL speed control over the entire speed region for zero overshoot design: a), b) speed overshoot and settling time for step application of large speed command; c), d) dip in speed and restoration time for rated load torque applications; e), f) speed undershoot and settling time for small step-wise change in speed command. (🟦 for PI, 🟥 for FL)

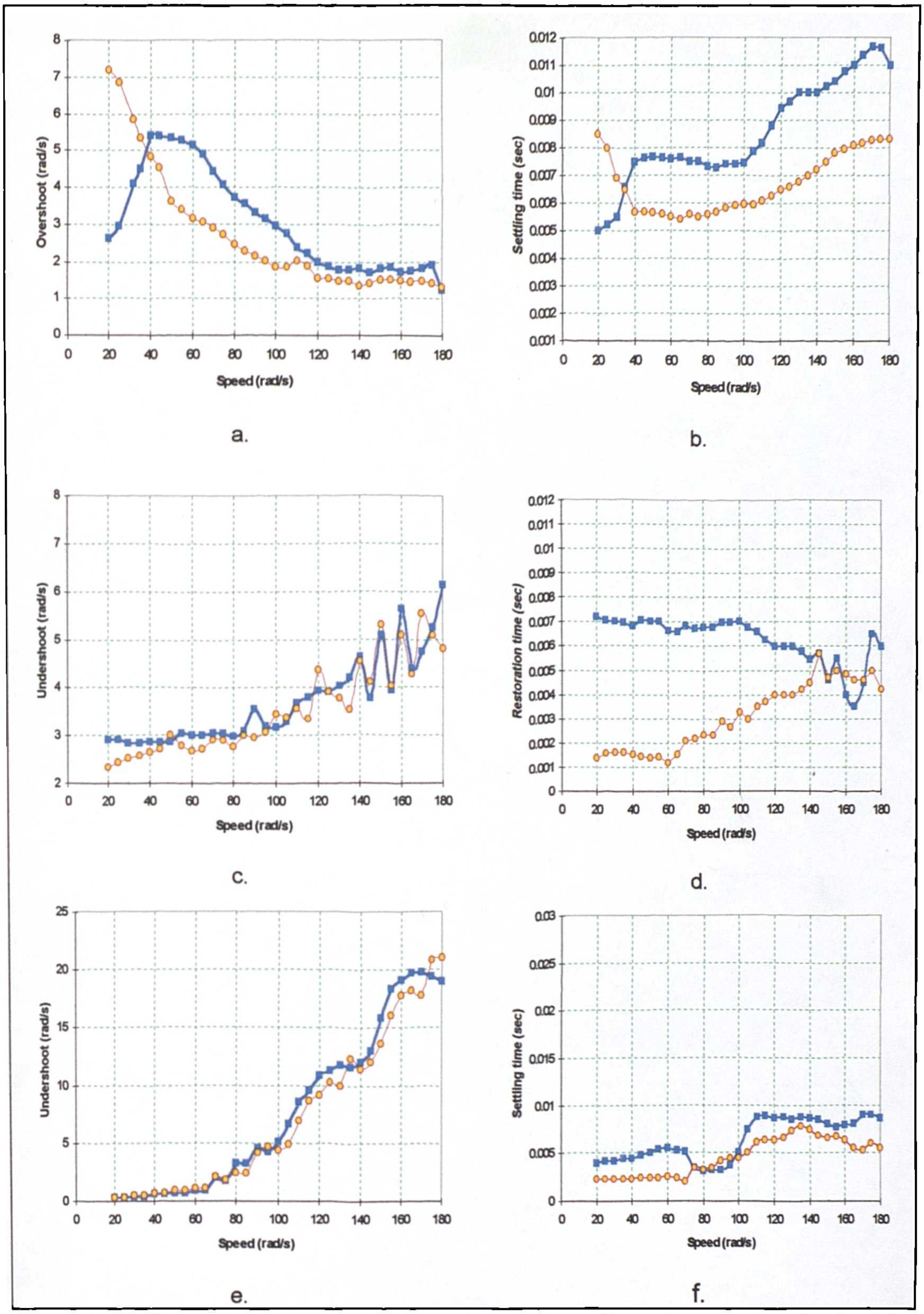


Figure 5.12: Comparison of PI and FL speed control over the entire speed region for 1.3 rad/s overshoot design: a), b) speed overshoot and settling time for step application of large speed command; c), d) dip in speed and restoration time for rated load torque applications; e), f) speed undershoot and settling time for small step-wise change in speed command. (🟦 for PI, 🟥 for FL)

Figs. 5.11 and 5.12 show that the speed response for zero overshoot design and 1.3 rad/s overshoot design has small overshoot when the operating points are near to the design case. However, when the operating point is far from the design case, the overshoot in speed becomes large for both control schemes. On the other hand in 1.3 rad/s overshoot design, both control schemes exhibit a long settling time in response to the step speed command. With regard to the step load torque application, the good performance of FL control is constantly maintained for both designs (zero and 1.3 rad/s overshoot design). For zero overshoot design only a set of fixed FL controller's gains (scaling factors) is required to enable the AC drive to operate satisfactorily in different operating points. There is no significant advantage of the FL control scheme when larger gains are utilised in 1.3 rad/s overshoot design. In addition, this causes large undershoot for small step-wise change in speed command (Fig. 5.11e, 5.12e). However, the advantage of increased gains exists for the PI control scheme, in terms of dip in speed and restoration time for load torque application (Fig. 5.11c,d and Fig. 5.12c,d). If the zero overshoot design of the PI control scheme is used in the implementation of the drive, different sets of gains should be used for load disturbance period and for speed command change, otherwise the performance of the PI controller will be degraded. In this case, the gain of PI controller must be higher than the gain used during the initial step speed command. On the other hand this value of gain cannot be applied for small step-wise change in speed command because large undershoot in speed and long settling time will occur (Figs. 5.11e and 5.12e). That means that another set of PI parameters is necessary. In reality, this is not easy to implement based on the conventional method. The FL control scheme has good performance for medium and large initial step speed commands, load rejection and small step reduction in speed command, without any requirement for different sets of controller gains. Further improvements in FL control scheme would be required to improve the response to a small step speed command. Detailed comparison of performance over the entire speed range for two initial designs shows that, in general, there are speed regions where PI control provides better response, as well as regions where FL and PI control yield more or less the same response. Further investigation and improvement is described in Chapter 6.

5.5.1 Variation of inertia

All the previous considerations apply to operation with rated inertia. Two tests incorporating an inertia change are conducted next. Off-line optimised PI and FL speed controllers, for zero overshoot design, are used. At first, it is assumed that the inertia is ten times higher than the one used for the controller design. The response to rated speed command and subsequent rated load torque application (at t = 0.2 s) is compared for this case in Fig. 5.13. As expected, operation of the FL speed controller is in this case superior. This is indeed the situation most frequently discussed in literature and used to underpin the conclusion that the operation of the drive with FL speed controller is far better than operation with PI controller. The PI speed controller yields higher speed overshoot with much longer settling time for the rated speed command and higher dip with longer restoration time for load torque application (Fig. 5.13b).

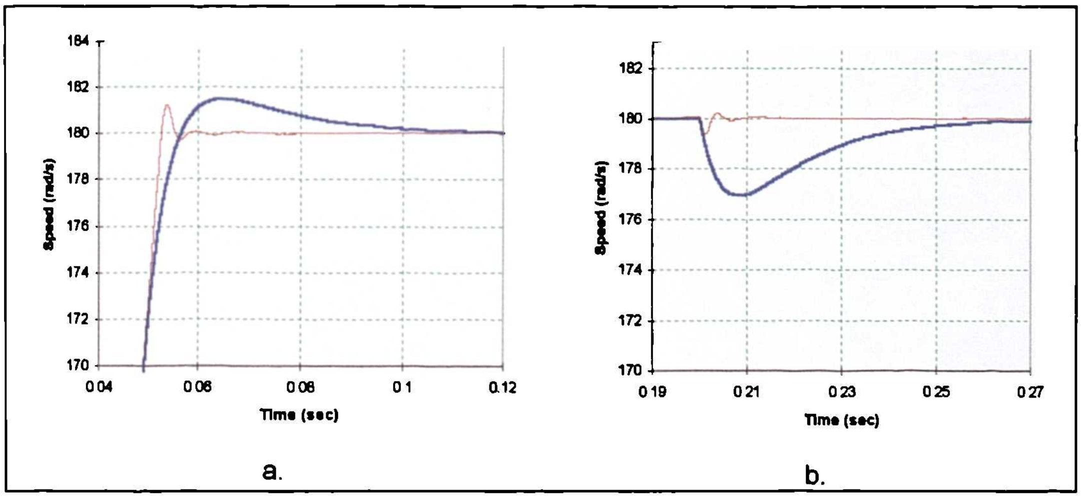


Figure 5.13: Comparison of responses obtained with PI and FL speed controllers for inertia ten times higher than in controller design: a) response to rated step speed command; b) response to step rated torque application. (🟦 for PI, 🟥 for FL)

The second test is the same as the previous one, except that now it is assumed that the inertia is only 0.3 times the value used in the controller design. Controller design normally relies on the combined inertia of the motor and load. Therefore, in the case of loads with variable inertia, the inertia can both reduce and increase with respect to the design value. Comparison of results for this decrease in system inertia is shown in Fig. 5.14. It is evident that now both controllers give poor, highly oscillatory response to rated speed command. However, the FL controller takes more time to establish steady-state operation. The dip in speed response to rated load torque application is the same for both controllers. The PI controller has a long restoration time, while the FL controller leads to oscillations in speed.

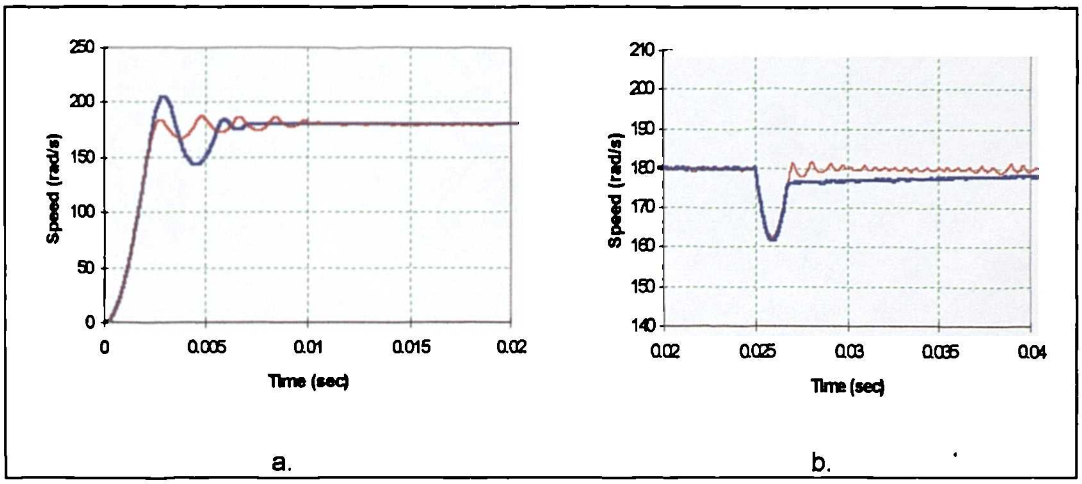


Figure 5.14: Comparison of responses obtained with PI and FL speed controllers for inertia 3.3 times lower than in controller design: a) response to rated step speed command; b) response to step rated torque application. (🟦 for PI, 🟥 for FL)

5.6 COMPARISON BASED ON INTEGRAL SPEED ERROR CRITERION

The desired performance index for a control system is often expressed graphically, or in the form of a desired response curve such as overshoot and undershoot of speed response. The limitation of this approach depends on the accuracy of the measurement and visual interpretation from plotted graphs. Therefore, integral time error (ITAE) method is used to present the overall performance of the variable speed drive in the numerical value form. In addition, this criterion has been chosen with the idea of making a global comparison of the system response, that includes responses to the step speed command, load rejection and step down speed command. However, the disadvantage of this approach is that the calculations are performed for a set of transients and it is therefore not possible to gain an insight into system behaviour for an individual transient. Figure 5.15a and Figure 5.15b are plotted based on different values of ITAE for the entire speed range. ITAE is calculated based on equation (5.6). Figure 5.15a shows two plots of ITAE for zero overshoot design, that apply to PI and FL control. It is evident that the ITAE values for the PI control scheme are higher than ITAE values for the FL control scheme. This is so because of significantly longer restoration time required by PI control to return speed to commanded after sudden load application at all speeds (Fig. 5.11).

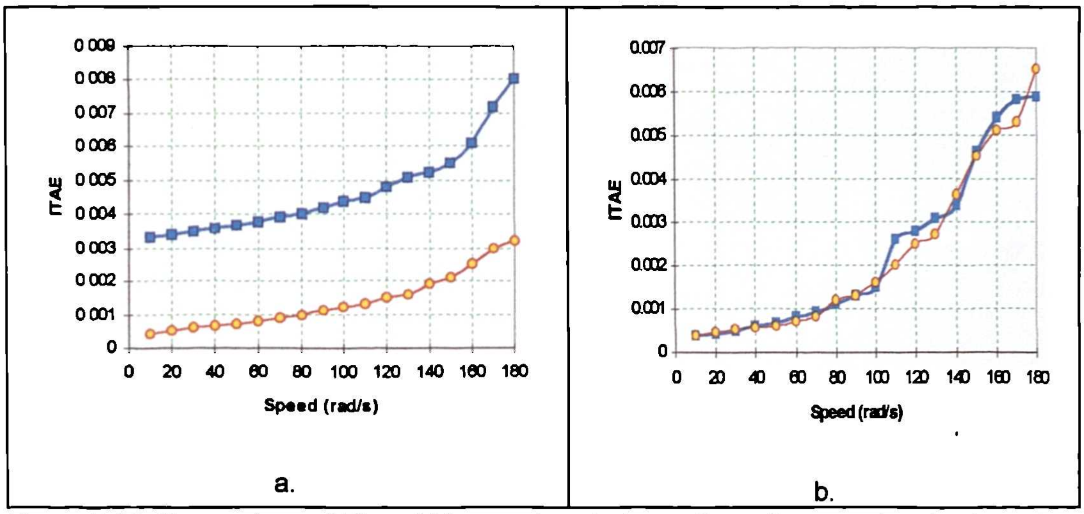
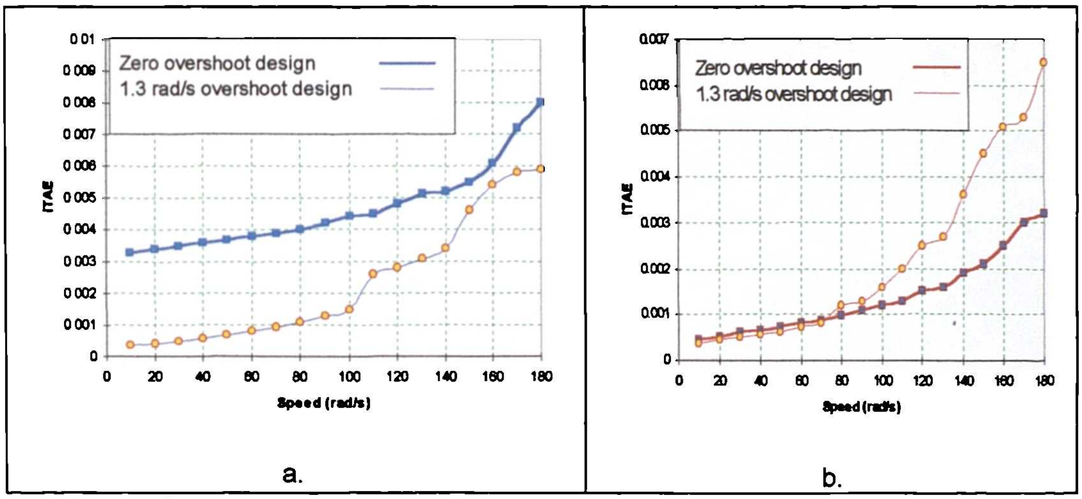


Figure 5.15: Comparison of PI and FL speed control over the entire speed region based on ITAE: a) zero overshoot design; b) 1.3 rad/s overshoot design.
(🟦 for PI, 🟥 for FL)

On the other hand, Fig. 5.15b shows that both control schemes have approximately the same performance and behaviour in terms of ITAE performance index for 1.3 rad/s overshoot design. The performance of the drive for two different speed control schemes is therefore very similar for 1.3 rad/s overshoot design and very different for zero overshoot design, in terms of ITAE.

Next, a comparison can be made of PI and FL control scheme for different design specifications. PI speed controller in 1.3 rad/s overshoot design shows better performance in terms of ITAE with respect to zero overshot design (Fig. 5.16a). Once again, PI speed controller has better tracking or regulating capability if the gains are higher than for the zero overshoot design. However, even the small overshoot cannot be tolerated in many high performance drive applications. In addition, ITAE is relatively large in the region of large initial step speed command because of the large contributions of errors caused by the small step-wise change in speed command (Figs. 5.11e, 5.12e). On the other hand, two designs of FL speed controller have the same performance in the small to medium speed region (Fig. 5.16b), while in medium to large speed region the FL speed controller with zero overshoot design has better performance than the one with 1.3 rad/s overshoot design. It therefore follows that a constant parameter FL speed controller is more robust than a constant parameter PI speed controller.



a. b.

Figure 5.16: Comparison of two design specifications of speed controllers: a) PI control; b) FL control. (🟦 for PI, 🟥 for FL)

PI speed control design with 1.3 rad/s overshoot and FL speed control design with zero overshoot are chosen for further comparison, due to their good performance in terms of ITAE (Fig. 5.17). Both controllers show the same performance in the small and medium speed region, while the performance of PI control scheme is significantly degraded for large speed region.

The ITAE is calculated during the simulation and the average of ITAE for the entire speed range is calculated based on sample interval and number of samples, as shown in Table 5.3. The results from Table 5.3 show that in zero overshoot design the FL control gives superior overall performance for the entire speed range, 0.1364% as compared to 0.467% for the PI control scheme. Comparison for 1.3 rad/s overshoot design shows significant improvement for the PI control scheme (0.232%), while the FL control degrades but maintains better overall performance (0.223%).

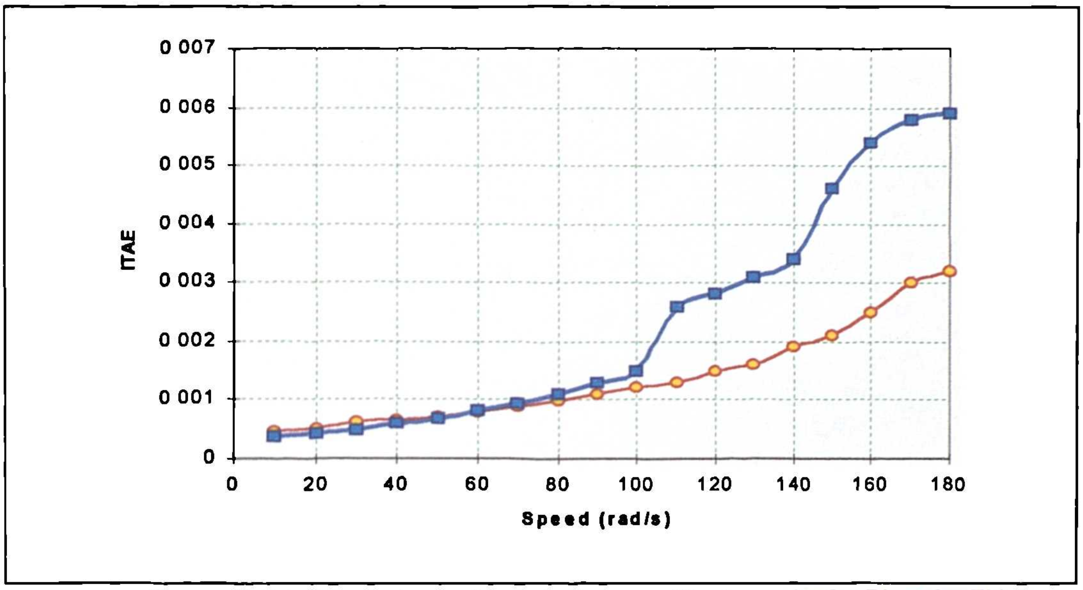


Figure 5.17: Comparison of two design specifications. (🟦for PI, 🟥for FL)

No. of samples, sample interval 10 rad/s	Zero overshoot design		1.3 rad/s overshoot design	
	PI	FL	PI	FL
1	0.0033	0.0004	0.0004	0.0004
2	0.0034	0.0005	0.0004	0.0005
3	0.0035	0.0006	0.0005	0.0005
4	0.0036	0.0006	0.0006	0.0006
5	0.0037	0.0007	0.0007	0.0006
6	0.0038	0.0008	0.0008	0.0007
7	0.0039	0.0009	0.0009	0.0008
8	0.0040	0.0009	0.0011	0.0012
9	0.0042	0.0011	0.0013	0.0013
10	0.0044	0.0012	0.0015	0.0016
11	0.0045	0.0013	0.0026	0.002
12	0.0048	0.0015	0.0028	0.0025
13	0.0051	0.0016	0.0031	0.0027
14	0.0052	0.0019	0.0034	0.0036
15	0.0055	0.0021	0.0046	0.0045
16	0.0061	0.0025	0.0054	0.0051
17	0.0072	0.0030	0.0058	0.0053
18	0.0080	0.0032	0.0059	0.0065
Average of ITAE for entire speed range (%)	0.467 %	0.136 %	0.232 %	0.223 %

Table 5.3: The overall performance based on average ITAE performance index.

5.7 SUMMARY

The basis for drive behaviour comparison is at first established, by classifying the performance indices into four main categories. These are quickness of speed response, control robustness, speed error criteria and economic criteria. The first category is related to the rise and settling time, overshoot and undershoot of speed response. Control robustness can be used to determine capability of control system to compensate the variations of motor and load inertia and load disturbance. The speed error criterion is used to give a unique performance level measure for a set of transients.

The chapter compares operation of a high performance drive with PI and FL speed control over the entire speed region. Two initial designs of speed controllers are considered, firstly the zero overshoot design and secondly, the 1.3 rad/s overshoot design. In order to provide a fair comparison, both controllers are initially designed to yield an identical speed response to the rated step speed command. A detailed comparison between the operation of the drive with speed control by PI and FL technique is made for the cases of application of large and small step speed command other than rated, application of step load torque, operation with inertia other than rated and small step-wise command speed decrease. The comparison is at first made on the basis of numerous transient results obtained by simulation and the speed response traces are therefore overlapped and zoomed. The results are evaluated and discussed. The results are based on many simulation runs and the accuracy of the curves plotted inevitably depends on the accuracy of the readings taken from graphs. A further comparison is based on speed error criterion, whereby the ITAE is calculated during the simulation. The investigation for the overall speed range is made by calculating the ITAE based on 10 rad/s speed interval, and then the average of ITAE is worked out to provide a final comparison of the PI and FL speed controllers.

It is noted that both controllers were evaluated based on a single operating condition design approach. The desired performance only results when the operating point corresponds to or is close to the design case operating condition. The impact of other operating points away from the design case is investigated by applying a small step speed command. The results show that the performance of both controllers is degraded in terms of overshoot and settling time. However, the FL speed controller had demonstrated a better behaviour in at least two thirds of the whole speed range, compared to PI speed controller. The FL speed controller has better capability of load rejection compared to PI speed controller. However, this can be improved by choosing higher PI controller gains than those used in the design case.

Both controllers behave like linear controllers, therefore the calculated controller parameters based on a certain operating condition during the design stage are not appropriate for other operating points. A new set of controller parameters needs to be determined when the operating conditions are changed. This can be automated by applying adaptive control mechanism such as self-tuning, MRAC, etc. With conventional PI controller, the proportional and integral gains can be varied to compensate for change in operating conditions. With FL speed controller there are more than two parameters that can be altered to adapt to the change in operating conditions. This topic will be investigated in Chapter 6.

CHAPTER 6

INVESTIGATION OF SELF-TUNING FUZZY LOGIC SPEED CONTROLLER

6.1 INTRODUCTION

Real-time, on-line implementation of adaptive controllers is an important problem in intelligent control [Vas et al, 1996a]. One of the main concerns of industry regarding the application of artificial-intelligence based controllers is related to the additional hardware or software requirements imposed by these controllers. An FLC is essentially a multi-parameter controller, whose performance depends on the selected shape of membership functions, rule base and scaling factors [Vas *et al*, 1996b], [Vas *et al*, 1994a]. These parameters are usually determined off-line, on the basis of simulations for specific operating conditions, using expert knowledge and/or trial-and-error. However, when such a fixed-parameter FL speed controller is applied in operating conditions that differ from those used in the design stage, performance of the drive may deteriorate [Eminoglu and Atlas, 1996], [Ibrahim and Levi, 1998], [Ibrahim *et al*, 1998a]. In particular, the response of the controller is affected by the variation of the total inertia of the drive and by the speed command and speed profile (for example, responses to step change of speed command and speed tracking performance are quite different, [Heber *et al*, 1997]). When a fuzzy logic controller is used, three main mechanisms can be employed to adapt the controller to differing operating conditions: 1) modification of the scaling factors (self-tuning controller); 2) modification of fuzzy rules (self-organising controller); 3) modification of membership functions [Vas and Stronach, 1996]. Each of the three options can be used independently or in combination with others.

An adaptive fuzzy logic controller, that will enable satisfactory operation under varying operating conditions, can be designed in a number of different ways. Frequently discussed methods encompass self-organising FLC [Ashrafzadeh *et al*, 1995], [Bolognani *et al*, 1994], [Stronach *et al*, 1997], where the rule base is modified on-line, and self-tuning FLC [Eminoglu and Atlas, 1996], [Vas *et al*, 1996b], [Barrero *et al*, 1995], [Feng and Chen, 1996], [Zhen and Xu, 1996], [Parasiliti *et al*, 1996], where scaling factors are modified on-line. Modifying the controller rule base is very effective. The idea is to modify existing rules, discard inappropriate rules and even generate new rules based on the observed performance. However, according to Shenoi [1995], rule modification is complicated and time-consuming. Additional complexity is introduced in a look-up table of the controller. Each adjustment to a rule or membership function requires that the entire look-up table be re-computed in a short period of time. On-line modification of scaling factors appears therefore to be the favoured approach and is the one adopted in this study.

Different methods of scaling factor self-tuning have already been proposed. It is possible to continuously tune the output scaling factor only, while keeping input scaling factors at a constant value [Eminoglu and Atlas, 1996], [He *et al*, 1997]. Alternatively, one may decide to tune the input error scaling factor and output scaling factor [Feng and Chen, 1996] or all the three scaling factors [Parasiliti *et al*, 1996] using only two sets of values for scaling factors: one set for transients and the other set in the vicinity of steady-state [Feng and Chen, 1996, Parasiliti *et al*, 1996]. The adopted methods for scaling factor selection are essentially empirical and based on a trial-and-error approach. Furthermore, in majority of these studies operation for a single speed command is considered. The approach used in this study is to calculate automatically the input scaling factors while at the same time maintaining the desired speed response under different operating conditions. Therefore the difficulty associated with a trial-and-error approach to scaling factor tuning can be minimised.

The adaptation technique should be easy to implement. In classical PI control, the adaptation of the controller gains has been successfully implemented in many types of applications, for example, in process control [Ebach and Graser, 1995] and in variable speed drives [Meshkat, 1996] where, the self-tuning method is based on a search strategy using a motor model. In [Stronach *et al*, 1994] a discrete transfer function-based pole placement algorithm is used to automatically determine the controller gains. There are many examples in which a fuzzy logic mechanism is used to tune the gains of the classical controller in AC motor drive applications [Zhao, 1993], [Chiricozzi *et al*, 1996], [Inoue *et al*, 1996], [Panda *et al*, 1997] and in DC drives [Vas *et al*, 1994b], [Kim *et al*, 1994].

The purpose of this Chapter is to develop a number of self-tuning methods that are applicable for a fuzzy logic speed controller in high performance drives. The purpose of self-tuning is to provide the required quality of speed response for an arbitrary speed reference. Additionally, one of the developed self-tuning mechanisms is capable of compensating for varying inertia of the system. The need for self-tuning of scaling factors with regard to a variable speed command is at first shown in Section 6.2, where the behaviour of scaling factors that is needed to preserve the same quality of speed response over the entire speed region is investigated. Selection of self-tuning procedures is discussed in Section 6.3 and the four methods that are to be developed further are introduced. Sections 6.4 to 6.6 present these four methods, while Section 6.7 gives a comparison of the results. Finally, Section 6.8 summarises the Chapter.

6.2 NON-LINEAR NATURE OF FL SPEED CONTROLLER SCALING FACTORS

In Chapter 4, a CPFL controller was designed using fixed scaling factors. The initial values of scaling factors were calculated off-line, based on the rated data and specific operating condition. Next, an optimal set of scaling factors for the controller was obtained by manual tuning (Section 4.4.2.3). A good performance of the CPFL controller was achieved, as shown in Chapter 5, only for the large and medium step speed commands [Ibrahim and Levi, 1998]. The performance deteriorated for the small step speed references. The main reason is that a set of constant scaling factors is applied in the whole speed region. The good performance of the FL speed controller can only be maintained in the entire speed region if differing sets of scaling factors are used for different initial step speed command.

In order to investigate how scaling factors of a FL speed controller need to be changed in order to preserve the desired quality of speed response over the entire speed region, an exhaustive simulation study is performed. Output scaling factor is kept fixed at a value of G_u = 3 in all the cases. Speed response is required to be aperiodic (or at worst nearly aperiodic with overshoot of less than 0.1 rad/s), with minimum settling time. The change of error scaling factor is at first fixed and simulations are then performed for a particular speed command with varying values of error scaling factor, until the desired speed response is achieved. The procedure is repeated for speed commands from 10 rad/s up to 180 rad/s in steps of 10 rad/s and the required values of error scaling factor are found. The same study is performed next with interchanged role of error scaling factor and change of error scaling factor. The actual behaviour of scaling factors for a wide range of operating conditions is depicted in Fig. 6.1. This figure applies to the FL controller described in Chapter 4 and termed "standard" CPFL speed controller (Section 4.4.1). The non-linear characteristic of error scaling factor and change of error scaling factor is evident from Fig. 6.1a,b. The required input scaling factors for speed commands from 100 rad/s up to 180 rad/s are approximately linear, as shown in Fig. 6.1a,b, while non-linear input scaling factors are needed for the step speed command from 10 rad/s up to 100 rad/s. Figure 6.2 shows results of a similar study conducted for the FL controller that was termed "off-line optimised" CPFL controller in Chapter 4 (Section 4.4.2). It is evident that the non-linearity of input scaling factors is more pronounced in Fig. 6.2 than in Fig. 6.1. Thus it follows that the degree of the non-linearity of the scaling factors is also affected by the controller parameters selected for the design case (Section 4.4.2.3).

Results given in Figs. 6.1 and 6.2 clearly visualise the non-linear nature of scaling factors for the entire speed range. If the desired quality of performance is to be preserved over the entire speed region, it is necessary to somehow provide a mechanism that will vary the scaling factors as a function of the speed reference. Possible approaches to adaptation and developed self-tuning mechanisms are elaborated in the forthcoming sections of this Chapter.

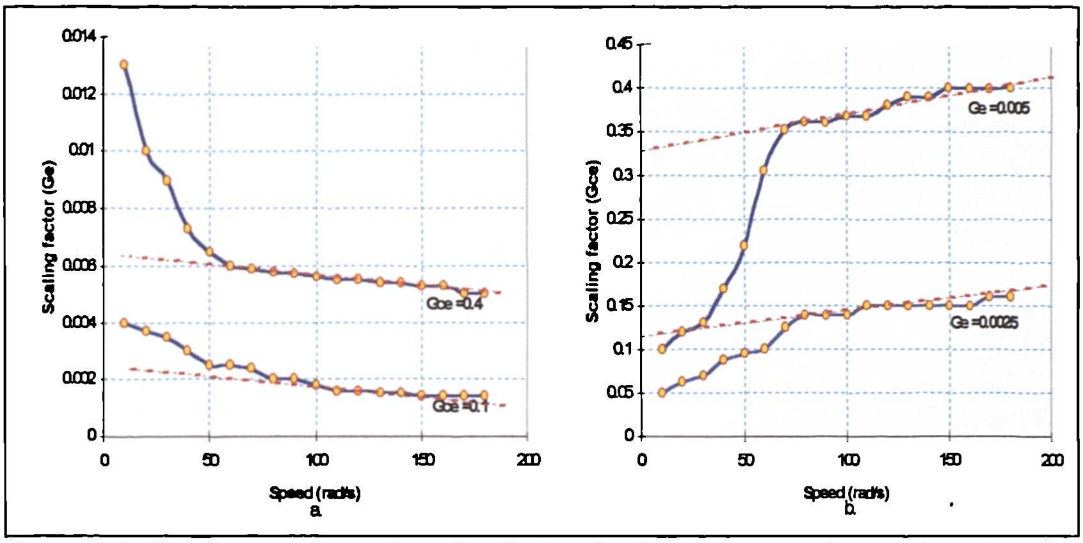


Figure 6.1: Non-linear nature of scaling factors for the entire speed range based on the standard CPFL speed controller: a) change of error scaling factor; b) error scaling factor.

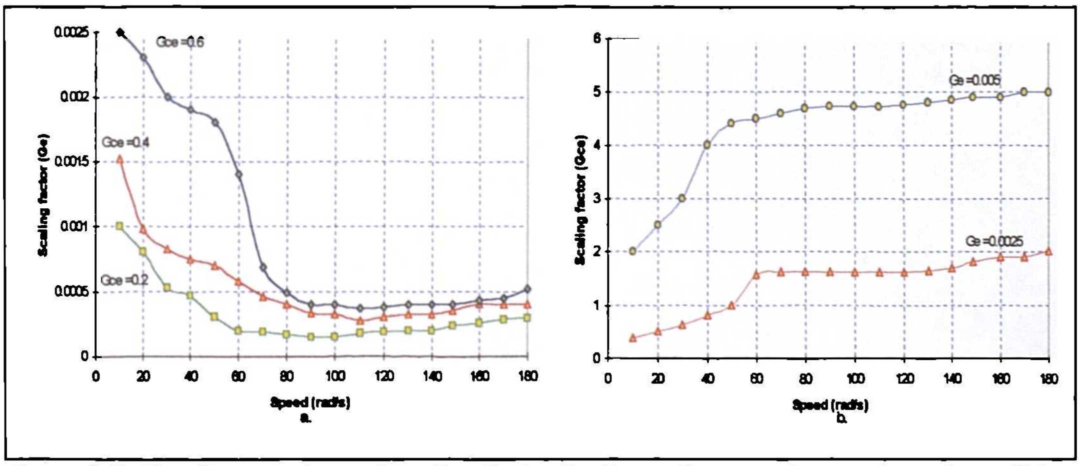


Figure 6.2: Non-linear nature of scaling factor for the entire speed range based on off-line optimised CPFL speed controller: a) error scaling factor; b) change of error scaling factor.

6.3 SELECTION OF SELF-TUNING PROCEDURES

The basic idea of a self-tuning system is to construct an algorithm that automatically changes controller parameters to meet a particular requirement. This is done by the addition of an adjustment mechanism which handles the initial tuning (initial adjustment) of a controller and adjusts on-line the coefficients of the controller by monitoring the system, so that a required performance is maintained. After the initial tuning, the adjustment mechanism can be disabled in which case on-line adaptation does not take place. If the adjustment mechanism is not disabled, it provides a mean of continually adapting the controller to changes in the system [Wellstead and Zarrop, 1991].

There are a number of ways of approaching the task of providing the adjustment mechanism:

1. An auxiliary controller can be used as an automatic on-line adjustment mechanism. It identifies the system using measured input and output data and then modifies the parameters of the main controller on the basis of system identification (indirect self-tuning) or adaptation mechanism (direct self-tuning). In this approach the "adaptation mechanism" observes the signals from the control system and adapts the parameters of the controller to maintain performance.

2. Model reference adaptive controller (MRAC) can be used to specify the desired performance of the control system. Measured input/ output data are used to monitor the system performance. These are then combined with the reference model output according to an adaptation rule and the result is used to adjust the controller. The overall aim is to force the output of the plant to track the model output.

3. Expert tuning systems is another possibility. In this approach measured input/output data are compared in an expert system against qualitative performance criteria. The results of these comparisons are used to adjust the controller settings.

4. In gain scheduling, the controller parameters need to be changed from one known set to another. This can be done by pre-calculating the controller parameters for specific operating regimes.

The two approaches utilised in this research are those numbered 1 (direct self-tuning) and 4.

The design process of a self-tuning system requires three steps: modelling of the system, initial design of the controller and development of the self-tuning mechanism. Design of the self-tuning FL speed controller is to be elaborated in this Chapter and it involves two main steps, namely initial tuning of scaling factors and subsequent on-line adaptation of scaling factors.

Behaviour of a CPFL speed controller has been studied in detail in the previous chapter. The CPFL controller has exhibited its good performance for large reference speed. The performance deteriorated for the medium and small reference speed. The fixed parameter controller therefore cannot give the desired performance for an operating condition that differs from the design case. It is believed that good performance of the speed controller can be preserved if the self-tuning procedure is used. The self-tuning mechanism is required to calculate a new set of the controller parameters when the operating conditions change.

Performance index criteria such as overshoot, settling time and integral speed error criteria can be used to characterise the controller performance [Vas, 1998], [Afonso *et al*, 1997], [da-Silva and Acarnley, 1997]. Tuning of the controller parameters is terminated when the desired performance is achieved. A traditional way to tune industrial PI controller gains, with the aim of minimising the overshoot and reducing the rise-time, is to perform the following actions (Section 3.8.1):

- To decrease overshoot, the integral time constant T_i is increased (or K_i decreased) when the output is approaching the set point. To reduce the rise time, the integral time constant T_i is decreased during the transient.

- To decrease rise time, the proportional gain K_p is increased during the transient and decreased when the output is approaching the set point.

These rules are rather straight-forward and are easy to implement with a fuzzy-logic mechanism if the controller is of standard PI (or PID) type. For example, a fuzzy gain scheduler can be used to tune the gains of a PID controller [Zhao *et al*, 1993]. Rule 1 is used in order to generate a "stronger" control action, while rule 2 is taken in order to avoid a large overshoot or small proportional gain:

Rule 1: if $e(k)$ is PL and $\Delta e(k)$ is ZE, then K_p is PL, and K_d is PS, and K_i is PS

Rule 2: if $e(k)$ is ZE and $\Delta e(k)$ is NL, then K_p is PS, and K_d is PL, and K_i is PM

However, if the fuzzy gain scheduler is used to tune the scaling factors of the FL controller, then the control rules are difficult to derive as the correlation between G_e, G_{ce}, and G_u and classical control action is very complex.

The impact of scaling factors on controller performance has been investigated in detail in Chapter 4. The new tuning rules for the specific control strategy are established based on this knowledge. The pre-calculated values of scaling factors, based on overshoot and steady-state error criteria can be used to improve the performance. A good speed controller should have the capability to yield a fast speed response without overshoot. However, overshoot may occur when the speed command changes or variation of inertia takes place.

Quantitative mathematical formulae, that can express the relationship between controller gains and system parameters, do not exist in fuzzy logic. Engineers and operators do however have experience and use intuitive rules for tuning or choosing gains for their controllers. Fuzzy logic is an effective tool for conceiving qualitative information and expressing human experience. In this sense the fuzzy inference method may be an ideal alternative to tackling the non-linear gain tuning problem. In common practice, the tuning of the controller gains can be based on two main methods: the step-based gain tuning method and the cycle-based gain tuning method [Chiricozzi *et al*, 1996]:

i) Step-based gain tuning: using the information derived from each sampling step, tune the gains accordingly. In this case, the step information includes the error and the change of error.

ii) Cycle-based gain tuning: using the information from each step reference change, tune the gains so as to improve the performance. In this case, the commonly used cycle parameters include overshoot, rising time etc. This technique has been implemented successfully in [Inoue *et al*, 1993] and [Inoue *et al*, 1996] where a FLC is used to tune the gains of a PI speed controller. Three sets of controller gains are employed in order to maintain the desired controller performance over the complete operating range.

The methods investigated here are based on scaling factor adjustment. Four different techniques for tuning the scaling factors of a FL speed controller are considered, with the idea of maintaining the desired controller performance under different operating conditions. These techniques are the following:

i) The first two methods automatically calculate the optimal input scaling factors for different speed command changes. The main purpose of automating the procedure is the tediousness of trial-and-error tuning. A self-tuning controller can perform this task on-line and consequently save time for the operator and improve the response of the process. Thus the difficulty of tuning the input scaling factors can be eliminated. The cycle-based approach is therefore adopted (in step-based gain tuning the controller parameters are calculated for every step in each sampling interval, so that a long processing time is required). The tuning strategy is to minimise the overshoot and settling time. In method 1, overshoot and integral speed error criteria are used in the tuning process, while in method 2 only overshoot criterion is considered. The correct values of scaling factors are found faster with method 2 than with method 1. The main reason is that in method 2 different step changes of scaling factors are utilised based on an auxiliary FLC, while fixed step changes of error scaling factor are used in method 1 (Section 6.4). In addition, the tuning of scaling factors in method 2 is performed after the occurrence of each overshoot and undershoot in speed response (Section 6.5). Therefore, the adaptation of scaling factors is done more rapidly compared to method 1. As a result the speed response reaches non-oscillatory steady-state more quickly. In method 2 (Section 6.5), the tuning process execution depends on how often the overshoot occurs and tuning therefore takes place more frequently than in method 1.

ii) The remaining two methods are based on on-line tuning of scaling factors by means of an auxiliary FLC. The knowledge gained from the first two methods is used to generate the tuning rules and tuning strategies. Method 3 is designed to compensate the variation of inertia and speed command change on-line. Thus input and output scaling factors have to be tuned on-line. Method 4 is developed to compensate for the variation of step speed commands only and the implementation is much simpler than for method 3. In method 3, the adaptation is performed based on 16 fuzzy rules, while only one rule is employed in method 4. In general, the concepts used in both methods are similar to classical gain scheduling technique. The difference is that rule base forms the scaling factor scheduler in method 3, (Section 6.6.1), while in method 4 the non-linear input and output membership functions form the scaling factor scheduler for the entire speed range (Section 6.6.2). In addition, in the classical gain scheduling the gain has to be calculated off-line for every single operating point, while in fuzzy logic scaling factor scheduling the scaling factor is scheduled based on heuristic rule and membership function.

6.4 AUTOMATIC TUNING OF SCALING FACTORS BASED ON IAE AND OVERSHOOT CRITERIA (Method 1)

The structure of the FL speed controller used in the section is identical to the off-line optimised CPFL controller of Chapter 4 (Section 4.4.2). In this method the IAE and overshoot of the speed response are used together as means for tuning the input scaling factors of the FL speed controller. The value of change of error scaling factor is constant (G_{ce} = 0.41), while the initial value of the error scaling factor is chosen to be the same as in the design case, G_e= 0.005. The optimal error scaling factor is then obtained by adjusting G_e, based on the fixed step change of G_e. Figure 6.3 shows the structure of the FLC and the adjustment mechanism. The error scaling factor is considered only in this method because of its ability to compensate for a large overshoot of speed response (Chapter 4). The error scaling factor is updated in every iteration. The flowchart for the tuning procedure is given in Fig. C1 (Appendix C).

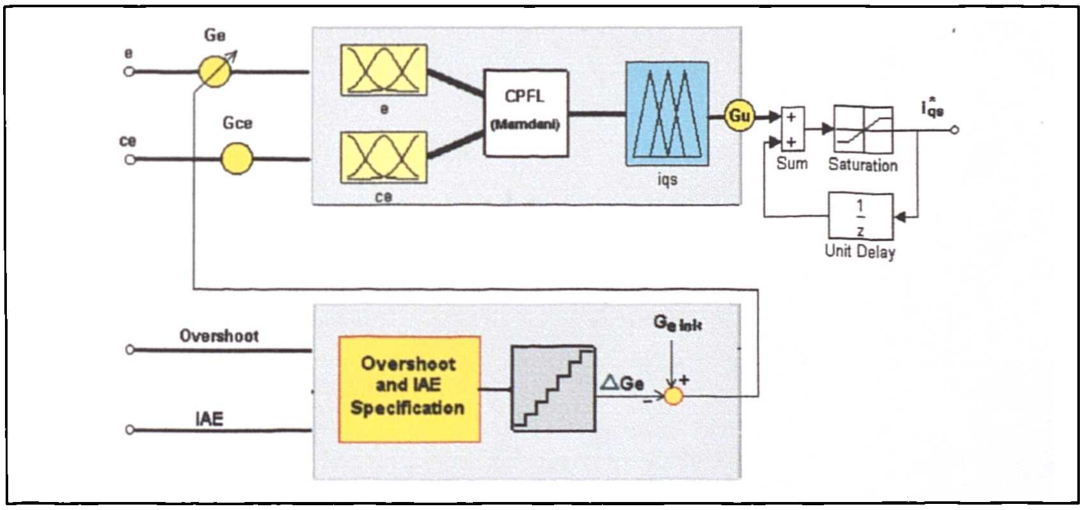


Figure 6.3: Structure of the FL controller with an adjustment mechanism (Method 1).

Figure 6.4 shows the progress of the tuning process for the different step speed command, based on IAE performance index. The procedure for error scaling factor tuning is as follows. A speed command is selected, say rated speed. A sequence of step speed commands of the sequence 180 rad/s, 0 rad/s, 180 rad/s, 0 rad/s is then applied. The error scaling factor is gradually tuned until oscillatory behaviour and overshoot are completely eliminated.

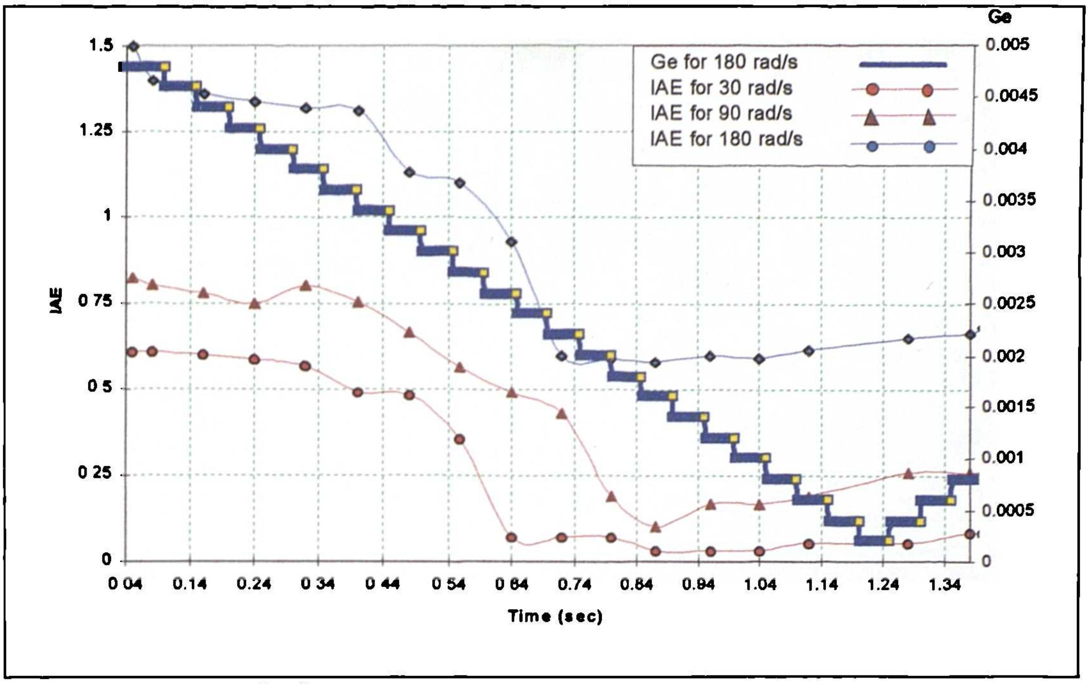


Figure 6.4: IAE and G_e for different step speed commands during the tuning process.

Three cases are considered in this study: large, medium and small step speed command application. Figure 6.5a shows that the oscillation in the speed response disappeared after 1.04 sec or 13 cycles of iteration. The final value of the error scaling factor is G_e = 0.0011. Large overshoot and oscillatory speed response take place for medium step speed application, Fig. 6.5b. The response becomes non-oscillatory after 12 cycles of iteration or 1 sec. At this time the final values of G_e is 0.0012. For small step speed command application, the speed response has very large overshoot compared to medium and large step applications, Fig. 6.5c. The speed response becomes non-oscillatory after 11 cycles of iteration, that is 0.9 sec. At this stage, the final value of G_e is 0.0014. It therefore follows that, in order to maintain the controller performance during the small step speed command application, the input scaling factor G_e has to be increased by at least 30% with respect to the scaling factor obtained for large speed command (G_e= 0.0011). This method of tuning is based on the fixed step change of G_e. If the size is too small, slow convergence results but the final value of the tuned variable is more accurate. If the step size is too big, then fast convergence occurs, but at the same time it does not guarantee the good result. Therefore, the size of the step G_e change is normally chosen based on experience. In this case, the value is 0.0002 which is 25 times smaller than the initial value of G_e. The simulation results are summarised in Table 6.1.

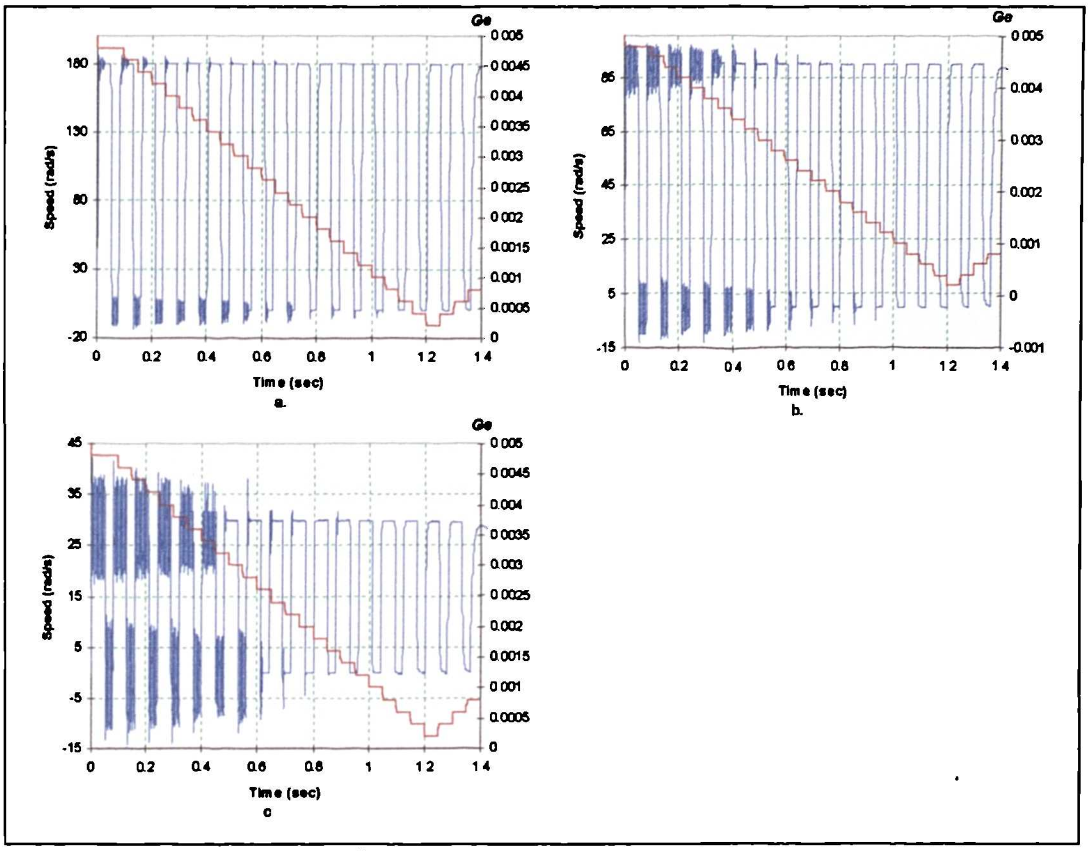


Figure 6.5: Tuning of the G_e for different step speed reference: a) large step speed command; b) medium step speed command; d) small step speed command. (🟦 speed responses, 🟥 G_e)

speed command (rad/s)	G_e (Initial, G_e = 0.005)	convergence time (sec)	cycles of iteration
180	0.0011	1.04	13
90	0.0012	1.00	12
30	0.0014	0.90	11

Table 6.1: Scaling factor for different speed commands.

Figure 6.6 shows plots of IAE and overshoot versus simulation time. It shows that the IAE and overshoot of speed response can be used as a performance measure in the process of tuning of the FL speed controller. Large overshoot in speed response results in large IAE value, while small overshoot creates small IAE value. The tuning process is terminated when the system achieves non-oscillatory speed response.

The method of automatic tuning, described in this section, performs automatic tuning of only error scaling factor while the change of error scaling factor is fixed and essentially detuned. This is regarded as the major shortcoming of the procedure. Next, the automatic tuning algorithm has to be repeated over the entire region of the operating speeds. It is therefore appropriate for use during commissioning of the drive and is essentially not an on-line self-tuning method. However, once when the appropriate values of error scaling factor are determined, these could be stored in a look-up table and used for on-line self-tuning during actual operation of the drive.

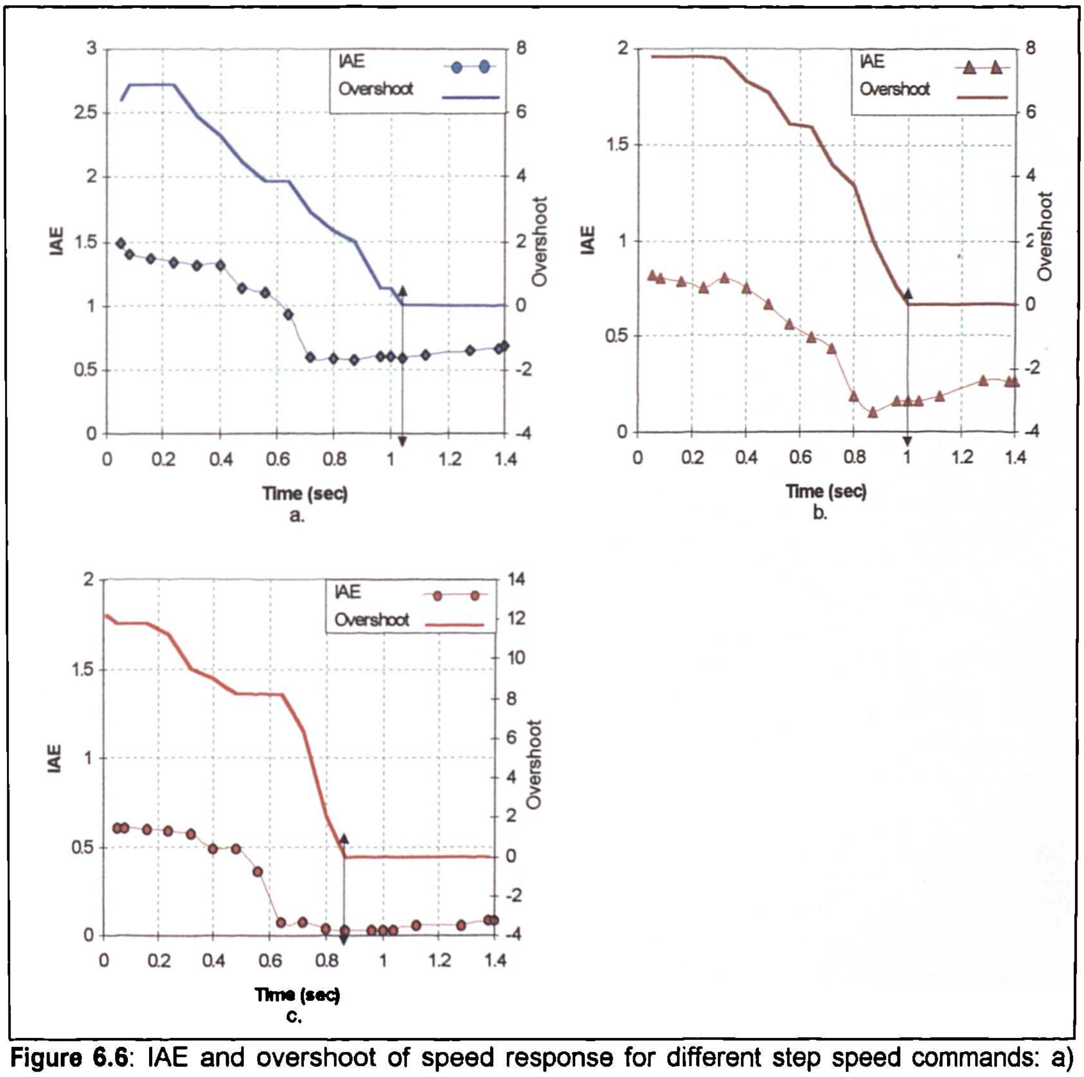


Figure 6.6: IAE and overshoot of speed response for different step speed commands: a) large step speed command; b) medium step speed *command*; *c) small step* speed command.

6.5 AUTOMATIC TUNING OF SCALING FACTORS BASED ON AUXILIARY FL CONTROLLER AND SPEED RESPONSE OVERSHOOT (Method 2)

In this section, automatic tuning of scaling factors is performed by an auxiliary FL controller. Overshoot of speed response is used as the input variable of the auxiliary FLC. The scaling factors of the FL speed controller are updated depending on the overshoot specification (Fig. C2, Appendix C). This technique can substitute the trial-and-error tuning method used in Chapter 4 and in Section 6.2. In contrast to the method of automatic tuning developed in Section 6.4, the method elaborated in this section performs automatic tuning of both change of error and error scaling factors. Output scaling factor of the FL speed controller is once more constant and equal to G_u = 3.

The structure of the FL system used in this section is depicted in Fig. 6.7. The design of the auxiliary FL controller is based on one input variable, namely overshoot of speed response. Its output variables are scaling factors for error and change of error, G_e and G_{ce}. The output from the auxiliary FLC is used to update the scaling factors of the FL speed controller for every cycle comprising one overshoot and one undershoot. The structure of the FL speed controller is again based on the off-line optimised CPFLC discussed in Chapter 4 (Section 4.4.2).

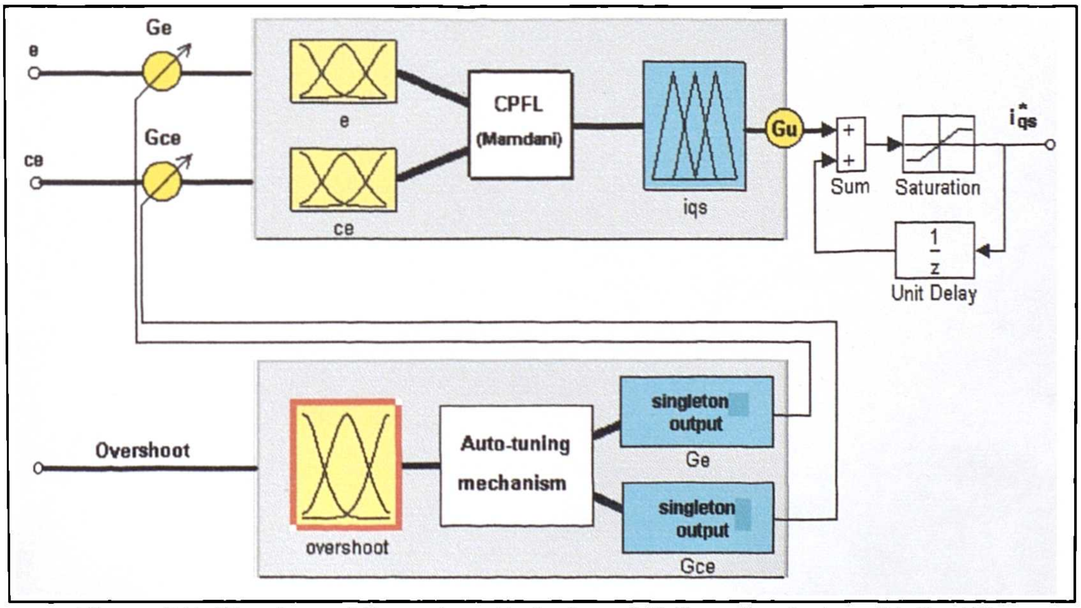
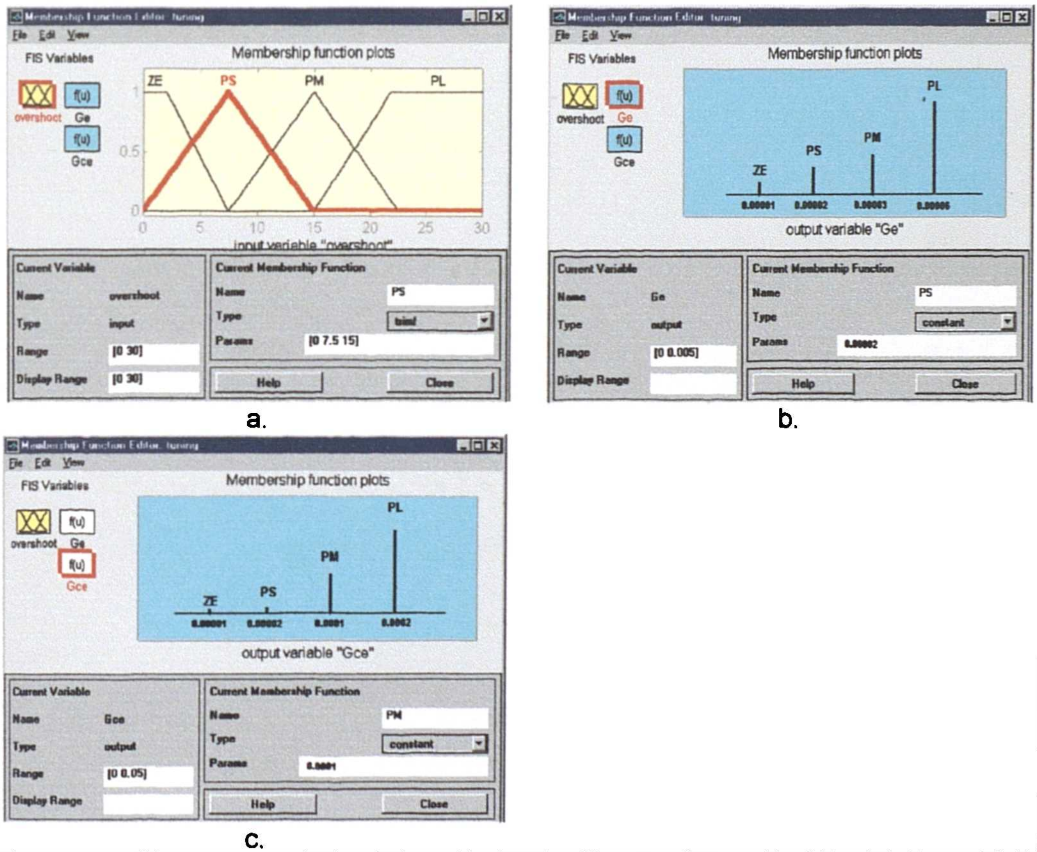


Figure 6.7: Structure of the automatic-tuning of FLC scaling factors (Method 2).

Four sets of membership functions are used to represent the input and output fuzzy logic variables of the auxiliary FL controller, as shown in Fig. 6.8. These are ZE, PS, PM and PL and the output membership functions of the auxiliary FL controller are of singleton type (Table 6.2, Fig. 6.8b,c). The maximum overshoot of speed response is assumed to be 30 rad/s. The tuning strategy is to decrease G_e when the output response has a large overshoot (Chapter 4). However, if G_e is too small, the steady-state accuracy will be affected (Chapter 4). The error scaling factor G_e has more pronounced influence on the speed response than G_{ce} (Chapter 4). Therefore, the step that is used to update the G_e for large overshoot is three times smaller than the step used to update G_{ce} (Fig. 6.8b,c and Table 6.2). On the other hand, reducing G_{ce} will improve the rise time of the speed response (Chapter 4). The fuzzy rules are generated based on Table 6.3.



Current Variable
Name overshoot
Type input
Range [0 30]
Display Range [0 30]

Current Membership Function
Name PS
Type trimf
Params [0 7.5 15]

a.

Current Variable
Name Ge
Type output
Range [0 0.005]
Display Range

Current Membership Function
Name PS
Type constant
Params 0.00002

b.

Current Variable
Name Gce
Type output
Range [0 0.05]
Display Range

Current Membership Function
Name PM
Type constant
Params 0.0001

c.

Figure 6.8: Automatic-tuning of FLC scaling factors: a) membership functions of input variable of auxiliary FL controller; b) membership functions of output variable G_e; c) membership functions of output variable G_{ce} of auxiliary FL controller.

Membership functions	G_e	G_{ce}
ZE	0.00001	0.00001
PS	0.00002	0.00002
PM	0.00003	0.0001
PL	0.00006	0.0002

Table 6.2: Output membership functions of the auxiliary FL controller.

Rules	Overshoot	G_e	G_{ce}
1	ZE	ZE	ZE
2	PS	PS	PS
3	PM	PM	PM
4	PL	PL	PL

Table 6.3: Rule base of auxiliary FLC.

Figure 6.9 shows in detail how the singleton output from auxiliary FLC is used to tune the scaling factors of the speed FLC. It clearly indicates that when the overshoot/undershoot is large, the large step change in scaling factor takes place, when the overshoot/undershoot is medium, the medium step change is activated and for the small overshoot/undershoot small change takes place (Fig. 6.9). It is evident that the tuning process is performed for every cycle comprising one overshoot and one undershoot (Fig. C2, Appendix C). The oscillatory speed response is significantly improved (Fig. 6.9a,b) when rule no 1 exists (Table 6.3), (Fig. 6.9c,d are obtained without rule 1). Thus, the implementation of the auxiliary FLC is based on the rules shown in Table 6.3.

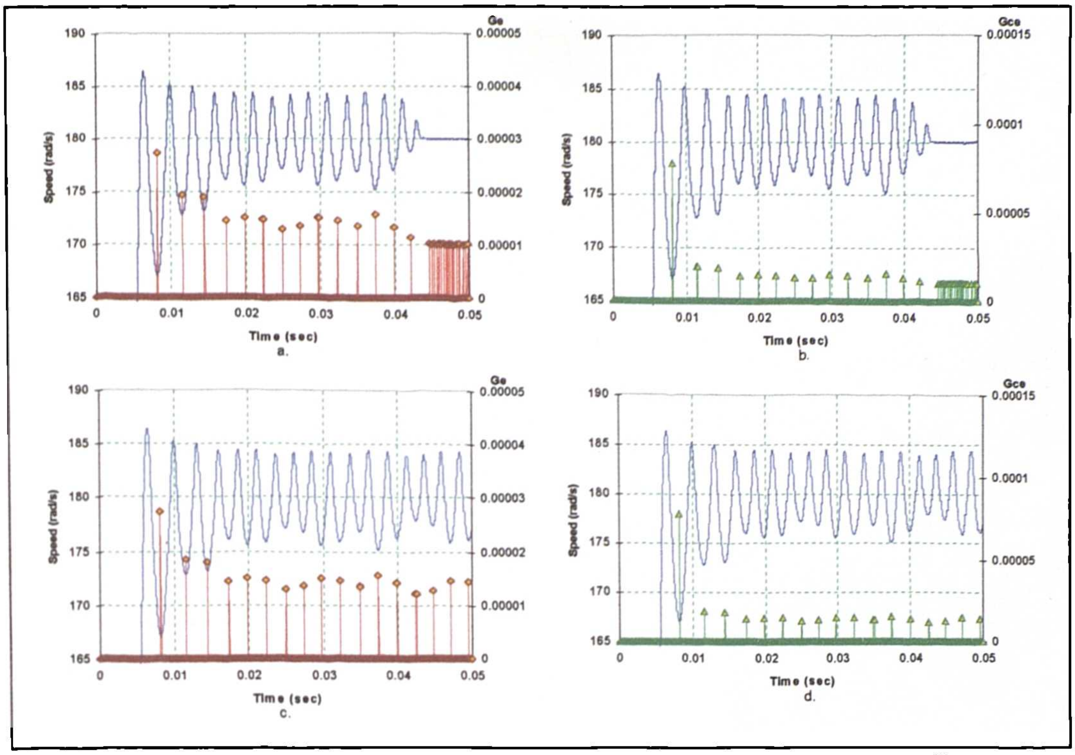


Figure 6.9: Output from the auxiliary FLC for every overshoot/undershoot: a) G_e with rule 1; b) G_{ce} with rule 1; c) G_e without rule 1; d) G_{ce} without rule 1 . (🟦 speed responses, 🟥 G_e, 🟩 G_{ce})

The responses shown in Fig. 6.10 are obtained using the same procedure as in Section 6.4. The overshoot of the response is measured during rising cycles of the speed. The output from auxiliary FLC is used to update the scaling factors of the CPFLC. There is no tuning during the falling cycles of the speed (Fig. 6.10). An optimal scaling factor for a large step speed command (Fig. 6.10a) is small, G_e= 0.0004 compared to an optimal scaling factor for the medium impulse speed command (Fig. 6.10b) G_e= 0.0015 and small step speed command (Fig. 6.10c) G_e = 0.0011. However, only small changes are required in G_{ce}, 0.3954 for large, 0.395 for medium and 0.3927 for small step speed command. The time required to achieve non-oscillatory speed response for small step speed response is longer compared to medium and large step responses (0.42 sec. for small, 0.32 sec. for medium and 0.2 sec. for large). The simulation results reported in Fig. 6.10 are summarised in Table 6.4.

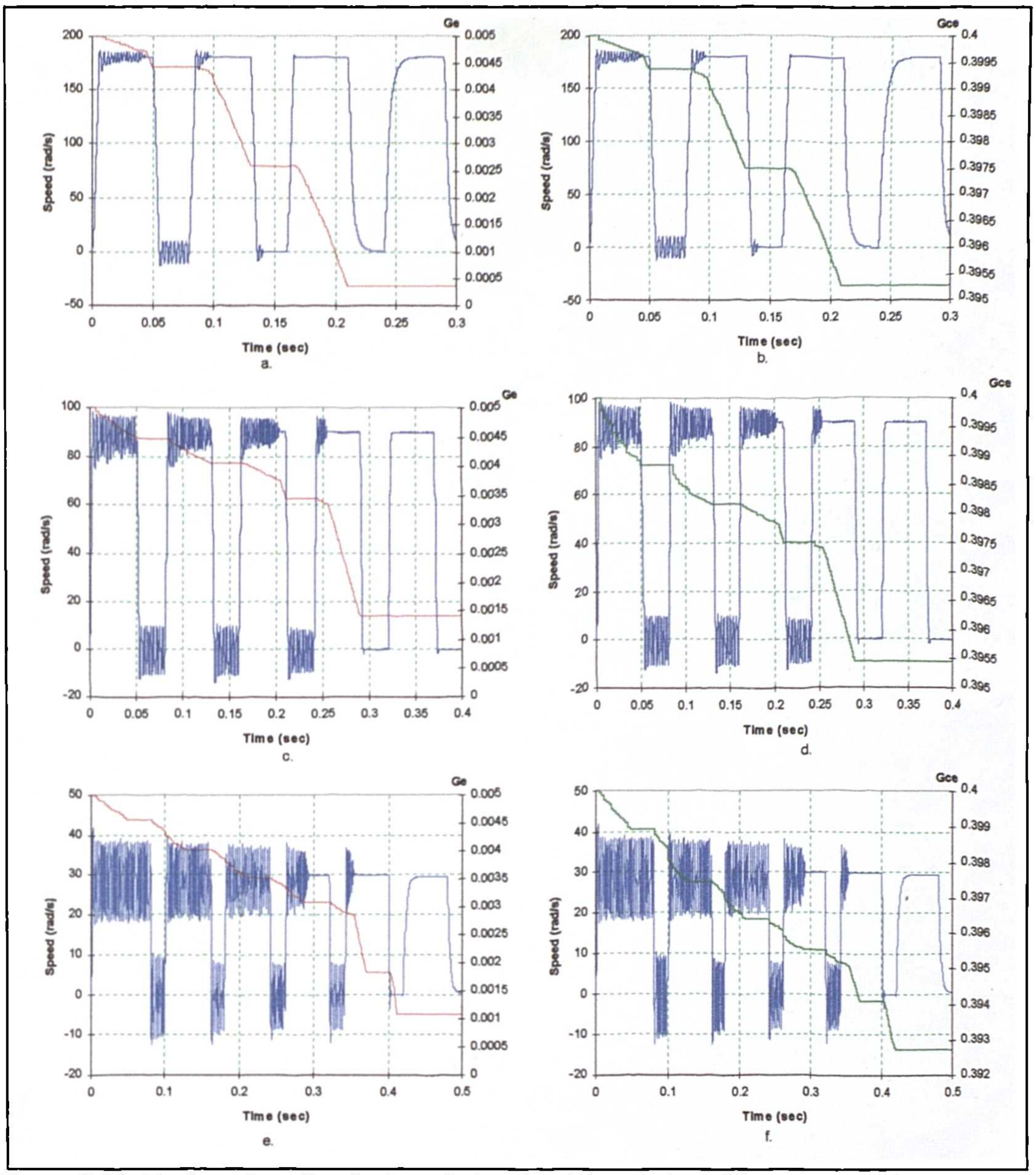


Figure 6.10: Tuning of scaling factors during the rising cycles of the step speed responses: a) G_e for large speed command; b) G_{ce} for large speed command; c) G_e for medium speed command; d) G_{ce} for medium speed command; e) G_e for small speed command; f) G_{ce} for small speed command. (🟦 Speed responses, 🟥 G_e, 🟩 G_{ce})

Speed command (rad/s)	G_e (Initial, G_e = 0.005)	G_{ce} (Initial, G_{ce} = 0.4)	Time to achieve non-oscillatory response (sec)
180	0.0004	0.3954	0.20
90	0.0015	0.3950	0.32
30	0.0011	0.3927	0.42

Table 6.4: Scaling factors to obtain non-oscillatory speed response.

Tuning of the scaling factors is performed next in both rising and falling cycles of the speed response. The simulation results, from the study identical to the one illustrated in Fig. 6.10, are reported in Fig. 6.11. The only difference is that the time required to reach the non-oscillatory condition is shorter compared to the previous tuning method. Tuning of the scaling factors based on the different initial values of scaling factors, is shown in Fig. 6.11. The optimal scaling factors for large speed command are obtained after 2 cycles of the tuning process or in 0.13 sec (Fig. 6.11a,b).

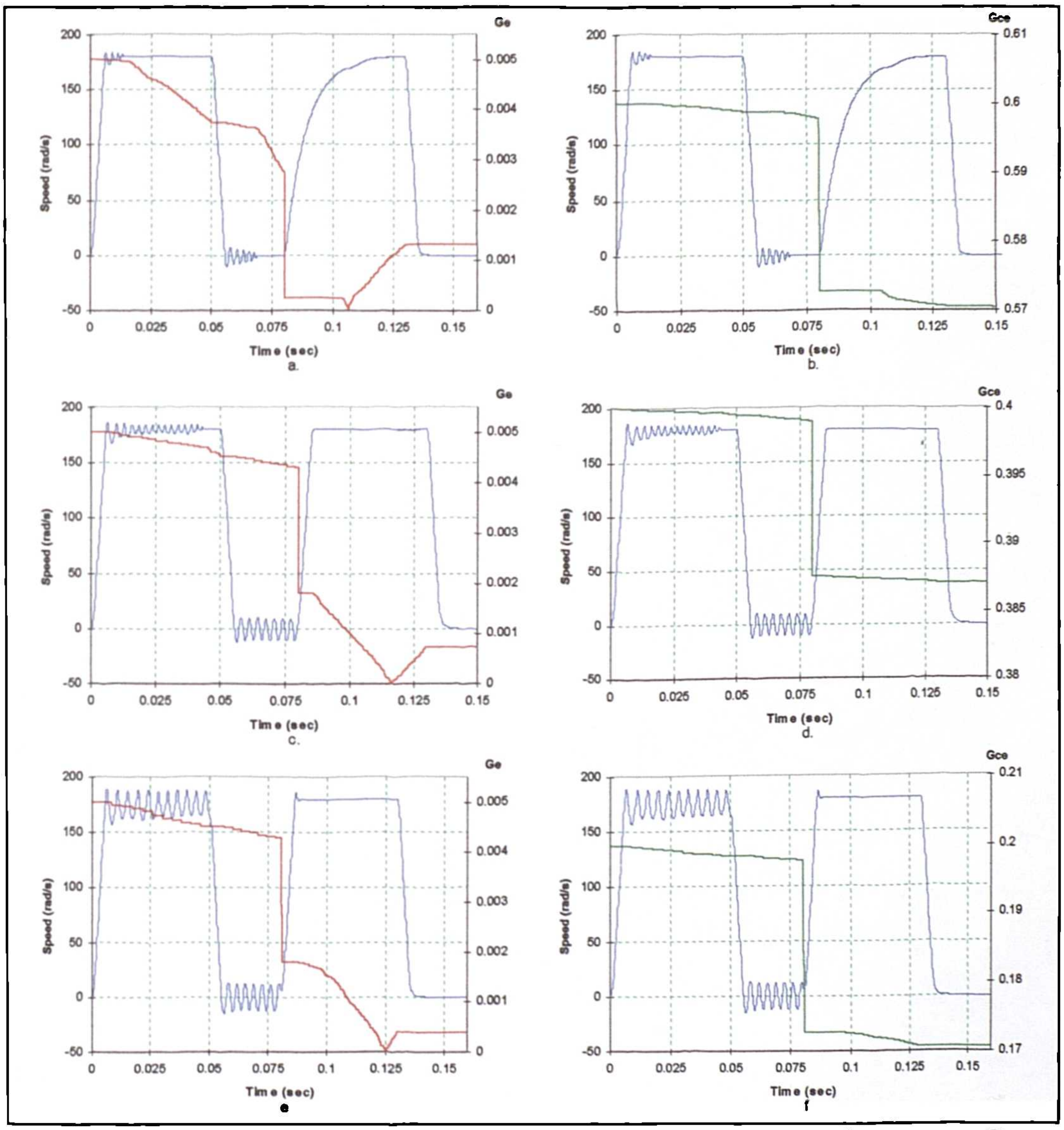


Figure 6.11: Tuning of scaling factors: a) tuning based on initial values G_e = 0.005, G_{ce} = 0.6; b) tuning based on initial values G_e = 0.005, G_{ce} = 0.4; c) tuning based on initial values G_e = 0.005, G_{ce} = 0.2. (🟦 Speed responses, 🟥 G_e, 🟩 G_{ce})

The simulation results shown in Fig. 6.11 are summarised in Table 6.5. Three sets of inputs scaling factors are obtained, based on different initial values of scaling factors.

Initial, G_e = 0.005	G_e	G_{ce}	rise time	settling time
G_{ce} = 0.6	0.0012	0.57	0.02	0.045
G_{ce} = 0.4	0.00074	0.395	0.005	0.007
G_{ce} = 0.2	0.00048	0.17	0.007	0.009

Table 6.5: Settling and rise time for different sets of scaling factor values.

Fig. 6.11 also indicates that the required speed response can be obtained even when the tuning process is based on different initial values of scaling factors. The solution for the final set of scaling factors is dependent on the correlation between G_e and G_{ce}. Table 6.5 shows that short rising time and fast settling time can be obtained if the final values G_e= 0.0074, G_{ce}= 0.395 are used. Further investigation will be based on these values.

When the step speed command is one half of the nominal speed, the overshoot in speed response becomes large compared to the nominal speed case. The same test described for large speed command is now done for medium speed command. The simulation results are summarised in Table 6.6. The final values of input scaling factors are 0.0012 for G_e and 0.37 for G_{ce} (Fig. 6.12a,c). The system settled within 0.13 sec or after finished 2 cycles of iteration. For the small step speed command, the system response requires longer tuning time, about 0.29 sec or 5 cycles of iteration (Fig. 6.12b,d).

speed command	G_e Initial, G_e = 0.005	G_{ce} Initial, G_{ce} = 0.4	cycles of iteration	settling time (sec)
90 rad/s	0.0012	0.37	2	0.13
30 rad/s	0.0013	0.3925	5	0.29

Table 6.6: Scaling factors, cycles of iteration and settling time for different speed commands.

This automatic tuning method can be used for tuning of scaling factors of a variable speed drive during commissioning. The method can be used to substitute the trial-and-error method by enabling the tuning controller to estimate the optimal scaling factors during the start-up process. Once the values have been determined, the tuning controller can be disabled.

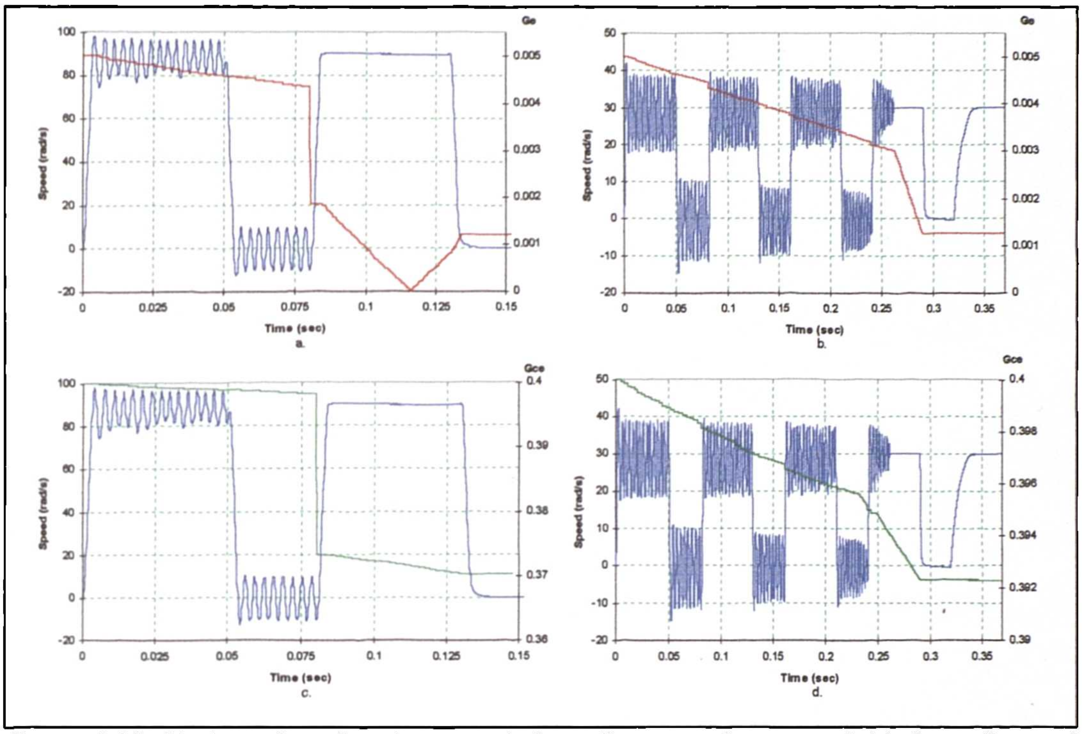


Figure 6.12: Tuning of scaling factors: a),c) medium speed command; b),d) small speed command. (🟦 Speed responses, 🟥 G_e, 🟩 G_{ce})

A further comparison can be made between results depicted in Fig. 6.10 and in Figs. 6.11 and 6.12. The tuning method used in the case of Figs. 6.11, 6.12 has demonstrated a better performance in terms of the time needed to achieve non-oscillatory speed response (0.13 sec. compared to 0.21 sec. for large speed command, 0.13 sec. compared to 0.29 sec. for medium speed command and 0.29 sec. compared to 0.42 sec. for small speed command). The tuning method used in Figs. 6.11 and 6.12 is therefore used in further investigation.

The scaling factors, obtained based on manual tuning procedure for the entire speed range, were plotted in Fig. 6.1 and Fig. 6.2. Corresponding scaling factors (blue traces), shown in Fig. 6.13, are based on the automatic tuning method of this section. One notes that the behaviour of G_e and G_{ce} is similar. In general, large speed command requires small error scaling factor (G_e) and large change of error scaling factor (G_{ce}), while small speed command requires large value of G_e and small value of G_{ce} (Fig. 6.13a,b).

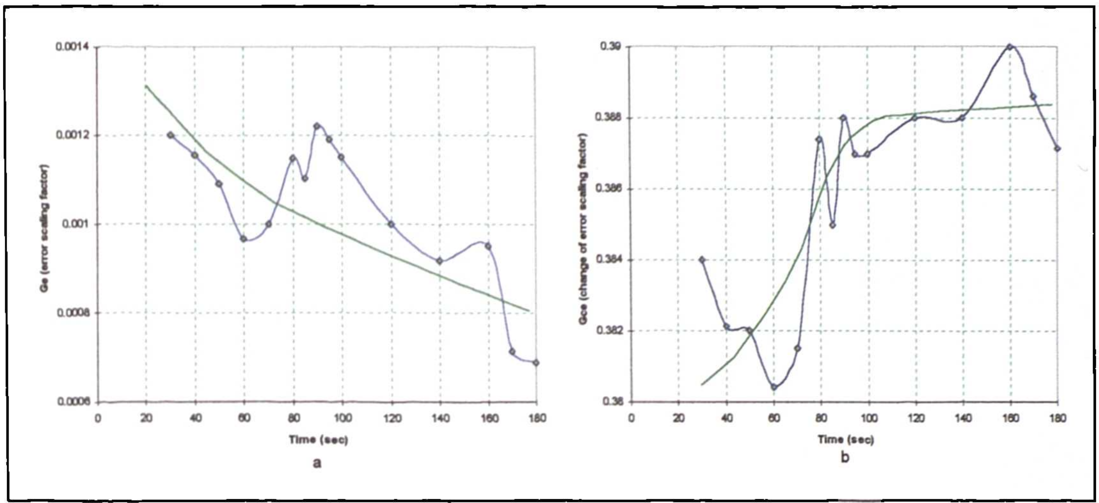


Figure 6.13: Scaling factors calculated based on automatic method🟦; approximation of scaling factors🟩 : a) G_e ;b) G_{ce}.

6.6 ON-LINE SELF-TUNING OF SCALING FACTORS BASED ON AUXILIARY FL CONTROLLER

6.6.1 Self-tuning of scaling factors based on different initial step speed inputs and motor inertia (Method 3)

The structure of the FL speed controller used in this section corresponds to the off-line optimised CPFLC discussed in Section 4.4.2. However, instead of tuning the scaling factors off-line, an on-line method, based on an auxiliary FL mechanism, for tuning of scaling factors is developed. It should be noted that all the three scaling factors of the FL speed controller are now tuned on-line, namely G_e, G_{ce} and G_u.

The structure of the adaptive speed FLC, whose scaling factors are tuned on-line, is shown in Fig. 6.14. Tuning of scaling factors is performed by means of the second, auxiliary FLC. Outputs of the auxiliary FLC are scaling factors for the speed controller. Its inputs are inertia of the drive and the reference speed setting.

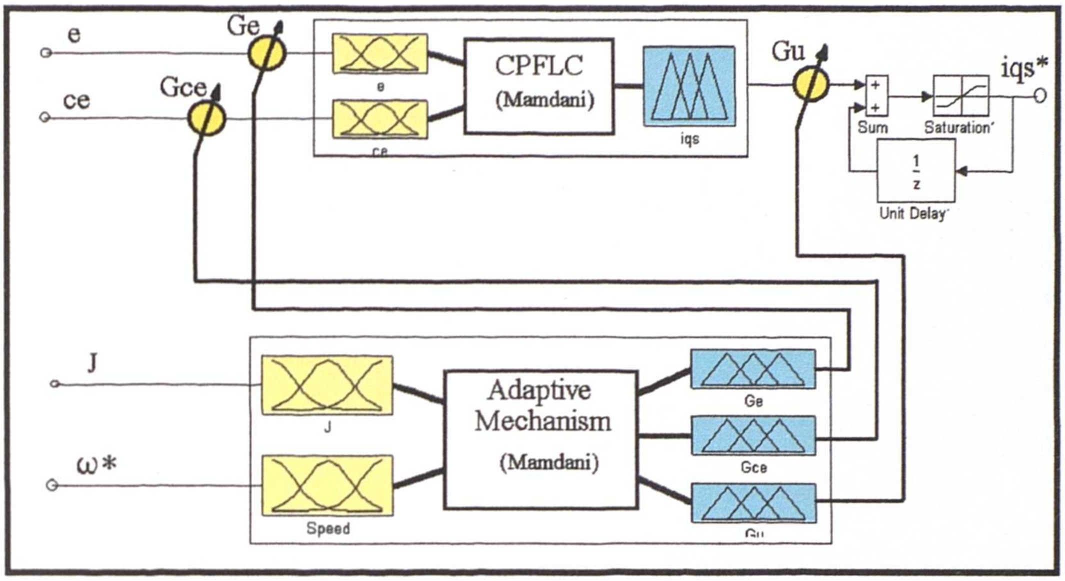


Figure 6.14: Self-tuning of speed FLC scaling factors with an auxiliary FLC.

It is possible to use intuition to write down a rule-base that characterises how the scaling factors should be tuned based on certain operating conditions. In this method the design of the auxiliary FLC for various operating conditions is not necessary as it may be possible to simply specify, in a heuristic fashion, how the scaling factors should be changed. The rule-base is used to schedule the scaling factors of the adaptive speed FLC based on the knowledge of step speed command and motor inertia. It is noted that these inputs are not normally used in the design of a CPFLC. The fuzzy rule base of the auxiliary FLC is constructed by considering three cases: inertia equal to rated, inertia larger than rated and inertia smaller than rated.

1. Inertia equal to rated (J = Jo)

   The inputs of the auxiliary FLC are the initial step speed command and inertia that is equal to rated. The scaling factors illustrated in Fig. 6.1 and 6.2 can be described with four membership functions, namely ZE, PS, PM and PL. The same applies to the speed command (Fig. 6.15a). A triangular shape membership function is chosen to represent the rated inertia, namely as PS (Fig. 6.15b). Finally, four rules are derived based on this description:

Rule 1: if (J is PS) and (speed is ZE) then (G_e is PL) and (G_{ce} is ZE)

Rule 2: if (J is PS) and (speed is PS) then (G_e is PM) and (G_{ce} is PS)

Rule 3: if (J is PS) and (speed is PM) then (G_e is PS) and (G_{ce} is PM)

Rule 4: if (J is PS) and (speed is PL) then (G_e is ZE) and (G_{ce} is PL)

The control surface of the auxiliary FLC for error and change of error scaling factors is depicted in Fig. 6.15c,d.

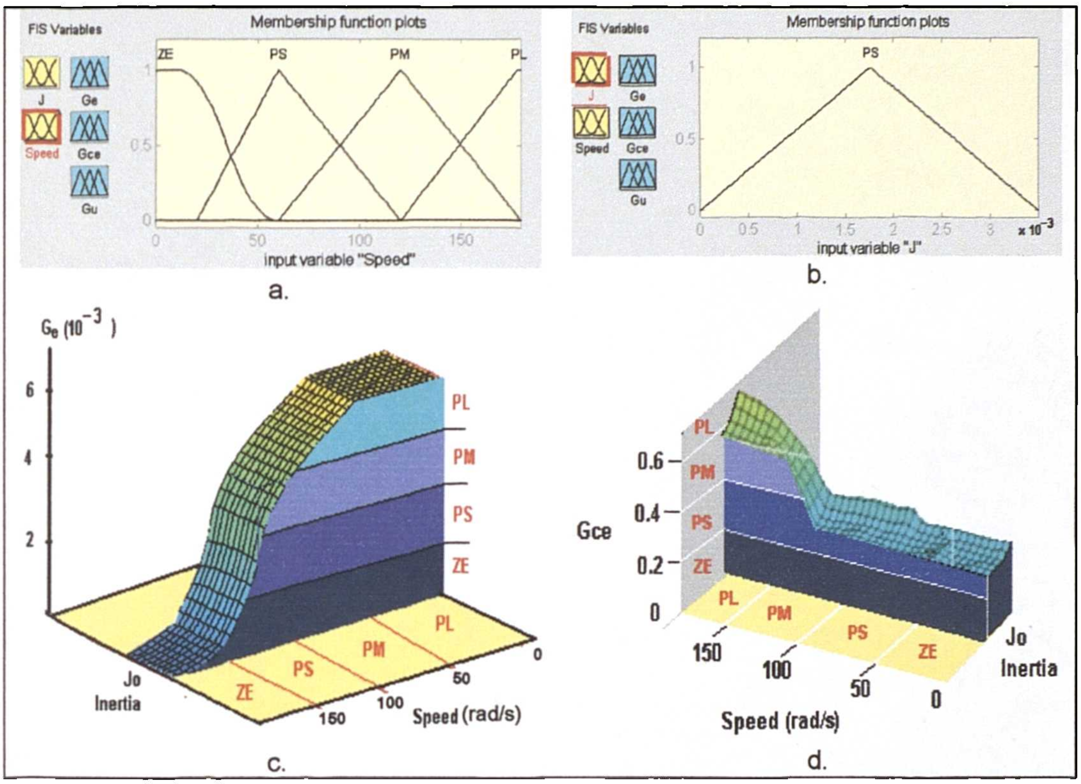


Figure 6.15: Case study for different initial speed commands and rated inertia: a) membership functions for initial step speed command; b) membership function for rated inertia; c) G_e; d) G_{ce}.

2. Inertia larger than rated (J = 2Jo)

The procedure used in case 1 is repeated here, with the inertia assumed to be PL or J > Jo (Fig. 6.16a), while initial step speed command is still the same as in case 1. The same number of rules are produced:

Rule 1: if (J is PL) and (speed is ZE) then (G_e is PL and G_{ce} is PS)

Rule 2: if (J is PL) and (speed is PS) then (G_e is PM and G_{ce} is PM)

Rule 3: if (J is PL) and (speed is PM) then (G_e is PS and G_{ce} is PL)

Rule 4: if (J is PL) and (speed is PL) then (G_e is ZE and G_{ce} is PL)

The control surface of the auxiliary FLC for error and change of error scaling factors is depicted in Fig. 6.16c,d.

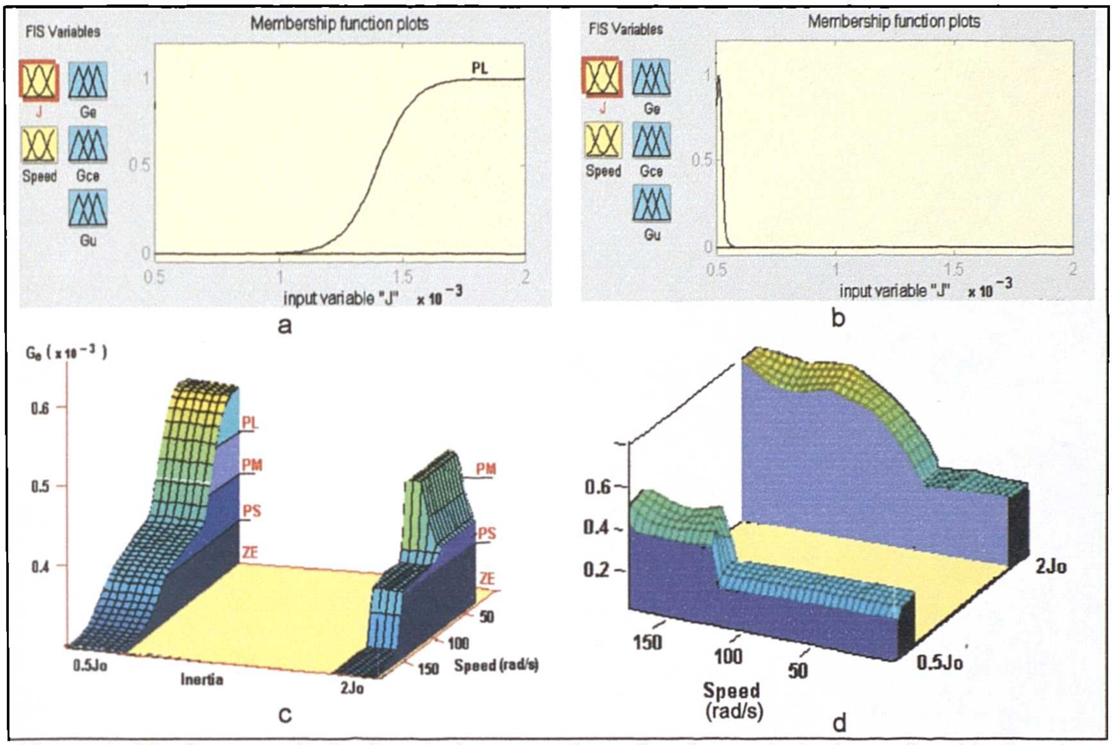


Figure 6.16: Case study for inertia larger and smaller than rated: a) membership function for inertia larger than rated; b) membership function for inertia smaller than rated; c) G_e; d) G_{ce}.

3. Inertia smaller than rated (J = 0.5Jo)

The procedure to derive the fuzzy rules for inertia smaller than rated (J < Jo) is basically the same as in cases 1 and 2. The only difference is that the inertia is assumed to be smaller than rated and so it has ZE membership function (Fig. 6.16b). It is noted that inertia has significant influence on the motor speed response. A motor which has small inertia can accelerate faster than the motor which has large inertia. On the other hand, the same amount of control action will produce satisfactory performance for the design case (Fig. 5.1, Section 5.4.1) but slow speed response for inertia larger than rated (Fig. 5.13, Section 5.5.1) and sluggish speed response when inertia is smaller than rated (Fig. 5.14, Section 5.5.1). A small control action can be obtained by decreasing the K_p or K_i (Section 3.8.1). The same effect can be achieved by decreasing the G_e or increasing G_{ce} (Section 4.5.3). It is noted that the influence of G_e is more significant than G_{ce} (Section 4.5.3). Therefore, in order to improve the speed response and to maintain a fast rise time, G_e has to be decreased while G_{ce} has to be increased (Table 6.7). The control action surface for inertia smaller than rated is given in Fig. 6.16 and is based on the final rules as follows:

Rule 1: if (J is ZE) and (speed is ZE) then (G_e is PM and G_{ce} is ZE)
Rule 2: if (J is ZE) and (speed is PS) then (G_e is PM and G_{ce} is ZE)
Rule 3: if (J is ZE) and (speed is PM) then (G_e is PS and G_{ce} is PS)
Rule 4: if (J is ZE) and (speed is PL) then (G_e is ZE and G_{ce} is PS)

Initial step speed command	Jo	J > Jo (J = 2Jo)	J < Jo (J = 0.5Jo)
ZE	PL	PL	PM
PS	PM	PM	PM
PM	PS	PS	PS
PL	ZE	ZE	ZE

a)

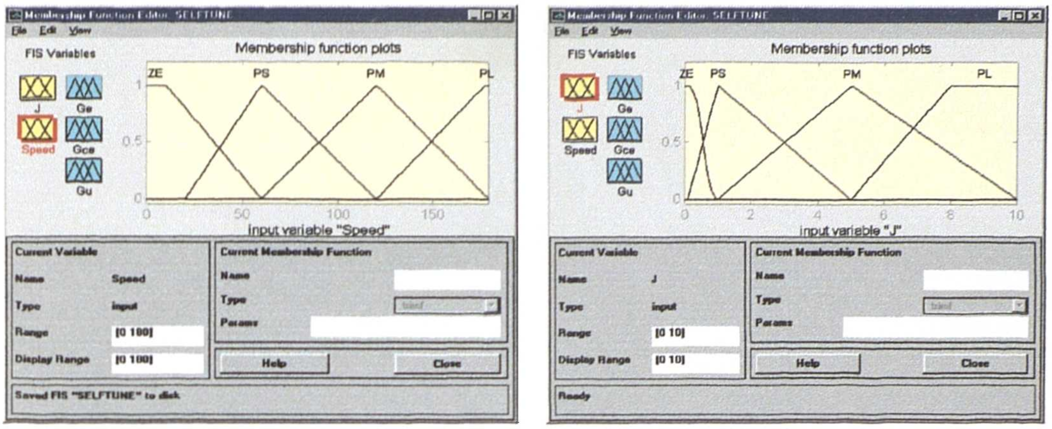
Initial step speed command	Jo	J > Jo (J = 2Jo)	J < Jo (J = 0.5Jo)
ZE	ZE	ZE	ZE
PS	PS	PM	ZE
PM	PM	PL	PS
PL	PL	PL	PS

b)

Table 6.7: Scaling factors for different operating conditions: a) G_e ; b) G_{ce}

The solution for the self-tuning of scaling factors based on auxiliary FL controller is given next based on these case study. The ultimate goal is therefore to adapt scaling factors for both variable reference speed setting and inertia variation. In the research, the variation of inertia is assumed to be ± 10 times of the rated inertia. Inertia is chosen as the input variable because it significantly influences performance of the drive. A very small inertia will result in a large overshoot and sluggish speed response of the drive. However, the total inertia is very difficult to measure directly on-line. There are several techniques that evaluate inertia indirectly. In [Chiricozzi *et al*,1996] inertia is estimated using fuzzy logic. In this research inertia is evaluated on the basis of mechanical motion equation (3.25). The estimation of inertia is performed on the basis of T_e*, $\Delta\omega$ and Δt. Load torque is estimated on the basis of known q-axis stator current command. During a large transient, when the torque command is at its limit, the inertia is evaluated on the basis of the known time between two speed samples and the measured change in speed. It is noted that the results may be somewhat inaccurate because the actual value of inertia is very small and approximation of differentiation is involved. The rule base of the auxiliary FLC, illustrated in Table 6.8, incorporates inertia values from 0.1Jo to 10 Jo. However, only the inertia smaller than the one used in the design case will be investigated (Section 4.4). The reason for this is that, as shown in Chapter 5, the FL speed controller exhibits an excellent performance when inertia is higher than rated. Four membership functions are used to represent the inertia and speed reference setting, which are the input variables (Fig. 6.17). The rule base therefore consists of 16 rules.

The final controller control surface is plotted in 3D view in Fig. 6.18. It is evident that G_e increases and G_{ce} decreases when the initial speed command decreases, while the output scaling factor decreases when the inertia decreases. A large output scaling factor can only be applied when the inertia is larger than in the design case. The behaviour of this self-tuning FL speed controller is investigated in detail in Section 6.7, where a number of simulation results are presented. As will be shown in Section 6.7, the performance of the self-tuning FL speed controller is superior to the one obtainable with off-line optimised CPFLC [Ibrahim *et al*, 1998b].



Current Variable

Name Speed

Type input

Range [0 100]

Display Range [0 100]

Current Membership Function

Name

Type

Params

Saved FIS "SELFTUNE" to disk

a.

Current Variable

Name J

Type input

Range [0 10]

Display Range [0 10]

Current Membership Function

Name

Type

Params

Ready

b.

Figure 6.17: Membership functions of the auxiliary FLC: a) reference speed membership functions; b) inertia membership functions.

Motor Inertia

	ZE	**PS**	**PM**	**PL**
REF SPEED	0 1J_o<J<Jo	0 8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	PM	PM	PL	PL
10<ω*<80	PM	PM	PM	PM
40<ω*<120	PS	PS	PS	PS
80<ω*<180	ZE	ZE	ZE	ZE

Table 6.8a: Error scaling factor.

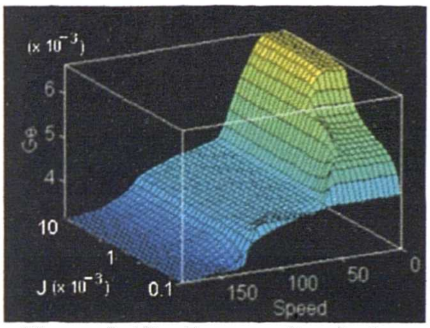


Figure 6.18a: Error scaling factor.

	ZE	**PS**	**PM**	**PL**
REF SPEED	0 1J <J<Jo	0 8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	ZE	PS	PM	PL
10<ω*<80	ZE	PS	PM	PL
40<ω*<120	PS	PS	PM	PL
80<ω*<180	PS	PM	PL	PL

Table 6.8b: Change of error scaling factor.

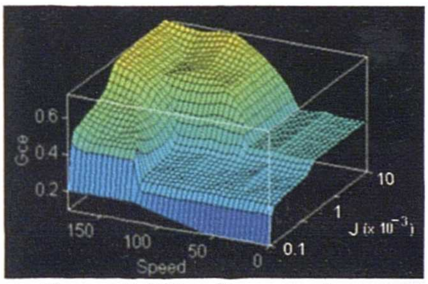


Figure 6.18b: Change of error scaling factor.

	ZE	**PS**	**PM**	**PL**
REF. SPEED	0 1J_o<J<Jo	0.8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	ZE	PS	PM	PM
10<ω*<80	ZE	PS	PM	PM
40<ω*<120	ZE	PS	PM	PM
80<ω*<180	ZE	PS	PM	PM

Table 6.8c: Output scaling factor.

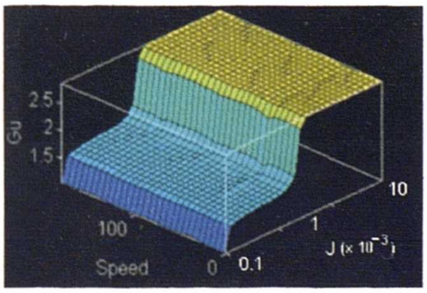


Figure 6.18c: Output scaling factor.

6.6.2 Scaling factor scheduling method for the FL speed controller (Method 4)

The method described in this section relies on an auxiliary FLC and it utilises the non-linear membership functions to perform the on-line scheduling of the scaling factors of a FL speed controller. The block diagram of the scaling factors scheduling method for the FL speed controller is shown in Fig. 6.19. It should be noted that the method developed in this section adjusts scaling factors for variable reference speed setting only and it therefore does not provide compensation for variation in inertia. Furthermore, only input scaling factors are adapted.

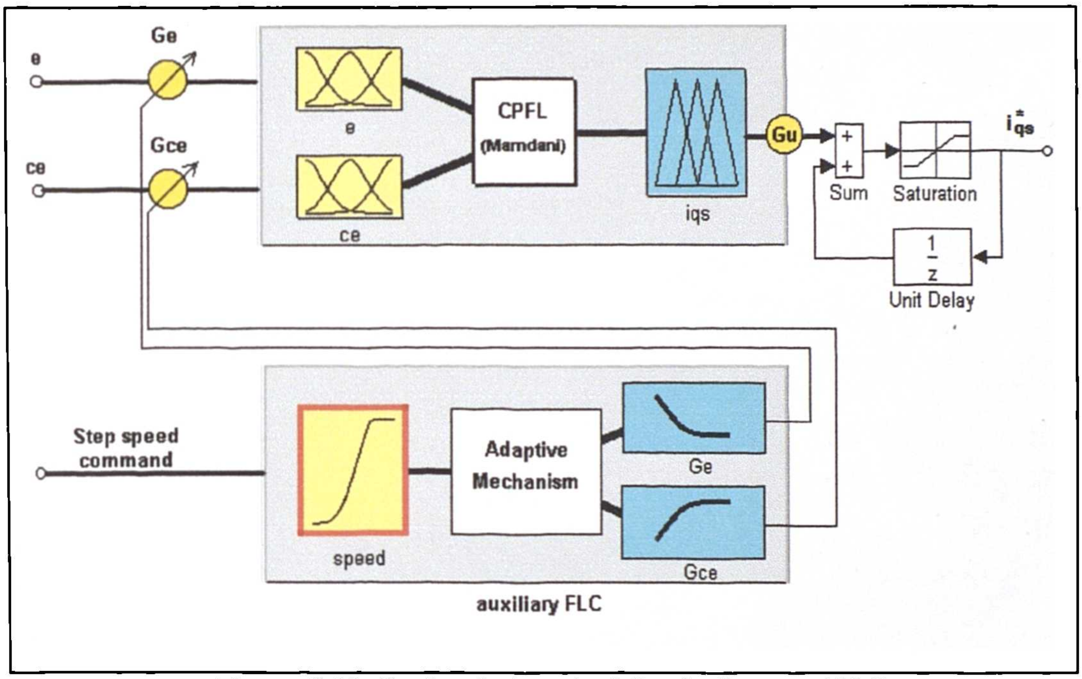


Figure 6.19: Scaling factor scheduling for the speed FLC.

Only one input variable of the auxiliary FLC is considered. The input of the auxiliary FLC is reference speed while its outputs are error and change of error scaling factors. Output scaling factor of the speed FL controller is kept constant at value G_u= 3. Firstly, the scaling factors that produce non-oscillatory response are calculated using the method of Section 6.5 for different cases, i.e. different initial step speed command, large, medium and small. Secondly, the knowledge gained from the first design, i.e. the maximum and minimum limit of scaling factors, is used to form the non-linear membership functions to represent the required scaling factors. In this research, the sigmoidal membership function is used as an initial membership function. There is no specific membership function that has to be used, the objective is that the auxiliary FLC has to produce the required output scaling factors. After tuning of membership functions, membership functions of the auxiliary FL mechanism became as shown in Fig. 6.20. This final design of output membership functions is based on the knowledge obtained in section 6.4 and can be summarised as follows: the input scaling factor is approximately linear for two thirds of the overall speed range, while the remaining one third is non-linear as shown in Figs. 6.1, 6.2. Output of auxiliary FLC obtained in this way is shown in Fig. 6.21a,b. On the other hand, the desired change of scaling factors can also be obtained from automatic tuning method, as discussed in Section 6.5 (blue lines in Fig. 6.13). The goal of this method is to improve the response especially for small step speed command. Final output of auxiliary FLC obtained in this way is shown in Fig. 6.22a,b. This output corresponds to the final outlook of membership functions shown in Fig. 6.20 and this form of auxiliary FLC is used in Section 6.7 in simulation studies.

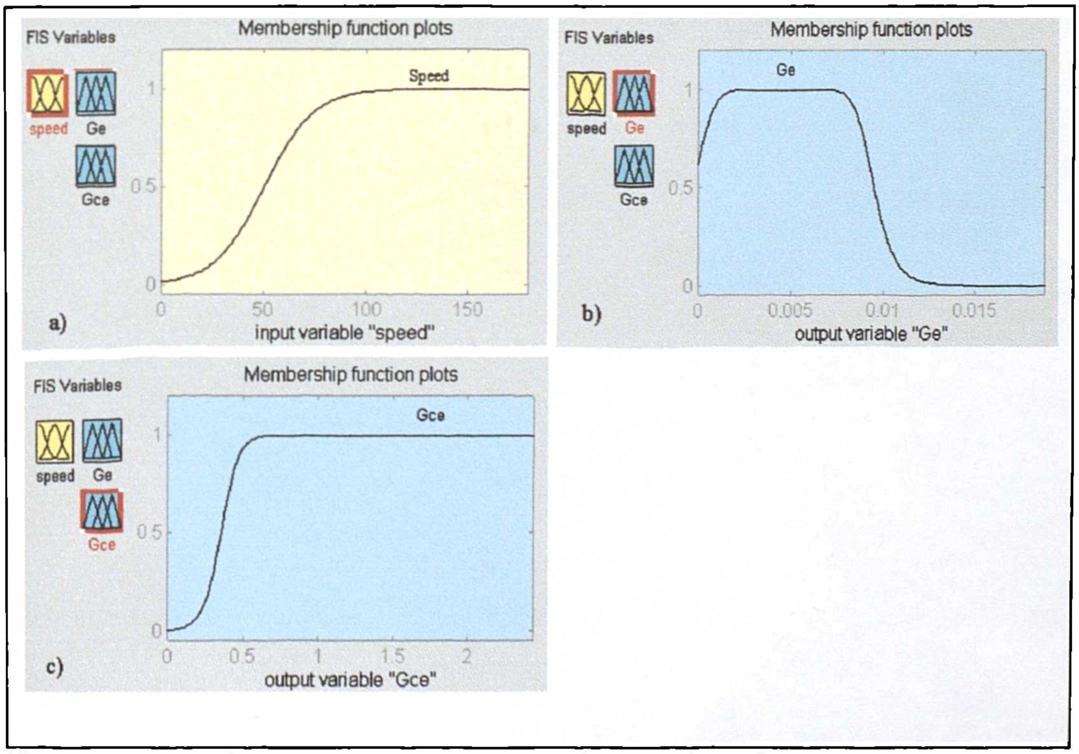


Figure 6.20: Membership functions of the auxiliary FLC: a) input variable membership function; b) membership function of G_e; c) membership function of G_{ce}.

The trial-and-error method to calculate an optimised scaling factor set for every single value of the step speed command is eliminated. It is possible to obtain the optimised scaling factors based on several case studies, as follows. The optimal scaling factors for nominal case, two third of nominal case and one sixth of nominal case are estimated using method 2. The membership functions for scaling factors can then be designed based on this approximation. The pre-calculated scaling factors and look-up table are not required and the auxiliary FL controller is built based on minimum number of rules (only one in this case).

The validity of this technique is investigated by simulation for small step speed commands, one third and one sixth of the rated speed. The results are shown in Fig. 6.23, where speed responses obtained with two CPFLCs (off-line optimised CPFLC and initial design CPFLC of Chapter 4) are included. The large overshoot in speed response is reduced, while the problem of tuning the non-linear scaling factors for the entire speed range is solved as well. Further investigations, undertaken to verify this method for different types of applications, will be based, as already pointed out, on scaling factors shown in Fig. 6.22a.b. The reason for this selection, superior speed response, is evident from Fig. 6.23.

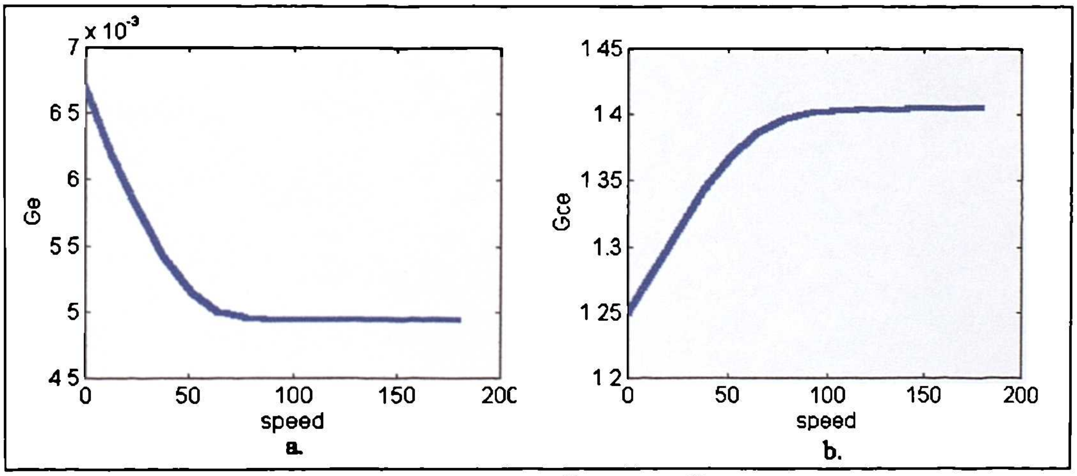


Figure 6.21: Output from auxiliary FLC: a) G_e and b) G_{ce} based on knowledge from method 1.

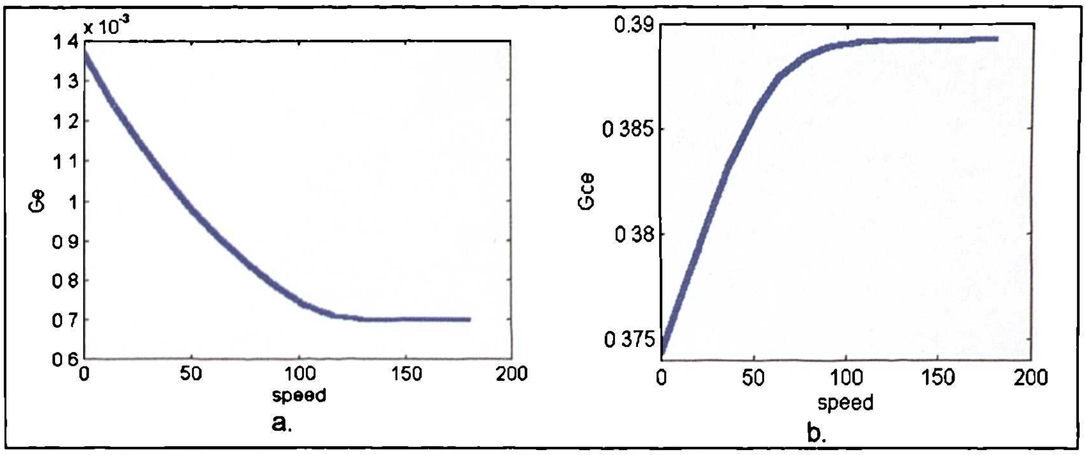


Figure 6.22: Output from auxiliary FLC: a) G_e and b) G_{ce} based on knowledge gained from method 2.

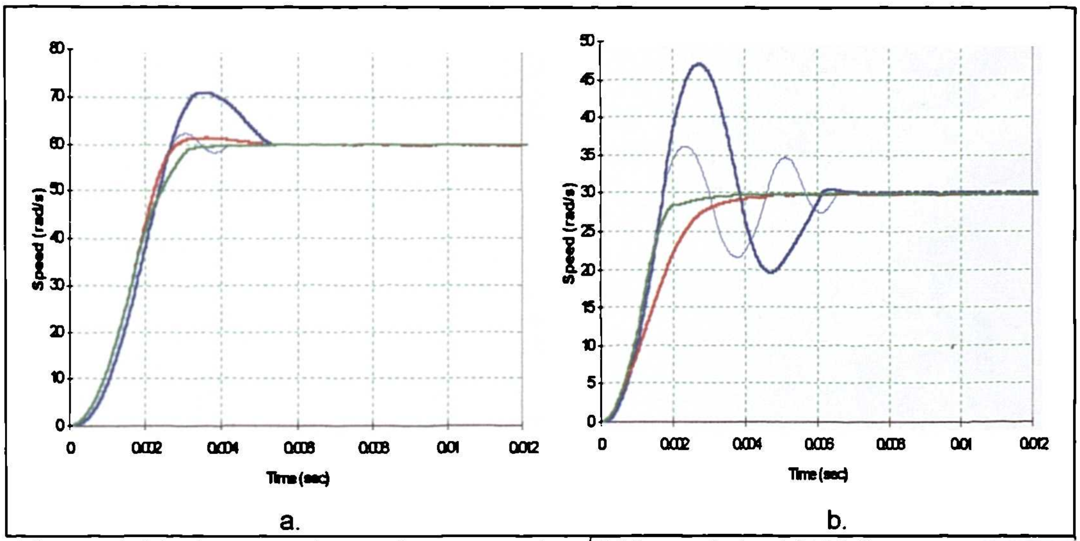


Figure 6.23: A comparison of the speed response: a), b) small speed command.

Off-line optimised CPFLC	—
Initial design of CPFLC	—
Method 4: based on method 1	—
Method 4: based on method 2	—

6.7 SIMULATION AND ANALYSIS OF SELF-TUNING FLCs

In this section, the comparison is made between off-line optimised CPFLC (for zero overshoot) and various procedures developed in this Chapter for automatic tuning and self-tuning. The discussion involves the controller parameters such as membership functions, number of rules and scaling factors. The evaluation of the controller performance is based on the dynamic response characteristic, steady state accuracy and the sensitivity to the inertia variation. The increase in inertia is not considered in this discussion because the FLC has interesting and very good behaviour in this case

(Chapter 5). In normal practice, the controller is tuned for the rated operating conditions. When the operating point is far from the design case, i.e. small step speed command, the different set of controller parameters is required. One can therefore expect that a FL speed controller with variable scaling factors will exhibit a superior performance to the one obtained with off-line optimised CPFLC.

6.7.1 Controller parameters

Table 6.9 gives the summary of the controller parameters that have been used in the simulation studies. In adaptive self-tuning, the auxiliary FLC is used to tune the scaling factors, while all the other parameters of the FL controller are kept constant. Forty nine rules have been used to design the off-line optimised CPFLC. The most time consuming is the tuning of the off-line optimised CPFLC. Further analysis will show that the good behaviour of the adaptive controller is not only dependent on how well tuned is the off-line optimised CPFLC, but also on the selection of the tuning control strategies. Four types of adaptive mechanisms which have been developed in the previous sections are analysed and compared in what follows. These are:

A. Speed FL controller based on off-line optimised CPFLC (Chapter 4) with scaling factor tuning according to Method 1 (Section 6.4).

B. Speed FL controller based on off-line optimised CPFLC (Chapter 4) with scaling factor tuning according to Method 2 (Section 6.5).

C. Speed FL controller based on off-line optimised CPFLC (Chapter 4) with scaling factor tuning according to Method 3 (Section 6.6.1).

D. Speed FL controller based on off-line optimised CPFLC with scaling factor tuning according to Method 4 (Section 6.6.2).

Speed controller Type	Variables		Membership functions		No. of rules
	input	output	inputs	output	initial rule base
Off-line optimised CPFLC	2	1	7 triangular & tuned	7 triangular & tuned	49
A. speed FLC	2	1	7 triangular & tuned	7 triangular & tuned	49
B. Speed FLC	2	1	7 triangular & tuned	7 triangular & tuned	49
Auxiliary FLC	1	2	4 triangular	4 singleton	4
C. Speed FLC	2	1	7 triangular & tuned	7 triangular & tuned	49
Auxiliary FLC	2	3	4 triangular & tuned	4 triangular & tuned	16
D. Speed FLC	2	1	7 triangular & tuned	7 triangular & tuned	49
Auxiliary FLC	1	2	1 sigmoidal & tuned	1 sigmoidal & tuned	1

Table 6.9: Summary of parameters of controllers used in the simulation.

6.7.2 Sensitivity to initial speed command change

The comparison of the adaptive controller sensitivity to initial step speed command is made based on the performance obtained from simulation. In all the simulation studies three adaptive systems based on the auxiliary FLC, namely B, C and D, are analysed. The reason for omission of the adaptive method A from further analysis is that it is not based on an auxiliary FLC. It is noted that the scope of the research is to develop a tuning mechanism based on fuzzy controller. Apart from that, it adjusts only one scaling factor (G_e). The drive speed response to the step speed command is illustrated in Fig. 6.24. Four different reference speed settings, ω^* = 180, 120, 60 and 30 rad/s, are considered. In each case five traces are shown, one obtained with initial design of CPFLC, one obtained with off-line optimised CPFLC and three obtained with the three of the listed four types of adaptive FL speed controllers, namely B, C and D.

Figure 6.24a. shows that the off-line optimised speed CPFLC gives the best response to rated speed command (the case for which manual optimal adjustment was performed). Speed response of all the adaptive FLCs is slightly worse, while initial design of the CPFLC yields inferior speed response. A similar situation is observed for the speed command of 120 rad/s; however Fig. 6.24b. shows that the optimised CPFLC now gives a very small, but nevertheless undesirable, overshoot, which does not take place with any of the adaptive FLCs. As the speed reference is further lowered (Fig. 6.24c,d), the speed response of the off-line optimised CPFLC further deteriorates: overshoot increases and becomes over 6 rad/s for 30 rad/s speed command. Furthermore, settling time increases as well, as the speed response becomes more and more oscillatory. The behaviour of the "initial design" CPFLC is the worst for all the speed commands and this type of CPFLC is therefore not considered further on. As far as adaptive FLCs are concerned, they yield desired aperiodic response at very low speed command (30 rad/s) and at 60 rad/s speed command. Figure 6.24 thus verifies the superiority of the adaptive FLCs when compared to both CPFLCs: they are capable of realising an aperiodic speed response at all speed settings, with very short settling time.

Cases considered in Fig. 6.24 all apply to stepping the speed reference from zero up to a certain value. In order to verify the ability of the adaptive speed FLCs to maintain a required aperiodic speed response when speed reference changes in a step-wise manner from an initial speed different from zero, the following two sequences of reference speed variation are investigated:

i) zero to 180 rad/s, 180 to 150 rad/s and 150 to 20 rad/s (Fig. 6.25a),

ii)zero to 30 rad/s, 30 to 120 rad/s and 120 to 180 rad/s (Fig. 6.25b).

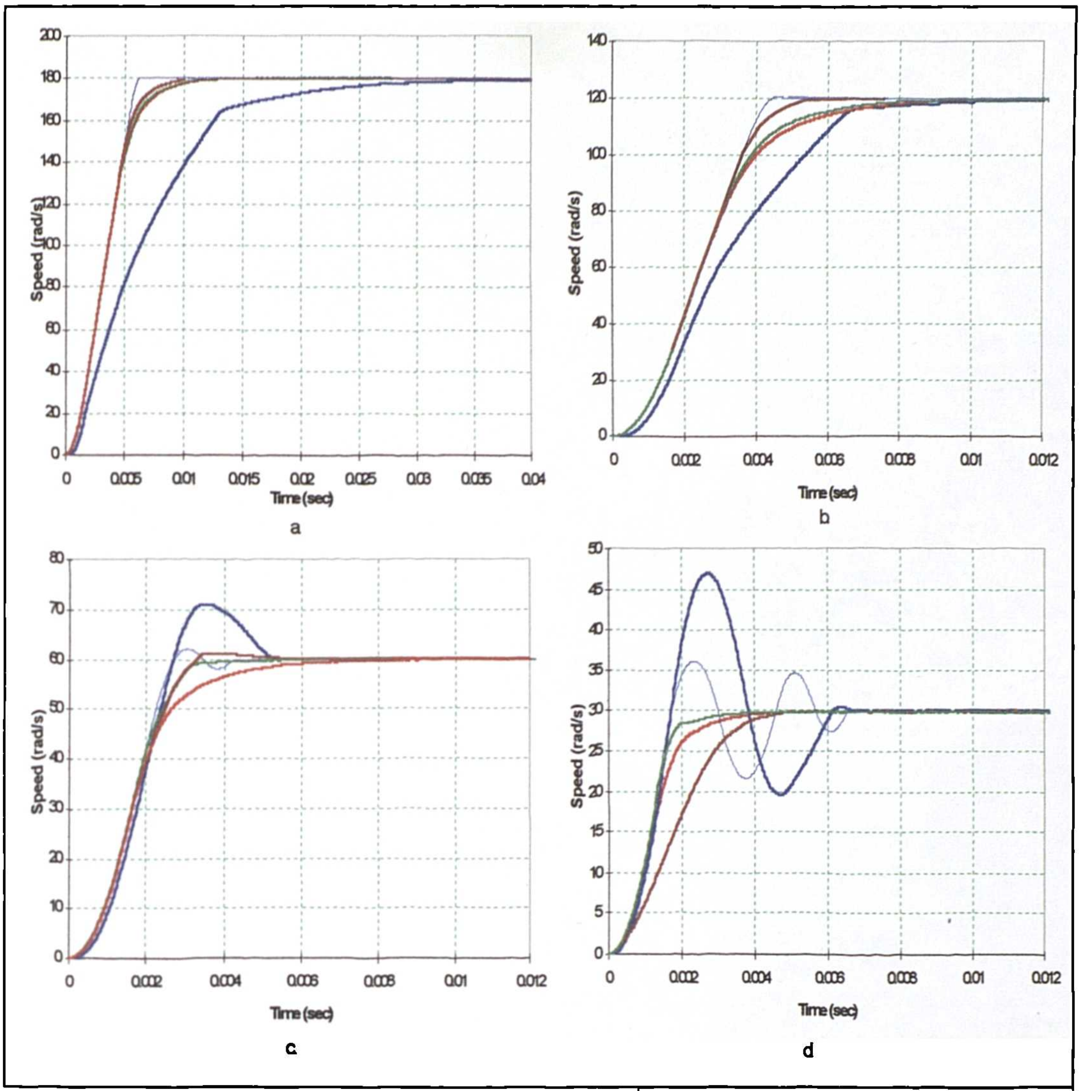


Figure 6.24: A comparison of the speed response for different speed controllers: a) large speed command; b) medium speed command; c), d) small speed command.

Off-line optimised CPFLC	—
Initial design of CPFLC	—
B	—
C	—
D	—

The results obtained with adaptive speed FLCs are compared with those obtained with off-line optimised CPFLC. Figure 6.25 clearly indicates superiority of the adaptive FLCs over the off-line optimised CPFLC and shows that adaptive FLCs give desired aperiodic speed response for all the considered transients. All the adaptive speed FLCs have demonstrated similar performance in terms of settling and rise time, and with respect to sensitivity to the initial step speed command change.

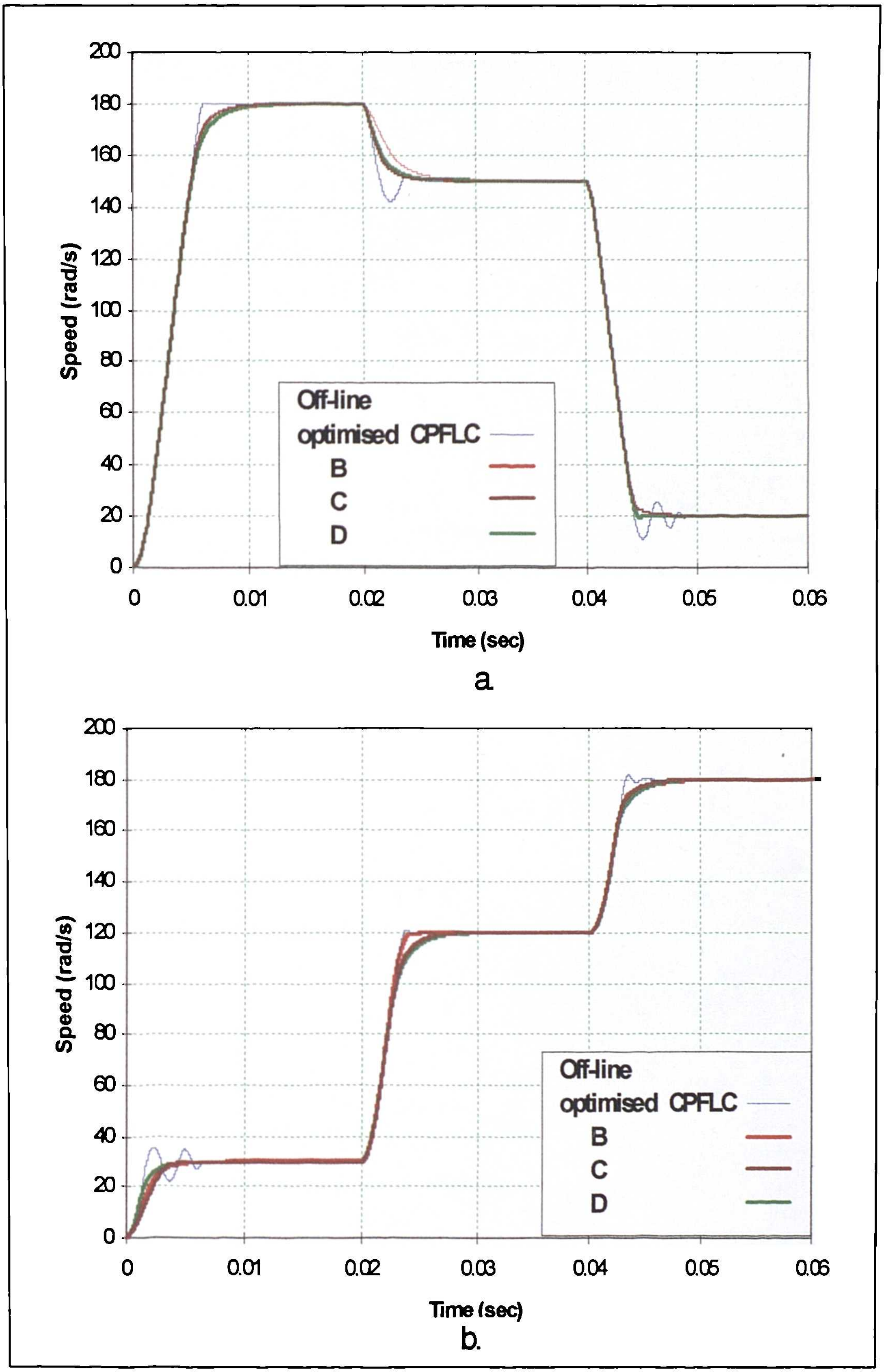


Figure 6.25: Speed response with different types of adaptive speed FLCs and with off-line optimised CPFLC: a) zero to 180 rad/s, 180 to 150 rad/s and 150 to 20 rad/s; b) zero to 30 rad/s, 30 to 120 rad/s and 120 to 180 rad/s.

6.7.3 Robustness to total inertia change

In this section, the robustness of the controller with respect to the total inertia change is investigated. Off-line optimised CPFLC and self-tuning FLC (Method 3, controller labelled C), which was specifically designed to compensate the variation of the inertia and initial step speed command variation, are compared. The following three sequences of reference speed variation are investigated:

a) zero to 180 rad/s, 180 to 120 rad/s and 120 to 30 rad/s (Fig. 6.26a).

b) zero to 180 rad/s, 180 to 30 rad/s (Fig. 6.26b,c).

c) zero to 30 rad/s, 30 to 120 rad/s and 120 to 180 rad/s (Fig. 6.26c).

Corresponding profiles of inertia variation are included in Fig. 6.26. Inertia is reduced from Jo to 0.1Jo at t = 0.04s for cases a) and b) while it is increased from 0.1Jo to Jo at t = 0.004s in case c).

The blue trace is the response obtained with off-line optimised CPFLC, while the red trace is the response with self-tuning FLC. From Fig. 6.26 it follows that the oscillatory speed response occurs with the off-line optimised CPFLC when the total inertia is reduced from Jo to 0.1Jo (Fig. 6.26a,b). It is interesting to note that the response obtained with the off-line optimised CPFLC is improved when the small inertia is applied during the small step speed command instead of rated inertia (Fig. 6.26c). On the other hand, the self-tuning FLC (Method 3) exhibits superiority in speed response even with the inertia as small as 0.1Jo (Fig. 6.26a,b,c). The actual inertia is estimated on-line, as shown on the right hand side of Fig. 6.26. Although the estimation of inertia is not perfectly accurate, the self-tuning FLC still gives significantly better behaviour than the off-line optimised CPFLC.

6.7.4 Overall controller performance

A global comparison of the system performance can be made based on integral speed error criterion, as mentioned in Chapter 5.

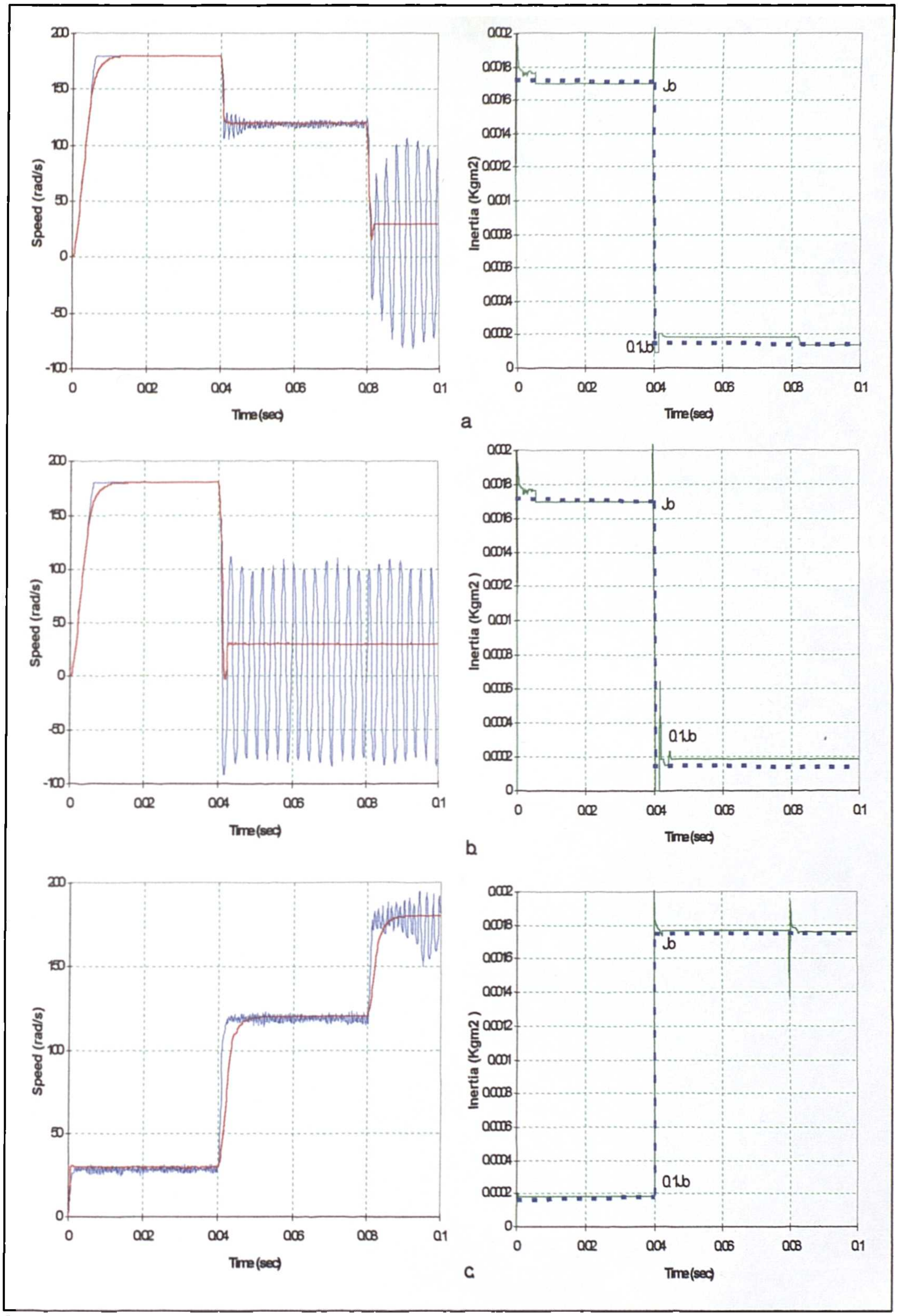


Figure 6.26: Speed response obtained with off-line optimised CPFLC and with self-tuning FLC (Method 3): a) zero to 180 rad/s, 180 to 120 rad/s and 120 to 30 rad/s; b) zero to 180 rad/s, 180 to 30 rad/s; c) zero to 30 rad/s, 30 to 120 rad/s and 120 to 180 rad/s . (🟦 optimised CPFLC, 🟥 self-tuning FLC, 🟩 estimated inertia)

The comparison is made here between the adaptive self-tuning speed FLCs and the off-line optimised CPFLC. The readings are captured from many simulation runs under the same conditions as those established in Chapter 5. Table 6.10 shows the overall performance index in terms of ITAE and IAE for three different adaptive methods. In general, these methods have the performance that is at least as good as the one of the off-line optimised FLC (ITAE = 0.223%) and off-line optimised PI controller (ITAE = 0.232%) for 1.3 rad/s overshoot design (Table 5.1, Chapter 5). The self-tuning method D has demonstrated a better performance with smaller ITAE (0.1988%). The behaviour of error criteria performance indices across the entire speed range can be further investigated by plotting data of Table 6.10, as shown in Fig. 6.27. It follows that the off-line optimised speed CPFLC with zero overshoot design is still the best method with 0.1364% of ITAE performance index. Self-tuning method D exhibits consistent performance for the whole speed range, as is evident from Fig. 6.27. On the other hand, the ITAE performance index for both methods B and C diverges from optimal performance (off-line optimised CPFLC) especially for medium and small speed command. The ITAE and IAE indices in the region of small speed commands are almost the same for all self-tuning methods.

It is difficult to make a viable conclusion based on Table 6.10 and Fig. 6.27 alone. The data apply only to the performance indices and do not reveal much regarding dynamic response and robustness for inertia variation. Therefore, the overall performance of the variable speed drive in terms of dynamic response, error criteria and robustness to inertia variation is summarised in Table 6.11. Further investigation has shown that the small value of ITAE index is not a guarantee of better controller performance, especially when the overshoot and settling time of the speed response are considered. The zero overshoot design of the off-line optimised CPFLC is very poor for the small step speed applications (Section 5.4 and Fig. 6.23, Table 6.11).

No. of samples, sample interval 10 rad/s	ITAE Rated load and inertia			IAE Rated load and inertia		
	Adaptive speed FL controller type			Adaptive speed FL controller type		
	B	C	D	B	C	D
1	0.00065	0.00063	0.0007	0.0246	0.0233	0.0390
2	0.00082	0.00069	0.0009	0.0422	0.0362	0.0460
3	0.00094	0.00078	0.0011	0.0604	0.0530	0.0571
4	0.0011	0.00091	0.0012	0.0818	0.0729	0.0782
5	0.0013	0.0010	0.0014	0.1069	0.0946	0.1020
6	0.0016	0.0011	0.0015	0.1403	0.1206	0.1277
7	0.0017	0.0014	0.0016	0.1664	0.1600	0.1552
8	0.0016	0.0018	0.0017	0.1892	0.2027	0.1865
9	0.0017	0.0022	0.0018	0.2187	0.2458	0.2198
10	0.0019	0.0026	0.0019	0.2586	0.2887	0.2553
11	0.0023	0.0028	0.0022	0.3042	0.3314	0.2958
12	0.0025	0.0031	0.0024	0.3496	0.3767	0.3369
13	0.0028	0.0034	0.0024	0.3966	0.4211	0.3796
14	0.0031	0.0037	0.0026	0.4454	0.4695	0.4279
15	0.0036	0.0039	0.0027	0.5076	0.5159	0.4768
16	0.0042	0.0043	0.0029	0.5730	0.5777	0.5296
17	0.0049	0.0047	0.0033	0.6451	0.6431	0.5891
18	0.0056	0.0054	0.0035	0.7169	0.7140	0.6462
Average of ITAE/IAE	0.235 %	0.2467 %	0.1988 %	29.04 %	29.70 %	27.49 %

Table 6.10: The error criteria performance indices for the overall speed range.

Furthermore, the response in the whole speed range is oscillatory when the total inertia is smaller than the design value (Fig. 6.26, Table 6.11). The rise time and settling time for large speed command for adaptive speed controllers are not much different compared to the off-line optimised CPFLC, while response for small speed command is much better with the adaptive speed controller. Therefore, the main conclusion is that the adaptive self-tuning speed FLCs have shown their good performance for the overall speed range (Figs. 6.24, 6.25, Table 6.11) and also for variation of total inertia (Fig. 6.26, Table 6.11). Adaptive speed controllers, in contrast to the off-line optimised CPFLC, are capable of maintaining the required nature and quality of the speed response over the entire speed region.

By comparing characteristics and behaviour of adaptive speed controllers, it should be noted that self-tuning method 4 might be the best solution, as it is very easy to implement. It has to be observed though that this method adapts the FL speed controller parameters only for variation in the reference speed setting. If compensation in inertia variation is required, the method is not applicable (method 3 is the preferred choice in such a case). Characteristics of method 4 are:

i) small ITAE and IAE performance indices of 0.1988% and 27.49%, respectively.

ii) good performance for the entire speed range.

iii) designed with a minimum number of controller parameters such as membership functions and fuzzy rule base. Increase in the number of parameters used will contribute to the complexity and will lead to longer processing time.

iv) the adaptation of scaling factor is based on the non-linear shape of the controller output surface, which is obtained from three case studies, a) rated step speed command, b) two thirds of the rated step speed command, c) one sixth of the rated step speed command. The scaling factors corresponding to these three operating conditions are obtained based on automatic tuning method 2.

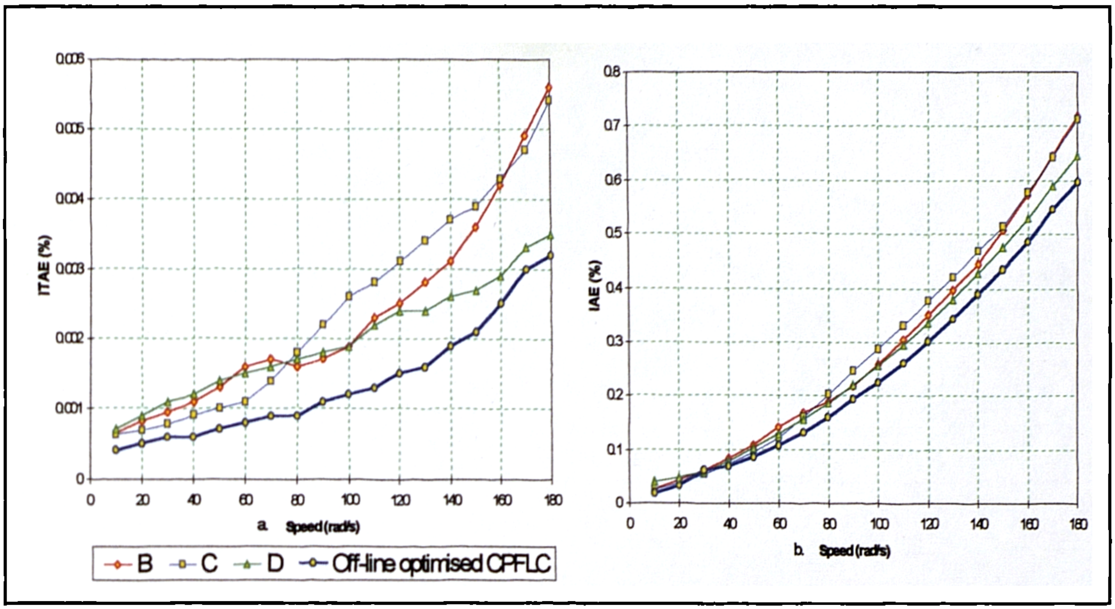


Figure.6.27: Speed error criteria: a) ITAE; b) IAE.

Type of Speed Controller	Dynamic response performance, Jo		Speed error criterion		Variation of inertia
	180 rad/s (large speed command)	30 rad/s (small speed command)	ITAE (%)	IAE (%)	0.1Jo
Off-line optimised CPFLC	aperiodic response, $T_s = 0.00625$ $T_r = 0.0054$	large overshoot 5.15 rad/s with oscillatory response, $T_r = 0.007$	0.1364	not calculated	oscillatory and sluggish response
B	aperiodic response, • $T_s = 0.011$ 1.76 times slower • $T_r = 0.0065$ 1.2 times slower	aperiodic response, • $T_s = 0.004$ 1.75 times faster • $T_r = 0.0025$	0.2350	29.04	N/A
C	aperiodic response, • $T_s = 0.01$ 1.6 times slower • $T_s = 0.006$ 1.1 times slower	aperiodic response, • $T_s = 0.005$ 1.4 times faster • $T_r = 0.0035$	0.2467	29.70	good response with zero overshoot
D	aperiodic response, • $T_s = 0.011$ 1.76 times slower • $T_r =$ 0.0062 1.77 times slower	aperiodic response, • $T_s = 0.0035$ 2 times faster • $T_r = 0.0019$	0.1988	27.49	N/A

Table 6.11: Summary of the overall controller performance (T_s = settling time, T_r = rise time both in seconds).

6.8 SUMMARY

The need for development of self-tuning strategies is at first established by investigating the relationship between the nature of scaling factor variation and the desired quality of step speed response, over the entire speed range. It is shown that the relationship is non-linear. The scaling factors are tuned manually until a minimum settling time with zero overshoot is obtained. The graph of scaling factor variation versus speed is then plotted. Two possible approaches to adaptation are employed next and four types of self-tuning mechanisms are developed and investigated based on simulation procedures. Method 1 is developed to automate the tuning of scaling factors based on IAE and overshoot criterion. Method 2 is designed to automate the tuning of scaling factors using an auxiliary FL controller and data regarding speed response overshoot. In method 3 the on-line self-tuning of scaling factors is performed based on a different initial step speed command inputs and motor inertia. Method 4 is based on knowledge gained from the previous designs. The outputs from the auxiliary FLC of methods 2, 3 and 4 are used to adjust the input scaling factors of the FL speed controller, whose structure corresponds to the off-line optimised CPFLC. Method 3 is specifically designed to compensate the variation of both step speed command and the variation of total inertia and adjusts the output scaling factor as well. The simulation study is performed with different speed command profiles and variation in inertia. Further investigation is made to compare the adaptive controllers with the off-line optimised CPFL speed controller with regard to parameters used in the controller design, overshoot in speed response, settling and rise time, integral speed error criteria, robustness with respect to variation of step speed command and variation of total inertia. The study based on integral speed error criteria is then performed and graphs of ITAE and IAE are plotted for the entire speed range. The overall performance for three types of adaptive self-tuning FLCs is finally discussed and conclusions regarding their applicability for different cases are drawn.

CHAPTER 7

CONCLUSION

7.1 SUMMARY

Since the first applications of the FL control in industrial processes in the early eighties, a lot of research has been carried out for different types of plants. Fuzzy logic can be used to control a highly non-linear process or a process that is very difficult to model because of the unknown plant parameters. It is robust, yields good performance and requires no mathematical model. Recently, applications of fuzzy logic control in variable speed drives have attracted a lot of interest. However, good features of fuzzy logic can be offset by the need to apply a trial-and-error approach in the design, since a systematic design method does not exist at present. Further difficulty is that many parameters need to tuned (number of rules, the width, shape and the number of membership functions, and the scaling factors). Finally, due to rather complex structure, FL controller is computationally intensive and the executable speed of control algorithm in on-line applications is therefore limited. Summarising, design of a FL controller is very time-consuming and difficulties in on-line implementation are numerous.

The SPMSM, hysteresis current controllers, voltage source inverter, speed sensor, and rotor flux oriented controller are all modelled using Simulink. The main reason why Simulink is used is that it is user friendly and it supports built-in function blocks (e.g. differentiator, integrator, multiplier, trigonometric functions, relay, S-functions, etc.), so that it can be used to form the complete drive model with little effort. Furthermore, it can be linked to other toolboxes such as Fuzzy logic toolbox, Image processing toolbox, Neural network toolbox, Control toolbox etc. With available motor data, the SPMSM drive was modelled and simulated using simulation programme developed in a short period of time. Next, two different types of speed controllers, namely PI and FL speed controller, are designed. The design of the PI speed controller with anti-windup was easy, fast and straight forward, compared to the FL speed controller design. The selection of the FL controller parameters is much more flexible. Normally, some experience is required to design the FL controller. The development of the first prototype of the FL speed controller took a long period of time.

At the next stage an attempt is made to evaluate the drive performance when PI and FL speed control are used. Approach used in this investigation is based on comparison of the FL speed controller to the conventional PI speed controller. To guarantee the validity of the results, both controllers are designed based on known motor rated values, and are initially tuned to yield the same aperiodic speed response with minimum settling time to the application of the rated speed command under no-load conditions. The configuration of the drive under test (torque current command limit, computational and sampling time, DC link voltage and hysteresis band of the hysteresis current controllers) is the same in all the cases. The PI speed control is initially designed using Zingler-Nichols method, while the final setting is based on the trial-and-error manual optimisation until aperiodic speed response is obtained with the fastest settling time. The tuning process is simple because only two parameters need to be tuned. For FL speed controller, the impact of variation of the controller parameters on the response can be studied by changing one parameter while other parameters are kept constant. Tuning the fuzzy logic speed controller is very tedious due to a large number of parameters involved. The tuning procedure can be simplified by fixing some parameters (e.g. number of rules, membership functions and defuzzification). The three most important parameters were considered: fuzzy rules, width of membership functions and scaling factors. A specific fuzzy rule influences a specific operating region, while membership functions and scaling factors have global influence on the overall drive performance. In general, the rules can be classified into four main groups. The first group covers the vicinity of steady state region, while the other three groups are related to positive and negative speed command regions. A specific fuzzy rule in a certain region can be tuned to improve operation in a certain operating point. Further tuning of fuzzy rules can be based on this approach. To simplify the design procedure, the initial rules and membership functions are based on those commonly used by other researchers. Seven triangular membership functions of identical width and with symmetrical peak position are used in this study. The triangular shape of membership functions is used because it can be represented by linear equations. This leads to reduced processing time in real time implementation. Determination of scaling factors is discussed next. However, like with fuzzy rules and membership functions, there is not a specific technique that can be used to calculate the initial scaling factors and in practice the trial-and-error approach is normally applied to determine the scaling factors. In this research, two techniques were proposed to calculate the initial values for scaling factors. The first technique is termed 'design case' and the initial scaling factors are calculated based on known motor data. The second technique is based on correlation of FLC scaling factors to the proportional and integral gains of the PI speed controller. The FL speed controller designed based on these premises is termed as 'standard FLC'. The speed response produced by this controller is aperiodic with long settling time.

In order to obtain the desired performance with shorter rise time and settling time and zero overshoot, an optimised FL controller is designed. The controller structure, initial rules, membership functions and scaling factors are based on the standard FLC. However, fuzzy rules, membership functions and scaling factors are further fine-tuned. It is noted that drive performance can be improved by tuning the fuzzy rules and membership functions. Further improvement can be achieved by tuning the scaling factors. The influence of the membership functions on the drive behaviour is at first investigated. The widths and peak positions of the triangular membership functions are adjusted by increasing and decreasing the width, with simultaneous movement of their peaks toward zero position and away from zero position. Tuning the width and moving the peak value position of the membership functions of speed error towards the zero, or tuning away from zero for the change in speed error membership functions will cause the speed controller to be more sensitive to a small change in speed error and produce a large control action. On the other hand, a smaller control action is produced when the membership functions of speed error are tuned away from zero, or when change in speed error membership functions are tuned towards zero. The same type of study is repeated for scaling factors. By keeping the sampling time and output scaling factor constant, the error and change of error scaling factors can be analysed and correlated more clearly. It is noted that increasing the error scaling factor of FL speed controller is similar to increasing the proportional gain of PI controller. For fuzzy logic speed controller, very long settling time is present if very small error scaling factor is used. On the other hand, decrease in the change of error scaling factor will have the same effect as an increase in integral gain of PI controller. The simulation results have shown that small rise time of speed response can be obtained if the change of error scaling factor is small. However, the speed response will overshoot if the change of error scaling factor is too small. The characteristic of the FL controller produced based on this procedure is more non-linear than for standard FLC, with better behaviour. It is shown that better performance can be obtained by properly tuning the FL controller. This asks for a long development time though. It is interesting to note that response produced by off-line optimised PI speed controller is still as good as the response produced by off-line (manually) optimised FL speed controller, as long as the response for the operating point used in the design (rated step speed command, under no load conditions) is under consideration. The results are however different if different operating conditions are considered. The conclusion of this study is that the error scaling factor is more dominant than the change of error scaling factor. Furthermore, tuning the scaling factors is easier to implement on-line than tuning the rules and membership functions. This is one of the main reasons why adaptation of the scaling factors was investigated for self-tuning FL controllers.

In general, FL controller can be evaluated by comparing its behaviour with other conventional controllers such as PI or PID. The comparison can be based on different criteria such as design simplicity, quality of response and implementation requirements. Approach used here is to compare the SPMSM drive controlled by PI and FL for different types of transients. These are different step speed commands, variation of inertia, and load disturbance. Two different methods are used to analyse the results. The first method is based on graphical comparison, where the transients are plotted, zoomed and compared. The overshoot/undershoot, settling time and restoration time are measured to indicate the quality of the step speed response and the properties of the controller in terms of load rejection capability. In the second method, the performance indices such as ITAE and IAE are used to measure the controller performance. This procedure is done not only for a single operating point but for the overall speed range. It is noted that the results from the simulation studies are only valid for the controller that has been designed based on known motor rated values and for the given operating point. The behaviour is different if the controller is designed for a different operating point. The same controller with different parameters will produce different response and different behaviour. Controller that has high gains will produce fast control actions and as a result the drive can be accelerated more quickly. The load rejection capability can be also improved by using higher gains. This is the reason why two designs are considered in this research, namely zero overshoot design and 1.3 rad/s overshoot design. The gain used in the second design is larger than in the first design and this causes small overshoot in response to rated step speed reference. The desired performance only results when the set point corresponds to or is close to the design case operating condition. The impact of other operating points away from the design case is investigated by applying a series of different step speed commands. The results show that the performance of both controllers degrade in terms of overshoot and settling time. However, the FL speed controller has demonstrated a better behaviour in at least two thirds of the whole speed range, compared to PI speed controller. The FL speed controller has better properties with regard to load rejection compared to PI speed controller.

Both controllers behave like linear controllers. Therefore the calculated controller parameters, based on a certain operating condition, are not appropriate for other operating points. Thus, in order to maintain the desired performance under all operating conditions, it is necessary to employ some kind of adaptive FL controller, whose parameters are tuned in accordance with specific operating point. As the most important FLC parameters, that have the greatest impact on the overall system behaviour, are the scaling factors, a number of procedures for adaptive scaling factors tuning are developed. It is possible to use trial-and-error method to determine the most suitable scaling factors for each operating point. However, the procedure can be simplified by developing the automatic tuning mechanism to calculate the suitable scaling factors automatically. This procedure can be based on measuring overshoot of step speed response or it can be based on integral error criteria. It is shown that it is possible to develop adaptive FL controller based on scaling factor scheduling. The results show that required scaling factors for the whole speed range have to exhibit non-linear behaviour if the desired performance is to be maintained. The error scaling factor has to increase while the change of error has to decrease from their original values when small step speed command is applied. Required variation of the scaling factors, obtained in the described way, can be built in the structure of the FL controller and is essentially off-line adaptation.

The research has also shown that the performance of the drive is affected by variation of inertia. The negative effects of inertia variation can be compensated by employing fuzzy reasoning. Similarly, response to a variable speed reference can be improved as well, if an on-line mechanism for adaptation is developed. Two on-line adaptive FL speed control schemes are therefore designed. Both rely on an auxiliary FL controller in the process of on-line scaling factor adaptation. One scheme provides on-line tuning for variable speed command only, while the second scheme enables simultaneous adaptation to both the speed reference setting and the inertia variation.

The selection of the initial parameters and structure of the FL controller depend on the type of application and the speed of the target processor. If it is possible to implement more rules and membership functions for process control application, a smoother control action can be obtained. However this causes difficulty in implementation because large processing time results, and high speed processor is required. In motion control applications, a very short sampling time (i.e. high processing speed) is required in order to realise a vector controlled drive. Therefore, the selection of the suitable structure of the FL controller is very important and it is noted that this is one of the most difficult tasks in implementation of a fuzzy logic controller. Normally the minimum sampling frequency of speed controller is at least 1 kHz while the sampling frequency for current loop is typically around 10 kHz. The need for co-ordinate transformation and resolver-to-digital conversion significantly reduce the attainable speed of the processor. DSP has become a good choice for such a demanding application. However, the cost of this controller board is still very high. A solution to these problems can be use of combined digital-analogue realisation in the development of the experimental rig.

7.2 CONCLUDING REMARKS

The thesis represents a research into the FL speed control of a PWM inverter-fed SPMSM drive. The performance of the controller is evaluated based on the comparison of the simulation results to those obtained with the conventional PI speed controller under the same design and operating conditions. In general, it is not permissible to draw any conclusion based on a single transient comparison. The comparison should be carried out for the transients of the whole speed region.

Fuzzy logic can produce better performance than PI control if the controller is properly tuned. The research has proved that the constant parameter FL speed controller is more robust with respect to variation of inertia, variation of the step speed command and is characterised with better load rejection capability. It is confirmed that FL controller can be a good alternative to replace the conventional PI controller for the future developments in motion control applications. However, difficulties encountered in the design and implementation of a FL speed controller are substantial. These primarily stem from the complex structure of the FL controller and from the lack of formal simple tools for its design at present. These factors could prevent wider acceptance of the FL speed control in high performance drives by drive manufacturers in near future. The major problem is believed to be the need to tune the FL speed controller individually to each machine. While for PI speed control well established and simple methods exist for the self-commissioning of the drive, no such methods are in sight for the FL speed controller. Fuzzy logic control may therefore remain confined to some other auxiliary functions within the drive structure, where the dependence of the operating characteristics on the specific motor is less pronounced and tuning during self-commissioning is not required.

The research described in this work is related to the permanent magnet synchronous motor drive. However, as the speed control loop is studied, and as there exists a complete analogy between the vector controlled SPMSM drive and other types of high performance drives (such as separately excited DC motor, vector controlled induction machines, etc.), all the findings described here are essentially universally applicable to any other high performance electric drive.

7.3 FURTHER WORK

The most important directions for further research in the area of FL speed control of high performance drives appear to be: 1) development and possible standardisation of the most appropriate structure of the FL speed controller for the given hardware platform; 2) development of some formal method for calculation of scaling factors, that could guarantee required performance for a variety of machines and that could be incorporated into a self-commissioning procedure. Provided that these two tasks are completed satisfactorily, FL speed controller would become acceptable to the drive manufacturers and would be likely to replace PI speed control in commercially available products.

Further research needs be done to simplify the FL control algorithms and further develop fuzzy logic tools. Some fuzzy tools have capability to communicate directly to the target processor and to generate executable code from the simulation blocks. This is a very good feature because a new algorithm can be tested quickly. However, in many cases, the code generated is not optimised, and works based on 16 bit operating system. Therefore, there is not much difference, in terms of speed even when the program is running on a very high speed processor.

A further possible research direction is development of current controllers based on fuzzy logic and neuro-fuzzy approach. As an alternative, the fuzzy-neuro technique can be used to estimate the suitable scaling factors for the FL speed controller as well, instead of trial-and-error.

The scope for further development of adaptive fuzzy logic speed controllers is rather huge. The self-tuning methods, developed in this thesis, could be implemented and their performance experimentally evaluated. It should be noted that the tuning and self-tuning methods presented here did not use optimisation toolbox. A further improvement in performance is likely to be achievable by using appropriate optimisation techniques. One such method, that can be used to optimise the scaling factors for the given performance index, is the genetic algorithm.

CHAPTER 8

REFERENCES

Abut, N., Cakir, B., Inanc, N., Yildiz, A.B., and Bilgin, M.Z., (1997), "Switched reluctance motor drive by using fuzzy controller", *Proceedings of the IEEE 23th International Conference on Industrial Electronics, Control, and Instrumentation IECON,* New Orleans, Louisiana, pp. 348-353.

Ackva, A., and Reinold, H., (1992), "A simple and self-adapting high-performance current control scheme for three phase voltage source inverters", *Proceedings of IEEE Power Eelectronics Specialist Conference PESC*, Toledo, Spain, pp. 435-442.

Afonso, J.L., Fonseca, J., Martins, J.S., and Couto, C.A., (1997), "Fuzzy logic techniques applied to the control of a three-phase induction motor", *Proceedings of the IEEE International Symposium on Industrial Electronics ISIE,* Guimarâes, Portugal, pp. 1179-1184.

Altrock, C.V., (1997), "Fuzzy logic relies on linguistic rules", *Control Engineering Magazine,* pp. 12-15

Ashrafzadeh, F., Nowicki, E.P., and Salmon, J.C., (1995), "Self-organising and self-tuning fuzzy logic controller for field oriented control of induction motor drives", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS*, Orlando, Florida, pp. 1656-1662.

Assilian, S., and Mamdani, E.H., (1975), "A fuzzy logic controller for dynamic plant", *International Journal of Man- Machine Studies,* Vol. 7, pp. 1-13.

Aström, K.J., and Wittenmark, B., (1990), "*Computer controlled system: Theory and design*", Prentice Hall.

Aström, K.J., and Hâgglund, T., (1988), "*Automatic tuning of PID controllers*", Instrument Society of America.

Baghli, L., Razik, H., and Rezzoug, A., (1997a), "Comparison between fuzzy and classical speed control within a field oriented method for induction motors", 7th *European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 2.444-2.448.

Baghli, I., Razik, H., and Rezzoug, A., (1997b), "Neuro-fuzzy controller in a field oriented control for induction motors" 7th *European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp.1.096-1.101.

Baliga, B.J., (1995), "Power ICs in the saddle", *IEEE Spectrum*, July, pp. 34-49.

Barrero, F., Galvan, E., Torralba, A., and Franquelo, L.G., (1995), "Fuzzy self-tuning system for induction motor controllers", *European Conference on Power Electronics and Applications EPE,* Seville, Spain, pp. 1.283-1.285.

Basehore, P., (1993), "Fuzzy logic outperforms PID controller", *Power Conversion and Intelligent Motion Magazine PCIM,* pp. 40-46.

Bebic, M., and Jeftenic, B., (1998), "Fuzzy logic speed controller for a variable frequency AC drive system", 8th *International Power Electronics & Motion Control Conference PEMC,* Prague, Czech Republic. Vol. 5, pp. 5.184 - 5.189.

Beierke, S., Vas, P., and Stronach, A.F., (1997), "TMS320 DSP implementations of fuzzy-neural controlled drives", 7th *European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 2.449-2.453.

Bird, I.G., Zelaya, H., and Parra, De La, (1997), "DSP implementation of a fuzzy based direct flux and torque controller", 7th *European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 2.415-2.420.

Birou, I., Imecs, M., and Kelemen, A., (1996), "Implementation of fuzzy-control induction motor drive using MC 68000 processor", *Proceedings of International Power Conversion and Intelligent Motion Conference PCIM,* Nurnberg, Germany, pp. 225-235.

Blaschke, F., (1972), "The principle of field orientation as applied to the new transvector closed loop control system for rotating-field machines", *Siemens Review,* Vol. 34, pp. 227-220.

Bodin, F., Siala, S., and Sicot, L., (1997), "State space fuzzy controller for a brushless DC motor", *Proceedings of the IEEE Power Electronics Specialists Conference PESC,* St. Louis, Missouri, pp. 869-874.

Boldea, I., and Nasar, S.A., (1992), "*Vector control of ac drives*", CRC Press.

Bolognani, S., and Zigliotto, M., (1996), "Fuzzy logic control of a switched reluctance motor drive", *IEEE Transactions on Industry Applications,* Vol. 32, No. 5, pp. 1063-1068.

Bolognani, S., Tursini, M., Zhang, D.Q., and Zigliotto, M., (1994), "Application of a fuzzy self organising controller to a motor drive", *Proceedings Symposium on Power Electronics, Electrical Drives, Advanced Electrical Motor SPEEDAM,* Taormina, Italy, pp. 299-304.

Bose, B.K., (1989), "Power electronics - An emerging technology", *IEEE Transactions on Industrial Electronics,* Vol. 36, No. 3, pp. 403-412.

Bose, B.K., (1990), "An adaptive hysteresis-band current control technique of a voltage-fed PWM inverter for machine drive system", *IEEE Transactions on Industrial Electronics,* Vol. 37, No.5, pp. 402-408.

Bose, B.K., (1994), "Expert system, fuzzy logic, and neural network applications in power electronics and motion control", *Proceedings of the IEEE,* Vol. 82, No. 8, pp. 1303-1323.

Bose, B.K., (1997a), "Intelligent control and estimation in power electronics and drives", *Proceedings of the IEEE International Conference on Electric Machines and Drives,* Milwaukee, Wisconsin, pp. TA2-2.1-TA2-2.6.

Bose, B.K., (1997b), "High performance control and estimation in ac drives", *Proceedings of the IEEE 23th International Conference on Industrial Electronics, Control, and Instrumentation IECON,* New Orleans, Louisiana, pp. 377-385.

Boussak, M., and Bauer, M., (1996), "Robust speed and position control of an indirect field oriented controlled induction motor drive using fuzzy logic regulator", *Proceedings of the International Conference on Electric Machines ICEM*, Vigo, Spain, pp. 219-224.

Brickwedde, A., (1985), "Microprocessor-based adaptive speed and position control for electrical drives", *IEEE Transactions on Industry Applications,* Vol. 1A-21, No. 5, pp. 1154-1161

Brod, D.M., and Novotny, D.W., (1985), "Current control of VSI-PWM inverters", *IEEE Transactions on Industry Applications,* Vol. IA-21, pp. 562-570.

Cerruto, E., Consoli, A., Raciti, A., and Testa, A., (1995), "A robust adaptive controller for PM motor drives in robotic applications", *IEEE Transactions on Power Electronics,* Vol. 10, No. 1, pp. 62-71.

Chang, K-T., Low, T-S, and Lee, T-H., (1994), "An optimal speed controller for permanent-magnet synchronous motor drives", *IEEE Transactions on Industrial Electronics,* Vol. 41, No. 5, pp. 503-510.

Chern, T.L., and Wu, Y.C., (1993), "Design of brushless DC position servo systems using integral variable structure approach", *IEE Proceedings- Part B,* Vol. 140, pp. 160-170.

Cheung, J.Y.M, Cheng, K.W.E., and Kamal, A.S., (1996), "Motor speed control by using a fuzzy logic model reference adaptive controller", *Proceedings of IEE Power Electronics and Variables Speed Drives PEVD,* IEE Conference Publication No. 429, Nottingham, United Kingdom, pp. 430-435.

Chiricozzi, E., Parasiliti, F., Tursini, M., and Zhang, D.Q., (1996), "Fuzzy self tuning PI control of PM synchronous motor drive", *International Journal of Electronics,* Vol. 80, No. 2, pp. 211-221.

Costa, A., Gloria, A.D., Faraboschi, P., Pagni, A., and Rizotto, G., (1995), "Hardware solution for fuzzy control", *Proceedings of the IEEE,* Vol. 83, pp. 422-432.

da Silva, W. G., and Acarnley, P.P., (1997), "Fuzzy logic controlled DC motor drive in the presence of load disturbance", *7th European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 2.386-2.391

Daley, S., and Gill, K.F., (1985), "A justification for the wider use of fuzzy logic control algorithms" *Proceedings of Institute of Mechanical Engineers*, Vol. 199, No. C1, pp. 43-49.

Daley, S., and Gill, K.F., (1987), "Atitude control of a spacecraft using an extended self-organising fuzzy logic controller", *Proceedings of the Institute of Mechanical Engineers,* Vol. 201, No. C2, pp. 97-106.

Dash, P.K., Panda, S.K., and Xu, J.X., (1997), "Fuzzy and neural controllers for dynamic systems: an overview", *Proceedings of the IEEE International Conference on Power Electronics and Drive System,* Singapore, pp. 810-816.

Davies, W.D.T., (1970), "*System identification for self-adaptive control*", John Wiley & Sons Ltd.

Depenbrock, M., (1988), "Direct self-control (DSC) of inverter-fed induction machine", *IEEE Transactions on Power Electronics,* Vol. 3, No. 4, pp. 420-429.

Donescu, V., Neacsu, D.O., Griva, G., and Profumo, F., (1996), "A systematic design method for fuzzy logic speed controller for brushless DC motor drives", *Proceedings of the IEEE Power Electronics Specialist Conference PESC*, Baveno, Italy, pp. 689-694.

Dote, Y., (1995), "Introduction to fuzzy logic", *Proceedings of the IEEE International Conference on Industrial Electronics, Control, and Instrumentation IECON*, Orlando, Florida, pp. 50-56.

Driels, M., (1995), "*Linear control system engineering*", McGraw Hill International.

Ebach, U. and Graser, A., (1995), "Design of an extended Smith controller with gain adaptation", *Control Engineering Practice*, Vol. 3, No. 10, pp. 1467-1470.

Eminoglu, I., and Atlas, I.H., (1996), "A method to form fuzzy control rules for a PMDC motor drive system", *Electric Power Systems Research*, Vol. 39, pp. 81-87.

Elkan, C., (1994), "The pradoxical success of fuzzy logic", *IEEE Expert - Intelligent systems & their applications,* August, pp. 3-8.

Enjeti, P.N., Ziogas, P.D., and Lindsay, J.F., (1992), "A new current control scheme for ac motor drive", *IEEE Transactions on Industry Applications,* Vol. 28, No. 4, pp. 842-848.

Ertugrul, N., and Acarnley, P., (1994), "A new algorithm for sensorless operation of PM motors", *IEEE Transactions on Industry Applications,* Vol. 30, No. 1, pp. 126-133.

Feng, X., and Chen, B., (1996), "Fuzzy-controlled DC drive system with load observer", *Proceedings of the 4th International Workshop on Advanced Motion Control AMC,* Mie University, Tsu-City, Japan, pp. 354-358.

Ficarra, M. C., Eguliaz, J.M.M., and Peracaula, J., (1996), "Fuzzy control of an induction motor with compensation of system dead-time", *Proceedings of the IEEE Power Electronics Specialist Conference, PESC*, Baveno, Italy, pp. 677-681.

Fischle, F., and Schroder, D., (1997), "Stable model reference neuro control for electrical drives systems", *7th Eropean Conference on Power Electronics and Applications EPE*, Trondheim, Norway, pp. 2.432-2.437.

Fodor, D., Katona, Z., and Vass, J., (1996), "On fuzzy logic speed control for vector controlled ac motors" *Proceedings 4th International Workshop on Advanced Motion Control AMC,* Mie University, Tsu-City, Japan, pp. 186-191.

Fodor, D., Vas, J., and Katona, J., (1997), "Embedded controller board for field-oriented AC drives", *Proceedings of the IEEE 23rd International Conference on Industrial Electronics, Control, and Instrumentation IECON*, New Orleans, Louisiana, pp. 1022-1027.

Gabriel, R., Leonhard, W., and Nordby, C., (1980), "Field-oriented control of standard ac motor using microprocessors", *IEEE Transactions on Industry Applications,* Vol. IA-16, No. 2, pp. 186-192.

Galichet, S., and Foulloy, L., (1995), "Fuzzy controller: Synthesis and Equivalences", *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 2, pp.140-147.

Gawthrop, P.J., (1987), "*Continuous-time self-tuning control*", Research Studies.

Ghribi, M., and Le-Huy, H., (1994), "Optimal control and variable structure combination using a permanent-magnet synchronous motor", *Proceedings of the IEEE Power Electronics Specialist Conference PESC*, Toledo, Spain, pp. 408-415.

Ghwanmeh, S.H., (1996), "The investigation of on-line self learning fuzzy logic control for non-linear process", *Ph.D Thesis,* Liverpool John Moores University, Liverpool, United Kingdom.

Ghwanmeh, S.H., Jones, K.O., and Williams, D., (1995), "Comparison of different fuzzy implication approaches under self-learning fuzzy logic control", *LAAS International Conference on Computer Simulation*, Beirut, Lebanon, pp. 221-228.

Ghwanmeh, S.H., Jones, K.O., and Williams, D., (1996), "Self-learning fuzzy logic control of non-linear processes", *Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Fukuoka, Japan, pp. 213-222.

Goureau, P., and Ibaliden, A., (1996), "Robustness of a fuzzy controller for induction machine", *Proceeding of 5th International Conference on Modelling and Simulation of Electric Machines, Converters & System, ELECTRIMACS,* St. Nazaire, France, pp. 551-556.

Groover, M.P., Weiss, M., Nagel, R.N., and Odrey, N.G., (1986), *"Industrial robotics: Technology, Programming, and Applications*", McGraw-Hill.

Guo, S., and Peters, L., (1996), " Design and application of an analog fuzzy logic controller", *IEEE Transactions on Fuzzy Systems,* Vol. 4, No. 4, pp. 429-436.

Hagl, R., Beilski, S., and Heidenhain, J., (1996), *"Rotary encoders make digital drives dynamic*", Hohner Automation Limited.

Hanus, R., Kinnaert, M., and Henrotte J.L., (1987), "Conditioning technique, A general anti-windup and bumpless transfer method", *Automatica*, Vol. 23, No. 6, pp. 729-739.

Harris, C.J., (1981), "*Self-tuning and adaptive control: Theory and applications*", Peter Peregrinus.

Hashimoto, H., Yamamoto, H., Yanagisawa, S., and Harasima, F., (1988), "Brushless servo motor control using variable structure approach", *IEEE Transaction on Industrial Electronics,* Vol. 34, pp. 160-170.

He, G., Jiang, J.P., Shen, J., and Lei, J., (1997), "A fuzzy variable structure control scheme for AC servo drive system", *Proceedings of the IEEE International Conference on Electric Machines and Drives,* Milwaukee, Wisconsin, pp. MC3-3.1-MC3-3.3.

Heber, B., Xu L., and Tang, Y., (1997), "Fuzzy logic enhanced speed control of indirect field-oriented induction machine drive", *IEEE Transactions on Power Electronics,* Vol. 12, No. 5, pp. 772-778.

Henneberger, G., (1993), "Servo drives: A status review", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Nurnberg, Germany, pp. 1-14.

Higgins, C.M., and Goodman, R.M., (1994), "Fuzzy rule-based networks for control", *IEEE Transactions on Fuzzy Systems,* Vol. 2, No. 1, pp. 82-88.

Hikita, H., (1988), "Servomechanisms based on sliding mode control", *International Journal of Control,* Vol. 48, No. 2, pp. 435-447.

Hissel, D., Maussion, P., Gateau, G., and Faucher, J., (1997), "Fuzzy logic control optimization of electrical systems using experimental designs", *7th European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 1.090-1.095.

Ho, E., and Sen, P.C., (1995), "High-performance decoupling control techniques for various rotating field machines", *IEEE Transactions on Industrial Electronics,* Vol. 42, No. 1, pp. 40-49.

Holtz, J., (1992), "Pulsewidth modulation - A survey", *IEEE Transactions on Industrial Electronics,* Vol. 39, No. 5, pp. 410-419.

Huy, H.L., (1989), "An adaptive current control scheme for PWM synchronous motor drives: Analysis and simulation", *IEEE Transactions on Power Electronics,* Vol. 4, No. 4, pp. 486-495.

Ibrahim, Z., and Levi, E., (1998), "A Detailed Comparative Analysis of Fuzzy Logic and PI Speed Control in High Performance Drives", *Proceedings of IEE International Power Electronics and Variable Speed Drives Conference PEVD*, London, United Kingdom, IEE Conference Publication No. 456, pp. 638-643.

Ibrahim, Z., Levi, E., and Williams, D., (1998a), "Fuzzy logic versus PI speed control of servo drives: a comparison", *8th International Power Electronics & Motion Control Conference PEMC*, Prague, Czech Republic, pp. 4-34 - 4-39.

Ibrahim, Z., Levi, E., and Williams, D., (1998b), "A Self-Tuning Method For Fuzzy Logic Speed Controller In High Performance Drives", *33rd Universities Power Engineering Conference UPEC,* Napier University, Edinburgh, United Kingdom, pp. 819-822.

Ide, K., Bai, Z-G., Yang, Z-J., and Tsuji, T., (1995), "Vector approximation method with parameter adaptation and torque control of CSI-fed Induction motor", *IEEE Transactions on Industry Applications,* Vol. 31, No. 4, pp. 830-840.

Inoue, K., Izumi, T., and Nakaoka, M., (1996), "An advanced DC brushless servo drive system with a fuzzy logic-based self-tuning control scheme and its practical evaluation", *International Journal of Electronics*, Vol. 80, No. 2, pp. 223-233.

Inoue, K., Takakado, Y., and Nakaoka, M., (1993), "New auto-tuning implementation for fuzzy-reasoning based DC brushless servo systems", *Proceedings of the International Conference on Power Conversion & Intelligent Motion PCIM*, Nurnberg, Germany, pp. 98-103.

Jager, R., (1995), "*Fuzzy logic in control*", Ph.D Thesis, Technische Universiteit Delft, Amsterdam, Holland.

Jahns, T.M, (1994), "Motion control with permanent-magnet ac machines", *Proceedings of the IEEE*, Vol. 82, No. 8, pp. 1241-1252.

Jansen, P.L., Lorenz, R., and Novotny, D.W., (1994), "Observer-based direct field orientation: Analysis and comparison of alternative methods", *IEEE Transactions on Industry Applications,* Vol. 30, No. 4, pp. 945-953.

John, A.J., and Herschel, B. S., (1995), "Advantages of an alternative form of fuzzy logic", *IEEE Transactions on Fuzzy Systems,* Vol. 3, No. 2, pp. 149-156.

Kazmierkowski, M.P., Sobczuk, D.L., and Dzieniakowski, M.A., (1995), "Neural network current control of VS-PWM inverter", *European Conference on Power Electronics and Applications EPE,* Seville, Spain, pp. 1.415-1.420.

Keuer, D., (1997), "Fuzzy logic smooths PID-resistant loop", *Control Engineering Magazine,* pp. 20-24.

Kickert, W.J., and Lemke, V.N., (1976), "Application of a fuzzy controller in warm water plant", *Automatica,* Vol. 12, No. 4, pp. 301-308.

Kickert, W.J., and Mamdani, E.H., (1978), "Analysis of a fuzzy logic controller", *Fuzzy Sets and Systems,* Vol. 1, No. 1, pp. 29-44.

Kim, D-H., Kim, H-S., Kim, J-J., Won, C-Y., and Kim, S-C., (1996), "Induction motor servo system using variable structure control with fuzzy sliding surface", *Proceedings of the IEEE 22nd International Conference on Industrial Electronics, Control, and Instrumentation IECON,* Taipei, Taiwan, pp. 977-982.

Kim, J.S., and Sul, S.K., (1995), "New approach for the low speed operation of the PMSM drives without rotational position sensors", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS,* Orlando, Florida, pp. 766-771.

King, R.E., and Mamdani E.H., (1977), "Application of fuzzy control system to industrial process", *Automatica,* Vol. 13, No. 3, pp. 235-242.

Klafter, R.D., Chmielewski, T.A., and Negin, M., (1989*), "Robotic engineering: An integrated approach",* Prentice Hall International.

Kothare, M.V., Campo, P.J, Morari M., and Nett C.N., (1994), "A unified framework for study of anti-windup designs", *Automatica,* Vol. 30, No. 2, pp.1869-1883.

Lee, C.C., (1990a), "Fuzzy logic in control system: Fuzzy logic controller-part 1", *IEEE Transactions on Systems, Man, Cybernetics*, Vol. 20, No. 2, pp. 404-417.

Lee, C.C., (1990b), "Fuzzy logic in control system: fuzzy logic controller-part 2", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp. 419-434.

Lee, J., (1993), "On methods for improving performance of PI-type fuzzy logic controllers", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 4, pp. 298-301.

Le-Huy, H., (1994), "Microprocessors and digital IC's for motion control", *Proceedings of the IEEE*, Vol. 82, No. 8, pp. 1140-1160.

Le-Huy, H., (1995), "An adaptive fuzzy controller for permanent-magnet AC servo drives", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS*, Orlando, Florida, pp. 104-110.

Le-Huy, H., Viarouge, P., and Kamwa, I., (1995), "Model reference adaptive fuzzy control of a permanent magnet synchronous motor", *Proceedings of the IEEE Industrial Conference on Electronics, Control, and Instrumentation IECON*, Orlando, Florida, pp. 1440-1445.

Leonhard, W., (1986), "*Control of Electrical Drives*", Springer-Verlag, Berlin, Germany.

Li, H. X., and Gatland, H.B., (1995), "A new methodology for designing a fuzzy logic controller", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 3, pp. 505-512.

Li, H., (1997), "A comparative design and tuning for conventional fuzzy control", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 27, No. 5, pp. 884-889.

Liaw, C.M., and Cheng, S.Y., (1995), "Fuzzy two-degrees of freedom speed controller for motor drives", *IEEE Transactions on Industrial Electronics*, Vol. 42, No. 2, pp. 209-216.

Lin, F.J., and Chiu, S.L., (1998), "Adaptive fuzzy sliding-mode control for PM synchronous servo motor drives", *IEE Proceedings on Control Theory Applications*, Vol. 145, No. 1, pp. 63-72.

Linkens, D.A., and Abbod, M.F., (1991), "Self organising fuzzy logic controller and the selection of its scaling factors", *Transactions of the Institute of Measurement and Control*, Vol. 14, No. 3, pp. 114-125.

Lipo, T.A., and Plunkett, A.B., (1974), "A novel approach to induction motor transfer functions", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-93, No. 5, pp. 1410-1418.

Liu, T-H., and Liu, C-H., (1990), "A multiprocessor-based fully digital control architecture for permanent magnet synchronous motor drive", *IEEE Transactions on Power Electronics,* Vol. 5, No. 4, pp. 413-423.

Lorenz, R.D., Lipo, T.A., and Novotny, D.W., (1994), "Motion control with induction motors", *Proceedings of the IEEE,* Vol. 82, No. 8, pp. 1215-1239.

Lu, Y.S., and Chen, J.S., (1994), "A self-organising fuzzy sliding-mode controller design for a class of non-linear servo system", *IEEE Transactions on Industrial Electronics,* Vol. 41, No. 5, pp. 492-502.

Malesani, L., and Tomasin, P., (1993), "PWM current control techniques of voltage source converter - a survey", *Proceedings of the IEEE International Conference on Industrial Electronics, Control and Instrumentation IECON*, Miaui, Hawaii, pp. 670-675.

Mamdani, E.H., (1974), "Application of fuzzy algorithm for control of simple dynamic process", *Proceedings of the IEE*, Vol. 121, No. 12, pp. 1585-1588.

Mamdani, E.H., and Assilian, S., (1975), "An experiment in linguistic synthesis with a fuzzy logic control", *International Joumal of Man Machine Studies*, Vol. 7, No. 1, pp. 1-13.

Mamdani, E.H., (1994), "Fuzzy control - a misconception of theory and application", *IEEE Expert - Intelligent systems & their applications,* August, pp. 27-28.

Matsuo, T., and Lipo, T.A., (1995), "Rotor position detection scheme for synchronous reluctance motor based on current measurements", *IEEE Transactions on Industry Applications*, Vol. 31, No. 4, pp. 860-868.

Meshkat, S., (1996), "Advanced DSP motion control system: Part I: Self-tuning techniques", *Power Conversion and Intelligent Motion Magazine*, pp. 47-56.

Miller, T.J.E., (1993), "*Switched reluctance motors and their controľ*, Magna Physics Publishing/Clarendon Press.

Monti, A., and Scaglia, A., (1997), "A closed-form approach to the design of fuzzy-PI controller", *7th European Conference on Power Electronics and Applications EPE,* Trondheim, Norway, pp. 2.471-2.476.

Namuduri, C., and Sen, P.C., (1987), "A servo-control system using a self-controlled synchronous motor with sliding mode control", *IEEE Transactions on Industry Applications,* Vol. IA-23, No. 2, pp. 283-295.

Nasar, S.A., Boldea, I., and Unnewehr, L.E., (1993), "*Permanent magnet, reluctance, and self-synchronous motors*", CRC Press.

Nonaka, S., and Kawaguchi, T., (1991), "Variable-speed control of brushless half-speed synchronous motor by voltage source inverter", *IEEE Transactions on Industry Applications,* Vol. 27, No. 3, pp. 545-551.

Novotny, D.W., and Lipo, T.A, (1996), "*Vector control and dynamics of ac drives*", Clarendon Press, Oxford, United Kingdom.

Ogasawara, S., and Akagi, H., (1991), "An approach to position sensorless drive for brushless dc motors", *IEEE Transactions on Industry Applications*, Vol. 27, No. 5, pp. 928-933.

Ohtani, T., Takada, N., and Tanaka, K., (1992), "Vector control of induction motor without shaft encoder", *IEEE Transactions on Industry Applications,* Vol. 28, No. 1, pp. 157-164.

Palm, R., (1995), "Scaling of fuzzy controllers using the cross-correlation", *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 1, pp. 116-122.

Panda, S.K., Zhu, X.M., and Dash, P.K., (1997), "Fuzzy gain scheduled PI speed controller for switched reluctance motor", *Proceedings of the IEEE International Conference on Industrial Electronics, Control, and Instrumentation IECON*, New Orleans, Louisiana, pp. 989-994.

Parasiliti, F., Tursini, M., and Zhang, D.Q., (1996), "Adaptive fuzzy logic control for high performance PM synchronous drives", *Proceedings* *of the IEEE Mediterranean Electrotechnical Conference MELECON*, Bari, Italy, pp. 323-327.

Parasiliti, F., Tursini, M., and Zhang, D.Q., (1997), "An analytical approach to the derivation of fuzzy PI scaling factors", *Proceedings of the IEEE International Conference on Power Electronics and Drive Systems*, Singapore, pp. 280-285.

Passino, K.M., and Yurkovich, S., (1998), "Fuzzy Control", Addison Wesley Logman , Inc.

Paszkiewic, D., and Lin, P., (1987), " Fuzzy self-adaptive control of pH using decoupling theorem", *Proceeding of the International Conference on Applied Informatics IASTED*, Geneva, Switzerland, pp. 209-212.

Pedrycz, W., (1993), *"Fuzzy control and fuzzy systems"*, John Wiley and Sons.

Peng, Y., Vrancic, D., and Hanus, R., (1996), "Anti-windup, bumperless, and conditional transfer techniques for PID controllers", *IEEE Control Systems*, pp. 48-56.

Pillay, P., and Krishnan, R., (1988), "Modeling of permanent magnet motor drives", *IEEE Transactions on Industry Electronics*, Vol. 35, No. 4, pp. 986-996.

Pilay, P., editor, (1989), "Performance and design of permanent magnet ac motor drives", *Tutorial course IEEE Industry Application Society Annual Meeting IAS*, San Diego, Califonia.

Pillay, P., and Krishnan, R., (1989), "Modeling, simulation and analysis of permanent-magnet motor drives, Part I: The permanent magnet synchronous motor drive", *IEEE Transactions on Industry Applications*, Vol. 25, No. 2, pp. 265-273.

Pillay, P., and Krishnan, R., (1990), "Control characteristics and speed controller design for a high performance permanent magnet synchronous motor drive", *IEEE Transactions on Power Electronics*, Vol. 5, No. 2, pp. 151-159.

Pool, K., (1995), "High resolution optical encoders", Computer Optical Products, Inc.

Qin, S.J., and Borders, G., (1994), "A multiregion fuzzy logic controller for nonlinear process control", *IEEE Transactions on Fuzzy Systems,* Vol. 2, No.1, pp. 74-80.

Rajashekara, K., Kawamura, A., and Matsuse, K., (1996), "*Sensorless Control of AC Motor Drives*", IEEE Press, Piscataway, New Jersey.

Raviraj, V.S.C., and Sen, P.C., (1995), "Comparative study of proportional-integral, sliding mode and fuzzy logic controllers for power converters", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS,* Orlando, Florida, pp. 1651-1655.

Sanz, M.A., Cabanas, M.F., Melero, M.G., and Aleixandre, J.G., (1996), "Vector control of surface mounted permanent magnet synchronous motors with speed and position estimation", *International Conference on Electrical Machines ICEM,* Vigo, Spain, pp. 213-218.

Sathiakumar, S., Hui, S.Y.R., and Gosden, D.F., (1994), "Speed control of PM ac motor without position sensor", *International Conference on Power Electronics and Motion Control PEMC,* Warsaw, Poland, pp. 378-382.

Schauder, C.D., and Caddy, R., (1982), "Current control of voltage source inverters for four quadrant drive performance", *IEEE Transactions on Industry Applications,* Vol. IA-18, pp. 163-171.

Schofield, J.R.G., (1995), "The next revolution in drive technology direct torque control DTC", *Proceeding of Drives and Controls Conference,* Telford, United Kingdom, pp. 34-38.

Schroedl, M., (1994), "Sensorless control of permanent magnet synchronous motors", *Electric Machines and Power Systems,* Vol. 22, pp.173-185.

Sebastian, T., Slemon, G.R., and Rahman, M.A., (1986), " Modelling of permanent magnet synchronous motor", *IEEE Transactions on Magnetics,* Vol. MAG-22, No. 5, pp.1068-1071.

Self, K., (1990), "Designing with fuzzy logic", *IEEE Spectrum,* pp. 42-45.

Sen, P.C., (1990), "Electric motor drives and control - Past, present, and future", *IEEE Transactions on Industrial Electronics,* Vol. 37, No. 6, pp. 562-575.

Senjyu, T.T., Ashimine, S., and Uezato, K., (1996), "Robust position control of dc servomotors using fuzzy reasoning", *Proceedings 4th International Workshop on Advanced Motion Control AMC,* Mie University, Tsu-City, Japan, pp.115-120.

Sepe, R.B., (1991), "Real-time adaptive control of the permanent-magnet synchronous motor", *IEEE Transactions on Industry Applications,* Vol. 27, No. 4, pp. 706-714.

Shenoi, S., Ashenayi, K., Timmerman, M., (1995), "Implementation of a learning fuzzy controller", *IEEE Control Systems,* Vol. 15, No. 3, pp. 73-80.

Silva, J.F., Pina, F., and Quadrado, J.C., (1996), "DSP based fuzzy controller for electromechanical drives", *Proceeding of 5th International Conference on Modelling and Simulation of Electric Machines, Converters & System ELECTRIMACS,* St. Nazaire, France, pp. 409-414.

Silva, N.J.L., and Le-Huy, H., (1997), "A fuzzy controller with a fuzzy adaptive mechanism for the speed control of a PMSM", *Proceedings of the IEEE 23rd International Conference on Industrial Electronics, Control, and Instrumentation IECON,* New Orleans, Louisiana, pp. 995-1000.

Slemon, G.R., (1994), "Electrical machines for variable-frequency drives", *Proceedings of the IEEE,* Vol. 82, No. 8, pp. 1123-1138.

Smith, M.L., (1994), "Sensors, appliance control, and fuzzy logic", *IEEE Transactions on Industry Applications,* Vol. 30, No. 2, pp. 305-309.

Sokola, M., Knezevic, M., and Levi, E., (1992), "A comparative study of current control techniques for permanent magnet synchronous motor drives", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Numberg, Germany, pp. 58-69.

Sonaglioni, L., (1995), "Predictive digital hysteresis current control", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS,* Orlando, Florida, pp.1879-1886.

Sousa, G.D.C., and Bose, B.K., (1994), "A fuzzy set theory based control of a phase-controlled converter DC machine drive", *IEEE Transactions on Industry Applications*, Vol. 30, No. 1, pp. 34-44.

Sousa, G.C.D., and Bose, B.K., (1995), "Fuzzy logic applications to power electronics and drives - An overview", *Proceedings of the IEEE International Conference on Industrial Electronic, Control and Instrumentation IECON*, Orlando, Florida, pp. 57-62.

Stronach, A.F., and Vas, P., (1995), "Variable-speed drives incorporating interacting multiloop adaptive controllers", *IEE Proceedings on Control Theory Applications,* Vol. 142, No. 5, pp. 411-419.

Stronach, A.F., and Vas, P., (1998), "Design, DSP implementation, and performance of artificial-based speed estimators for electromechanical drives", *IEE Proceedings on Control Theory Applications,* Vol. 145, No. 2, pp. 197-203.

Stronach, A.F., Vas, P., and Lees, P., (1994), "The application of self-tuning digital controllers in variable-speed drives", *Proceedings of IEE Power Electronics and Variable Speed Drives Conference' PEVD,* London, United Kingdom, IEE Conference Publication No. 399, pp. 119-124.

Stronach, A.F., Vas, P., and Neuroth, M., (1997), "Implementation of intelligent self-organising controllers in DSP controlled electromechanical drives", *IEE Proceedings on Control Theory Applications,* Vol. 144, No. 4, pp. 324-330.

Stronach, A.F., Vas, P., and Neuroth, M., (1998), "The design and DSP implementation of self-configuring controllers for high performance electromechanical drives", *Proceeding of IEE Power Electronics and Variable Speed Drives Conference PEVD,* London, United Kingdom, IEE Conference Publication No. 456, pp. 644-649.

Stumberger, B., Kreca, B., and Hribernik, B., (1996), "Variation of parameters of two axis model of synchronous motor with permanent magnets", *International Conference on Electrical Machines ICEM,* Vigo, Spain, Vol.1, pp.12-16.

Ta-Cao, M., and Huy, H. L., (1996), "Model reference adaptive fuzzy controller and fuzzy estimator for high performance induction motor drives", *Proceedings of the IEEE 31th Industry Applications Society Annual Meeting IAS,* San Diego, Califonia, pp. 380-387.

Tagaki, T., and Sugeno, M., (1983), "Derivation of fuzzy control rules from human operator's control actions", *Proceeding of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis,* Marseilles, France, pp. 50-60.

Takahashi, I., and Noguchi, T., (1986), "A new quick response and high efficiency strategy of an induction motor" *IEEE Transactions on Industry Applications,* Vol. IA-22, No. 5, pp. 820-827.

Tana, Y., and Xu, L., (1994), "Fuzzy logic application for intelligent control of variable speed drive", *IEEE Transactions of Energy Conversion,* Vol. 9, No. 4, pp. 679-685.

Tang, K.L., and Mulholland, R. J., (1987), "Comparing fuzzy logic with classical controller designs", *IEEE Transactions on Systems, Man and Cybemetics,* Vol. SMC-17, No. 6, pp. 1085-1087.

Thomas, D.E., and Armstrong-Hèlouvry, B., (1995), "Fuzzy logic control - A taxonomy of demonstrated benefits", *Proceedings of the IEEE,* Vol. 83, No. 3, pp. 407-420.

Ting, C.S., Li, T.H., and Kung, F.F., (1996), "An approach to systematic design of fuzzy control system", *Fuzzy Sets and Systems,* Vol. *77,* pp. 151-166.

Tong, R.M., Beck, M.B., and Latten, A., (1980), " Fuzzy control of the activated sludge waste water treatment process", *Automatica,* Vol. 16, No. 6, pp. 695-701.

Tripathi, A., and Sen, P.C., (1992), "Comparative analysis of fixed and sinusoidal band hysteresis current controllers for voltage source inverters", *IEEE Transactions on Industrial Electronics,* Vol. 39, No. 1, pp. 63-72.

Trzynadlowski, A.M., (1994), "*The field orientation principle in control of induction motors*", Kluwer Academic Publishers, Norwell, Massachusetts.

Trzynadlowski, A.M., (1996), "An overview of modern PWM techniques for three-phase, voltage-controlled, voltage-source inverters", *IEEE International Symposium on Industrial Electronics ISIE*, Warsaw, Poland, pp. 25-37.

Vas, P., (1990), "*Vector control of AC machines*", Oxford University Press, Oxford, United Kingdom.

Vas, P., (1992), "*Electrical machines and drives*: A space vector theory approach", Oxford University Press, Oxford, United Kingdom.

Vas, P., (1998), "*Sensorless vector and direct torque control*", Oxford University Press, Oxford, United Kingdom.

Vas, P., (1999), "*Artificial-intelligence-based electrical machines and drives: application of fuzzy, neural, fuzzy-neural, and genetic algorithms based techniques*", Oxford University Press, Oxford, United Kingdom.

Vas, P., and Drury, W, (1994), "Vector-control drives", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Nurnberg, Germany, pp. 213-228.

Vas, P., and Drury, W., (1995), "Future trends and developments in electrical machines and variable-speed drives", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Numberg, Germany, pp. 1-27.

Vas, P., and Stronach, A.F., (1996a), "Minimum configuration soft-computing based DSP controlled drives", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Numberg, Germany, pp. 299- 313.

Vas, P., and Stronach, A.F., (1996b), "Adaptive fuzzy-neural DSP-control of high-performance drives", *Proceedings of IEE Power Electronics and Variable Speed Drives Conference PEVD,* London, United Kingdom, IEE Conference Publication No. 429, pp. 424-429.

Vas, P., Chen, J., and Stronach, A.F, (1994a), "Fuzzy control of dc drives", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM*, Nurnberg, Germany, pp.11-31.

Vas, P., Chen, J., and Stronach, A.F., (1994b), "Fuzzy control of AC drives", *Proceedings of IEE Power Electronics and Variable Speed Drives Conference PEVD*, London, United Kingdom, IEE Conference Publication No. 399, pp. 113-118.

Vas, P., Stronach, A.F., and Neuroth, M., (1996a), "Design and DSP implementation of fuzzy controllers for servo drives", *Electrical Engineering,* Vol. 79, pp. 265-276.

Vas, P., Drury, W., and Stronach, A.F., (1996b), "Recent developments in artificial intelligence based drives - a review", *Proceedings of International Power Conversion & Intelligent Motion Conference PCIM,* Nurnberg, Germany, pp. 59-70.

Vas, P., Stronach, A.F., and Neuroth, M., (1997), "Full fuzzy control of a DSP-based high performance induction motor drive", *IEE Proceedings - Control Theory and Applications*, Vol. 144, No. 5, pp. 361-368.

Vas, P., Stronach, A.F., and Neuroth, M., (1998), "DSP-based speed-sensorless vector controlled induction motor drives using AI-based speed estimator and two current sensors", *Proceedings of IEE Power Electronics and Variable Speed Drives Conference PEVD*, London, United Kingdom, IEE Conference Publication No. 456, pp. 442-649.

Vivek, G.M., and Kwong, W.N., (1995), "Fuzzy learning control for a flexible-link robot", *IEEE Transactions on Fuzzy System,* Vol. 3, No. 2, pp. 199-210.

Walgama, K.S., Ronnback, S., and Sternby, J., (1992), "Generalisation of conditioning technique for anti-windup compensators", *IEE Proceedings, Part D,* Vol. 139, pp.109-118.

Wei, D., (1993), "Variable structure system for the field-oriented control of a permanent magnet synchronous motor drives", *Ph.D Thesis,* Liverpool University, Liverpool, United Kingdom.

Wellstead, P.E., and Zarrop, M.B., (1991), "*Self-tuning systems control and signal processing*", John Wiley and Sons.

Wijenayake, A.H., Bailey, J.M., and Naidu, M., (1995), "A DSP-based position sensor elimination method with an on-line parameter identification scheme for permanent magnet synchronous motor drives", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS*, Orlando, Florida, pp. 207-215.

Wu, J.C., and Liu, T.S., (1996), "A sliding-mode approach to fuzzy control design", *IEEE Transactions on Control System Technology,* Vol. 4, pp. 141-151.

Yager, R. R., and Filev, D., (1993), "On the issue of defuzzification and selection based on a fuzzy set", *Fuzzy Sets and Systems*, Vol. 55, pp. 255-271.

Ying, H., (1993), "A nonlinear fuzy controller with linear control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear proportional-integral controller", *Automatica*, Vol. 29, No. 2, pp. 499-505.

Ying, H., Siler, W., and Buckley, J.J., (1990), "Fuzzy control theory: A nonlinear case", *Automatica,* Vol. 26, No. 3, pp. 513-520.

Yoon, B-D., Kim, Y-H., Rhew, H-W., and Kim, C-K., (1996), "An approach for the design of fuzzy sliding adaptive controller in induction motor drives", *Proceedings of the IEEE International Conference on Industrial Electronics, Control, and Instrumentation IECON,* Taipei, Taiwan, pp. 971-976.

Young, K., (1995), "The introduction of fuzzy logic to drive", *Proceedings of Drives and Controls Conference,* Telford, United Kingdom, pp. 46-51.

Zadeh, L.A., (1965). "Fuzzy sets" , *Information and control*, Vol. 8 , pp. 338-353.

Zhao, Z., Tomizuka, M., and Isaka, S., (1993), "Fuzzy gain scheduling of PID controllers", *IEEE Transactions on System, Man, and Cybernetics*, Vol. 23, No. 5, pp. 1392-1398.

Zhen, L., and Xu, L., (1996), "A comparison study of three fuzzy schemes for indirect vector control of induction machine drives", *Proceedings of the IEEE Industry Applications Society Annual Meeting IAS*, San Diego, California, pp. 1725-1732.

Zheng, L., (1992), "A practical guide to tuning of PI like fuzzy controllers", *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, California, pp. 633-640.

Ziegler, J.G., and Nichols, N.B., (1942), "Optimum setting for automatic controllers", *Transaction on America Society of Mechanical Engineers ASME*, Vol. 64, pp. 759-768.

Appendix A

Data of the motor used in simulations

A1. Motor used in the research

	Motor data
Motor type	-
Peak torque, $T_{e\,max}$ (Nm)	*20.7*
Rated torque, $T_{e\,n}$ (Nm)	*6.1*
Rated current, I_n(A)	*6.2*
Peak current, I_{max}(A)	*21*
Rated speed, ω_n(rad / s) n_n(rpm)	*180* *(1720 rpm)*
Inertia J (kgm^2)	*0.00176*
Winding resistance, $R(\Omega)$ (phase to neutral)	*1.4*
Winding inductance, L (mH) (phase to neutral)	*5.6*
Magnet flux, ψ_m(Vs / rad)	*0.1546*
Mech. time constant, τ_m(ms)	-
Elec. time constant, τ_e(ms)	-
Rated frequency, f_n(Hz)	*86*
Pole pair number	*3*
DC link voltage, V_{dc}(V)	*220*

Appendix B

Simulation block diagrams

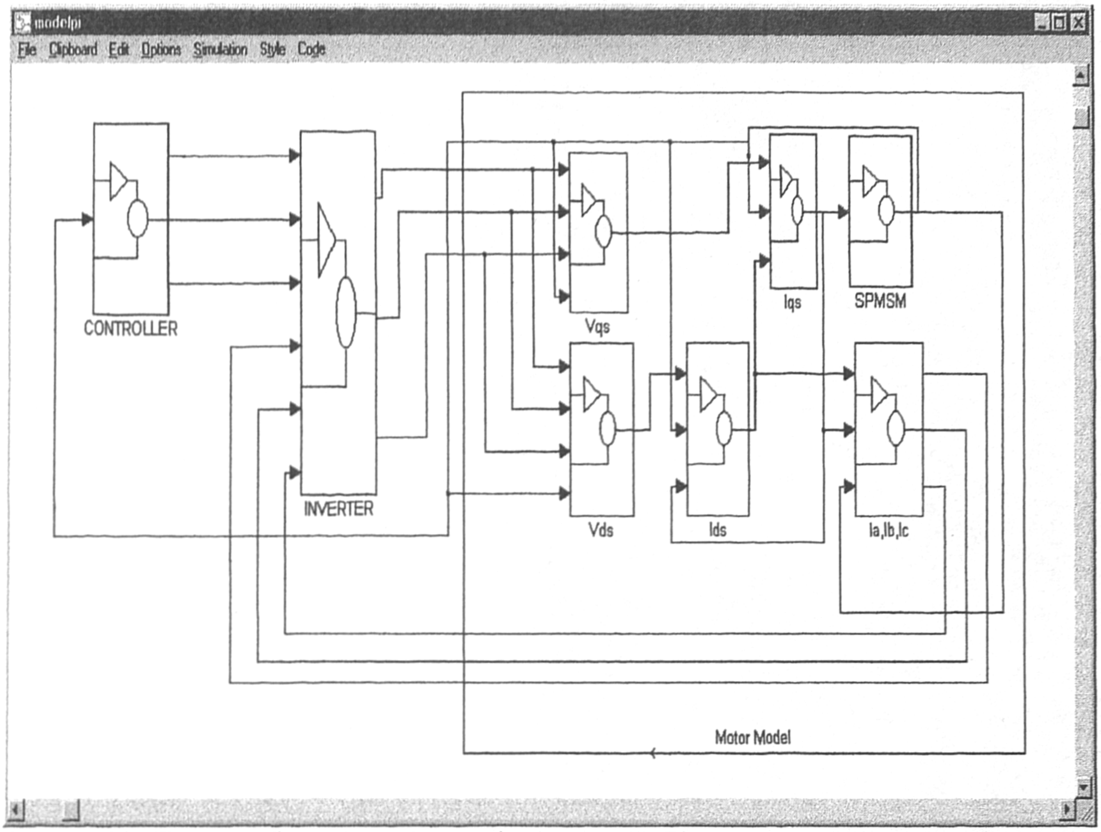


Figure B1: The rotor flux oriented control scheme with PI speed controller and hysteresis current control: Overall diagram of SPMSM drive.

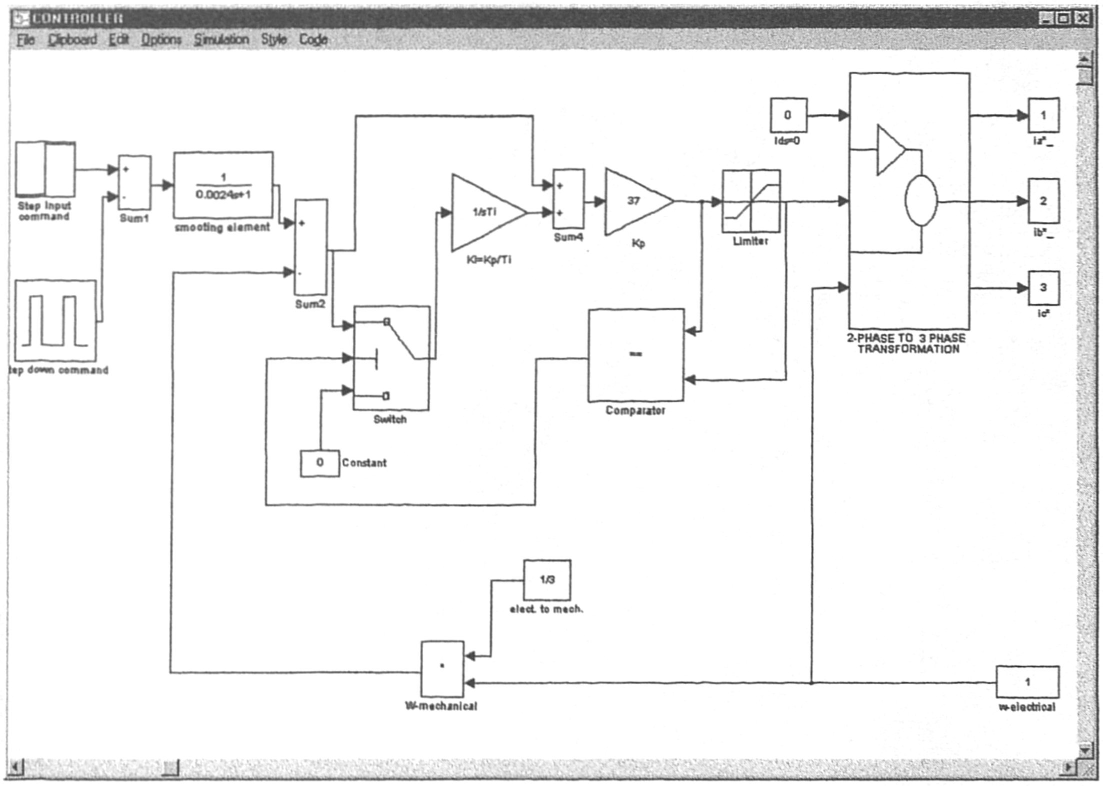


Figure B2: PI speed controller with conditional integration method: Inside the controller block of SPMSM drive.

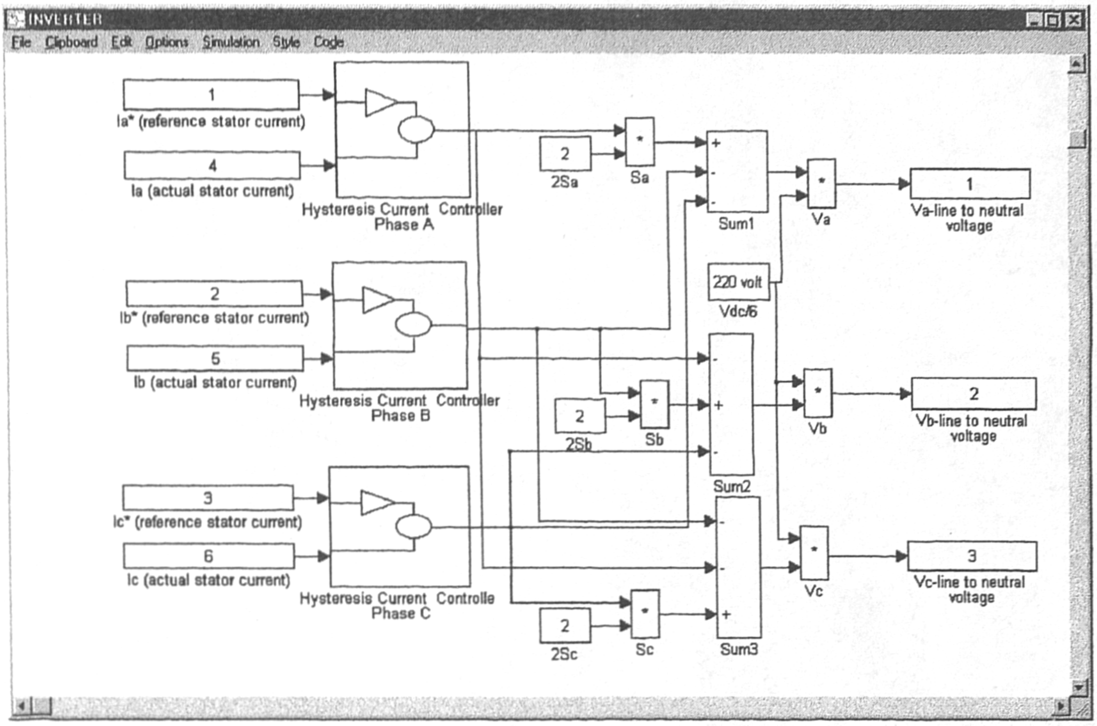

INVERTER
File Clipboard Edit Options Simulation Style Code
1
Ia* (reference stator current)
4
Ia (actual stator current)
Hysteresis Current Controller Phase A
2
Ib* (reference stator current)
5
Ib (actual stator current)
Hysteresis Current Controller Phase B
3
Ic* (reference stator current)
6
Ic (actual stator current)
Hysteresis Current Controlle Phase C
2
2Sa
Sa
2
2Sb
Sb
2
2Sc
Sc
Sum1
Sum2
Sum3
220 volt
Vdc/6
Va
Vb
Vc
1
Va-line to neutral voltage
2
Vb-line to neutral voltage
3
Vc-line to neutral voltage

Figure B3: Three phase voltage source inverter and hysteresis current controller: Inside the inverter block of SPMSM drive.

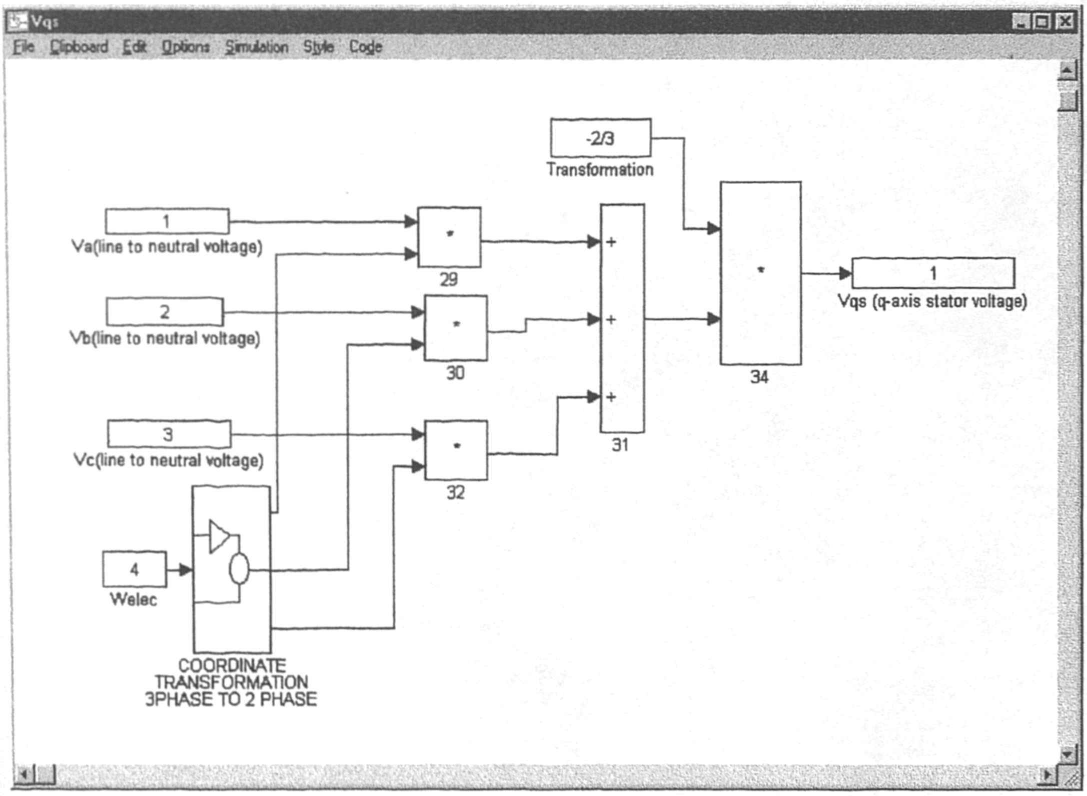

Vqs
File Clipboard Edit Options Simulation Style Code
-2/3
Transformation
1
Va(line to neutral voltage)
2
Vb(line to neutral voltage)
3
Vc(line to neutral voltage)
4
Welec
COORDINATE TRANSFORMATION 3PHASE TO 2 PHASE
29
30
32
31
34
1
Vqs (q-axis stator voltage)

Figure B4: Transformation of three phase stator voltages to q-axis stator voltage: Inside the Vqs block of SPMSM drive.

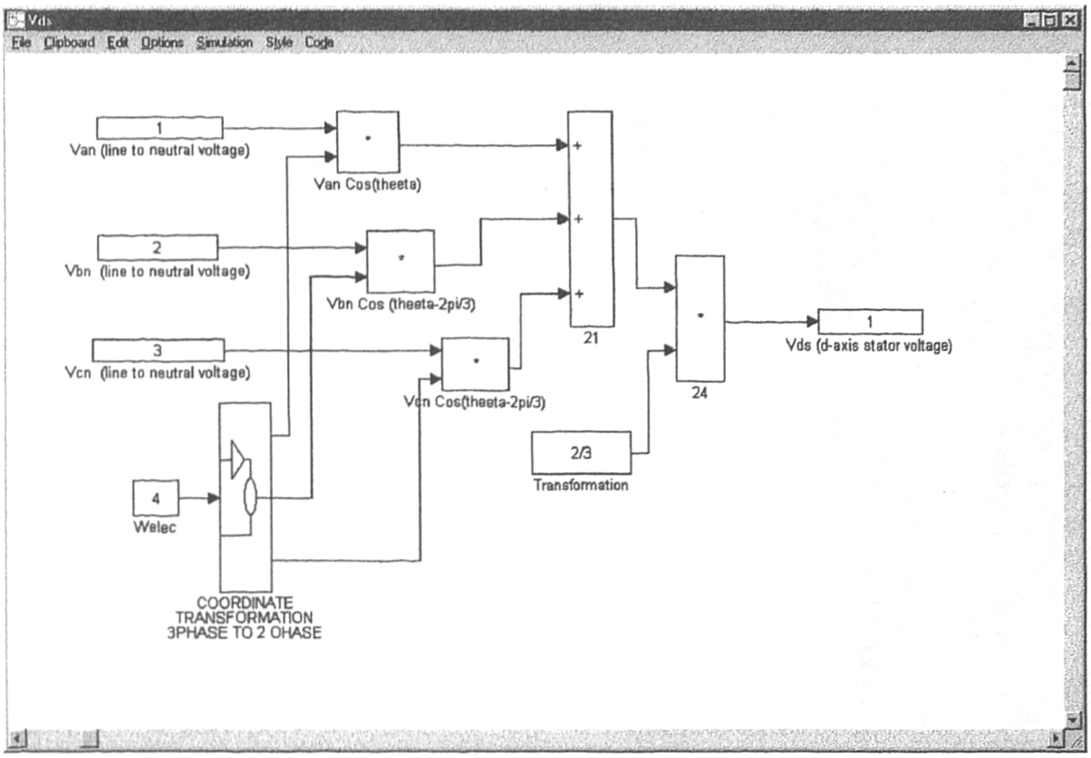

Vds
File Clipboard Edit Options Simulation Style Code
1
Van (line to neutral voltage)
Van Cos(theeta)
2
Vbn (line to neutral voltage)
Vbn Cos (theeta-2pi/3)
3
Vcn (line to neutral voltage)
Vcn Cos(theeta-2pi/3)
21
24
1
Vds (d-axis stator voltage)
2/3
Transformation
4
Welec
COORDINATE
TRANSFORMATION
3PHASE TO 2 OHASE

Figure B5: Transformation of three phase stator voltages to d-axis stator voltage: Inside the Vds block of SPMSM drive.

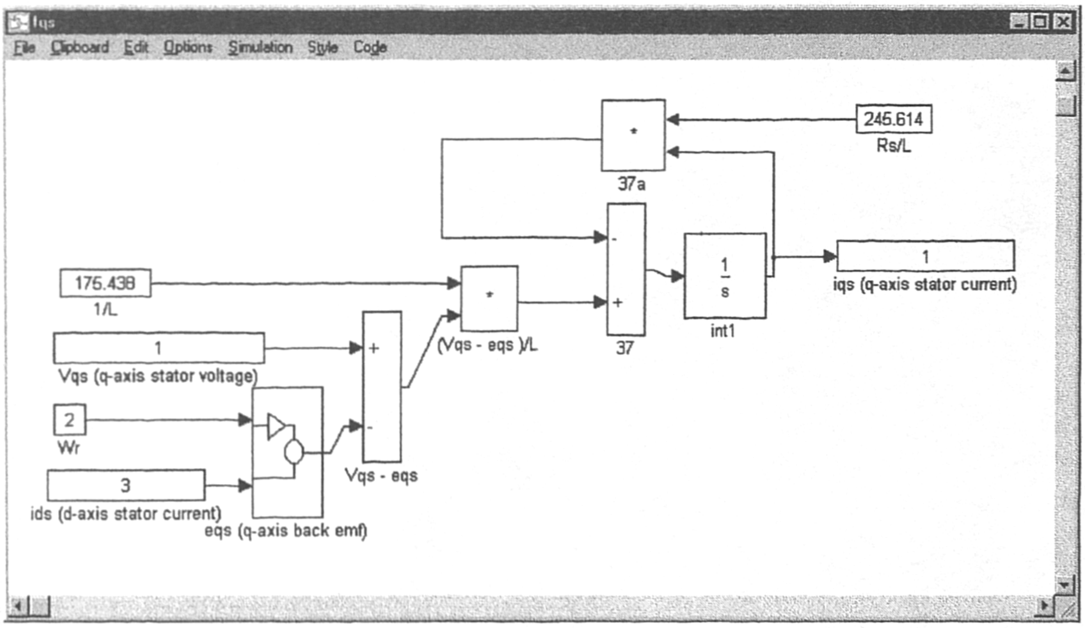

Iqs
File Clipboard Edit Options Simulation Style Code
245.614
Rs/L
37a
175.438
1/L
1
Vqs (q-axis stator voltage)
(Vqs - eqs)/L
37
1/s
int1
1
iqs (q-axis stator current)
2
Wr
3
ids (d-axis stator current)
Vqs - eqs
eqs (q-axis back emf)

Figure B6: Calculation of q-axis stator current based on two-phase stator voltages: Inside the iqs block of SPMSM drive.

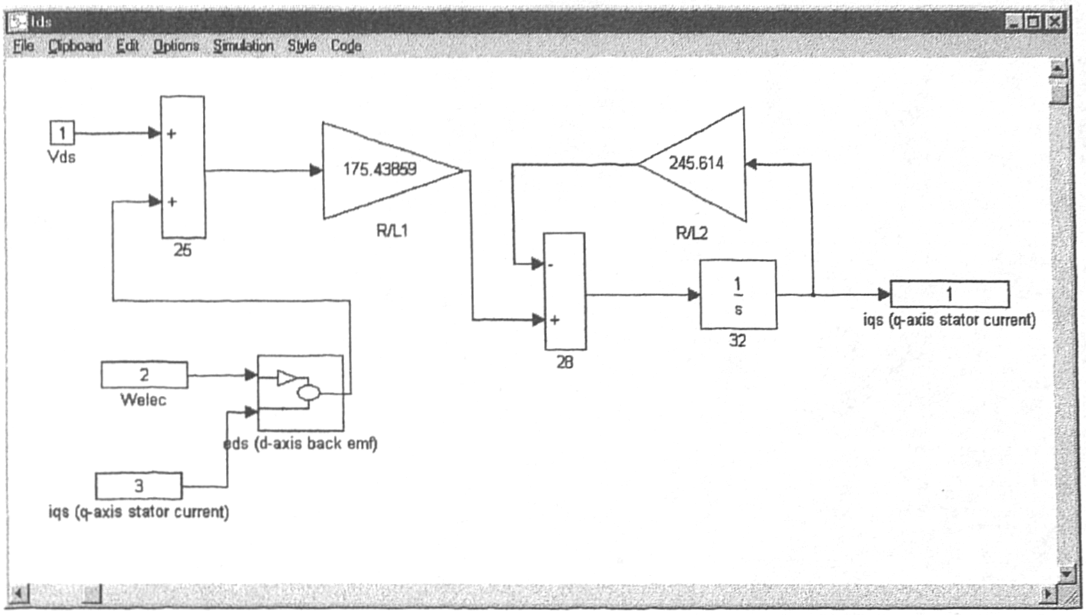

Ids
File Clipboard Edit Options Simulation Style Code
1
Vds
+
+
25
175.43859
R/L1
245.614
R/L2
-
+
28
1/s
32
1
iqs (q-axis stator current)
2
Welec
eds (d-axis back emf)
3
iqs (q-axis stator current)

Figure B7: Calculation of d-axis stator current based on two-phase stator voltages: Inside the Ids block of SPMSM drive.

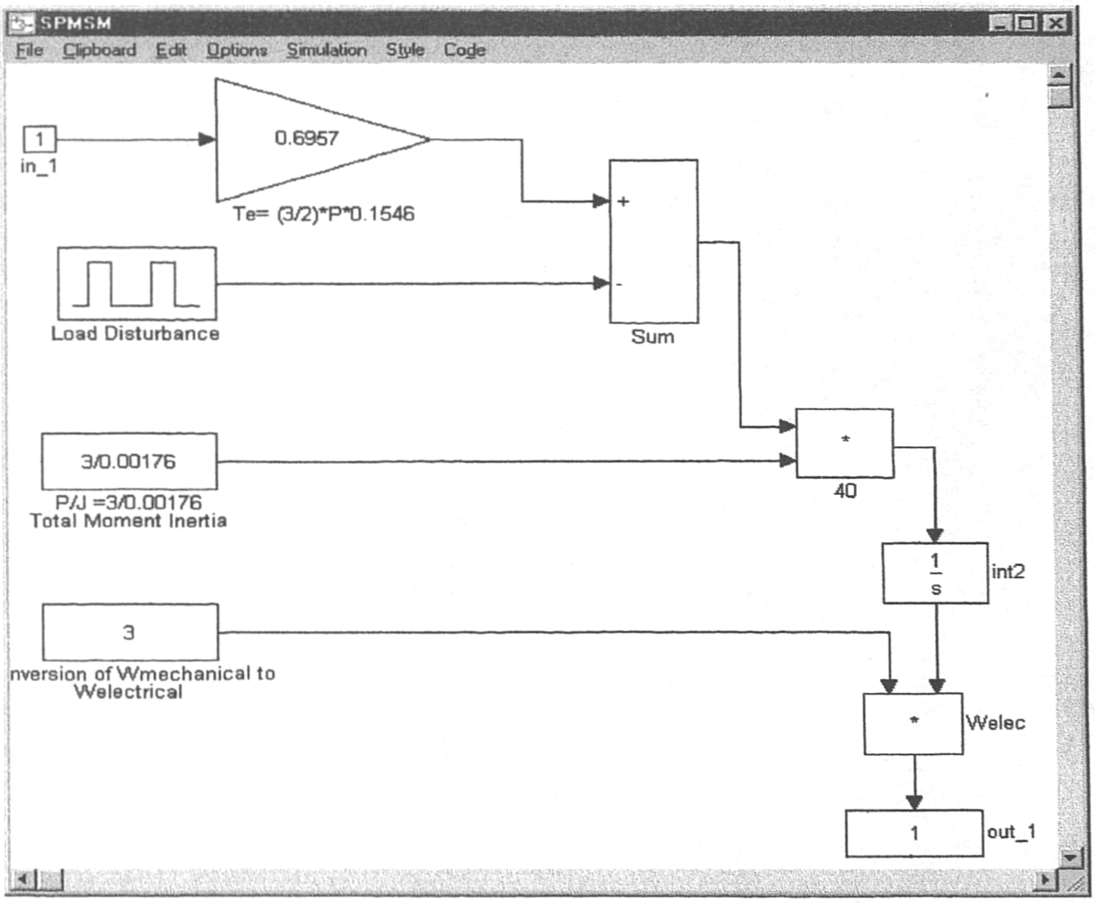


Figure B8: Electromechanical torque and rotor speed: Inside the SPMSM block of SPMSM drive.

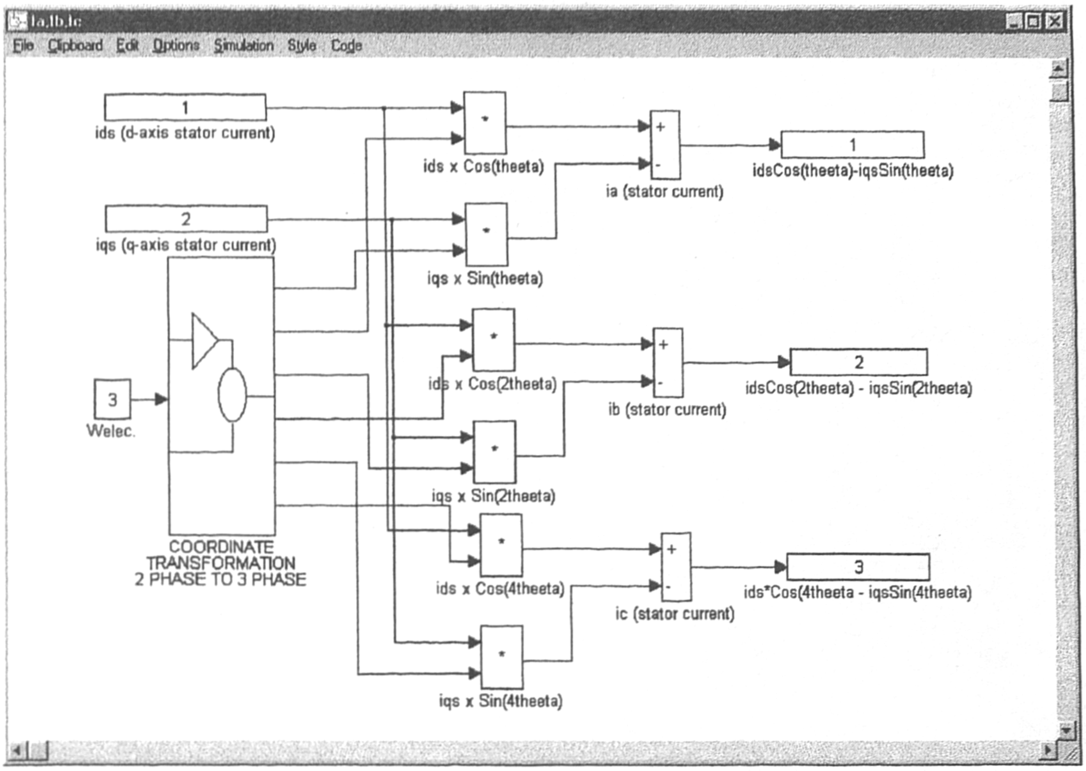
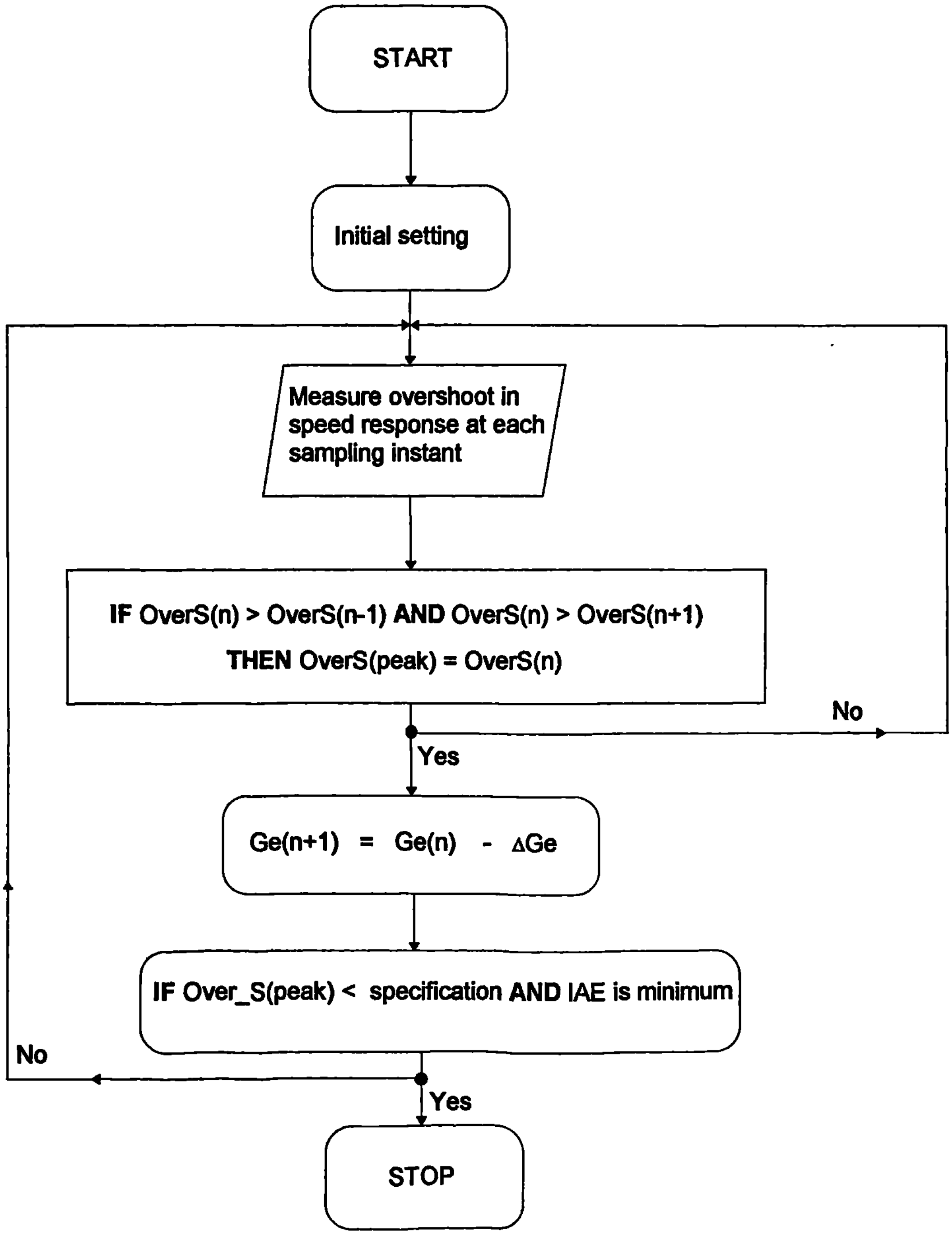


Figure B9: Transformation of two-phase stator currents to three phase stator currents: Inside the i_a, i_b, i_c block of SPMSM drive.

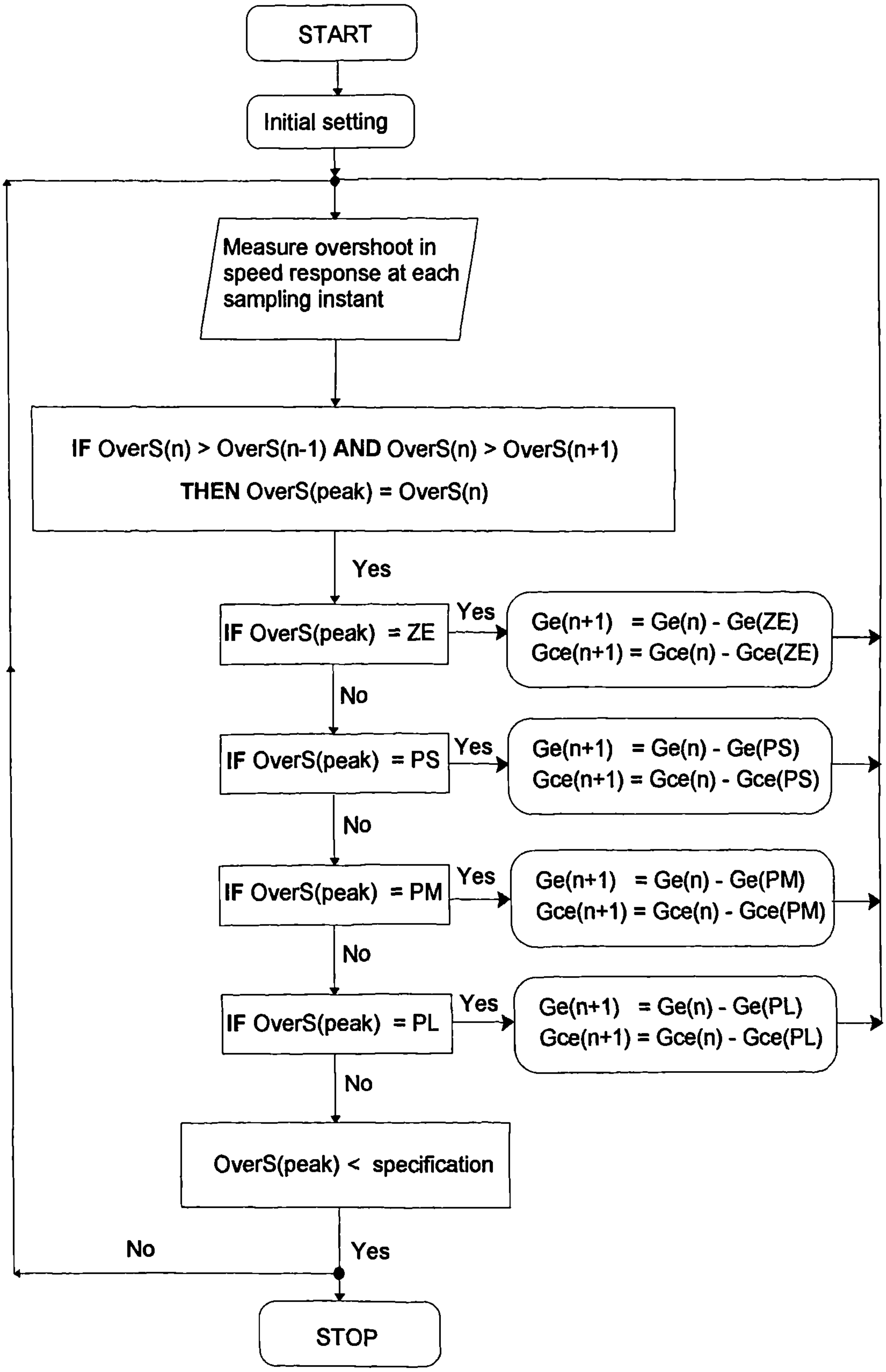
Appendix C

Flow charts

C1. Flow chart for method 1 of Chapter 6 (Section 6.4)


START
Initial setting
Measure overshoot in speed response at each sampling instant
IF OverS(n) > OverS(n-1) AND OverS(n) > OverS(n+1)
THEN OverS(peak) = OverS(n)
No
Yes
Ge(n+1) = Ge(n) - ΔGe
IF Over_S(peak) < specification AND IAE is minimum
No
Yes
STOP

C2. Flow chart for method 2 of Chapter 6 (Section 6.5)


START
Initial setting
Measure overshoot in speed response at each sampling instant
IF OverS(n) > OverS(n-1) AND OverS(n) > OverS(n+1)
THEN OverS(peak) = OverS(n)
Yes
IF OverS(peak) = ZE
Yes
Ge(n+1) = Ge(n) - Ge(ZE)
Gce(n+1) = Gce(n) - Gce(ZE)
No
IF OverS(peak) = PS
Yes
Ge(n+1) = Ge(n) - Ge(PS)
Gce(n+1) = Gce(n) - Gce(PS)
No
IF OverS(peak) = PM
Yes
Ge(n+1) = Ge(n) - Ge(PM)
Gce(n+1) = Gce(n) - Gce(PM)
No
IF OverS(peak) = PL
Yes
Ge(n+1) = Ge(n) - Ge(PL)
Gce(n+1) = Gce(n) - Gce(PL)
No
OverS(peak) < specification
No
Yes
STOP

Appendix D

Publications

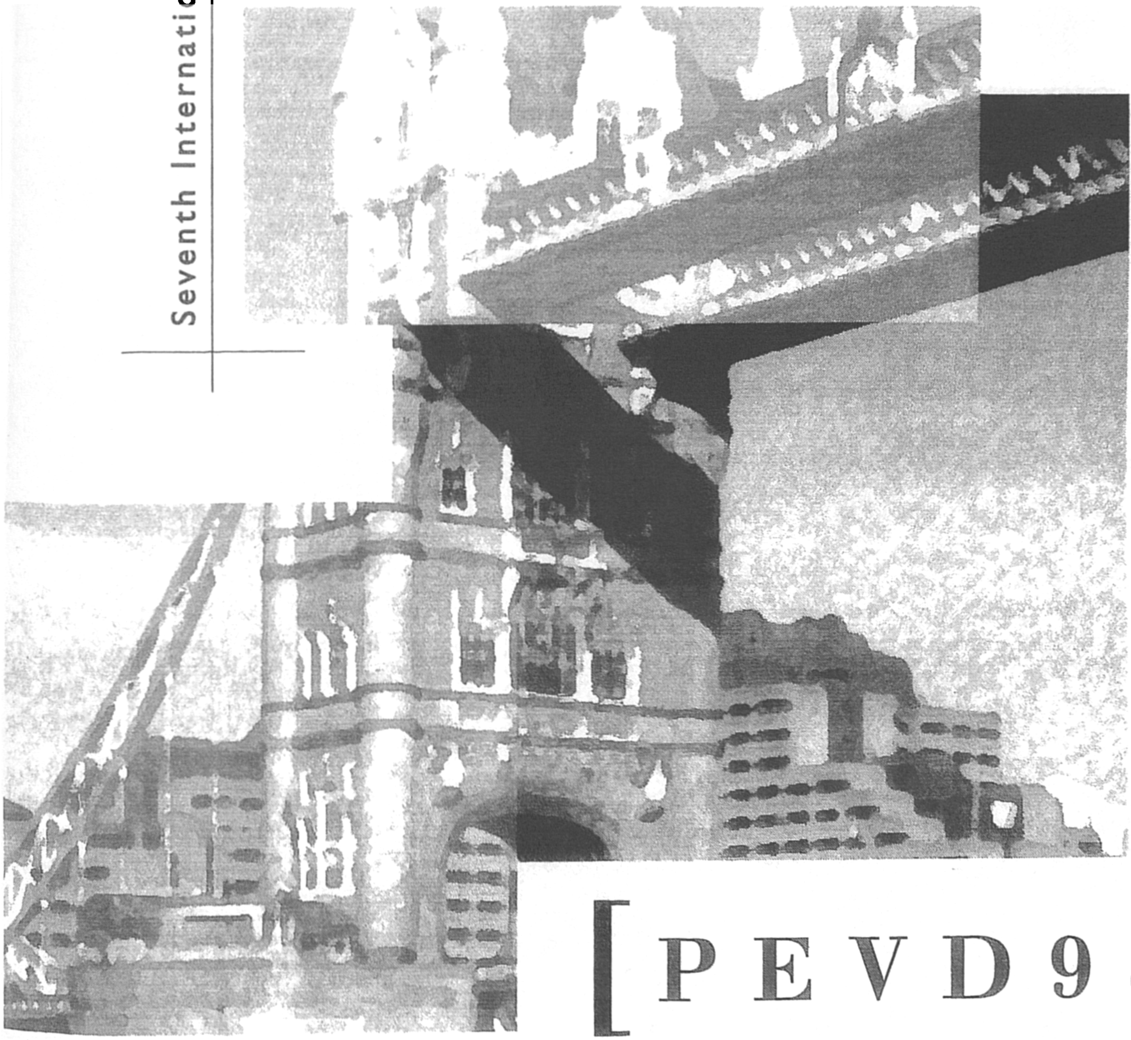
1. Ibrahim, Z., and Levi, E., (1998), "A Detailed Comparative Analysis of Fuzzy Logic and PI Speed Control in High Performance Drives", *Proceedings of IEE International Power Electronics and Variable Speed Drives Conference PEVD*, London, United Kingdom, IEE Conference Publication No. 456, pp. 638-643.

2. Ibrahim, Z., Levi, E., and Williams, D., (1998a), "Fuzzy logic versus PI speed control of servo drives: a comparison", *8th International Power Electronics & Motion Control Conference PEMC*, Prague, Czech Republic, pp. 4-34 - 4-39.

3. Ibrahim, Z., Levi, E., and Williams, D., (1998b), "A Self-Tuning Method For Fuzzy Logic Speed Controller In High Performance Drives", *33rd Universities Power Engineering Conference UPEC,* Napier University, Edinburgh, United Kingdom, pp. 819-822.

Seventh International Conference on

Power Electronics & VARIABLE SPEED DRIVES



[PEVD98]

IEE

Conference Publication No. 456

A DETAILED COMPARATIVE ANALYSIS OF FUZZY LOGIC AND PI SPEED CONTROL IN HIGH PERFORMANCE DRIVES

Z Ibrahim, E Levi

Liverpool John Moores University, UK

INTRODUCTION

A standard approach for speed control in industrial drives is to use a PI controller. Recent developments in fuzzy logic have brought into focus a possibility of replacing a PI speed controller with a fuzzy logic (FL) equivalent, Bose [1] and Bose [2].

Many publications contain comparison between the operation of a drive with speed control by PI and FL techniques (Vas et al [3], Sousa and Bose [4], Ibaliden and Goureau [5], Bossak and Bauer [6], Ficcara et al [7], Fodor et al [8], Feber et al [9], Donescu et al [10], da Silva and Acarnley [11], Baghli et al [12], Hissel et al [13]). Their main conclusion seems to be that FL control provides superior performance. In vast majority of cases however, comparison is done for a very limited selection of transients. These typically include one reference speed setting and application/removal of the load torque at one reference speed. Comparison is usually based on the controller design for aperiodic speed response, in which case PI control is known to exhibit sluggish disturbance rejection properties. More detailed comparisons are rather rare, the exceptions being [8,9,10,11]. Study reported in [11] performs initial design of the controllers in such a way that the same response to a selected step change in speed command is obtained. Such an approach enables a fair comparison to be performed further on for different operating conditions. Improvement of response obtained by FL control in [11] appears to be marginal. Similarly, [8] provides an indication that there are transients in which PI control will yield better response, while [10] shows some transients for which response of PI and FL control is essentially the same.

Vector control of a three-phase current-fed induction or permanent magnet synchronous machine (PMSM) converts the machine, from the control point of view, into its dc equivalent. Therefore, in speed control loop design, it is irrelevant whether the actual machine is a dc machine or a vector controlled ac machine and the results are universally applicable. The drive used in the study described here comprises a surface mounted permanent magnet synchronous machine with rotor flux oriented control. Current control of the PWM voltage source inverter is performed using hysteresis current controllers. The goal of the paper is to provide a detailed comparison regarding drive operation with constant parameter type PI and FL speed controllers.

Both controller types are initially tuned to yield essentially an identical speed response to the application of a step rated speed command under no-load conditions, assuming rated inertia. It is believed that only such an approach enables a fair comparison. Two initial controller designs are considered. The first one yields aperiodic speed response (no overshoot), while the second one allows for a small overshoot of 0.72% of the rated speed (1.3 rad/s). Studies of numerous transients are then undertaken by simulation. These include application of large step speed command other than rated, application of step load torque and small step-wise reference speed decrease. The speed response of the control schemes is compared and it is found that there are situations in which PI control offers a better response than the FL control. The findings of this paper confirm that situation, with regard to performance of the FL speed control, is not as clear as often presented in the literature.

DESCRIPTION OF THE SYSTEM

A PMSM drive with rotor flux oriented control is shown in Fig. 1. Data of the machine are given in Appendix. The PWM voltage source inverter is controlled by means of three independent hysteresis current controllers. The hysteresis band is constant and equal to ± 0.5 A. The inverter input dc voltage is set to 220 V, so that sufficient voltage reserve is provided for operation with good current control at rated speed and maximum torque. *Stator q-axis current is limited to 3.39 times* rated stator current. Speed PI controller is provided with anti wind-up. The FL speed controller, illustrated in Fig. 2, is of standard structure. Inputs are speed error and change of speed error. As a FL controller on its own is a PD controller equivalent, output of the speed FL controller is integrated in order to yield PI like behaviour. Also, an equivalent anti wind-up feature is included.

The PI controller is designed first, on the basis of the speed response to the step rated speed command (180 rad/s) under no-load conditions assuming rated inertia. Two designs are performed. The criterion for the first one is a speed overshoot less than 0.1 rad/s. The second design allows for an overshoot of 1.3 rad/s. In both cases minimum settling time is required, taking into account the limit on the stator current maximum value.

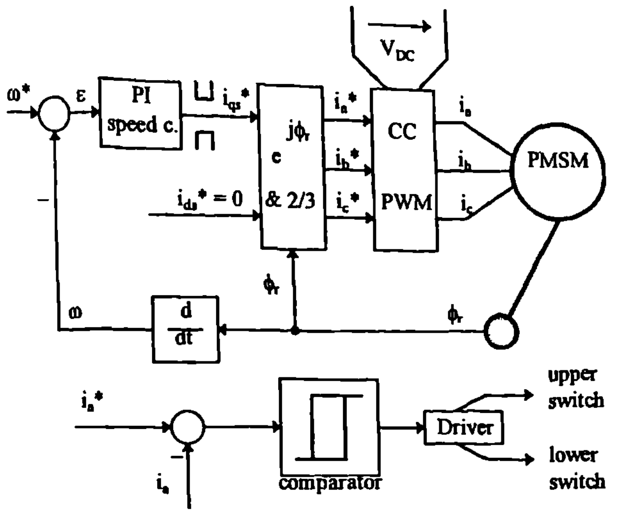


Fig. 1: Configuration of a rotor flux oriented PMSM drive and illustration of a hysteresis current controller.

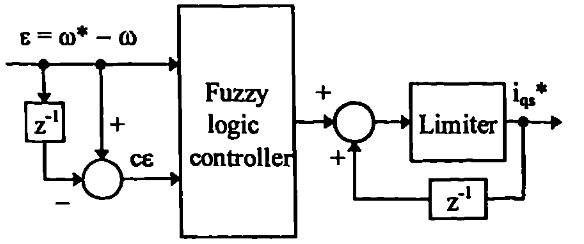


Fig. 2: Fuzzy logic speed controller.

Ziegler - Nichols method is used for determining the initial settings of the PI controller parameters; fine tuning is then achieved by trial and error.

Two FL controllers are designed next. The FL controller consists of triangular membership functions with overlap, seven for speed error and seven for the change of speed error, so that a 7x7 rule base is created. The width of the membership functions is reduced in vicinity of zero error, zero change of error and zero stator q-axis current command in order to obtain better steady-state accuracy. Input and output scaling factors are selected, through many simulation runs, in such a way that speed response to the step rated speed command matches as closely as possible the response obtained with the corresponding PI controller. Design of the PI like FL speed controller closely parallels guidelines given by Zheng [14].

Further investigations include: response to the large step speed command, application of the step rated load torque and a small step decrease in reference speed setting. Figures 3 shows a sample of simulation results that applies to operation with FL speed controller, designed for 1.3 rad/s overshoot. The test sequence is as follows: 1) Rated step speed command is applied at zero time under no-load conditions; 2) Rated load torque is applied at t = 0.025 sec.; 3) Speed reference is reduced to 0.9 of the previous setting at t = 0.08 sec.

SIMULATION RESULTS

Comparison of the drive behaviour under PI and FL speed control is performed by overlapping and zooming speed responses of the type shown in Fig. 3. Figure 4 illustrates comparison of speed responses to the step application of the rated speed command, obtained for the two design cases with PI (bold trace in all the figures) and FL control. It verifies that both controllers yield identical speed response for zero overshoot design. For 1.3 rad/s overshoot design both controllers yield the same overshoot, while settling time of the FL controller is slightly shorter. It is believed that responses for the both design cases are sufficiently similar to enable a fair and thorough comparison between the controllers for other operating regimes.

Figure 5 applies to initial controller design for zero overshoot. Response to the step change of speed reference from zero to 40 rad/s (Fig. 5a), response to the step application of the rated load torque at speed equal to 90 rad/s speed (Fig. 5b) and response to small reference speed change from 60 rad/s to 0.9 times the previous setting (Fig. 5c) are shown. As can be seen from Fig. 5a overshoot obtained with both controllers is the same, while FL controller needs longer time to achieve steady-state operation. Load torque application, Fig. 5b, results in the same dip in speed with both controllers; however, FL control returns the speed to the reference value much quicker. Response to the small reference speed change, Fig. 5c, is similar with both controllers, except that FL controller needs more time to establish new steady-state. One can conclude from Fig. 5 that PI control yields a better response for selected large and small reference speed changes (Figs. 5a, 5c), while FL control provides better disturbance rejection (Fig. 5b).

Similar type of study is reported in Fig. 6, where results apply to the design with 1.3 rad/s overshoot. Step change of speed reference from zero to 30 rad/s (Fig. 6a), step rated load torque application at 120 rad/s speed (Fig. 6b) and small reference speed change from 100 rad/s to 0.9 times the previous setting (Fig. 6c) are displayed. Response to small reference speed change has the same first undershoot with both PI and FL control. However, FL control gives a subsequent overshoot which is not present with PI control, so that PI control results in better overall response. Response to large speed reference change (Fig. 6a) is superior with PI control. Load torque application causes a smaller dip in speed with PI control (Fig. 6b), although restoration time is slightly shorter with FL control.

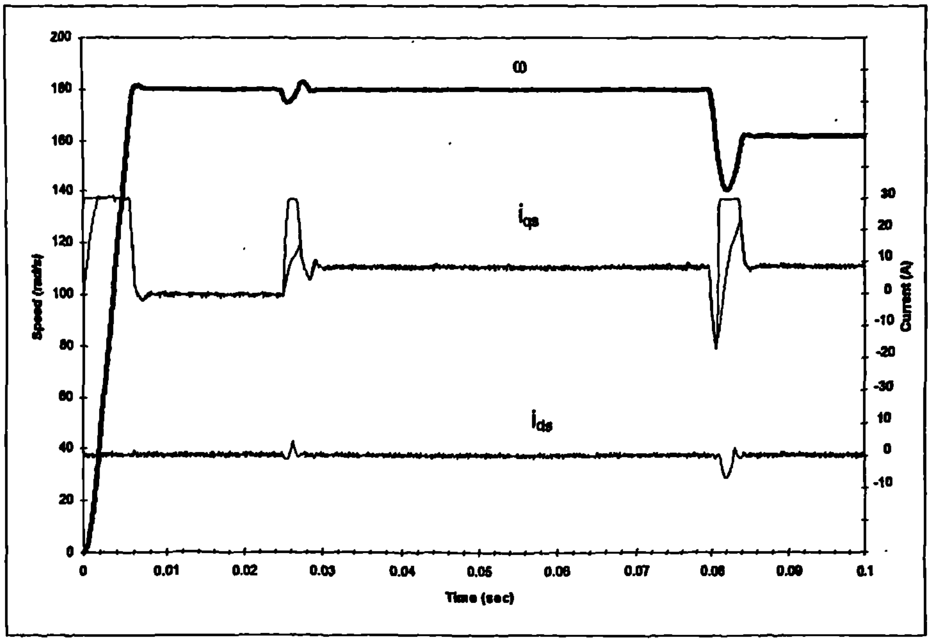


Fig. 3: Response of the drive to step rated speed command, application of rated load torque and step reduction of the speed reference to 0.9 times the previous setting with FL speed control (speed control design for 1.3 rad/s overshoot). Speed and stator d-q axis currents (actual and reference values) are shown.

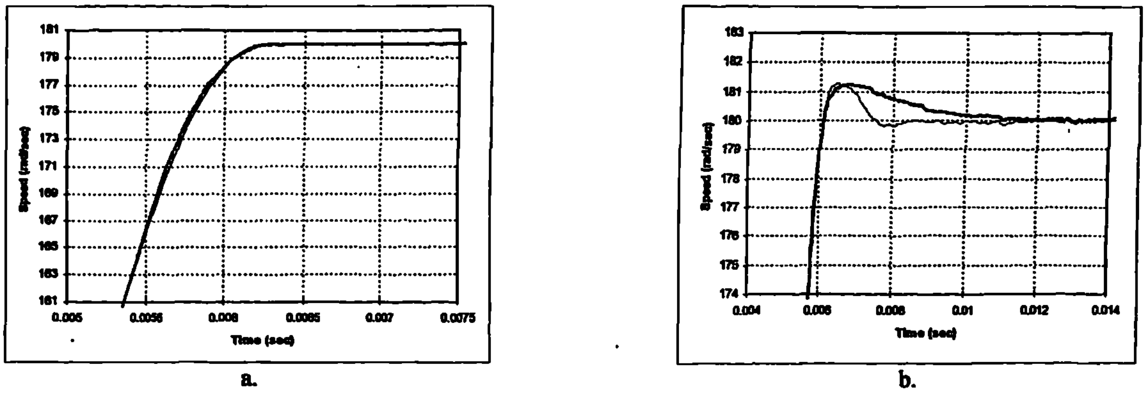
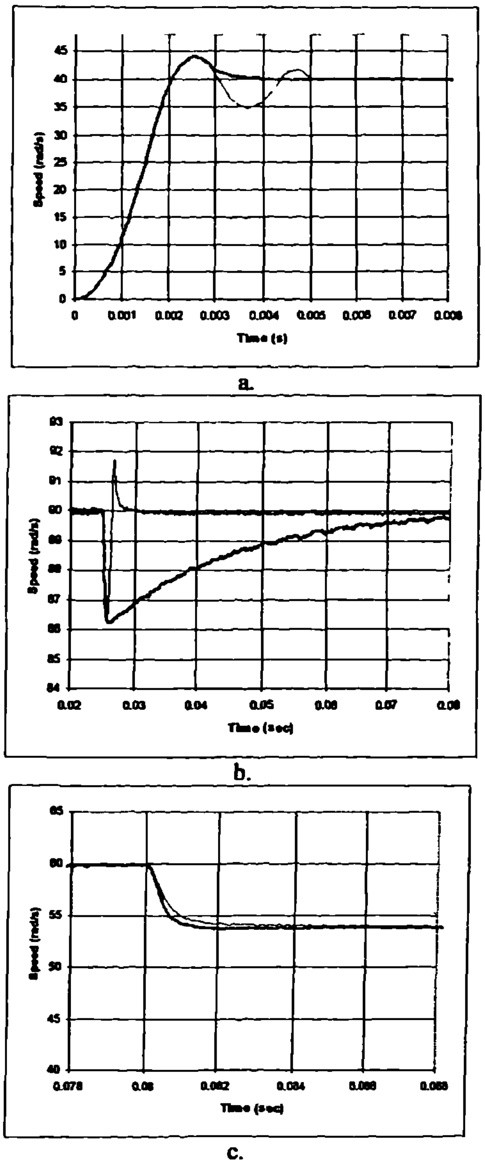


Fig. 4: Comparison of speed response to application of the rated speed command, obtained with PI and FL speed control, for a) zero overshoot design and b) 1.3 rad/s overshoot design.

The results reported in Figs. 5 and 6 are deliberately selected in such a way that operation with PI control appears to be better than operation with FL control. Indeed, FL control gives superior response only in Fig. 5b (load torque application, zero overshoot design). In all the other five transients, illustrated in Figs. 5 and 6, PI control yields better speed response. Hence the only viable conclusion that one could arrive at on the basis of Figs. 5 and 6 is that PI control is to be preferred to FL control. Such a conclusion would however be quite erroneous and the previous analysis indicates how misleading improper selection of transients for comparison can be. The overall situation, across the entire speed control range, is much more complex, as shown next.

The same type of study, reported in Fig. 3, is performed over the range of operating speeds from 10 rad/s for zero overshoot design and 20 rad/s for 1.3 rad/s overshoot design up to the rated speed (180 rad/s). Overshoot in speed response for large reference speed change, dip due to load torque application and undershoot that follows small reference speed change are measured for PI and FL control, together with the duration of the transient (which is taken as the time needed for the speed error to become smaller than 0.1 rad/s). Results are summarised in Fig. 7 for the zero overshoot design and in Fig. 8 for the 1.3 rad/s overshoot design. Every attempt was made to provide as accurate readings as possible in compiling data given in Figs. 7 and 8.

Response to the large step speed reference change is basically the same with PI and FL control for zero overshoot design, Fig. 7a, for speed commands between 120 and 180 rad/s. FL control is superior between 40 and :20 rad/s, while PI control is better up to 40 rad/s. Disturbance rejection is better with FL control in all the

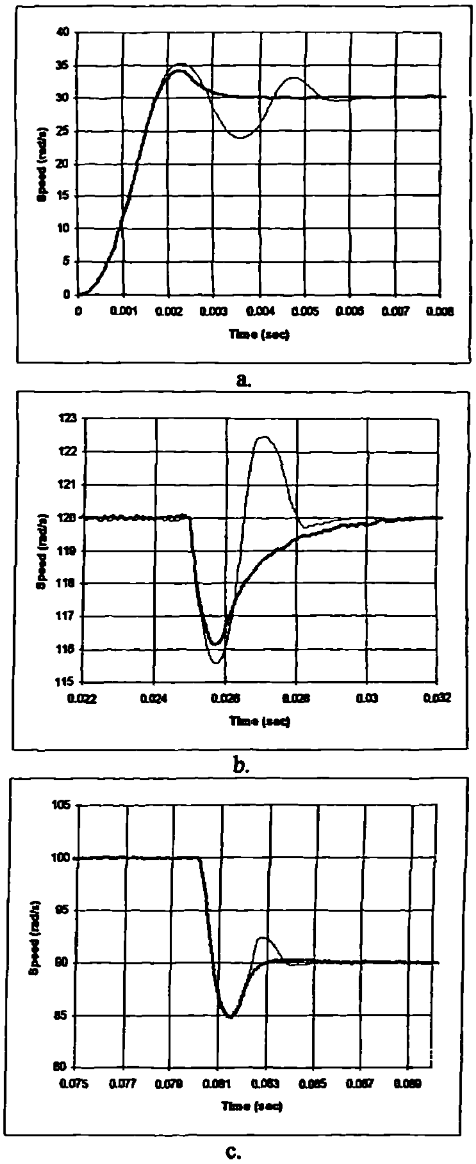


a.

b.

c.

Fig. 5: Comparison of PI and FL speed control response for zero overshoot design: a) step change of speed reference from zero to 40 rad/s; b) rated load torque application at 90 rad/s speed; c) change of speed reference from 60 rad/s to 0.9 times previous setting.



a.

b.

c.

Fig. 6: Comparison of PI and FL speed control response for 1.3 rad/s overshoot design: a) step change of speed reference from zero to 30 rad/s; b) rated load torque application at 120 rad/s speed; c) change of speed reference from 100 rad/s to 0.9 times previous setting.

cases (Fig. 7b). Response to small step speed reference change (Fig. 7c) is better with PI control for initial speeds up to 60 rad/s; from 60 to 180 rad/s response is better with FL control. Response to the large step speed reference change, with 1.3 rad/s overshoot design (Fig. 8a), is better with PI control for speeds up to 35 rad/s; above this value FL control gives better response, especially in terms of settling time. Note however that in initial design FL control has shorter settling time, Fig. 4b. This appears to be reflected throughout Fig. 8a and shows how important it is to provide the same response of the two controllers for initial design. Disturbance rejection of the PI and FL control is similar for speeds from 140 to 180 rad/s, Fig. 8b. FL control is superior below 140 rad/s in terms of restoration time. Response to small reference speed change, Fig. 8c, is marginally better with FL control.

CONCLUSION

The paper compares operation of a high performance drive with PI and FL speed control. Two initial designs of speed controllers are considered and it is emphasised that any attempt to provide a fair comparison must aim at achieving the same response with both controllers for the design case. Next, it is shown how a selective approach to choice of transients that are used to underpin certain desirable conclusion (usually superiority of FL control) can be misleading. Detailed comparison of performance over the entire speed range for two initial designs shows that, in general, there are speed regions where PI control provides better response, as well as regions where FL and PI control yield more or less the same response. The major conclusion of this

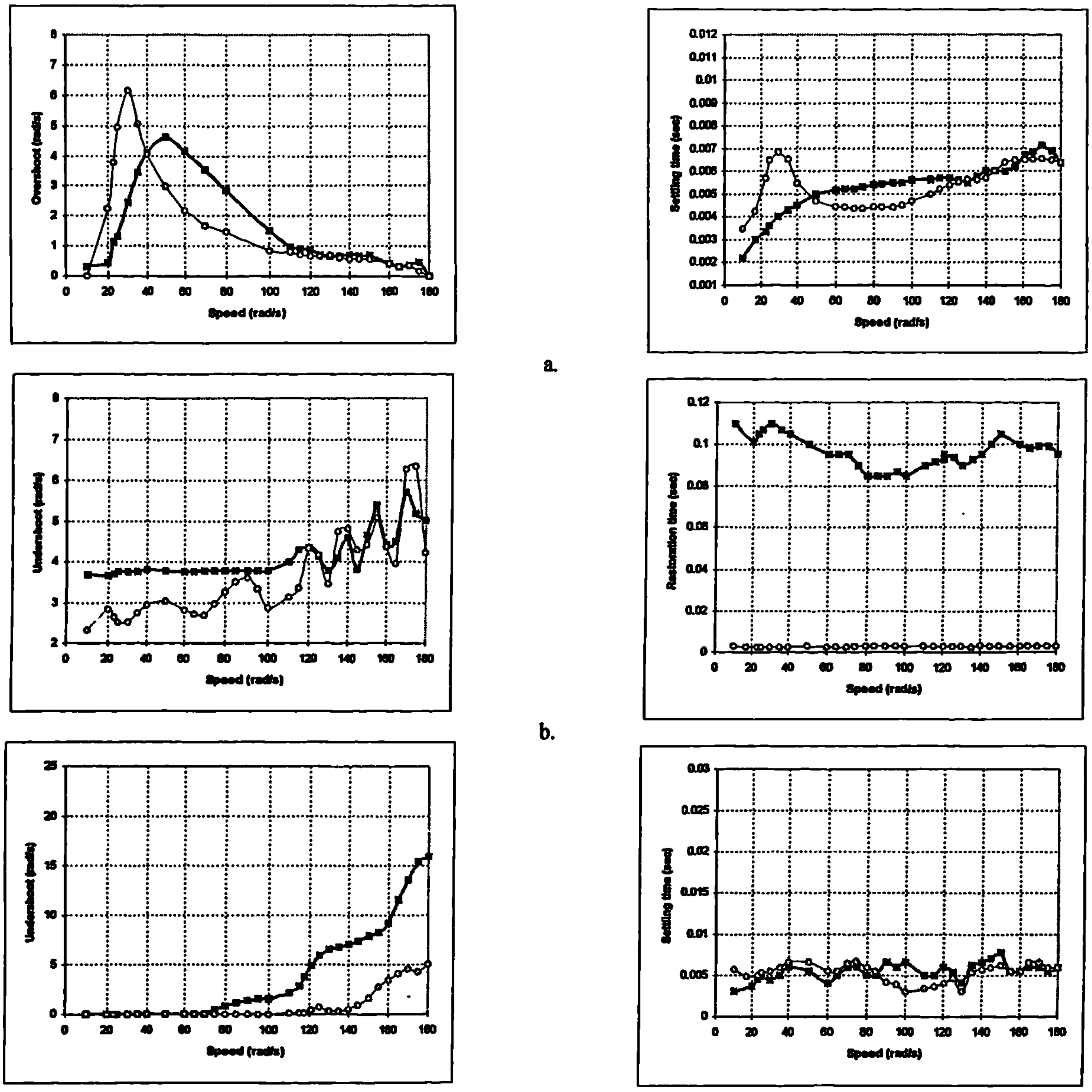


Fig. 7: Comparison of PI and FL speed control over the entire speed region for zero overshoot design:
a) speed overshoot and settling time for step application of large speed command;
b) dip in speed and restoration time for rated load torque application;
c) speed undershoot and settling time for small step-wise change in speed command.

study is therefore that certain care needs to be exercised when claiming superiority of the FL control. Such a conclusion should not be arrived at on the basis of comparison for a single operating point. Undoubtedly though, overall performance with FL control does appear to be better than with PI control.

REFERENCES

[1] Bose BK, 1994, "Expert systems, fuzzy logic, and neural network applications in power electronics and motion control", Proceedings of the IEEE, 82, pp. 1303-1323.

[2] Bose BK, 1997, "Intelligent control and estimation in power electronics and drives", IEEE Int. Electric Machines and Drives Conf., pp. TA2-2.1-TA2-2.6.

[3] Vas P, Chen J, Stronach AF, 1994, "Fuzzy control of dc drives", 25th Int. Power Conversion and Intelligent Motion Conf. PCIM, pp. 11-31.

[4] Sousa GCD, Bose BK, 1994, "A fuzzy set theory based control of a phase-controlled converter DC machine drive", IEEE Trans. on Industry Applications, 30, pp. 34-44.

[5] Ibaliden A, Goureau P, 1996, "Fuzzy robust speed control of induction motor", Int. Conf. on Electr. Machines ICEM, vol. III, pp. 168-173.

[6] Bossak M, Bauer M, 1996, "Robust speed and position control of an indirect field oriented controlled induction motor drive using fuzzy logic regulator", Int. Conf. on Electr. Machines ICEM, vol. I, pp. 219-224.

[7] Ficcara MC, Eguliaz JMM, Peracaula J, 1996, "Fuzzy control of an induction motor with compensation of

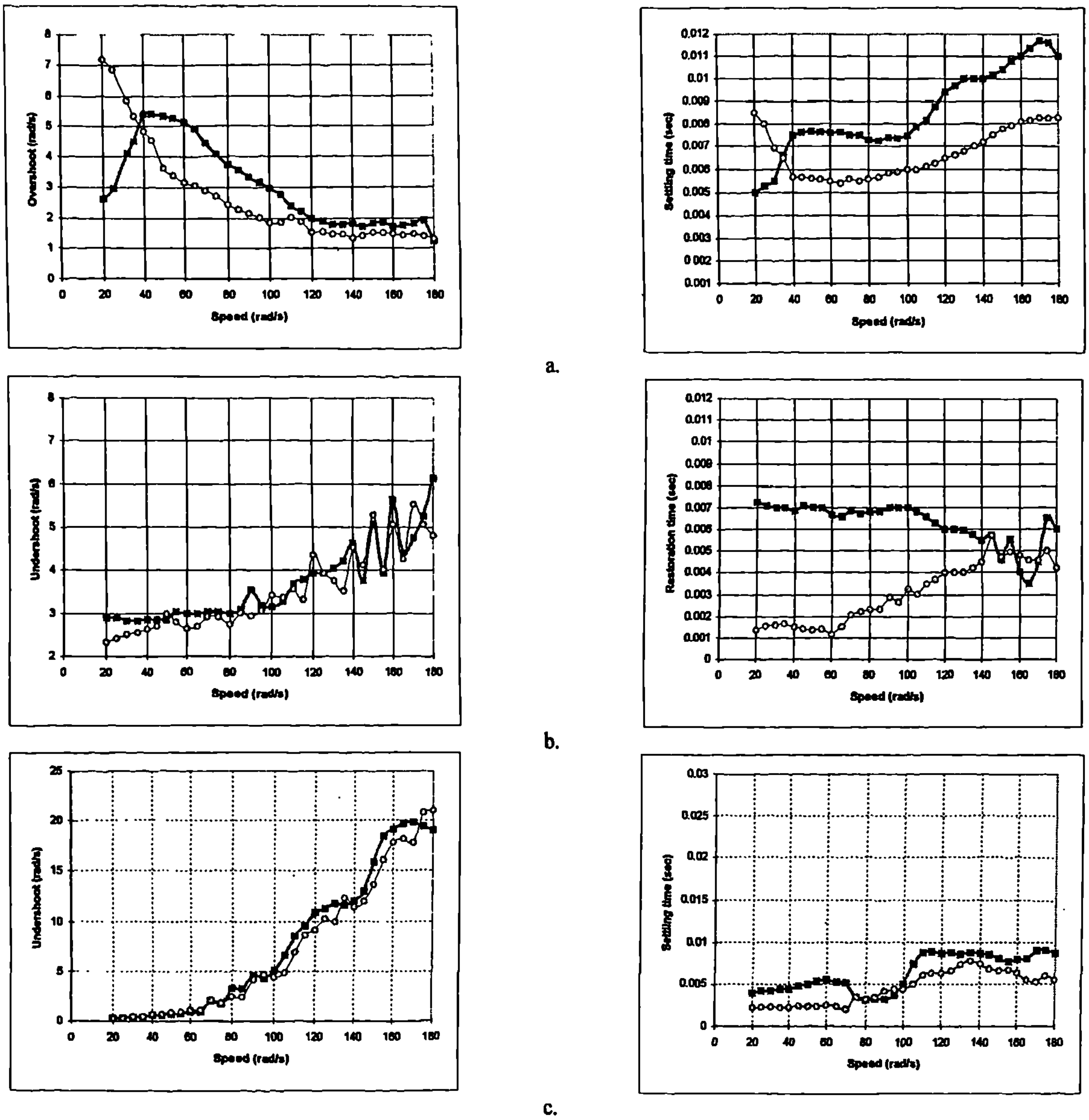


Fig. 8: Comparison of PI and FL speed control over the entire speed region for 1.3 rad/s overshoot design:
a) speed overshoot and settling time for step application of large speed command;
b) dip in speed and restoration time for rated load torque application;
c) speed undershoot and settling time for small step-wise change in speed command.

system dead-time", IEEE Power Elec. Spec. Conf. PESC, pp. 677-681.

[8] Fodor D, Katona Z, Vass J, 1996, "On fuzzy logic speed control for vector controlled AC motors", 4th Int. Workshop on Advanced Motion Control AMC'96-MIE, pp. 186-191.

[9] Heber B, Xu L, Tang Y, 1995, "Fuzzy logic enhanced speed control of an indirect field oriented induction motor drive", IEEE Power Elec. Spec. Conf. PESC, pp. 1288-1294.

[10] Donescu V, Neacsu DO, Griva G, Profumo F, 1996, "A systematic design method for fuzzy logic speed controller for brushless DC motor drives", IEEE Power Elec. Spec. Conf. PESC, pp. 689-694.

[11] da Silva WG, Acarnley PP, 1997, "Fuzzy logic controlled dc motor drive in the presence of load disturbance", 7th Eur. Conf. Power Elec. & Appl. EPE, pp. 2.386.-2.391.

[12] Baghli L, Razik H, Rezzoug A, 1997, "Comparison between fuzzy and classical speed control within a field oriented method for induction motors", 7th Eur. Conf. Power Elec. & Appl. EPE, pp. 2.444-2.448.

[13] Hissel D, Maussion P, Gateau G, Faucher J, 1997, "Fuzzy logic control optimization of electrical systems using experimental designs", 7th Eur. Conf. Power Elec. & Appl. EPE, pp. 1.090-1.095.

[14] Zheng L, 1992, "A practical guide to tuning of proportional and integral (PI) like fuzzy controllers", IEEE Int. Conf. on Fuzzy Systems, pp. 633-640.

APPENDIX: Motor data

$R_s = 1.4\ \Omega$	$L_s = 5.6$ mH	$\psi_m = 0.1546$ Vs / rad
$\omega_n = 180$ rad / s	2P = 6	$J_n = 0.00176$ kgm^2
$I_n = 6.2$ A	$I_{\max} = 21$ A	
$T_{en} = 6.1$ Nm	$T_{e\max} = 20.7$ Nm	

PEMC`98

Organized by
CZECH TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING

In cooperation with
CZECH ELECTROTECHNICAL SOCIETY
ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

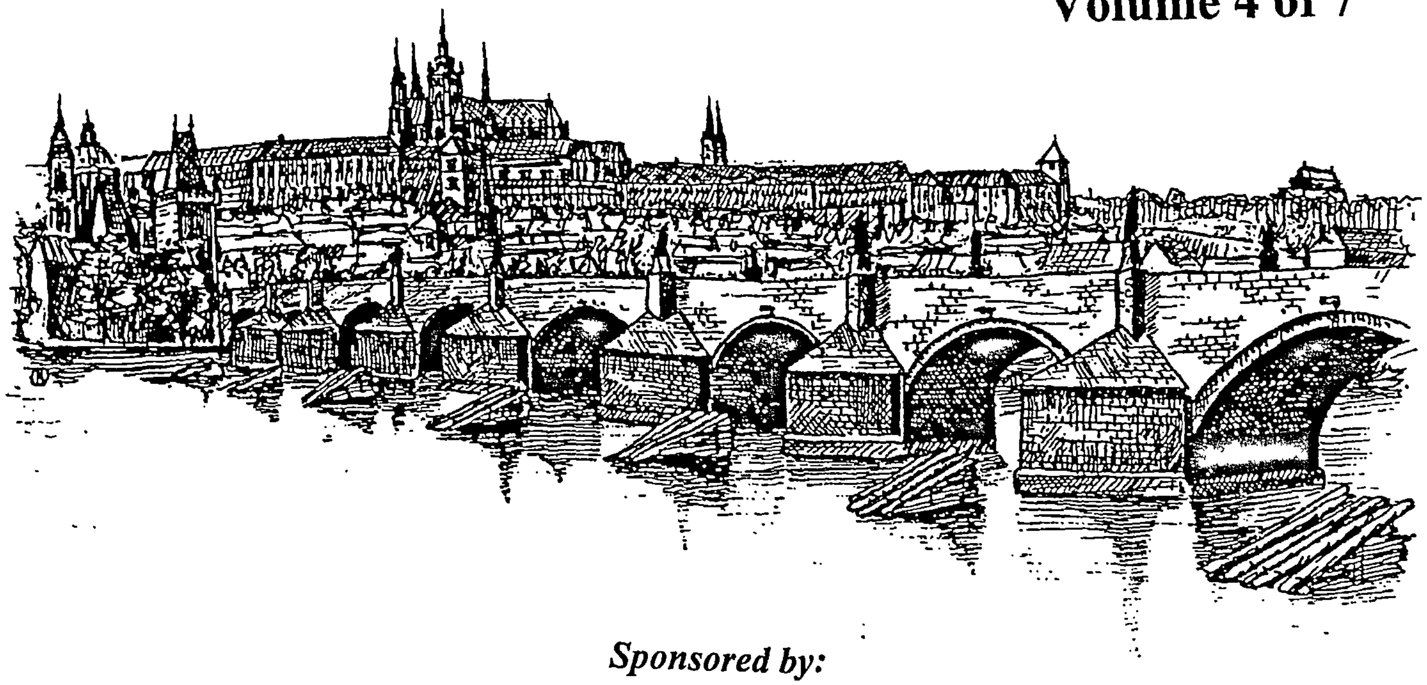
Within the framework of
EUROPEAN POWER ELECTRONICS & DRIVES ASSOCIATION

Proceedings of the

8th International Power Electronics & Motion Control Conferei

8 - 10 September, 1998
PRAGUE - CZECH REPUBLIC

Volume 4 of 7



Sponsored by:

ČKD PRAHA HOLDING, a.s.
SIEMENS AG

FUZZY LOGIC VERSUS PI SPEED CONTROL OF SERVO DRIVES: A COMPARISON

Z.Ibrahim, E.Levi, D.Williams

Liverpool John Moores University, UK

Abstract. Fuzzy logic control has recently attracted considerable interest in the electric drive area. In particular, the possible replacement of proportional plus integral (PI) speed controller with a fuzzy logic (FL) controller has been widely considered, with the common conclusion being that performance of a servo drive can be greatly enhanced by the use of a FL speed controller. This paper attempts to provide a thorough comparative insight into the behaviour of a servo drive controlled by PI and FL speed controllers. A rotor flux oriented permanent magnet synchronous machine is simulated under varying operating conditions. It is shown that although the application of an FL speed controller does yield better performance in some cases, there are situations when either the PI controller is superior, or when both controllers are equally poor.

Keywords. Fuzzy logic control, PI control, Servo drives.

INTRODUCTION

A standard approach for speed control in industrial drives is to use a PI controller. Recent developments in fuzzy logic have brought into focus a possibility of replacing a PI speed controller with a fuzzy logic (FL) equivalent [1,2]. Numerous publications are available that describe the development of a FL speed controller and its simulation and experimental verification in conjunction with high performance dc drives [1-6], vector controlled induction [3,7-14] and permanent magnet synchronous machines [15-17], and switched reluctance motors [18].

Many of the publications [1-18] contain comparison between the operation of a drive with speed control by PI and FL techniques. Their main conclusion seems to be that FL control provides superior performance [4,6-15]. However, results from experiments and simulations, that form the basis of such a statement in many papers, often do not seem to provide enough evidence for this conclusion [4,6-9,11,12]. Examples of attempts to provide an in-depth comparison are rather rare [10,13,15]. Furthermore, numerous difficulties encountered in FL speed controller applications, such as design difficulties [1,15,17], steady-state speed error [5,7,14] and chattering [5] are rarely described.

Vector control of a three-phase current-fed induction or permanent magnet synchronous machine converts the machine, from the control point of view, into its dc equivalent. Therefore, in speed control loop design, it is irrelevant whether the actual machine is a dc machine or a vector controlled ac machine and the results are universally applicable. The drive used in the study described here comprises a surface mounted permanent magnet synchronous machine with rotor flux oriented control. Current control of the PWM voltage source inverter is performed in stationary reference frame, using hysteresis current controllers. The goal of the paper is to attempt to provide a detailed comparison regarding drive operation with constant parameter type PI and FL speed control.

Both these controllers are initially tuned to yield essentially an identical speed response to the application of a step rated speed command under no-load conditions, assuming rated inertia. Studies of numerous transients are then undertaken by simulation. These include: application of step speed command other than rated, application of step load torque (disturbance rejection) and operation with inertia other than rated. The speed response of these control schemes is compared and it is found that there are situations in which PI control offers a better response than the FL controller. The findings of this paper support the conclusion that situation, with regard to performance of the FL speed controller, is not as clear as often presented in the literature.

DESCRIPTION OF THE SYSTEM

A permanent magnet synchronous motor drive with rotor flux oriented control is illustrated in Fig. 1. All the relevant data of the machine are given in Appendix. The PWM voltage source inverter is controlled by means of three independent hysteresis current controllers. The hysteresis band is adjusted to ± 0.5 A (i.e., ± 5.7% of the rated current) and is kept at this value in all the simulations. The inverter input dc voltage is set to the constant value of 220 V, so that sufficient voltage reserve is provided for operation with good current control at rated speed and maximum torque. Stator q-axis current is limited in accordance with the maximum allowed stator current rms value. Speed PI controller is provided with an anti wind-up mechanism.

The FL speed controller, illustrated in Fig. 2, is of standard structure with inputs of speed error and change of speed error. As a FL controller on its own is essentially a PD controller equivalent, output of the speed FL controller is integrated in order to yield PI like behaviour. Also, an equivalent anti wind-up feature is included.

Initially, the PI controller is designed, on the basis of the speed response to the step rated speed command (180 rad/s) under no-load conditions with rated combined inertia of the motor and the load (J_n value given in Appendix). The design criteria are set in terms of a speed overshoot less than 0.1 rad/s and minimum rise time considering the limited current capability of the inverter (essentially, aperiodic speed response). Ziegler - Nichols method is used for determining the initial settings of the PI controller parameters; fine tuning to the specification is then achieved by trial and error.

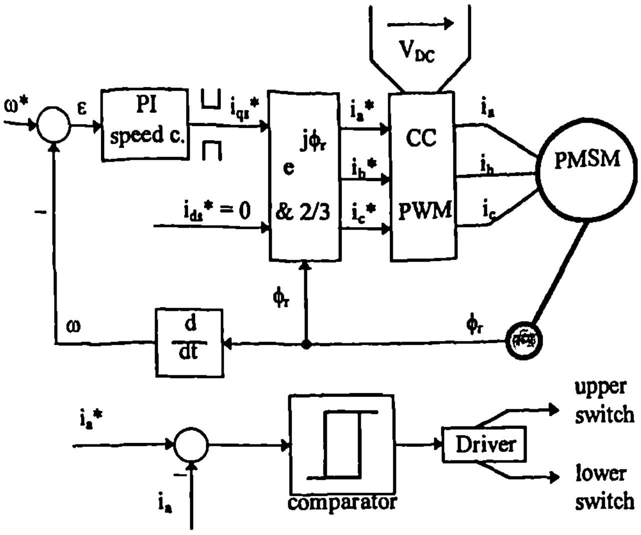


Fig. 1: Configuration of a rotor flux oriented permanent magnet synchronous motor (PMSM) drive and illustration of a hysteresis current controller.

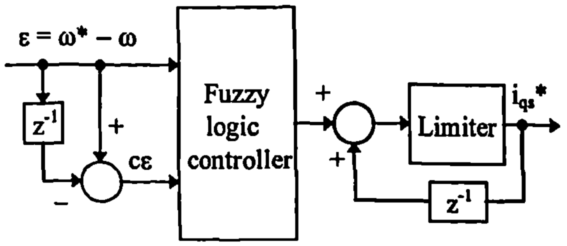


Fig. 2: Fuzzy logic speed controller.

The fuzzy logic controller consists of triangular membership functions with overlap, seven for speed error and seven for the change of speed error, so that a 7x7 rule base is created. The width of the membership functions is reduced in vicinity of zero error, zero change of error and zero stator q-axis current command in order to obtain better steady-state accuracy. Input and output scaling factors are selected, through many simulation runs, in such a way that speed response to the step rated speed command matches as closely as possible the response obtained with PI controller. Design of the PI like fuzzy logic speed controller closely parallels guidelines given in [19].

Further transient investigations include: response to the large step speed command (prolonged operation in the limit of stator q-axis current required), application of the step load torque equal to rated and a small step change in reference speed setting (operation in the limit of stator q-axis current only short-term or not required). Figures 3a and 3b show a sample of simulation results and apply to operation with PI and FL speed controllers, respectively. The test sequence is as follows:

1) Rated step speed command is applied at zero time under no-load conditions;

2) Rated load torque is applied at time instant $t = 0.025$ sec.;

3) Speed reference is reduced to 0.9 of the previous setting at time instant $t = 0.08$ sec.

The speed response, stator d-axis current and reference and actual stator q-axis current are shown.

SIMULATION RESULTS

It is difficult to draw any viable conclusions from traces of Fig. 3 regarding comparison of the drive behaviour under PI and FL speed control. The speed response traces are therefore overlapped and zoomed. The same approach is utilised further in displaying simulation results that apply to other operating conditions.

'Design' case

Figure 4 shows zoomed speed responses to the three transients depicted in Fig. 3. In all the figures bold trace represents response obtained with PI speed control. Response of the PI control to step speed command of 180 rad/s results in no overshoot, while FL control yields an overshoot of less than 0.1 rad/s. Rise and settling times of the two controllers are effectively the same. The two responses are believed to be close enough, so that a fair comparison is possible for other operating regimes.

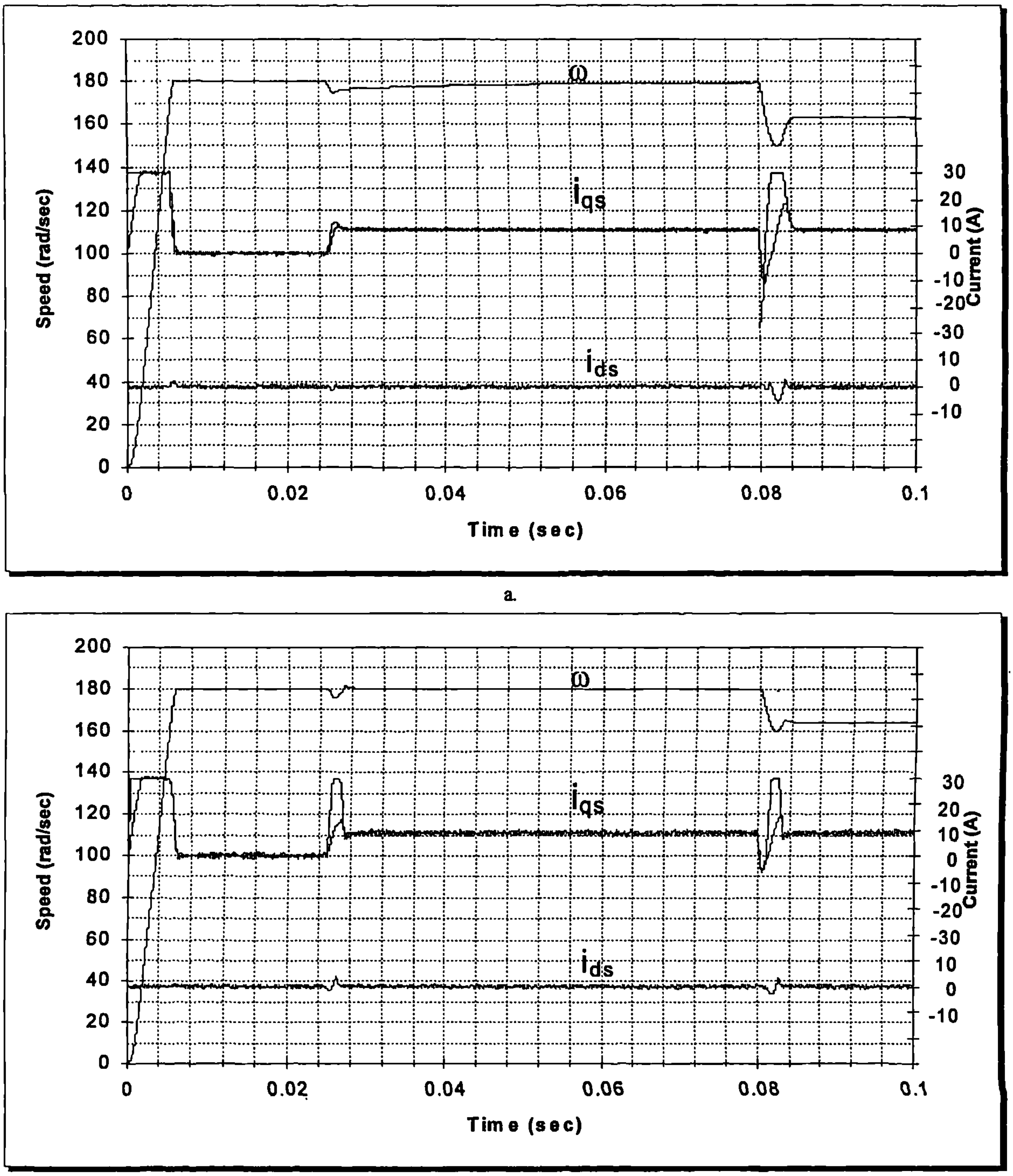
The disturbance rejection comparison in Fig. 4 shows that with PI control the initial dip in speed is larger. PI control response is aperiodic and FL controller restores operation to rated speed command more quickly. The load rejection capability of the FL controller is therefore far superior. The nature of response to the reference speed change from rated to 0.9 time rated speed is similar with both PI and FL controller. However, the speed undershoot with FL control is significantly smaller, although the time needed to achieve the new steady-state is the same for both controllers.

Different initial step speed command setting

The response to step changes in speed command, subsequent rated load torque application and step-wise reduction of speed command to 0.9 times the previous value is investigated next for two arbitrarily selected speed commands: one third of the rated speed (60 rad/s) and one sixth of the rated speed (30 rad/s). A comparison of results for these two cases is illustrated in Figs. 5 and 6.

When the step speed reference is reduced, an overshoot appears in the speed response of both controllers. For a speed reference of 60 rad/s, PI controller generates a larger overshoot, while the settling time in both cases is again effectively the same (Fig. 5a). However, at a reference speed of 30 rad/s, the response of the FL controller is highly oscillatory with a significantly larger overshoot and settling time (Fig. 6a). Thus it appears that response of the PI controller, for low reference speed settings, is far better. The response to the load application remains similar in both cases (Figs. 5b and 6b) to the one in Fig. 4, indicating superiority of the FL controller with respect to disturbance rejection.

The response to a small reference speed change of the PI controller improves as the initial speed command is altered. Indeed, the time needed to achieve a new steady-state is significantly shorter with PI control in Fig. 5c, with no undershoot. Similar conclusions apply to the case illustrated in Fig. 6c, where a small reference speed change is initiated at $t = 0.12$ s.



a.

b.

Fig. 3: Response of the drive to step rated speed command, application of rated load torque and step reduction of the speed reference to 0.9 of the previous setting (a. PI speed controller, b. FL speed controller).

Variation of inertia

Two tests incorporating the inertia change are conducted. At first, it is assumed that inertia is ten times higher than the one used for controller design. The response to rated speed command and subsequent rated load torque application (at t = 0.2 s) is compared for this case in Fig. 7. As expected, operation of the FL speed controller is in this case superior. This is indeed the situation that is most frequently discussed in the literature and used to underpin the conclusion that operation of the drive with FL speed control is far better than operation with PI control. PI speed controller yields a higher overshoot with much longer settling time for the rated speed command and higher undershoot with longer restoration time for load torque application.

The second test is the same as the previous one, except that now it is assumed that inertia is only 0.3 times the value used in the controller design. Controller design normally relies on combined inertia of the motor and load. Therefore, in the case of loads with variable inertia, inertia can both reduce and increase with respect to the design value. Comparison of results for this decrease in system inertia is shown in Fig. 8. It is evident that now both controllers give poor, highly oscillatory response to rated speed command. However, the

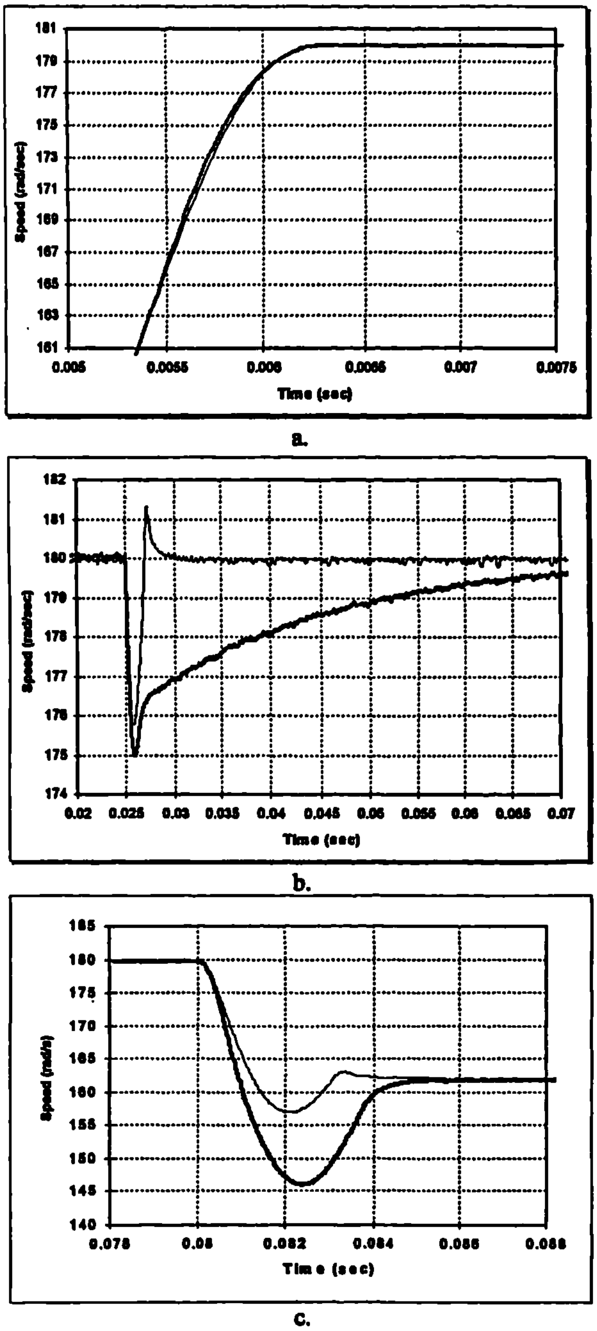


Fig. 4: Comparison of responses with PI and FL speed control (zoomed extracts from Fig. 3): response to rated step speed command (a.), step rated torque application (b.) and 10% change of speed command (c.).

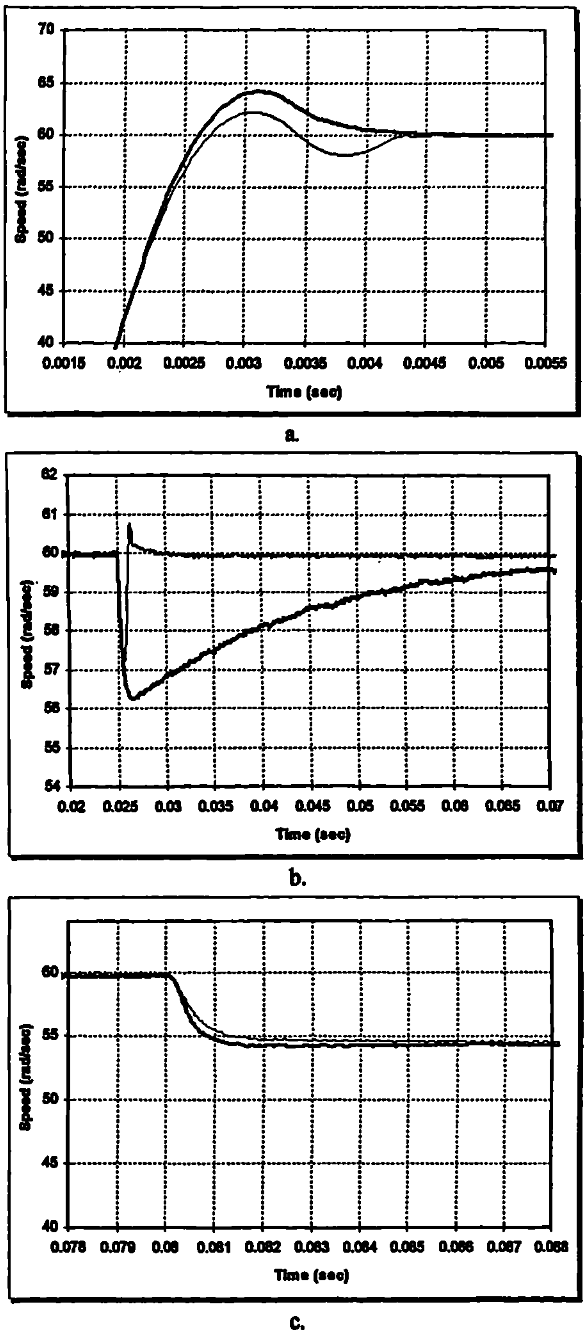


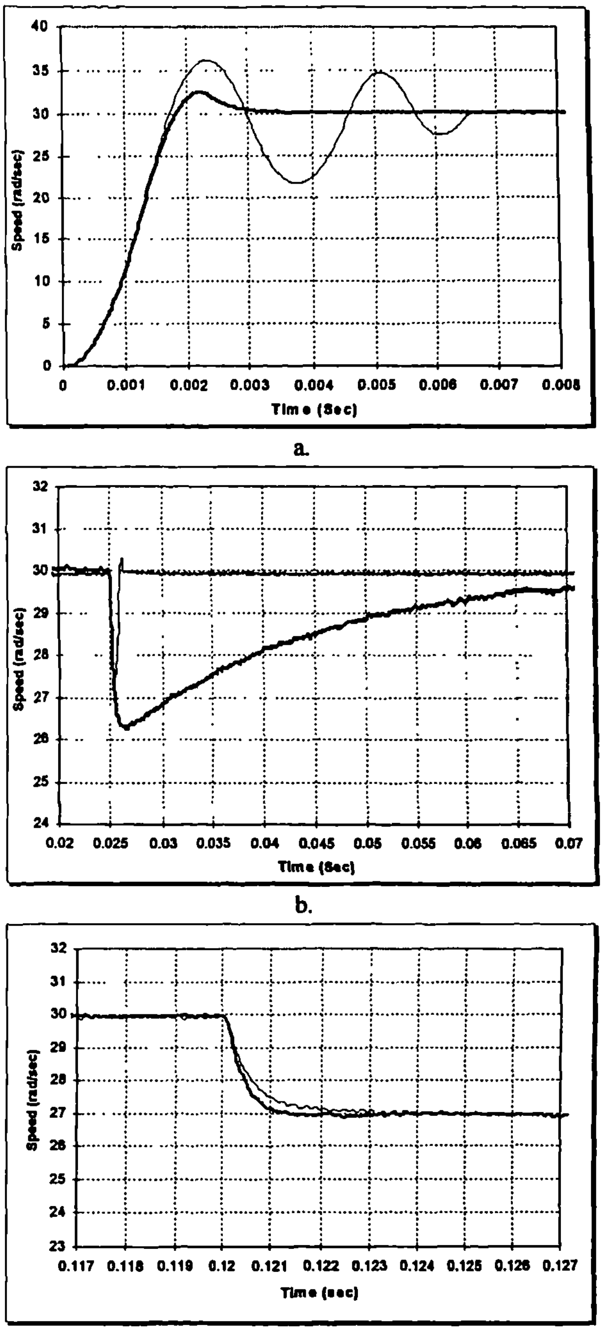
Fig. 5: Comparison of responses with PI and FL speed control: response to step speed command of 60 rad/s (a.), response to step rated torque application (b.) and response to change of speed command from 60 rad/s 54 rad/s (c.).

FL controller takes more time to establish steady-state operation. Undershoot in speed response to rated load torque application is the same for both controllers. PI controller has a long restoration time, while FL controller leads to oscillations in speed response.

DISCUSSION

On the basis of the simulation results presented in the preceding section, it is possible to state the following:

- FL speed control has a better disturbance rejection properties, provided that inertia is equal to or higher than the one used in the controller design.
- FL controller provides superior response to a large speed command if inertia of the drive is higher than the value used in the controller design.
- PI speed control provides superior response to certain, in this study relatively low values of the step speed command, with significantly smaller overshoot and shorter settling time. For other values of speed reference, response obtained with FL controller is better.
- Response to small reference speed change is sometimes better with PI controller and in other cases it is better with FL controller. This depends on the initial speed value and on the amount of the desired speed reference change.
- Response of both controllers is sluggish when inertia is significantly smaller than the one used in the controller



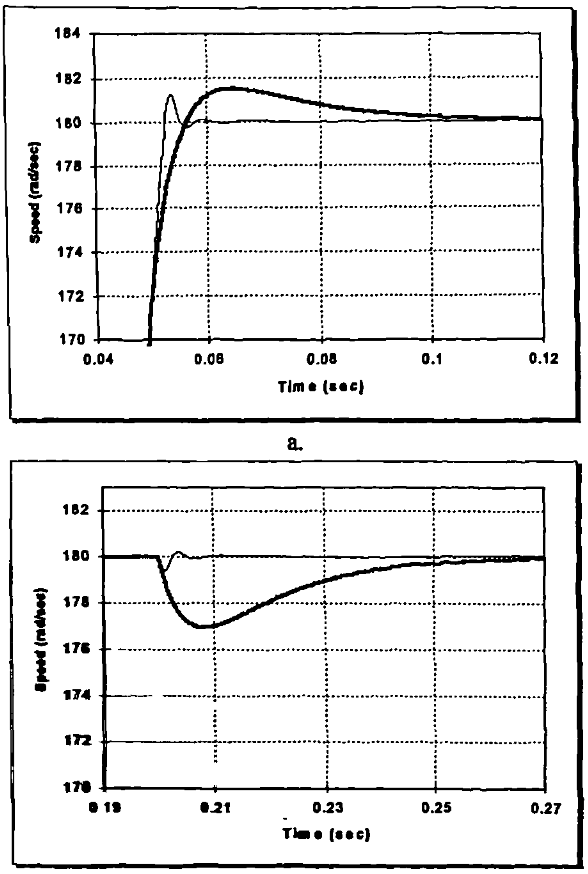
c.

Fig. 6: Comparison of responses obtained with PI and FL speed controllers: response to step speed command of 30 rad/s (a.), response to step rated torque application (b.) and response to change of speed command from 30 rad/s to 0.9 times 30 rad/s (c.).

design. PI control gives a better response to a step change in speed command, while FL control yields insignificantly better response to load torque application.

CONCLUSION

An attempt is made in this paper to compare performance of a servo drive with PI and FL speed control. Both speed controllers are initially designed to yield essentially an identical speed response to a step rated speed command. Operation is further investigated by simulation for a number of cases: different large step speed commands, load torque application and small step speed command changes. Sensitivity to inertia variation is also examined.



b.

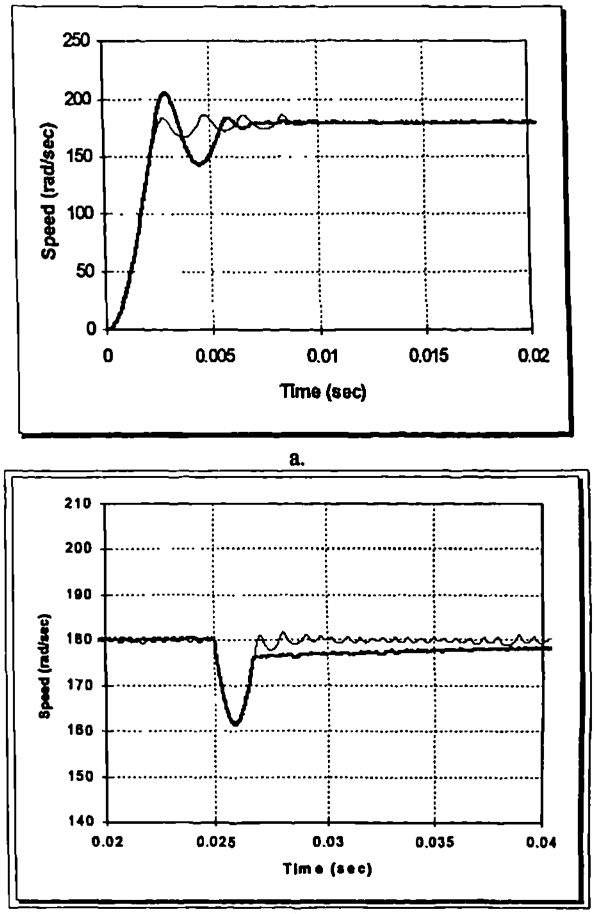
Fig. 7: Comparison of responses obtained with PI and FL speed controllers: response to rated step speed command (a.) and response to step rated torque application (b.) (inertia ten times higher than in controller design).

Indeed, as frequently claimed, FL controller performance is superior when inertia is higher than the one used for controller design. Similarly, disturbance rejection is better with FL controller, provided that the inertia is at least equal to the rated value. However, response to a large step speed command with PI controller is found to be better for relatively low settings of the step speed command. Similarly, PI control can lead to a better response for a small reference speed change. Step speed response with PI control appears to be better when inertia is lower than rated.

The main conclusion of this study is that certain care should be exercised when claiming superiority of the FL control with respect to PI speed control in electric drives. Last but not least, it should be stressed that the whole comparison is valid only for the case when initial controller design aims at achieving aperiodic response to the rated step speed command. The situation is likely to be different if certain amount of overshoot can be tolerated and controllers are initially designed to give an overshoot.

APPENDIX: Motor Data

$R_s = 1.4\ \Omega$	$L_s = 5.6\ \mathrm{mH}$	$\psi_m = 0.1546\ \mathrm{Vs/rad}$
$\omega_n = 180\ \mathrm{rad/s}$	$2P = 6$	$J_n = 0.00176\ \mathrm{kgm}^2$



a.

b.

Fig. 8: Comparison of responses obtained with PI and FL speed controllers: response to rated step speed command (a.) and response to step rated torque application (b.) (inertia 0.3 times the one used in controller design).

$I_n = 6.2$ A $\quad I_{\max} = 21$ A

$T_{en} = 6.1$ Nm $\quad T_{e\max} = 20.7$ Nm

$V_{DC} = 220$ V

REFERENCES

[1] Bose, B.K.: Expert systems, fuzzy logic, and neural network applications in power electronics and motion control. Proceedings of the IEEE, 82, 1994, 1303-1323.

[2] Bose, B.K.: Intelligent control and estimation in power electronics and drives. Proc. IEEE International Electric Machines and Drives Conference, 1997, TA2-2.1-TA2-2.6.

[3] Vas, P., Stronach, A.F. and Neuroth, M.: Design and DSP implementation of fuzzy controllers for servo drives. Electrical Engineering, 79, 1996, 265-276.

[4] Vas, P., Chen, J. and Stronach, A.F.: Fuzzy control of dc drives. Proc. 25th Int. Power Conversion & Intelligent Motion Conf. PCIM, 1994, 11-31.

[5] Feng, X. and Chen, B.: Fuzzy-controlled DC drive system with load observer. Proc. 4th Int. Workshop on Advanced Motion Control AMC'96-MIE, 1996, 354-358.

[6] Sousa, G.C.D. and Bose, B.K.: A fuzzy set theory based control of a phase-controlled converter DC machine drive. IEEE Trans. on Industry Applications, 30, 1994, 34-44.

[7] Ibaliden, A. and Goureau, P.: Fuzzy robust speed control of induction motor. Proc. ICEM, 1996, vol. III, 168-173.

[8] Boussak, M. and Bauer, M.: Robust speed and position control of an indirect field oriented controlled induction motor drive using fuzzy logic regulator. Proc. ICEM, 1996, vol. I, 219-224.

[9] Ficarra, M.C., Eguliaz, J.M.M. and Peracaula, J.: Fuzzy control of an induction motor with compensation of system dead-time. Proc. IEEE PESC, 1996, 677-681.

[10] Fodor, D., Katona, Z. and Vass, J.: On fuzzy logic speed control for vector controlled AC motors. Proc. 4th Int. Workshop on Advanced Motion Control AMC'96-MIE, 1996, 186-191.

[11] Ta-Cao, M. and Le-Huy, H.: Model reference adaptive fuzzy controller and fuzzy estimator for high performance induction motor drives. Proc. IEEE IAS Annual Meeting, 1996, pp. 380-387.

[12] Zhen, L. and Xu, L.: A comparison study of three fuzzy schemes for indirect vector control of induction machine drives. Proc. IEEE IAS Annual Meeting, 1996, 1725-1732,

[13] Heber, B., Xu, L. and Tang, Y.: Fuzzy logic enhanced speed control of an indirect field oriented induction motor drive. Proc. IEEE PESC, 1995, 1288-1294.

[14] Goureau, P. and Ibaliden, A.: Robustness of a fuzzy controller for induction machine. Proc. ELECTRIMACS'96, 551-556.

[15] Donescu, V., Neacsu, D.O., Griva, G. and Profumo, F.: A systematic design method for fuzzy logic speed controller for brushless DC motor drives. Proc. IEEE PESC, 1996, 689-694.

[16] Eminoglu, I. and Atlas, I.H.: A method to form fuzzy control rules for a PMDC motor drive system. Electric Power Systems Research, 39, 1996, 81-87.

[17] Le-Huy, H.: An adaptive fuzzy controller for permanent-magnet AC servo drives. Proc. IEEE IAS Annual Meeting, 1995, 104-110.

[18] Bolognani, S. and Zigliotto, M.: Fuzzy logic control of a switched reluctance motor drive. IEEE Trans. on Industry Applications, 32, 1996, 1063-1068.

[19] Zheng, L.: A practical guide to tune of proportional and integral (PI) like fuzzy controllers. Proc. IEEE Int. Conf. on Fuzzy Systems, 1992, pp. 633-640.

Addresses of the authors

Z.Ibrahim, E.Levi, D.Williams

Liverpool John Moores University
School of Engineering
Byrom St
Liverpool L3 3AF
England, UK

33rd Universities
Power Engineering Conference

UPEC ‘98
Conference Proceedings

Napier University, Edinburgh, UK
8 – 10 September 1998

Volume 2

NAPIER UNIVERSITY
EDINBURGH

A SELF-TUNING METHOD FOR FUZZY LOGIC SPEED CONTROLLER IN HIGH PERFORMANCE AC DRIVES

Z.Ibrahim, E.Levi and D.Williams

Liverpool John Moores University, UK

ABSTRACT

It is well recognised that scaling factors of a speed fuzzy logic controller (FLC) significantly influence operation of a high performance ac drive. In general, fixed scaling factors do not provide satisfactory response over wide range of operating conditions. Self-tuning of scaling factors is therefore often discussed as a mean for achieving desired response when parameters of the plant and/or reference settings vary. The paper presents a self-tuning mechanism for input and output scaling factors of a speed FLC. Self-tuning is achieved by means of another FLC, whose outputs are scaling factors for the speed controller. Design of the fuzzy logic controllers is described and their performance is investigated for the case of a vector controlled permanent magnet synchronous machine (PMSM). Simulation results, that verify appropriateness of the approach, are included.

INTRODUCTION

A possibility of replacing a standard PI speed controller with a fuzzy logic speed controller in high performance drives is a topic that has recently attracted significant attention in the research community world-wide [1,2]. An FLC is essentially a multi-parameter controller, whose performance depends on the selected shape of membership functions, rule base and scaling factors [3,4]. These parameters are usually determined off-line, on the basis of simulations for specific operating conditions, using expert knowledge and trial-and-error method. However, when such a fixed-parameter FL speed controller is applied in operating conditions that differ from those used in the design stage, performance of the drive deteriorates [5]. In particular, response of the controller is affected by the variation of the total inertia of the drive and by reference speed set point and reference speed profile (for example, responses to step change of set point and reference speed tracking performance are quite different, [6]).

An adaptive fuzzy logic controller, that will enable satisfactory operation under varying operating conditions, can be designed in a number of different ways. Frequently discussed methods encompass self-organising FLC [7-9], whose rule base is modified on-line, and self-tuning FLC [5,10-15], whose scaling factors are modified on-line. On-line modification of scaling factors appears to be the favoured approach and is the one adopted in this study. Different methods of scaling factor self-tuning have already been proposed. It is possible to continuously tune output scaling factor only, while keeping input scaling factors at constant value [5,13]. Alternatively, one may decide to tune input error scaling factor and output scaling factor [12] or all the three scaling factors [15] using only two sets of values for scaling factors: one set for transient and the other set in vicinity of steady-state [12,15]. In majority of these studies operation for a single speed set point is considered. The method described in this paper performs on-line tuning of all the three scaling factors as function of the reference speed setting, by means of another fuzzy logic controller. Fuzzy logic speed controller is required to provide speed response without any overshoot with minimum settling time for all the reference speed set points. As shown in the paper, this goal is achievable by the developed auxiliary fuzzy logic controller whose outputs are scaling factors for the speed fuzzy logic controller.

DESCRIPTION OF THE SYSTEM

A permanent magnet synchronous motor drive with rotor flux oriented control, used in this study, is illustrated in Fig. 1. All the relevant data of the machine are given in Appendix. The PWM voltage source inverter is controlled by means of three independent hysteresis current controllers. The hysteresis band is adjusted to ± 0.5 A (i.e., ± 5.7% of the rated current) and is constant in all the simulations. The inverter input dc voltage is set to the constant value of 220 V, so that sufficient voltage reserve is provided for operation with good current control at rated speed and maximum torque. Stator q-axis current is limited in accordance with the maximum allowed stator current rms value.

The FL speed controller, Fig. 2, is of standard structure with inputs of speed error and change of speed error. Symbol T_s stands for sampling time. Scaling factors for speed error, change of speed error and output scaling factor (G_e, G_{ce}, and G_u) are defined in Fig. 2. As an FLC is essentially a PD controller equivalent, output of the speed FL controller is integrated in order to yield PI like behaviour. An equivalent anti wind-up feature is added, and stator q-axis current command is limited to the maximum permissible value.

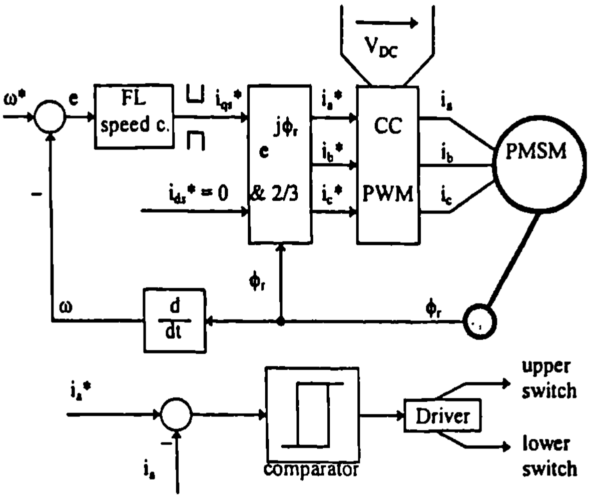


Figure 1 Rotor flux oriented permanent magnet synchronous motor (PMSM) drive with hysteresis current control

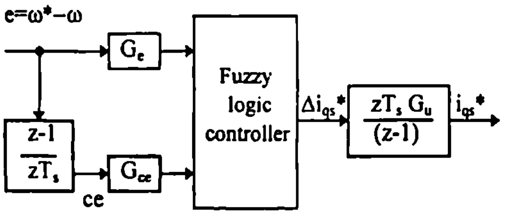


Figure 2 Fuzzy logic speed controller

The FL controller consists of triangular membership functions with overlap, seven for speed error and seven for the change of speed error, so that a 7x7 rule base is created. The width of the membership functions is reduced in vicinity of zero error, zero change of error and zero stator q-axis current command in order to obtain better steady-state accuracy.

DESIGN OF THE FUZZY LOGIC SPEED CONTROLLER

Initial Design of the Speed FLC

At first, a constant parameter FLC (CPFLC) is designed for aperiodic response to step speed command equal to rated. Scaling factors are calculated using known motor data. Rated speed of the motor is 180 rad/s electrical and operation in forward direction of rotation is required only. Thus, assuming that this is the maximum speed of operation of the motor, maximum speed error is 180 for start-up from standstill and the scaling factor for the speed error is obtained as

$$G_e = 1/180 \approx 0.005 \tag{1}$$

Scaling factor for the change of error is calculated on the basis of the rated inertia and the maximum torque that the motor is allowed to develop (both values are given in the Appendix), taking sampling time as 20 μs (*e* and *ce* stand for speed error and change of error):

$$\begin{gathered} T_{emax} = J_n / P(\Delta\omega / T_s) \quad \Rightarrow \quad \Delta\omega = 0.706 \text{ rad/s} \\ G_{ce} = 1/\text{ce} = 1/(e(T_s) - e(0)) = 1/\Delta\omega = 1.4 \end{gathered} \tag{2}$$

Output scaling factor is set to $G_u = 1$.

Off-line Optimised Speed FLC

Speed response, obtainable with the scaling factors of sub-section 3.1, is not satisfactory. All the three scaling factors have therefore been optimised, by performing large number of simulation runs, in such a way that aperiodic speed response to rated speed command is obtained with minimum settling time. The second CPFLC is created in this way, with scaling factors

$$G_e = 0.0023 \qquad G_{ce} = 0.41 \qquad G_u = 3 \tag{3}$$

Excellent speed response to the rated speed command is obtainable with scaling factors of (3), as shown in the next section. However, as the initial step in speed reference is reduced, speed response deteriorates and becomes highly oscillatory at very low speed reference settings. An adaptive FLC is therefore developed next, whose scaling factors are adjusted in accordance with the speed reference, by means of an auxiliary FLC.

Adaptive Speed FLC

Adaptive speed FLC, whose scaling factors are tuned on-line, is shown in Fig. 3. Tuning of scaling factors is performed by means of the second, auxiliary FLC.

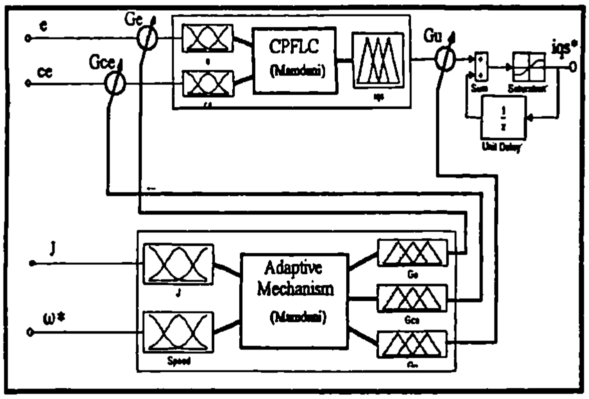


Figure 3 Tuning of speed FLC scaling factors with an auxiliary FLC

Outputs of the auxiliary FLC, used for scaling factor adaptation, are scaling factors for the speed controller. Its inputs are inertia of the drive and the reference speed setting. The ultimate goal is therefore to adapt scaling factors for both variable reference speed setting and inertia variation. The aspect of inertia variation, its on-line identification and its compensation by adaptation of scaling factors is however beyond the scope of interest in this paper. Only adaptation of scaling factors to reference speed settings is considered and inertia at all times equals rated. Rule base of the auxiliary FLC, illustrated in Table I, however incorporates inertia. Rated inertia, according to Table I, is encompassed by PS membership function. There are four membership functions for both inertia and speed reference setting. Rule base therefore consists of 16 rules.

Table I Rule base for error scaling factor, change of error scaling factor and output scaling factor

	MOTOR INERTIA			
	ZE	**PS**	**PM**	**PL**
REF. SPEED	0.3J<J<Jo	0.8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	PM	PM	PL	PL
10< ω*<80	PM	PM	PM	PM
40< ω*<120	PS	PS	PS	PS
80< ω*<180	ZE	ZE	ZE	ZE

	MOTOR INERTIA			
	ZE	**PS**	**PM**	**PL**
REF. SPEED	0.3J<J<Jo	0.8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	ZE	PS	PM	PL
10< ω*<80	ZE	PS	PM	PL
40< ω*<120	PS	PS	PM	PL
80< ω*<180	PS	PM	PL	PL

	MOTOR INERTIA			
	ZE	**PS**	**PM**	**PL**
REF. SPEED	0.3J<J<Jo	0.8Jo<J<2Jo	Jo<J<6Jo	2Jo<J<10Jo
0<ω*<20	ZE	PS	PM	PM
10< ω*<80	ZE	PS	PM	PM
40< ω*<120	ZE	PS	PM	PM
80< ω*<180	ZE	PS	PM	PM

SIMULATION RESULTS

Speed response of the drive to step speed command is illustrated in Fig. 4 for four different reference speed settings: ω* = 180, 120, 60 and 30 rad/s. In each case three traces are shown, that correspond to the three previously discussed designs of the speed FLC.

Figure 4 shows that off-line optimised speed CPFLC gives the best response to rated speed command (the case for which manual optimal adjustment was performed). Speed response of adaptive FLC is slightly worse, while initial design of the CPFLC yields inferior speed response. Similar situation is observed for speed command of 120 rad/s; however, optimised CPFLC now gives a very small, but nevertheless undesirable, overshoot, which does not take place with adaptive FLC. As speed reference is further lowered, speed response of the off-line optimised CPFLC further deteriorates: overshoot increases and becomes over 6 rad/s for 30 rad/s speed command. Furthermore, settling time increases as well, as speed response becomes more and more oscillatory. Behaviour of the CPFLC is the worst for all the speed commands and this type of CPFLC is therefore not considered further on. As far as adaptive FLC is concerned, it yields desired aperiodic response at very low speed command (30 rad/s) and almost aperiodic response at 60 rad/s speed command (overshoot is less than 2 rad/s). Figure 4 thus verifies the superiority of the adaptive FLC when compared to both CPFLCs: it is capable of realising an aperiodic (or almost aperiodic) speed response irrespective of the reference speed setting, with very short settling time.

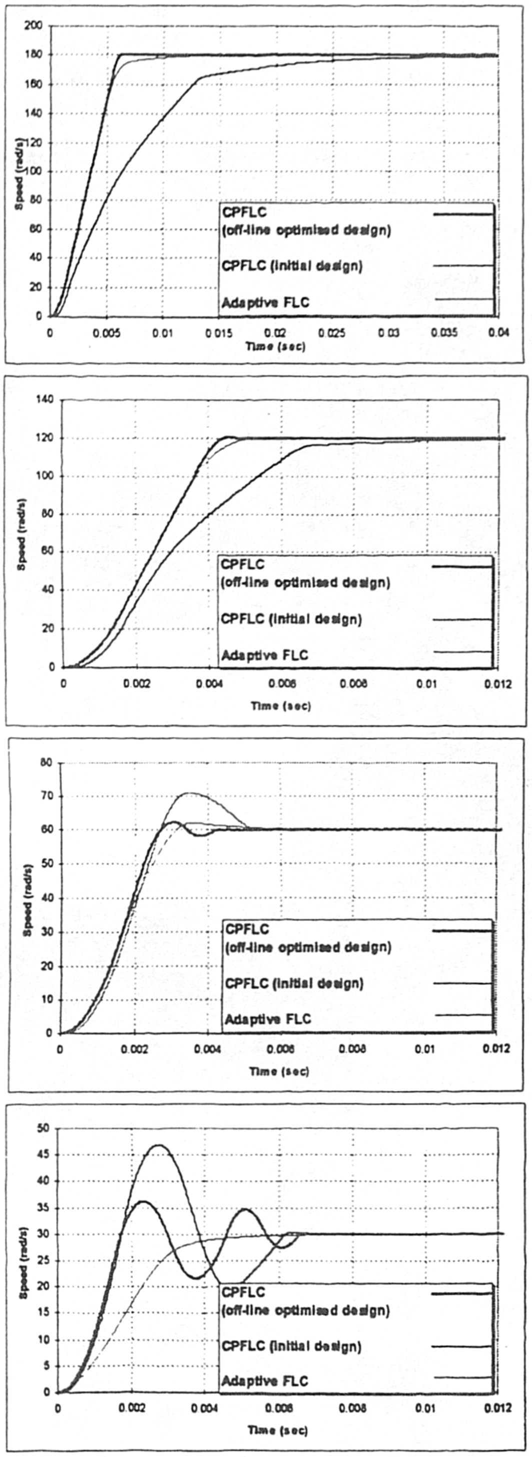


Figure 4: Speed response of the drive for four reference speeds (180, 120, 60 and 30 rad/s, respectively), obtained with three speed FLCs of Section 3

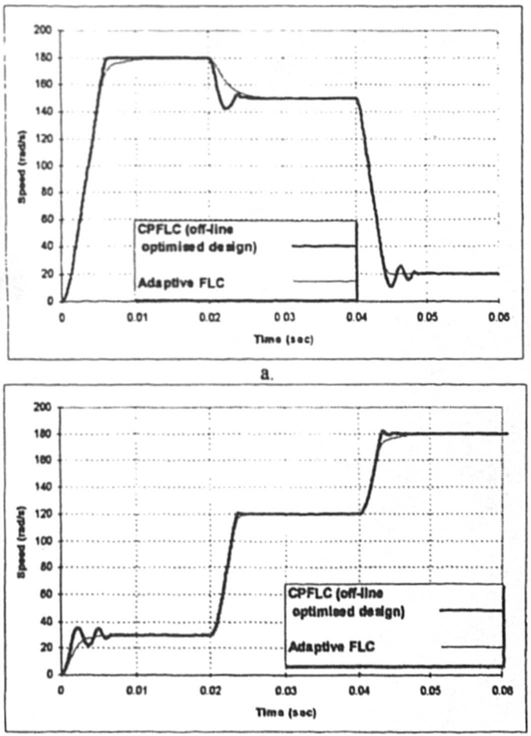
Cases considered in Fig. 4 all apply to stepping of the speed reference from zero up to a certain value. In order to verify the ability of the adaptive speed FLC to maintain required aperiodic speed response when speed reference changes in a step-wise manner from an initial speed different from zero, the following two sequences of reference speed variation are investigated: a) zero to 180 rad/s, 180 to 150 rad/s and 150 to 20 rad/s (Fig. 5a), and b) zero to 30 rad/s, 30 to 120 rad/s and 120 to 180 rad/s (Fig. 5b). Results obtained with adaptive speed FLC are compared with those obtained with off-line optimised CPFLC. Figure 5 clearly indicates superiority of the adaptive FLC over the CPFLC and shows that adaptive FLC gives desired aperiodic speed response for all the considered transients.

CONCLUSION

A fuzzy logic based on-line tuning of scaling factors for the fuzzy logic speed controller in high performance drives is described in the paper. It is shown that such an adaptive FLC can provide desired response to any speed reference setting and that the speed response is superior to the one obtainable with off-line optimised constant parameter fuzzy logic speed controller. Tuning of scaling factors by means of the proposed fuzzy reasoning rule base is relatively easy to implement and execution time is sufficiently small for on-line applications.

REFERENCES

[1] B.K. Bose; Expert systems, fuzzy logic, and neural network applications in power electronics and motion control, *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1303-1323, 1994.



a.

b.

Figure 5 Speed response for step change in reference speed setting: a) 0 to 180, 180 to 150 and 150 to 20 rad/s; b) 0 to 30, 30 to 120 and 120 to 180 rad/s

[2] B.K. Bose; Intelligent control and estimation in power electronics and drives, *Proc. IEEE Int. EMD Conf.*, Milwaukee, WI, pp. TA2-2.1-TA2-2.6, 1997.

[3] P.Vas, A.F.Stronach, M.Neuroth; Design and DSP implementation of fuzzy controllers for servo drives, *Electrical Engineering*, vol. 79, pp. 265-276, 1996.

[4] P.Vas et al.; Fuzzy control of dc drives, *Proc. 25th PCIM*, Nuremberg, Germany, pp. 11-31, 1994.

[5] I.Eminoglu, I.H.Atlas; A method to form fuzzy control rules for a PMDC motor drive system, *Electric Power Systems Research*, vol. 39, pp. 81-87, 1996.

[6] B.Heber et al.; Fuzzy logic enhanced speed control of an indirect field-oriented induction machine drive, *IEEE Trans. Power Elec.*, vol. 12, no. 5, pp. 772-778, 1997.

[7] F.Ashrafzadeh, E.P.Nowicki, J.C.Salmon; Self-organising and self-tuning fuzzy logic controller foe field oriented control of induction motor drives, *Proc. IEEE IAS*, Orlando, FL, pp. 1656-1662, 1995.

[8] S.Bolognani, M.Tursini, D.Q.Zhang, M.Zigliotto; Application of a fuzzy self organising controller to a motor drive, *Proc. Symp. SPEEDAM*, Taormina, Italy, pp. 299-304, 1994.

[9] S.H.Ghwanmeh; The investigation of on-line self-learning fuzzy logic control for control of non-linear process, *PhD Thesis*, JMU, Liverpool, UK, 1996.

[10] P.Vas, W.Drury, A.F.Stronach; Recent developments in artificial intelligence based drives - a review, *Proc. 29th PCIM*, Nuremberg, Germany, pp. 59-70, 1996.

[11] F.Barrero, E.Galvan, A.Torralba, L.G.Franquelo; Fuzzy self-tuning system for induction motor controllers, *Proc. EPE'95*, Seville, Spain, pp. 1.283-1.285, 1995.

[12] X.Feng, B.Chen; Fuzzy-controlled DC drive system with load observer, *Proc. 4th AMC'96-MIE*, Mie University, Tsu-City, Japan, pp. 354-358, 1996.

[13] G.He, J.P.Jiang, J.Shen, J.Lei; A fuzzy variable structure control scheme for AC servo drive system, *Proc. IEEE Int. EMD Conf.*, Milwaukee, WI, pp. MC3-3.1-MC3-3.3, 1997.

[14] L.Zhen, L.Xu; A Comparison study of three fuzzy schemes for indirect vector control of induction motor drives, *Proc. IEEE IAS*, San Diego, CA, pp. 1725-1732, 1996.

[15] F.Parasiliti, M.Tursini, D.Q.Zhang; Adaptive fuzzy logic control for high performance PM synchronous drives, *Proc. IEEE MELECON'96*, Bari, Italy, pp. 323-327, 1996.

APPENDIX: Motor Data

$R_s = 1.4\ \Omega$	$L_s = 5.6$ mH	$\psi_m = 0.1546$ Vs/rad
$\omega_n = 180$ rad/s	$2P = 6$	$J_n = 0.00176$ kgm^2
$I_n = 6.2$ A	$I_{max} = 21$ A	
$T_{en} = 6.1$ Nm	$T_{emax} = 20.7$ Nm	
$V_{DC} = 220$ V		

AUTHOR'S ADDRESS

The first author can be contacted at

School of Engineering, John Moores University
Byrom St
Liverpool L3 3AF, UK