INTELLIGENT SPEED CONTROL OF INTERIOR PERMANENT MAGNET MOTOR DRIVES

CASEY BENJAMIN BUTT

Your file Votre référence
ISBN: 978-0-494-33470-6
Our file Notre référence
ISBN: 978-0-494-33470-6

Intelligent Speed Control of Interior Permanent Magnet Motor Drives

by

Casey Benjamin Butt

A thesis submitted to the school of graduate studies
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

July 2007

St. John's Newfoundland Canada

Abstract

The advantages of the interior permanent magnet synchronous motor (IPMSM) for modern high performance drive (HPD) applications are well established. The IPMSM has become increasingly popular in these applications due to its high torque to current ratio, high power to weight ratio, high efficiency, high power factor, low noise and robust operation. In HPDs, it is required that automated control systems be implemented that can respond quickly and accurately to sudden and unknown disturbances, recovering the correct working speed of the motor promptly. Conventional proportional-integral (PI) and proportional-integral-derivative (PID) speed controllers have been widely utilized to meet these control challenges in IPMSM drives. However, difficulties in obtaining the exact d-q axis reactance parameters of the IPMSM make design approaches for these controllers cumbersome. Moreover, since the operation of the IPMSM is strongly affected by the rotor magnetic saliency, saturation and armature reaction effects, conventional fixed-gain PI and PID controllers are very sensitive to parameter variations, changes of command speed and load disturbances. Therefore, an intelligent speed controller is required, which overcomes these problems, for the IPMSM drive to be used in HPD systems.

This thesis presents intelligent speed controllers for the IPMSM based on multiple and single artificial neuron networks. Traditional artificial neural network-based motor controllers require extensive offline training, which is both time consuming and requires complex knowledge of motor behaviour for the specific drive system. In addition, drive behaviour is unpredictable when operating parameters outside the training set are encountered. The proposed drive systems overcome these limitations by utilizing a novel method of determining a reference command torque that is used to train the artificial neural network-based controller. Thus, no offline training of the networks is required, they are robust under varying operating conditions and they are adaptable to various drive systems.

Control of the IPMSM at rated speed and below is achieved by use of the maximum torque per ampere (MTPA) mode of operation utilizing a series approximated approach. Control above base speed is achieved based on an approximation of the flux-weakening (FW) mode of operation.

Comparisons are made, in simulation and experimentally, between the proposed drives and a conventional PID-based control technique utilizing the $i_d = 0$ approximation.

The complete vector control scheme is implemented in real-time using a digital signal processor (DSP) controller board and a laboratory 1 hp interior permanent magnet synchronous motor.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor Professor M. A. Rahman for his guidance, encouragement and advice throughout the preparation of this thesis. His influence has helped me learn the practicalities of modern research and scientific writing. I would also like to thank Dr. Glyn George and Dr. Tariq Iqbal for serving as my supervisory committee and helping with the revisions to this thesis – both have encouraged and helped me since my undergraduate years.

I would like to sincerely thank the Natural Sciences and Engineering Research Council of Canada (NSERC), the School of Graduate Studies, the Faculty of Engineering and Applied Science and Dr. Rahman for their financial support, without which the pursuit of this research would not have been possible. In addition, I would like to acknowledge Mr. Richard Newman for his expertise and assistance with the implementation of the lab equipment – there's simply no replacement for practical experience.

Finally, I would like to thank my wife Nadya and my parents Mr. Benjamin Butt and Mrs. Mary Butt for their support and encouragement. Without their support I doubt it would have been possible for me to complete this study.

Contents

2. Analysis and Modeling of the PWM VSI-Fed IPMSM Drive

3. Artificial Neural Network Based Speed Controller

6. Experimental Implementation of the SAN-based Vector Control of the IPMSM

7. Conclusions

List of Tables

List of Figures

List of Symbols

v_a, v_b, and v_c	a, b and c phase voltages, respectively
i_a, i_b, and i_c	actual a, b and c phase currents, respectively
i_a^*, i_b^*, and i_c^*	command a, b and c phase currents, respectively
v_d	d-axis voltage
v_q	q-axis voltage
i_d	stationary d-axis current
i_q	stationary q-axis current
i_d^r	rotating d-axis current
i_q^r	rotating q-axis current
i_{dm}^r	rotating d-axis magnetizing current
i_{qm}^r	rotating q-axis magnetizing current
L_{md}	d-axis magnetizing inductance
L_{mq}	q-axis magnetizing inductance
ω_s	stator angular frequency
ω_r	actual rotor speed
ω_r^*	motor command speed
e	error between actual and command speeds (ω_r^* - ω_r)
Δe	change in speed error
θ_r	rotor position
$\theta_r(0)$	initial rotor position

p	differential operator (d/dt)
P	number of pole pairs
T_e	developed electromagnetic torque
T_e^*	command electromagnetic torque
T_{ref}^*	reference command torque
T_{ANN}^*	neural network command torque
T_L	load torque
J_m	rotor inertia constant
B_m	friction damping coefficient
Ψ_m	magnetic flux linkage
V_B	dc bus voltage of inverter

Chapter 1

Introduction and Literature Review

1.1 General Introduction to Motor Drives

Conventional direct current (dc) electric machines and alternating current (ac) induction and synchronous electric machines have traditionally been the three cornerstones serving daily electric motor needs from small household appliances to large industrial plants. Recent technological advances in computing power and motor drive systems have allowed an even further increase in application demands on electric motors. The brushless DC (BLDC) machine, the switched reluctance machine (SRM), the permanent magnet hysteresis machine and the permanent magnet synchronous machine have all been developed, or further developed, in the past two decades [1-6]. This increasing development in motor technology and application demands of both newer and traditional machines has been accompanied by the need for sophisticated control schemes to maximize the performance and minimize the operating costs of electric motors in modern applications.

1.2 Permanent Magnet Synchronous Motors

1.2.1 General Introduction

The use of rotor windings on conventional synchronous motors requires additional power supply and maintenance-requiring slip-rings and brushes to supply the dc field excitation. The traditional rotor winding set-up of these machines, therefore, increases operating costs and reduces efficiency. This shortcoming has prompted researchers to develop different types of synchronous motors with the intent of eliminating slip-ring and brush related losses. One such motor is the permanent magnet (PM) synchronous motor.

PM synchronous motors have desirable features such as a high torque to current ratio, a high power to weight ratio, high efficiency, low noise and good overload capacity. Unlike in the wire-wound synchronous motor, the excitation field is provided by permanent magnets, thus eliminating the rotor field windings. This not only reduces the initial cost of the PM synchronous motor but also eliminates the additional power demands due to those windings and the costs necessary to maintain them. Furthermore, permanent magnet rotors allow for more compact motor designs for given output capacities.

As compared to the induction motor, the PM synchronous motor experiences no slip, as with all synchronous motors. Therefore, there are no slip dependent rotor losses, giving the PM synchronous motor a higher torque to inertia ratio and power density. For similar output ratings, the PM synchronous motor is smaller and lighter than the induction motor, making it preferable in high performance applications where size and weight are of concern.

Restrictions upon PM synchronous motors at this time are in the form of power and torque limitations due to the current limits of permanent magnet technology. Clearly, as magnetic materials with higher flux density to weight/size ratios are developed, and incorporated into PM synchronous motor designs, performance will improve further. The PM synchronous motor also presents some rather unique challenges from a control perspective due to its inherently varying q- and d-axis reactances during operation and back emf generation during deceleration.

1.2.2 Classification of PM motors

On the most fundamental level, PM motors fall into two categories: PM dc motors and PM synchronous ac motors. PM dc motors are separately excited dc motors with permanent magnets as the excitation source. In industry, they are widely used as control motors, often in precision applications such as computer disk drives. PM ac motors are, operationally, synchronous motors. Therefore, from this point in this thesis forward, PM ac synchronous motors will be referred to simply as PM ac motors. Intelligent speed control of these motors will be the primary focus of this thesis.

PM ac motors are categorized broadly based on the orientation of the permanent magnet magnetic fields. There are two such fundamental classifications: (a) radially oriented type, in which the rotor magnets are oriented such that the direction of magnetization is radial from the rotor, and (b) circumferential type, in which the rotor magnets are oriented such that the direction of magnetization is circumferential around the rotor.

In radially oriented machines the air gap flux density above the permanent magnets is approximately the same as the magnetic flux density. This means that for practical radially oriented PM motors to exist high-energy to weight/size ratio magnetic materials such as neodymium-boron-iron (Nd-B-Fe) and samarium-cobalt (Sm-Co) must be used. The use of low residual flux density magnetic materials such as ferrite magnets produces machines with very low air gap flux densities and, therefore, low output capacities. In this manner, radially oriented permanent magnet motor development has been directly dependent on recent advances in high-energy permanent magnet material technology.

Circumferentially oriented permanent magnet motors have a direction of magnetization that is circumferential in the rotor. Typically, their design incorporates a large numbers of poles, thus allowing for reasonable output ratings to be produced while utilizing traditional low residual flux density magnetic materials.

Based on the location of the permanent magnets in the rotor itself, there are three configurations of permanent magnet motors: (a) surface mounted type, where the magnets are mounted on the surface of the rotor; (b) inset type, where the magnets are fully or partially inset into the rotor core and (c) interior type, where the permanent magnets are buried within the rotor core [7]. It should be noted that surface mounted PM motors, by design, must be of the radially oriented type, whereas inset and interior type PM motors can have magnetization orientations either radial or circumferential in nature.

PM motor classification can also be based on control strategy, which produces two more classifications of PM motors [8]: (a) the brushless dc (BLDC) motor, which is an electronically commutated rectangular wave-fed synchronous motor with surface mounted permanent magnets; and (b) the conventional PM synchronous motor (PMSM), which is a sinusoidal wave-fed PM motor. Surface mounted type PM motors, suitable for use as BLDC motors, have large air gaps and, therefore, have weakened armature reaction effects. This restricts this type of motor to low speed and constant torque operations. These motors are commonly used as precision control motors and hard disk drive motors in computers. Inset and interior type PM motors have smaller, more uniform air gaps, allowing for operation at higher speeds in the constant power region. Interior type PM motors, in particular, have significant motor torque contributed by the reluctance component due to the large difference between the direct- and quadrature- axis reactances, as well as the permanent magnet field component. This makes the interior PM motor ideal for use at high speeds utilizing the flux-weakening method of control, as well as for low speed, constant torque operations. In addition, this type of PM motor is the most economical to manufacture. These motors are commonly controlled with three-phase sinusoidal EMF, making them fall into the second PM classification based on control method: conventional PMSM.

Depending on the rotor cage winding, the PM motor may be further classified as: (a) cage type or (b) cageless, based on whether the rotor is fitted with a cage winding or not. In the case of the cage type motor, the rotor cage winding provides the starting torque via induction and, hence, this type of motor can be line started with rated supply voltage and frequency. Cageless motors have no cage windings and, therefore, control strategies using variable stator frequencies must be used to start the motor from standstill and accelerate it smoothly to synchronous speed.

One further classification of the PMSM is based on whether or not the motor is equipped with a rotor position sensor. PMSMs without a rotor position sensor are classified, logically, as "sensorless" and PMSMs equipped with a rotor position sensor are known as "with sensor".

1.3 PMSM Drives

1.3.1 Vector Control Schemes

As with other types of synchronous motors, early controllers for the PMSM were based on the conventional open-loop volt/hertz (v/f) control method. Open-loop systems, by design, are incapable of compensating reactively in response to dynamic changes in operating parameters. Due to this limitation, closed-loop schemes with torque and angle control have been developed for applications where better drive performance is required [9]. These traditional scalar control techniques, however, have shortcomings due to the nonlinearity of the PMSM mathematical model and inherent coupling between direct and quadrature axis quantities. This leads to inaccuracies and slow responsiveness, which are unacceptable in high performance drive applications. To solve this problem, vector, or field-oriented, control techniques have been accepted almost universally for control of ac drives. The vector control technique, employing a current-controlled voltage source inverter (VSI), provides a method of variable speed control for the PMSM that has fast responsive and follows command speeds accurately and precisely.

In the vector control technique for PMSM drives, the phase angle and the magnitude of the phase currents are controlled to provide high precision control of the motor. This is an evolution of control techniques developed in the 1970s by Blasche [10] and Hasse [11] for ac drives. At that time, however, implementation of these schemes was difficult due to technological limitations. Today, with very large scale integrated (VLSI) technologies, as well as advancing power electronic and microprocessor technologies, the practical implementation of the vector control scheme is no longer a problem.

The operating principle of vector control is based on elimination of the coupling between the direct (d) and quadrature (q) axes. This can be achieved by coordinate transformation, producing control very similar in nature to that of a separately excited dc motor. However, unlike dc machine control, in an ac machine both the magnitude and phase angle of the stator current need to be controlled. This is achieved by employing a time-varying vector that corresponds to a sinusoidal flux wave moving in the air-gap of the machine, hence the name, "vector" control.

By referring the mmf wave of the stator current to the vector corresponding to the flux wave it becomes apparent that the quadrature-axis (q-axis) component of the stator current mmf wave contributes most significantly to the developed torque. The direct-axis (d-axis) current is seen to contribute, predominantly, to the magnitude of the flux. This makes it convenient to define the stator current in a frame of reference defined by the time-varying field, thus illustrating the close correspondence with dc machines. Such a comparison shows that the d-axis component of stator current in a PMSM is analogous to the field current and the q-axis component of stator current is analogous to the armature current in the dc motor. Figure 1.1 shows a cutaway of a 4-pole IPMSM with synchronously rotating d- and q-axes identified.

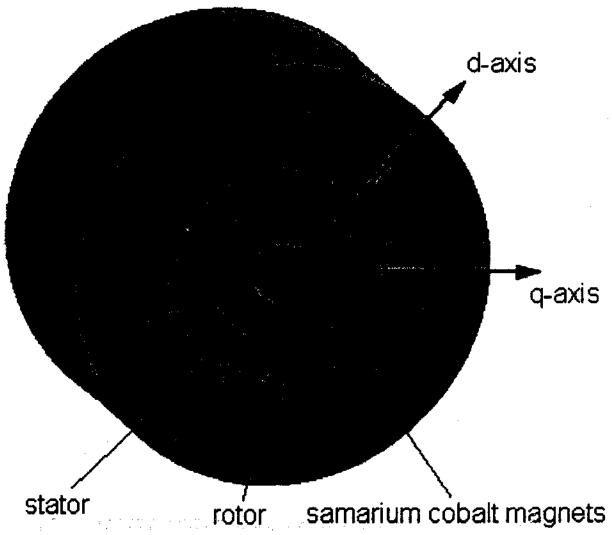


Figure 1.1 Cutaway of 4-pole IPMSM.

While vector control can be implemented in a reference frame fixed to the stator, rotor, or magnetizing flux space vector, with d- and q-axis stator currents defined in those frames, rotor flux oriented control is most commonly used in PMSM drives. This is because in the stator and magnetizing flux oriented control cases there exists a coupling between the torque producing stator current and the stator magnetizing current, whereas with rotor flux oriented control matters are simplified by a natural decoupling that occurs between the d- and q-axis components.

Vector control can be classified as either direct or indirect. Direct vector control, shown in Figure 1.2, depends on the direct measurement of the stator (or rotor) flux using flux sensors. The d- and q- axis flux components, ψ_{dm} and ψ_{qm},

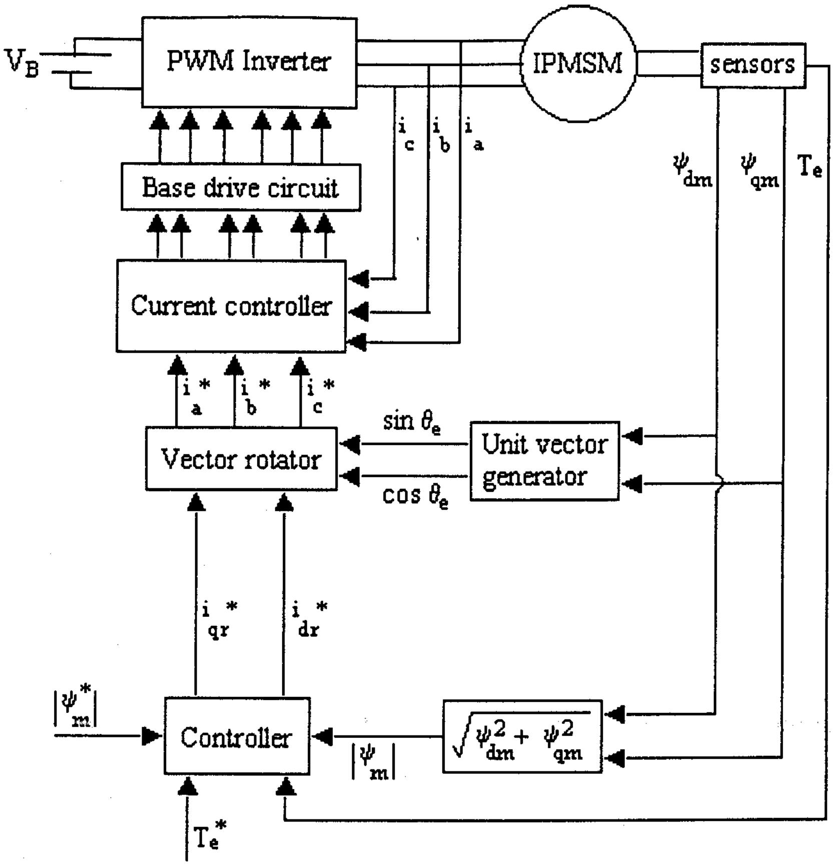


Figure 1.2 Direct vector control scheme of PMSM.

along with the appropriately calculated command torque and flux and actual torque and flux, are then used to generate the principal control parameters, the rotor flux frame d- and q-axis command currents, i_{dr}^* and i_{qr}^*. These dc currents, proportional to command torque and flux respectively, are then converted to a stationary reference frame and used to generate phase current commands for the VSI.

The indirect method of vector control, as shown in Figure 1.3, uses sensors to determine rotor position and stator currents. From this, and along with the command speed, the rotor flux frame d- and q-axis command currents, i_{dr}^* and i_{qr}^*,

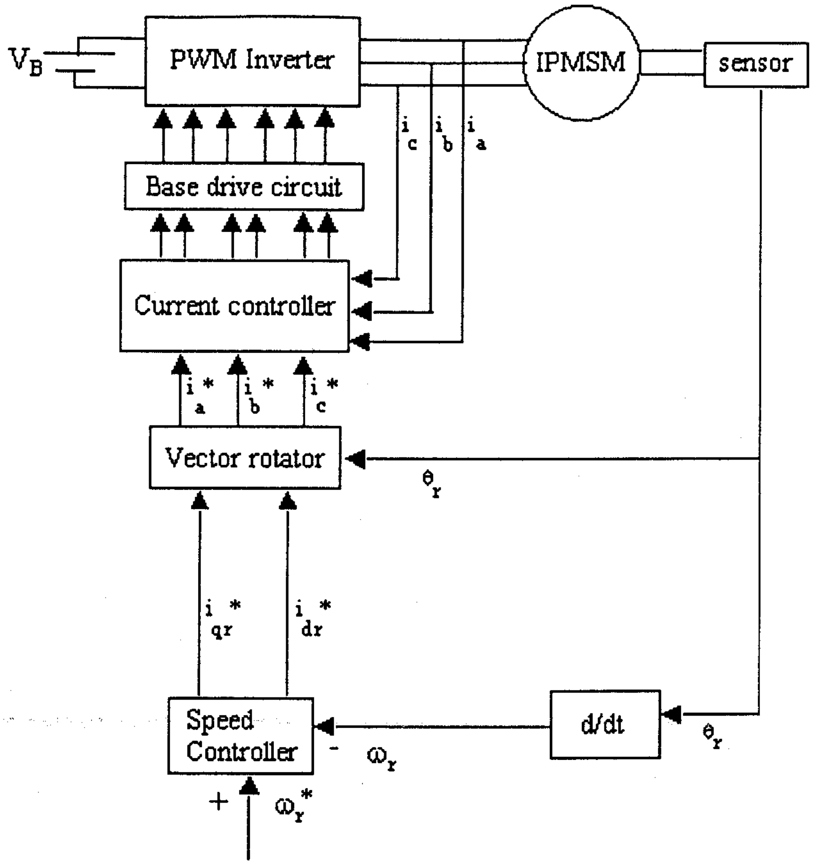


Figure 1.3 Indirect vector control scheme of PMSM.

are calculated. As with the direct method, these currents, after transformation to a stationary reference frame, are then used to generate phase current commands for the VSI. The indirect method of vector control is more sensitive to parameter variations than the direct method, and so motor parameters must be known accurately.

In both of these schemes, a current controlled VSI is used to apply the calculated command currents to the motor stator. This imposes the need for precise and accurate control of the VSI, as the current controller has direct influence on the drive performance. Low-loss current controllers that produce minimal harmonics and noise in the motor, as well as fast responsiveness for high performance under dynamic conditions, are required. The performance of various current control schemes for VSI-fed ac and PMSM drives have been investigated [12-19]. Each scheme has been shown to have its own unique drawbacks with regards to accuracy and dynamic response over varying speed ranges. As the purpose of my research is to investigate improvements in control of the motor drive itself and not the VSI specifically, this work employs a simple fixed-band hysteresis current control algorithm for the VSI. This scheme provides fast response and good accuracy in producing a stator current which tracks the command current within a hysteresis band, while avoiding unnecessary complexity. The drawback is that harmonic current is increased and a non-uniform PWM switching pattern is produced, resulting in a switching frequency that varies over the fundamental period [15].

1.3.2 PMSM Drive Controllers

Modern VLSI and advancing power electronic technologies have lead to the utilization of microprocessors in the control of the PMSM. This allows the current advancements in computing power to be applied to PMSM drive control, resulting in the implementation of complex control strategies. Bose and Szczesny originally proposed a microcomputer-based control system of an interior permanent magnet synchronous motor (IPMSM) in 1988 [20].

Because of the versatility of the IPMSM, it has become essential to develop robust controllers for use in high performance drive (HPD) systems. Various uncertainties like sudden changes of command speed, abrupt load changes and parameter variations have to be handled quickly and accurately in such applications. With this in mind, several types of controllers, including conventional fixed gain types such as proportional-integral (PI), proportional-integral-derivative (PID) and pseudo-derivative-feedback (PDF); adaptive types such as model reference adaptive controllers (MRAC), sliding mode controllers (SMC) and variable structure controllers (VSC); and intelligent types such as fuzzy logic controllers (FLC), artificial neural network (ANN) controllers and neuro-fuzzy controllers have been used for moderate to high performance drive systems. As this work utilizes the indirect control of an IPMSM through a current-controlled VSI, this is where the literature review for the present work will focus.

To date, many researchers have reported their work on the development of high performance IPMSM drives, with the majority of the control schemes reported involving the performance of the IPMSM fed by a voltage source or current source inverter. The reader is referred to the indirect vector control scheme shown in Figure 1.3.

Gummaste and Slemon [21] have proposed a vector control scheme to analyze the steady-state performance of a VSI-fed PMSM drive. They also presented a similar work utilizing the current source inverter-fed PMSM drive [22]. In both schemes they used position feedback control from a shaft position sensor, mounted on the motor shaft, relaying rotor position back to the microcomputer so that the inverter could be operated in a self-controlling mode in real-time. The constant torque mode and the constant power mode operations have been investigated, with strategies having been developed for torque control. They have suggested removing the damper winding for the VSI-fed PMSM drive in order to increase operating stability, as the damper winding provides a path for the flow of harmonic currents induced for non-sinusoidal voltage outputs of the inverter.

However, for the current source inverter (CSI) -fed PMSM drive system, the damper windings are helpful in reducing the commutating inductance and, therefore, their removal is not recommended.

An analysis of a microprocessor-based implementation of the PMSM drive has been presented by Liu et. al., [23]. In their work, the motor is fed by the hysteresis current-controlled VSI. To overcome some of the limitations of the hysteresis controller at low speeds, they have proposed a method utilizing the freewheeling period. The downfall of this proposal, however, is that this method reduces the average torque delivered by the motor with only modest performance improvements at low speeds.

Pillay and Krishnan [24-27] have presented a number of papers on modeling, simulation, analysis and controller design for high performance vector-controlled PMSM drives using a state-space model. They have investigated the transient and steady-state performance of the drive using a d-q axis model of the PMSM, and have also investigated the performances of the hysteresis and ramp comparator current controllers for the VSI-fed PMSM drive. In these contributions, a PID type speed controller, based on the linear model of the PMSM, has been used. In real-time, however, the PMSM torque does not behave linearly, [28] thus the linear model approximation introduces error. Hence, it is very difficult to accurately predict the performance of the drive accurately using this approximation. In addition, the inherent nature of the PID speed controller makes drive performance very parameter and load sensitive. Pillay and Krishnan have not reported drive performances over wide speed ranges, but rather only for certain command speeds.

In other work, Pillay et. al., [29] have proposed a control scheme for a PMSM based on dual digital signal processors (DSP). One DSP is used to implement the current controller algorithm and the other the vector control algorithm. The performance of this drive is diminished due to the slower speed of the DSPs employed; and their use of a look-up table in generating command currents may not be suitable for a wide range of speed operations. Therefore, there is a lack of robustness with this drive, and comprehensive test results at different dynamic operating conditions have not been reported. However, the experimental results do show the effectiveness of the controller and its potential.

B.K. Bose [30] has presented a high performance inverter-fed IPMSM drive system using a closed-loop torque control scheme in which command torque is determined using feedback, taking into account the effects of saturation, temperature variation and non-linearity. The drive system has been incorporated in the constant torque as well as constant power regions. However, the performance of this drive system has been reported only for a fixed speed, and while the sophistication of the drive is apparent, testing over a wide speed range and at different dynamic operating conditions is necessary to establish the efficacy of the drive.

An adaptive current control scheme for the PMSM drive has been proposed by Huy and Dessiant [31]. Their controller uses two modes, a hysteresis current control scheme for transient operation and a predictive current control scheme for steady-state operation. The drive appeared somewhat promising within the confines of the test parameters, but the performance of the drive at low operating speeds has not been reported.

Huy et. al., [32] have presented an analysis and implementation of a real-time predictive current controller for a PMSM synchronous motor servo drive. Although a high performance standard has been obtained with this controller, its implementation requires hardware incorporating an EPROM-based approach, making this scheme less flexible than the microprocessor-based approach due to the nature of EPROM programming and necessity for a dedicated EPROM board.

Bose and Szczesny [33] have proposed a microcontroller-based controller of an IPMSM drive for electric vehicle propulsion. The control system incorporates both the constant torque region operation as well as the constant power region operation, where the flux-weakening mode is used. This drive system has been implemented on a multiprocessor architecture, making the overall system somewhat cumbersome and costly.

Jahns et. al., [34] reported an adjustable speed drive using a torque control technique for an IPMSM by providing control of the magnitude and phase angle of the sinusoidal phase currents with respect to the rotor orientation. This method, while apparently effective, cannot provide a smooth transition from the constant torque mode to the constant power mode while the motor is in operation. In addition, the performance of this drive system has not been investigated over a wide speed range.

Jahns [35] also proposed a flux-weakening mode operating scheme for the IPMSM. This allows for investigation of the performance of the drive over an extended speed range. In this method, the d-axis rotor current is calculated from sensor-obtained stator phase currents and the d-axis command current.

Likewise, Morimoto et. al., [36-41] have proposed a flux-weakening mode-based controller for an IPMSM drive. In this work, the magnetic saturation and demagnetization effects of permanent magnets has been accounted and compensated for. This results in high torque and high efficiency operation within the maximum voltage and current limit ratings of the inverter and the motor. Different control methods, such as $i_d = 0$, unity power factor and constant flux linkage schemes have been investigated, and a comparison of the various methods has been made over a wide speed range. Regarding the magnitude and phase errors inherent when using a current controlled VSI, a compensating technique based on a calculated value of q-axis inductance, L_q, from the actual q-axis current has been used. Then, in order to overcome saturation, the d-axis command current is generated from the calculated value of L_q. The shortcomings of this rather comprehensive work lie in the high specificity of the algorithms and the effects of parameter variations due to noise, temperature, etc. are not considered.

Rahman et. al. [42] have proposed a flux-weakening mode based torque controller for the IPMSM drive for operation exceeding base speed. Maximum voltage and current capabilities of the motor and the inverter during operation are also accounted for in this work. The drive has not been tested for variable speed operation.

Vaez et. al. [43] have proposed a vector control strategy of the IPMSM drive aimed at producing minimum loss operation. They have used a PI-based speed controller, making this scheme sensitive to parameter variations, load changes, etc..

Radwan et. al. [44] have developed a hybrid current controller for the IPMSM drive which incorporates a ramp comparator controller for low speed operation and a hysteresis current controller for high speed operation. This controller produces stable operation over a wide speed range. However, as with

Vaez et. al. the speed controller is PI-based, and thus is also sensitive to parameter variations, load changes, etc.

Hoque et. al. [45] have reported a vector control strategy of the IPMSM drive based on the maximum torque per ampere (MTPA) scheme. In this work, d-axis command current is obtained from q-axis command current by use of a Taylor series expansion to overcome mathematical difficulties arising from nonlinearities of the model. A PI-based speed controller has been used. This approach was proposed by myself as part of the research team, and was done in conjunction with the work presented in my Masters of Engineering thesis.

Other works [46-49] investigating PMSM drive performance while in flux-weakening mode have been reported. Most of these works are based on conventional PI and PID based speed controllers. These controllers offer the advantages of simplicity and ease of implementation in real-time, however, they are very sensitive to parameter variations due to load changes, sudden changes of command speed, saturation, temperature variations and other system uncertainties. For these reasons, it is difficult to tune the controller parameters precisely for an optimal implementation. Consequently, these types of controllers are not suitable for high performance applications, and researchers have been prompted to develop adaptive control schemes for PMSM drive systems in which the controller parameters can be adapted, in real-time, in response to system parameter variations and load changes.

1.3.3 PMSM Drives with Adaptive Controllers

To date, adaptive controllers have been used in PMSM drives to achieve fast transient responses, parameter insensitivity, nonlinear load handling capabilities and high adaptabilities to other types of uncertainties.

The model reference adaptive controller (MRAC) is one such scheme in which the drive forces the motor response to follow the output of a reference model, regardless of drive parameter changes, by adjusting the controller parameters based on the error between the reference model response and actual motor response. An MRAC may be used with a PI controller to adapt the controller parameters compensatively for system parameter changes. Parameters are adapted by trial and error such that the error between the actual and the desired responses remains within specified limits.

Choy et. al., [50] have proposed a vector control position servo PMSM drive system using an MRAC. In their work, an MRAC is used to reactively tune a PI controller. The steady-state error gain component of the PI controller is used to compensate the chattering problem that occurs due to discontinuous control inputs. However, this drive still does not completely overcome the chattering problem.

Cerruto et. al., [51] have proposed an MRAC-based PMSM drive for robotic applications. An MRAC has been used to compensate for changes in system parameters such as inertia and torque constant. As explained previously, the error between the reference model speed and the actual speed is used to adjust the parameters. The shortcoming of this model is the computational burden that the algorithm imposes on the microcomputer. This limits the maximum operating speed of the drive.

Sozer and Torrey [52] have proposed an MRAC-based PMSM drive utilizing an adaptive flux-weakening mode controller that adjusts the d-axis current, i_d, compensatively. However, this drive has been reported in simulation only and has not been tested at variable speed conditions.

MRAC is only one of several types of adaptive controllers. Other popular schemes that have been used in PMSM drive systems are the variable structure controller (VSC) and a particular type of VSC called the sliding mode controller (SMC). These controllers utilize constraints of the state-space model to achieve effective control. They do, however, have a tendency to produce motor responses which exhibit significant chattering if traditional implementation techniques are used without some form of smoothing function.

Namudri and Sen [53] have presented an SMC-based vector control system of a synchronous motor drive for a position servo drive. In their work, a gate turn-off (GTO) inverter and phase-controlled chopper are used to provide the torque producing current component. The controller effectively accommodates parameter variations and load changes.

Consoli and Antonio [54] have proposed a DSP-based vector control scheme for an IPMSM drive using an SMC for torque control. In this work, both the actual motor currents and terminal voltages are used as feedback signals to generate the torque and flux commands for flux-weakening mode operation above base speed. The effects of constant acceleration, constant speed and constant deceleration have been accounted for in the design of the SMC. However, this drive system has been presented in simulation only and not in real-time where the parameter variations are not defined.

An ac servo drive using a variable structure controller (VSC) for position and speed control of a PMSM has been presented by Ghirby and Le-Huy [55]. Two control loops have been used, an inner loop for predictive current controllers and an outer loop for a position or speed controller. The performance of the drive has not been reported for wide variable speed operations, and suffers somewhat from chattering.

Sepe and Lang [56] have proposed an adaptive speed controller for the PMSM drive system in which the mechanical parameters of the motor have been estimated in real-time to continually tune the gain of the controller. As with Ghirby and Le-Huy, the system is composed of two loops: An inner loop consisting of the motor, inverter, current controller, speed controller and filter; and a slower, outer loop consisting of a motor parameter estimator and the control algorithm for the controller. This system has been implemented on a microprocessor of limited computational capacity, therefore, the performance of the drive is impaired by a significant lack of precision in steady-state operation.

Zhu et. al. [57] have proposed a DSP-based adaptive vector control scheme for various types of IPMSM drives using an online flux-weakening control strategy. Using an iterative process, the optimal q- and d-axis command currents are estimated and applied to the motor stator terminals through conversion to phase currents and a hysteresis PWM. No knowledge of machine parameters is required in the calculation of command currents. This work presents an improvement over several previous online adaptive schemes. However, because of the nature of the iterative search process for the optimal command current values, responses to changes in command speed are slow.

Clearly, compared to conventional fixed-gain PI and PID controllers, adaptive controllers show improved performance in several areas because of their relative insensitivities to parameter variations, load changes and other uncertainties. However, search algorithms can result in slow responses to step changes in command speed and almost all of the adaptive controller-based systems suffer from chattering, overreaching and steady-state errors due to finite switching. In addition, the unavailability of the exact system parameter model can still make for cumbersome design approaches for these types of controllers.

1.3.4 PMSM Drives with Intelligent Controllers

To solve some of the problems associated with fixed gain PI, PID and various adaptive controllers, recent researchers have investigated intelligent controllers such as fuzzy logic (FL), artificial neural network (ANN), neuro-fuzzy (NF), and genetic algorithm -based systems. These controllers do not require precise knowledge of motor parameters and advance information about system nonlinearities.

Inoue et. al., [58] have presented a fuzzy algorithm for the brushless dc servo motor drive. The fuzzy algorithm is used to tune the gain of the PI controller in response to load changes, parameter changes and system disturbances. The actual speed, reference speed and output of the reference filter are used to generate the membership functions. The experimental results show optimum response after several auto-tuning calculations. However, as the drive system incorporates a reference generator, somewhat complex fuzzy calculations and two PI controllers, the system imposes high computational burden, thus diminishing performance.

Erenay et. al., [59] have proposed a fuzzy logic approach for the brushless dc motor drives used in washing machines. They have made a comparison among various control techniques, namely, conventional PI, fuzzy PI, fuzzy reset rate and fuzzy gain scheduled PI. However, they have investigated the speed responses only for fixed speed conditions. Moreover, because of the large number of fuzzy rules incorporated, high computational burden may render these FLC-based systems incapable of high-speed operations. In experimental results, it is shown that the motor cannot follow the command speed smoothly.

A fuzzy logic-based MRAC for the PMSM drive has been proposed by Koviac et. al. [60]. Simulation results verify the effectiveness of the proposed algorithm, which was designed using the linearized model of the PMSM (the reference model being parameter dependent). This may not necessarily translate well to real-time implementation, where nonlinear load changes and parameter variations exist with different operating conditions.

Uddin and Rahman [61] have presented an FLC-based vector control scheme for the IPMSM. Their work uses a fuzzy algorithm to determine q-axis command current from the error between actual and reference speeds, and the difference between the current sample and previous sample speed errors. D-axis command current is set to be zero as a simplifying approximation. Heavy computational burden and the inaccuracy of the $i_{dr}^{*} = 0$ approximation limit the capacity of this scheme.

ANNs have been reported for use in controllers for PMSM drives. El-Sharkawi et. al., [62] have proposed one such scheme for a high performance brushless dc motor drive. In their work, an MRAC is used to implement a multi-layer ANN. The inputs of the ANN are the estimated speed from the reference model, three consecutive past samples of actual speed, a past sample of the converter input voltage, and the error between the reference model speed and the actual speed. A back-propagation algorithm is used to train the network. However, the speed control of this drive is not precise because the ANN must be trained offline. This shortcoming may render the drive incapable of effectively handling different dynamic operating conditions such as load changes, parameter variations and system disturbances.

Similarly, Shigou et. al. [63] have proposed an offline-trained ANN-based controller for a brushless dc servo motor drive system. In their work, they have used an analog speed controller in order to obtain better servo performance. However, they still have not produced satisfactorily precise speed control.

M.A. Hoque et al. [64] presented an ANN-based controller for permanent magnet dc motor drives. The shortcoming of this drive control system is that offline training of the network is necessary, as well as the nature of the static backpropagation training utilized has since been shown to be unsatisfactory for the control of high-performance drive systems under dynamically changing conditions.

Rahman and Hoque [65] have presented an online ANN-based PMSM drive system utilizing a back-propagation training algorithm and combined offline and online training. There are two artificial neural networks: One to generate the command signal and the other to generate the estimated signal. Controller parameters are then updated in accordance with the error between the two signals. In this work, however, the d-axis command current, i_d^*, is assumed to be zero, which makes it impossible to utilize the flux-weakening mode of operation to control the motor above base speed. There is also a reliance on an estimated ideal command signal. This means the drive will, to a degree, be limited by the inaccuracy of the model.

M.N. Uddin et al. [66] have proposed an IPMSM drive system which utilizes an ANN to tune the parameters of a genetic algorithm-based PI controller. Controller complexity and the inherent weaknesses of the PI strategy are the two prominent shortcomings of this work.

Another work, by Yang Yi et. al. [67] has utilized an ANN-based controller for an IPMSM. This work shows encouraging results, but, as with other ANN-based systems, the controller has an innate reliance on offline training. In addition, a reference model and the $i_d^* = 0$ approximation are utilized.

1.4 Problem Identification and Thesis Objectives

High Performance Drive (HPD) systems must provide fast and accurate responses, quick recoveries of reference speed from sudden disturbances of all natures, and show insensitivity to parameter variations. Each of the systems presented to date have shortcomings that require remedy if the IPMSM is to be implemented under HPD standards. Thus, it is necessary to further develop control algorithms and approaches to produce this high standard of performance in a practical manner.

While the IPMSM has many advantages over conventional motors, its operation is strongly affected by motor magnetic saliency, saturation and armature reaction effects, making an accurate motor model difficult to attain [68]. Particularly, due to the relative permeability of permanent magnets being very nearly unity, the magnet space itself behaves like air, resulting in the effective airgap length in the d-axis being larger than the q-axis in the IPMSM. The resultant saliency ($L_q > L_d$) causes the effects of magnetic saturation in the q-axis to be dominant (the q-axis exhibits less magnetic reluctance) and the q-axis inductance, L_q, to vary significantly with q-axis current and torque angle. Additionally, the saturation of the iron portion of the rotor around the permanent magnets produces a distortion of the air-gap flux that affects reactance parameters. As a result, the control performance is affected by magnetic saturation [37, 69-71]. This makes the control of the IPMSM for HPD applications an engineering challenge.

The objective of this work is the development and implementation of a complete IPMSM drive system for use in HPD applications. The indirect vector control scheme, incorporating a speed controller and a current controller, is used because it decouples the torque and flux, thus providing faster transient responses and making the control task easier. The test motor utilized and modeled is a 4-pole radially oriented interior permanent magnet type. An efficient speed controller, incorporating heretofore undeveloped methods, is presented for the high performance IPMSM drive.

As discussed in the literature survey, fixed-gain proportional integral (PI) and proportional integral derivative (PID) controllers, model reference adaptive controllers (MRAC) and variable structure controllers (VSC), including sliding mode controllers (SMC), all require the accurate and precise knowledge of system model parameters. Moreover, the fixed-gain PI and PID controllers are especially sensitive to parameter variations, load changes and other system disturbances. Fuzzy logic-based controllers (FLC) do not require information about the system mathematical model and can handle any kind of system non-linearity, but they incorporate complex algorithms that impose heavy computational burdens, thus they can only be incorporated by use of sufficiently powerful computer systems or by making performance compromises. They also require extensive offline tuning.

This work presents the development and real-time implementation of the IPMSM drive incorporating an intelligent, self-adaptive controller. Moreover, this control is achieved with minimal complexity and computational burden. In addition, the common simplification of the non-linear model of the IPMSM to a linear one by forcing the d-axis current to zero (i.e. $i_d = 0$) is not employed. This simplification is not accurate and, as a result, produces a motor control that requires increased stator current to produce a given torque. This work includes a practical approximation of maximum torque per ampere (MTPA) and flux-weakening (FW) schemes of operation with the real case of $i_d \neq 0$. This produces motor torque and command speed tracking with the minimum possible stator currents below rated speed and also for reliable operation above rated speed.

To this end, an efficient speed controller, incorporating an online, adaptively-tuned artificial neural network (ANN), which is insensitive to machine parameter inconsistencies such as dynamic d- and q- axis reactance variations, has been designed and implemented for the high performance IPMSM drive. This ANN-based speed controller differs from existing ANN-based controllers in similar applications, as training is done continuously by the network without reliance on a predetermined training set. The need for extensive off-line training of the ANN is eliminated. Speed error and an approximate torque error estimate are used for online tuning of the ANN to ensure accurate tracking of command speed. MTPA and FW modes of operation are achieved using mathematical calculations.

The end result is an intelligent control scheme for the IPMSM which is self-adaptive to uncertainties, can handle any kind of system non-linearity, and achieves this with maximum efficiency and minimal computational burden.

1.5 Thesis Organization

This thesis consists of seven chapters. The introduction and literature survey of vector control techniques for PMSM drives, as well as the objectives of this work have been covered in this chapter.

Chapter Two contains the theoretical development of the mathematical model of the IPMSM and the analysis and modeling of the PWM VSI-fed IPMSM drive.

Chapter Three outlines the development of the ANN-based speed controller for the IPMSM drive, as well as the maximum torque per ampere (MTPA) and flux weakening (FW) schemes of operation, and presents the ANN controller structure.

Chapter Four presents a four-neuron ANN-based speed controller for the IPMSM along with drive simulation results. For comparison purposes, simulation results of a traditional PID controller-based IPMSM drive are also presented.

Chapter Five presents a single artificial neuron (SAN) -based speed controller for the IPMSM along with drive simulation results.

Chapter Six presents real-time experimental results for the SAN controller-based IPMSM drive. Real-time experimental traditional PID controller-based IPMSM drive responses are included for comparison.

Chapter Seven contains a summary of this work and recommendations for future research.

Chapter 2

Analysis and Modeling of the VSI-Fed IPMSM Drive

2.1 General Introduction

This chapter presents the development of the mathematical model of a complete current-controlled voltage source inverter (VSI) -fed interior permanent magnet synchronous motor (IPMSM) drive using the d-q axis model of the IPMSM. A fixed-band hysteresis current controller has been used to apply the stator currents to the motor through the VSI.

In order to operate the vector control scheme, an artificial neural network based speed controller is used. This will be developed in Chapter 3.

2.1.1 Mathematical Modeling of the IPMSM

The IPMSM is similar to the conventional synchronous motor with the exception that the field excitation is provided by permanent magnets instead of a wire-wound dc rotor field. Because of this, the mathematical model of the IPMSM can be derived from the standard model of the synchronous motor by removing the equation related to the field current and other associated terms.

The flux linkages in the three stator phase windings due to the permanent magnets of the rotor are given in matrix form as [72]:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \psi_m \begin{bmatrix} \sin\theta_r \\ \sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \tag{2.1}$$

The three phase air gap flux linkage equations are given in matrix form as:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & M_{ab} & M_{ac} \\ M_{ba} & L_{bb} & M_{bc} \\ M_{ca} & M_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \psi_m \begin{bmatrix} \sin\theta_r \\ \sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \tag{2.2}$$

where i_a, i_b, i_c are the three phase currents, L_{aa}, L_{bb}, L_{cc} are the self inductances and M_{ab}, M_{bc}, M_{ca} are the mutual inductances, respectively, ψ_m is the constant flux supplied by the permanent magnets and θ_r is the rotor position angle. Rotor position angle θ_r is defined as,

$$\theta_r = \int_0^t \omega_r(t)dt + \theta_r(0) \tag{2.3}$$

The voltage equations of the three phases of the IPMSM can be defined as,

$$v_a = r_a\, i_a + \frac{d\psi_a}{dt} \tag{2.4}$$

$$v_b = r_b\, i_b + \frac{d\psi_b}{dt} \tag{2.5}$$

$$v_c = r_c\, i_c + \frac{d\psi_c}{dt} \tag{2.6}$$

where v_a, v_b, v_c are the three phase voltages and r_a, r_b, r_c are the three stator phase resistances. In matrix form, this is,

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} r_a & 0 & 0 \\ 0 & r_b & 0 \\ 0 & 0 & r_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + p \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \tag{2.7}$$

where p is the time differential operator, $\frac{d}{dt}$. Inspection of Equation (2.2) reveals that the flux linkages are functions of rotor position and, therefore, functions of rotor speed. This means that the coefficients of the voltage equations are time varying (except, of course, when the motor is stationary). In order to avoid the complexity of calculations, all of the equations can be transformed to the synchronously rotating rotor reference frame, as shown in Figure 1.1, where the machine equations are no longer dependent on the rotor position. This is accomplished using Park's transformation equations. First, the machine equations are transformed from the stationary a-b-c frame to the stationary d-q frame, then they are transformed from the stationary d-q frame to the synchronously rotating d^r-q^r frame. Using x to represent the machine phase variables, the inverse Park's transform gives,

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 1 \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & \sin\left(\theta_r - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & \sin\left(\theta_r + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} x_q \\ x_d \\ x_o \end{bmatrix} \tag{2.8}$$

The corresponding Park transform is,

$$\begin{bmatrix} x_q \\ x_d \\ x_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta_r & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \sin\theta_r & \sin\left(\theta_r - \frac{2\pi}{3}\right) & \sin\left(\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.9}$$

where x_o is the zero sequence component. The matrix element x may represent either voltage or current. Equations (2.8) and (2.9) are both in a stationary reference frame, so θ_r is only the initial rotor position $\theta_r(0)$ which is also the angle difference between the q-axis and a-phase. For balanced 3-phase, x_o does not exist and it is also convenient to set $\theta_r(0) = 0$ so that the q-axis coincides with the a-phase. Under these conditions, Equations (2.8) and (2.9) become,

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_q \\ x_d \end{bmatrix} \tag{2.10}$$

and

$$\begin{bmatrix} x_q \\ x_d \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.11}$$

To convert these variables to the rotating d^r-q^r frame we use Fig 2.1 as reference. From this, we see that the quantities in the stationary d-q frame can be converted to the synchronously rotating frame as follows:

$$\begin{bmatrix} x_q^r \\ x_d^r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x_q \\ x_d \end{bmatrix} \tag{2.12}$$

The inverse relation can be written as,

$$\begin{bmatrix} x_q \\ x_d \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x_q^r \\ x_d^r \end{bmatrix} \tag{2.13}$$

Now, using Equations (2.7), (2.10) and (2.13) the d^r-q^r model of the IPMSM can be written as,

$$v_q^r = r_s i_q^r + p\psi_q^r + \omega_s \psi_d^r \tag{2.14}$$

$$v_d^r = r_s i_d^r + p\psi_d^r - \omega_s \psi_q^r \tag{2.15}$$

where v_d^r and v_q^r are the d- and q-axis voltages, i_d^r and i_q^r are the d^r- and q^r-axis currents, respectively, r_s is the per phase stator resistance and ω_s is the stator frequency.

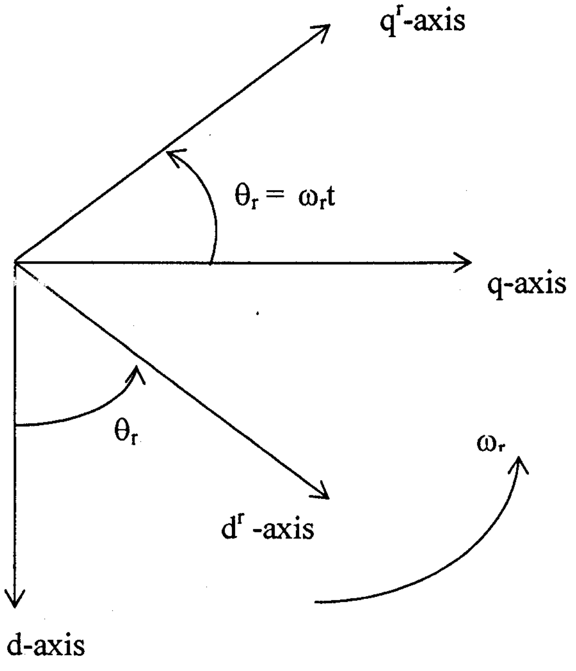


Figure 2.1 Relative positions of stationary d-q axes and rotating d^r-q^r axes.

2.1.2 IPMSM Model Including Iron Losses

The mathematical model traditionally used for the IPMSM neglects the effects of iron losses. In permanent magnet motors, lack of rotor slip and the reduction of stator losses due to operation at near-unity power factors means that iron losses form a significant portion of the total losses [73]. The resultant influence on the flux linkage and output torque of the motor due to these iron losses must be accounted for in high performance applications. Including iron losses in the model derivation helps to alleviate error and provides a more accurate IPMSM model for use with the vector control scheme [74]. The following section presents the development of the IPMSM model considering iron losses.

The q^r-axis flux linkage ψ_q^r and the d^r-axis flux linkage ψ_d^r can be written as,

$$\psi_q^r = L_q i_{qm}^r \tag{2.16}$$

$$\psi_d^r = L_d i_{dm}^r + \psi_m \tag{2.17}$$

where,

$$L_q = L_l + L_{qm} \tag{2.18}$$

$$L_d = L_l + L_{dm} \tag{2.19}$$

L_d and L_q are the d^r- and q^r-axis inductances, L_{dm} and L_{qm} are the d^r- and q^r-axis magnetizing inductances, respectively, and L_l is the leakage inductance per phase; $i_{dm}{}^r$ and $i_{qm}{}^r$ are the d^r- and q^r-axis magnetizing currents, respectively.

The stator frequency ω_s is related to the rotor frequency ω_r as,

$$\omega_s = P\omega_r \tag{2.20}$$

where P is the number of pole-pairs.

Therefore, Equations (2.14) and (2.15) can be rewritten as,

$$\begin{bmatrix} v_q^r \\ v_d^r \end{bmatrix} = r_s \begin{bmatrix} i_q^r \\ i_d^r \end{bmatrix} + \begin{bmatrix} pL_q & P\omega_r L_d \\ -P\omega_r L_q & pL_d \end{bmatrix} \begin{bmatrix} i_{qm}^r \\ i_{dm}^r \end{bmatrix} + \begin{bmatrix} P\omega_r \psi_m \\ 0 \end{bmatrix} \tag{2.21}$$

Figure 2.2 shows the d^r- and q^r-axis equivalent circuit diagrams of the IPMSM considering iron loss [74, 75]. To represent the effects of iron losses an iron loss resistance r_i is inserted in a parallel fashion with the magnetizing branch. Thus, the d^r- and q^r-axis line currents are divided into iron loss currents (i_{qi}^r and i_{di}^r) and magnetizing currents (i_{qm}^r and i_{dm}^r).

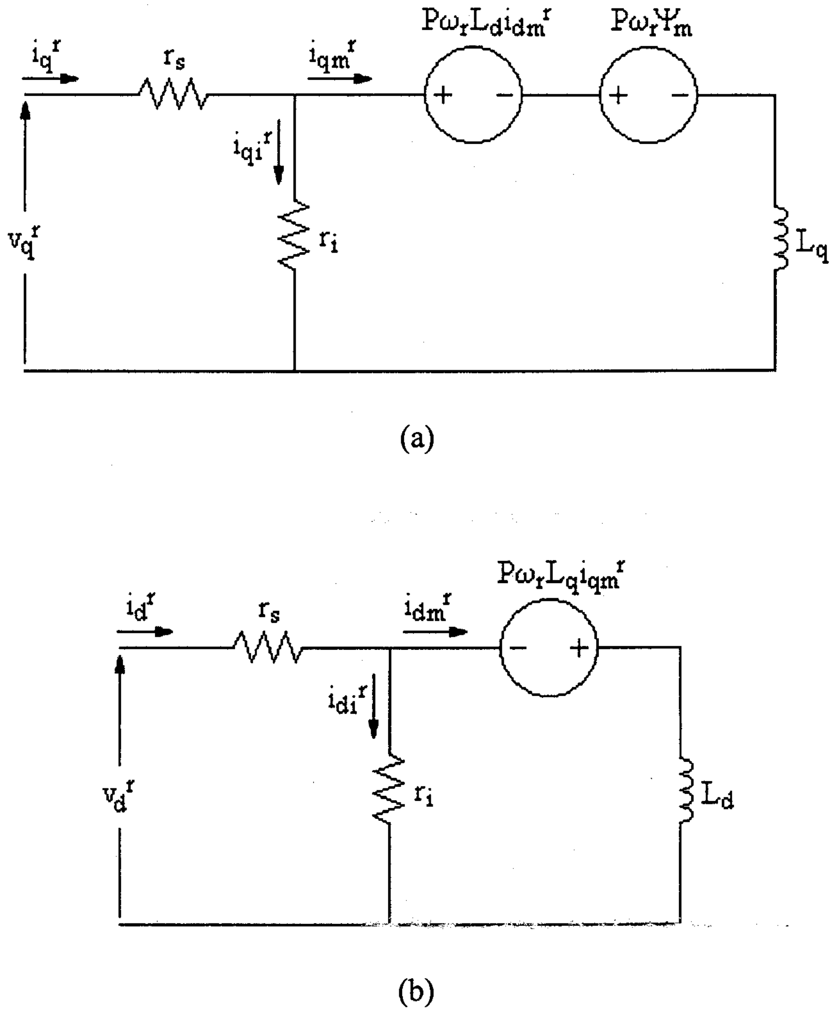


Figure 2.2 Equivalent circuit model of the IPMSM considering iron losses: (a) q^r-axis, (b) d^r-axis.

The magnetizing currents can be solved for as follows:

$$i^r_{qm} = i^r_q - \frac{P\omega_r}{r_i}\left(L_d i^r_{dm} + \psi_m + \frac{p i^r_{qm} L_q}{P\omega_r}\right) \qquad (2.22)$$

$$i^r_{dm} = i^r_d + \frac{P\omega_r L_q i^r_{qm}}{r_i} - \frac{p i^r_{dm} L_d}{r_i} \qquad (2.23)$$

At steady-state, the magnetizing currents can be found by substituting (2.23) into (2.22), giving the q^r-axis magnetizing current as

$$i^r_{qm} = i^r_q - \frac{P\omega_r L_d i^r_d}{r_i} - \left(\frac{P\omega_r L_d}{r_i}\right)\left(\frac{P\omega_r L_q}{r_i}\right) i^r_{qm} - \frac{P\omega_r \psi_m}{r_i} \qquad (2.24)$$

Substituting (2.22) into (2.23) gives the d^r-axis magnetizing current as

$$i^r_{dm} = i^r_d + \frac{P\omega_r L_q i^r_q}{r_i} - \left[\left(\frac{P\omega_r L_d}{r_i}\right)\left(\frac{P\omega_r L_q}{r_i}\right) i^r_{dm} + \left(\frac{P\omega_r}{r_i}\right)^2 L_q \psi_m\right] \qquad (2.25)$$

Using the assumption that iron loss resistance r_i is much greater than the reactances $P\omega_r L_d$ and $P\omega_r L_q$ yields

$$i^r_{qm} = i^r_q - \frac{P\omega_r L_d i^r_d}{r_i} - \frac{P\omega_r \psi_m}{r_i} \qquad (2.26)$$

$$i^r_{dm} = i^r_d + \frac{P\omega_r L_q i^r_q}{r_i} - \left(\frac{P\omega_r}{r_i}\right)^2 L_q \psi_m \qquad (2.27)$$

Similarly, the d^r-q^r axis currents can be solved for as follows:

$$i_q^r = i_{qm}^r + \frac{P\omega_r L_d i_{dm}^r}{r_i} + \frac{P\omega_r \psi_m}{r_i} \tag{2.28}$$

$$i_d^r = i_{dm}^r - \frac{P\omega_r L_q i_{qm}^r}{r_i} \tag{2.29}$$

The power developed per phase is given by,

$$P_{phase} = \left(-P\omega_r L_q i_{qm}^r i_{dm}^r + P\omega_r L_d i_{dm}^r i_{qm}^r + P\omega_r \psi_m i_{qm}^r\right)\left(\frac{1}{2}\right) \tag{2.30}$$

So, the total power developed by the machine is,

$$P_{mach} = \frac{3P\omega_r}{2}\left\{\psi_m i_{qm}^r + \left(L_d - L_q\right) i_{dm}^r i_{qm}^r\right\} \tag{2.31}$$

And the developed electromagnetic torque is given by,

$$T_e = \frac{P_{mach}}{\omega_r} = \frac{3P}{2}\left[\psi_m i_{qm}^r + \left(L_d - L_q\right) i_{dm}^r i_{qm}^r\right] \tag{2.32}$$

which can also be represented as,

$$T_e = T_L + B_m\omega_r + J_m p\,\omega_r \tag{2.33}$$

where T_L is the load torque, P is number of pole pairs of the motor, p is the differential operator, B_m is the friction damping coefficient and J_m is the rotor inertia constant.

Finally, the IPMSM model equations may be expressed as follows:

$$p i_{qm}^{r} = (v_q^{r} - r_s\, i_q^{r} - P\omega_r\, L_d\, i_{dm}^{r} - P\omega_r\, \psi_m)/\, L_q \tag{2.34}$$

$$p i_{dm}^{r} = (v_d^{r} - r_s\, i_d^{r} + P\omega_r\, L_q\, i_{qm}^{r})/\, L_d \tag{2.35}$$

$$p\omega_r = (T_e - T_L - B_m\, \omega_r)/J_m \tag{2.36}$$

The above three fundamental equations are used to model the IPMSM. The specific motor parameters used for simulation are given in Appendix A.

The following assumptions have been made:

(a) Iron loss is modeled as an equivalent copper loss.

(b) There is no squirrel cage on the rotor side.

(c) The induced emf is sinusoidal.

(d) The effects of magnetic saturation are neglected.

(e) There are no field current dynamics or magnet imperfections.

(f) The stator resistances of the three phases are balanced.

2.2 Vector Control of the IPMSM Drive

From Equation (2.32) one obtains the expression of torque:

$$T_e = \frac{3P}{2}\left[\psi_m i^r_{qm} + \left(L_d - L_q\right) i^r_{dm} i^r_{qm}\right].$$

We see that the second term in the electrical torque equation represents an interaction of inductances, L_d and L_q and also the currents, $i_{dm}{}^r$ and $i_{qm}{}^r$. However, in the case of the IPMSM, L_q is larger than L_d, and L_q, in particular, undergoes significant variations under different steady-state and dynamic loading conditions – varying as a function of $i_{qm}{}^r$ [67, 69, 70]. Hence, complexity of control of the IPMSM drive arises due to the nonlinear nature of the torque Equation (2.32).

Using phasor notations, and taking the d^r axis as reference, the steady state phase voltage V_a can be derived from the d^r, q^r-axis voltages described in Equation (2.21) by omitting the time varying components:

$$V_a = v^r_d + jv^r_q = r_s i^r_d + jr_s i^r_q + L_d \frac{di^r_{dm}}{dt} - P\omega_r L_q i^r_{qm} + jL_q \frac{di^r_{qm}}{dt} + jP\omega_r L_d i^r_{dm} + jP\omega_r \psi_m,$$

$$V_a = r_s I_a - \omega_s L_q i^r_{qm} + j\omega_s L_d i^r_{dm} + j\omega_s \psi_m \qquad (2.37)$$

Where $\omega_s = P\omega_r$ and the phase current I_a is given as,

$$I_a = i_d{}^r + ji_q{}^r, \qquad (2.38)$$

In the case of the IPMSM motor, the d^r-axis current is negative and it demagnetizes the main flux provided by the permanent magnets. Thus, in order to take only the absolute value of i_{dm}^r one can re-write the Equation (2.37) as follows:

$$V_a = r_s I_a - \omega_s L_q i_{qm}^r - j\omega_s L_d i_{dm}^r + j\omega_s \psi_m \tag{2.39}$$

where, the phase current, I_a is now given as,

$$I_a = -i_d^r + ji_q^r \tag{2.40}$$

Based on Equation (2.37) the vector diagram of the IPMSM is shown in Figure 2.3. It can be seen that the stator current can be controlled by the d^r-and q^r-axis current components, hence, the basis of the vector control scheme is illustrated. A complete indirect vector control scheme for the current controlled VSI-fed IPMSM drive is shown in Figure 2.4.

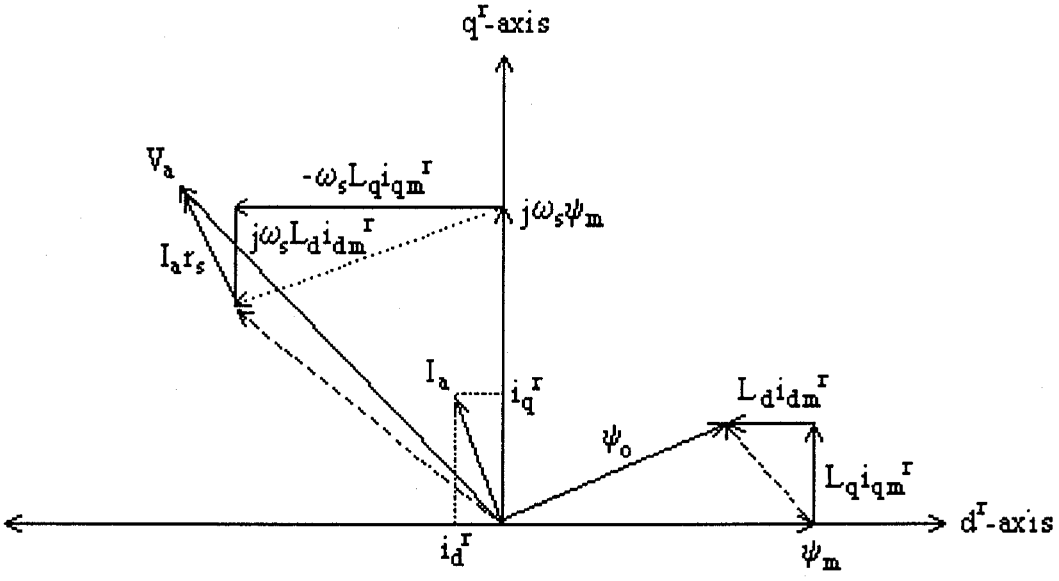


Figure 2.3 Vector diagram of IPMSM parameters.

2.2.1 Working Principle of the Drive System

The IPMSM drive consists, essentially, of the current controller and the speed controller. The speed controller generates the torque command from the command speed, the actual speed samples, and samples of the actual phase currents i_a, i_b and i_c. Then the command d^r-and q^r-axis currents i_d^{r*} and i_q^{r*} are generated from the command torque and the command phase currents i_a^*, i_b^* and i_c^* are generated from i_q^{r*} and i_d^{r*} using Park's transformation. From these command currents and the actual motor stator currents, i_a, i_b and i_c, the current controller provides the appropriate operating gating signals to the current-controlled voltage source pulse-width modulated (PWM) 3-phase, 6-pulse inverter, forcing the actual motor currents to follow the command currents as closely as possible and, hence, forcing the motor to follow the command speed by feedback control.

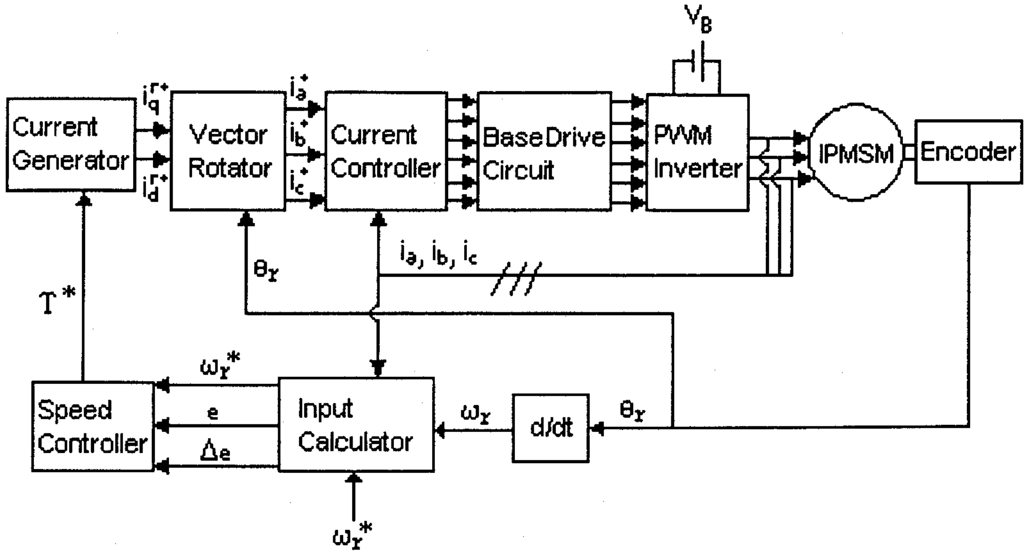


Figure 2.4 Block diagram of complete current-controlled VSI-fed IPMSM drive.

Therefore, in order to operate the motor in a vector control scheme the feedback quantities will be the rotor angular position and the actual motor currents.

2.2.2 Maximum Torque per Ampere Mode Speed Controller

The speed controller generates the torque command from the command speed and the actual speed samples. Traditionally, this has been accomplished by use of PI and PID controllers. However, these controllers produce unsatisfactory performance for high performance drive systems, so alternatives must be found (as discussed in Chapter 1). An Artificial Neural Network (ANN) -based speed controller, the development of which will be outlined in detail in Chapter 3, is proposed for this purpose. The controller relies on samples of rotor speed (as calculated from rotor position), command speed, and the d^r-q^r axis magnetizing currents, $i_{dm}{}^r$ and $i_{qm}{}^r$, which are determined from samples of i_a, i_b, and i_c via Equations (2.27), (2.26), (2.12) and (2.11). It is then necessary to calculate the required d^r-q^r axis magnetizing command currents, $i_{dm}{}^{r*}$ and $i_{qm}{}^{r*}$, to produce the command speed, from the command torque given by the ANN-based speed controller.

One of the main problems associated with the control of the IPMSM, which can be seen directly from inspection of the torque equation, is the non-linear nature of the developed torque as a function of $i_{dm}{}^r$ and $i_{qm}{}^r$. Many researchers have simplified the vector control of the IPMSM drive by forcing $i_{dm}{}^r = 0$, which linearizes the torque equation [25, 28, 32, 34]. However, in real-time the electromagnetic torque is truly non-linear in nature. In order to incorporate this non-linearity in a practical IPMSM drive, a control technique known as maximum torque per ampere (MTPA) has been derived which provides maximum motor torque with minimum stator current [25, 28, 32, 40, 76]. This MTPA strategy is advantageous as it optimizes drive efficiency when operating at or below base speed. Figure 2.5 shows the maximum torque per ampere trajectory that provides information for the control strategy.

Another difficulty associated with MTPA control technique is that its implementation in real-time is complicated by the complex relationship between d^r-axis and q^r-axis currents. Some researchers have solved this problem by the use of look-up tables for the d^r- and q^r-axis currents, based on the MTPA trajectory [42].

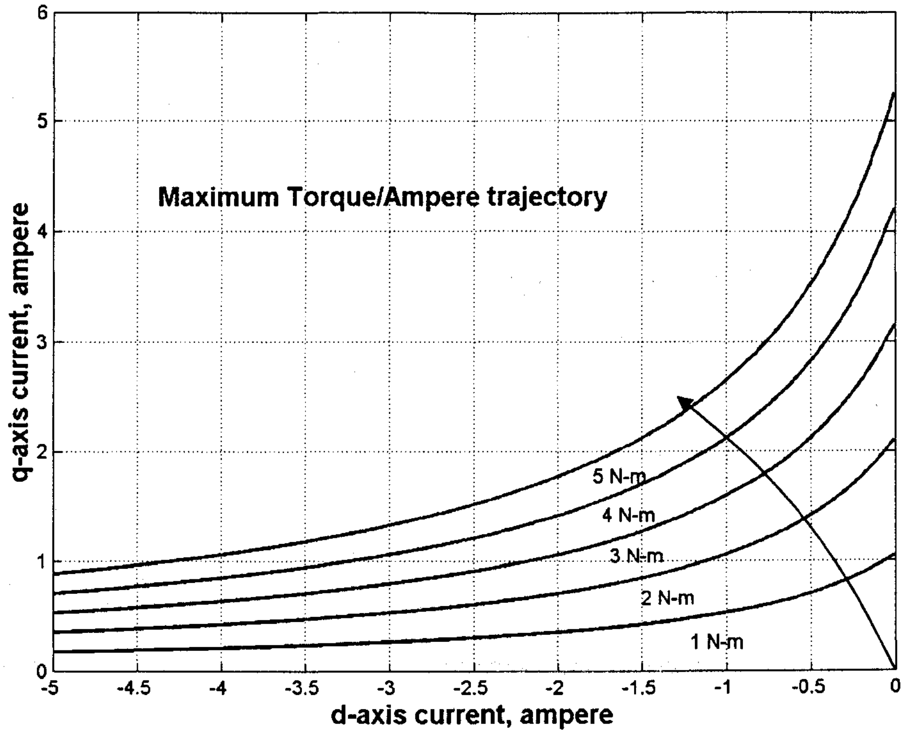
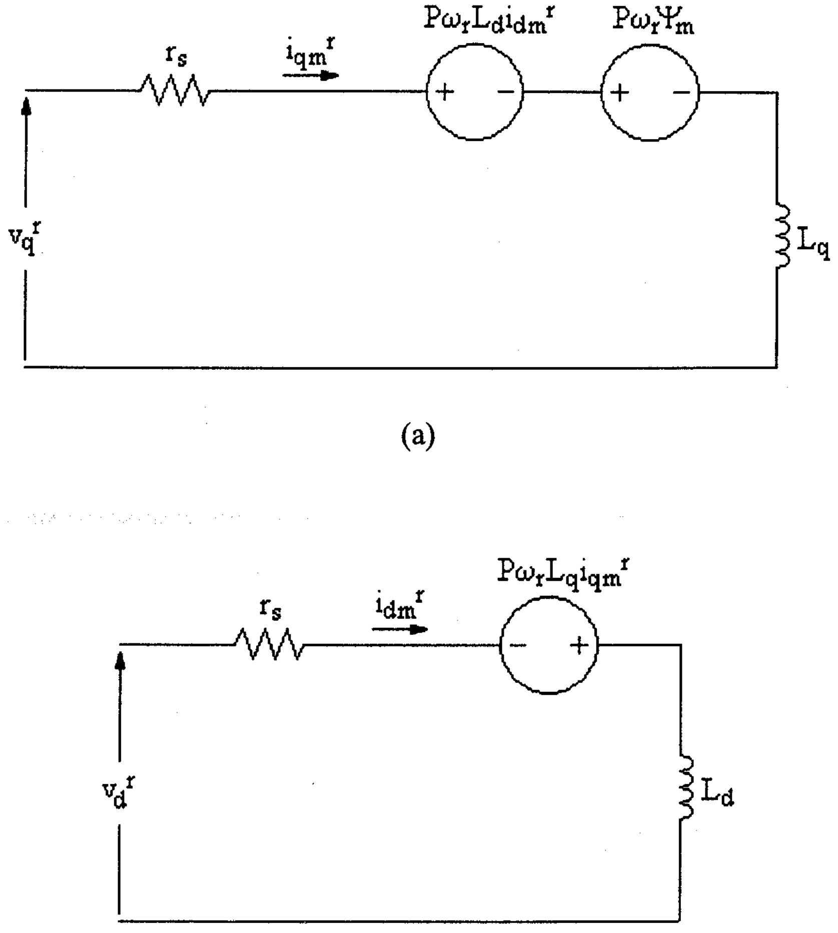


Figure 2.5 Maximum torque per ampere (MTPA) trajectory on constant torque loci.

For application of the MTPA technique, the lossless model of the IPMSM, as shown in Figure 2.6, is used. This is necessary to avoid mathematical complexities involved in trying to implement the MTPA scheme utilizing the IPMSM model including iron losses. In this model the d^r-q^r axis magnetizing currents are equal to the d^r-q^r axis currents themselves.



(a)

(b)

Figure 2.6 Equivalent circuit model of the IPMSM considering iron losses: (a) q^r-axis, (b) d^r-axis.

Neglecting iron losses, Equation (2.21) becomes,

$$\begin{bmatrix} v_q^r \\ v_d^r \end{bmatrix} = \begin{bmatrix} r_s + pL_q & P\omega_r L_d \\ -P\omega_r L_q & r_s + pL_d \end{bmatrix} \begin{bmatrix} i_{qm}^r \\ i_{dm}^r \end{bmatrix} + \begin{bmatrix} P\omega_r \psi_m \\ 0 \end{bmatrix} \tag{2.41}$$

Equation (2.40) becomes,

$$I_a = -i_{dm}{}^r + ji_{qm}{}^r \tag{2.42}$$

To implement the MTPA strategy, the relationship between $i_{dm}{}^r$ and $i_{qm}{}^r$ is obtained by differentiating the torque equation, Equation (2.32), with respect to $i_{qm}{}^r$ and setting the resulting expression to zero. The relationship between $i_{dm}{}^r$, $i_{qm}{}^r$ and I_a can be derived from Equation (2.42) as

$$i_{dm}^r = \sqrt{I_a^2 - i_{qm}^{r\ 2}}\ . \tag{2.43}$$

Giving,

$$\frac{dT_e}{di_{qm}^r} = \frac{3P}{2}\left(\psi_m + (L_d - L_q)i_{dm}^r - (L_d - L_q)i_{qm}^{r\ 2}\frac{1}{\sqrt{I_a^2 - i_{qm}^{r\ 2}}}\right) = 0 \tag{2.44}$$

$$\psi_m + (L_d - L_q)i_{dm}^r - (L_d - L_q)\frac{i_{qm}^{r\ 2}}{i_{dm}^r} = 0. \tag{2.45}$$

Rearranging the order of L_d and L_q and solving for i_{dm}^r gives,

$$i_{dm}^r = \frac{\psi_m}{2(L_q - L_d)} - \sqrt{\frac{\psi_m^2}{4\left(L_q - L_d\right)^2} + {i_{qm}^r}^2} \qquad (2.46)$$

Substituting Equation (2.46) into Equation (2.32), one can get a non-linear relationship between i_{qm}^r and T_e as,

$$T_e = \frac{3P}{2}\left(\psi_m i_{qm}^r - \frac{\psi_m i_{qm}^r}{2} - \left(L_d - L_q\right)\sqrt{\frac{\psi_m^2 {i_{qm}^r}^2}{4\left(L_q - L_d\right)^2} + {i_{qm}^r}^4}\right) \qquad (2.47)$$

Because of the square root terms, the real-time implementation of the drive system becomes difficult using Equations (2.46) and (2.47). This work presents a simplified relationship between the d^r-q^r axis magnetizing currents, which is obtained by expanding the square root term of Equation (2.46) using a MacClaurin series expansion. Substituting the appropriate experimental machine parameters given in Appendix A into Equation (2.46) gives,

$$i_{dm}^r = 4.22839 - \sqrt{17.87925 + {i_{qm}^r}^2} \qquad (2.48)$$

and expanding Equation (2.48) in a MacClaurin series about $i_{qm}^r = 0$, and neglecting higher terms, gives,

$$i_{dm}^{r} = -0.11825 i_{qm}^{r\,2} \tag{2.49}$$

Substituting Equation (2.49) into Equation (2.32) and solving for i_{qm}^{r} gives,

$$i_{qm}^{r} = 1.06157 T_{e} \tag{2.50}$$

Equations (2.49) and (2.50) are used in the speed controller to determine the rotating frame d^r-q^r axis magnetizing command currents for the MTPA control of the IPMSM. The appropriate d^r-q^r axis currents are then calculated via Equations (2.28) and (2.29).

It is of interest to note that the torque equation of the IPMSM, Equation (2.32), suggests that maximum torque cannot be obtained unless the reluctance component of the equation is exploited and is positive, and since L_q is greater than L_d in the IPMSM, i_{dm}^{r} must be negative in order to achieve this [70]. This confirms the negative values of i_{dm}^{r} given by Equation (2.49).

2.2.3 Flux-Weakening Mode Speed Controller

The IPMSM has a relatively large armature inductance and experiences enough flux-weakening (FW) effect due to d-axis armature reactance that the FW mode of constant power operation can be utilized to control the motor at high speeds [37, 41, 69, 77]. Therefore, above base speed, roughly where the magnet-generated back EMF causes rated stator voltage to be exceeded, the FW mode is used to generate the appropriate command currents.

In the FW scheme, the d- and q-axis components of phase current are constrained to maintain phase voltage at an optimum value – with rated phase voltage set as maximum. This is done to keep phase voltages and currents within rated values.

$$V_o = \sqrt{v_d^{r^2} + v_q^{r^2}} \le V_{om} \tag{2.51}$$

The lossless d^r-q^r axis model given in Equation (2.41) is redefined as follows [69]:

$$V_{om} = V_{am} - r_s I_{am} \tag{2.52}$$

$$\begin{bmatrix} v_q^r \\ v_d^r \end{bmatrix} = \begin{bmatrix} P\omega_r L_d & \\ -P\omega_r L_q & \end{bmatrix} \begin{bmatrix} i_{qm}^r \\ i_{dm}^r \end{bmatrix} + \begin{bmatrix} P\omega_r \psi_m \\ 0 \end{bmatrix} \tag{2.53}$$

where V_{am} and I_{am} are the maximum phase voltage and current, respectively. Steady-state is assumed, so the current derivatives with respect to time are reduced to zero, and Equation (2.52) allows stator resistance to be removed from the v_d^r-v_q^r model, resulting in the lossless model of Equation (2.53).

Solving Equation (2.53) for i_{dm}^r gives:

$$i_{dm}^r = -\frac{\psi_m}{L_d} + \sqrt{\frac{v_q^{r^2}}{P^2 \omega_r^2 L_d^2}} \tag{2.54}$$

Solving Equation (2.51) for v_q^r gives:

$$v_q^{r^2} = V_o^2 - v_d^{r^2} \tag{2.55}$$

Substituting Equation (2.55) into Equation (2.54) results in the following equation for i_{dm}^r:

$$i_{dm}^r = -\frac{\psi_m}{L_d} + \frac{1}{L_d}\sqrt{\frac{V_o^2}{P^2\omega_r^2} - L_q^2 i_{qm}^{r\ 2}} \tag{2.56}$$

Using the motor specifications given in Appendix A into Equation (2.56) gives,

$$i_{dm}^r = -7.39868 + \frac{1}{0.08488}\sqrt{\frac{V_o^2}{\omega_r^2} - 0.006331\ i_{qm}^{r\ 2}} \tag{2.57}$$

Expanding Equation (2.57) in a MacClaurin series about $i_{qm}^r = 0$, and neglecting higher terms, gives,

$$i_{dm}^r = -7.39868 + \frac{11.7813}{\omega_r} V_o \tag{2.58}$$

where

$$V_o \le V_{om} = V_{am} - r_s I_{am}$$

Substituting Equation (2.58) into Equation (2.32) with the appropriate experimental machine parameters and solving for i_{qm}^r gives:

$$i_{qm}^r = \frac{T_e}{1.76614 - \frac{1.31232}{\omega_r} V_o} \tag{2.59}$$

Equations (2.59) and (2.58) are used in the speed controller to determine the rotating frame d^r-q^r axis magnetizing command currents for the FW control of the IPMSM. The appropriate d^r-q^r axis currents are then calculated via Equations (2.28) and (2.29).

2.2.4 Vector Rotator

The command current outputs of the speed controller are in the rotating rotor reference frame. Therefore, they must be transferred to the stator reference frame. The inputs of the vector rotator block are the d^r-q^r axis command currents, i_d^{r*} and i_q^{r*}, from the speed controller, and the rotor position θ_r from the sensor on the motor. The outputs are the three phase command currents i_a^*, i_b^* and i_c^*.

The transformation is done by first converting the synchronously rotating d^r – q^r axis quantities to the stationary d-q axis quantities and then transforming these stationary d-q axis quantities into the corresponding a-b-c phase currents.

2.2.5 Current Controller and Voltage Source Inverter

Once the appropriate a-b-c phase command currents are generated by the vector rotator a current controller is used to control the voltage source inverter (VSI) to produce these command currents on the motor stator. The outputs of the current controller are the firing pulses for the inverter switches. Therefore, the VSI forces the motor to follow the command speed by feedback control.

The current control principle for the VSI used in this work is the based on the fixed-band hysteresis controller. The hysteresis current controller is most commonly used for high performance drive applications because of its simplicity and ease of implementation while providing acceptable results.

2.3 Current Control of the Voltage Source Inverter

2.3.1 Inverter Operation

The purpose of the current controller is to force the motor stator currents to follow the a-b-c phase command currents as closely as possible. These command currents are produced from the error between the command speed and samples of the actual speed of the motor with the intent of producing a motor speed which accurately tracks command speed. Therefore, under the scheme of speed control, the current controller forces the motor to follow the command speed.

The current-controlled VSI operation can be described, with reference to Figure 2.7, as follows: The errors between the actual motor currents and the command currents are processed by the hysteresis current controller to generate the pulse-width modulated (PWM) firing pulses for the transistors of the inverter.

Each stator phase of the motor is connected to the corresponding leg of the three-phase VSI. The center point of the two equal-valued capacitors is considered ground. The neutral of the stator is not connected to this ground.

In the figure, NA, NB and NC represent three binary logic variables of the three legs of the inverter, respectively. These logic variables determine the conduction state of the inverter. When NA is 1, T_1 is conducting and T_4, is not, and when NA is 0, T_4 is conducting and T_1 is not. NB and NC produce similar conduction patterns in the other two legs.

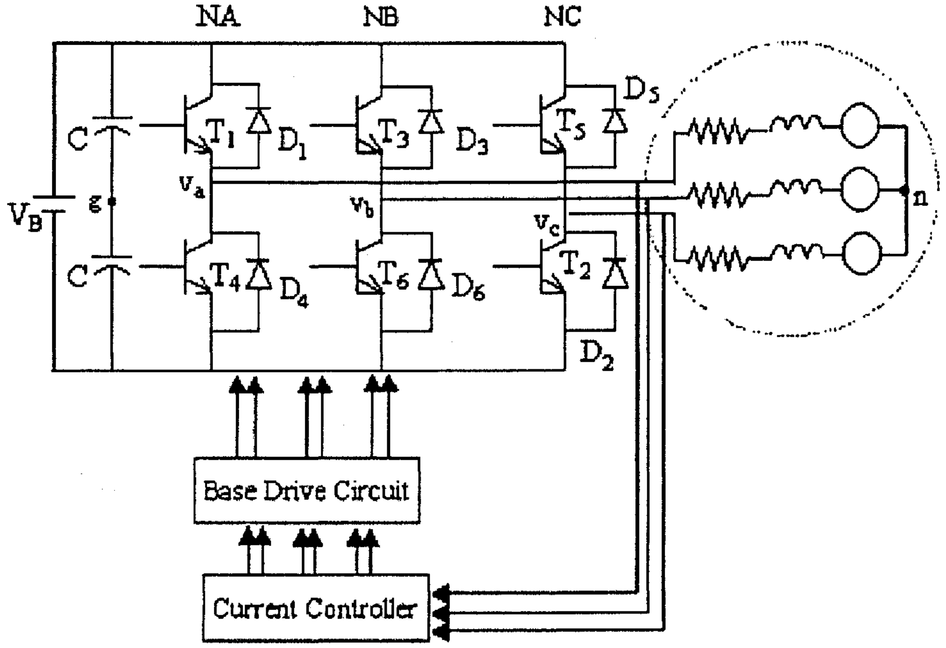


Figure 2.7 Current-controlled voltage source inverter for the IPMSM drive.

To simplify the analysis of the three phase voltages, voltage and space vectors are used to analyze the current controller. The inverter voltage vector can be defined as [16],

$$v = (2/3)\,(v_a + av_b + a^2 v_c) \tag{2.60}$$

where $a = e^{j2\pi/3}$, and v_a, v_b and v_c are the phase voltages.

The phase voltages v_a, v_b and v_c are expressed as functions of bus voltage, V_B, and the logic variables as,

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} NA \\ NB \\ NC \end{bmatrix} V_B\,. \tag{2.61}$$

There are eight switch combinations for the six switches of the inverter. Using Equation (2.60) the inverter voltage vector can be written as,

$$\begin{aligned} v_L &= (2/3)\,V_B\, e^{j(L-1)\pi/3} && \text{for L= 1, 2, ..., 6} \\ &= 0 && \text{for L= 0, 7.} \end{aligned} \tag{2.62}$$

The logic operation of the current-controlled VSI is summarized in Table 2.1.

The voltage vectors corresponding to the six active states are shown in Figure 2.8. The magnitude of each voltage vector is $(2/3)V_B$. The magnitudes of the voltage vectors corresponding to the freewheeling states, v_0 and v_7, are zero. In these cases, no voltage is applied to the motor because in these states no conduction path exists across any leg of the inverter.

Table 2.1 Conduction modes of the VSI under current control.

State, L	Leg 'a'		Leg 'b'		Leg 'c'		Operating modes	Voltage phasor
	T_1	T_4	T_3	T_6	T_5	T_2		
0	0	1	0	1	0	1	Freewheeling	v_o
1	1	0	0	1	0	1	Active	v_1
2	0	1	1	0	0	1	Active	v_2
3	1	0	1	0	0	1	Active	v_3
4	0	1	0	1	1	0	Active	v_4
5	1	0	0	1	1	0	Active	v_5
6	0	1	1	0	1	0	Active	v_6
7	1	0	1	0	1	0	Freewheeling	v_7

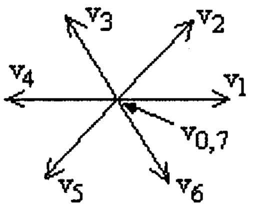


Figure 2.8 VSI voltage vectors.

2.3.2 Hysteresis Current Controller

With the hysteresis controller, the actual phase currents are allowed to vary within prescribed higher and lower limits known as a hysteresis band. This causes the switching frequency of the inverter to vary over the fundamental period, due to the sinusoidal nature of the command phase current – resulting in an irregular operation of the inverter with time and, therefore, an increase in switching losses as compared to some other types of current controllers. Still, this type of controller is the most extensively used because of its simplicity and excellent dynamic response.

There are conventionally two types of hysteresis controllers: Sinusoidal-band and fixed-band hysteresis controllers. In the sinusoidal-band type, the hysteresis band varies sinusoidally over the fundamental period. The advantage of this scheme is that the harmonic content of the current is low, but this scheme produces very high switching frequencies near zero crossings. As a result, the maximum switching frequency of the inverter increases substantially when this type of controller is used.

With the fixed-band hysteresis controller the hysteresis band is maintained constant. This reduces the maximum switching frequency of the inverter but the harmonic current is increased. As it is desired to keep switching frequency of the inverter to a minimum (to allow the use of BJT-based inverters), but still produce accurate dynamic command current tracking, this is the type of current controller used in this work.

Figure 2.9 illustrates the operating principle of the fixed-band hysteresis controller. In the figure, N1, N3 and N5 are the logic signals for the high transistors of the inverter and N4, N6 and N2 are the logic signals for the corresponding low transistors of the inverter. When the logic signal N1 is 1 then transistor T_1 is on and when it is 0 then T_1 is off. The other transistors follow logic signals likewise.

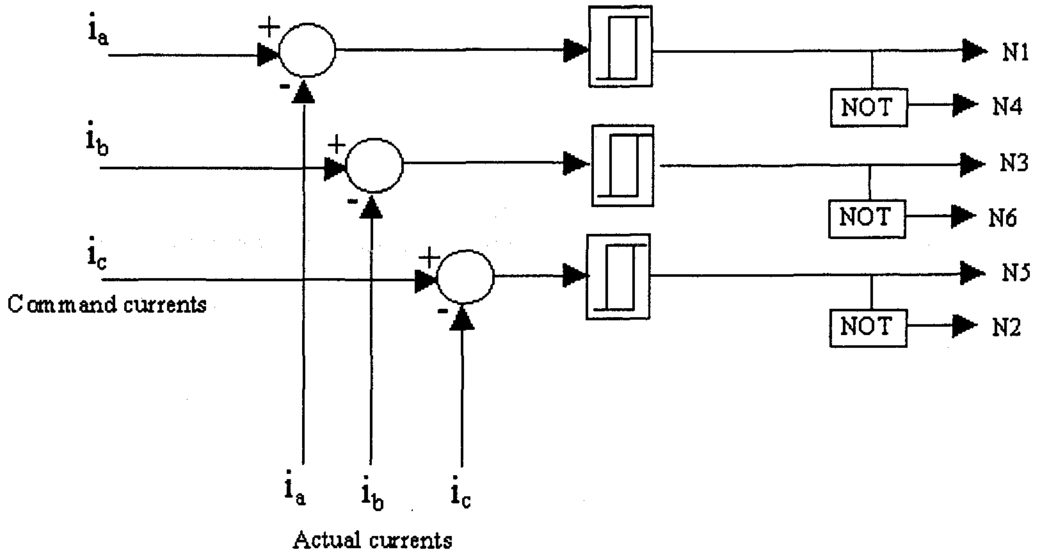


Figure 2.9 Fixed-band hysteresis current controller scheme.

The operational logic of the fixed-band hysteresis controller is illustrated in Figure 2.10. It can be described as follows:

(i) For $i_a^* > 0, N_4 = 0$: if $i_a > i_{up}$, then $N_1 = 0$, else if $i_a < i_{lo}$, then $N_1 = 1$.

(ii) For $i_a^* < 0, N_1 = 0$: if $i_a > i_{up}$, then $N_4 = 1$, else if $i_a < i_{lo}$, then $N_4 = 0$.

where i_a is actual 'a' phase current, i_a^* is command current, $i_{up} = i_a^* + H$ is the upper band, $i_{lo} = i_a^* - H$ is the lower band, and H is the fixed hysteresis band.

Looking at the case for $i_a^* \geq 0$, when the actual phase current reaches the lower limit, the upper transistor of the half bridge for that particular phase is turned 'ON' and the lower transistor is turned 'OFF' (for example, T_1 would be turned 'ON' and T_4 turned 'OFF' in the case of the 'a' phase). This switching action causes the phase current to increase positively in that phase. Then, as the phase current just reaches the upper limit of the hysteresis band, the upper transistor is turned 'OFF' while the lower one remains 'OFF' as well. Due to it's inductance, the current in the machine winding cannot go to zero instantaneously, therefore, as the upper transistor turns 'OFF', the free-wheeling diode across the lower transistor begins to conduct, changing the 'a' phase voltage from $+V_B/2$ to $-V_B/2$. The 'a' phase current is thus forced to decrease negatively. Control when $i_a^* < 0$ follows a similar logic.

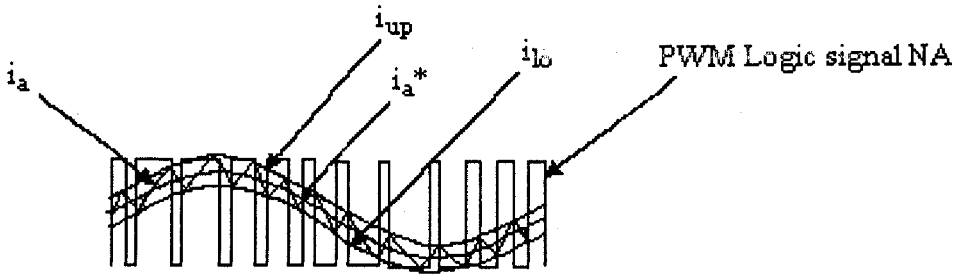


Figure 2.10 Hysteresis current controller waveforms.

The switching logic of the VSI for phase 'a' is summarized in Table 2.2. The operation of phases 'b' and 'c' would be similar. In this fashion, the actual current wave is forced to track the command current wave within the hysteresis band.

Table 2.2 Switching logic of VSI for phase 'a'.

i_a^*	i_a	T_1	T_4	V_a
≥ 0	$i_a \leq (i_a^* - \Delta i)$	ON	OFF	$+ V_B/2$
≥ 0	$i_a \geq (i_a^* + \Delta i)$	OFF	OFF	$- V_B/2$
< 0	$i_a \geq (i_a^* + \Delta i)$	OFF	ON	$- V_B/2$
< 0	$i_a \leq (i_a^* - \Delta i)$	OFF	OFF	$+ V_B/2$

2.3.3 Limitation of dc Bus Voltage and Inverter Switching Frequency

In order for the inverter to produce and follow the command currents there must be sufficient dc voltage across the inverter terminals to force the line currents in the desired direction. As the back emf from the motor stator is increased a point is reached where the dc bus voltage is insufficient, and the line-to-neutral voltage of the inverter (and therefore also the stator) becomes a six-step quasi-square wave. At this point the current controller is saturated and actual stator current is not able to follow command current.

Inverter switching frequency, which depends on the type of current controller used, also imposes a limitation on the operation of the motor. In the case of the hysteresis controller, the switching frequency depends on several factors that can be described by the following equation representing one phase of the motor:

$$L\frac{di_a}{dt} = v_a - Ri_a - e_a\,, \tag{2.63}$$

where L is phase inductance, i_a is phase current, R is per-phase resistance, v_a is phase voltage and e_a is phase back emf. The actual switching pattern is shown in Figure 2.11. In this figure, Δt_1 represents the time during which the line current will increase by Δi_{a1}, and Δt_2 is the time during which the current will decrease by Δi_{a2}. Assuming v_a and e_a are constant during the interval $\Delta t_1 + \Delta t_2$, the equation describing Δt_1 and Δt_2 can be written as,

$$\Delta t_1 = \frac{L\Delta i_{a1}}{v_a - Ri_a - e_a}; \quad \Delta t_2 = \frac{L\Delta i_{a2}}{v_a - Ri_a - e_a}, \tag{2.64}$$

Inverter switching frequency can be expressed as,

$$f = 1/(\Delta t_1 + \Delta t_2). \tag{2.65}$$

Therefore, switching frequency depends on the phase current (which is determined by the dc bus voltage), the inductance, and the magnitude and ripple content of the

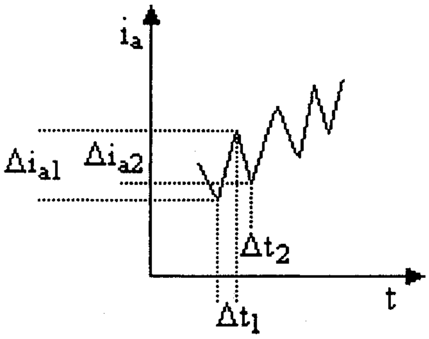


Figure 2.11 Hysteresis current controller switching pattern.

motor current. As the phase voltage and current vary periodically, so the inverter switching frequency will vary over the fundamental period.

2.3.4 Effect of Unconnected Neutrals

If the neutral of the Y-connected motor stator phases is not connected to the dc bus midpoint, the switching of any one phase depends on the states of the other two phases. This means that the individual line-to-neutral voltages are dependent on each other and each phase current will depend not only on the corresponding inverter phase but also on the state of the other two phases. Therefore, the current controller experiences interaction between the phases of the inverter, and this may introduce inaccuracy and imprecision into the tracking of command current.

2.4 Summary

This chapter has presented the development of the mathematical model of the IPMSM, including iron losses, and the indirect method of vector control of the IPMSM through a PWM-controlled VSI utilizing a fixed-band hysteresis current controller. The maximum torque per ampere and flux-weakening schemes of operation have been derived and developed for use in the IPMSM drive. Current-controlled voltage source inverter modeling and operation is explained.

Further chapters will develop intelligent speed control methods for the IPMSM drive to be used with the vector control scheme outlined here.

Chapter 3

Artificial Neural Network Based Speed Controller

3.1 Introduction

An artificial neural network (ANN) is an information processing method inspired by the way biological nervous systems, such as the brain, process information. In a mathematical analogy of real biological systems, highly interconnected processing elements (neurons) work in unison to produce the final output(s) in response to the given input(s). The ANN has been used successfully in recent years in many application areas.

By virtue of their highly trainable and adaptive nature, and in a process very similar to biological nervous systems, ANNs learn by example. “Learning” involves adjustments to the “synaptic” connections between the neurons and, in this manner, the network is trained and configured for a specific application.

Applications to date include pattern recognition, data classification and, of significance to the proposed research, motor control.

Neural networks take a different approach to problem solving than that of the traditional algorithmic approach (i.e. the microprocessor follows a prescribed sequence of instructions) where, unless the specific steps that the computer needs to follow are known, the computer cannot solve the problem. Conventional algorithms' problem solving capabilities, therefore, are restricted to problems that are already understood and are already solvable by traditional methods. Neural networks, however, by nature of their interconnected neurons working in parallel, cannot be programmed to perform such specific tasks – they learn only by example.

This means that the "answer" to the "problem" posed to the ANN does not necessarily need to be known, nor does any specific problem solving procedure. All that is needed is the input(s) to the ANN and acceptable output(s) which the ANN is required to produce in response. In this manner, the ANN is trained, by way of example, to "behave" in the desired manner. Therefore, the selection of the proper "teaching" examples must be done very carefully, otherwise useful time is wasted or, even worse, the network "learns" to function incorrectly or in a non-robust manner.

The disadvantage of the ANN is that because the network trains itself (by way of a training algorithm), its internal operation cannot easily be analyzed and its output can be unpredictable if the training data set is not complete. The primary advantage is that the ANN can be trained without knowledge of a solution procedure, and that the ANN can provide significant fault tolerance, since damage to a few neuron interconnections need not impair the overall performance of the network – it can re-learn [78].

3.1.1 Classification of ANNs

The two major classes of ANNs, which have received considerable attention, are:

1. Multi-layer Feed Forward Neural Network (FFNN)
2. Recurrent Networks

Multi-layer FFNNs have proved to be extremely useful in pattern recognition, image processing and speech recognition [79, 80]. Recurrent networks have been used primarily in associative memories and optimization problems [81].

Drive systems are inherently complex, with characteristics such as non-linearity, time varying parameters, time delay and saturation. This makes model uncertainty a major problem in control applications. Fixed gain controllers such as PI and PID controllers are tuned for specific operating ranges and conditions only, unable to adapt reactively to the changing dynamics of the system. And the large number of unknown parameters makes implementation with conventional adaptive controllers, which can react appropriately to such things as dynamically changing system parameters, complex and cumbersome. The ability of the FFNN to learn system characteristics through non-linear mapping, via a proper training scheme such as the back-propagation training algorithm, has resulted in the FFNN having been used widely in control applications [82, 83]. It is the FFNN type of ANN which has been utilized in this work.

3.2 The Neuron and FFNN Structure

3.2.1 Neurons

The "heart" of any ANN is its component neurons. An artificial neuron is a processing unit which has at least one, but perhaps many, input(s) and just one output. Neurons operate in a threshold manner of activation, meaning that they either produce output in response to a given input(s), or they do not.

The neuron has two modes of operation – "training" mode and "using" mode. In training mode, the neuron is trained to produce output (or no output), for particular input patterns. In using mode, when a previously encountered input pattern is detected at the input, the neuron outputs its learned response. If the input pattern is unfamiliar, then the response can be unpredictable.

Neurons are connected to each other in layers, with the interconnections between them "weighted" so that the weight of an input to a neuron is multiplied by a prescribed amount that is determined during training (see Figure 3.1). All of the weighted inputs are then added together and if they exceed a preset threshold value, the neuron is activated to produce an output. In any other case the neuron does not "fire".

Static FFNNs, such as those used in this work, have no feedback elements within them and contain no delays. This means that their outputs are calculated directly from the inputs through the networks' feed forward connections. In dynamic networks the outputs depend not only on the current inputs to the network, but also on the current or previous inputs, outputs, and/or states of the network. Dynamic networks may be FFNNs with imbedded tapped delay lines or recurrent networks with feedback connections [84].

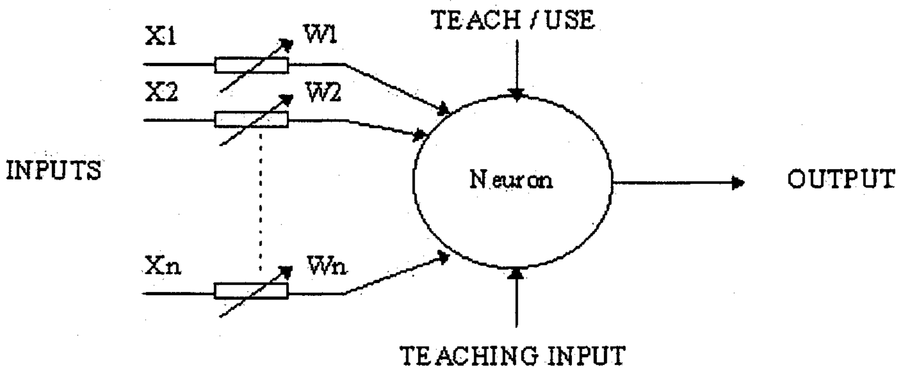


Figure 3.1 The Artificial Neuron.

3.2.2 Multi-layer Feed Forward Neural Networks

Typically, FFNN ANNs will have an input layer of input nodes (a "node" corresponding to each input), at least one "hidden" layer of neurons and an output layer (consisting of both neurons and outputs). A neuron, along with all of its interconnection inputs and outputs, is called a "unit". Because of the nature of the interconnections, the output layer will receive as its input(s) the output(s) of the last hidden layer; the input(s) to the first hidden layer will be the output(s) of the input layer. If there are several hidden layers they will be interconnected similarly. As previously mentioned, dynamic FFNNs will also have imbedded tapped delay lines feeding back into the neurons' inputs.

In linear systems, the output of a neuron, upon activation, will simply be a linear multiple (with possible biasing by addition of a constant) of the sum of the weighted inputs to the neuron. Non-linear ANNs typically use sigmoidal or other non-linear functions to compute neuron output. In both types of systems the mathematical operation performed on the summed input by the neuron itself is referred to as the neuron's "transfer function".

3.2.3 The Back-Propagation Algorithm

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. To minimize the error the neural network calculates the error derivative (the total rate of change of error as each weight is increased or decreased slightly) for each adjustment of weights. The back-propagation algorithm is the most widely used method for adjusting these weights and calculating the error derivative, and is the method used in this work.

The back-propagation algorithm is most easily understood if all of the units in the network are assumed to be linear. The algorithm computes each error derivative by first computing the rate at which the error changes (dE) as the activity level of a unit is changed. For output units, this rate is simply the difference between the actual and the desired output. To compute the dE for a hidden unit in the layer just before the output layer, all the weights between that hidden unit and the output units to which it is connected must be considered. Those weights are multiplied by the dEs of the output units and the products added up. This sum equals the error derivative for the given hidden unit. After calculating all of the error derivatives in the hidden layer just before the output layer we can compute, in similar fashion, the error derivatives for other layers – moving from layer to layer backwards through the network in a direction opposite to the way activities actually propagate when the network is operating – hence the name "back-propagation".

For non-linear units, the back-propagation algorithm includes an extra step – before back-propagating, the dE must be converted into the rate at which the error changes as the total input received by a unit is changed.

This work utilizes static FFNNs which employ the gradient-based back-propagation training algorithm outlined above.

3.3 ANN Training

To train a FFNN several known input-output patterns must be utilized. The inputs are "fed" to the network and the output then adjusted by back-propagation until the output is correct to within acceptable tolerances. Each input pattern is applied to the network and then propagates forward through each level until the output is reached. In each processing unit the net sum of weighted inputs is calculated first and then the sum is passed through the corresponding neuron (transfer function), with appropriate bias, and the output is then calculated for that unit. Thus, for a given pattern $X = (X_1, X_2, \ldots, X_N)^t$, the net-input value of the jth hidden-layer unit is:

$$sum_j^h = \sum_{i=1}^{N} W_{ji}^h X_i + B_j^h \qquad (3.1)$$

where the superscript h stands for the hidden layer, W_{ji}^h 's are the weights associated with the connections between the input layer and the hidden layer and

B_j^h 's are the biases associated with the hidden-layer neurons. The output of the j^th^ neuron of the hidden layer is found by applying the transfer function on the net-input as given below:

$$O_j^h = f_j^h\left(sum_j^h\right) \tag{3.2}$$

Similarly the net-input value and the output of the k^th^ output-layer unit can be calculated as:

$$sum_k^o = \sum_{j=1}^{L} W_{kj}^o O_j^h + B_k^o \tag{3.3}$$

where the superscript o stands for the output layer, W_{kj}^o 's are the weights associated with the connections between the hidden layer and the output layer and B_k^o 's are the biases associated with the output-layer neurons and

$$O_k^o = f_k^o\left(sum_k^o\right) \tag{3.4}$$

is the output of the k^th^ output-layer neuron. This output, O_k^o, is then compared with the desired or target output T_k and depending on the error between the obtained output and desired output, the weight of each connection and the bias of each processing unit are updated. To update the weights and biases, a sum-squared, or "cost", function is defined and the delta rule applied. This cost function is defined as:

$$E = \frac{1}{2}\sum_{k}\left(T_k - O_k^o\right)^2 \tag{3.5}$$

The main objective of ANN training is to minimize this function for all applied input patterns.

3.3.1 Output Layer Weight and Bias Updating

Because there exists a non-linear, multi-dimensional mapping between inputs and outputs, the steepest descent method is used to adjust the weights and biases to minimize the error function. This approach dictates that the change in weights and biases are proportional to the negative slope of the cost function [78],

$$\Delta W_{kj}^o \propto -\frac{\delta E}{\delta W_{kj}^o} \tag{3.6}$$

$$\Delta B_k^o \propto -\frac{\delta E}{\delta B_k^o} \tag{3.7}$$

For the output layer weights, in accordance with the chain rule, Equation (3.6) can be written as:

$$\frac{\delta E}{\delta W_{kj}^o} = -\left(T_k - O_k^o\right) \times \frac{\delta O_k^o}{\delta sum_k^o} \times \frac{\delta sum_k^o}{\delta W_{kj}^o} \tag{3.8}$$

Differentiating Equations (3.3) and (3.4) with respect to W_{kj}^{o} and sum_{k}^{o}, respectively, yields the following:

$$\frac{\delta sum_{k}^{o}}{\delta W_{kj}^{o}} = O_{j}^{h} \tag{3.9}$$

$$\frac{\delta O_{k}^{o}}{\delta sum_{k}^{o}} = f_{k}^{o'}\left(sum_{k}^{o}\right) \tag{3.10}$$

Substituting Equations (3.9) and (3.10) into Equation (3.8) gives:

$$\frac{\delta E}{\delta W_{kj}^{o}} = -\left(T_{k} - O_{k}^{o}\right) f_{k}^{o'}\left(sum_{k}^{o}\right) O_{j}^{h} \tag{3.11}$$

From Equation (3.6), the change in weights and biases for the output layer can be expressed as,

$$\Delta W_{kj}^{o} = -\eta \frac{\delta E}{\delta W_{kj}^{o}} \tag{3.12}$$

where η is a user-chosen proportionality factor known as the "learning rate", which is determined by trial and error.

Substituting Equation (3.11) into (3.12) gives:

$$\Delta W_{kj}^{o} = \eta\left(T_{k} - O_{k}^{o}\right) f_{k}^{o'}\left(sum_{k}^{o}\right) O_{j}^{h} \tag{3.13}$$

Setting,

$$\delta_k^o = \left(T_k - O_k^o\right) f_k^{o'}\left(sum_k^o\right) \tag{3.14}$$

where δ_k^o is referred to as the "local gradient", Equation (3.13) can be rewritten as,

$$\Delta W_{kj}^o = \eta \delta_k^o O_j^h . \tag{3.15}$$

This is the updating weight change which is applied to W_{kj}^o. Following a similar procedure starting with Equation (3.7), the updating bias adjustments are found to be:

$$\begin{aligned} \Delta B_k^o &= \eta\left(T_k - O_k^o\right) f_k^{o'}\left(sum_k^o\right) \\ &= \eta \delta_k^o \end{aligned} \tag{3.16}$$

In summary, the change in output layer weights and biases and the adjusted weights and biases on the k^{th} neuron of the output layer are expressed as:

$$\Delta W_{kj}^o = \eta \delta_k^o O_j^h \tag{3.17}$$

$$\Delta B_k^o = \eta \delta_k^o \tag{3.18}$$

$$W_{kj}^o(n+1) = W_{kj}^o(n) + \Delta W_{kj}^o(n) \tag{3.19}$$

$$B_k^o(n+1) = B_k^o(n) + \Delta B_k^o(n) \tag{3.20}$$

It should be noted that the learning rate is a very important factor – too small a value of η leads to a slow learning process, i.e. a large number of iterations are required; but, on the other hand, too large a value of η, while speeding up the learning process, can cause instability.

3.3.2 Hidden Layer Weight and Bias Updating

For the weights updating of the hidden layer, the gradient of the mean-squared error of the j^{th} neuron is given by:

$$\frac{\delta E}{\delta W_{ji}^{h}} = \frac{1}{2}\sum_{k}\frac{\delta\left(T_k - O_k^o\right)^2}{\delta W_{ji}^{h}} \tag{3.21}$$

$$= -\sum_{k}\left(T_k - O_k^o\right)\times\frac{\delta O_k^o}{\delta sum_k^o}\times\frac{\delta sum_k^o}{\delta O_j^h}\times\frac{\delta O_j^h}{\delta sum_j^h}\times\frac{\delta sum_j^h}{\delta W_{ji}^{h}} \tag{3.22}$$

Differentiating Equations (3.1) to (3.4) with respect to W_{ji}^h, sum_j^h, O_j^h and sum_k^o, respectively, one gets:

$$\frac{\delta sum_j^h}{\delta W_{ji}^{h}} = X_i \tag{3.23}$$

$$\frac{\delta O_j^h}{\delta sum_j^h} = f_j^{h'}\left(sum_j^h\right) \tag{3.24}$$

$$\frac{\delta sum_k^o}{\delta O_j^h} = W_{kj}^o \tag{3.25}$$

$$\frac{\delta O_k^o}{\delta sum_k^o} = f_k^{o'}\left(sum_k^o\right) \tag{3.26}$$

Substituting Equations (3.23) to (3.26) into Equation (3.22) yields:

$$\frac{\delta E}{\delta W_{ji}^h} = -\sum_k \left(T_k - O_k^o\right) f_k^{o'}\left(sum_k^o\right) W_{kj}^o f_j^{h'}\left(sum_j^h\right) X_i \tag{3.27}$$

The change in weight on the j^{th} neuron of the hidden layer can be found by applying the delta rule to Equation (3.27):

$$\begin{aligned}
\Delta W_{ji}^h &= -\eta \frac{\delta E}{\delta W_{ji}^h} \\
&= \eta f_j^{h'}\left(sum_j^h\right) X_i \sum_k \left(T_k - O_k^o\right) f_k^{o'}\left(sum_k^o\right) W_{kj}^o \\
&= \eta f_j^{h'}\left(sum_j^h\right) X_i \sum_k \delta_k^o W_{kj}^o \\
&= \eta \delta_j^h X_i
\end{aligned} \tag{3.28}$$

where,

$$\delta_j^h = f_j^{h'}\left(sum_j^h\right) \sum_k \delta_k^o W_{kj}^o \tag{3.29}$$

Similarly, the change in bias on the j^{th} neuron of the hidden layer is found to be:

$$\Delta B_j^h = \eta \delta_j^h \tag{3.30}$$

Finally, the adjusted weights and biases on the j^{th} neuron of the hidden layer are expressed as:

$$W_{ji}^h(n+1) = W_{ji}^h(n) + \Delta W_{ji}^h(n) \tag{3.31}$$

$$B_j^h(n+1) = B_j^h(n) + \Delta B_j^h(n) \tag{3.32}$$

Much research has gone into the development of faster, more stable learning algorithms such as conjugate gradient, quasi-Newton, Gauss-Newton, Levenberg-Marquardt and extended Kalman filter methods [84-88]. However, these algorithms are typically more suitable for larger, more complex networks than the static FFFNs used in this work and require batch training (unsuitable for real-time updating) and impose increased computational burdens of their own. For the purpose of this research, a simple method, which helps in speeding up the learning process while avoiding the danger of instability, is to introduce a momentum factor α in the delta rule for the hidden layer(s). The hidden layer weight change equation can be rewritten according to the modified delta rule as:

$$\Delta W_{ji}^h(n) = \alpha \Delta W_{ji}^h(n-1) + \eta \delta_j^h X_i \tag{3.33}$$

3.4 ANN Controller for IPMSM

3.4.1 Reference Command Torque

Equation (2.33) can be rearranged and discretized to give:

$$T_L(n) = T_e(n) - J_m\, p\omega_r(n) - B_m\omega_r(n) \tag{3.34}$$

Since J_m and B_m are known, $\omega_r(n)$, the present sample of rotor speed, can be read from a position sensor mounted to the rotor shaft and $p\omega_r(n) = (\omega_r(n) - \omega_r(n-1))/t_s$ is the change in rotor speed over the sampling period; current load torque at the present sample, $T_L(n)$, can be calculated directly from Equation (3.34) if $T_e(n)$ is known at that sampling instant. $T_e(n)$ can be calculated from Equation (2.32) as:

$$T_e = \frac{3P}{2}\left[\psi_m i_{qm}^r + \left(L_d - L_q\right) i_{dm}^r i_{qm}^r\right]. \tag{3.35}$$

$i_{qm}^r(n)$ and $i_{dm}^r(n)$ can be calculated via Equations (2.11), (2.12), (2.26) and (2.27) from samples of $i_a(n)$, $i_b(n)$, $i_c(n)$ taken by current sensors. P and Ψ_m are known and L_d and L_q are assumed to have constant, known values.

With the current sample value of T_L now having been calculated, Equation (2.33) can be utilized again to calculate the reference command torque as follows:

$$T_{ref}^*(n) = K_{ref} J_m\, p\omega_r(n) + B_m\omega_r^*(n) + T_L(n) \tag{3.36}$$

where K_{ref} is a constant used to tune the inertial response of the drive, $p\omega_r = (\omega_r^*(n) - \omega_r(n))/t_s$ is the current sample value of speed error divided by the sampling time, $\omega_r^*(n)$ is the current sample of command speed and $\omega_r(n)$ is the current sample of rotor speed.

Thus, a reference command torque is calculated based on the present sample of calculated load torque, $T_L(n)$, the inertia and friction coefficients of the machine, J_m and B_m, and the present samples of rotor speed and speed error. The shortcoming of this value of command torque is that it is calculated with the assumption that L_d and L_q have constant, known values. Due to the effects of magnetic saturation and interaction between the axes, this assumption is, of course, incorrect. However, in this work this approximate command torque is to serve merely as a reference for the ANN, and for this purpose it is sufficiently accurate.

3.4.2 Motor Dynamics

The dynamic model of the IPMSM can be rewritten from Equations (2.21) and (3.34) as

$$L_q p i_{qm}^r + L_d P\omega_r i_{dm}^r = v_q^r - r_s i_q^r - K_b\omega_r \tag{3.37}$$

$$p\omega_r = (T_e - T_L - B_m\,\omega_r)/J_m \tag{3.38}$$

where $K_b = P\psi_m$. The load will be considered as having unknown nonlinear mechanical characteristics and can be modeled using the following equation as [61],

$$T_L = A\omega_r^2 + B\omega_r + C \tag{3.39}$$

where A, B and C are arbitrary constants. To make the control task easier, the equations of an IPMSM are expressed as a single input and single output system by combining Equations (3.38) and (3.39) in continuous time domain form as,

$$J_m \frac{d\omega_r}{dt} = T_e - (B_m + B)\omega_r - A\omega_r^2 - C. \tag{3.40}$$

A small increment ΔT_e in T_e causes a small increment $\Delta\omega_r$ in ω_r:

$$J_m \frac{d(\omega_r + \Delta\omega_r)}{dt} = (T_e + \Delta T_e) - (B_m + B)(\omega_r + \Delta\omega_r) - A(\omega_r + \Delta\omega_r)^2 - C \tag{3.41}$$

Subtracting Equation (3.40) from (3.41) gives,

$$J_m \frac{d(\Delta\omega_r)}{dt} = \Delta T_e - (B_m + B + 2A\omega_r)(\Delta\omega_r) - A(\Delta\omega_r)^2 \tag{3.42}$$

By replacing all the continuous quantities of Equation (3.42) by their finite differences, the discrete-time small signal model of the simplified IPMSM with nonlinear load can be given as,

$$\Delta T_e(n) = \frac{J_m}{t_s}\Delta e(n) + (B_m + B + 2A\omega_r(n))\Delta\omega_r(n) + A\{\Delta\omega_r(n)\}^2 \tag{3.43}$$

Hence,

$$T_e^*(n) = \int_{\text{discrete}} \Delta T_e(n) = f(\Delta e(n),\ \Delta \omega_r(n),\ \omega_r^*(n)) \qquad (3.44)$$

where, $\Delta e(n) = \Delta\omega_r(n) - \Delta\omega_r(n-1)$ is the change of speed error, $\Delta\omega_r(n) = e(n) = \omega_r^*(n) - \omega_r(n)$ is the present sample of speed error, $\Delta\omega_r(n-1)$ is the past sample of speed error, $\omega_r(n)$ is the present sample of actual speed, $\omega_r^*(n)$ is the present sample of command speed, t_s is the sampling time interval and f denotes the nonlinear function. Thus, the purpose of using the ANN speed controller is to obtain the command torque by mapping the nonlinear functional relationship between ANN command torque, T_{ANN}^*, and rotor command speed ω_r^*, speed error e(n) and change in speed error $\Delta e(n)$, with no reliance on knowledge of motor parameters.

From this command torque, T_{ANN}^*, Equations (2.49) and (2.50) or (2.58) and (2.59) are used to calculate the necessary d^r- and q^r-axis magnetizing currents to produce the desired rotor speed ω_r^* – the MTPA scheme is used at, and below, base speed and the FW scheme is used above base speed. The appropriate d^r-q^r axis currents are then calculated via Equations (2.28) and (2.29).

In real-time, the motor position information and the output of the ANN in terms of the command d^r- and q^r-axis currents, i_d^{r*} and i_q^{r*}, are used to get the motor command phase currents, i_a^*, i_b^* and i_c^*, by first calculating the corresponding stationary d-q axis command currents, i_d^* and i_q^*, and then using the inverse Park's transformation.

3.4.3 ANN Layout

The number of hidden layers and the number of neurons in the hidden layer(s) were chosen by trial and error with the goal of achieving the optimal balance between drive performance and computational complexity. The number of nodes in the input layer is dependent on the number of inputs to the ANN. Command speed, actual rotor speed and previously sampled rotor speed give three input vector elements for this application. The output of the ANN is command torque, requiring only one neuron in the output layer.

An ANN structure with three units in the hidden layer was initially utilized for this research. Further investigation was done with simpler ANN structures and a simplified "network" consisting of only one neuron, with no hidden layers, was also found to produce acceptably fast and accurate command speed tracking. Both ANN structures have been implemented in this work.

Because of their precision, ease of implementation, capacity to produce both positive and negative outputs, and quick convergence rates when used in the ANN for IPMSM, tan-sigmoidal transfer functions, as illustrated in Figure 3.2, are used for all neurons. They are defined as follows:

$$f(x) = tansig(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{3.45}$$

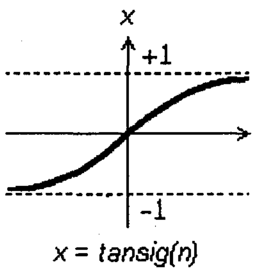


Figure 3.2 Tan-sigmoid function.

3.4.4 Initial Weights and Biases

The initial T_{ref}^* is calculated, as outlined in Section 3.4.1, based on command speed and speed error with the assumption that the initial load torque, T_L, is zero. This assumption is of little consequence as the initial speed error term itself invariably produces a torque command which is in excess of the motor's maximum rating (maximum torque command applied to the motor is limited for safety reasons). The ANN command torque value, T_{ANN}^*, is compared to this reference command torque and the network trains itself, by back-propagation, to produce an output that is within an acceptable tolerance of the reference. Substituting the lab motor parameters given in Appendix A into the torque equation, Equation (2.32), it can be seen that a 40% change in L_q produces less than a 10% variation in electric torque at full-load (2 Nm) and rated conditions. Changes in L_d have less effect. Therefore, as even relatively large variations in the L_q and L_d parameters have only modest effects on the command torque values given by Equation (3.36), the reference torque command serves as an acceptable approximate standard. The initial weights and biases are quickly "tuned" to bring the ANN output torque command error to within a predetermined percentage of this reference. Thus, the ANN produces its own initial weights and biases, and no specific initial weights and biases are necessary for startup and no off-line training need be performed.

Further samples of T_L are calculated, via Equation (3.34), based on the motor's response to the previous command torque, and sufficiently accurate calculations of T_L are made at each iteration.

Torque error-based retraining is done only when T_{ANN}^* differs unacceptably from T_{ref}^*.

3.4.5 Online Weight and Bias Adjusting

From Equations (2.34), (2.35), (2.36) and (2.32), it can be seen that the IPMSM model can be expressed in terms of d^r-q^r axis currents and rotor speed. As i_{dm}^r and i_{qm}^r are determined directly from the output command torque of the ANN-based speed controller, the back-propagation training algorithm can be used to update the ANN weights and biases of the torque-based speed controller using speed error directly [89]. Thus making the controller adaptive to load variations, parameter changes, noise, temperature, etc. Retraining is done based on speed error whenever the error exceeds a user-chosen acceptable threshold – if the speed error is below that threshold the existing weights and biases are reused. This means that retraining may take place at every iteration for many consecutive iterations. Once steady-state has been achieved, however, retraining will not occur unless disturbances, parameter variations, changes in command speed, etc. cause motor speed to differ unacceptably from the command speed.

At each iteration the command torque generated by the ANN is compared to the reference command torque, $T_{ref}^*(n)$, to ensure agreement. If the ANN command torque differs unacceptably from the reference then retraining is done, during that iteration, to return the ANN's output value to within acceptable limits. (Error tolerance determination is outlined in Chapter 4, Section 4.2) If retraining is not successful within a certain number of back-propagation iterations (to fix a limit on the computing time the retraining can take) then the reference command torque is used to generate motor currents for that iteration. This ensures that the ANN does not cause the drive to become unstable.

The speed error term is defined as:

$$E_\omega(n) = \frac{1}{2}e_\omega^{\ 2}(n) \tag{3.46}$$

where

$$e_\omega(n) = \omega^*(n) - \omega_r(n) \tag{3.47}$$

and $\omega^*(n)$ is the command speed at iteration n and $\omega_r(n)$ is the actual speed at that sample.

The torque error term is defined as:

$$E_T(n) = \frac{1}{2}e_T^{\ 2}(n) \tag{3.48}$$

where

$$e_T(n) = error_T = T_{ANN}(n) - T_{ref}(n)\,. \tag{3.49}$$

$T_{ANN}(n)$ is the command torque produced by the ANN for iteration n and $T_{ref}(n)$ is the reference command torque for iteration n.

The flow chart of this procedure is outlined in Figure 3.3.

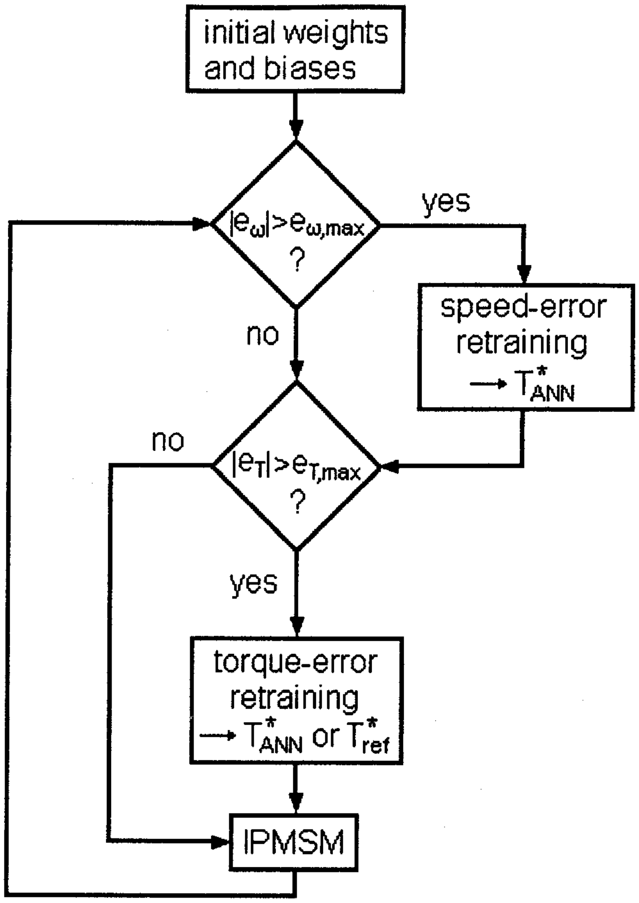


Figure 3.3 Flow Chart of Speed-based ANN Controller Operation.

3.4.6 Control Characteristics

As outlined, the three inputs to the ANN are speed error, change of speed error over the previous sample iteration and command speed. With fixed weights and biases these inputs would seem to produce control which is proportional-like, derivative-like and integral-like, respectively. However, due to the speed-error based self-tuning of the ANN and the interconnections between layers, the drive, when in closed-loop, will seek command speed even if one or more of the input weights are fixed. This occurs because the non-fixed weights and biases of any interconnections and neurons allowed to remain adaptive will be continually adjusted as long as speed error exists and if ANN output command torque differs unacceptably from the reference torque command. The more complex the network the less appropriate such a comparison to PID-based control becomes – with a "network" consisting of only one neuron being, generally speaking, most aptly compared to a very precise and adaptive PID controller which continually adjusts itself in real-time based on speed error and by comparison to a mathematically-derived reference torque command.

3.5 Summary

This chapter has presented the development of an ANN-based speed controller for the IPMSM drive that utilizes real-time back propagation training and requires no specific initial weights and biases and thus no offline training. The controller uses a reference torque command which produces command torque values accurate to within 10% of ideal values. The theoretical and mathematical basis of this reference command torque scheme has been thoroughly developed and presented.

Further chapters will continue the development and implementation of drives based on the theory presented in this chapter.

Chapter 4

Artificial Neural Network Based Speed Controller for IPMSM Drive

4.1 Introduction

A high performance IPMSM drive system is characterized by an efficient speed controller which causes the drive to follow the command speed under a wide range of varying operating conditions. This tracking of command speed is achieved rapidly and precisely regardless of system parameter variations and unexpected load excursions. Conventional controllers suffer from the fact that an accurate system model of the IPMSM is necessary for their design. This model is difficult to obtain due to necessary approximations, parameter variations, saturation effects, load disturbances, temperature variations, noise, etc. The consequence is that these typically controllers perform well only over the narrow range of specific conditions under which they were designed.

Adaptive controllers such as model reference adaptive controllers, self-tuning regulators, sliding mode controllers, etc., partially solve these problems by reacting adaptively to disturbances and parameter variations. However, they typically require IPMSM model knowledge and involve large numbers of unknown parameters, making their implementation in real-time difficult and often complex. An ANN can solve these problems by reacting adaptively to the speed error of the drive while not requiring accurate modeling of the drive system.

This chapter introduces a novel approach to the design and implementation of an ANN for the IPMSM drive. An online back-propagation training algorithm based on speed error is used for this controller, enabling the drive to track any arbitrary reference speed under unknown load dynamics and parameter variations. In addition, a back-propagation training algorithm based on approximate torque error is utilized to ensure the drive command torque remains within acceptable limits. This ensures the stability of the drive system. The complete drive system consists of an ANN-based speed controller, a voltage source inverter and an interior permanent magnet synchronous motor. The ANN-based controller consists of the input layer containing three input nodes, one hidden layer containing three neurons, and the output layer containing one neuron. The design has been developed in Chapter 3 and simulation model details are presented in Appendix B.

For comparative evaluation purposes, a PID controller-based IPMSM drive system has also been implemented and tested. Extensive simulations have been done in order to evaluate the performances of both drives at different dynamic operating conditions.

Finally, the simulation results are presented and discussed.

4.1.1 Vector Control of IPMSM with ANN Controller

The indirect vector control scheme of the IPMSM drive, as described in Chapter 2, has been implemented – with the ANN fulfilling the role of speed controller. The system is illustrated in Figure 4.1

The position sensor on the rotor shaft relays rotor position information, from which rotor speed is calculated, back to the input calculator algorithm. Along with this and the desired command speed, the input calculator uses phase current readings to determine the appropriate reference command torque. Command speed, rotor sample speed and speed error, are then fed to the ANN-based speed controller. The ANN controller generates the appropriate command torque to track command speed, as outlined in Chapter 3. Then, via the maximum torque per ampere or flux-weakening approximations – depending on whether command speed is at, below or above rated speed – the correct phase currents to produce the command torque are determined and applied to the motor stator via vector rotation calculations and the hysteresis current controller through a voltage source inverter.

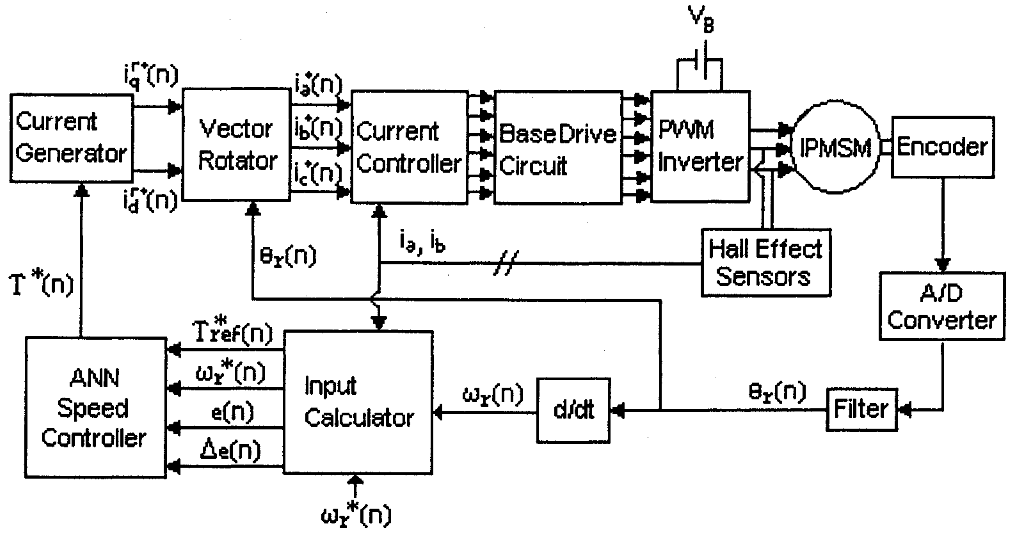


Figure 4.1 ANN based IPMSM drive.

4.2 ANN Based Speed Controller for IPMSM Drive

The Feed Forward Neural Network (FFNN) structure, as described in Chapter 3, is employed. The network consists of one input layer, one hidden layer and one output layer. The input layer contains three input nodes, the hidden layer consists of three neurons and their interconnections, and the output layer consists of one neuron, for a total of four neurons in the entire network. This ANN is illustrated in Figure 4.2.

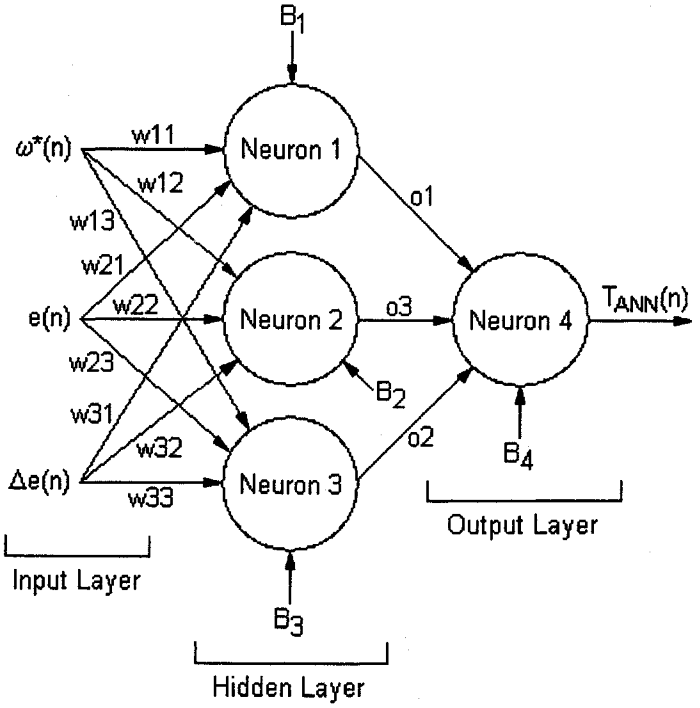


Figure 4.2 Four-neuron ANN for IPMSM drive.

Inputs to the ANN are present sample of command speed $\omega^*(n)$, present sample of speed error $e(n) = \omega^*(n) - \omega(n)$ and present sample of change of speed error $\Delta e(n) = \Delta\omega(n) - \Delta\omega(n-1)$. The weights applied to the inputs are represented as w11, w12, w13, w21, w22, etc. The weights applied to the outputs of the hidden layer neurons are o1, o2 and o3. The biases added to each neuron's summed input are represented as B1, B2, B3 and B4. As the output neuron, via the tan-sigmoid transfer function, can produce outputs only between -1 and 1, this output is multiplied by the maximum torque capacity of the motor (3 Nm, in the case of the experimental lab motor) in order to produce the proper range of command torque values. Thus, ANN output is the present sample of command torque $T_{ANN}(n)$.

From this output the maximum torque per ampere scheme, as outlined in Chapter 2, is used to determine the appropriate q- and d-axis command currents, i_q^* and i_d^* respectively, for operation at or below base speed. Above base speed, the flux-weakening mode, also outlined in Chapter 2, is employed. From these currents, the phase command currents, i_a^*, i_b^* and i_c^*, are calculated and applied to the motor through the VSI by a hysteresis controller.

In order to provide adaptive control, the weights and biases of the ANN are updated, by back-propagation, whenever the speed error exceeds an appropriately chosen threshold – in this case an error of 0.1 rad./sec. If this threshold is not exceeded the weights and biases will be used again in the following iteration.

To maintain drive stability, the output command torque of the ANN is compared to the reference torque as outlined in Chapter 3. If the relative error between the two, as expressed by $\left|\frac{T_{ANN}(n)-T_{ref}(n)}{T_{ref}(n)}\right|$, exceeds 0.10 then retraining will be done to return $T_{ANN}(n)$ to within 10% of $T_{ref}(n)$. 10% is chosen as this is the approximate deviation in torque caused by a 40% change in L_q at full-load (2 Nm external load plus inertia and friction effects). This is a realistic limit as L_d can be assumed to be constant and L_q is unlikely to change more than 40% during safe operation [69, 70].

Hence, the speed error, $e_\omega(n) = \omega^*(n) - \omega(n)$, as well as the approximated torque error $e_T(n) = T_{ANN}(n) - T_{ref}(n)$, are used for back-propagation updating, as described in Chapter 3. The flow chart of this procedure is outlined in Figure 4.3.

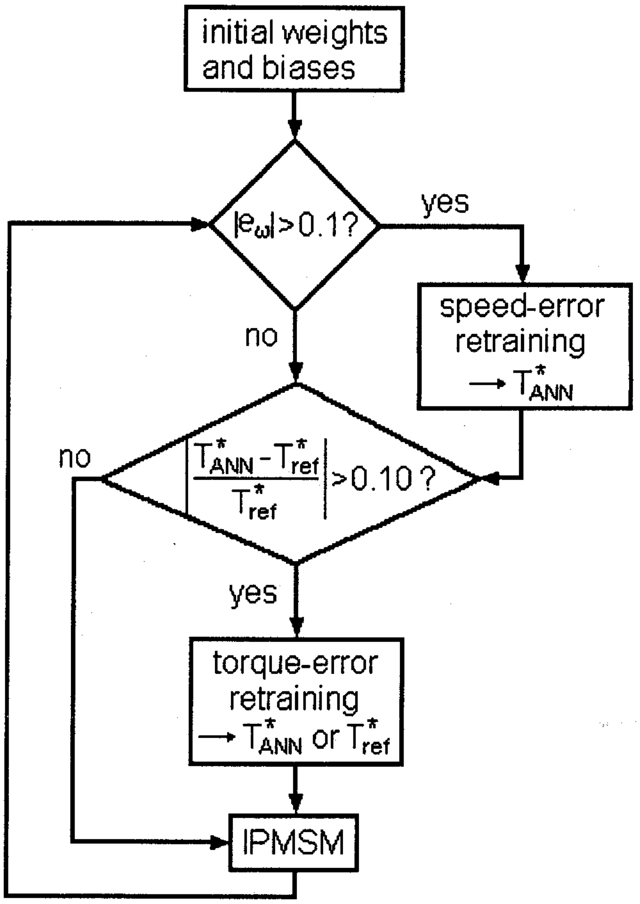

initial weights and biases
$|e_\omega|>0.1?$
yes
no
speed-error retraining $\longrightarrow T^*_{ANN}$
no
$\left|\frac{T^*_{ANN}-T^*_{ref}}{T^*_{ref}}\right|>0.10\,?$
yes
torque-error retraining $\longrightarrow T^*_{ANN}$ or T^*_{ref}
IPMSM

Figure 4.3 Flow Chart of Speed-based ANN Controller Operation.

4.3 Simulation Results and Discussion

This section presents simulation results of the four-neuron ANN controller-based IPMSM drive described above. The startup weights and biases of this network are as follows:

All weights (W_{ij}) = 1

All biases (B_n) = 0

No off-line training was done to determine these weights. They were chosen arbitrarily as precise initial weights and biases, as outlined in Chapter 3, are not necessary. Sampling frequency was set at 5000 Hz.

4.3.1 Rated Speed and Below Operation (with MTPA)

Figure 4.4 shows the startup response of the four-neuron ANN controller-based IPMSM drive under no load for a command speed of 188.5 rad./sec. This is base speed for the modeled four-pole IPMSM. The drive is shown to track the command speed accurately with minimal oscillation. No overshoot of command speed is exhibited in the transient state.

Figure 4.5 shows the startup response of an appropriately tuned PID controller-based IPMSM drive, utilizing the $i_d^r = 0$ approximation, under no load for a command speed of 188.5 rad./sec. A significant initial overshoot is exhibited as well as a longer time necessary to settle at the command speed.

Figure 4.6 shows the current in phase 'a' of the ANN controller-based IPMSM drive under no load for a command speed of 188.5 rad./sec. Figure 4.7 shows the current in phase 'a' of the PID controller-based IPMSM drive under the same conditions. In both cases, peak current values are roughly 1.5 A, with the ANN-based controller settling faster due to its quicker startup response.

Figure 4.8 shows the response of the ANN controller-based IPMSM drive to step changes of command speed (from 90 to 188.5 to 130 rad./sec.) at no load. The drive accurately and precisely follows these changes. Figure 4.9 shows the response of the PID controller-based IPMSM to step changes of command speed (from 90 to 188.5 to 130 rad./sec.) also at no load. The drive tracks the command speed changes, however, experiences overshoot, undershoot and increased settling time.

Figure 4.10 shows the current in phase 'a' of the ANN controller-based IPMSM drive under step changes of command speed (from 90 to 188.5 to 130 rad./sec.). Figure 4.11 shows the current in phase 'a' of the PID controller-based IPMSM drive under the same conditions. In both cases, peak steady-state current values are roughly 1.5 A, with the ANN-based controller settling faster after command speed changes due to its quicker speed response. It can also be seen that the ANN-based drive experiences higher currents during the transient acceleration and deceleration periods. This is explained by the drive's faster acceleration and deceleration.

Figure 4.12 shows the response of the ANN controller-based IPMSM drive to a sudden application of full-load (0 to 2 Nm) at a command speed of 188.5 rad./sec. The speed disturbance due to the load application is almost indiscernible. The rapid response of the drive is due, in large part, to the instantaneous reaction of the reference command torque calculation after one iteration of the disturbance. This produces a command torque within 10% of optimum within one sampling period. Figure 4.13 shows the response of the PID controller-based IPMSM drive to the sudden increase in load (0 to 2 Nm) at a command speed of 188.5 rad./sec. The speed disturbance due to the load application is quite dramatic.

Figure 4.14 shows the current response in phase 'a' of the ANN controller-based IPMSM drive under sudden application of full-load (from 0 to 2 Nm) at rated command speed. Figure 4.15 shows the current response in phase 'a' of the PID controller-based IPMSM drive under the same conditions. In both cases, peak steady-state current values increase to approximately 3 A when full-load is applied. Again, the ANN-based controller settles faster due to its quicker response.

Figure 4.16 shows the q-axis current response of the ANN controller-based IPMSM drive to a sudden application of full-load (0 to 2 Nm) at 188.5 rad./sec. speed command. Figure 4.17 shows the q-axis current response of the PID controller-based IPMSM drive under the same conditions. Again, the ANN-based drive reaches steady-state faster, but it also experiences a slightly lower peak value of i_q^r under load. This is, at least partially, due to the MTPA scheme of determining d-q axis command currents being utilized with the ANN controller-based drive.

Figure 4.18 shows the corresponding d-axis current response of the ANN controller-based IPMSM drive to a sudden application of full-load (0 to 2 Nm) at 188.5 rad./sec. speed command. Figure 4.19 shows the d-axis current response of the PID controller-based IPMSM drive under the same conditions. Here it can be seen that the maximum torque per ampere (MTPA) scheme produces an increased negative d-axis current when load is applied, taking advantage of the reluctance torque due to the saliency of the IPMSM. This can be seen in the second term of the IPMSM torque equation, Equation (2.32), when i_d^r is negative.

Figure 4.20 shows the startup speed response of the ANN controller-based IPMSM drive to a sinusoidal speed command with an amplitude of 25 rad./sec. and 163.5 rad./sec. offset. The drive quickly and accurately tracks command speed. Figure 4.21 shows the startup response of the PID controller-based IPMSM drive to the same speed command. The drive cannot accurately track command speed.

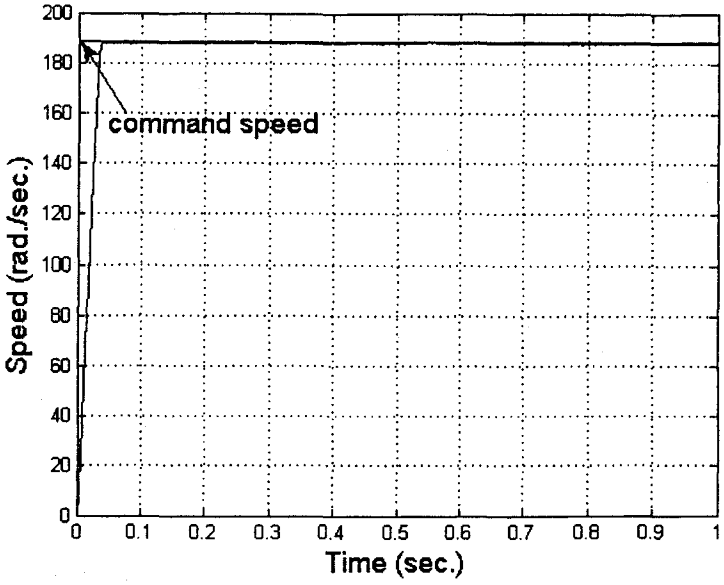


Figure 4.4. Response of ANN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

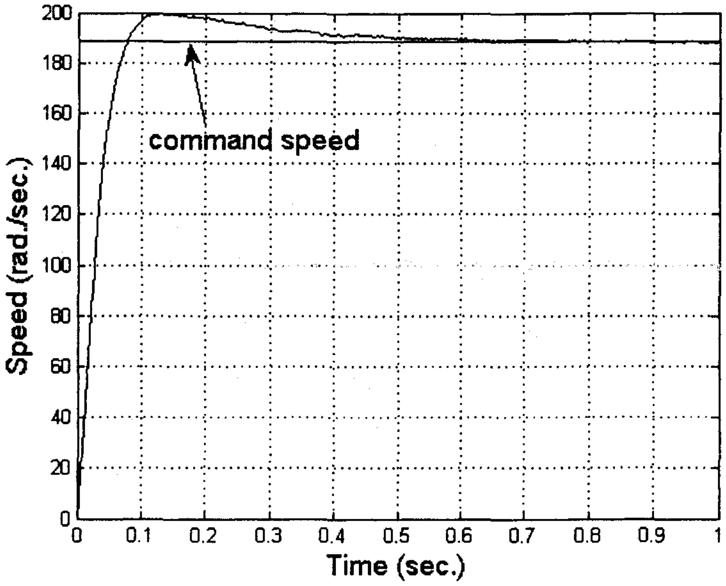


Figure 4.5. Response of PID controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

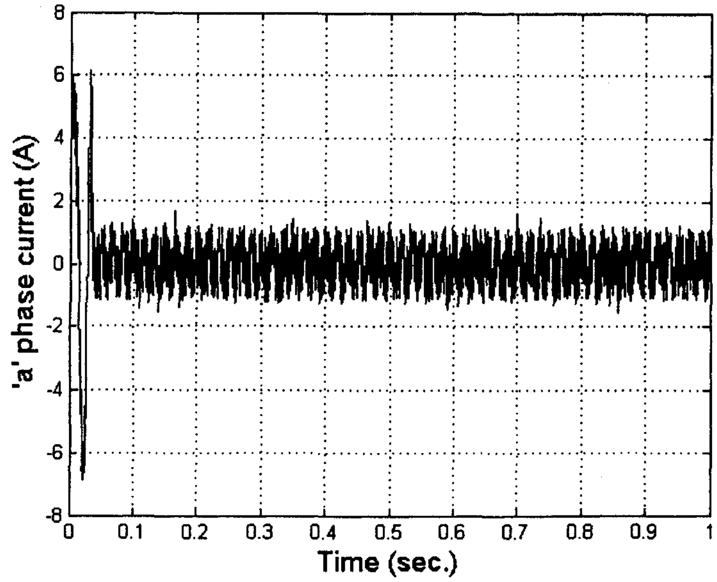


Figure 4.6. Current in phase 'a' of ANN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

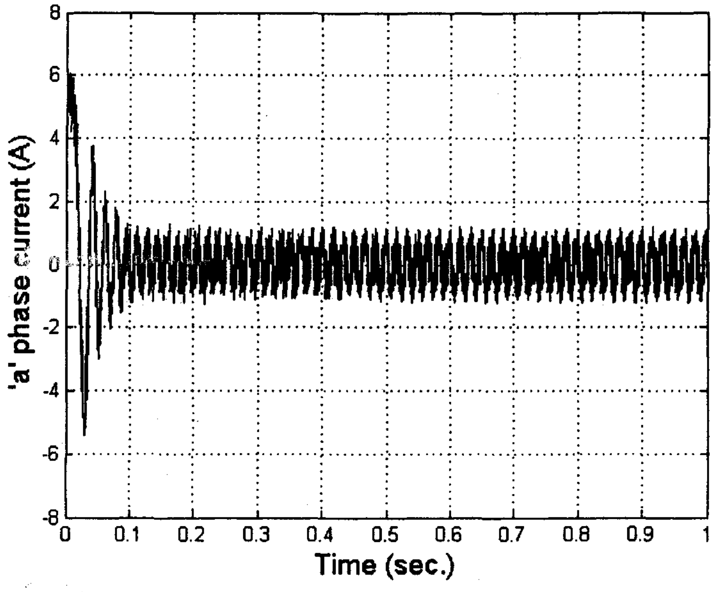


Figure 4.7. Current in phase 'a' of PID controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

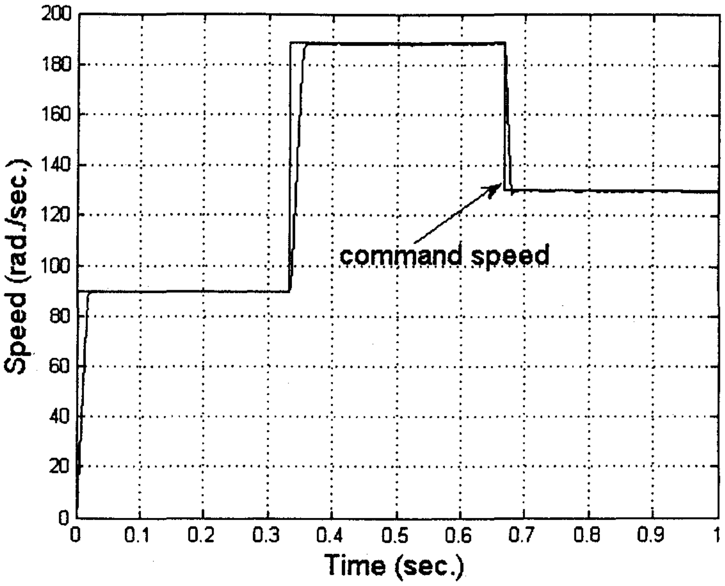


Figure 4.8. Response of the ANN controller-based IPMSM drive to step changes of command speed (90 → 188.5 → 130 rad./sec.).

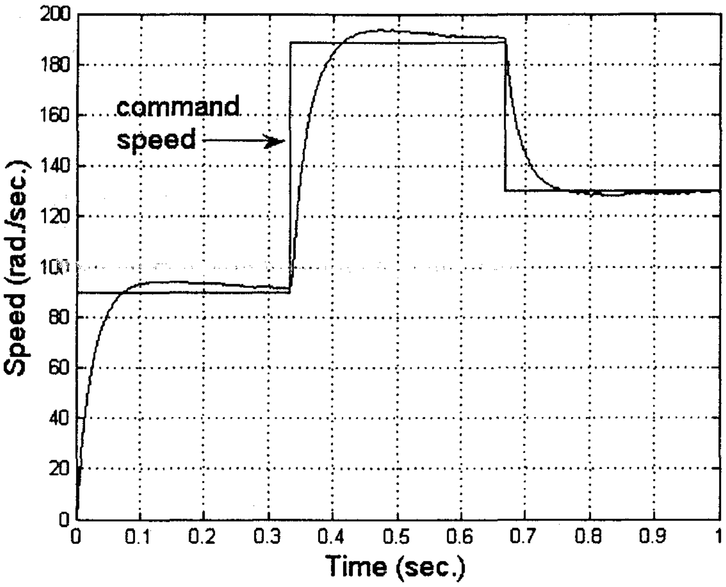


Figure 4.9. Response of the PID controller-based IPMSM drive to step changes of command speed (90 → 188.5 → 130 rad./sec.).

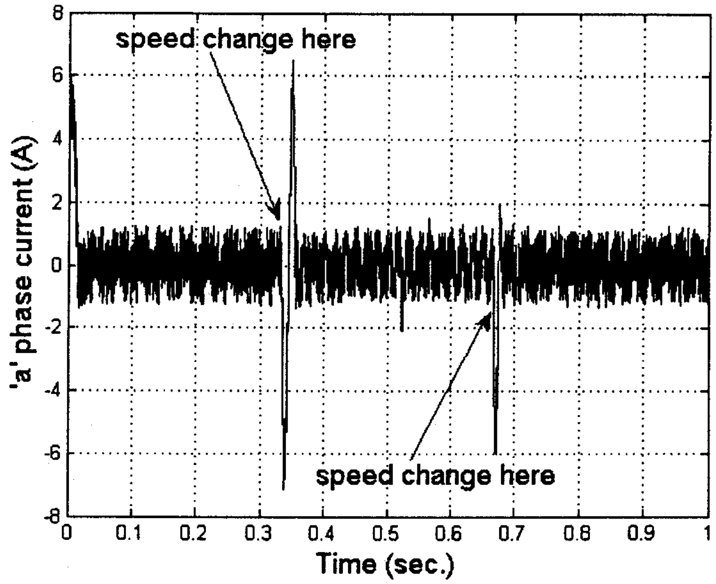


Figure 4.10. Current in phase 'a' of ANN controller-based IPMSM drive under step changes of command speed (90 → 188.5 → 130 rad./sec.).

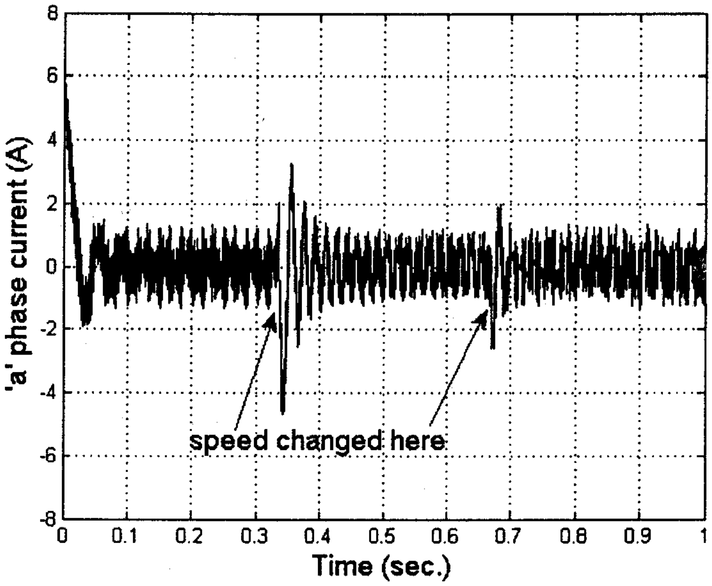


Figure 4.11. Current in phase 'a' of PID controller-based IPMSM drive under step changes of command speed (90 → 188.5 → 130 rad./sec.).

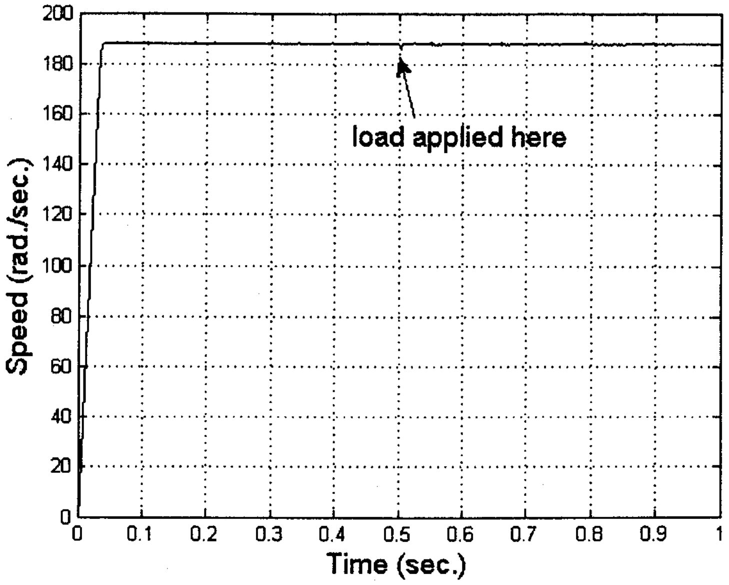


Figure 4.12. Response of ANN controller-based IPMSM to sudden application of full-load (0 → 2 Nm) at a command speed of 188.5 rad./sec.

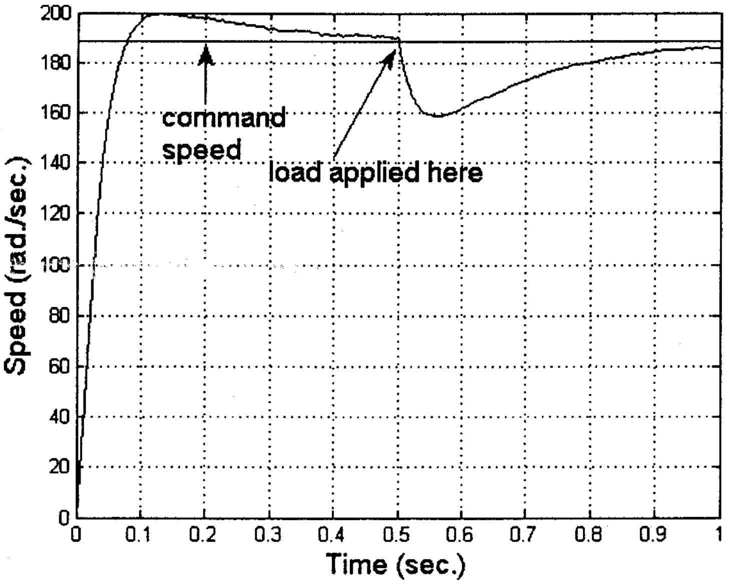


Figure 4.13. Response of PID controller-based IPMSM to sudden application of full-load (0 → 2 Nm) at a command speed of 188.5 rad./sec.

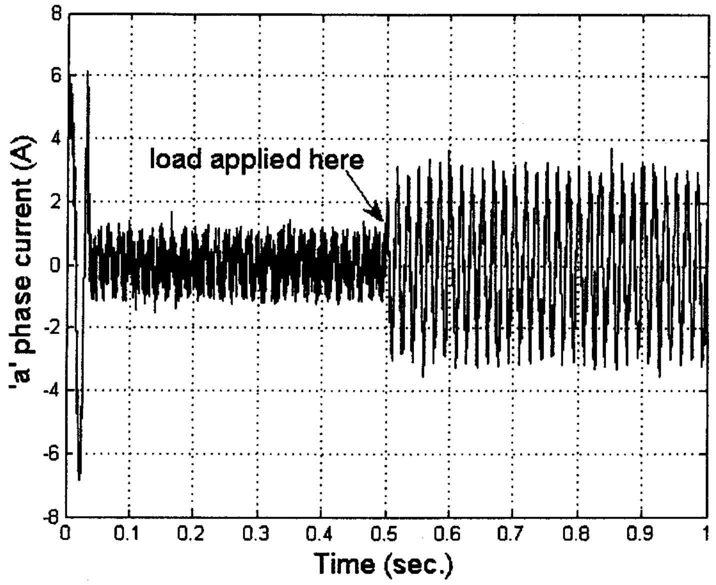


Figure 4.14. Current response in phase 'a' of ANN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

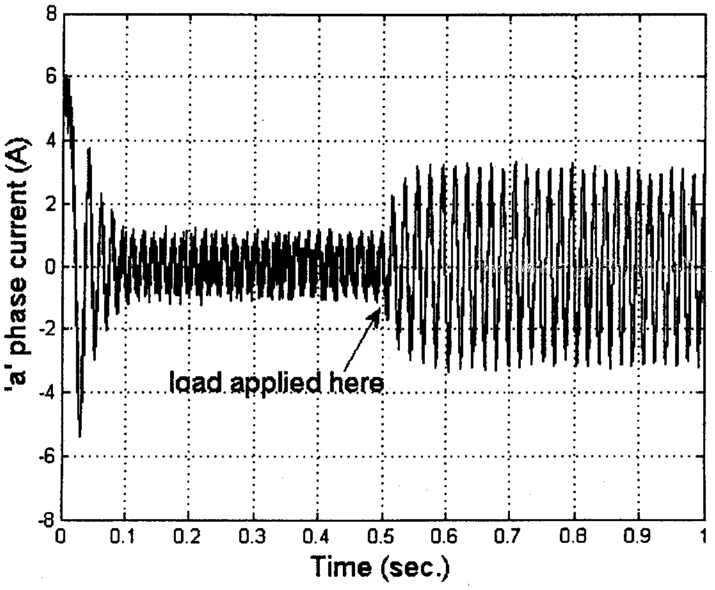


Figure 4.15. Current response in phase 'a' of PID controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

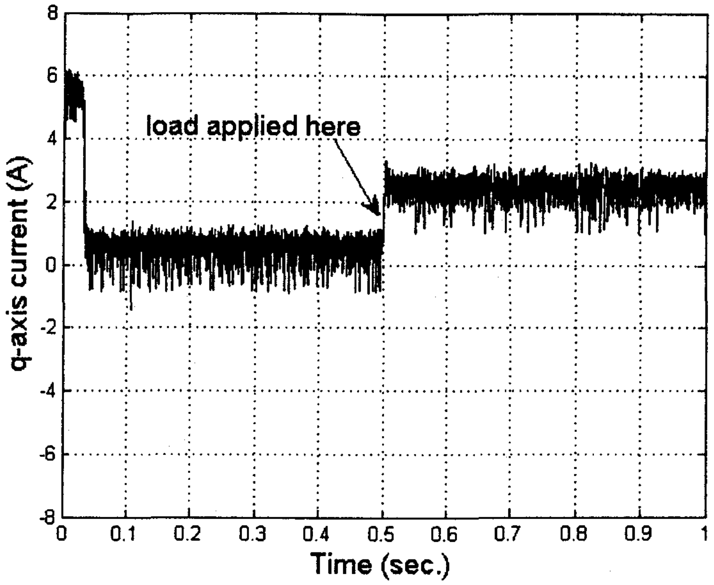


Figure 4.16. Q-axis current response of ANN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

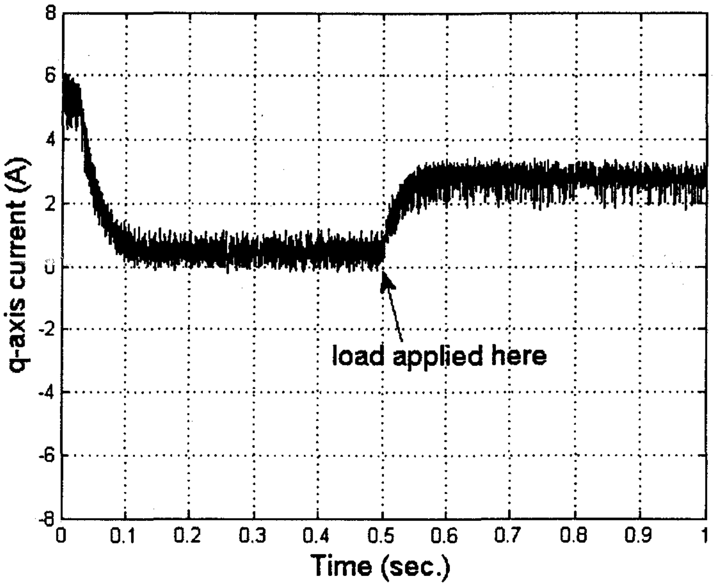


Figure 4.17. Q-axis current response of PID controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

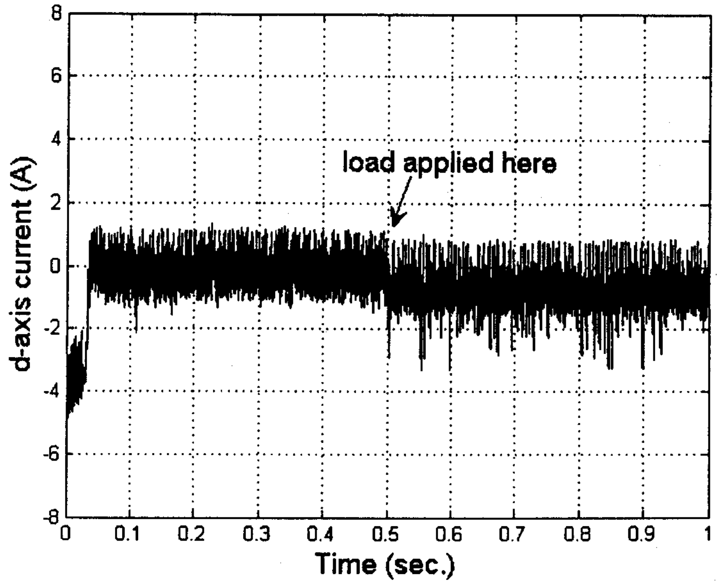


Figure 4.18. D-axis current response of ANN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

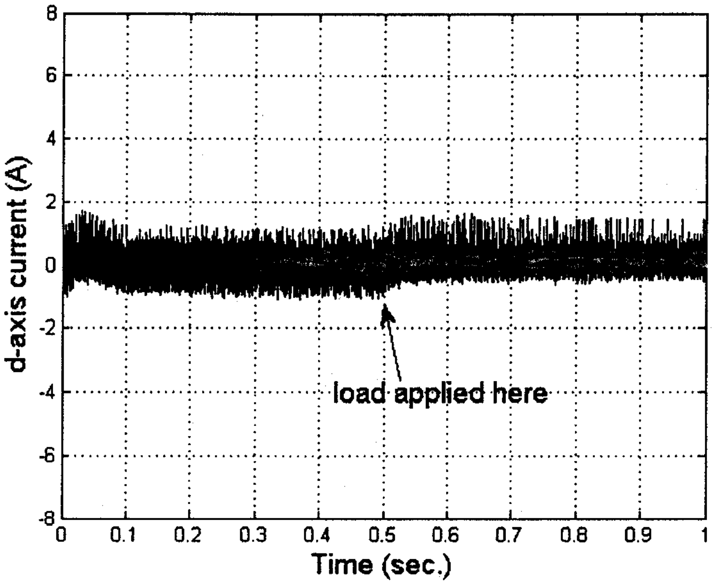


Figure 4.19. D-axis current response of PID controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

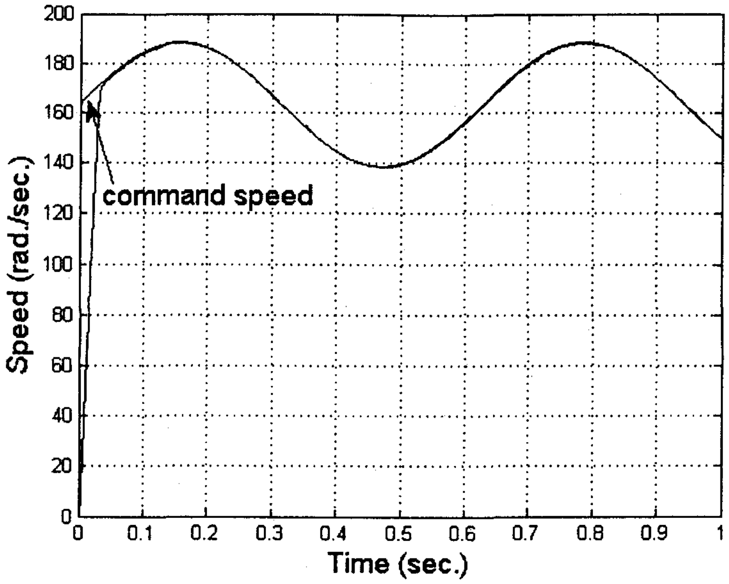


Figure 4.20. Response of ANN controller-based IPMSM drive, under no load, to a varying command speed (sine).

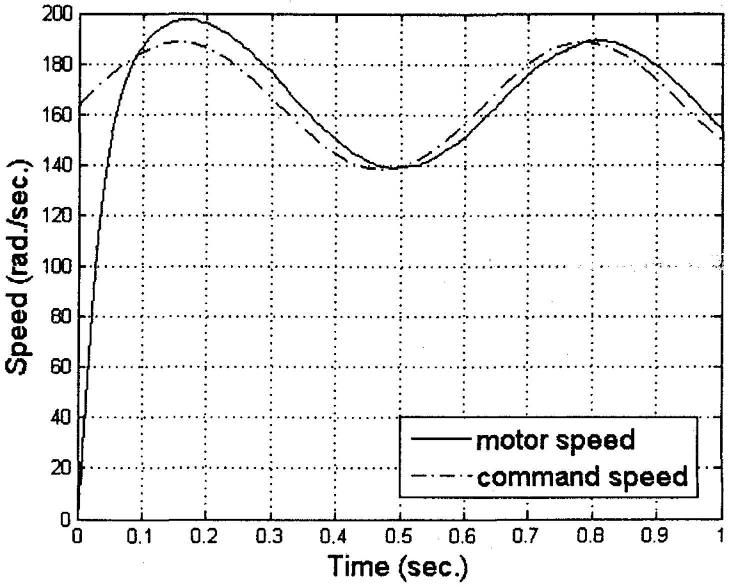


Figure 4.21. Response of PID controller-based IPMSM drive, under no load, to a varying command speed (sine).

4.3.2 Rated Speed and Below Operation with Parameter Variations (MTPA mode)

As discussed in Section 1.4, q-axis inductance varies with phase current and torque angle in the IPMSM, due to the effects of magnetic saturation. The remaining tests in this section investigate the behaviour of the drive when subjected to sudden parameter changes. It is not realistic, or indicative of actual operating conditions, to assume that such changes occur in sudden, step-like manners. However, the following simulated parameter changes are chosen as worse-case scenarios to investigate the limitations of the ANN-based drive. Real-time parameter variations will not be as severe as the cases presented here.

Figure 4.22 shows the response of the ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of q-axis inductance at base speed command (188.5 rad./sec.). The drive shows no speed disturbance due to this L_q parameter change. Figure 4.23 shows the response of the PID controller-based IPMSM drive, under no load, to the same increase of q-axis inductance. The drive does not smoothly track command speed, but it is difficult to determine whether this is due solely to the change in L_q.

Figure 4.24 shows the response of the ANN controller-based IPMSM drive, under no load, to a sudden doubling of stator resistance, R_s, at base speed command (188.5 rad./sec.) The drive shows no speed disturbance due to this R_s parameter change. Figure 4.25 shows the response of the PID controller-based IPMSM drive, under no load, to the same increase of stator resistance. The drive does not smoothly track command speed, but again it is difficult to determine whether this is due solely to the change in R_s. Comparison with Figure 4.23, however, shows that the PID-based drive is affected by these parameter changes, as differences in the two speed responses can be readily seen. The PID-based drive does seem, though, tolerable to these disturbances under no-load conditions.

Figure 4.26 shows the speed response of the ANN controller-based IPMSM drive to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2 Nm) at base speed command (188.5 rad./sec.). The drive appears minimally affected by these changes. Figure 4.27 shows the speed response of the PID controller-based IPMSM drive to the same parameter variations and full-load application. The drive clearly cannot maintain command speed tracking under these conditions.

Figures 4.28 (a) and (b) show the 'a' phase current response of the ANN controller-based IPMSM drive to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2 Nm) at base speed command (188.5 rad./sec.). Phase current increases to a peak value of roughly 3 A. Fluctuations in phase current due to the discrete nature of sampling and the hysteresis current controller can be clearly seen. Figures 4.29 (a) and (b) show the 'a' phase current response of the PID controller-based IPMSM drive to the same parameter variations and full-load application. The drive cannot track command speed under these conditions and phase current increases to in excess of 4 A. Again, fluctuations due to sampling and the hysteresis current controller are clearly visibile.

Figure 4.30 shows the 'a' phase command current of the ANN controller-based IPMSM drive with a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm) at base speed command (188.5 rad./sec.). The command phase current is clearly seen to be greater than the actual phase current shown in Figure 4.28. Figure 4.31 shows the 'a' phase command current of the PID controller-based IPMSM drive to the same parameter variations and full-load application. With the PID controller, the command phase current is much more sinusoidal in nature. This is due to the fact that the ANN-based controller uses a reference torque which is calculated from the motor currents themselves, and also that the torque command from the ANN-based controller may be modified quite reactively at each iteration.

Figure 4.32 shows the speed response of the ANN controller-based IPMSM drive, under a non-linear load of $T_L = \omega^2/17766.125$, to a sinusoidal command speed with an amplitude of 25 rad./sec. and 163.5 rad./sec. offset. At 0.5 seconds, L_q is increased by 50% and R_s doubled. The drive experiences no difficulty in tracking command speed under these conditions. Figure 4.33 shows the speed response of the PID controller-based IPMSM drive under these same conditions. The drive cannot track command speed.

Figure 4.34 shows the non-linear load of $T_L = \omega^2/17766.125$ applied to the ANN controller-based IPMSM drive, under sinusoidal speed command (amplitude of 25 rad./sec. and 163.5 rad./sec. offset) and sudden 50% increase of L_q and doubling of R_s. This is the load corresponding to the speed response of Figure 4.32. Figure 4.35 shows the current response in phase 'a' of the ANN controller-based IPMSM drive due to this load and sinusoidal speed command (amplitude of 25 rad./sec. and 163.5 rad./sec. offset). It can be seen that the amplitude of the phase current increases as load increases; it can also be seen that phase current frequency increases as motor speed increases. Figures 4.36 and 4.37 show the q- and d-axis current responses of the ANN controller-based IPMSM drive, respectively, under these same conditions. It can be seen that the q-axis current increases in the positive direction as load increases and the d-axis current experiences a corresponding increase in the negative direction.

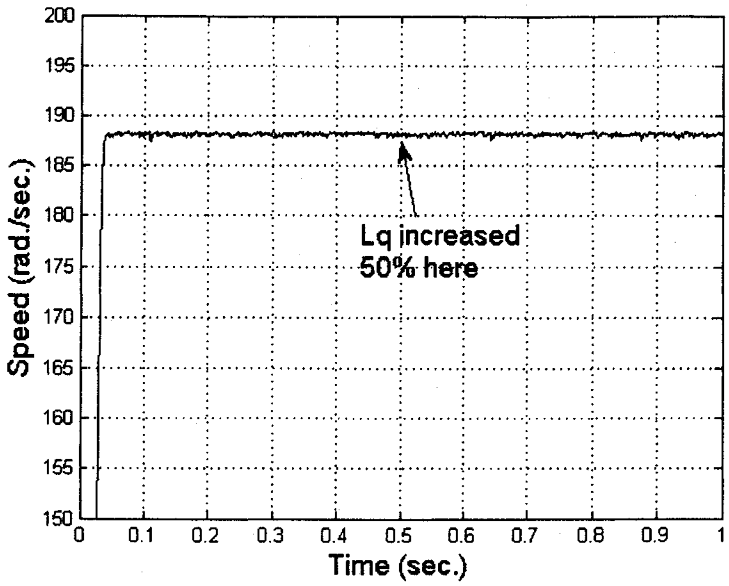


Figure 4.22. Response of ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

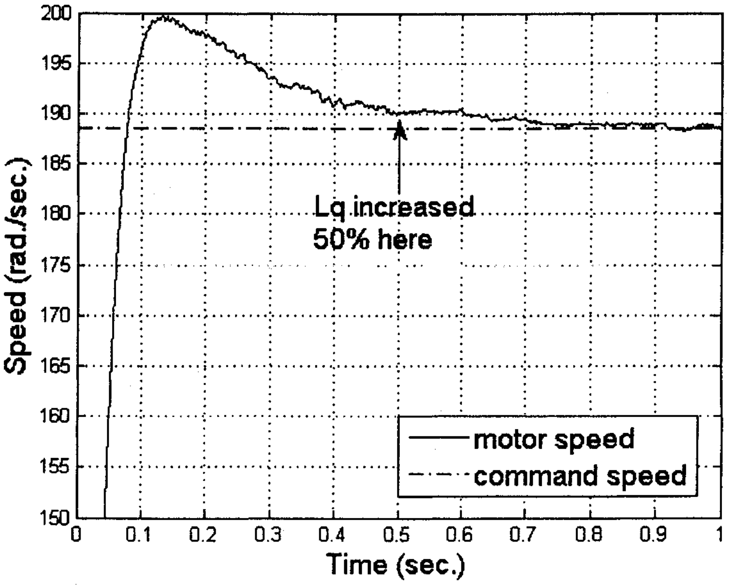


Figure 4.23. Response of PID controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

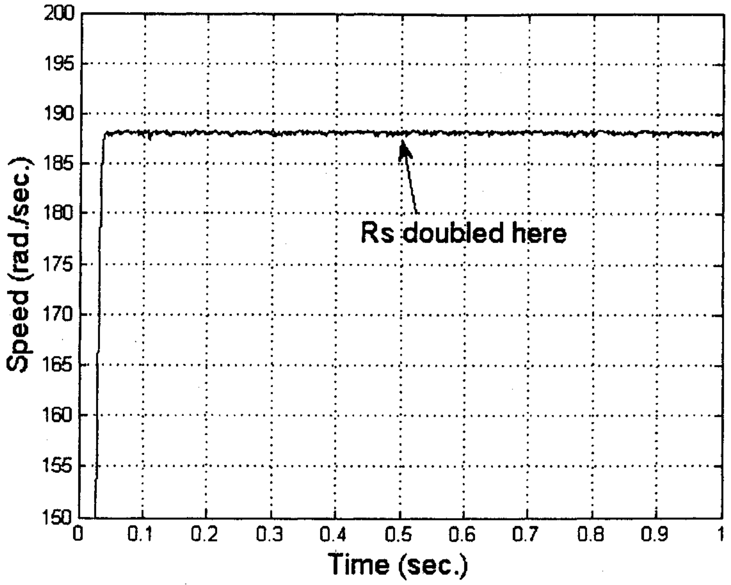


Figure 4.24. Response of ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

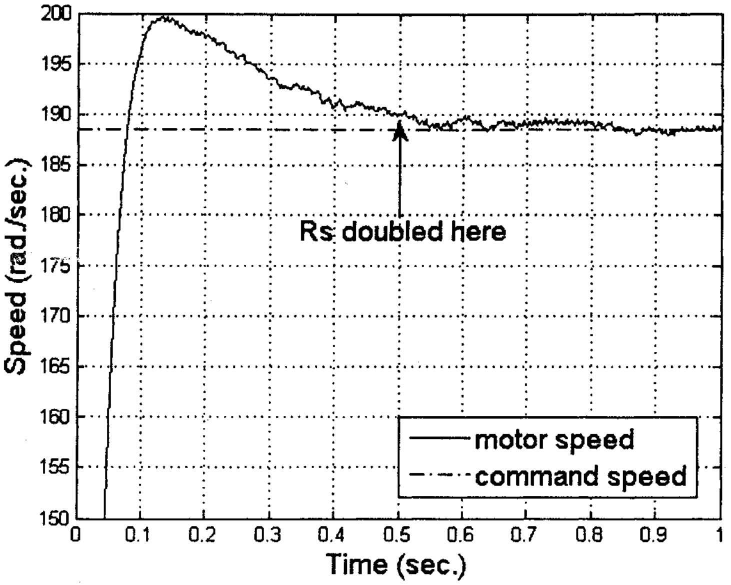


Figure 4.25. Response of PID controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

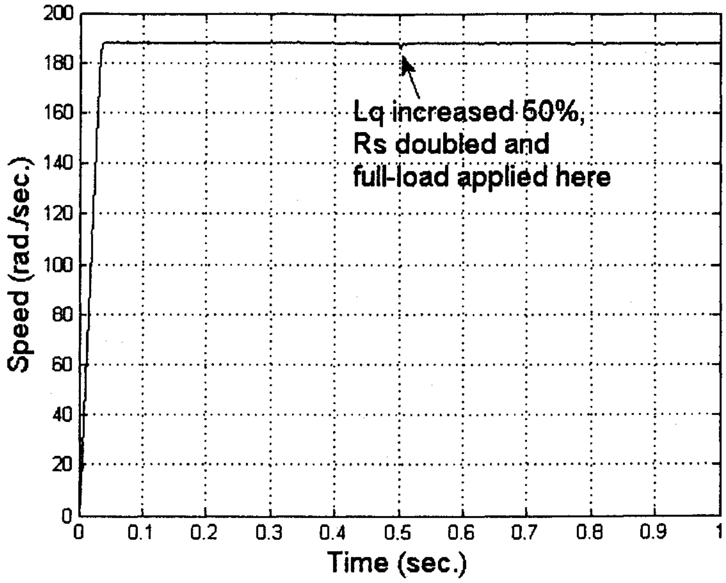


Figure 4.26. Response of ANN controller-based IPMSM drive to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2 Nm).

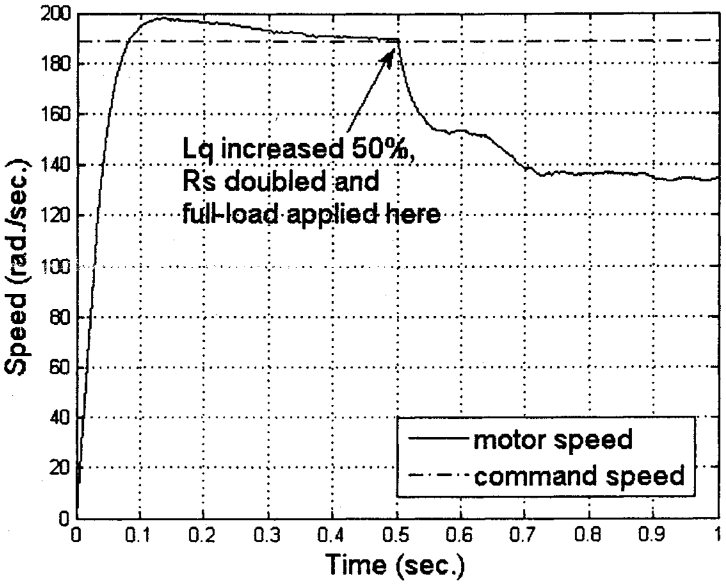
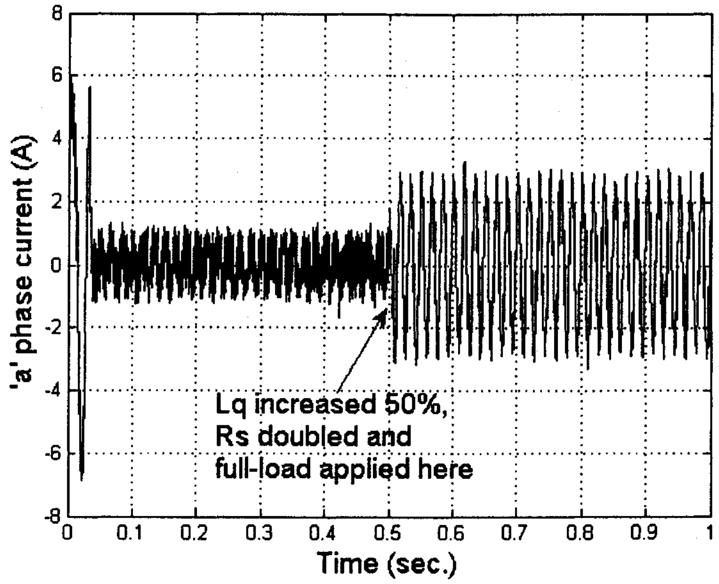
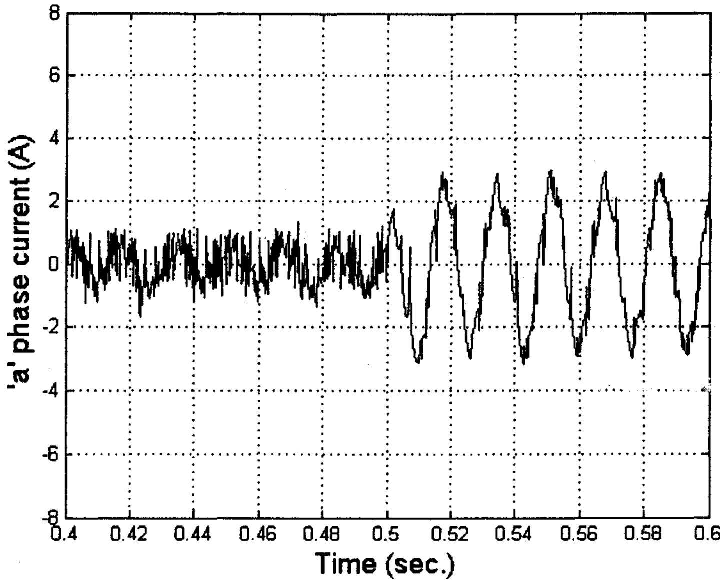


Figure 4.27. Response of PID controller-based IPMSM drive to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2 Nm).

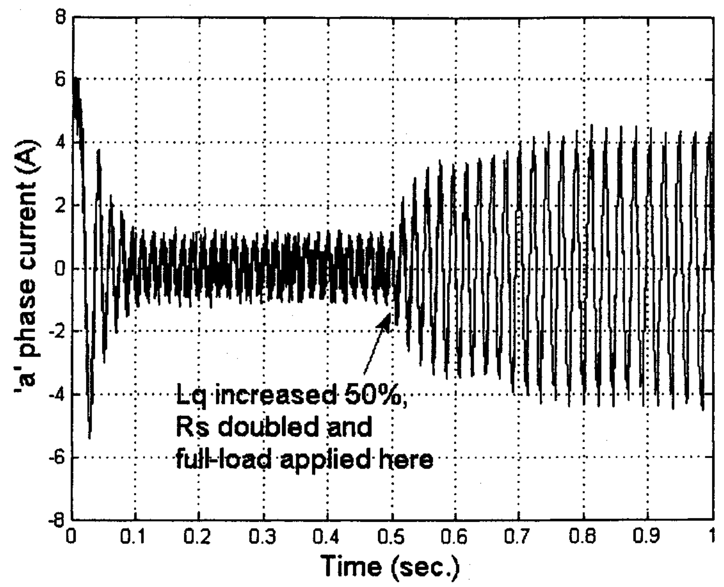


(a)

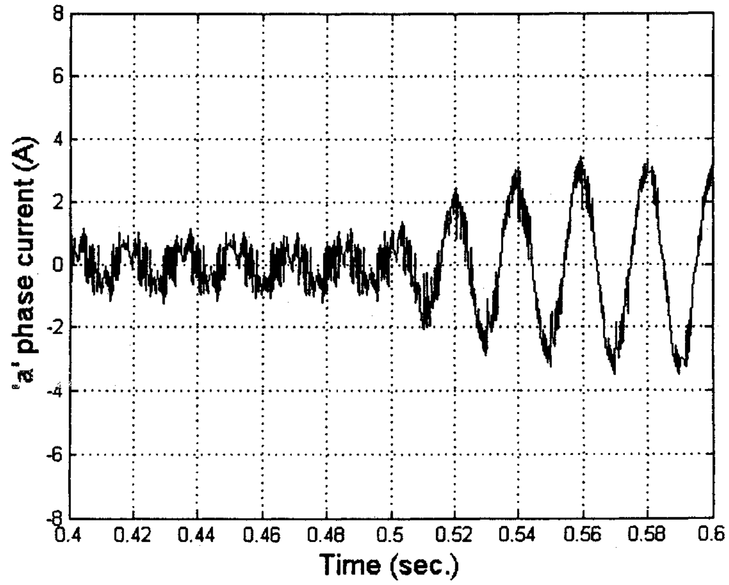


(b)

Figure 4.28 (a) and (b). Current response in phase 'a' of ANN controller-based IPMSM drive at base speed command to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm).



(a)



(b)

Figure 4.29 (a) and (b). Current response in phase 'a' of PID controller-based IPMSM drive at base speed command to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm).

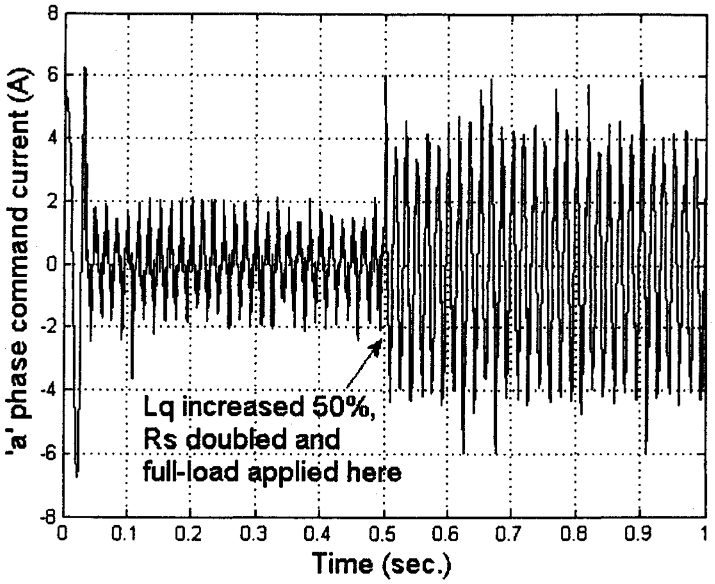


Figure 4.30. 'a' phase command current of ANN controller-based IPMSM drive at base speed command with a sudden 50% increase of L_q, doubling of R_s and application of full-load.

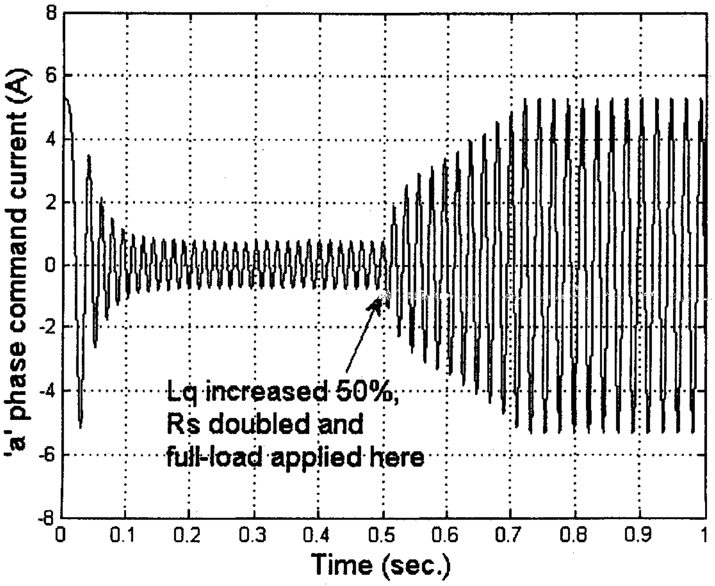


Figure 4.31. 'a' phase command current of PID controller-based IPMSM drive at base speed command with a sudden 50% increase of L_q, doubling of R_s and application of full-load.

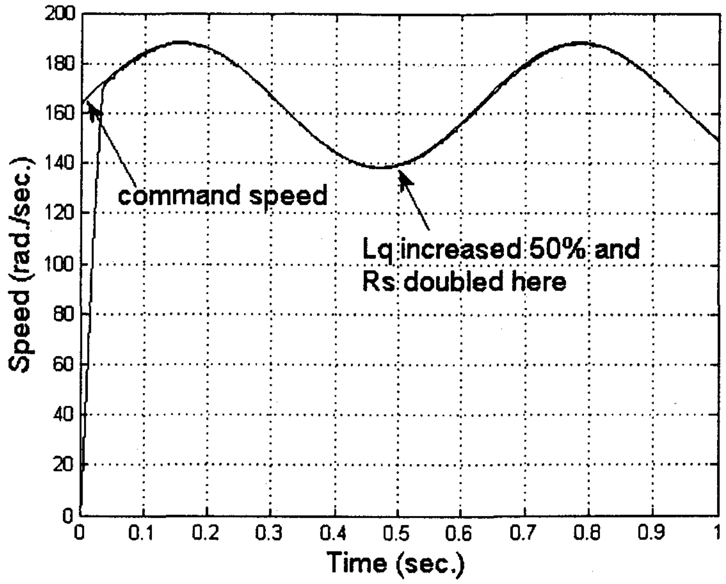


Figure 4.32. Response of ANN controller-based IPMSM drive, under non-linear load, to sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

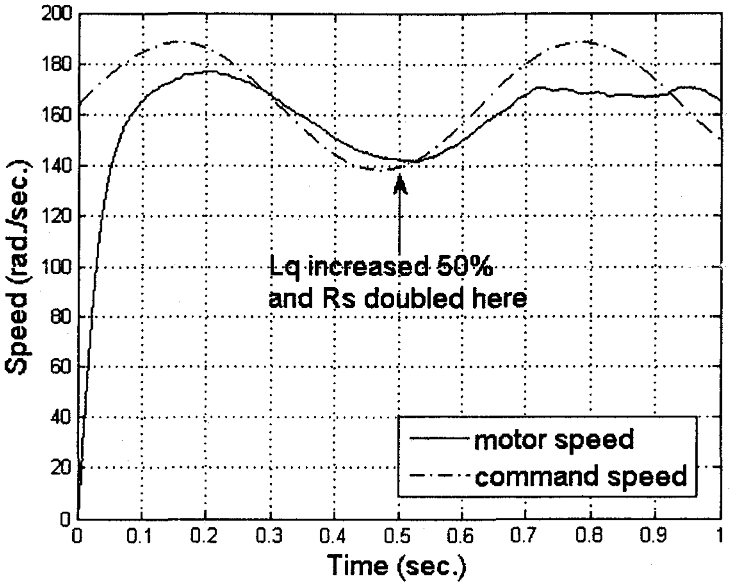


Figure 4.33. Response of PID controller-based IPMSM drive, under non-linear load, to sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

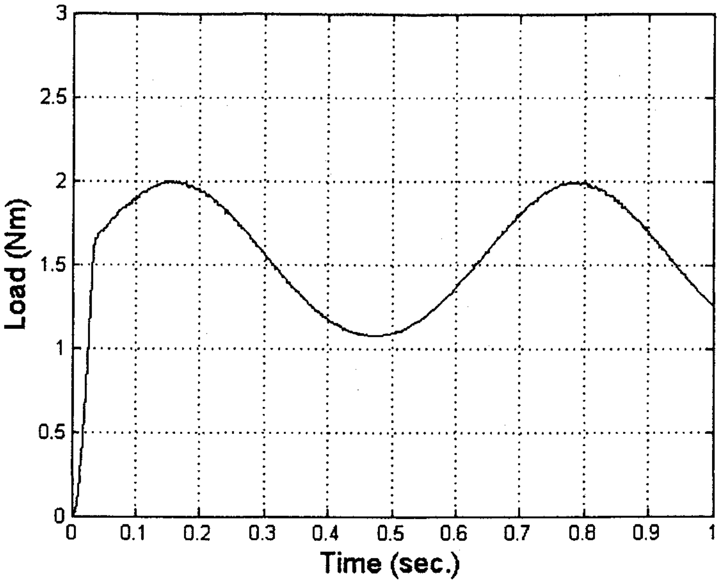


Figure 4.34. Non-linear load applied to ANN controller-based IPMSM drive under sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

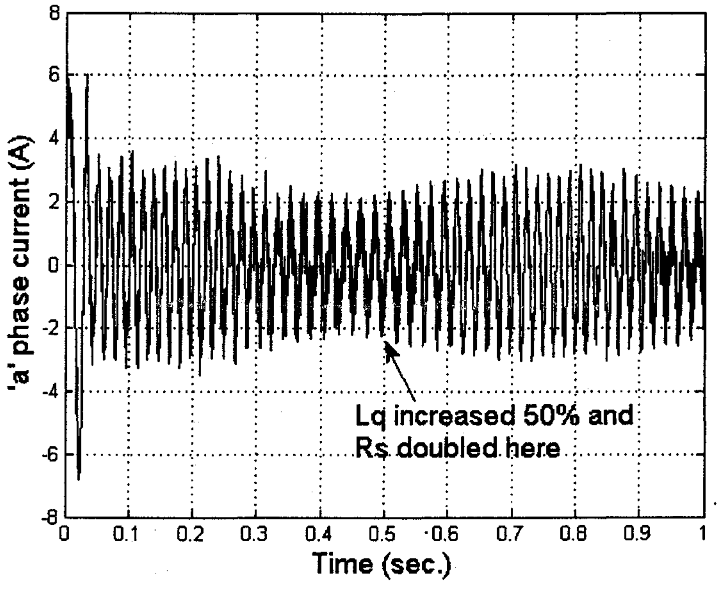


Figure 4.35. Current response in phase 'a' of ANN controller-based IPMSM drive under non-linear load to sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

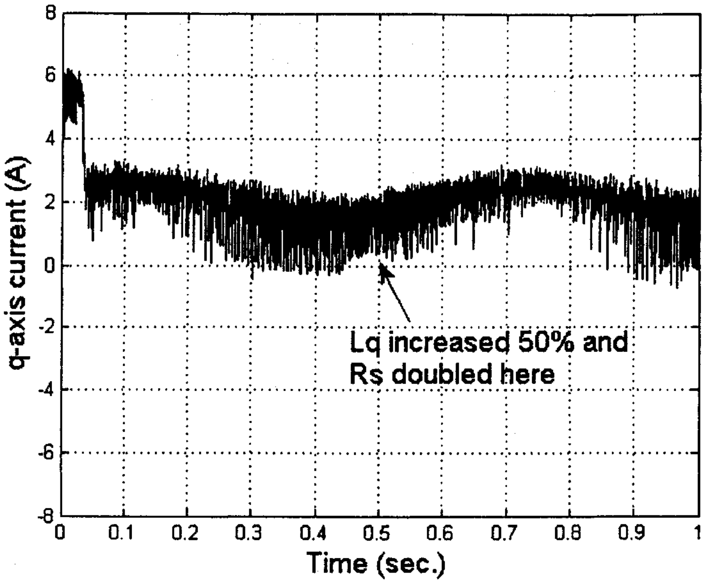


Figure 4.36. Q-axis current response of ANN-based IPMSM drive under non-linear load to sinusoidal speed command, 50% increase of L_q and doubling of R_s.

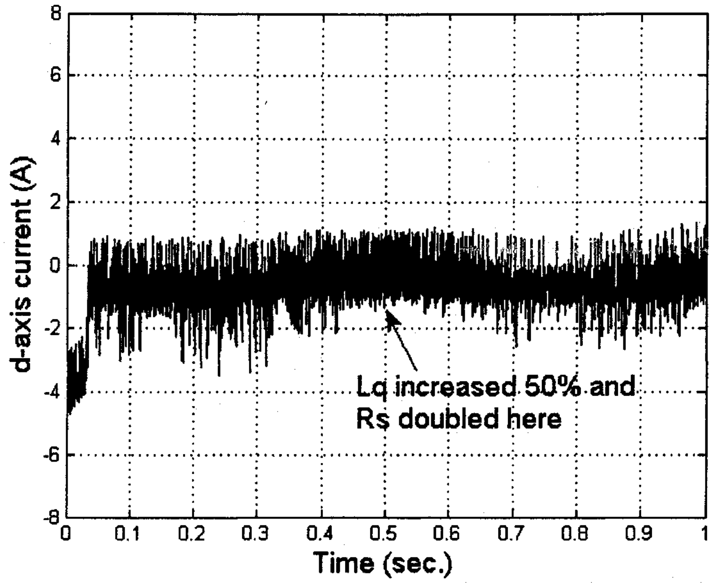


Figure 4.37. D-axis current response of ANN-based IPMSM drive under non-linear load to sinusoidal speed command, 50% increase of L_q and doubling of R_s.

4.3.3 Above Rated Speed Operation (with FW)

Figure 4.38 shows the startup response of the ANN controller-based IPMSM drive under no load for a command speed of 250 rad./sec, utilizing the flux-weakening mode of operation. In this case, V_o, in the flux-weakening (FW) Equations (2.58) and (2.59), has been chosen to yield a phase current of 3 A at 250 rad./sec. The PID controller-based drive is unable to safely operate above base speed because the $i_d^r = 0$ approximation does not allow for FW mode operation. The ANN controller-based drive quickly and accurately reaches and tracks command speed.

Figure 4.39 shows the response of the ANN controller-based IPMSM drive to a step change of command speed (from 188.5 to 250 rad./sec.) at no load. The drive accurately and precisely follows these changes. Figure 4.40 shows the 'a' phase current response of the ANN controller-based IPMSM drive under these same conditions. At no-load and command speed of 188.5 rad./sec. the FW-based ANN-controlled IPMSM drive is seen to experience a slightly higher 'a' phase current than when the MTPA scheme is used under these same conditions. This is to be expected due to the very nature of the MTPA scheme. Again, phase current increases to approximately 3 A at 250 rad./sec. Figures 4.41 and 4.42 show the corresponding q- and d-axis current responses of the ANN drive under the same conditions. Figure 4.42 clearly shows the increase in negative d-axis current as command speed is increased beyond base speed. Q-axis current spikes to generate the necessary torque as the motor is accelerated.

Figure 4.43 shows the startup speed response of the ANN controller-based IPMSM drive to a sinusoidal speed command with an amplitude of 25 rad./sec. and 225 rad./sec. offset. The drive very quickly and accurately reaches and tracks command speed. Figure 4.44 shows the current response in phase ‘a’ of the ANN controller-based IPMSM drive under these same conditions. The increase of phase current magnitude can be seen with increased command speed.

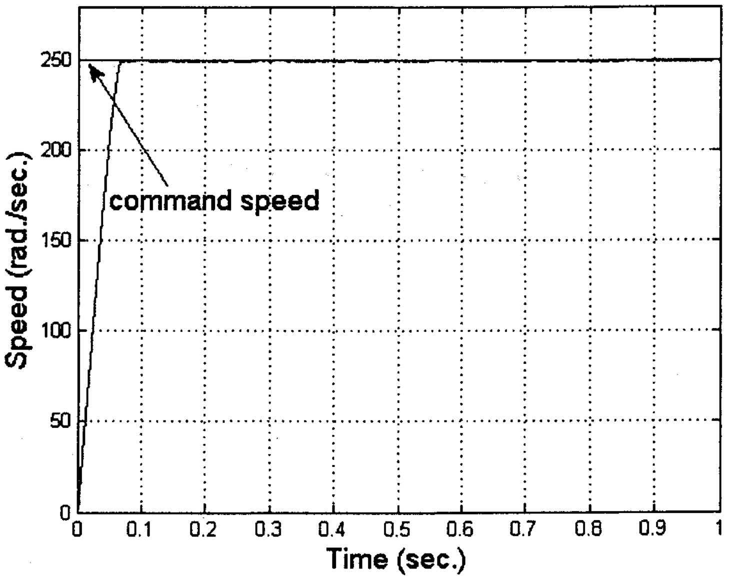


Figure 4.38. Response of ANN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

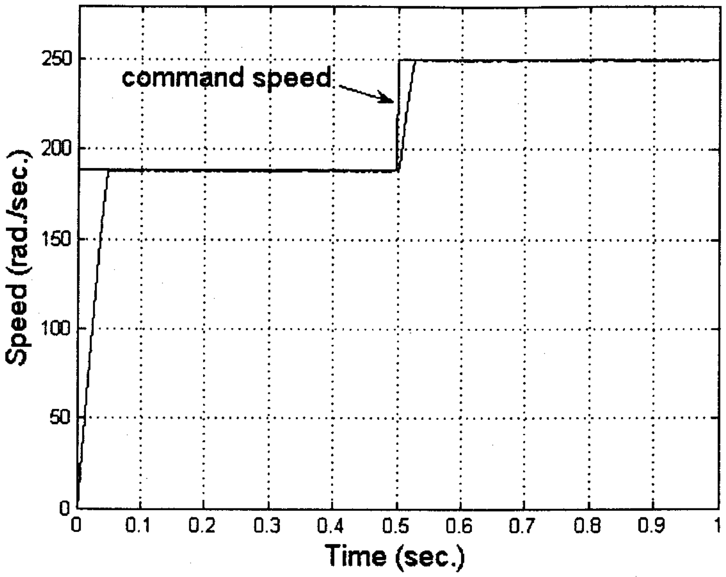


Figure 4.39. Response of the ANN controller-based IPMSM drive to a step change of command speed (188.5 → 250 rad./sec.).

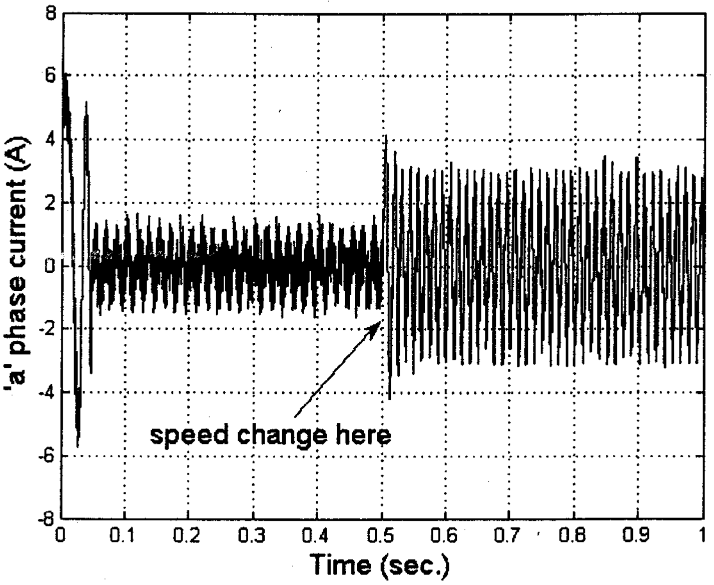


Figure 4.40. Current in phase 'a' of ANN controller-based IPMSM drive under step change of command speed (188.5 → 250 rad./sec.).

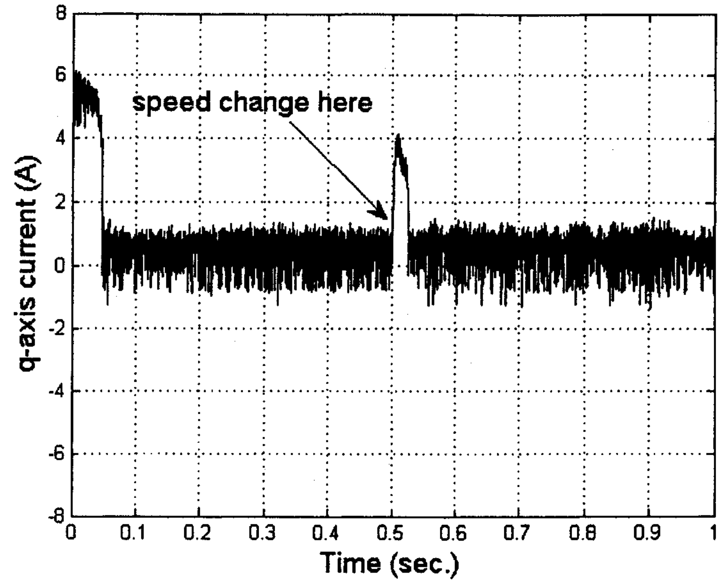


Figure 4.41. Q-axis current response of ANN controller-based IPMSM drive to sudden step change of speed command (188.5 → 250 rad./sec.).

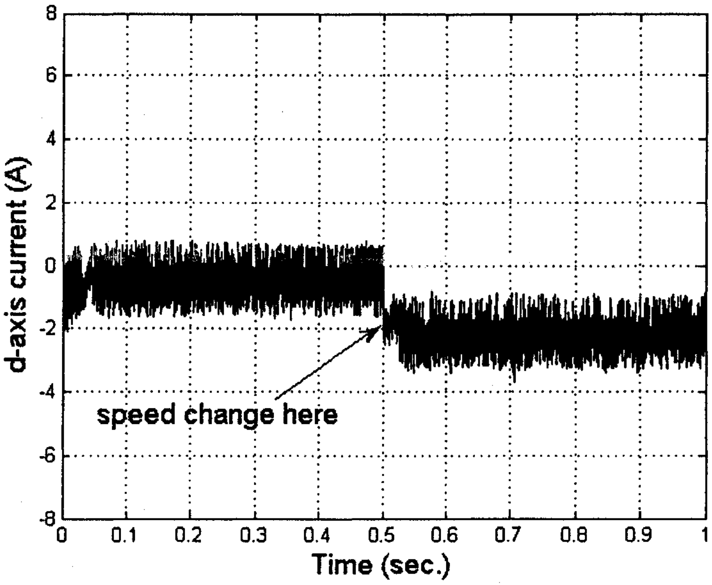


Figure 4.42. D-axis current response of ANN controller-based IPMSM drive to sudden step change of speed command (188.5 → 250 rad./sec.).

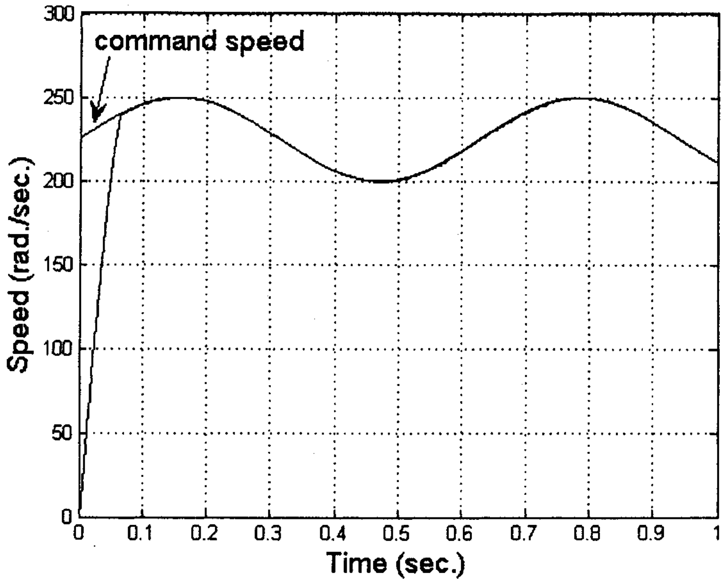


Figure 4.43. Response of ANN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

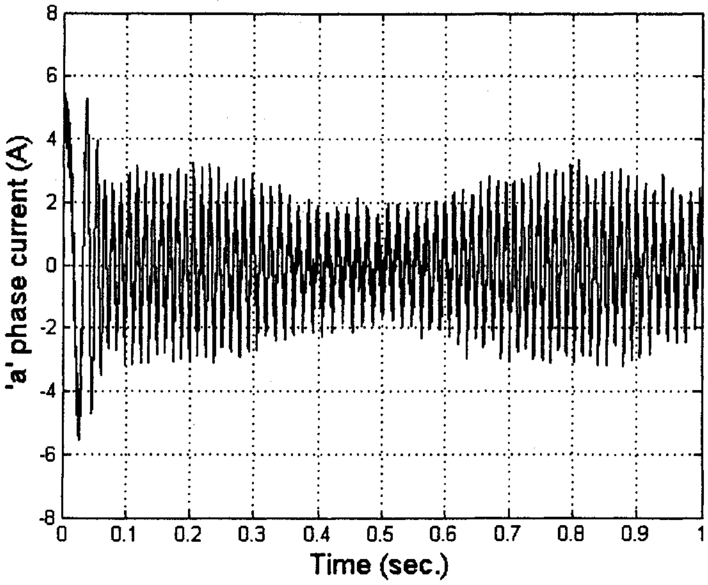


Figure 4.44. Current response in phase 'a' of ANN controller-based IPMSM drive, under no-load, to sinusoidal speed command above base speed.

4.3.4 Above Rated Speed Operation with Parameter Variations (FW mode)

Worst-case scenario parameter changes when operating above base speed, in FW mode, are investigated in this section.

Figure 4.45 shows the speed response of the ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec. Any disruption to motor speed is indiscernible. Figure 4.46 shows the phase current response of the drive under these same conditions. Again, the disruption appears to have no effect. Figures 4.47 and 4.48 show the corresponding q- and d-axis current responses, respectively. Here it can be seen that the change in L_q does affect d-q axis currents, but the effects are not visibly reflected in phase current.

Figure 4.49 shows the speed response of the ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec. Again, disruption to motor speed is indiscernible. Figure 4.50 shows the phase current response of the drive under these same conditions. Again, the disruption appears to have no effect. Figures 4.51 and 4.52 show the corresponding q- and d-axis current responses, respectively. It is difficult to visually ascertain whether the change of stator resistance has an identifiable affect on these currents.

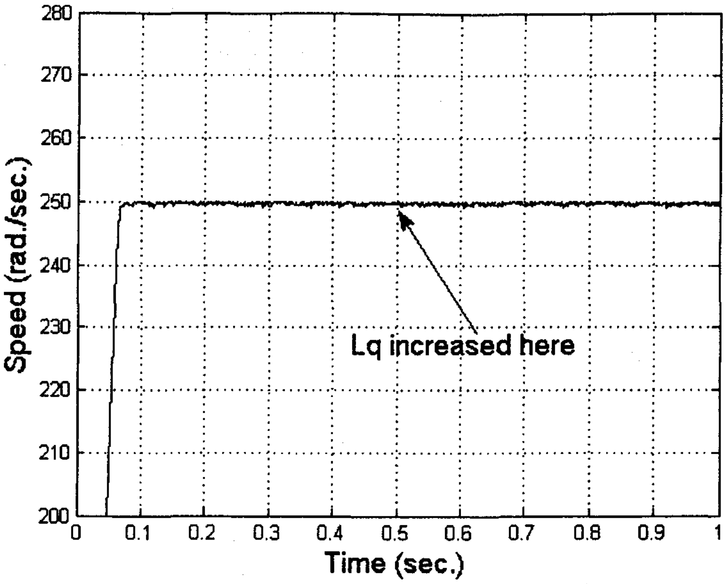


Figure 4.45. Response of ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

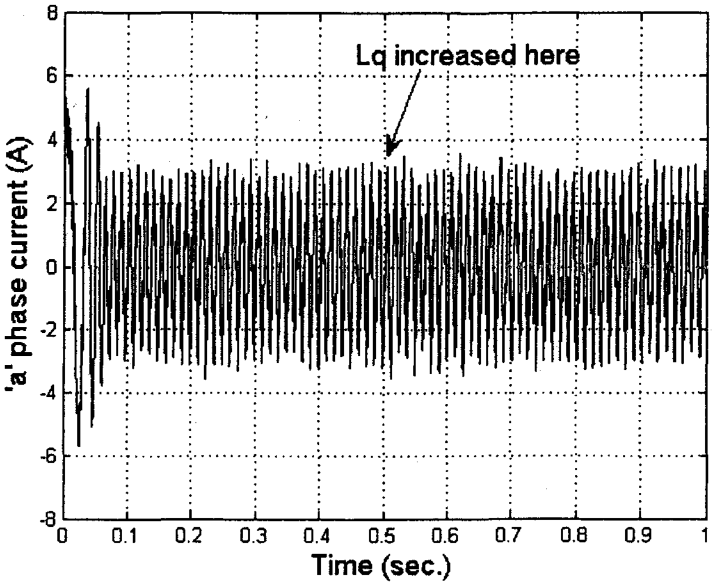


Figure 4.46. Current response in phase 'a' of ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

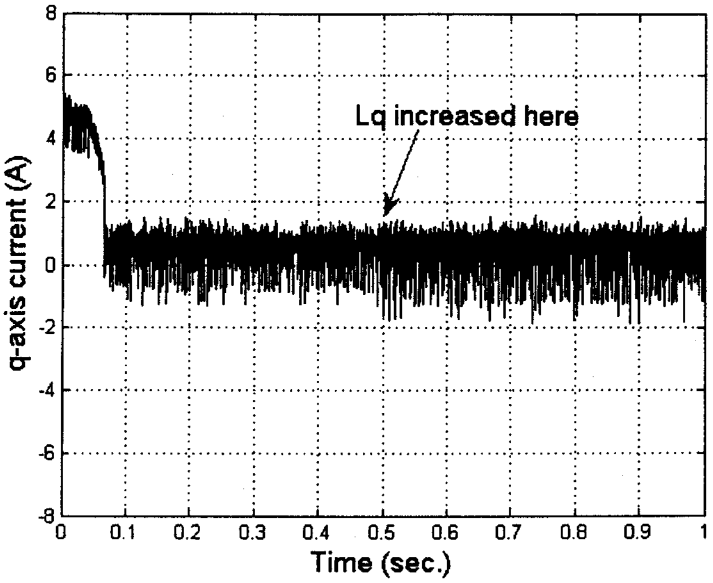


Figure 4.47. Q-axis current of ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

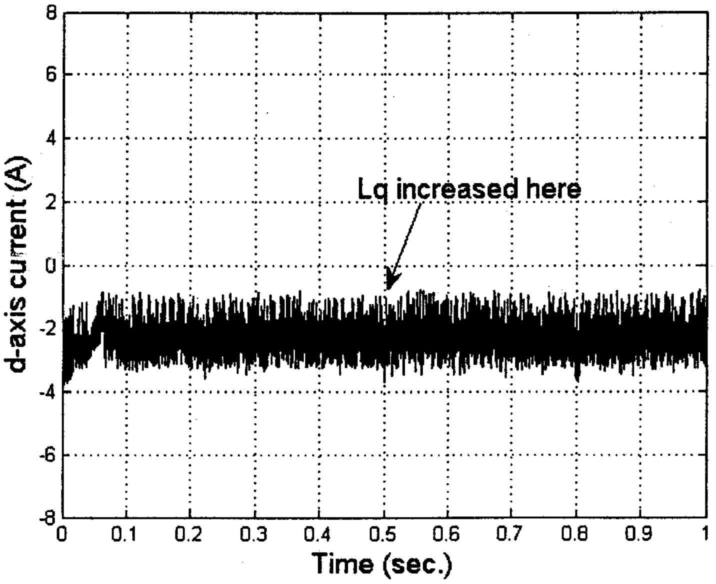


Figure 4.48. D-axis current of ANN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

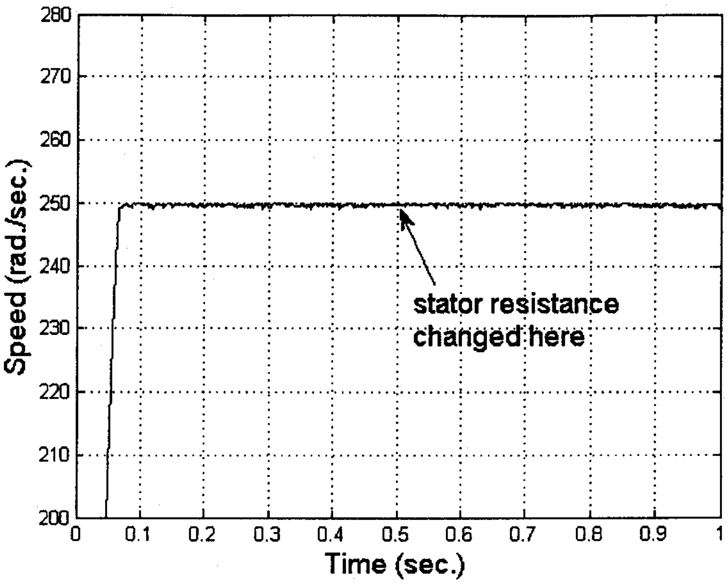


Figure 4.49. Response of ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

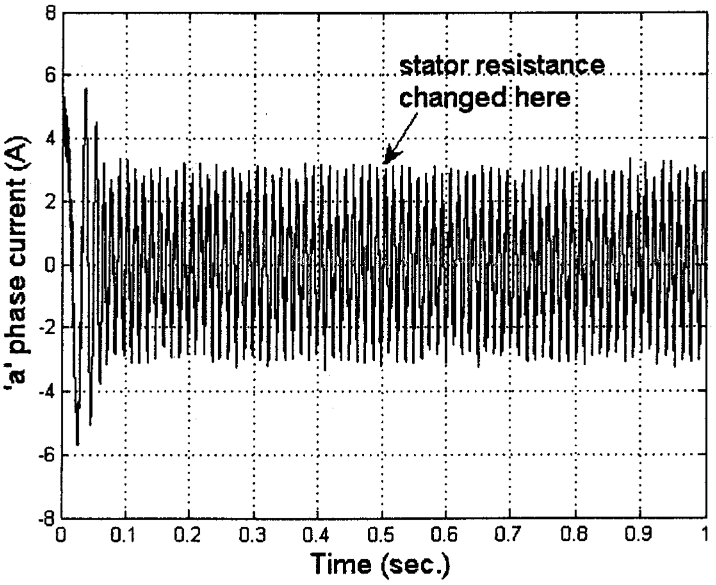


Figure 4.50. Current response in phase 'a' of ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

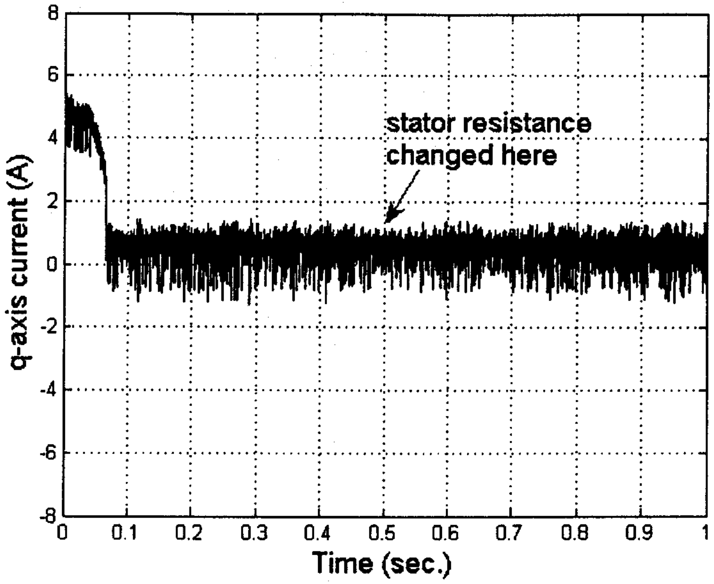


Figure 4.51. Q-axis current of ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

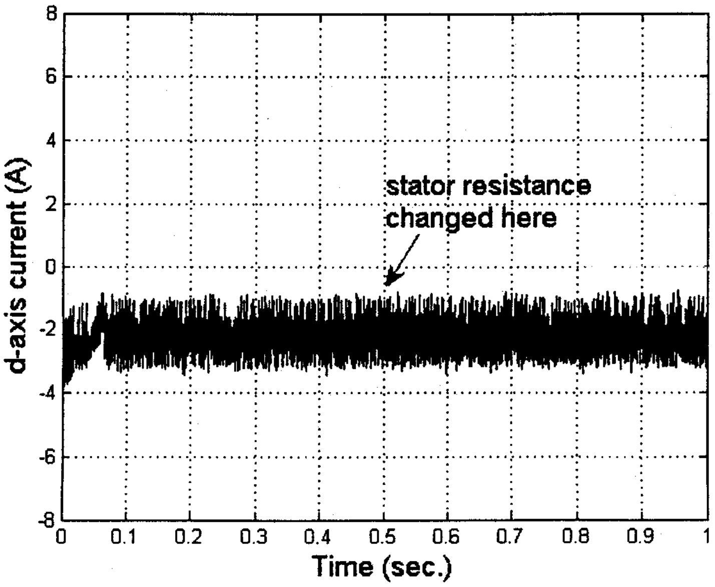


Figure 4.52. D-axis current of ANN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

4.4 Summary

From the simulated results, it is seen that the IPMSM drive can be successfully implemented with an ANN-based speed controller that trains itself adaptively in real-time based on the scheme covered both in this chapter and Chapter 3. This controller displays superior performance to a PID-based speed controller, and is resistant to disturbances due to both load applications and parameter variations. The drive is capable of accurately tracking non-linear speed commands and maintaining command speed tracking under non-linear loading.

In summary, fast and accurate speed control is possible at and below base speed via the MTPA scheme. Above base speed, the ANN-based controller can successfully utilize the FW mode of operation.

Chapter 5

Single Artificial Neuron Based Speed Controller for IPMSM Drive

5.1 Introduction

This chapter presents a simplified version of the ANN-based speed controller used in the IPMSM drive, as presented in Chapter 4. The same online back propagation training algorithm based on speed error is used, as well as online back propagation training based on torque error. Again, the MTPA mode is used for base-speed and below base-speed operation, and the FW mode is used for above base-speed operation. Thus, the drive can stably track any arbitrary reference speed under unknown load dynamics and parameter variations. The drive is illustrated in Figure 5.1.

In all respects, the drive system developed in this chapter is identical to that of the ANN-based drive presented in Chapter 4, with the exception that the speed

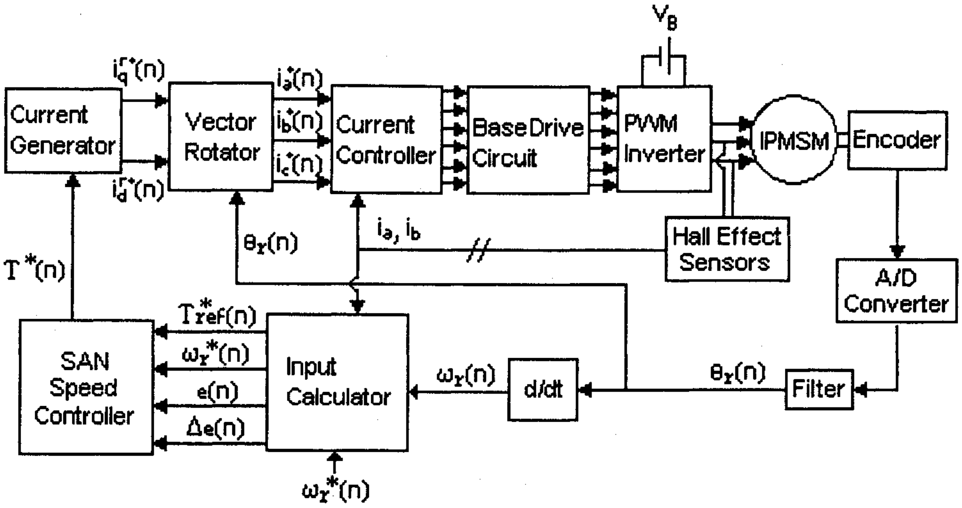


Figure 5.1 SAN-based IPMSM drive.

controller utilizes an artificial neural "network" consisting of only a single artificial neuron (SAN) rather than a network of four neurons. Single neuron theory has been applied successfully in fields such as communications, radar, sonar, biomechanical engineering and controls [78].

The simulation results presented in this chapter indicate that the SAN-based system achieves a performance standard comparable to the more complex ANN-based system.

5.2 SAN-based Speed Controller for IPMSM Drive

The Feed Forward Neural Network (FFNN) structure, as developed in Chapters 3 and 4, is again employed. The network consists of one input layer and one output layer, with the output layer containing a single neuron. This SAN is illustrated in Figure 5.2.

Inputs to the SAN are, again, present sample of command speed $\omega^*(n)$, present sample of speed error $e(n) = \omega^*(n) - \omega(n)$ and present sample of change of speed error $\Delta e(n) = \Delta\omega(n) - \Delta\omega(n-1)$. The weights applied to these inputs are represented as w1, w2 and w3. The bias added to the neuron's summed input is represented as B. Output is the present sample of command torque $T_{ANN}(n)$. As explained in Chapter 3, Section 3.4.6, such a network can be likened to a very precise and adaptive PID controller.

Q- and d-axis command currents, i_q^* and i_d^*, are then determined from T_{ANN} via either the MTPA or FW algorithm. They are then converted to command phase currents and applied to the stator terminals as described in Chapters 2, 3 and 4.

As with the ANN of Chapter 4, the weights and biases of the SAN are updated, by back propagation, whenever the speed error exceeds a threshold of 0.1 rad./sec. If this threshold is not exceeded the weights and biases from the previous iteration will be used again in the current iteration. The speed error, $e(n) = \omega^*(n) - \omega(n)$, as well as the approximated torque error $e_T(n) = T_{ref}(n) - T_{ANN}(n)$, are used for back propagation updating, as described in Chapter 3.

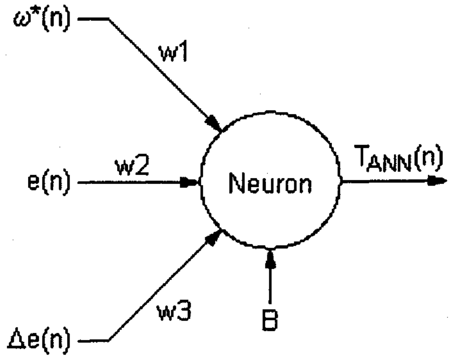


Figure 5.2 SAN for IPMSM drive.

To maintain drive stability, the output command torque of the SAN is compared to the reference torque and if the relative error between the two, as expressed by $\left|\frac{T_{ANN}(n) - T_{ref}(n)}{T_{ref}(n)}\right|$, exceeds 0.10 then retraining is done to return $T_{ANN}(n)$ to within 10% of $T_{ref}(n)$.

5.3 Simulation Results and Discussion

This section presents simulation results of the single artificial neuron controller-based IPMSM drive described above. The initial weights and biases of this network are as follows:

W1 = 1

W2 = 1

W3 = 1

B = 0

As with the more complex ANN-based controller presented in Chapter 4, no off-line training was done to determine these weights. They were chosen arbitrarily as precise initial weights and biases are not necessary. To maintain consistency, the same sampling frequency used with the four-neuron ANN controller-based drive is used with the SAN controller-based drive as well – 5000 Hz. Appendix C presents some graphical representations of the running weights and bias updating of the SAN, under various operating conditions, as a function of command torque

5.3.1 Rated Speed and Below Operation (with MTPA)

Figure 5.3 shows the startup response of the SAN controller-based IPMSM drive under no load for a base speed command of 188.5 rad./sec. The drive is shown to track command speed accurately and precisely with no overshoot in the transient state. Weights and bias training under these conditions are shown in Appendix C.

Figure 5.4 shows the current in phase 'a' of the SAN controller-based IPMSM drive under no load for a command speed of 188.5 rad./sec. Peak phase current is approximately 1.5 A, but several anomalous higher peaks can be seen. Figure 5.5 shows the command current in phase 'a' of the SAN-based under the same conditions. The amplitude of the command current is clearly greater than that of the actual phase current.

Figure 5.6 shows the response of the SAN controller-based IPMSM drive to step changes of command speed (from 90 to 188.5 to 130 rad./sec.) at no load. The drive accurately and precisely follows these changes. Figure 5.7 shows the current in phase 'a' of the SAN controller-based IPMSM drive under step changes of command speed (from 90 to 188.5 to 130 rad./sec.). Peak steady-state current is approximately 1.5 A, with higher currents exhibited during the transient acceleration and deceleration periods. Figure 5.8 shows the command current in phase 'a' of the drive under these same conditions. It can be clearly seen that command current frequency increases with increasing speed command.

Figure 5.9 shows the response of the SAN controller-based IPMSM drive to a sudden application of full-load (0 to 2 Nm) at a command speed of 188.5 rad./sec. The speed disturbance due to the load application is almost indiscernible. Figure 5.10 shows the current response in phase 'a' of the drive under these same conditions. Peak steady-state current increases to approximately 3 A when full-load is applied. Figures 5.11 and 5.12 show the corresponding q- and d-axis current responses of the drive, respectively. It can be seen that the q-axis current increases under load to produce more torque, and the d-axis current increases in the negative direction to take advantage of the reluctance component of the torque equation, as per the MTPA scheme.

Figure 5.13 shows the startup speed response of the SAN controller-based IPMSM drive to a sinusoidal speed command with an amplitude of 25 rad./sec. and 163.5 rad./sec. offset. The drive quickly and accurately tracks command speed. Figure 5.14 shows the current response in phase 'a' of the drive under these same conditions. It can be seen that phase current amplitude increases during acceleration and decreases during deceleration. This is to produce the torque necessary to accelerate and decelerate the rotor. Figures 5.15 and 5.16 show the corresponding q- and d-axis current responses of the drive, respectively. It can be seen that the q-axis current increases to produce the necessary torque during acceleration and decreases during deceleration. As the d-axis current under the MTPA scheme varies as a relatively small proportion of $(i_q^r)^2$, as can be seen in Equation (2.49), changes in the magnitude of i_d^r are more difficult to detect visually.

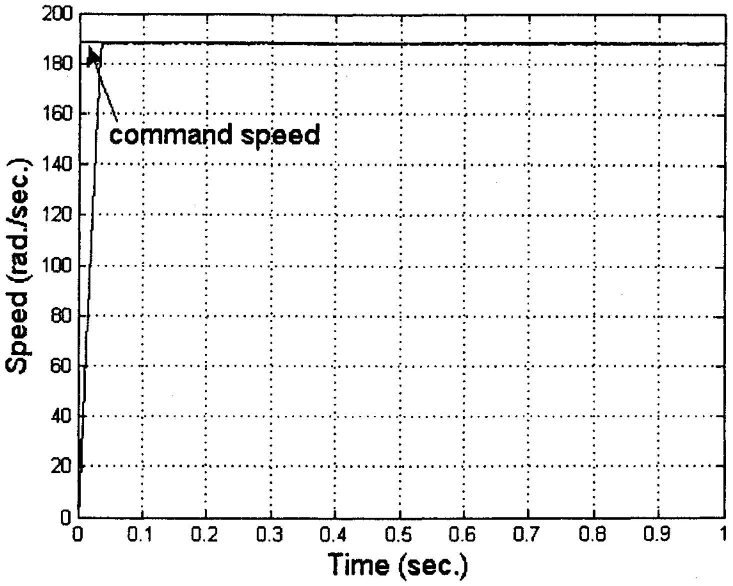


Figure 5.3. Response of SAN controller-based IPMSM drive under no load, command speed of 188.5 rad./sec.

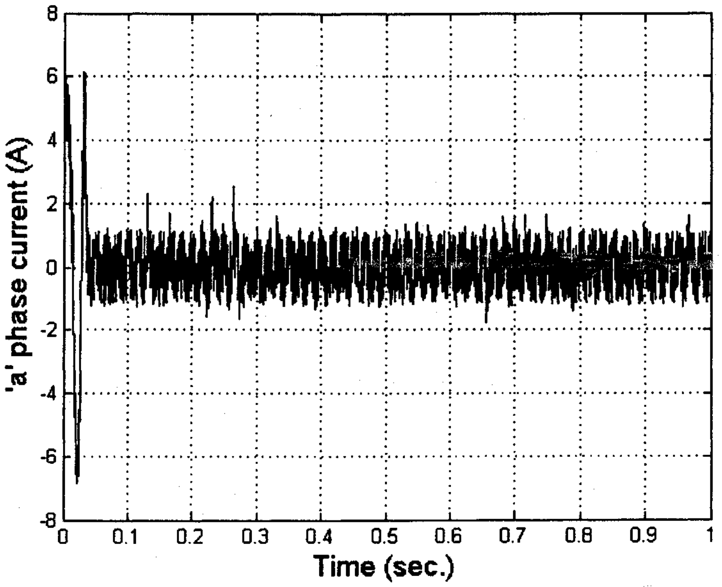


Figure 5.4. Current in phase 'a' of SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

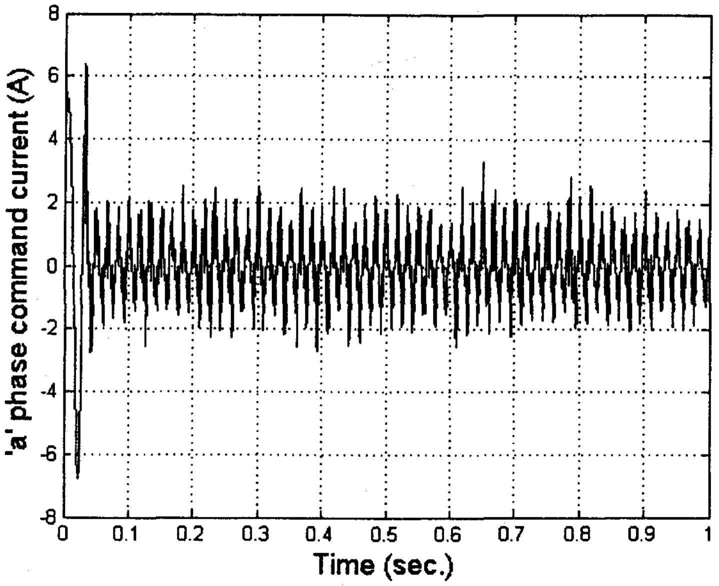


Figure 5.5. Command current in phase 'a' of SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

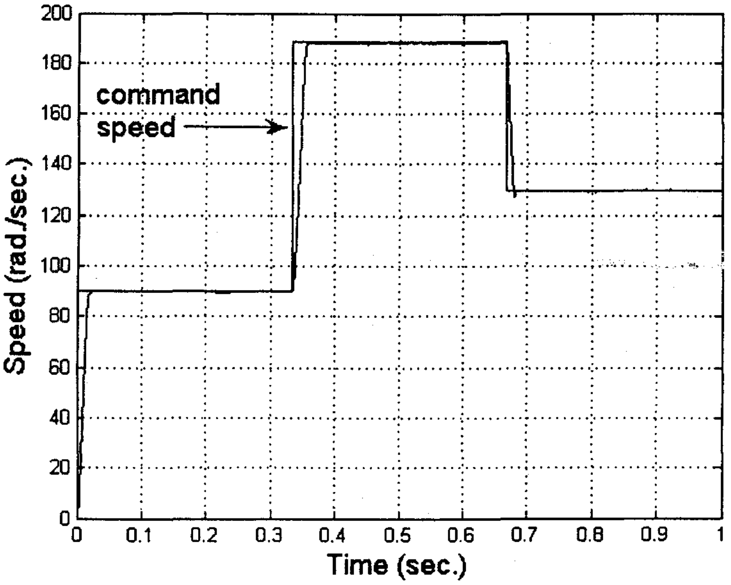


Figure 5.6. Response of the SAN controller-based IPMSM drive to step changes of command speed (90 → 188.5 → 130 rad./sec.).

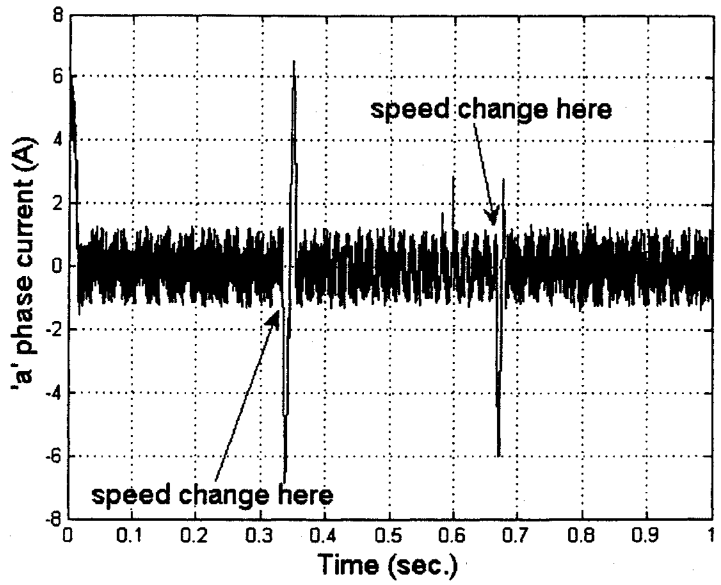


Figure 5.7. Current in phase 'a' of SAN controller-based IPMSM drive under step changes of command speed (90 → 188.5 → 130 rad./sec.).

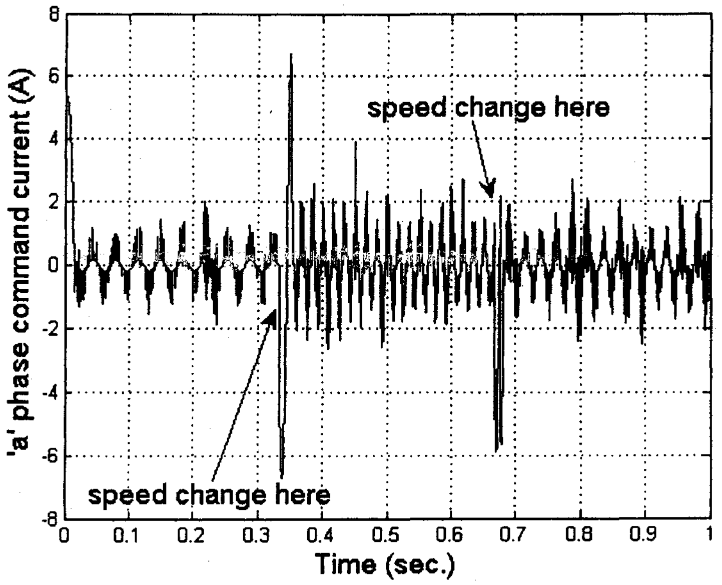


Figure 5.8. Command current in phase 'a' of SAN controller-based IPMSM drive under step changes of command speed (90 → 188.5 → 130 rad./sec.).

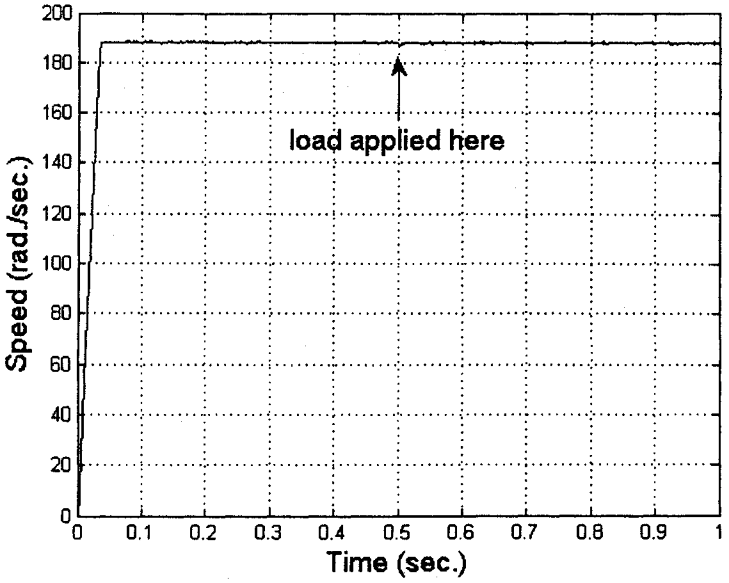


Figure 5.9. Response of SAN controller-based IPMSM to sudden application of full-load (0 → 2 Nm) at a command speed of 188.5 rad./sec.

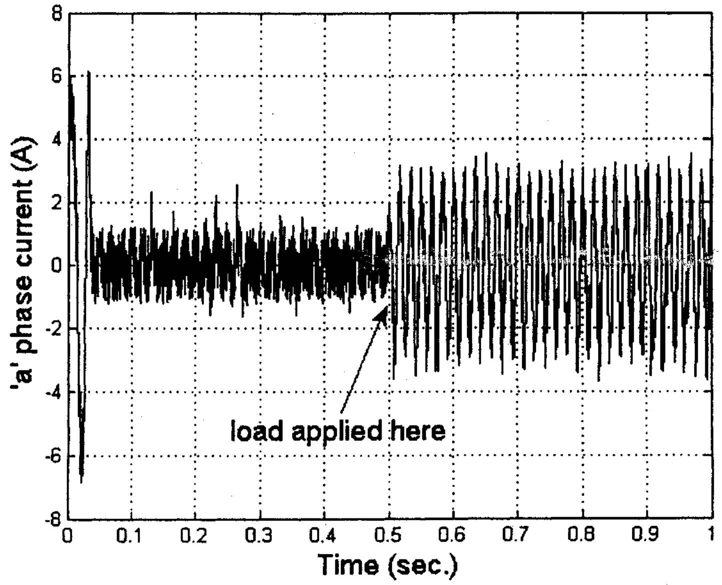


Figure 5.10. Current response in phase 'a' of SAN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

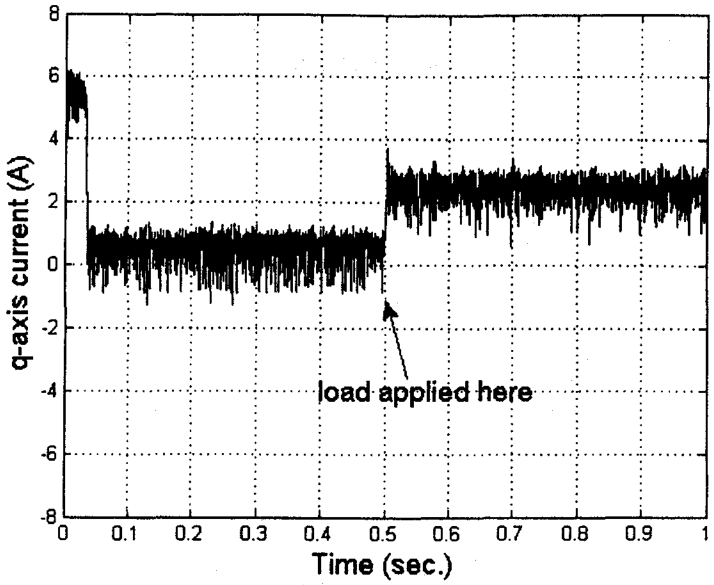


Figure 5.11. Q-axis current response of SAN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

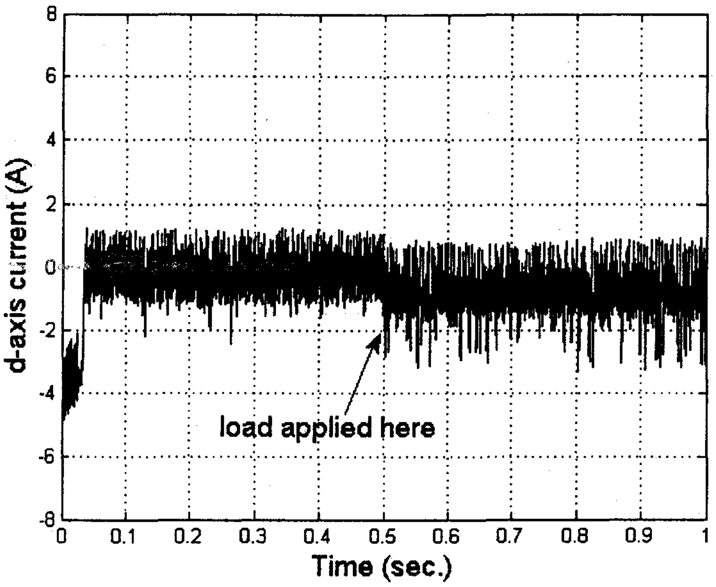


Figure 5.12. D-axis current response of SAN controller-based IPMSM drive to a sudden application of full-load (0 → 2 Nm) at 188.5 rad./sec. speed command.

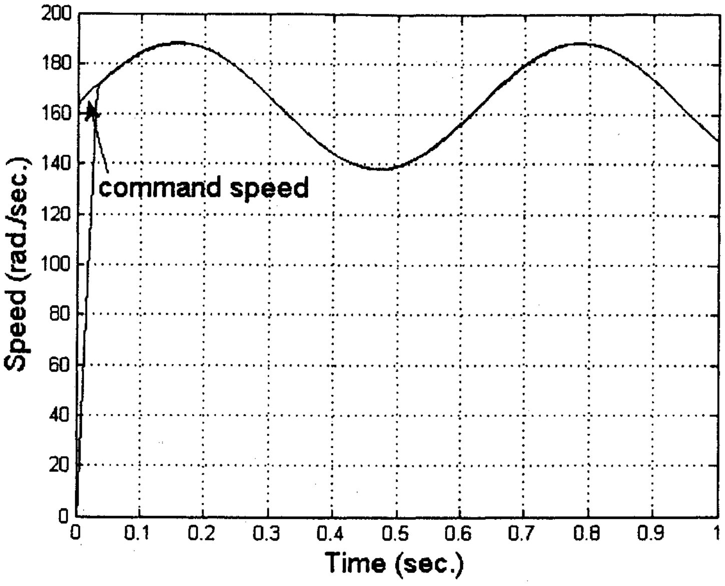


Figure 5.13. Response of SAN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

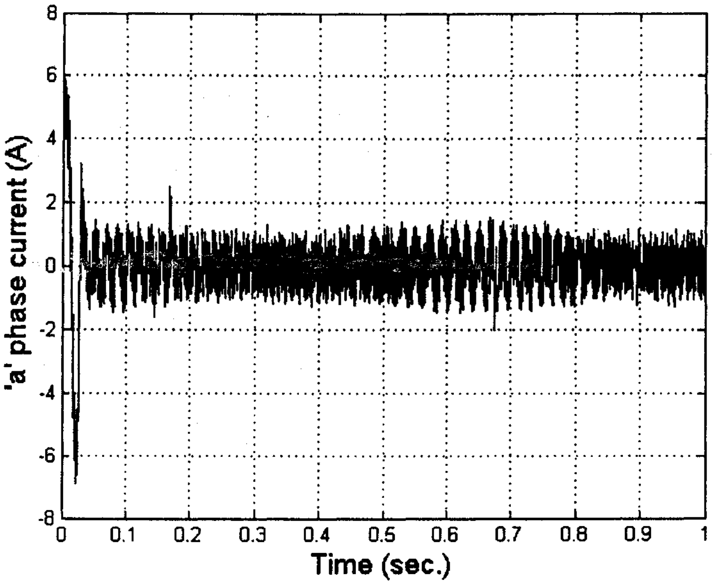


Figure 5.14. Current response in phase 'a' of SAN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

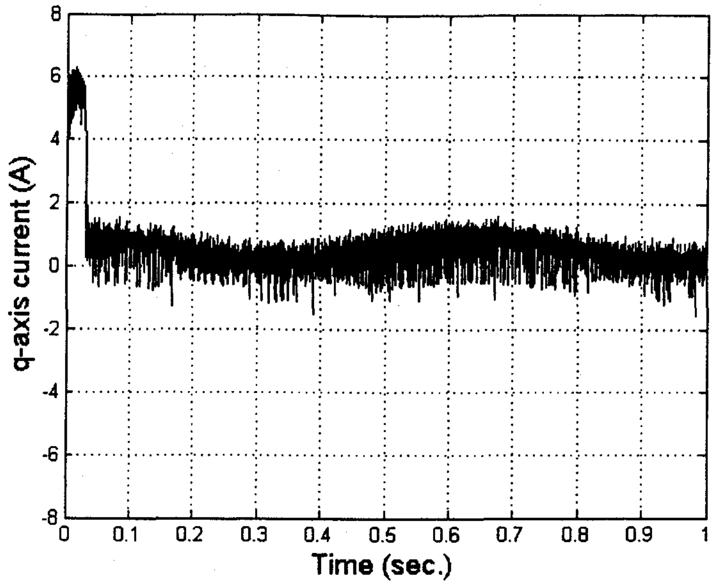


Figure 5.15. Q-axis current response of SAN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

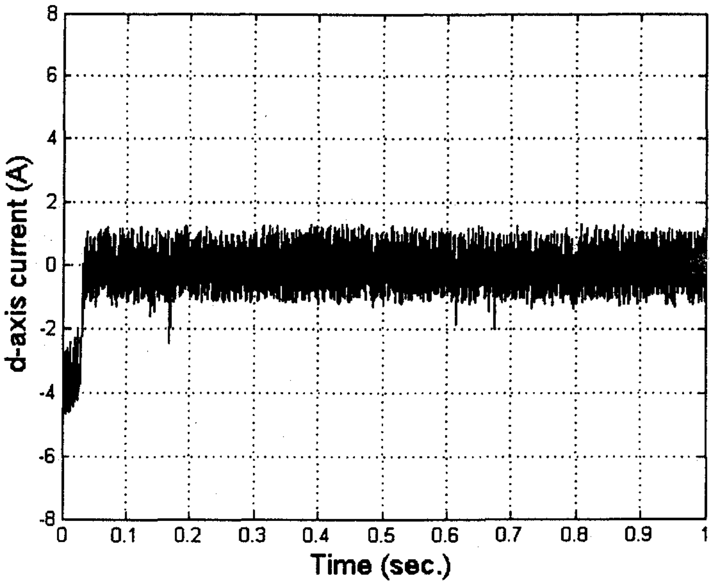


Figure 5.16. D-axis current response of SAN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

5.3.2 Rated Speed and Below Operation with Parameter Variations (MTPA mode)

Worst-case scenario parameter changes when operating at or below base speed, in MTPA mode, are investigated in this section.

Figure 5.17 shows the response of the SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of q-axis inductance at base speed command (188.5 rad./sec.). The drive shows no speed disturbance due to this L_q parameter change. Figure 5.18 shows the current response in phase 'a' of the drive under these same conditions. It is difficult to visually detect the disturbance. Figures 5.19 and 5.20 show the corresponding q- and d-axis current responses of the drive, respectively. The q-axis current appears to show a slight reaction when L_q is increased. Disruption to the d-axis current is more difficult to detect visually.

Figure 5.21 shows the response of the SAN controller-based IPMSM drive, under no load, to a sudden doubling of stator resistance at base speed command (188.5 rad./sec.). The drive shows no speed disturbance due to this sudden R_s parameter change. Figure 5.22 shows the current response in phase 'a' of the drive under these same conditions. Again, it is difficult to visually detect any disturbance. Figures 5.23 and 5.24 show the corresponding q- and d-axis current responses of the drive, respectively. These currents also appear to be not significantly affected by the parameter change.

Figure 5.25 shows the speed response of the ANN controller-based IPMSM drive to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm) at base speed command (188.5 rad./sec.). The drive appears minimally affected by these changes. Figures 5.26 (a) and (b) and 5.27 (a) and (b) show the 'a' phase current and 'a' phase command current responses, respectively, of the drive under the same conditions. Similarly to the four-neuron ANN-based drive, fluctuations due to discrete sampling and the hysteresis current controller can be seen. Actual phase current increases to a peak value of roughly 3 A when the disturbances are applied. Figures 5.28 and 5.29 show the corresponding q- and d-axis current responses of the drive. The absolute magnitudes of the q- and d-axis currents clearly increase in response to the sudden disturbances.

Figure 5.30 shows the speed response of the SAN controller-based IPMSM drive, under a non-linear load of $T_L = \omega^2/17766.125$, to a sinusoidal command speed with an amplitude of 25 rad./sec. and 163.5 rad./sec. offset. At 0.5 seconds, L_q is increased by 50% and R_s doubled. The drive experiences no difficulty in tracking command speed under these conditions. Figures 5.31, 5.32 and 5.33 show the corresponding 'a' phase, q- and d-axis current responses of the drive, respectively. The amplitude and frequency of the 'a' phase current can be seen to increase with increasing load and speed. Likewise, the peak absolute values of q- and d-axis currents also increase with load. However, no disruption due to the parameter changes is discernible.

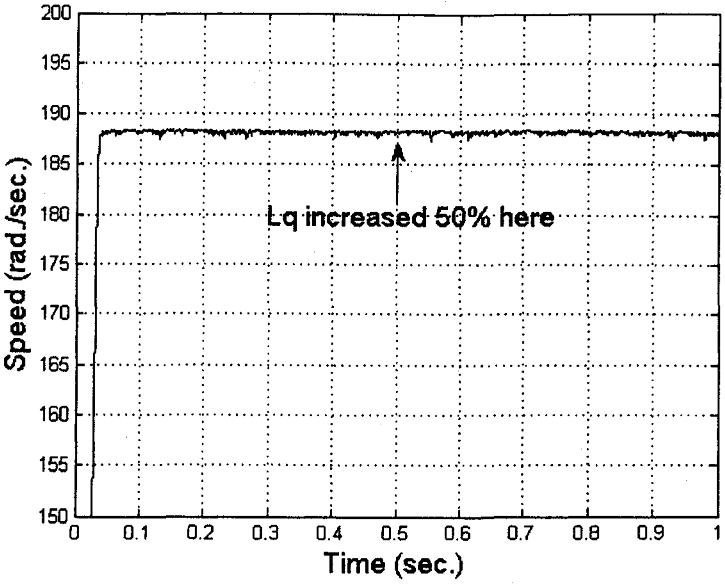


Figure 5.17. Response of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

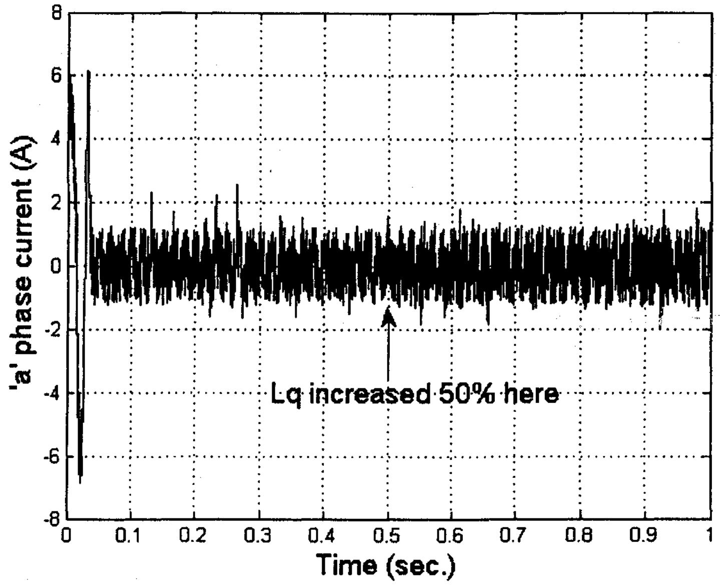


Figure 5.18. Current response in phase ‘a’ of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

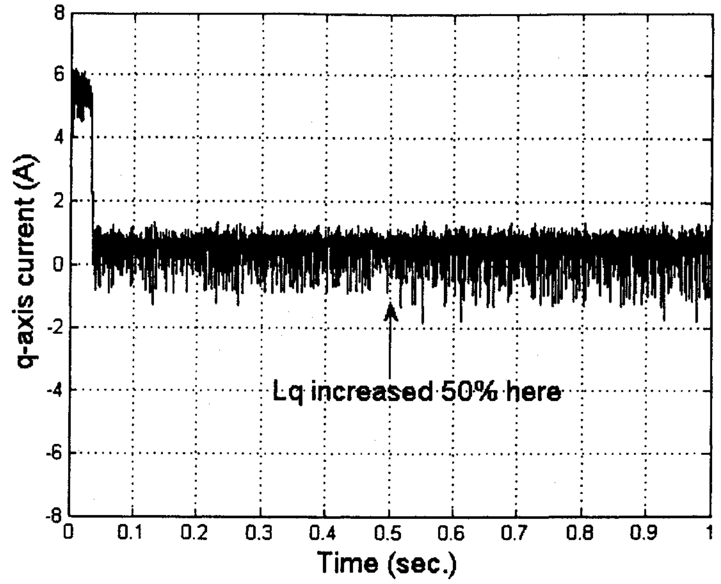


Figure 5.19. Q-axis current response of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

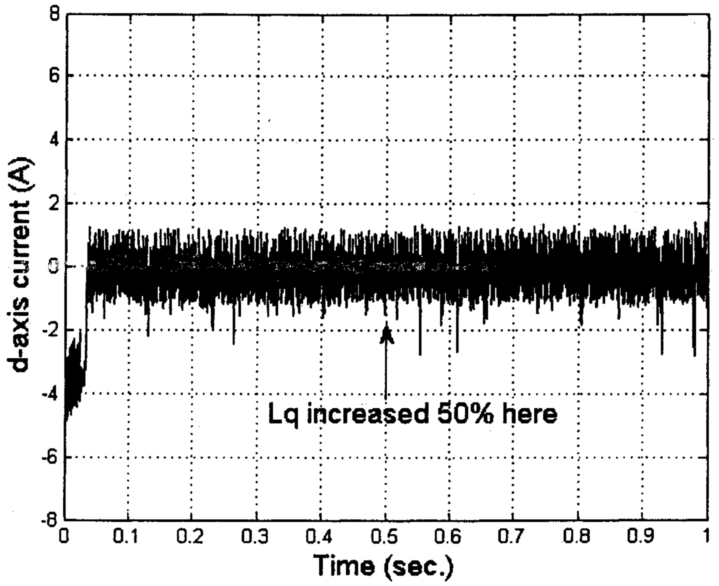


Figure 5.20. D-axis current response of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.).

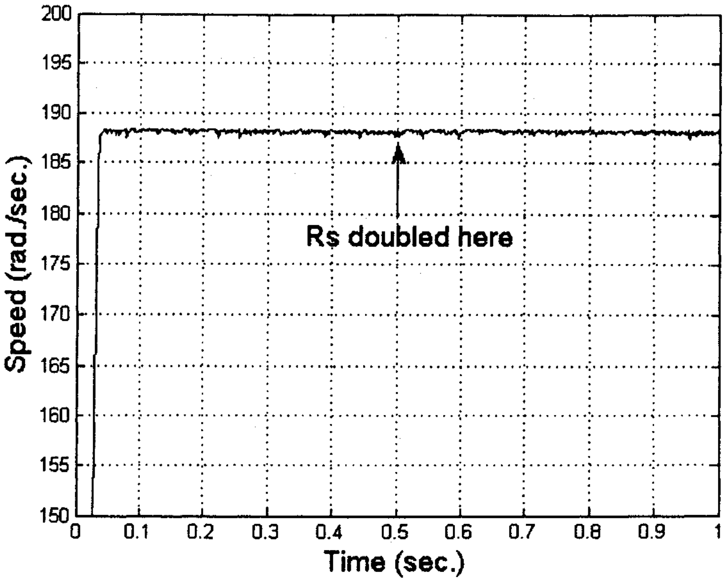


Figure 5.21. Response of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

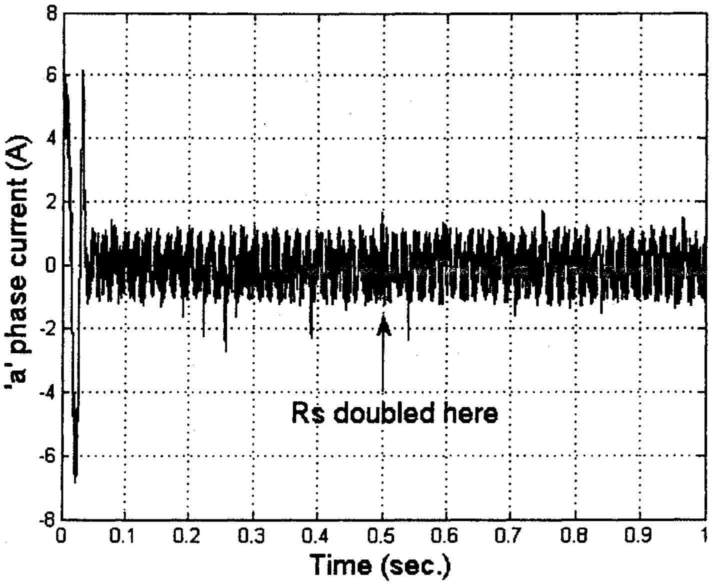


Figure 5.22. Current response in phase 'a' of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

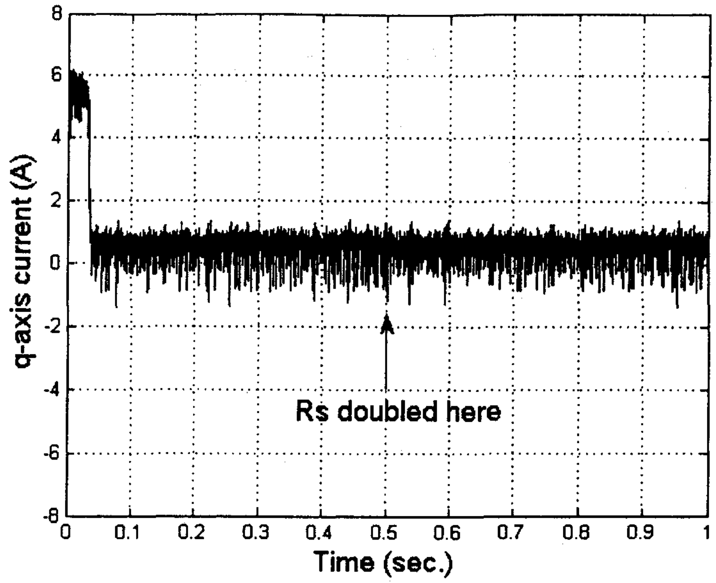


Figure 5.23. Q-axis current response of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

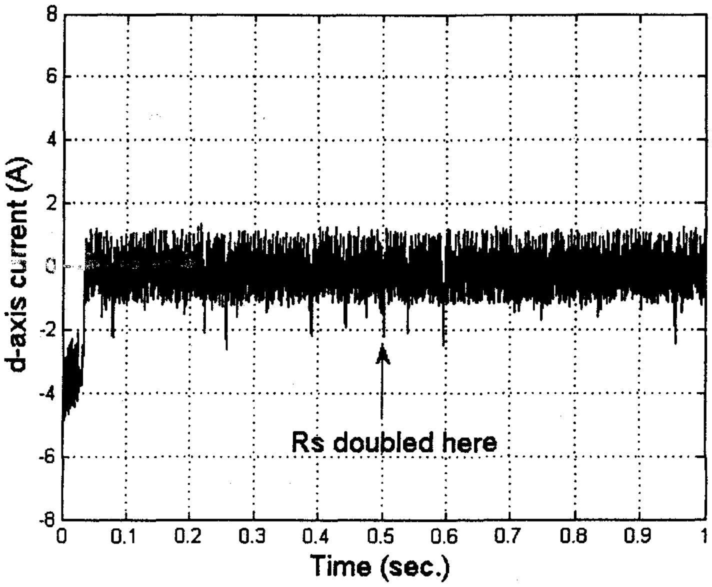


Figure 5.24. D-axis current response of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at base speed command (188.5 rad./sec.).

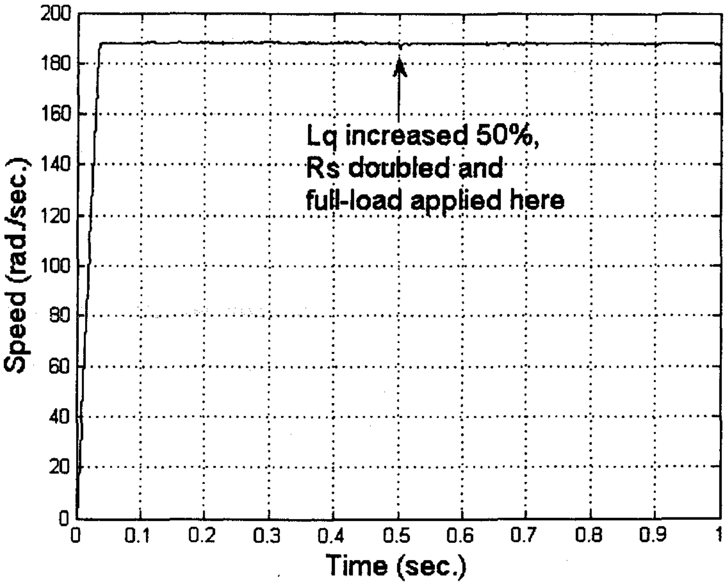
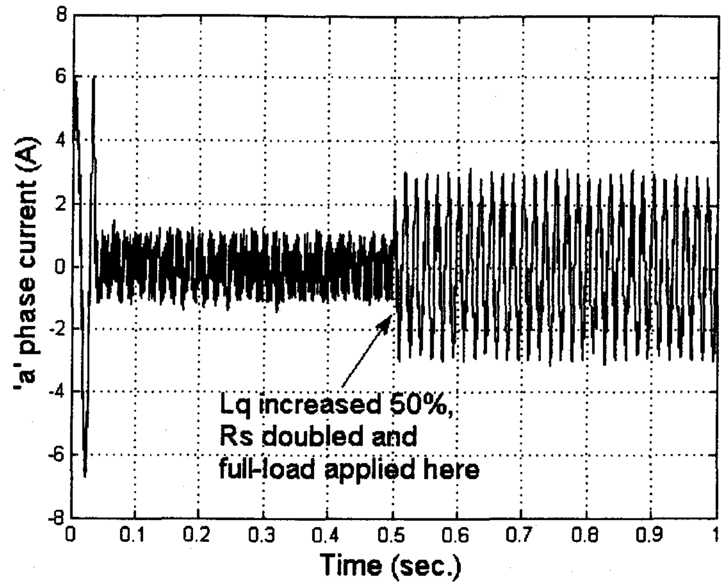
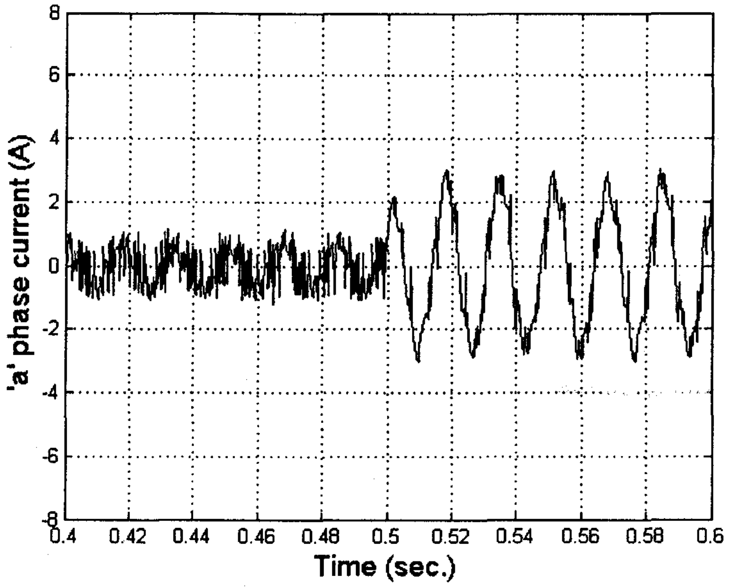


Figure 5.25. Response of SAN controller-based IPMSM drive at base speed command (188.5 rad./sec.) to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm).

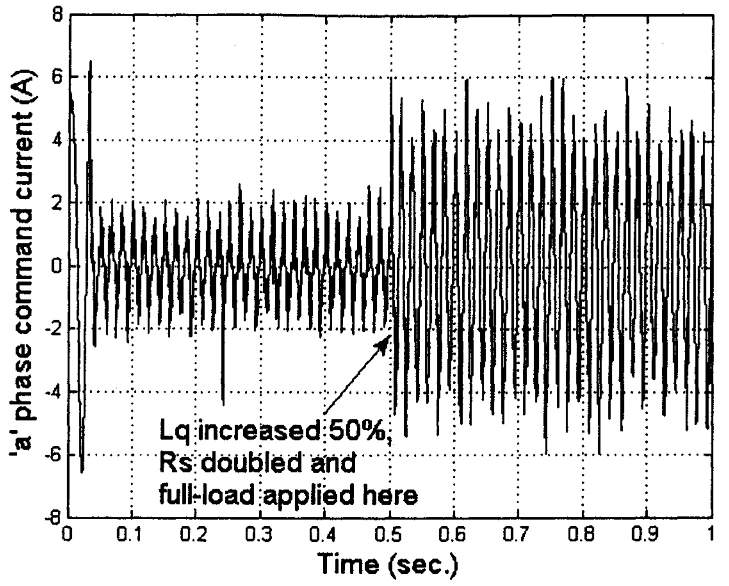


(a)

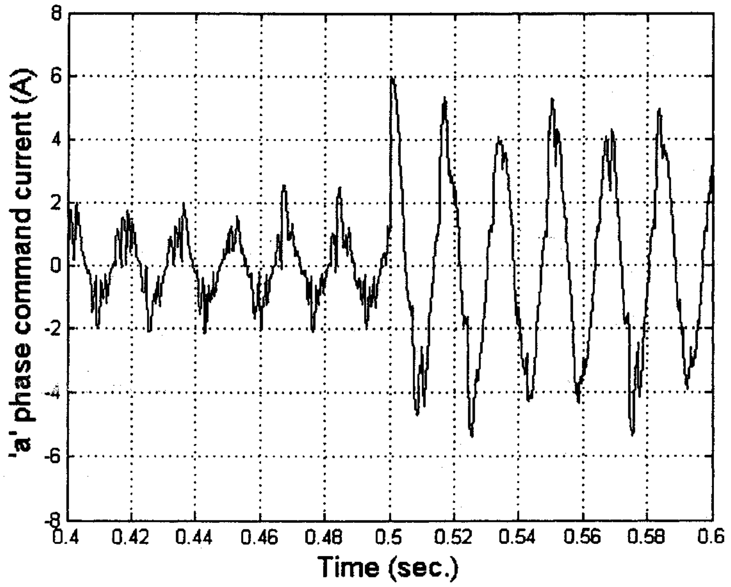


(b)

Figure 5.26 (a) and (b). Current response in phase 'a' of SAN controller-based IPMSM drive at base speed command (188.5 rad./sec.) to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2Nm).



(a)



(b)

Figure 5.27 (a) and (b). 'a' phase command current of SAN controller-based IPMSM drive at base speed command (188.5 rad./sec.) with a sudden 50% increase of L_q, doubling of R_s and application of full-load.

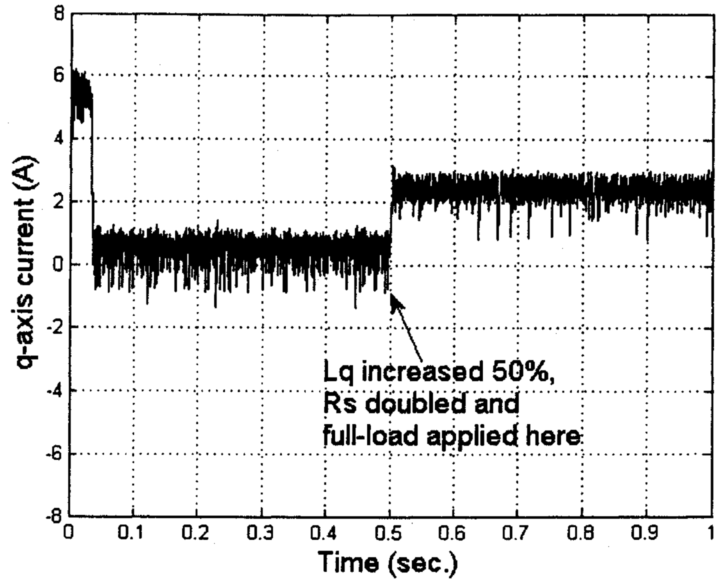


Figure 5.28. Q-axis current of SAN controller-based IPMSM drive at base speed command (188.5 rad./sec.) with a sudden 50% increase of L_q, doubling of R_s and application of full-load.

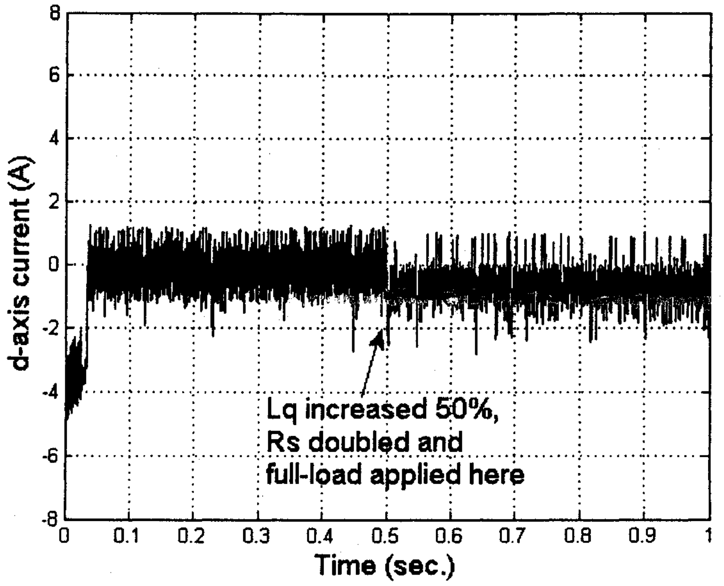


Figure 5.29. D-axis current of SAN controller-based IPMSM drive at base speed command (188.5 rad./sec.) with a sudden 50% increase of L_q, doubling of R_s and application of full-load.

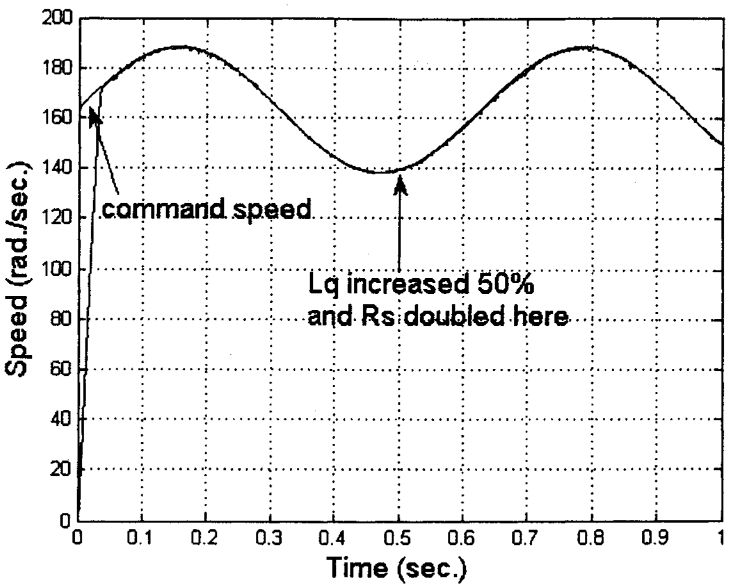


Figure 5.30. Response of SAN controller-based IPMSM drive, under non-linear load, to sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

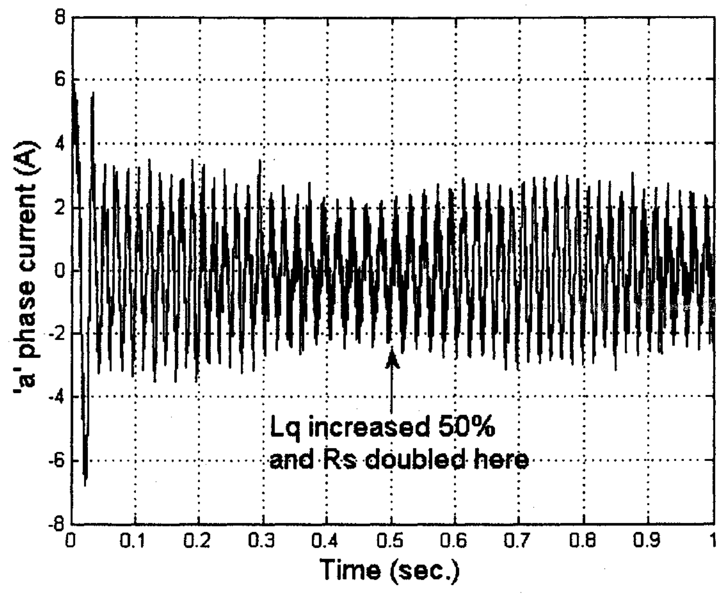


Figure 5.31. Current response in phase 'a' of SAN controller-based IPMSM drive under non-linear load to sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s.

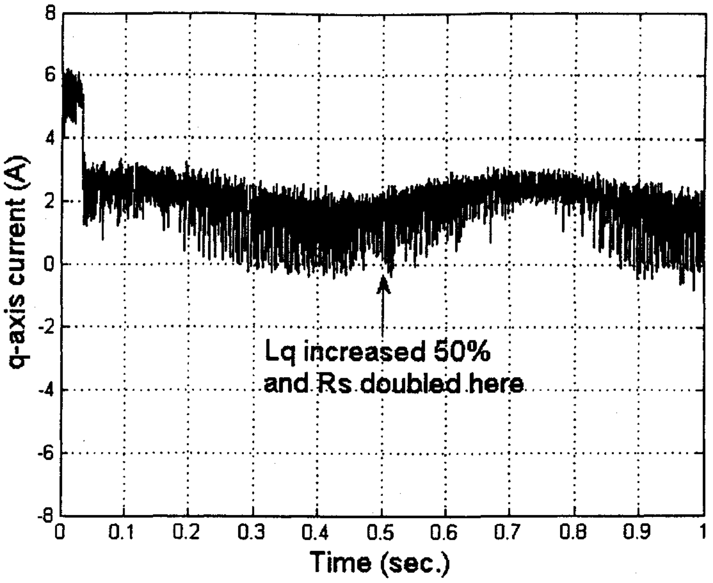


Figure 5.32. Q-axis current response of SAN-based IPMSM drive under non-linear load to sinusoidal speed command, 50% increase of L_q and doubling of R_s.

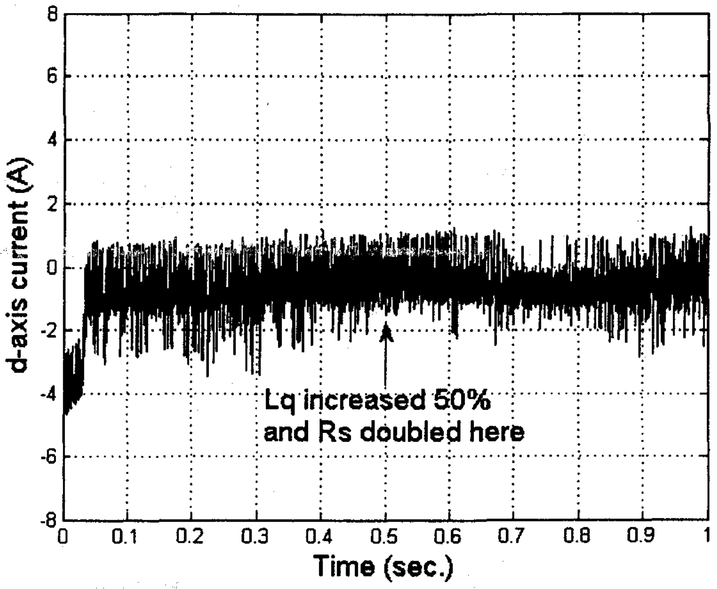


Figure 5.33. D-axis current response of SAN-based IPMSM drive under non-linear load to sinusoidal speed command, 50% increase of L_q and doubling of R_s.

5.3.3 Above Rated Speed Operation (with FW)

Figures 5.34 and 5.35 show the startup speed and 'a' phase current responses of the SAN controller-based IPMSM drive under no load for a command speed of 250 rad./sec, utilizing the flux-weakening mode of operation. In this case, V_o, in the FW Equations (2.58) and (2.59), has been chosen to yield a peak phase current of 3 A at 250 rad./sec., as seen in Figure 5.35. Figure 5.34 shows that the drive quickly and accurately reaches and tracks command speed. Weights and bias training of the SAN under these conditions are shown in Appendix C.

Figure 5.36 shows the response of the SAN controller-based IPMSM drive to step changes of command speed (from 188.5 to 250 to 188.5 rad./sec.) at no load. The drive accurately and precisely follows these changes. Figure 5.37 shows the 'a' phase current response of the SAN controller-based IPMSM drive under these same conditions. At no-load, and command speed of 188.5 rad./sec., the FW-based ANN-controlled IPMSM drive is seen to experience a slightly higher phase current than when the MTPA scheme is used under these same conditions. Phase current increases to approximately 3 A at 250 rad./sec. Figures 5.38 and 5.39 show the corresponding q- and d-axis current responses of the SAN drive under the same conditions. A clear increase in negative d-axis current as command speed is increased beyond base speed is seen. Q-axis current spikes to generate the necessary torque as the motor is accelerated.

Figure 5.40 shows the startup speed response of the SAN controller-based IPMSM drive to a sinusoidal speed command with an amplitude of 25 rad./sec. and 225 rad./sec. offset. The drive very quickly and accurately reaches and tracks command speed. Figure 5.41 shows the current response in phase 'a' of the SAN controller-based IPMSM drive under these same conditions. An increase in phase current magnitude can be seen with increased command speed.

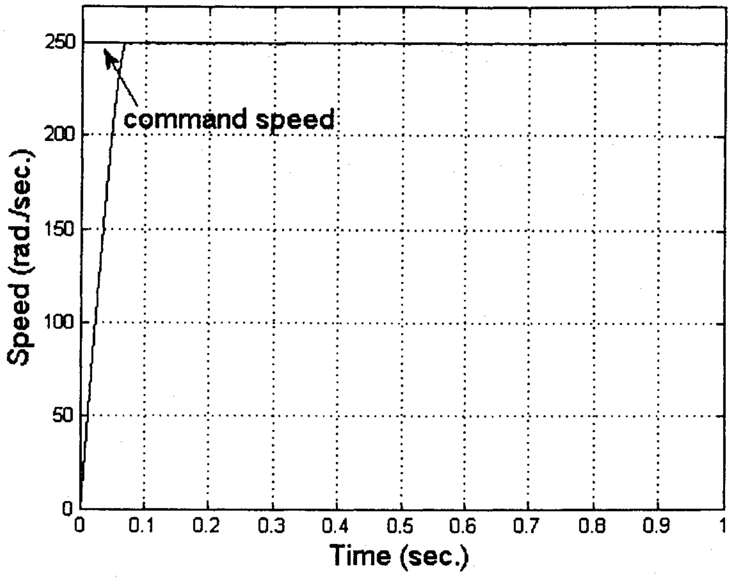


Figure 5.34. Response of SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

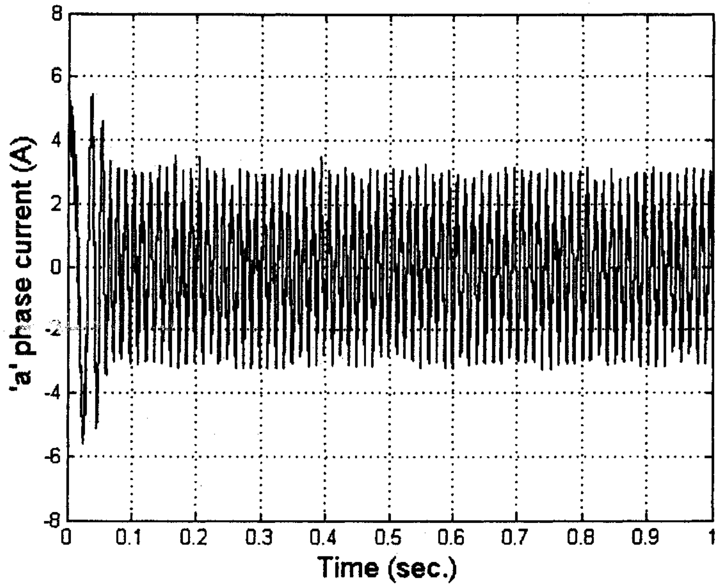


Figure 5.35. Current response in phase 'a' of SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

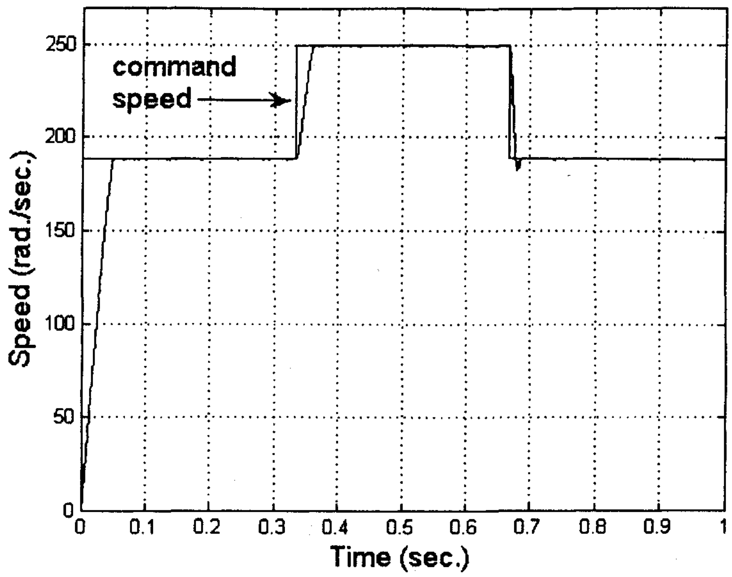


Figure 5.36. Response of the SAN controller-based IPMSM drive to step changes of command speed (188.5 → 250 → 188.5 rad./sec.).

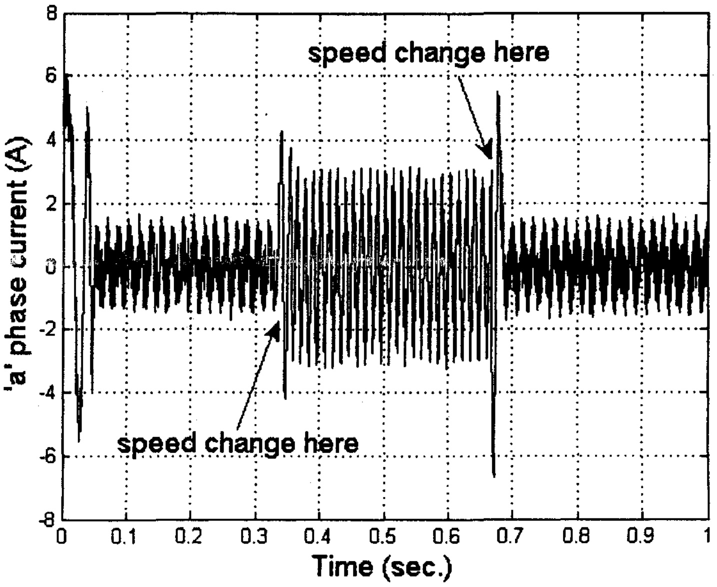


Figure 5.37. Current in phase 'a' of SAN controller-based IPMSM drive under step changes of command speed (188.5 → 250 → 188.5 rad./sec.).

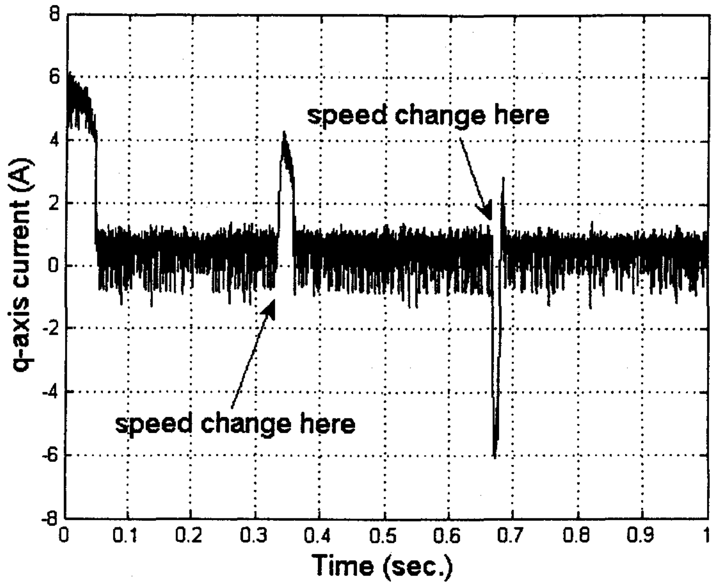


Figure 5.38. Q-axis current response of SAN controller-based IPMSM drive to step changes of speed command (188.5 → 250 → 188.5 rad./sec.).

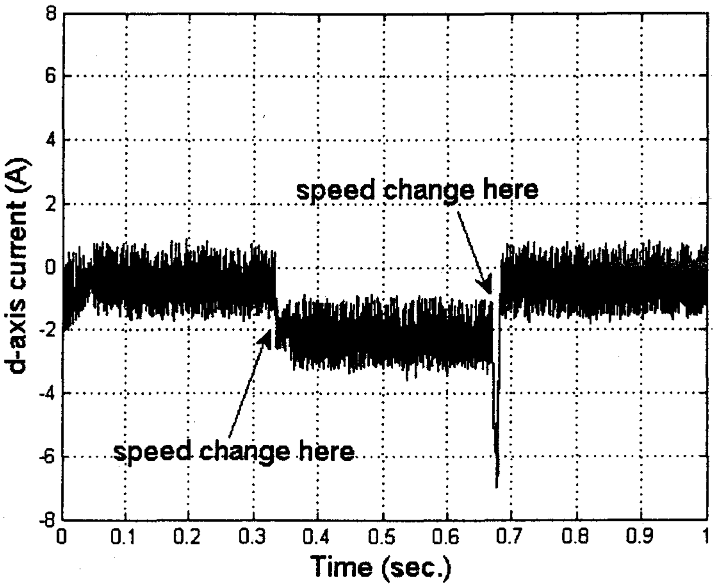


Figure 5.39. D-axis current response of SAN controller-based IPMSM drive to step changes of speed command (188.5 → 250 → 188.5 rad./sec.).

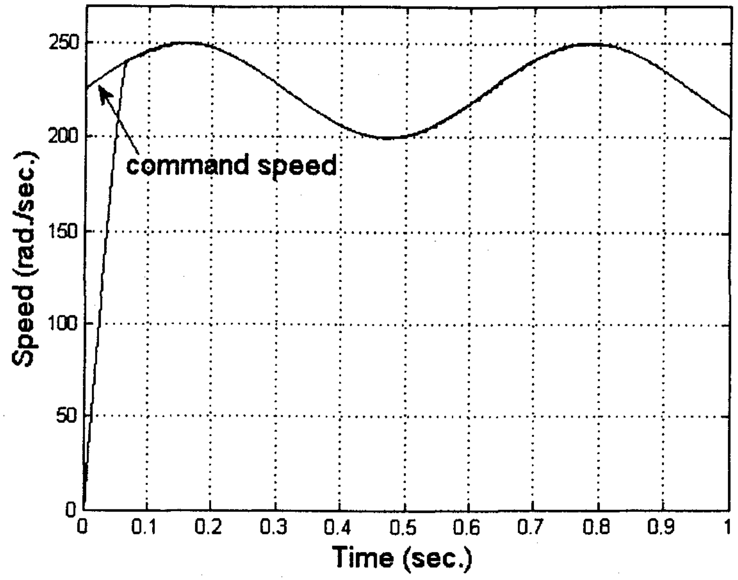


Figure 5.40. Response of SAN controller-based IPMSM drive, under no load, to a sinusoidal speed command.

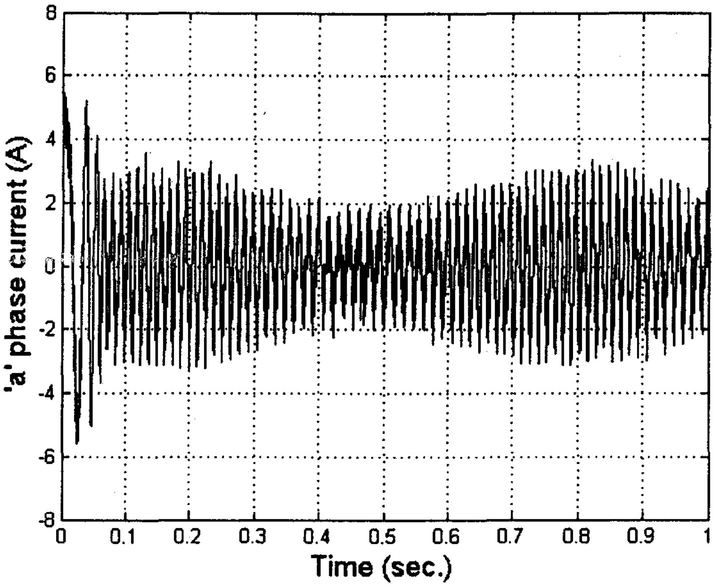


Figure 5.41. Current response in phase 'a' of SAN controller-based IPMSM drive, under no-load, to sinusoidal speed command above base speed.

5.3.4 Above Rated Speed Operation with Parameter Variations (FW mode)

Figure 5.42 shows the speed response of the SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec. Any significant disruption to motor speed is visually indiscernible. Figure 5.43 shows the 'a' phase current response of the drive under these same conditions. A minor disruption of phase current due to this parameter change seems to be detectable. Figures 5.44 and 5.45 show the corresponding q- and d-axis current responses, respectively. Here it can be seen that the change in L_q does result in an increase in q-axis current amplitude and peak negative value, but the d-axis current appears relatively unaffected.

Figure 5.46 shows the speed response of the SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec. Again, disruption to motor speed is visually indiscernible. Figure 5.47 shows the 'a' phase current response of the drive under these same conditions. Again, the disruption appears to have no significant effect. Figures 5.48 and 5.49 show the corresponding q- and d-axis current responses. It is difficult to visually ascertain whether the change of stator resistance has an identifiable affect on these currents.

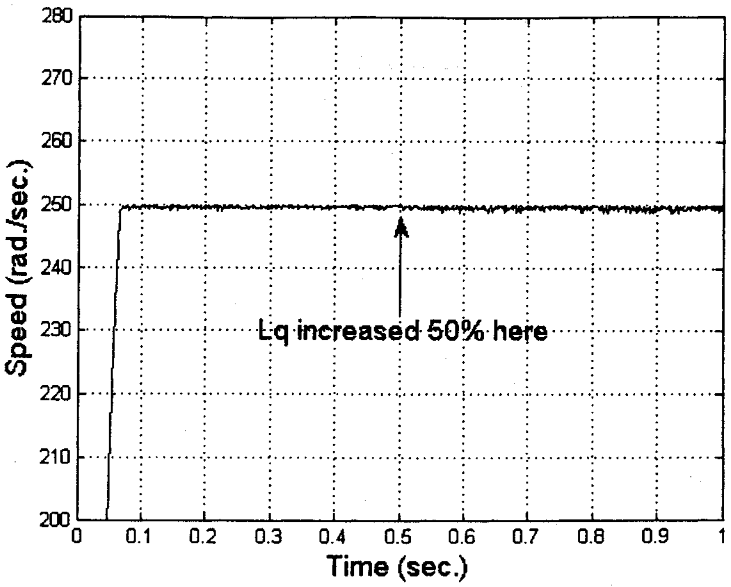


Figure 5.42. Response of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

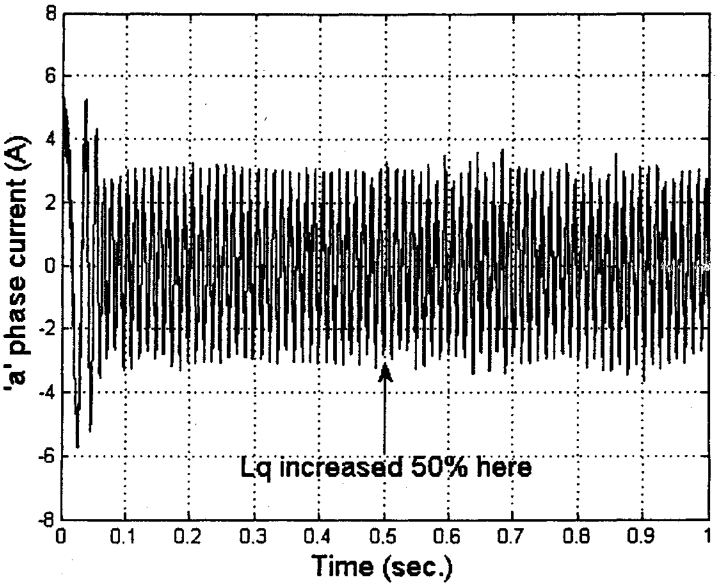


Figure 5.43. Current response in phase 'a' of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

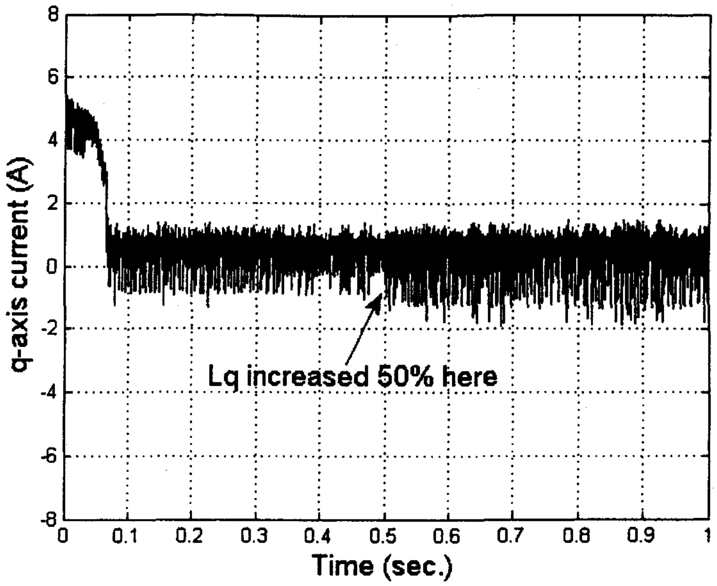


Figure 5.44. Q-axis current of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

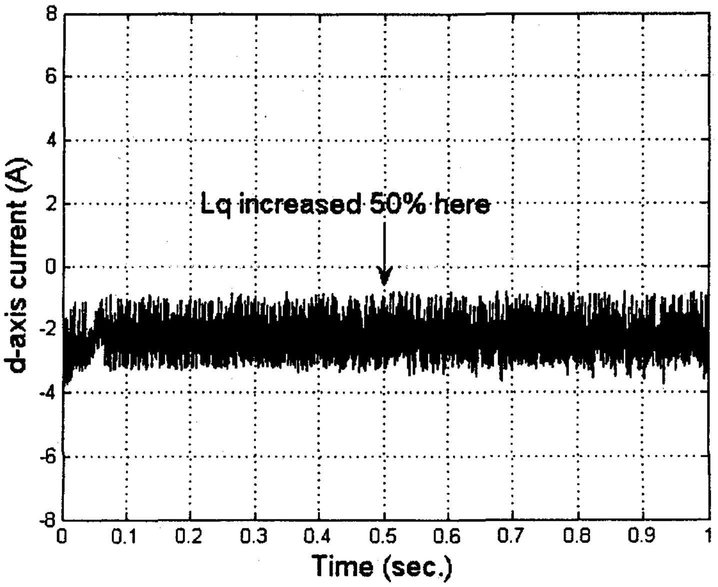


Figure 5.45. D-axis current of SAN controller-based IPMSM drive, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec.

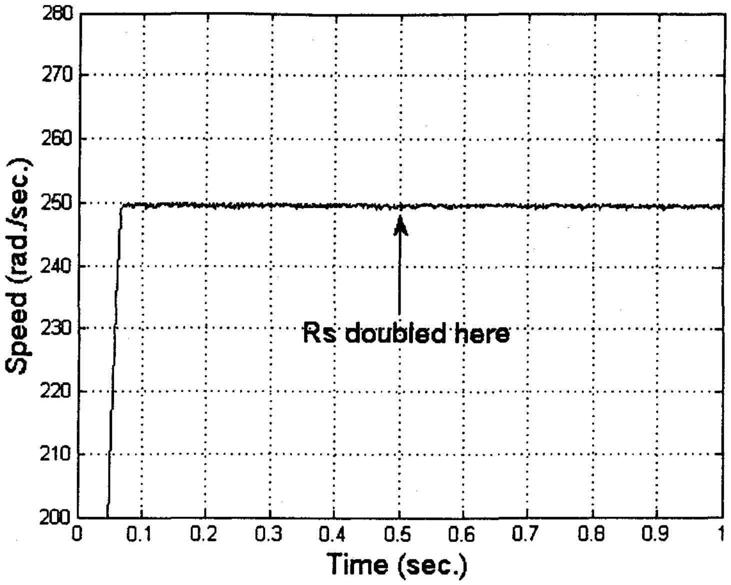


Figure 5.46. Response of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

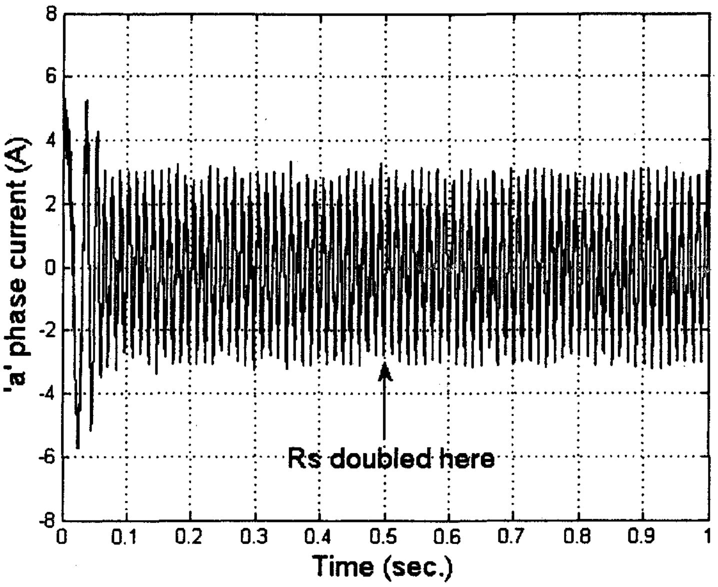


Figure 5.47. Current response in phase 'a' of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

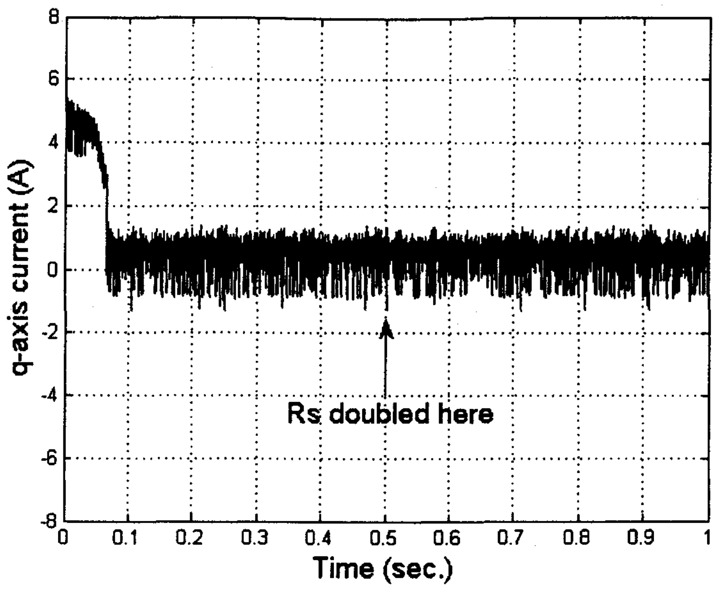


Figure 5.48. Q-axis current of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

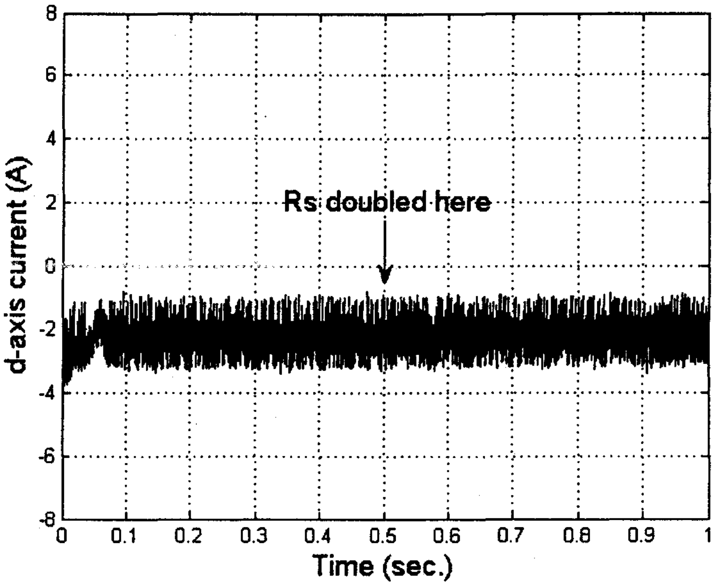


Figure 5.49. D-axis current of SAN controller-based IPMSM drive, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec.

5.4 Comparison to Four-Neuron ANN Controller

5.4.1 Rated Speed and Below Operation (with MTPA)

Figures 4.4 and 5.3 show the start-up speed responses of the IPMSM drives under no load to command speed of 188.5 rad./sec. with the four-neuron ANN (4-ANN) and SAN-based controllers, respectively. No observable difference is apparent between the responses of the two controllers.

Comparing the responses of the drives to step changes of command speed (90 → 188.5 → 130 rad./sec.), as shown in Figures 4.8 and 5.6, also indicates little performance difference. Though, the SAN-based drive does appear to experience smoother command speed tracking at 130 rad./sec. speed command. Phase current responses under these conditions, as shown in Figures 4.10 and 5.7, appear nearly identical.

A comparison of the responses of the 4-ANN and SAN-based drives to sudden application of full-load at rated speed command also yields little in the way of observable difference. This can be seen in Figures 4.12 and 5.9. The current responses of the drives under these conditions, as shown in Figure 4.14 and 5.10, seem to indicate that the 4-ANN drive produces a slightly smoother phase current response – though the difference is minute. The q, d-axis current responses of the drives, shown in Figures 4.16 and 4.18, and 5.11 and 5.12 indicate little difference between the responses of the 4-ANN and SAN-based drives. Though, again, the 4-ANN drive may produce a slightly smoother q-axis current response.

The responses of the 4-ANN and SAN controller-based IPMSM drives under no load to a sinusoidal speed command, as shown in Figures 4.20 and 5.13, respectively, appear practically identical.

Comparing Figures 4.22 and 5.17 indicates that the responses of drives, under no load, to a sudden 50% increase of L_q at base speed command (188.5 rad./sec.) is practically indiscernible – though the 4-ANN controller-based drive may produce a very slightly smoother speed response. Comparison of the drives' responses, under no load, to a sudden doubling of R_s at base speed command, as shown in Figures 4.24 and 5.21, yields a similar conclusion.

The speed responses of the drives to a sudden 50% increase of L_q, doubling of R_s and application of full-load (2 Nm) at base speed command (188.5 rad./sec.), as shown in 4.26 and 5.25, indicates little difference. Comparison of the phase current responses under these conditions, as shown in Figures 4.28 and 5.26, also produces nearly identical performances.

Comparison of the responses of the drives under non-linear load ($T_L = \omega^2/17766.125$) and subjected to a sinusoidal speed command, sudden 50% increase of L_q and doubling of R_s, as shown in Figures 4.32 and 5.30, indicates nearly identical performances. Phase and q, d-axis current responses under these conditions, as shown in Figures 4.35, 4.36, 4.37 and 5.31, 5.32, 5.33 seem to indicate very similar results. Though, again, the 4-ANN controller-based drive appears to exhibit slight smoother current responses.

5.4.2 Above Rated Speed Operation (with FW)

Figures 4.38 and 5.34 show the start-up speed responses of the 4-ANN and SAN controller-based drives under no load to command speed of 250 rad./sec., respectively. The SAN-based drive appears to exhibit slightly superior speed tracking, though the extent of the performance difference is rather minute.

A comparison of drive speed responses under no load, to sinusoidal speed command above base speed, as shown in Figures 4.43 and 5.40, yields no discernible performance differences. A comparison of phase current responses under these conditions, as shown in Figures 4.44 and 5.41, also indicates nearly identical performances.

Figures 4.45 and 5.42 show the speed responses of the 4-ANN and SAN controller-based IPMSM drives, under no load, to a sudden 50% increase of L_q at a command speed of 250 rad./sec., respectively. Performances appear very similar, though the 4-ANN may produce a slightly more accurate tracking of command speed after the application of parameter change. Phase and q, d-axis current responses under these conditions, as shown in Figures 4.46, 4.47, 4.48 and 5.43, 5.44, 5.45 seem to indicate nearly identical responses.

Figures 4.49 and 5.46 show the speed responses of the drives, under no load, to a sudden doubling of R_s at a command speed of 250 rad./sec. Performances appear nearly identical. Phase and q, d-axis current responses under these conditions, as shown in Figures 4.50, 4.51, 4.52 and 5.47, 5.48, 5.49 also appear to correspond closely.

5.5 Summary

From the simulated results it is seen that the IPMSM drive can be successfully implemented with a single artificial neuron-based speed controller that trains itself adaptively in real-time based on the scheme covered in both this chapter and Chapter 3. This SAN-based controller yields similar performance standards to the four-neuron ANN-based speed controller presented in Chapter 4, but with a substantial reduction in complexity and computational burden.

Chapter 6

Experimental Implementation of the SAN-based Vector Control of the IPMSM

6.1 Introduction

Online implementation of an ANN-controller based drive presents difficulties in that the high computational burdens imposed by the ANN necessitate both high-level hardware and software. This is due to the many calculations involved both in producing a network output by feed-forward operation of the network, and during back-propagation weights and biases updating. Because of this, the real-time implementation of the ANN for motor drives has presented a challenge for control engineers. Complex networks typically impose lower sampling frequencies than those used with simpler control algorithms to allow sufficient computing time to produce stable and accurate motor control. This lower sampling frequency results in a loss of precision, accuracy and response time of the drive.

To illustrate this loss of drive performance with decreased sampling frequency, Figures 6.1 and 6.2 simulate the speed and current responses, respectively, of the IPMSM drive subjected to a sudden application of full-load (2 Nm). The same ANN-based speed controller as presented in Chapter 4 is used, but with the sampling frequency decreased from 5000 Hz (as was used with both the ANN and SAN-based drive simulations) to 1000 Hz to reflect the additional computational burden imposed by the four-neuron ANN over the single artificial neuron (SAN) controller. The results clearly indicate a loss of drive accuracy and precision, although the drive recovers quickly from the sudden application of full-load. Phase current is considerably higher during no-load and loaded operation than with the other controllers tested. With this sampling frequency the drive does not produce an acceptable performance standard.

In real-time, the digital signal processor (DSP) used with the IPMSM drive in the laboratory allowed a maximum sampling frequency of 5000 Hz with the SAN controller-based drive and 1000 Hz with the four-neuron ANN controller-based drive. At this sampling frequency the ANN controller-based drive did not produce acceptable results. For that very reason the SAN controller-based drive was developed and implemented. Results indicate that the SAN-based controller achieves a performance standard equal to, or nearly equal to, the four-neuron ANN-based controller.

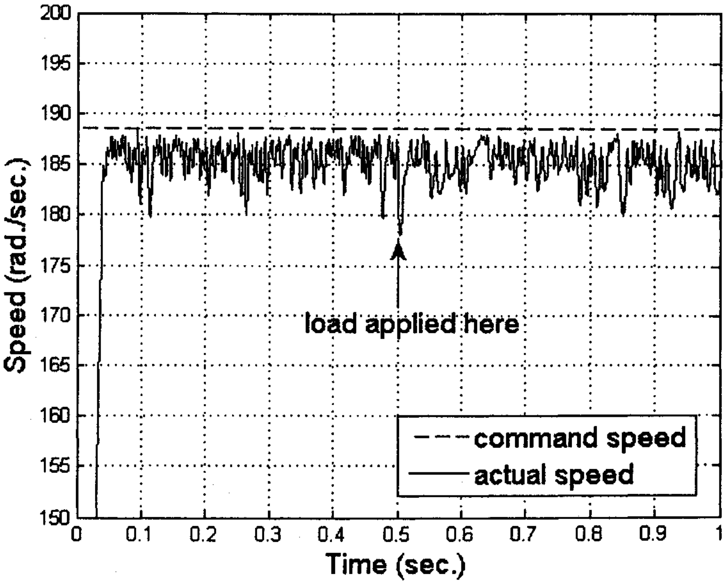


Figure 6.1 Response of ANN controller-based IPMSM drive at 1000 Hz sampling frequency to sudden application of full-load (0 → 2 Nm) at a rated speed command.

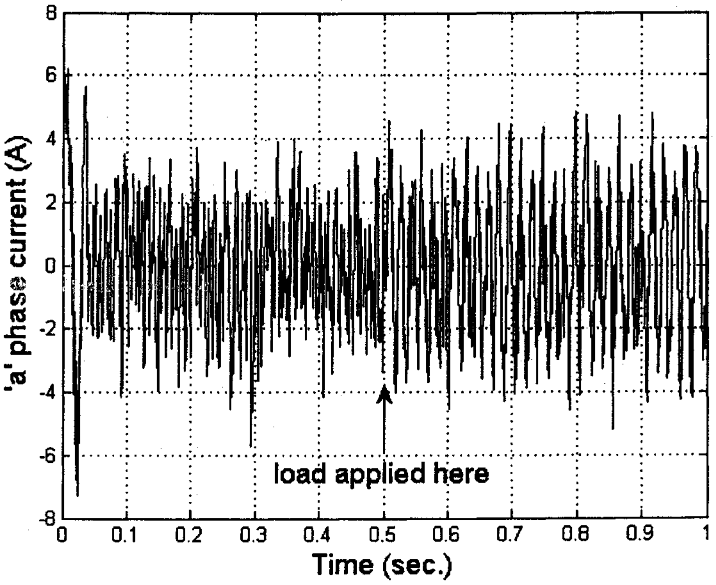


Figure 6.2 Current response in phase 'a' of ANN controller-based IPMSM drive at 1000 Hz sampling frequency to sudden application of full-load (0 → 2 Nm) at rated speed command.

The experimental implementation of a complete SAN controller-based IPMSM drive, utilizing the MTPA scheme for base speed and below base speed operation, and the FW mode for above base speed operation, is presented in this chapter. In addition a PID-based controller, utilizing the $i_d^r = 0$ approximation, has been implemented for comparison purposes.

The drives have been realized by use of a Pentium II 300 MHz personal computer (PC) interfaced with a current-controlled BJT voltage source inverter (VSI) through a dSPACE DS1102 DSP controller board. This chapter presents the detailed implementation of both hardware and software programming. In order to validate the efficacy of the SAN controller-based drive, various experimental results under different dynamic operating conditions are presented and discussed.

6.2 Experimental Setup

6.2.1 Hardware Implementation

The hardware schematic for real-time implementation of the IPMSM drive is shown in Figure 6.3. The DSP board is installed in a PC with uninterrupted communication capabilities through dual-port memory. The DS1102 board is based on a Texas Instruments (TI) TMS320C31 32-bit floating point digital signal processor. The DSP has been supplemented by a set of on-board peripherals used in digital control systems, such as analog to digital (A/D) and digital to analog (D/A) converters, and incremental encoder interfaces. The DS1102 board has one 4-channel (two 12-bit and two 16-bit) A/D converter, one 4-channel (12-bit each channel) D/A converter and two 16-bit incremental encoders.

The DS1102 is also equipped with a TI-TMS320P14 16-bit micro controller DSP that acts as a slave processor and provides the necessary digital input/output (I/O) ports and powerful timer functions such as input capture, output capture and PWM waveform generation. In this work, the slave processor is used only for digital I/O configuration.

The actual motor currents are measured by Hall effect sensors, which have acceptable frequency responses, and are fed to the DSP board through the A/D converter. As the motor neutral is not grounded, only two phase currents are fed back and the other phase current is calculated from them.

The rotor position angle is measured by an absolute incremental encoder mounted on the rotor shaft via a flexible coupler and fed to the DSP board through an encoder interface. The encoder generates 4096 pulses per revolution, which are fed to the incremental encoder interface of the board. By using a 4-fold pulse multiplication the output of the encoder is increased to 4x4096 pulses per revolution in order to produce better resolution. A 24-bit position counter is used to count the encoder pulses. The counter is reset once per revolution by the index pulse generated from the encoder. The motor speed is computed from the measured rotor position angles using numerical differentiation.

In order to implement the vector control algorithm, the fixed-band hysteresis controller was used as a current controller. The command currents are generated from the speed controller as outlined in Chapter 2. The hysteresis current controller compares the command currents with the corresponding actual motor currents and generates the logic signals, which act as firing pulses for the inverter switches. Thus, these six PWM logic signals are the output of the DSP board and are fed to a base drive circuit for the bipolar junction transistor (BJT) or insulated gate bipolar transistor (IGBT) inverter power module. The base drive circuit is used to provide isolation between the high power supply and the low power logic circuits and, in the case with the IGBT inverter, to improve the voltage level so that it is sufficient to drive the inverter. Although a BJT-based VSI was used for the results in this work, a base-drive circuit for an IGBT-based inverter was implemented and used with only limited success due to the lack of proper snubber circuitry for use with the IPMSM in our IGBT-based inverter. The dc power supply for the inverter was obtained by rectifying the ac power through a variac.

In order to provide loading, the 1 hp laboratory IPMSM was coupled to a LabVolt dynamometer using a belt. The dynamometer serves as a simple

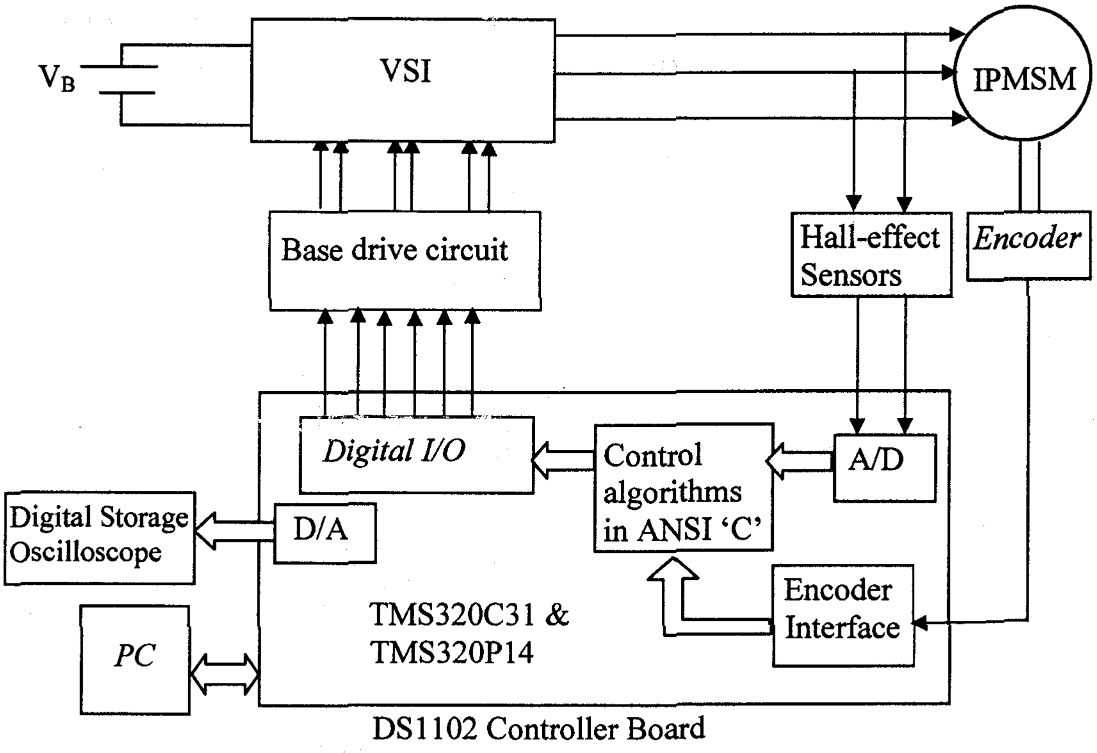


Figure 6.3 Hardware schematic for experimental implementation of the IPMSM drive.

mechanical load on the motor. A digital storage oscilloscope was used to capture the desired signals through the D/A port of the DSP board.

6.2.2 Software Implementation

Software was developed for the SAN-based speed controller, approximated MTPA and FW methods of command current calculation, and the fixed-band hysteresis current controller for the IPMSM drive in the ANSI 'C' programming language. The fastest sampling frequency allowed by the DSP board was 5 kHz, which is adequate for successful real-time implementation of the drive with SAN-based speed controller, and is within the capabilities of the BJT inverter. The detailed software implementation is described below.

Step 1 (Initializing the DSP board): All the peripherals of the DSP board are initialized as shown in Table 6.1. Macro functions supplied with the DS1102 controller board achieve this task. The macro function *init()* initializes the D/A converter subsystem of the DS1102 for output. This command also resets the interrupt request bits and calibrates the 16-bit A/D converters. The function *init_slave_DSP_digital_i_o()* initializes the I/O ports for use with outputs. The function *ds1102_inc_clear_counter(1)* clears the incremental encoder counter. The function *start_isr_t0(Ts)* is used to initialize the DS1102-TMS320C31's on-chip timer to generate timer interrupts at a predefined sampling rate, Ts – this is essential for real-time implementation. The detailed initialization and the I/O functions for the D/A and A/D converters and 16-bit I/O ports can be obtained from the reference manual of the DS1102 [90].

Table 6.1 Peripheral initialization.

```
void main()
{
init(); /*init DAC mode, calibrate ADCs */

init_slave_DSP_digital_i_o();  /*initialize i/o ports for output*/

*error=NO_ERROR;               /* initialize overload error flag*/

dp_mem[0].f=0.0;               /* init 1st dp-mem loc for float */

dp_mem[1].f=0.0;               /* init 2nd dp_mem loc for float */

ds1102_inc_clear_counter(1);   /* clear incr. encoder counter */

start_isr_t0(Ts);              /*initialize sampling clock timer*/

while(*error==NO_ERROR);       /* background process   */
}
```

Step 2 (Determining motor currents and rotor speed): After initializing all the required variables, an interrupt service routine (isr_t0) is used to read the actual motor currents and rotor position angles at the 5 kHz sampling frequency, following the steps shown in Table 6.2. In this table, the constants K_a and K_b are used to get the actual motor currents through A/D channels 3 and 4. These constants depend on the Hall-effect sensors used. The incremental gain (INCG) is used to get the actual rotor position depending on the incremental encoder. In this work, the constants K_a, K_b and INCG have been determined to be 10.714, 11.985 and 3216.990494, respectively. The rotor speed is calculated by differentiating the present and past samples of rotor position. The command torque is generated by the SAN-based speed controller from the command speed, the error between the command speed and present sample of actual speed, and the change in this speed error since the previous interval.

Table 6.2 Interrupt service routine to read the motor currents and rotor position.

```
void  isr_t0()
 {
      .
      .

   ds1102_ad_start();
   i_a = Ka*ds1102_ad(3);
   i_b = Kb*ds1102_ad(4);
   i_c = -(i_a+i_b);
   THI = ds1102_inc(1);
   TH = THI*INCG;
   θr = P*TH;
      .
      .
      .
 }
```

Step 3 (Reference Command Torque calculation): From the phase currents, as obtained from the Hall effect sensors (and calculation, in the case of the 'c' phase current), the d-q axis currents are calculated via Equation (2.11). Equation (2.12) is then used to convert these currents to the rotating d^r-q^r reference frame. Then i_q^r and i_d^r are used to calculate the rotating d^r-q^r axis magnetizing currents, i_{qm}^r and i_{dm}^r, via Equations (2.26) and (2.27), respectively. From this, Equations (3.16) and (3.15) are used to calculate the electrical torque T_e and load torque T_L at the current sampling iteration, respectively. Equation (3.17) is then used to calculate the reference command torque T_{ref}, from T_L, command speed ω_r^*, and the current sample value of speed error divided by the sampling time (an inertial factor).

This process is outlined in Table 6.3.

Table 6.3 'C' code for T_{ref} calculation.

```
iq = 2.*i_a/3. - i_b/3. - i_c/3.;
id = -i_b/SQR3 + i_c/SQR3;

iqr = iq*cos(θ_r) - id*sin(θ_r);
idr = iq*sin(θ_r) + id*cos(θ_r);

iqmr = iqr - P*WR*Ld*idr/Rs - P*WR*phi/Rs;
idmr = idr + P*WR*Lq*iqr/Rs - (P*WR/Rs)*(P*WR/Rs)*Lq*phi;

Te = 0.942*iqmr - 0.11139*idmr*iqmr;
Tl = Te - Bm*WR – Jm*(1./Ts)*DWR;

if (Tl > Tlmax){
     Tl = Tlmax;
     }
if (Tl < -Tlmax){
     Tl = -Tlmax;
     }

T_ref = Tl + Bm*WRC + Jm*(Kref/Ts)*DW;
```

Step 4 (SAN Command Torque calculation and real-time SAN training):

Initial weights and biases: The normalized motor speed (actual speed divided by command speed) is passed to the SAN and a check is done to determine if the motor is at startup (as indicated by $\omega_r/\omega_r^* = 0$). If that is the case, then the arbitrarily chosen initial weights and bias are used for the first pass of the inputs through the SAN. This is outlined in Table 6.4. The initial neuron weights and bias used are as follows:

W1 = 1

W2 = 1

W3 = 1

B = 0

Table 6.4 'C' code for initial neuron weights and bias.

```
if (In1 == 0)  /*setting initial weights and bias*/
    {
    W1 = 1;
    W2 = 1;
    W3 = 1;
    B = 0;
            -
            -
            -
}
```

Weights and bias updating based on speed error: Command speed ω_r^*, speed error e and change in speed error Δe are normalized by dividing by command speed and used as inputs to the SAN. The weights and bias of the SAN are updated, by back-propagation, whenever the speed error exceeds the chosen threshold. The threshold used was an error of 0.1 rad./sec. If this is not exceeded, the weights and bias from the previous iteration are used again in the present iteration.

If the speed error threshold is exceeded, SAN weights and bias are updated using speed error as per the back-propagation training algorithm outlined in

Chapter 3, sections 3.2.3 and 3.3. The ANSI 'C' code used for this is presented in Table 6.5.

Table 6.5 'C' code for speed error-based SAN retraining.

```
/* speed error training section */

if ( fabs(DW) > 0.1 )      /* retrain only if speed error */
                            /* is greater than 0.1 */
{

   fo = (T_ANN-T_old)/(Sum-Sum_old);
   sig_o = In2*fo;

   W1_old = W1;
   W2_old = W2;
   W3_old = W3;
   dW1_old = dW1;
   dW2_old = dW2;
   dW3_old = dW3;
   dW1 = lr1*sig_o*In1;
   dW2 = lr1*sig_o*In2;
   dW3 = lr1*sig_o*In3;
   W1 = W1_old + m1*dW1_old + dW1;
   W2 = W2_old + m1*dW2_old + dW2;
   W3 = W3_old + m1*dW3_old + dW3;

   B_old = B;
   dB_old = dB;
   dB = lr1*sig_o;
   B = B_old + m1*dB_old + dB;
         -
         -
         -
}
```

Calculation of Command Torque: After the weights and bias for the SAN are determined the normalized values of command speed, speed error and change in speed error are fed to the SAN and command torque is calculated by feed-forward propagation through the single neuron. These calculations are shown in Table 6.6.

Table 6.6 'C' code for T_{ANN} calculation.

```
Sum_old = Sum;
Sum = W1*In1 + W2*In2 + W3*In3 + B;

T_old = T_ANN;
T_ANN = ((1-exp(-Sum))/(1+exp(-Sum)))*Tmax;
```

Weights and bias updating based on torque error: To ensure drive stability, if the output command torque of the SAN is not within an acceptable deviation from the reference command torque, then T_{ref} and *not* T_{ANN} will be used to calculate the command currents applied to the motor terminals. Retraining will then be repeatedly executed at each iteration to update the weights and bias of the SAN, by back-propagation, until the maximum torque error threshold is no longer exceeded. As outlined in Chapter 4, section 4.2.1, if the torque error, as expressed by $\left|\frac{T_{ANN}(n)-T_{ref}(n)}{T_{ref}(n)}\right|$, exceeds 0.10 then retraining will be done to return T_{ANN} to within 10% of T_{ref}. The ANSI 'C' code used for this is presented in Table 6.7.

Table 6.7 'C' code for torque error-based SAN retraining.

```
/* torque error training section */

if ( fabs((T_ANN-T_ref)/T_ref) > .1 )          /* retrain only if torque error */
                                                /* is greater than 0.1 */
{

   fo = (T_ANN-T_old)/(Sum-Sum_old);
   sig_o = (T_ref-T_ANN)*fo;

   W1_old = W1;
   W2_old = W2;
   W3_old = W3;
   dW1_old = dW1;
   dW2_old = dW2;
   dW3_old = dW3;
   dW1 = lr2*sig_o*In1;
   dW2 = lr2*sig_o*In2;
   dW3 = lr2*sig_o*In3;
   W1 = W1_old + m2*dW1_old + dW1;
   W2 = W2_old + m2*dW2_old + dW2;
   W3 = W3_old + m2*dW3_old + dW3;

   B_old = B;
   dB_old = dB;
   dB = lr2*sig_o;
   B = B_old + m2*dB_old + dB;
        -
        -
        -
}
```

Step 5 (Command current calculation): From the command torque, T^*, Equations (2.49) and (2.50) or (2.58) and (2.59) are used to calculate the necessary d^r- and q^r-axis magnetizing currents to produce the desired rotor speed ω_r^*. The MTPA scheme is used at base speed and below and the FW scheme is used for above base speed operation. The appropriate d^r-q^r axis currents are then calculated via Equations (2.28) and (2.29). The command phase currents i_a^*, i_b^* and i_c^* are generated from the d^r-q^r axis command currents and the rotor position angle θ_r using the inverse Park's transformation. This procedure is outlined in Table 6.8.

Table 6.8 ‘C’ code for command current calculation.

```
/************************************************************************
 * MTPA and FW schemes *
************************************************************************/

    if(WRCC > W_SYNC) {
      idmrC = -7.39868 + (11.7813/WRCC)*Vo;
      iqmrC = TEC/(1.76614 – (1.31232/WRCC)*Vo);
                }
    else {
      iqmrC = 1.06157*TEC;
      idmrC = -0.11825*iqmrC*iqmrC;
            }

    iqrC = iqmrC + P*WRCC*Ld*idmrC/Rs + P*WRCC*phi/Rs;
    idrC = idmrC - P*WRCC*Lq*iqmrC/Rs;

    iqC = iqrC*cos(θr) + idrC*sin(θr);
    idC = -iqrC*sin(θr) + idrC*cos(θr);

    IAC = iqC;
    IBC = -0.5*(SQR3*idC+iqC);
    ICC = -(IAC+IBC);
```

Step 6 (Command current application): The command phase currents are compared with the actual motor currents in the hysteresis current controller, which provides the necessary driving pulses for the inverter switches. The hysteresis current controller algorithm is also written in the ANSI 'C' programming language, as outlined in Table 6.9. The digital I/O ports are configured as output ports for the six PWM logic pulses, which are fed to the isolation and base-drive circuits of the 3-phase voltage source inverter module.

Table 6.9 'C' code for command current application.

```
/************************************************************************
 *        hysteresis current controller                                  *
 ************************************************************************/

 if (IAC >= 0.0){
         if (i_a > 1.2*IAC) NA = 0;
         if (i_a < 0.8*IAC) NA = 1;
         }
        else NA = 0;
    if (IAC < 0.0){
             if (i_a < 1.2*IAC) NA1 = 0;
             if (i_a > 0.8*IAC) NA1 = 1;
             }
            else NA1 = 0;
 if (IBC >= 0.0){
         if (i_b > 1.2*IBC) NB = 0;
         if (i_b < 0.8*IBC) NB = 1;
         }
        else NB = 0;
   if (IBC < 0.0){
           if (i_b < 1.2*IBC) NB1 = 0;
           if (i_b > 0.8*IBC) NB1 = 1;
           }
           else NB1 = 0;
 if (ICC >= 0.0){
         if (i_c > 1.2*ICC) NC = 0;
         if (i_c < 0.8*ICC) NC = 1;
         }
        else NC = 0;
    if (ICC < 0.0){
             if (i_c < 1.2*ICC) NC1 = 0;
             if (i_c > 0.8*ICC) NC1 = 1;
             }
            else NC1 = 0;
         -
         -
         -

 IO = NA+2*NC1+4*NB+32*NA1+64*NC+128*NB1;     /* Output signals from I/O */
 ds1102_p14_write_io_register(0,0,IO);
```

Steps 1-6 are summarized in the flow chart of Figure 6.4.

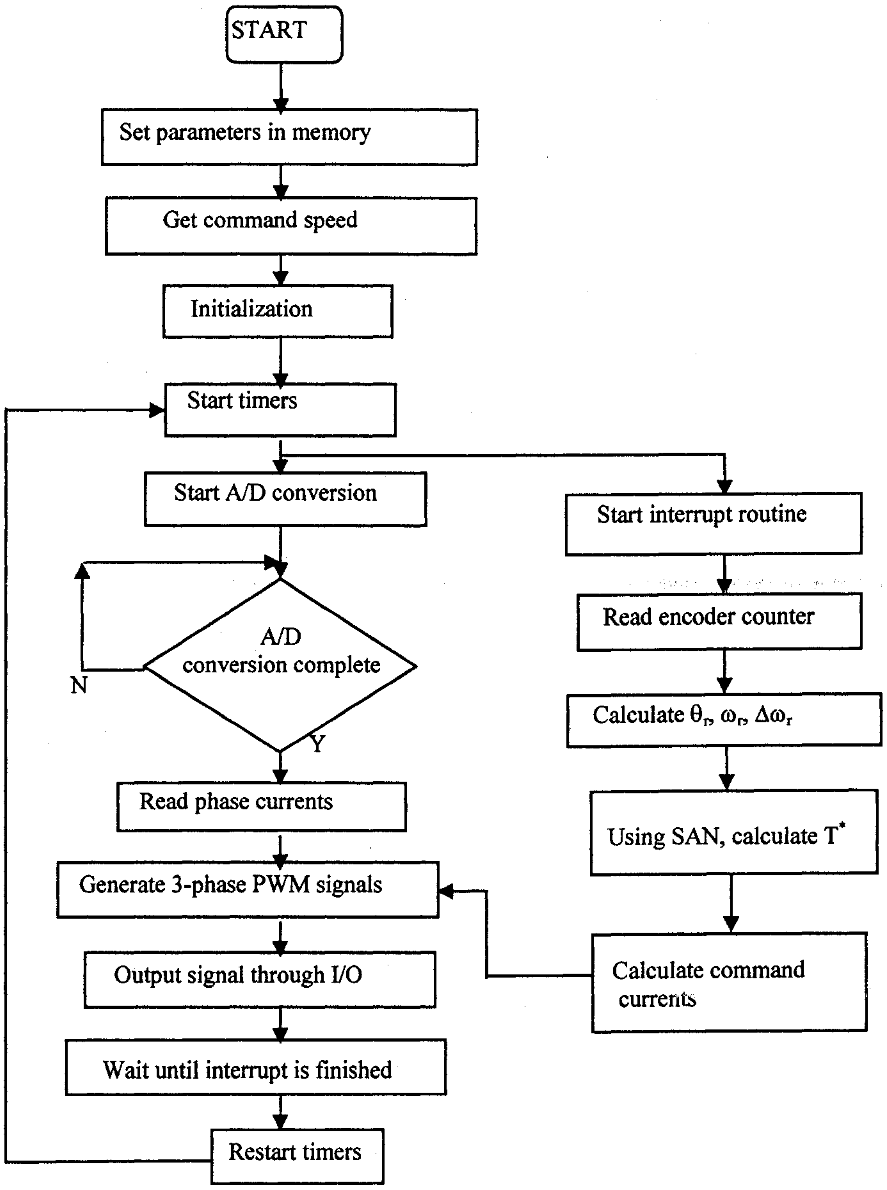


Figure 6.4 Software flow chart for real-time implementation of the SAN controller-based IPMSM drive.

The program is compiled by the TI 'C' compiler and downloaded to the DS1102 DSP board using the dSPACE loader program down1102.

The entire experimental setup is shown in Figure 6.5, where A denotes isolation and base drive circuits for the voltage source inverter, B donates a digital storage oscilloscope, C denotes Hall effect sensors, D denotes the dynamometer used for loading, E denotes the rectifier circuit to produce dc voltage for the VSI, F is the ac voltage source, G denotes the position sensor on the motor rotor and H denotes the voltage source inverter.

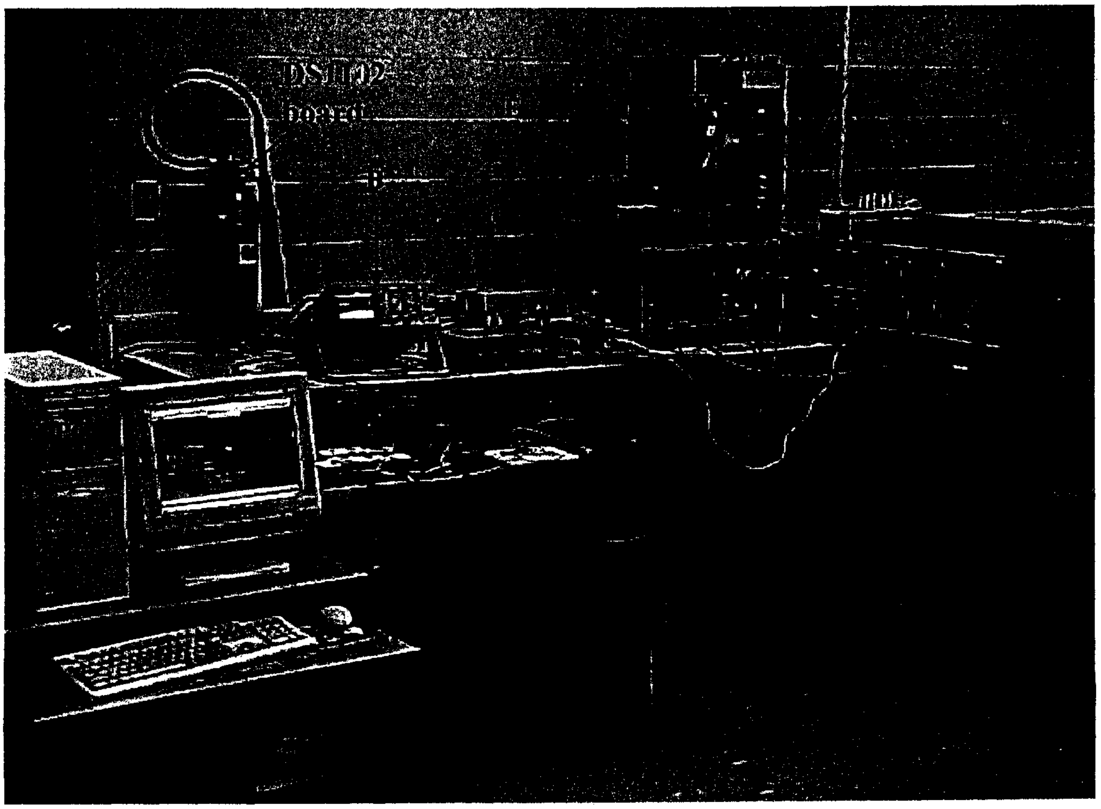


Figure 6.5 Experimental setup for the IPMSM drive.

6.3 Experimental Results and Discussion

Please note that in the following experimental results, unless otherwise noted, the time scale for speed responses is 0.5 seconds per horizontal division – for a full-graph time span of 5 seconds. The time scale for simulated speed response tests was 1.0 second for full-graph time span. Experimental current responses were taken on varying horizontal division scales, whereas simulation results were taken on either 1.0-second or 0.2-second full-graph time spans. These differences are due, in part, to the operational nature of the manually controlled experimental setup. Experimental startup responses are not as rapid as simulation results would indicate. This is because the experimental setup dictates that voltage be applied to the inverter manually in a ramp-like fashion and not instantaneously as is the case with simulation. In addition, the laboratory voltage supply was not capable of supplying the full 254.75 Vdc to the VSI that is required to produce rated votage at the motor terminals.

6.3.1 Rated Speed and Below Operation (with MTPA)

The startup speed response of the SAN controller-based IPMSM drive under light load to rated speed command (188.5 rad./sec.) is shown in Figure 6.6 (a). The results show that the drive achieves the command speed promptly, with no observable overshoot or oscillation, and accurately maintains command speed. Figure 6.6 (b) shows the corresponding startup 'a' phase current for the SAN-based drive under the same conditions. It is clearly seen that current amplitude is maximum during motor acceleration and then settles to within rated values. Figures 6.7 (a) and (b) show the steady-state phase current response of the SAN-based drive under light-load and rated command speed. The current waveform is not a clean sinusoid – primarily due to the hysteresis current controller and the reactive nature of the SAN controller. These results agree well with those predicted in simulation.

Figure 6.8 (a) shows the startup speed response of a PID-based IPMSM drive at light load and rated speed command (188.5 rad./sec.). The drive experiences an initial overshoot of command speed. This is typical for such a controller. Figure 6.8 (b) shows the steady-state 'a' phase current response of a PID-based IPMSM drive at light load and rated command speed (188.5 rad./sec.). The waveform is sinusoidal but experiences higher frequency variations due to the hysteresis current controller.

Figure 6.9 (a) shows the speed response of the SAN-based IPMSM drive to step changes of command speed at light load. The drive is seen to reach and maintain command speeds quickly and accurately during step increases (0 → 90 rad./sec. → 188.5 rad./sec.) and over a step decrease (188.5 rad./sec. → 130 rad./sec.) of command speed. There are no visually detectable over- or undershoots or steady-state speed errors. Figure 6.9 (b) shows the phase current response of the SAN-based IPMSM drive to step changes of command speed (90 rad./sec. → 188.5 rad./sec. → 90 rad./sec.). As the MTPA scheme is used to achieve these speeds, phase current magnitude does not appear to change significantly with motor speed (as load torque has remained constant in this case). Current frequency, however, is clearly seen to increase with increasing motor speed. A software program (Dspeed.c), included in Appendix D, was used to achieve real-time step changes of command speed.

Figures 6.10 (a) and (b) show the speed responses of the SAN and PID controller-based IPMSM drives to sudden applications of full-load at rated speed

(188.5 rad./sec.). Speed disturbance is practically undetectable with the SAN-based drive. The PID-based drive shows a clear disturbance of command speed due to the suddenly applied load, though command speed is recovered in approximately 0.65 seconds.

Figures 6.11 (a) and (b) show the phase current responses of the SAN-based IPMSM drive to sudden application of full-load at rated speed (188.5 rad./sec.). Increased phase current amplitude with the application of load is apparent. While these results generally show agreement with simulation results, it is clear from these and the previous experimental results that the experimental setup draws a higher current under no-load conditions, and current amplitude doesn't appear to increase as dramatically under loading as in simulation. This can be explained by the operational nature of the dynamometer used to load the motor. The dynamometer is coupled to the motor rotor by a belt, and even under a zero load setting the motor is subjected to loading due to the friction and inertia of the dynamometer/coupling-belt mechanism, thus leading to higher "no-load" phase currents drawn by the motor. In addition, load is applied manually by turning the control knob. This is imprecise and depends on the operator's judgment, thus applying only approximate load torque values.

Figures 6.12 (a) and (b) show the q- and d- axis current responses of the SAN-based IPMSM drive to sudden application of full-load at rated speed (188.5 rad./sec.). Absolute magnitude increases are visible in both currents when load is applied. Also, i_d appears to be a scaled mirror image of i_q – this is predicted by the nature of the MTPA approximation utilized.

Figure 6.13 (a) shows the startup speed response of the SAN-based IPMSM drive under full-load to rated speed command. The results show that even under an initial full-load the drive quickly reaches command speed and tracks it accurately and precisely, though comparison with Figure 6.6 (a) indicates that the drive requires roughly 67% more time to reach command speed than when only lightly loaded. Figure 6.13 (b) shows the corresponding startup phase current response of the drive under the same conditions. Steady-state phase current remains within rated values.

Figures 6.14 (a) and (b) show the experimental startup speed and phase current responses, respectively, of the SAN-based IPMSM drive at light load to 90 rad./sec. speed command. Again, the drive quickly and accurately tracks command speed and phase current settles to within rated values.

Figure 6.15 (a) shows the speed response of the SAN-based IPMSM drive to a sudden application of full-load at 90 rad./sec. speed command. Disruption to drive speed is visually undetectable. Figure 6.15 (b) shows the corresponding phase current response under the same conditions. As expected, current amplitude increases when load is applied, but remains within rated values.

Figure 6.16 shows the experimental speed response of a PID-based IPMSM drive to a sudden application of full-load at 90 rad./sec. speed command. Disturbance to the drive is clearly visible, but the drive recovers command speed within a little more than 1 second.

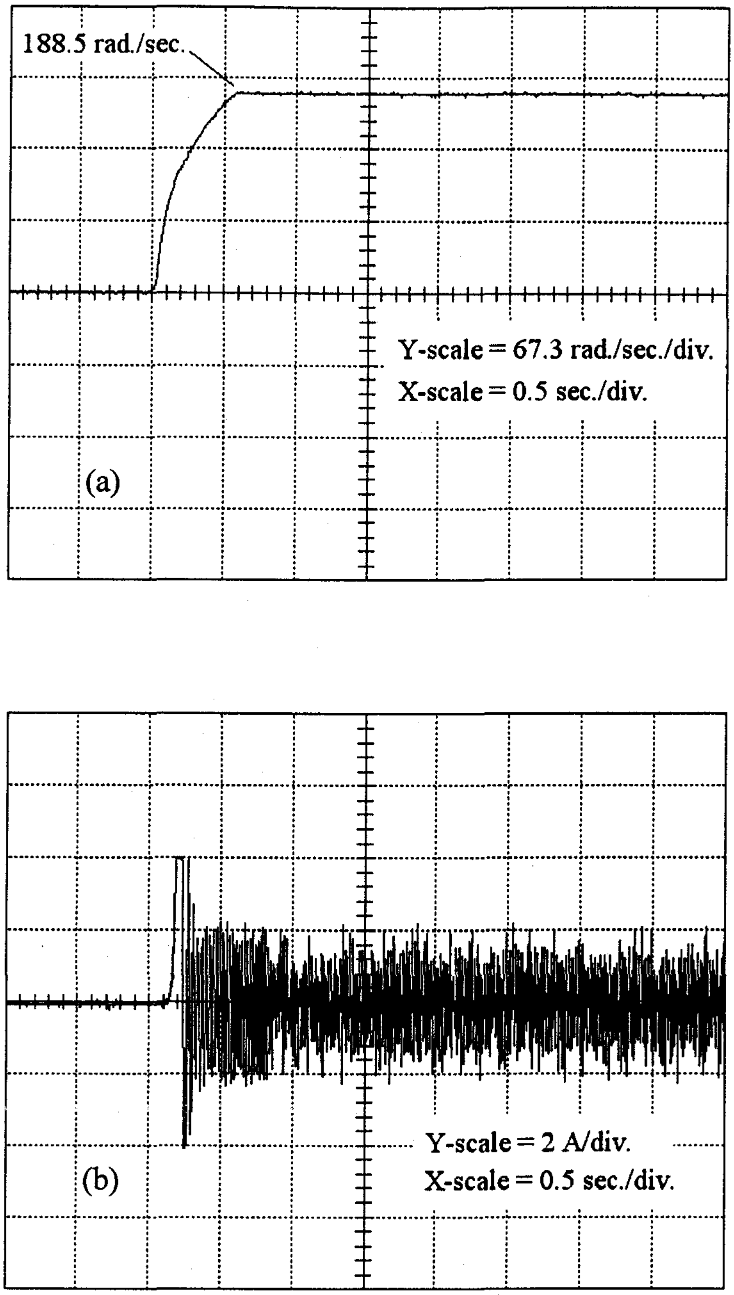


Figure 6.6. Startup responses of SAN-based IPMSM drive at light load and rated speed command: (a) speed and (b) phase current.

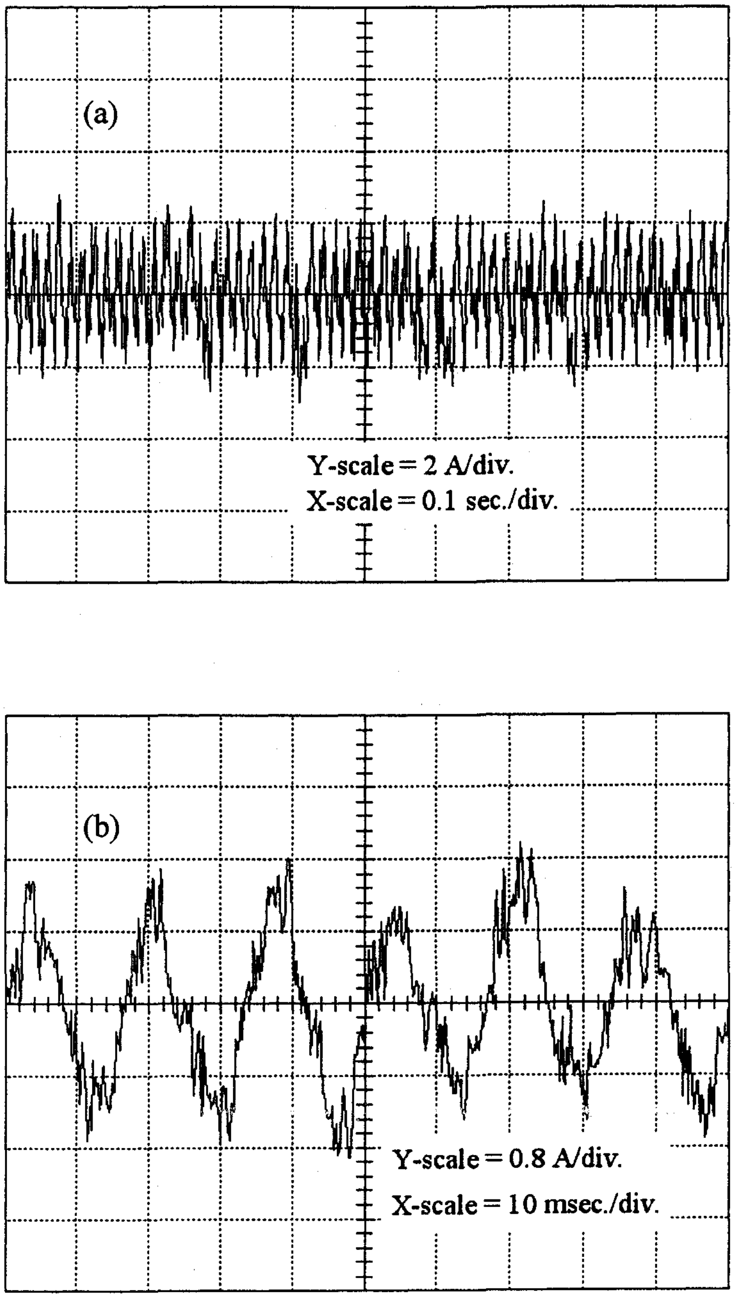


Figure 6.7 (a) and (b). Experimental steady-state phase current responses of SAN-based IPMSM drive at rated speed command.

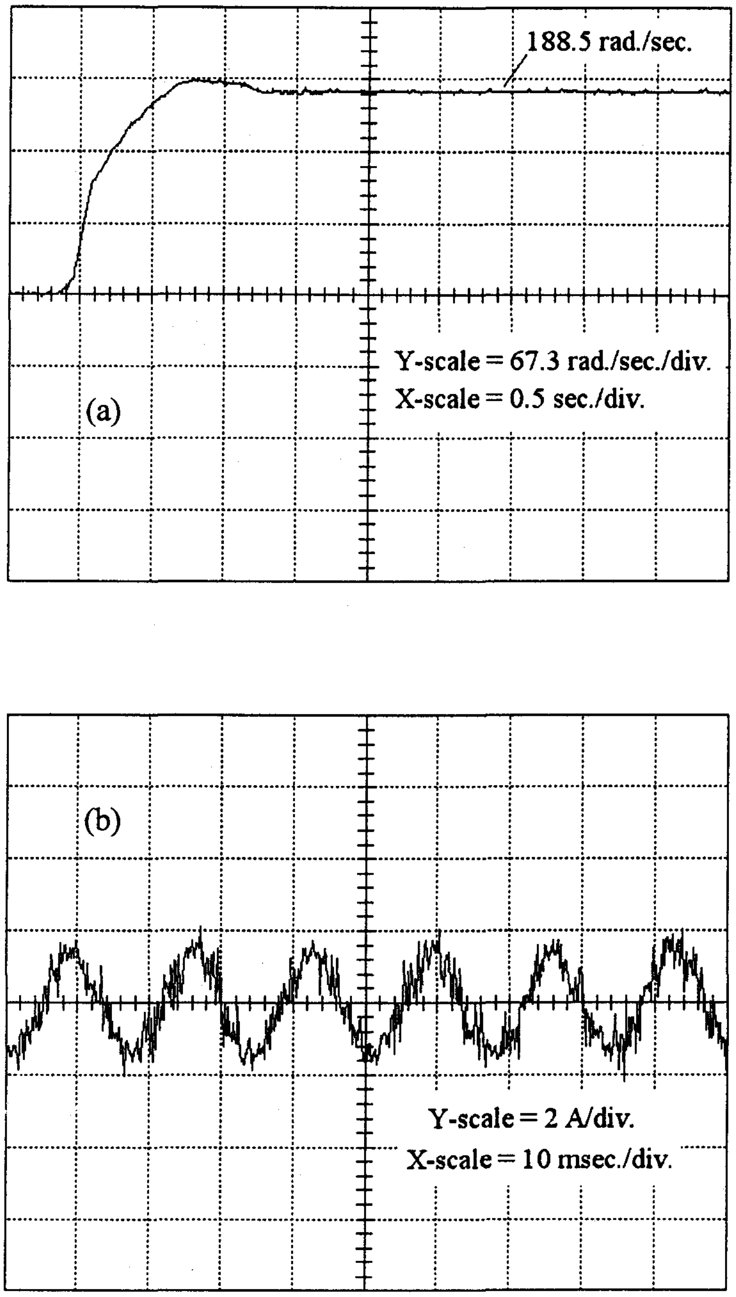


Figure 6.8. Experimental responses of PID-based IPMSM drive at light load and rated speed command: (a) startup speed response and (b) steady-state phase current.

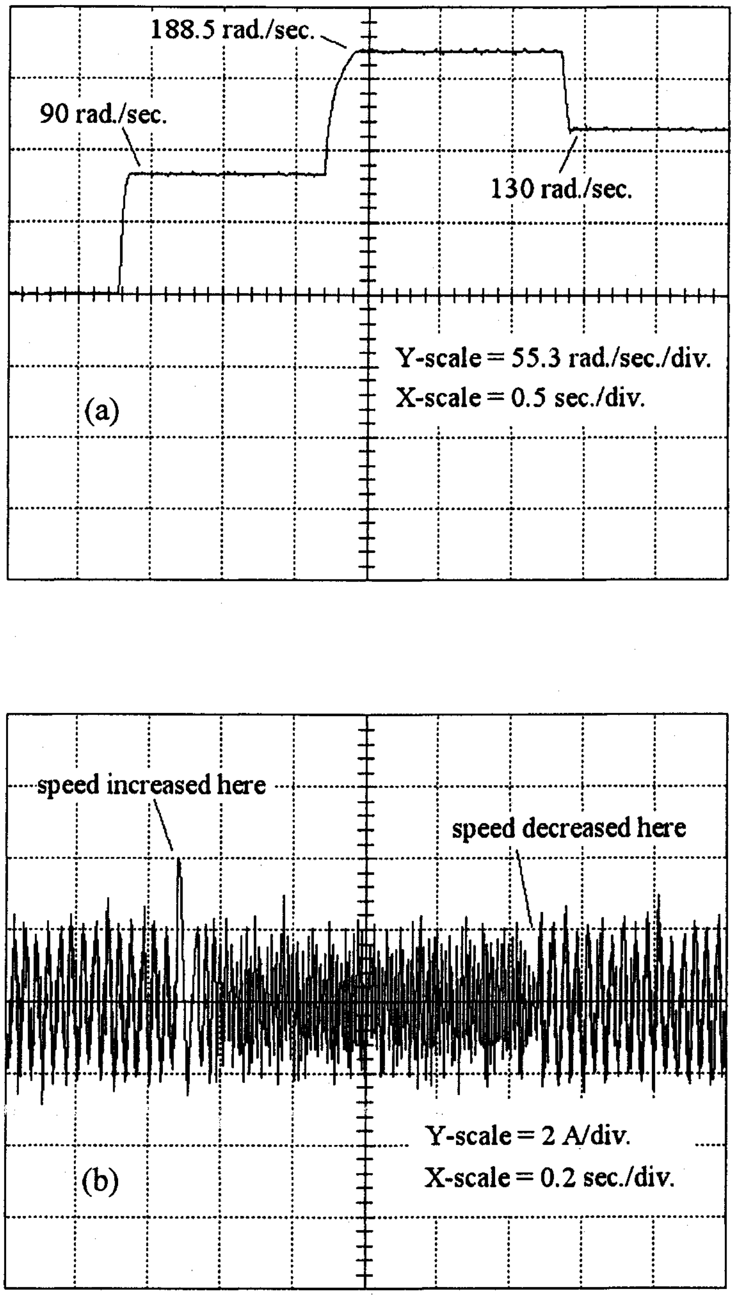


Figure 6.9. Experimental responses of SAN-based IPMSM drive to sudden changes in command speed at light load: (a) speed (90 → 188.5 → 130 rad./sec.) and (b) phase current (90 → 188.5 → 90 rad./sec.).

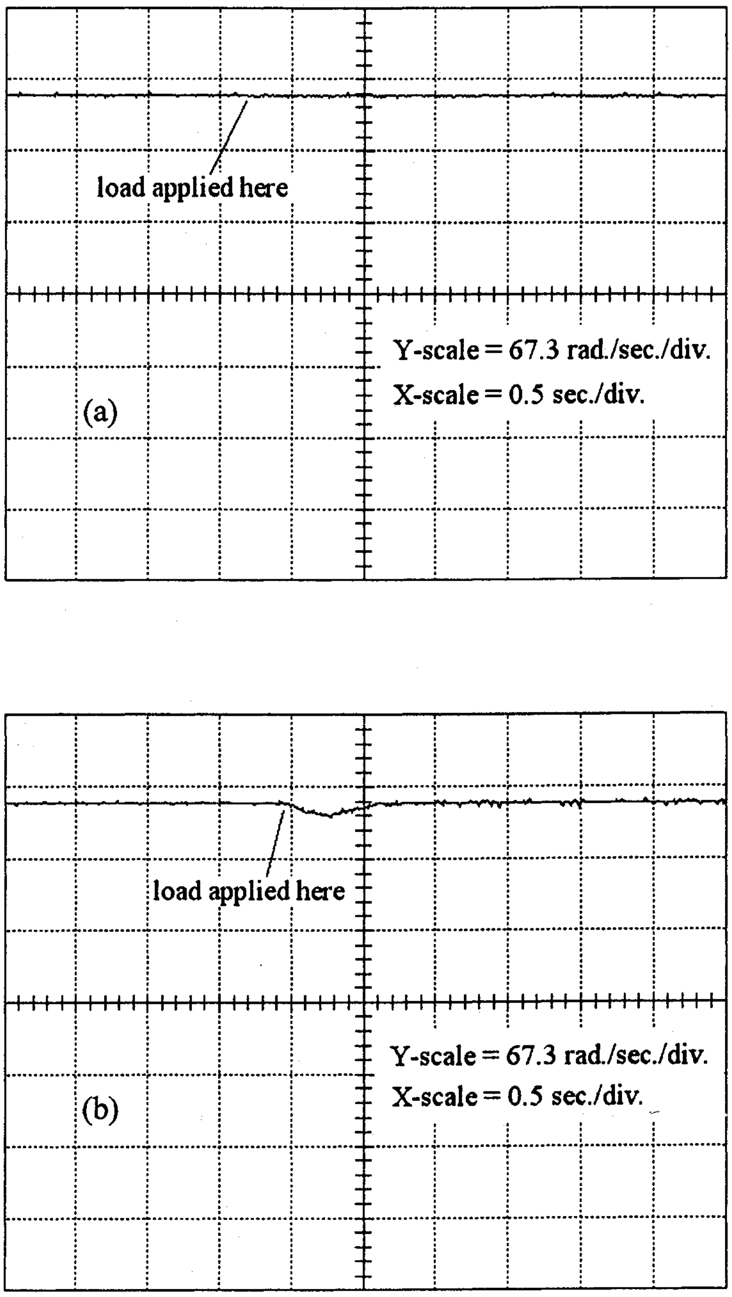


Figure 6.10. Experimental speed responses of IPMSM drives to sudden application of full-load at rated speed (188.5 rad./sec.): (a) SAN-based controller and (b) PID-based controller.

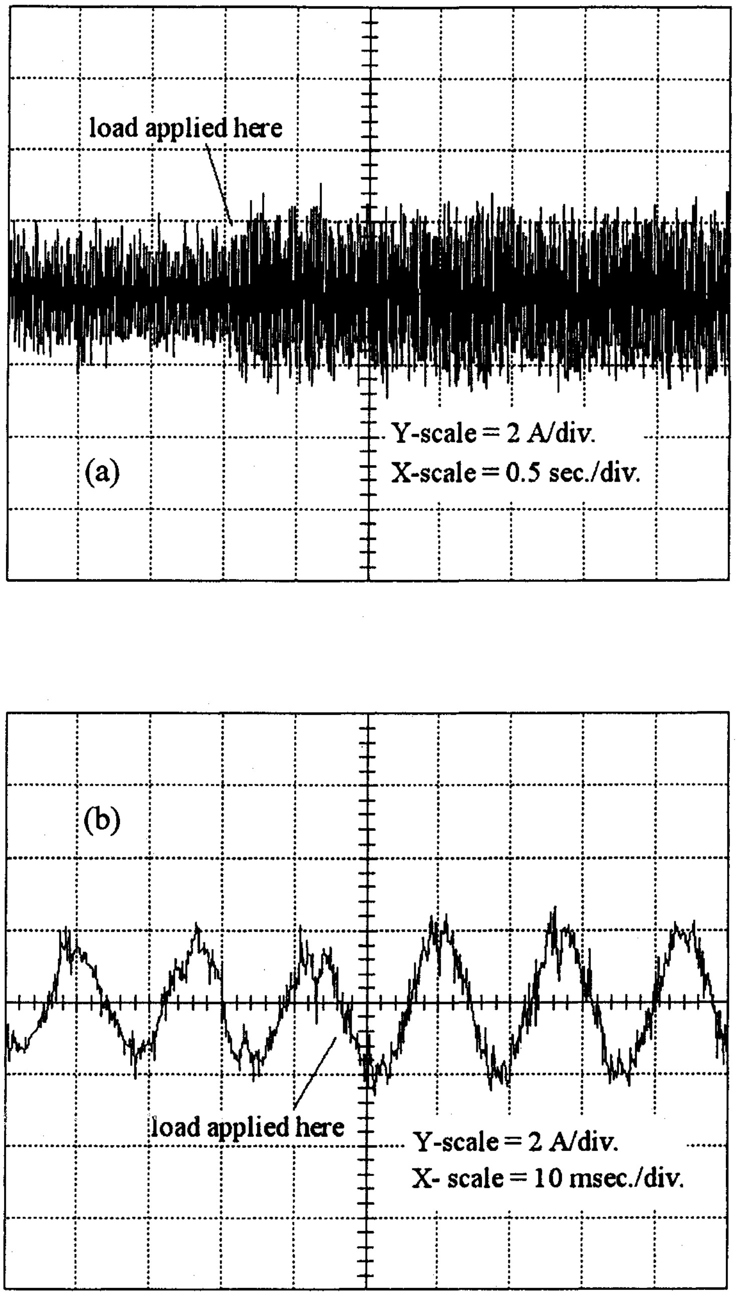
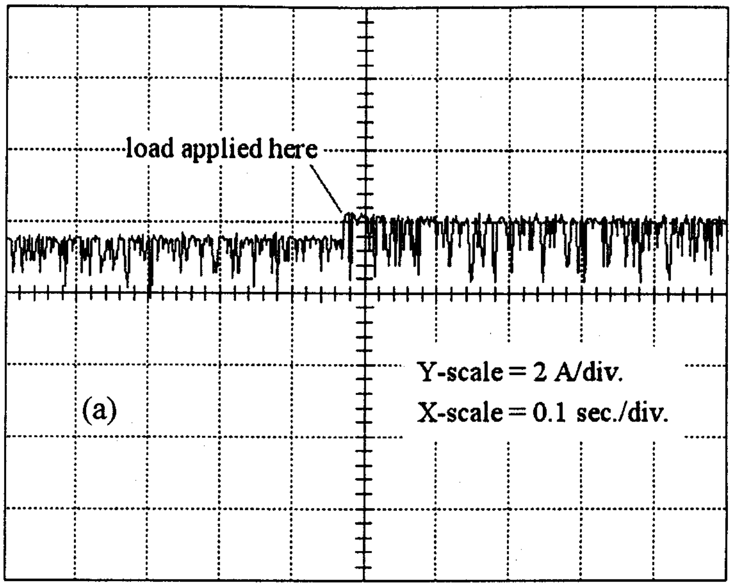


Figure 6.11 (a) and (b). Experimental phase current responses of SAN-based IPMSM drive to sudden application of full-load at rated speed (188.5 rad./sec.).



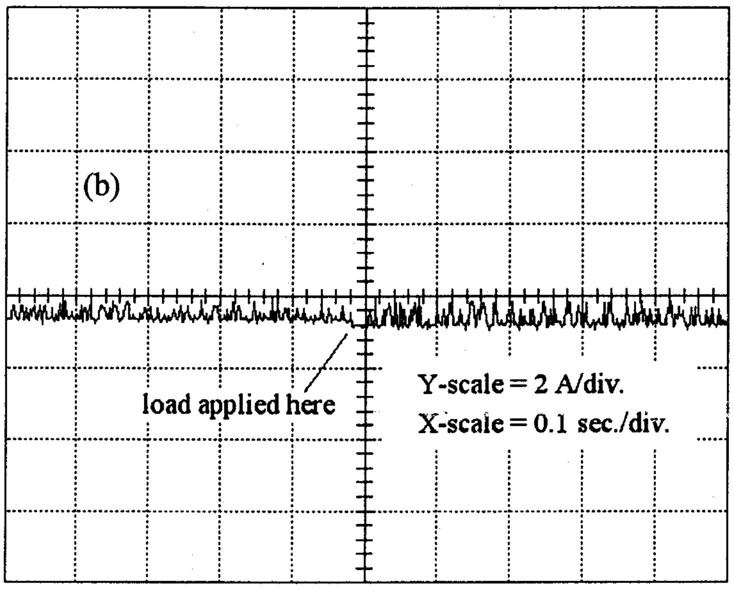


Figure 6.12. Experimental current responses of SAN-based IPMSM drive to sudden application of full-load at rated speed (188.5 rad./sec.): (a) q-axis current and (b) d-axis current.

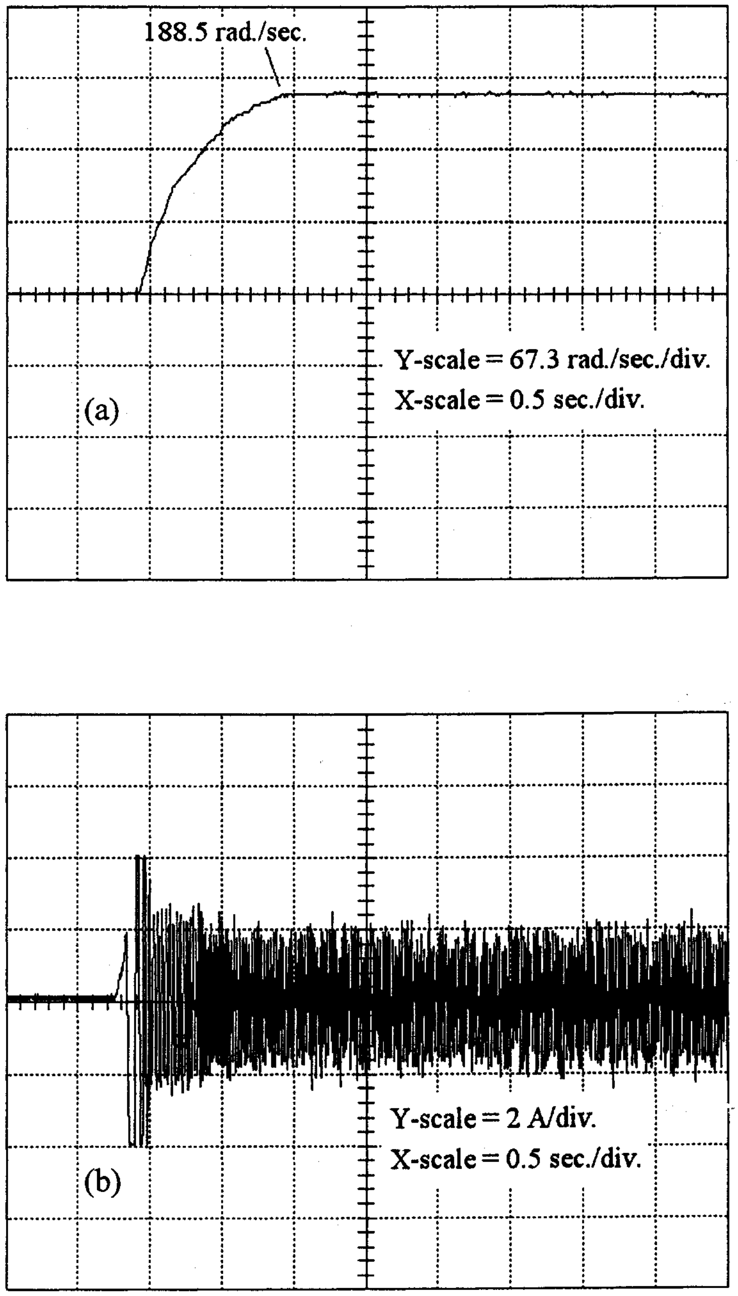
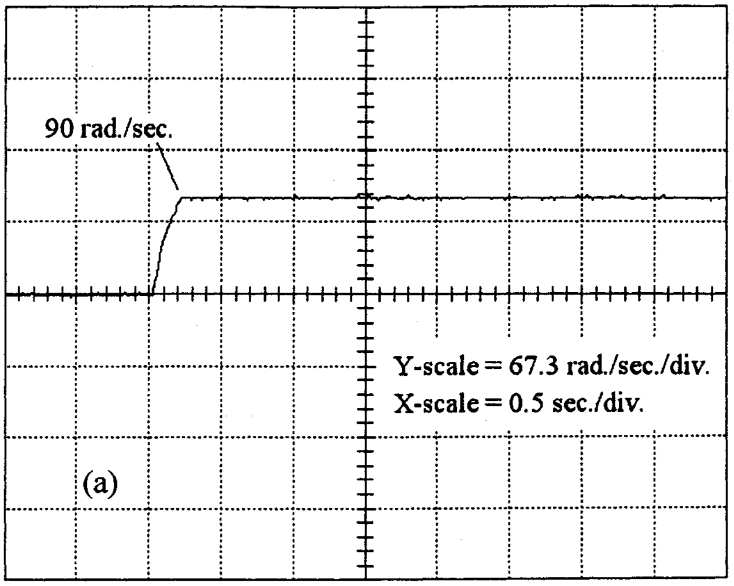


Figure 6.13. Experimental responses of SAN-based IPMSM drive under full-load to rated speed command: (a) startup speed and (b) startup phase current.



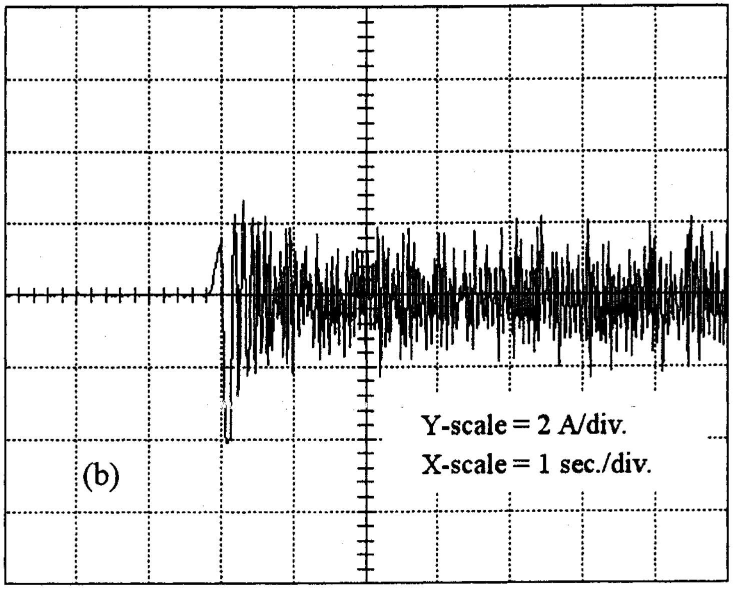


Figure 6.14. Experimental responses of SAN-based IPMSM drive at light load to 90 rad./sec. speed command: (a) startup speed and (b) startup phase current.

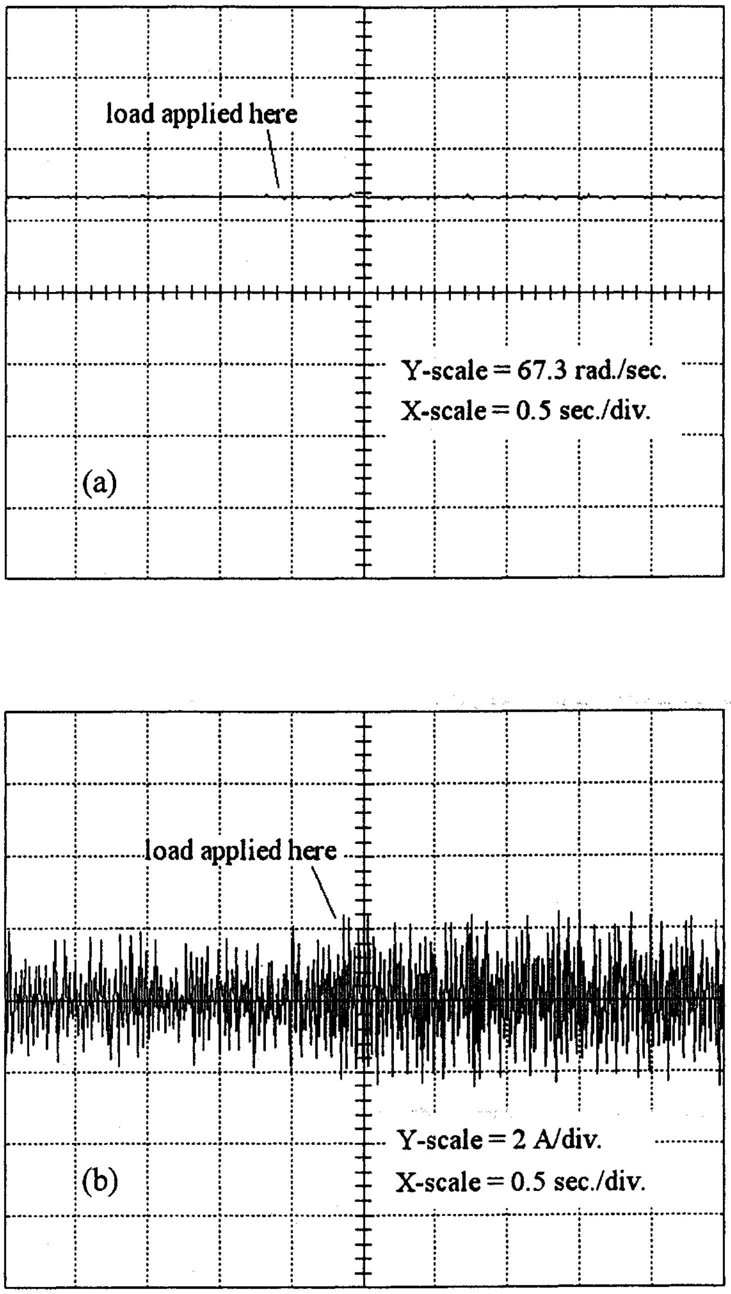


Figure 6.15. Experimental responses of SAN-based IPMSM drive to sudden application of full-load at 90 rad./sec. speed command: (a) speed and (b) phase current.

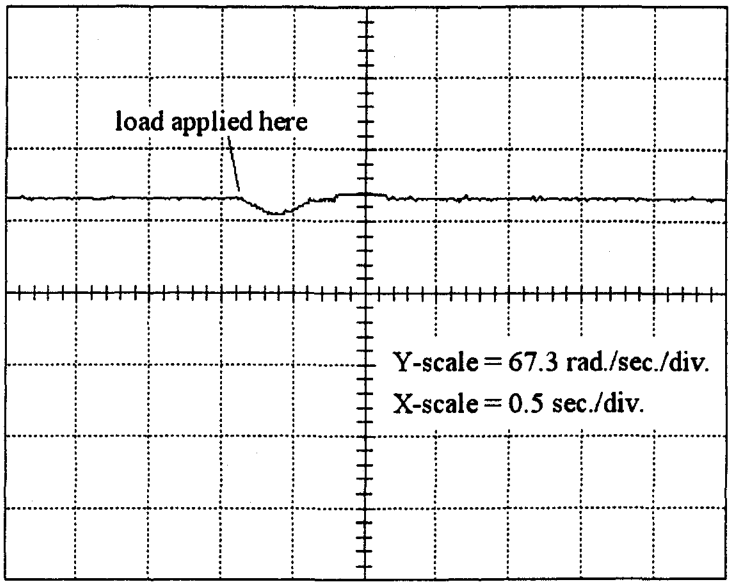


Figure 6.16. Experimental speed response of PID-based IPMSM drive to sudden application of full-load at 90 rad./sec. speed command.

6.3.2 Above Rated Speed Operation (with FW)

Figure 6.17 shows the startup speed response of the SAN-based IPMSM drive at light load to 250 rad./sec. speed command, utilizing the FW mode of operation. Command speed is quickly achieved and maintained. Figures 6.18 (a) and (b) show the startup and steady-state phase current responses, respectively, of the SAN-based IPMSM drive at light load to 250 rad./sec. speed command. Current response is roughly sinusoidal and remains within rated values. Again, a clean sinusoid is not achieved, primarily due to the reactive nature of the SAN controller and the hysteresis current controller. The results agree with those predicted in simulation.

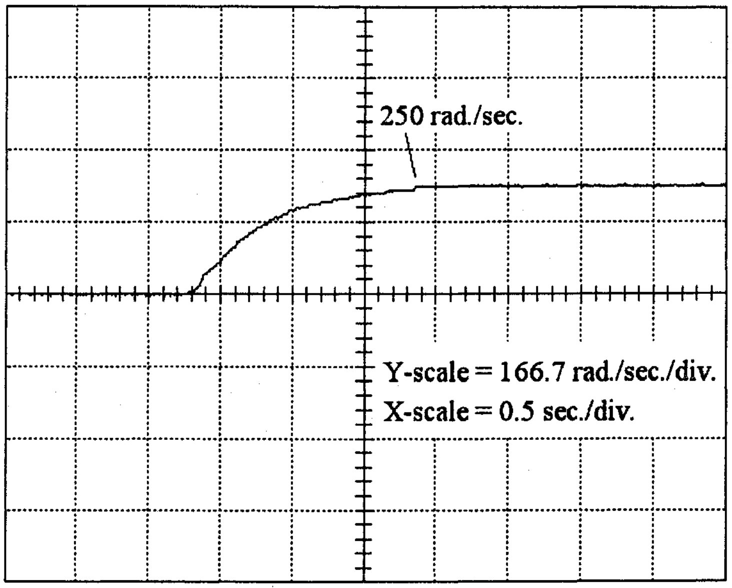


Figure 6.17. Startup speed response of SAN-based IPMSM drive at light load to 250 rad./sec. speed command.

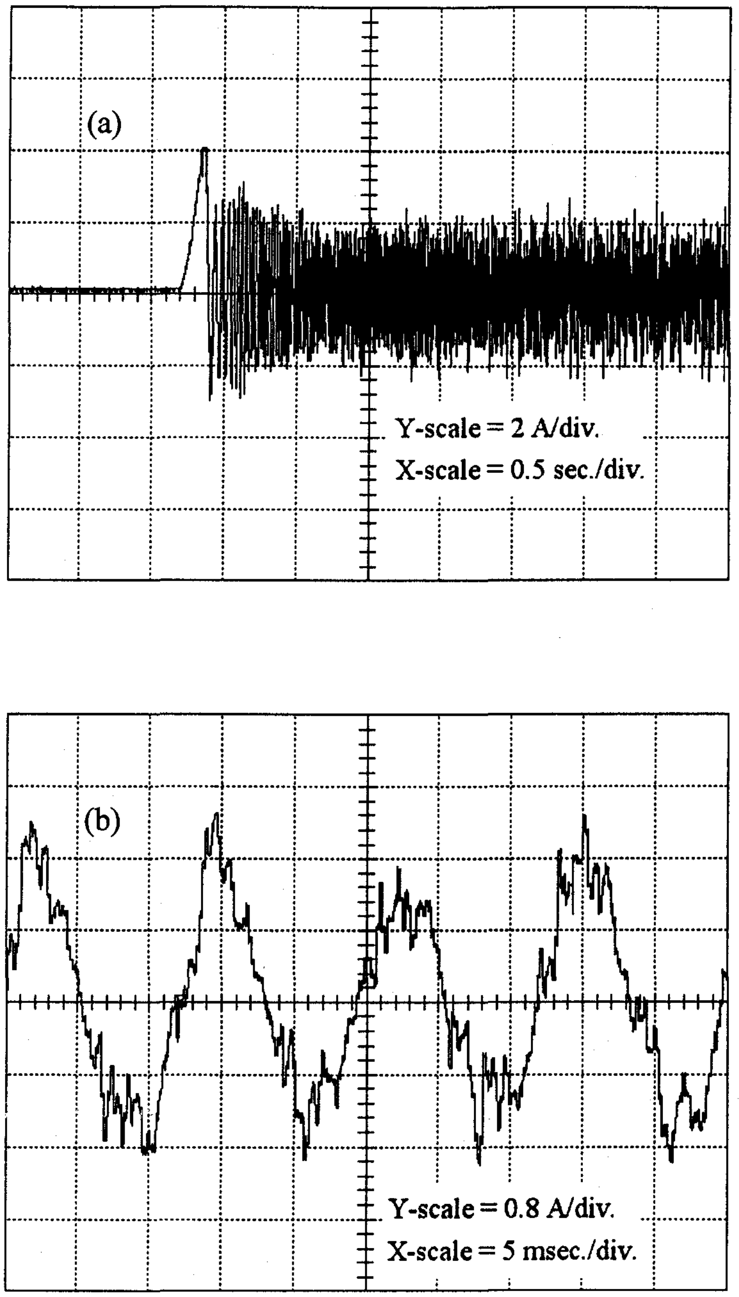


Figure 6.18. Experimental phase current responses of SAN-based IPMSM drive at light load to 250 rad./sec. speed command: (a) startup phase current response and (b) steady-state phase current.

6.4 Concluding Remarks

The detailed DSP based real-time implementation procedure for the SAN speed controller with approximated MTPA and FW modes of operation for the IPMSM drive has been presented in this chapter. The complete drive has been implemented, through both hardware and software, using a PC equipped with a digital signal processor board on a laboratory 1 hp interior permanent magnet synchronous motor.

The performance of the proposed IPMSM drive has been investigated experimentally at different dynamic operating conditions such as sudden changes of command speed and under varying loads. The drive is found to be robust for use in high performance drive applications. The experimental results have validated the simulation results presented in Chapter 5.

Chapter 7

Conclusions

7.1 General

When coupled with an appropriate controller, the interior permanent magnet synchronous motor (IPMSM) can be a favourable choice for use in variable speed high performance electric motor drives. Speed controllers for the IPMSM, however, typically require precise knowledge of machine parameters, which, in the case of the IPMSM, are not constant under dynamic operating conditions. Q-axis inductance, in particular, may change by 40% with loading of the motor. This lack of accurate knowledge of machine parameters at any given instant makes the design and implementation of traditional controllers imprecise and cumbersome.

Fixed gain controllers suffer from overshoot, undershoot, steady-state error and even instability in their applications for high performance IPMSM drives. Conventional adaptive controllers are typically complex, impose high computational burdens, and usually require extensive knowledge of machine behaviour for their implementation.

The ANN- and SAN-based controllers presented in this work, coupled with approximated maximum torque per ampere and flux-weakening schemes of operation, overcome these problems by providing adaptive speed control which can accommodate parameter variations, system uncertainties and non-linearities as well as load disturbances. The SAN controller-based drive, in particular, achieves this while requiring a minimum of computing power and complexity of hardware.

Chapter 2 has detailed the analysis and modeling of the pulse-width modulated voltage source inverter-fed IPMSM drive including iron losses. Included in this chapter are practical method for incorporating the maximum torque per ampere and flux-weakening modes of operation in real-time in the IPMSM drive.

The development of an artificial neural network based speed controller for the IPMSM drive, including the development of a method of determination of a reference torque command for use in real-time ANN training, has been included in the Chapter 3.

Chapter 4 presents the further development and comparative simulation results of a four-neuron SAN controller-based IPMSM drive, utilizing the MTPA and FW schemes, with a conventional PID controller-based drive utilizing the $i_d^* = 0$ approximation. Extensive simulations have verified that the proposed drive readily outperforms the conventional PID-based controller with regards to speed response, load handling, compensation of parameter changes, ability to follow non-linear speed commands and tracking of command speed under non-linear loading.

In Chapter 5 a single artificial neuron-based speed controller, utilizing the MTPA and FW schemes, is developed and simulation results presented. Again, extensive simulation results have verified the drives proficiency and robustness with regards to speed response, load handling, compensation of parameter changes, ability to follow non-linear speed commands and tracking of command speed under non-linear loading.

Chapter 6 presents the real-time implementation of the SAN controller-based IPMSM drive utilizing MTPA and FW modes of operation. For comparison purposes, a conventional PID controller-based drive utilizing the $i_d^* = 0$ approximation is also implemented. Experimental investigation of the proposed drive under varying dynamic operating conditions has verified that the drive can achieve and follow command speed promptly without over- or undershoot and with virtually zero steady-state error for a variety of operating conditions. The drive responds quickly and accurately to step changes of command speed and has excellent load-handling capability. The robustness of the SAN-based speed controller has been verified.

7.2 Major Contributions of this Work

- The practical, real-time implementation of the maximum torque per ampere and flux-weakening modes of operation of the IPMSM have been developed.

- New ANN- and SAN-based speed controllers have been developed for the IPMSM for operation in both the MTPA and FW modes. The SAN controller, particularly, has been designed to achieve a high performance drive standard with minimum complexity and computation burden.

- The proposed ANN- and SAN-based drives have been simulated along with a conventional PID-based drive with $i_d^* = 0$ for comparison purposes. Comparative simulations have illustrated the ability of the proposed controllers to handle unpredictably changing motor dynamics as well as non-linear speed commands and load disturbances.

- The proposed SAN/MTPA/FW IPMSM drive has been implemented in real-time through a laboratory 1 hp IPMSM using a digital signal processor controller board. The efficacy and robustness of the drive has been verified by experimental investigation of the performance of the drive under various dynamic operating conditions such as sudden changes in command speed and loading.

- The theoretical foundation for the ANN and SAN-based controllers presented in this work is based on "generic" interior permanent magnet motor equations. Controller output torque is attained through multiplication by the motor's maximum torque and all inputs are normalized by division by command speed. The controllers themselves are not dependent on, or designed around, any specific motor. For these reasons, the controllers are expected to produce similar performances on any IPMSM governed by the same torque equation – with the appropriate motor parameters of magnetic flux linkage, stator resistance and q-d axis inductances substituted in.

7.3 Recommended Future Direction of Research

It has been established that the ANN and SAN-based controllers are insensitive to changes in motor parameters. However, in the development of the maximum torque per ampere and flux-weakening schemes of operation, the q- and d-axis reactances, L_q and L_d, have been assumed to be constant. This is not true under dynamic operating conditions. Although any negative impact due to this assumption appears to have little significance, for maximum performance in the context of high performance adjustable speed drive standards this should be investigated further. Ideally, a method of applying the MTPA and FW modes of operation without reliance on precise knowledge of such motor parameters, and the inherent inaccuracy that this entails, should be developed.

As the ANN- and SAN-based controllers reference a precisely calculated torque command, the determination of this reference torque has a profound effect on the performance characteristics of the drive. For this reason, a more detailed investigation should be done to determine the optimum method for calculating this reference torque. In particular, manipulation of the $K_{ref} J_m p\omega_r(n)$ term of Equation (3.26) by varying K_{ref} affects the sensitivity of the drive and its response characteristics. Further investigation of the determination of this parameter should be undertaken for high performance applications.

In addition, the reference torque command is dependent on motor phase currents as determined by sensors. As these currents are dictated by the characteristics of the current controller, improvement in the method of stator phase current application could result in more sinusoidal command current waveforms from the ANN- and SAN-based controllers and less phase current harmonics. This may also be achieved by use of high frequency filtering on the phase current inputs to the DSP board through the A/D converter.

7.4 Conclusions Drawn from this Research

- The maximum torque per ampere and flux-weakening approximation techniques can be used in the high performance IPMSM drive for operation at, above and below rated speed. This allows the motor to develop maximum mechanical torque with minimum values of stator currents at and below rated speed, while achieving speeds in excess of rated speed without exceeding machine voltage and current ratings.

- The proposed drive can be used effectively for accurate, precise and robust speed control of the IPMSM under different dynamic operating conditions such as, command speed change, non-linear speed command, sudden load application, non-linear loading, system uncertainties and parameter variations.

- The proposed drive provides an adaptive, intelligent control technique for the IPMSM with minimal complexity, computational burden and hardware requirements.

- The efficacy of the new control system has been established theoretically, in simulation and experimentally.

References

[1] P. C. Sen, "Electric Motor Drives and Control – Past, Present and Future", *IEEE Trans. on Industrial Electronics*, vol. 37, no. 6, Dec. 1990, pp. 562-575.

[2] M. A. Rahman, "Modern Electric Motors in Electronic World", *IEEE/IECON'93 Conference Record*, (Hawaii), pp. 644-648.

[3] G. R. Slemon, *Electric Machines and Drives*, Addison-Wesley Publishing Company, 1992.

[4] B. K. Bose, *Power Electronics and AC Drives*, Englewood Cliffs, NJ: Prentice Hall, 1986.

[5] P. C. Sen, *Principles of Electric Machines and Power Electronics*, New York: Wiley, 1988.

[6] M. H. Rashid, *Power Electronics-Circuits, Devices and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[7] M. A. Rahman, “Permanent Magnet Synchronous Motors - A Review of the State of Design Art”, *Proceedings of International Conference On Electric Machines, Athens*, 1980, pp. 312-319.

[8] M. A. Hoque, “Speed Control of High Performance Permanent Magnet Motors”, Ph. D. Thesis, Memorial University of Newfoundland, Canada, Aug. 1996.

[9] T. S. Radwan, *Doctor of Philosophy Thesis*, Menoufiya University, Egypt, Aug. 1996.

[10] F. Blaschke, “The Principle of Field Orientation as Applied to the New Transvector Closed-Loop Control System for Rotating Field Machines”, *Siemens Review*, Vol. 34, May 1972, pp. 217-220.

[11] K. Hasse, “On the Dynamic Behaviour of Induction Machines Driven by Variable Frequency and Variable Voltage Sources”, *ETZ A, Bd. 89,* H.4, pp. 77-81, 1968.

[12] M. A. Rahman, T. S. Radwan, A. M. Osheiba and A. E. Lashine, “Analysis of Current Controllers for Voltage Source Inverters”, *IEEE Trans. Ind. Electronics*, vol. 44, Aug. 1997, pp. 477-485.

[13] H. Le-Huy, K. Silmani and P. Viarouge, “Analysis and Implementation of a Real-Time Predictive Current Controller for Permanent Magnet Synchronous Servo Drives”, *IEEE Trans. on IEEE Trans. on Industrial Electronics*, vol. 41, no. 1, Feb. 1994, pp. 110-117.

[14] H. Le-Huy and L. A. Dessaint, "An Adaptive Current Control Scheme for PWM Synchronous Motor Drive: Analysis and Simulation", *IEEE Trans. on Power Electronics*, vol. 4, no. 4, Oct. 1989, pp. 486-495.

[15] A. Tripathi and P. C. Sen, "Comparative Analysis of Fixed and Sinusoidal Band Hysteresis Current Controllers for Voltage Source Inverters", *IEEE Trans. on Ind. Electronics*, vol. 39, Feb. 1992, pp. 63-73.

[16] D. M. Brod and D. W. Novotny, "Current Control of VSI-PWM Inverters", *IEEE Transactions on Industry Applications*, Vol. IA-21, No. 4, May/June 1985, pp. 562-570.

[17] J. Holtz and S. Stadtfeld, "A Predictive Controller for the Stator Current Vector of the ac Machines fed from Switching Voltage Source", *Proc. Int. Power Engineering Conf.*, Tokyo, Japan, Apr. 1983, pp. 1665-1675.

[18] A. B. Plunkett, "A Current Controlled PWM Transistor Inverter Drive", *Conf. Rec. IEEE-IAS Annu. Meeting*, Oct. 1979, pp. 951-957.

[19] M. N. Uddin, T. S. Radwan, G. H. George and M. A. Rahman, "Performance of Current Controllers for VSI-Fed IPMSM Drive", *IEEE Transactions on Industry Applications*, Vol. 36, No. 6, Nov./Dec. 2000, pp. 1531-1538.

[20] B. K. Bose and P. M. Szczesny, "A Microcomputer-Based Control and Simulation of an Advanced IPM Synchronous Machine Drive System for Electric Vehicle Propulsion", *IEEE Trans. on Industrial Electronics*, vol. 3, no. 4, Nov. 1988, pp. 547-559.

[21] A. V. Gummaste and G. R. Slemon, "Steady State Analysis of a Permanent Magnet Synchronous Motor Drive with Voltage Source Inverter", *IEEE Trans. on Industrial Applications*, vol. IA-17, no. 2, March/April 1981, pp.143-151.

[22] G. R. Slemon and A. V. Gummaste, "Steady State Analysis of a Permanent Magnet Synchronous Motor Drive with Current Source Inverter", *IEEE Trans. on Industrial Applications*, vol. IA-19, no. 2, March/April 1983, pp. 190-197.

[23] T. H. Liu, C. M. Young and C. H. Liu, "Microprocessor Based Controller Design and Simulation of Permanent Magnet Synchronous Motor Drive", *IEEE Trans. on Industrial Electronics*, vol. 35, no. 4, Nov. 1988, pp. 516-523.

[24] P. Pillay and R. Krishnan, "Modeling of permanent magnet motor drives", *IEEE Trans. on Industrial Electronics*, vol. 35, no. 4, Nov. 1988, pp. 537-541.

[25] P. Pillay and R. Krishnan, "Modeling, Simulation and Analysis of Permanent Magnet Motor Drives, Part I: The Permanent-Magnet Synchronous Motor Drive", *IEEE Trans. on Industry Applications*, vol. 25, no. 2, March/April 1989, pp. 265-273.

[26] P. Pillay and R. Krishnan, "Modeling, Simulation and Analysis of a High Performance Vector Controlled Permanent Magnet Synchronous Motor Drive", *IEEE/IAS Annual Meeting Conference Record*, 1987, pp 253-261.

[27] P. Pillay and R. Krishnan, "Control Characteristics and Speed Controller Design for High Performance Permanent Magnet Synchronous Motor Drive", *IEEE Trans. on Power Electronics*, vol. 5, no. 2, April 1990, pp.151-159.

[28] M. A. Rahman, M. Vilathgamuwa, M. N. Uddin & K .J. Tseng, " Non-linear control of interior permanent magnet synchronous motor drive", *IEEE Trans. on Industry Applications*, Vol.39, Mar/April 2003, pp.408-416.

[29] P. Pillay, C. R. Allen and R. Budhabhathi, "DSP-Based Vector Control and Current Controllers for Permanent Magnet Synchronous Motor Drive", *IEEE/IAS Annual Meeting Conference Record*, Seattle, Washington, 1990, pp. 539-544.

[30] B. K. Bose, "A High Performance Inverter Fed Drive System of an Interior Permanent Magnet Synchronous Machine", *IEEE Trans. on Industry Applications*, vol. IA-24, no. 6, 1988, pp. 987-997.

[31] H. Le-Huy and L. A. Dessaint, "An Adaptive Current Control Scheme for PWM Synchronous Motor Drive: Analysis and Simulation", *IEEE Trans. on Power Electronics*, vol. 4, no. 4, Oct. 1989, pp. 486-495.

[32] H. Le-Huy, K. Silmani and P. Viarouge, "Analysis and Implementation of a Real-Time Predictive Current Controller for Permanent Magnet Synchronous Servo Drives", *IEEE Trans. on IEEE Trans. on Industrial Electronics*, vol. 41, no. 1, Feb. 1994, pp. 110-117.

[33] B. K. Bose and P. M. Szczesny, "A Microcomputer-Based Control and Simulation of an Advanced IPM Synchronous Machine Drive System for Electric Vehicle Propulsion", *IEEE Trans. on Industrial Electronics*, vol. 3, no. 4, Nov. 1988, pp. 547-559.

[34] T. M. Jahns, G. B. Kilman and T. W. Neumann, "Interior Permanent Magnet Synchronous Motors for Adjustable Speed Drives", *IEEE Trans. on Industry Applications*, vol. IA-22, no. 4, July/August 1986, pp. 738-747.

[35] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior Permanent Magnet Synchronous Motor Drive", *IEEE Trans. on Industry Applications*, vol. IA-23, no. 4, 1987, pp. 681-689.

[36] S. Morimoto, Y. Takeda and T. Hirasha, "Current Phase Control Methods for Permanent Magnet Synchronous Motors", *IEEE Trans. on Power Electronics*, vol. 5, no. 2, April 1990, pp. 133-139.

[37] S. Morimoto, Y. Takeda, T. Hirasa and K. Taniguchi, "Expansion of Operating Limits for Permanent Magnet Motor by Current Vector Control Considering the Inverter Capacity", *IEEE Trans. on Industry Applications*, vol. 26, no. 5, Sept./Oct. 1990, pp. 866-871.

[38] S. Morimoto, Y. Takeda, K. Hatanaka, Y. Tong, and T. Hirasa, "Design and Control System of Permanent Magnet Synchronous Motor for High Torque and High Efficiency Operation", *IEEE/IAS Annual Meeting Conference Record*, 1991, pp. 463-468.

[39] S. Morimoto, K. Hatanaka, Y. Tong, Y. Takeda and T. Hirasa, "Servo Drive System and Control Characteristics of salient Pole Permanent Magnet Synchronous Motor", *IEEE Trans. on Industry Applications*, vol. 29, no. 2, March/April 1993, pp. 338-343.

[40] S. Morimoto, Y. Takeda, and T. Hirasa, "Flux Weakening Control Method for Surface Permanent Magnet Synchronous Motors", *IPEC Conference. Record, Tokyo, Japan,* 1990, pp. 942-950.

[41] S. Morimoto, M. Sanada and Y. Takeda, "Effects and Compensation of Magnetic Saturation in Flux-Weakening Controlled Permanent Magnet Synchronous Motor Drives", *IEEE Trans. on Industry Applications*, vol. 30, no. 6, Nov./Dec. 1994, pp. 1632-1637.

[42] M.F. Rahman, L. Zhong and K.W. Lim, "A direct torque-controlled interior permanent magnet synchronous motor drive incorporating field weakening", *IEEE Transations on Industry Applications.,* Vol. 34, Nov/Dec, 1998, pp. 1246-1253.

[43] S. Vaez, V. I. John and M. A. Rahman, "An On-Line Loss Minimization Controller for Interior Permanent Magnet Motor Drives", *IEEE Trans. on Energy Conversion,* Paper No. PE-I345-EC-0-07-1998.

[44] T. S. Radwan, M. A. Rahman, A. M. Osheiba and A. E. Lashine, "Performance of a Hybrid Current-Controlled VSI Permanent Magnet Synchronous Motor Drive", *IEEE/PESC Conference Record*, 1996, pp.951-957.

[45] M.A. Hoque, Casey Butt and M.A.Rahman, “A Novel Approach for MTPA Speed Control of IPMSM Drive”, *Proc. 2nd IEEE International Conference of Electrical & Computer Engineering*, Dhaka, Dec. 26-28 2002, pp. 336-339.

[46] R. Krishnan, “Control and Operation of PM Synchronous Motor Drives in the Field Weakening Region”, *IEEE/IECON’93 Conference Record, Hawaii*, 1993, pp. 745-750.

[47] S. R. Macminn and T. M. Jahns, “Control Techniques for Improved High–Speed Performance of Interior PM Synchronous Motor Drives”, *IEEE Trans. on Industry Applications*, vol. 27, no. 5, Sep./Oct. 1991, pp. 997-1004.

[48] R. Dhanuadi and N. Mohan, “Analysis of Current-Regulated Voltage Source Inverters for Permanent Magnet Synchronous Motor Drives in Normal and Extended Speed Range”, *IEEE Trans. on Energy Conversion*, vol. 5, no. 1, March 1990, pp. 137-147.

[49] A. Kumamoto and Y. Hirame, “A Flux-Weakening Method for a Buried Permanent Magnet Motor with Consideration of Parameter Detunning on System Performance”, *IPEC Conference Record, Tokyo, Japan*, 1990, pp. 950-955.

[50] I. Choy, T. Yoon, K. Kim, and M. Park, “Microprocessor-Based Permanent Magnet Synchronous Motor Drives using MRAC”, *IPEC Conference Record, Tokyo, Japan*, 1990, pp. 481-488.

[51] E. Cerruto, A. Raciti and A. Testa, "A Robust Adaptive Controller for PM Motor Drives in Robotic Applications", IEEE Trans. on Power Electronics, vol. 10, no. 1, Jan. 1995, pp. 62-70.

[52] Y. Sozer and D. A. Torrey, "Adaptive Flux Weakening Control of Permanent Magnet Synchronous Motors", *IEEE/IAS Annual Meeting Conference Record*, 1998, pp. 475-482.

[53] C. Namudri and P. C. Sen, "A Servo-Control System using a Self-Controlled Synchronous Motor (SCSM) with Sliding Mode Controller", *IEEE/IAS Annual Meeting Conference Record*, 1987, pp. 283-295.

[54] A. Consoli and Antonio, "A DSP Based Sliding Mode Field Oriented Control of an Interior Permanent Magnet Synchronous Motor Drive", *IPEC Conference Record*, Tokyo, Japan, 1990, pp. 263-303.

[55] M. Ghirby and H. Le-Huy, "Optimal Control and Variable Structure Combination using Permanent Magnet Synchronous Motor", *IEEE/IAS Annual Meeting Conference Record*, 1994, pp. 408-415.

[56] R. B. Sepe and J. H. Lang, "Real Time Adaptive Control of the Permanent Magnet Synchronous Motor", *IEEE/IAS Annual Meeting Conference Record*, 1990, pp. 545-552.

[57] Z. Q. Zhu, Y. S. Chen and David Howe, "Online Optimal Flux-Weakening Control of Permanent-Magnet Brushless AC Drives", *IEEE Trans. on Industry Applications*, vol. 36, no. 6, Nov./Dec. 2000, pp. 1661-1668.

[58] K. Inoue, Y. Takakado and M. Nakaoka, “Auto Tuning Technology for Fuzzy Algorithms Based DC Brushless Servo Systems”, *IEEE/PESC Conference Record*, 1993, pp. 446-450.

[59] K. Erenay, I. Ciprut, L. Tezduyar, Y. Istefanopulos, “Application of Fuzzy Algorithms to the Speed Control of Washing Machines with Brushless DC Motors”, *Proceedings of International Conference on Electric Machines, Istanbul, Turkey*, 1998, pp. 1231-1236.

[60] Z. Koviac, S. Bogdan and P. Crnosija, “Fuzzy Rule-Based Model Reference Adaptive Control of Permanent Magnet Synchronous Motor”, *IEEE/IECON Conference Record*, 1993, pp. 207-212.

[61] M. N. Uddin and M. A. Rahman, “Fuzzy Logic Based Speed Control of an IPM Synchronous Motor Drive”, *Journal of Advanced Computational Intelligence,* vol. 4, no. 2, 2000, paper #JACI-CCECE-S-05.

[62] M. A. El-Sharkawi, A. A. El-Samahy and M. L. El-Sayed, “High Performance Drive of dc Brushless Motors Using Neural Network*”, IEEE Trans. on Energy Conversion*, vol. 9, no. 2, June 1994, pp. 317-322.

[63] C. Shiguo, D. G. Holmes and W. A. Brown, “Digital Control of a Servo System using Neural Network”, *IEEE/IAS Annual Meeting Conference Record*, 1995, pp. 129-133.

[64] M.A. Hoque, M.R. Zaman and M.A. Rahman, “Artificial Neural Networks Based Controller for Permanent Magnet DC Motor Drives”, *IEEE/IAS Annual Meeting,* Orlando, FL, Oct. 1995, Vol. 1, pp. 98-103.

[65] M.A. Rahman and M.A. Hoque, "On-Line Adaptive Artificial Neural Network Based Vector Control of Permanent Magnet Synchronous Motors", *IEEE Trans. on Energy Conversion*, vol. 13, no. 4, 1998, pp. 311-318.

[66] M.N. Uddin, C.B. Butt, M.A. Abido and M. A. Rahman, "Laboratory Implementation of an Artificial Neural Network for Online Tuning of a Genetic Algorithm Based PI Controller for IPMSM Drive," ***7th International Conference on Modeling and Simulation of Electric Machines, Converters and Systems ELECTRIMACS-2002,*** Montreal, Canada, August 18-21, 2002, CD-ROM.

[67] Yang Yi, D.M. Vilathgamuwa and M.A. Rahman, "A New Artificial Neural Network Controller for an Interior Permanent Magnet Motor Drive", *Proceeding of IEEE/IAS Annual Meeting*, Chicago, IL, USA, Oct 2001, Vol. 2, pp. 945-952.

[68] M. A. Rahman and Ping Zhou, "Field Circuit Analysis of Brushless Permanent Magnet Synchronous Motors", *IEEE Trans. on Ind. Elect.*, vol. 43, No.2, April 1996, pp. 256-267.

[69] S. Morimoto, T. Ueno, M. Sanada, A. Yamagiwa, Y. Takeda and T. Hirasa, "Effects and Compensation of Magnetic Saturation in Permanent Magnet Synchronous Motor Drives", Proceedings of *IEEE/IAS Annual Meeting*, Toronto, ON, Canada, Oct. 1993, Vol. 1, pp. 59-64.

[70] C. Mademlis and V.G. Agelidis, "On Considering Magnetic Saturation with Maximum Torque to Current Control in Interior Permanent Magnet Syn-

chronous Motor Drives", *IEEE Trans. on Energy Conversion,* Vol. 16, No. 3, 2001, pp. 246-252.

[71] B. J. Chalmers, "Influence of Saturation in Brushless Permanent-Magnet Motor Drives", *IEE Proceedings,* pt. B, Vol. 139, No. 1, 1992, pp. 51-52.

[72] P. C. Krause, *Analysis of Electric Machinery*, McGraw-Hill Inc., 1986.

[73] Chunting Yi, G.R. Slemon and R. Bonert, "Modeling of Iron Losses of Permanent-Magnet Synchronous Motors", *IEEE Trans. on Industry Applications,* Vol. 39, No. 3, 2003, pp. 734-742.

[74] N. Urasaki, T. Senjyu and K. Uezato, "A Novel Calculation Method for Iron Loss Resistance Suitable in Modeling Permanent-Magnet Synchronous Motors", *IEEE Trans. on Energy Conversion,* Vol. 18, No. 1, 2003, pp. 41-47.

[75] S. Morimoto, Y. Tong, Y. Takeda and T. Hirasa, "Loss Minimization Control of Permanent Magnet Synchronous Motor Drives", *IEEE Trans. on Ind. Electronics*, vol. 41, no. 5, 1994, pp. 511-517.

[76] S. Morimoto, M. Sanda and Y. Taketa, "Wide-Speed Operation of Interior Permanent Magnet Synchronous Motors with High-Performance Current Regulator", *IEEE Trans. on Ind. Applicat.,*Vol. 30, July/Aug, 1994, pp.920-926

[77] R.F. Schiferl and T.A. Lipo, "Power Capability of Salient Pole Permanent Magnet Synchronous Motors in Variable Speed Drive Applications", *IEEE Trans. on Industry Applications*, vol. 26, no. 1, 1990, pp. 115-123.

[78] Haykin, H., "Neural Networks: A Comprehensive Foundation", IEEE Press, Macmillan College Publishing Company Inc., 1994.

[79] B. Widrow, R.G. Winter and R.A. Baxter, "Layered Neural Nets for Pattern Recognition", *IEEE Trans. on Acoustics, Speech, Signal Processing,* Vol. 36, No. 7, 1988, pp. 1109-1118.

[80] D.J. Burr, "Experiments on Neural Net Recognition of Spoken and Written Text", *IEEE Trans. on Acoustics, Speech, Signal Processing,* Vol. 36, No. 7, 1988, pp. 1162-1168.

[81] Narendra, K.S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. on Neural Networks,*Vol. 1, No. 1, 1990, pp. 4-27.

[82] Fukuda, T. Shibata, T., "Theory and Applications of Neural Network for Industrial Control Systems," *IEEE Trans. on Ind. Electronic.,*Vol. 39, No. 6, 1992, pp. 473-489.

[83] Chen, T.C. and Liaw, C.Y., "Design of a Neural Fuzzy Controller for Induction Motor Speed Control", *Proceedings of IECON*, 1994, pp. 611-616.

[84] O. De Jesus and M. Hagan, "Backpropagation Algorithms for a Broad Class of Dynamic Networks", *IEEE Trans. on Neural Networks,* Vol. 18, No. 1, 2007, pp. 14-27.

[85] S. Weerasooriya and M.A. El-Sharkawi, "Identification and Control of a DC Motor Using Back-Propagation Neural Networks", *IEEE Trans. on Energy Conversion,* Vol. 6, No. 4, 1991, pp. 663-669.

[86] A. Rubaai and R. Koteru, "Neural Net-based Robust Controller Design for Brushless DC Motor Drives", *IEEE Trans. on Systems, Man and Cybernetics,* Vol. 29, No. 3, 1999, pp. 460-474.

[87] K.S. Narendra and K. Parthasarathy, "Gradient-Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Trans. on Neural Networks,* Vol. 2, No. 2, 1991, pp. 252-262.

[88] B. Burton and R.G. Harley, "Reducing the Computational Demands of Continually Online-Trained Artificial Neural Networks for System Identification and Control of Fast Processes", *IEEE Trans. on Industry Applications,* Vol. 34, No. 3, 1998, pp. 589-596.

[89] Yang Yi, D.M. Vilathgamuwa and M.A. Rahman, "A New Artificial Neural Network Controller for an Interior Permanent Magnet Motor Drive", *Proceeding of IEEE/IAS Annual Meeting*, Chicago, IL, USA, Oct 2001, Vol. 2, pp. 945-952.

[90] *DS1102 Program Loader User's Guide*, dSPACE, Paderborn, Germany, Ver. 2, 1996.

APPENDIX A

IPMSM Parameters

Number of phase = 3

Number of poles = 4

Rated current = 3 A

Rated frequency = 60 Hz

Rated power = 1 hp

Rated input line to line voltage = 208 V

q-axis inductance L_q = 0.07957 H

d-axis inductance L_d = 0.04244 H

Stator resistance per phase r_s = 1.93 Ω

Inertia constant J_m = 0.003 Kg.m^2

Rotor damping constant B_m = 0.0008 (N-m)/rad./sec.

Permanent magnet flux linkage ψ_m = 0.314 volts/rad./sec.

Magnet type = Samarium Cobalt

* Test motor parameters were calculated in reference [68] from phase parameters and voltage and current information during actual motor operation. The given parameters were calculated at rated conditions. These parameters have been used extensively and found to produce accurate controller responses.

APPENDIX B

Simulink Model of IPMSM Drive

In modelling the system in SIMULINK it is necessary to work from the principle equations governing the IPMSM drive system rather than base the individual components of the model strictly on the schematic of the vector control scheme. This is because working in the d-q reference frame necessitates the use of transformation operations that are not implemented, physically, in real-world systems. Figure B.1 presents the SIMULINK schematic of the complete current-controlled VSI-fed IPMSM drive system that was developed in this work.

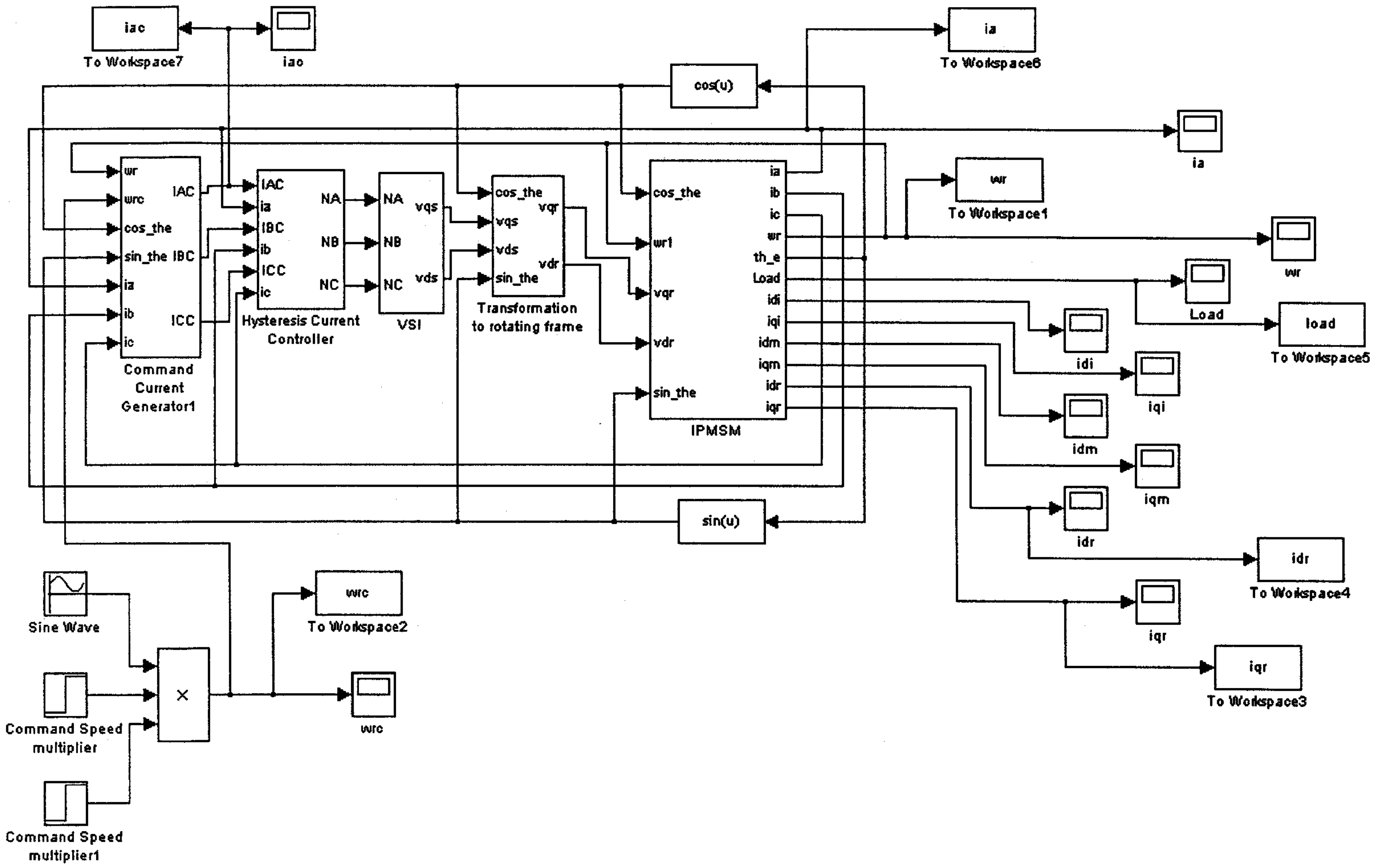


Figure B.1. Simulink model of IPMSM Drive.

The following subsections present the purpose and design of each of the component blocks of the model in detail. Only the SAN/MTPA scheme is illustrated.

B.1 Command Current Generator

The Command Current Generator block generates the command currents necessary to maintain the desired rotor speed of the motor. To do this, rotor speed and rotor angular position are necessary inputs. The schematic of the Command Current Generator block is shown in Figures B.2a and B.2b.

The command torque necessary to achieve the desired rotor speed is calculated as per detailed in Chapters 3, 4 and 5. This command torque is then used to obtain i_{qm}^{r*} and i_{dm}^{r*} by means of either the MTPA or FW scheme, depending on command speed (the MTPA scheme is shown). Then, i_q^{r*}, i_d^{r*} and i_a^*, i_b^*, and i_c^* are calculated. This block is shown in Figures B.2a and B.2b.

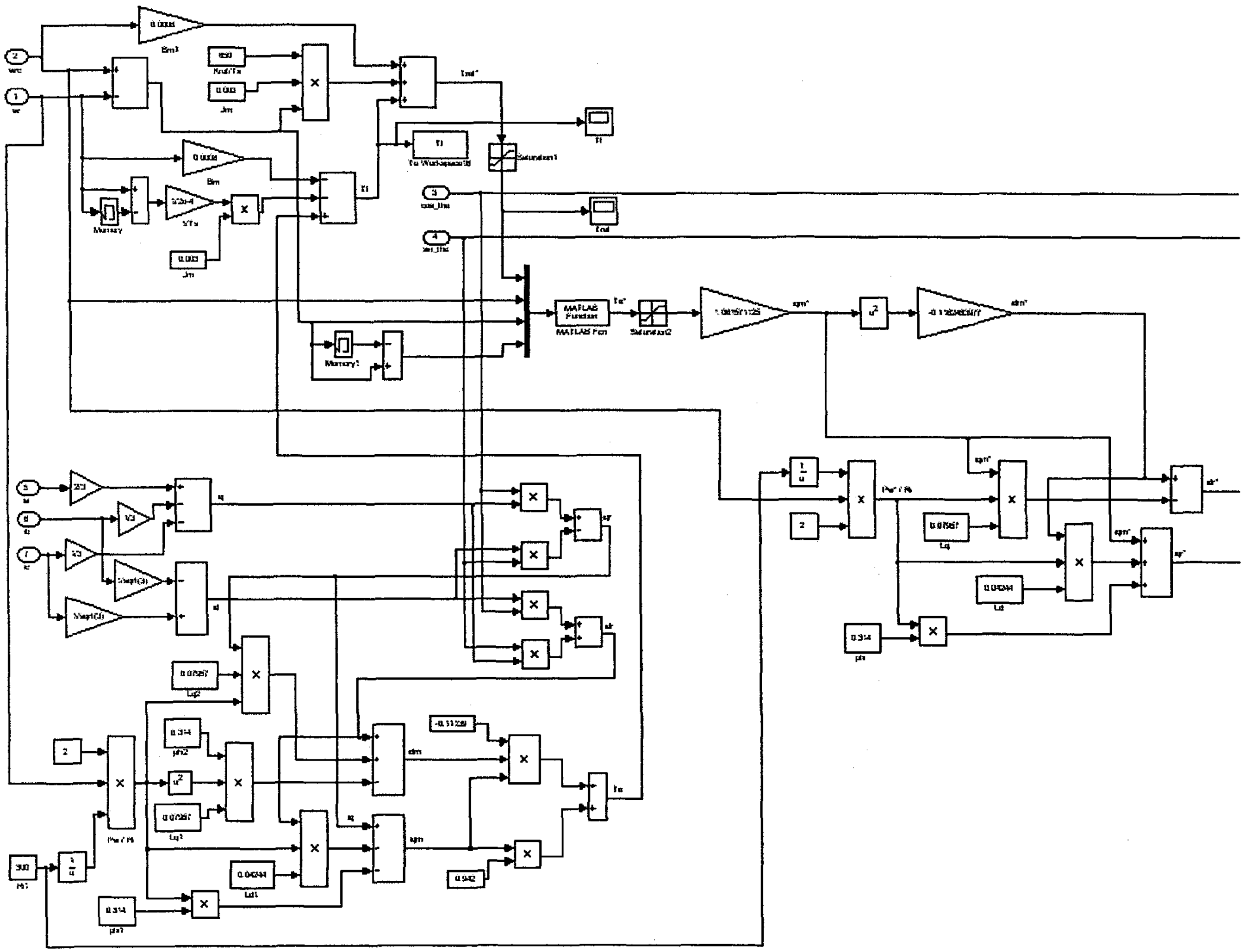


Figure B.2a. Command current generator block.

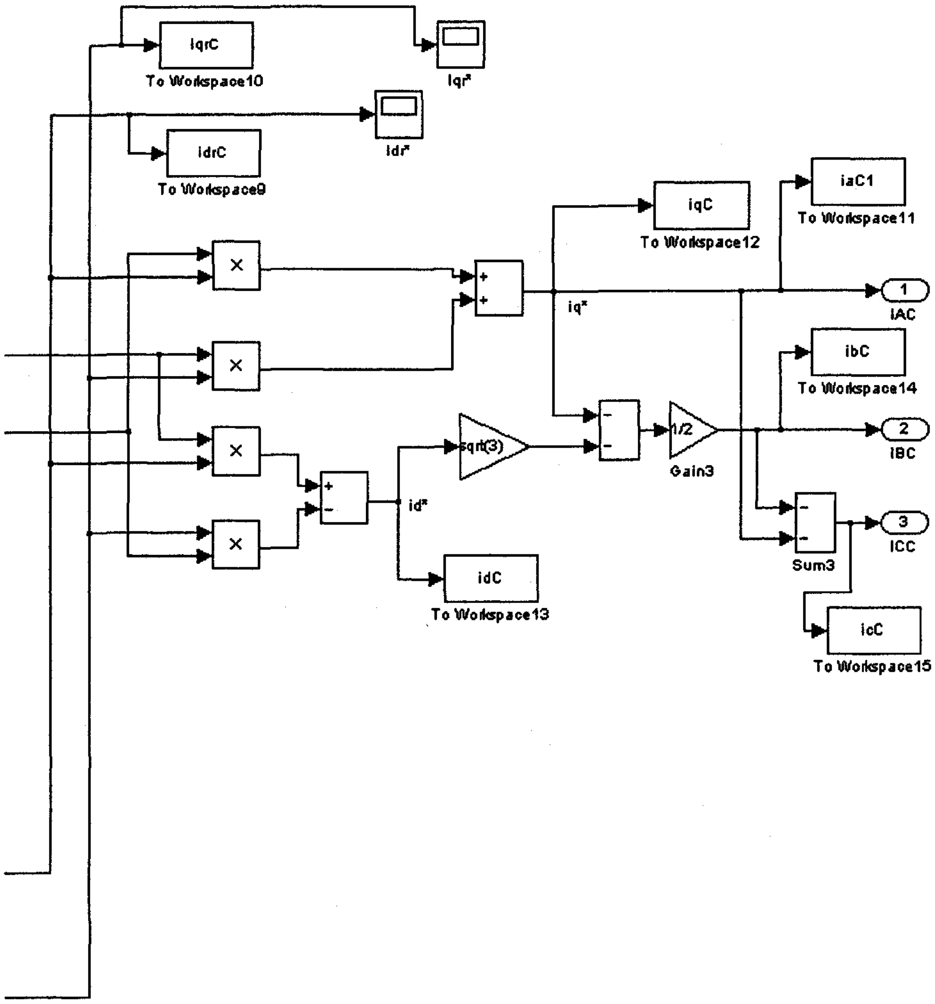


Figure B.2b. Command current generator block (continued).

B.2 Current Controller

The next block in the model represents the controller for the voltage source inverter. The current controller takes as input the three phase command currents from the Command Current Generator and outputs the logic variables NA, NB and NC used to control the VSI – a fixed-band hysteresis controller, as shown in Figure B.3, is employed.

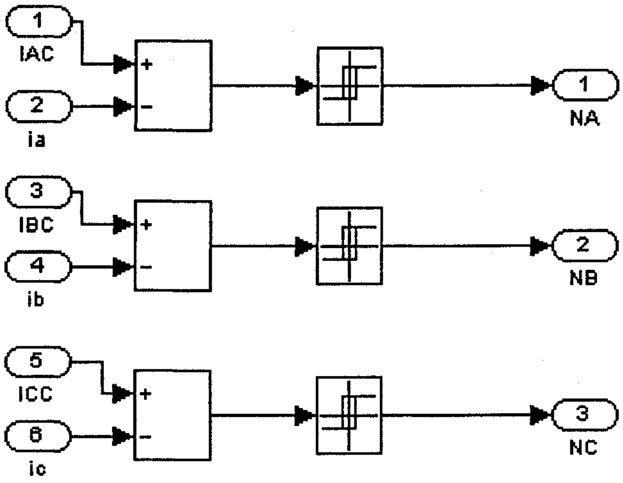


Figure B.3. Hysteresis current controller.

B.3 Voltage Source Inverter

The next block in the model represents the voltage source inverter and is designed according to the following equation,

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} NA \\ NB \\ NC \end{bmatrix} V_B \tag{B.1}$$

and Equation (2.11). Outputs are the v_q^* and v_d^* command voltages. This block is shown in Figure B.4.

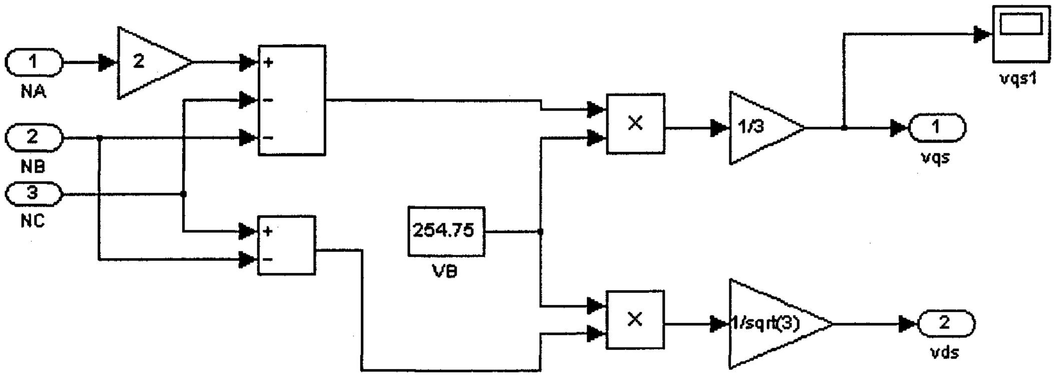


Figure B.4. Voltage source inverter block.

B.4 Transformation to Rotating Frame

The IPMSM model equations require i_q^{r*} and i_d^{r*} to give the outputs ω_r and θ_r. Therefore, v_q^* and v_d^*, as obtained from the VSI, must be transformed to the rotating d-q frame in order to give v_q^{r*} and v_d^{r*}. This is done, as shown in Figure B.5, using Equation (2.12).

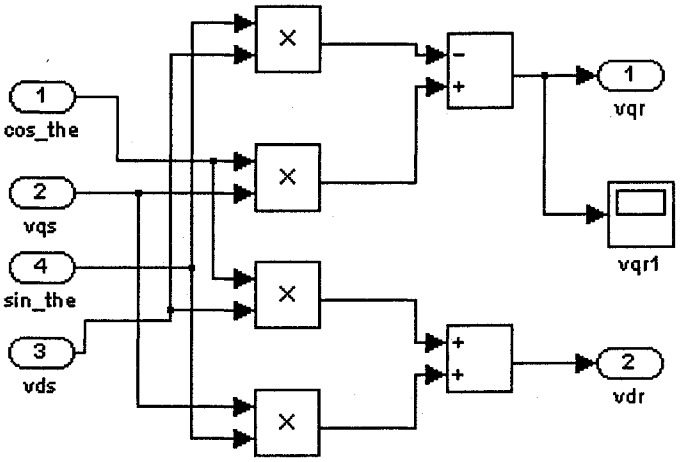


Figure B.5. Transformation to rotating reference frame block.

B.5 IPMSM

This block (Figure B.6) represents the actual motor. Inputs are v_q^{r*}, v_d^{r*} and ω_r. In addition, i_q^{r*} and i_d^{r*} must be converted to i_a, i_b and i_c, and these conversions require the input of $\cos\theta_r$ and $\sin\theta_r$.

Because several functions are performed in this block, it is further composed of three sub-blocks which each perform specific tasks.

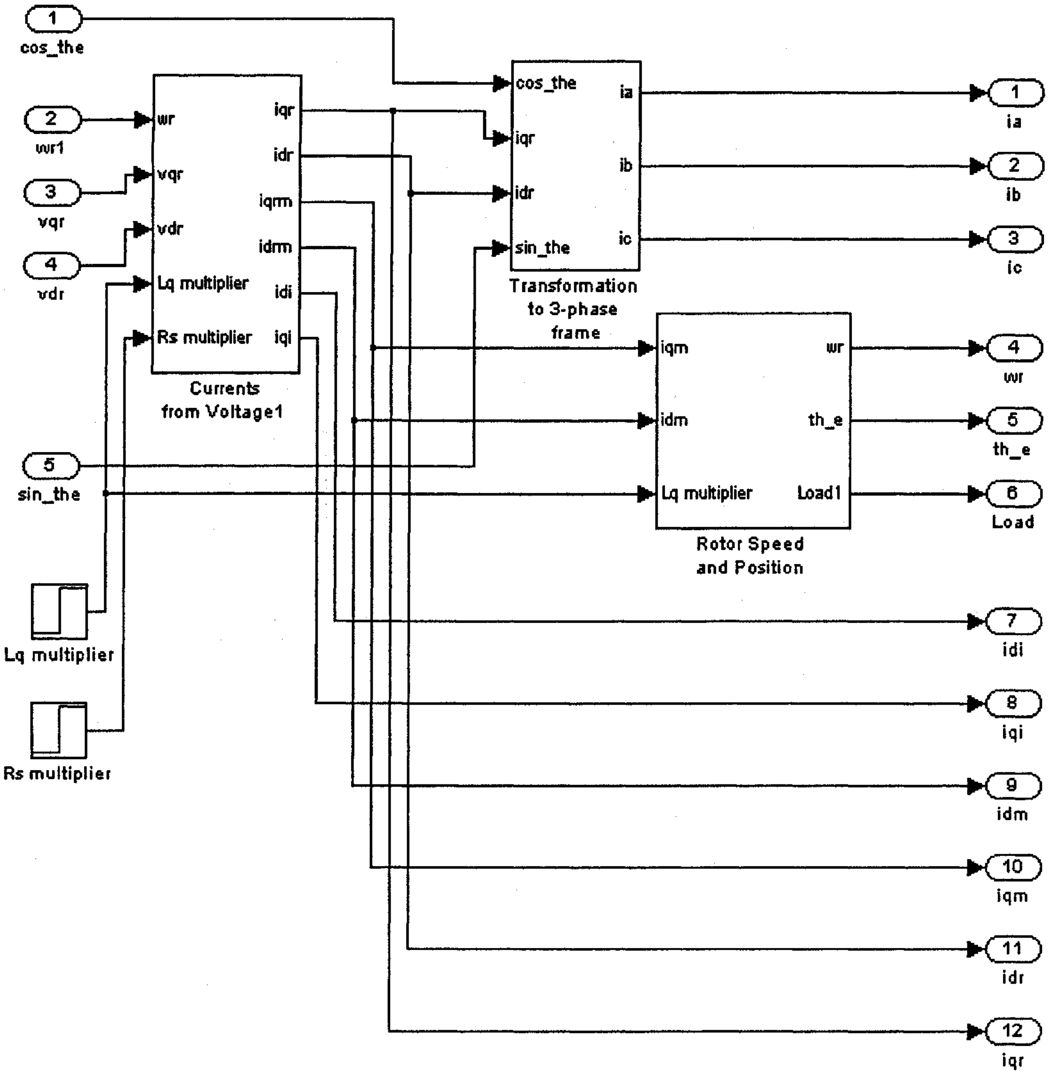


Figure B.6. IPMSM block.

The first sub-block (Figure B.7) calculates the actual motor currents, i_{qm}^r, i_{dm}^r, i_{qi}^r, i_{di}^r (iron loss currents) and i_q^r, i_d^r from v_q^{r*}, v_d^{r*} and ω_r. This is done via manipulation of the IPMSM equations, Equations (2.34) and (2.35).

The second sub-block (Figure B.8) converts i_q^r and i_d^r to i_a, i_b, and i_c via Equations (2.13) and (2.10).

The third sub-block calculates the motor's rotor speed and position via Equations (2.32) and (2.36). This sub-block is shown in Figure B.9.

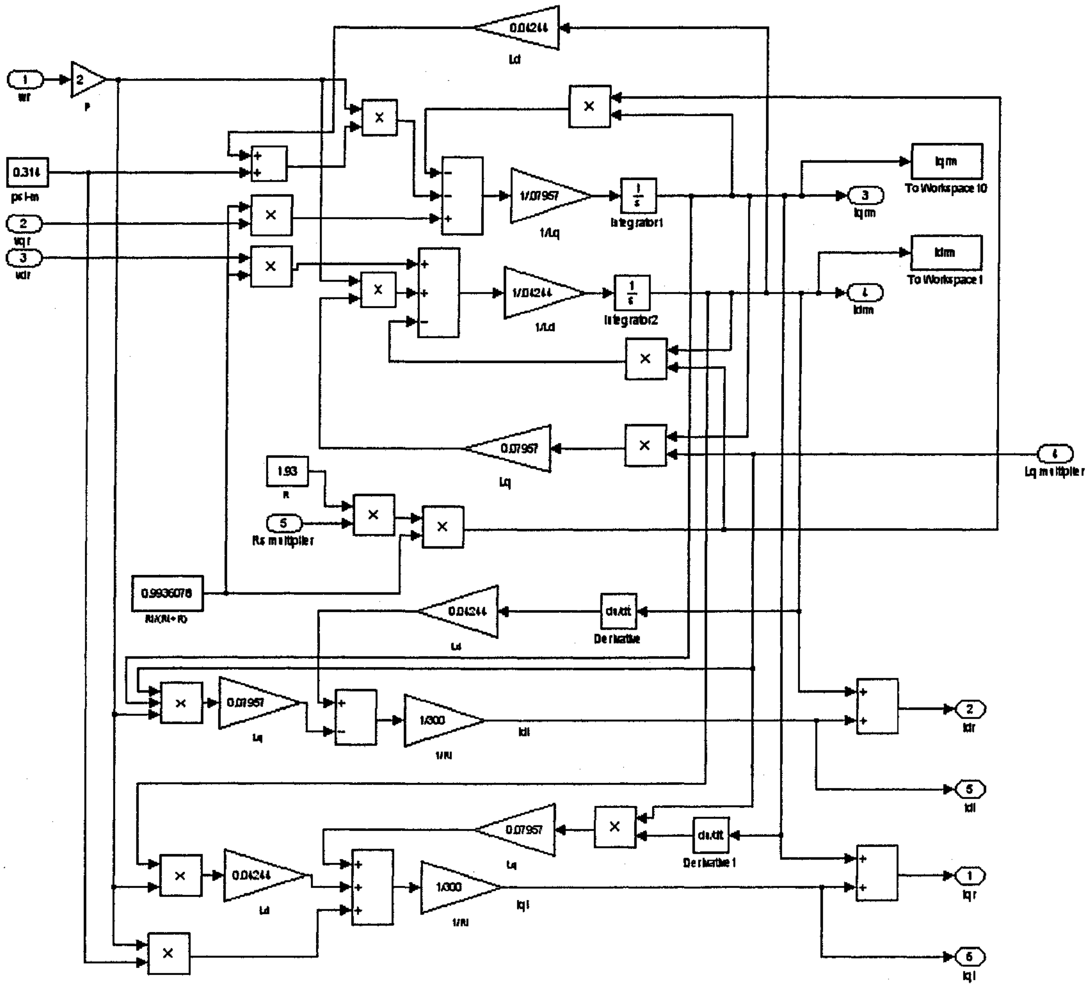


Figure B.7. Calculation of q- and d-axis currents from q- and d-axis command voltages.

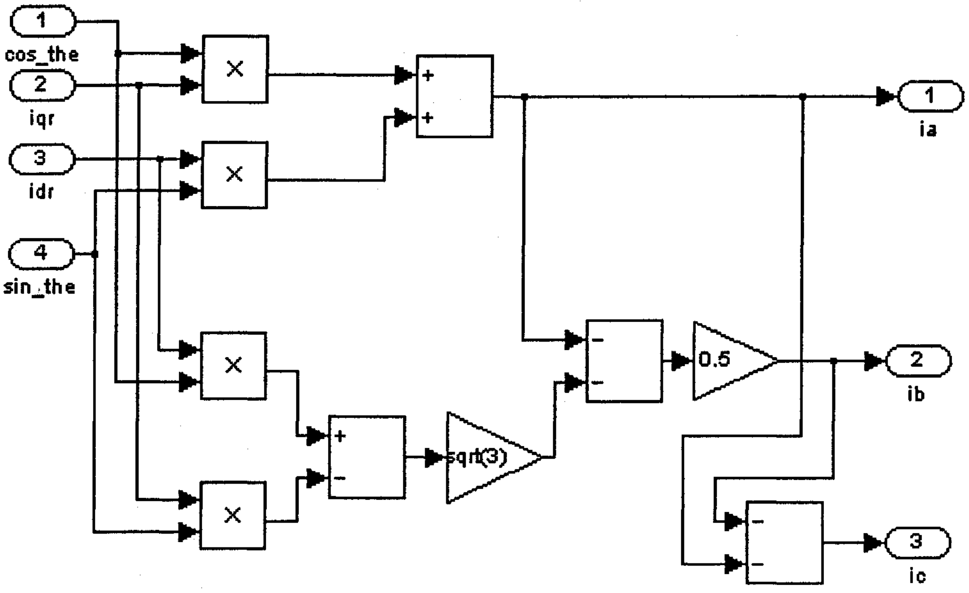


Figure B.8. Calculation of stator phase currents from q- and d-axis currents.

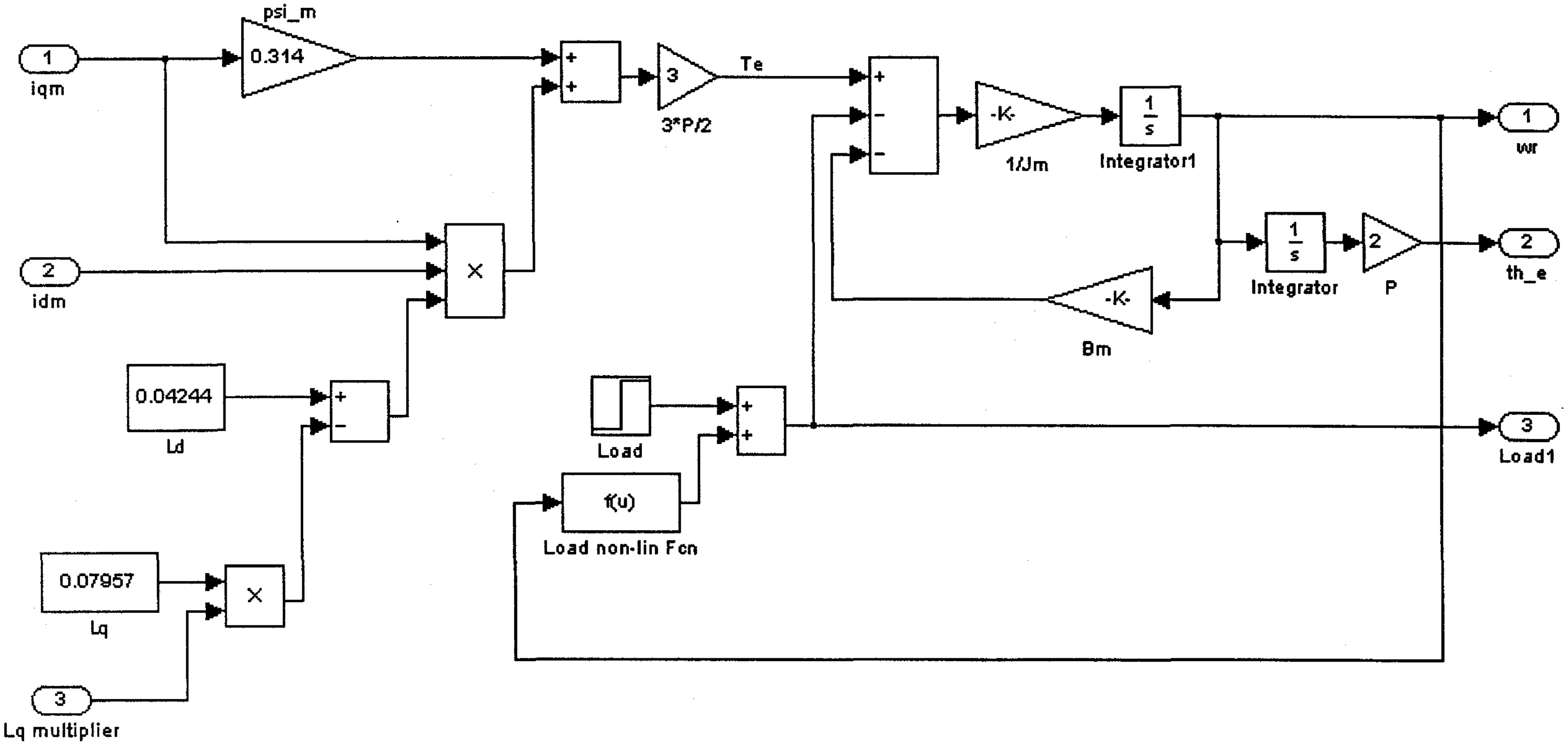


Figure B.9. Motor speed and position sub-block.

APPENDIX C

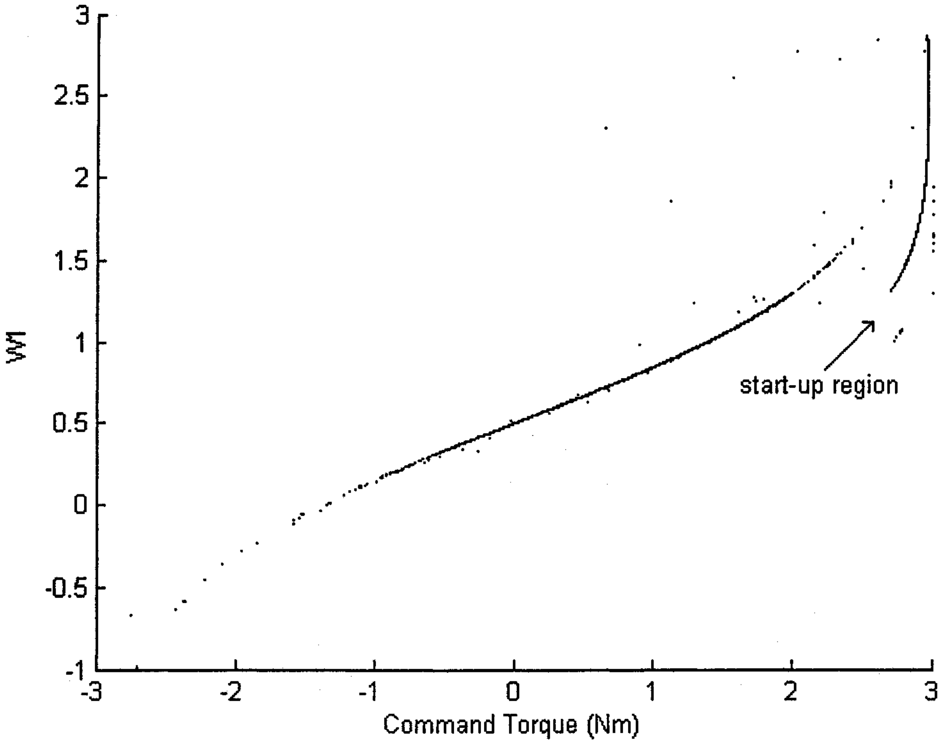


Figure C1. W1 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

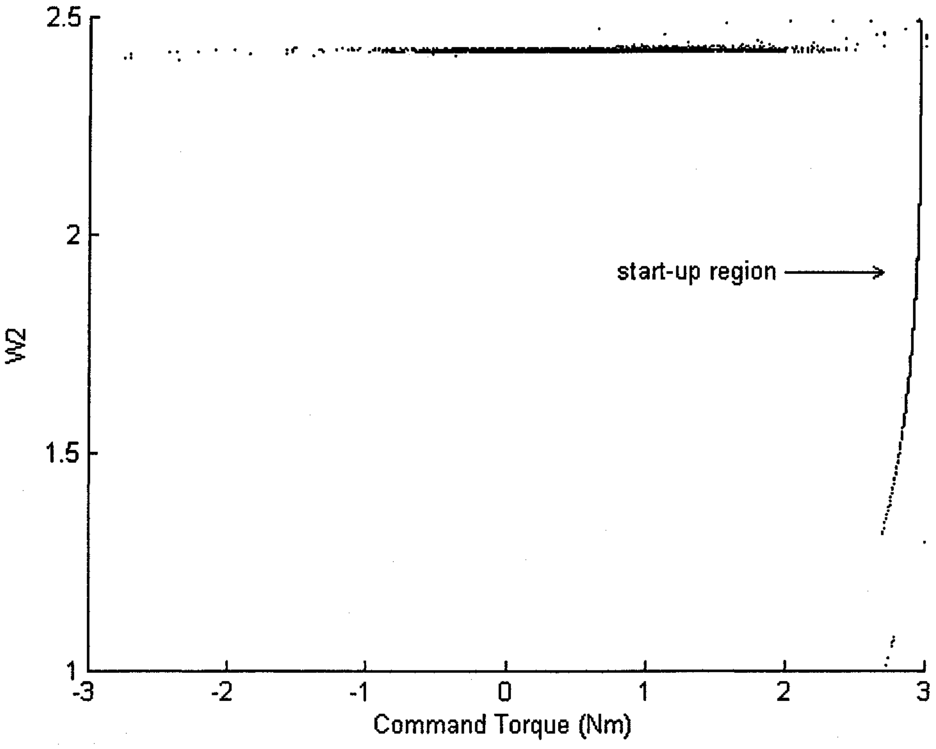


Figure C2. W2 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

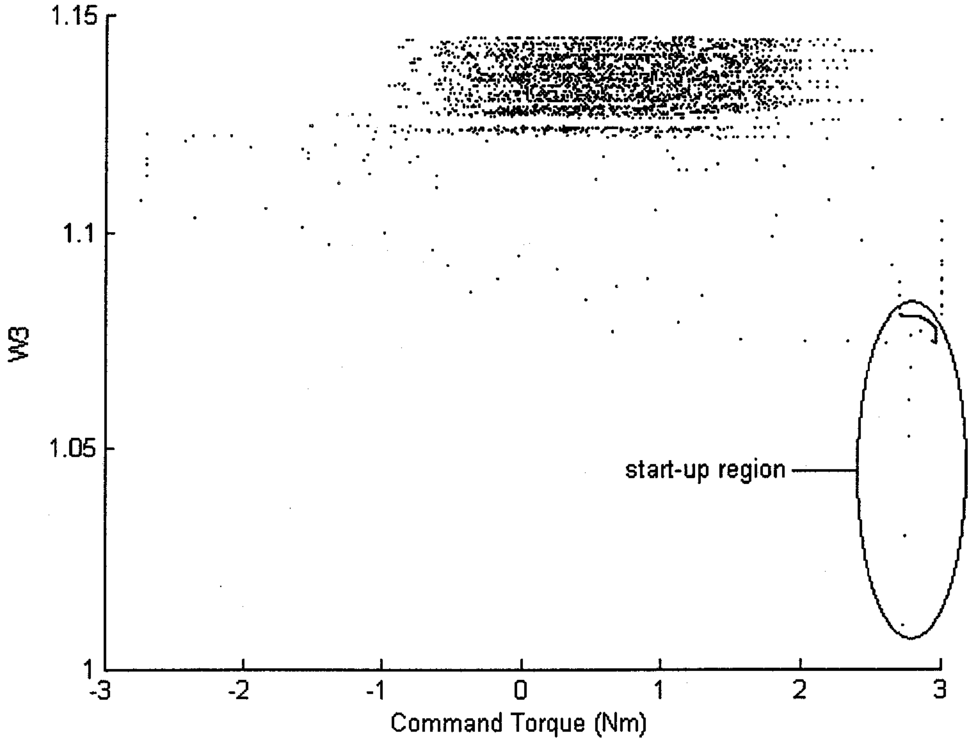


Figure C3. W3 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

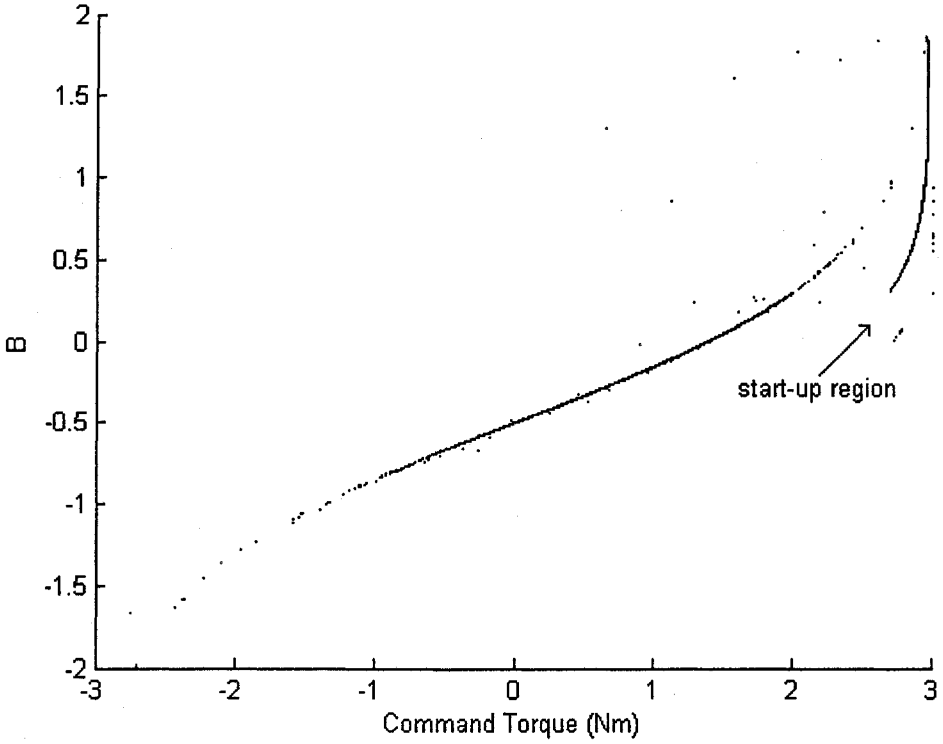


Figure C4. B versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 188.5 rad./sec.

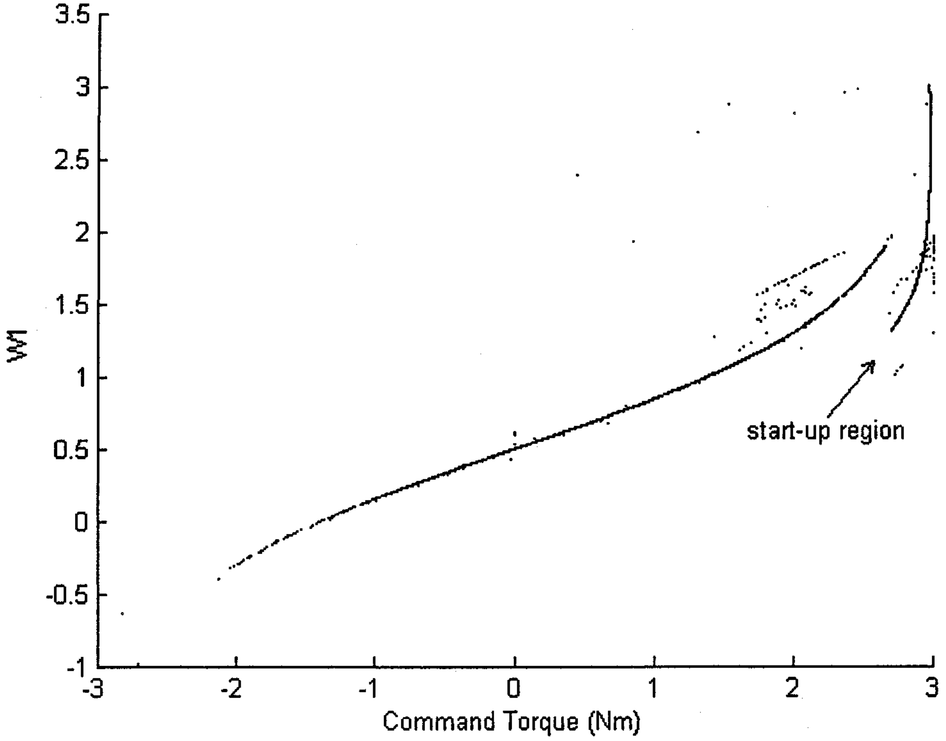


Figure C5. W1 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

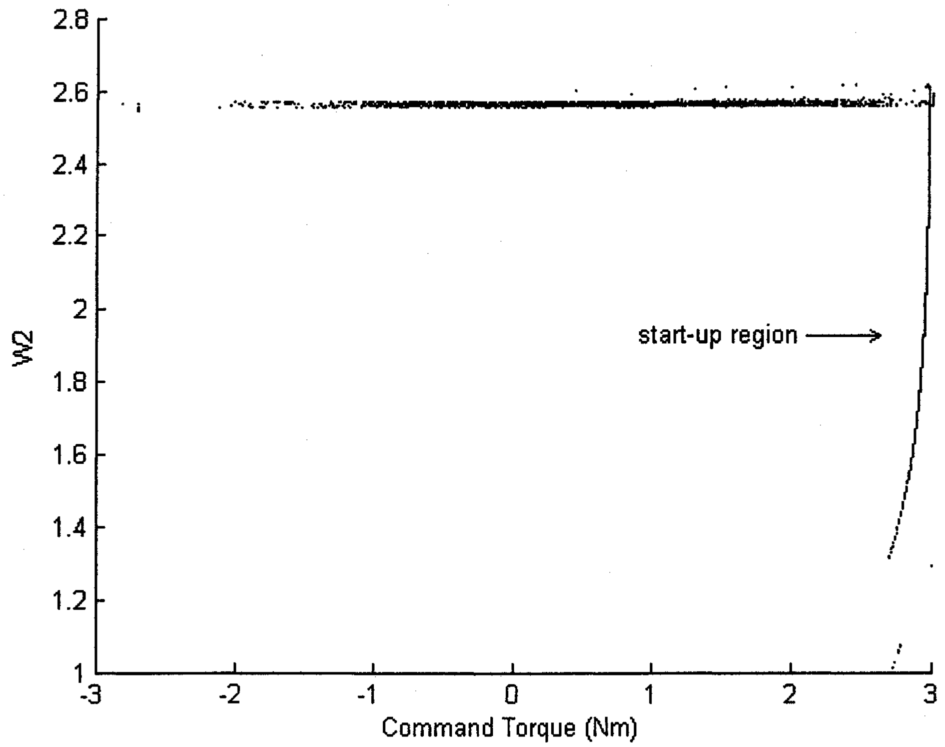


Figure C6. W2 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

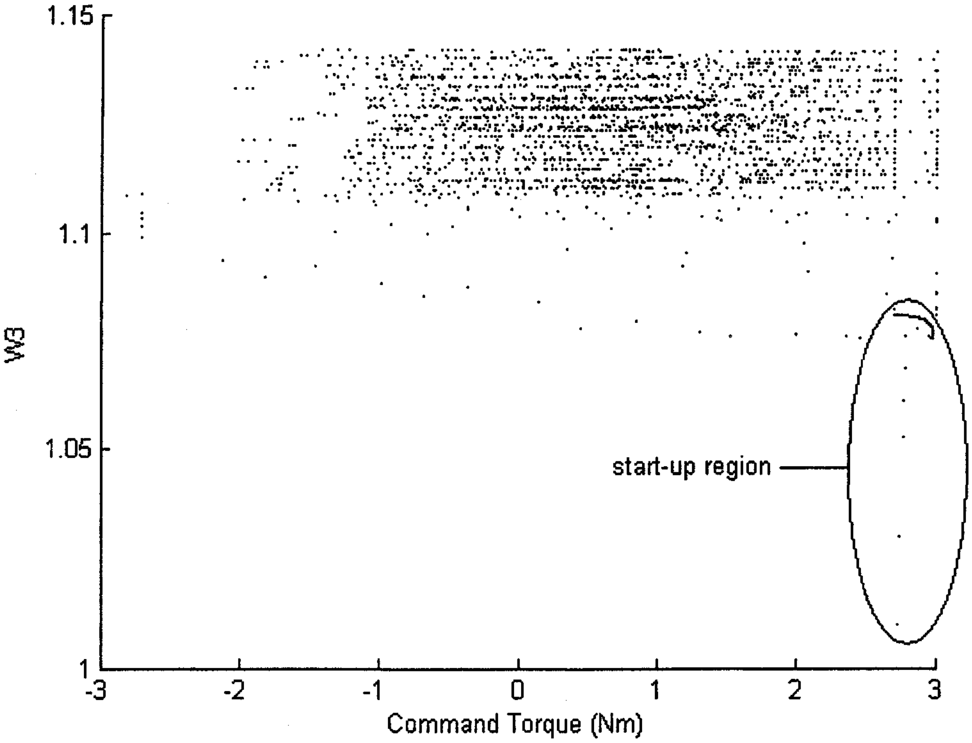


Figure C7. W3 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

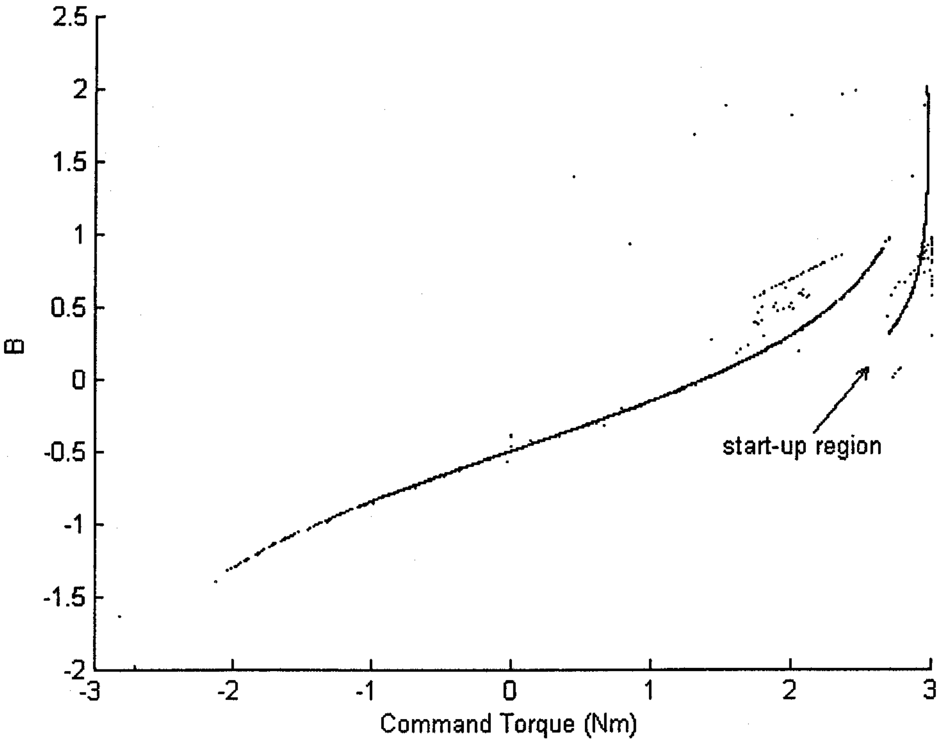


Figure C8. B versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 250 rad./sec.

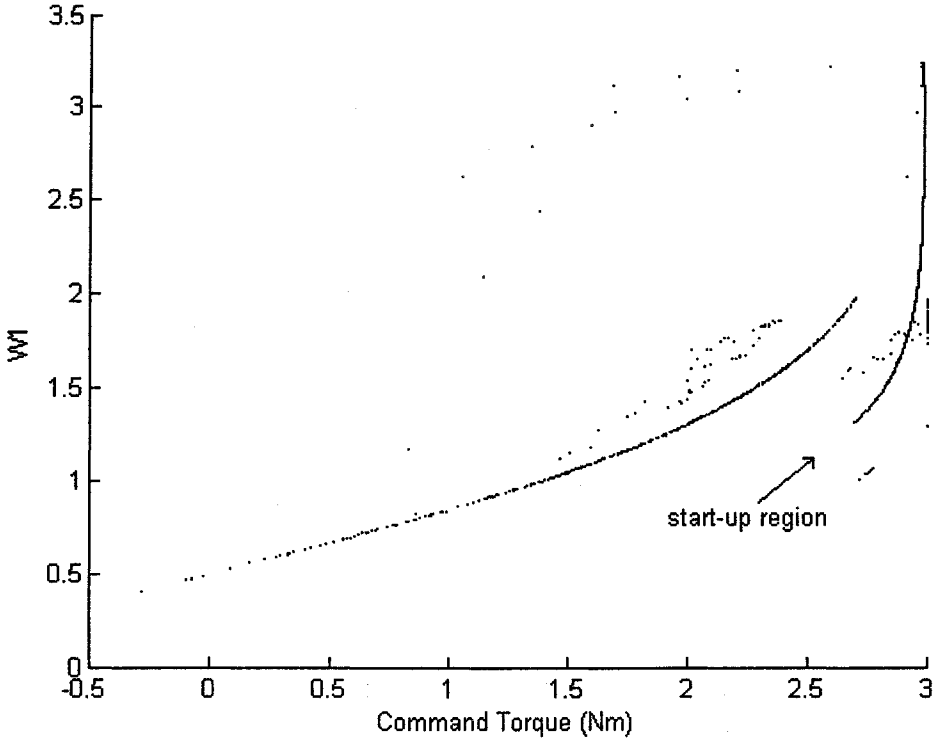


Figure C9. W1 versus Command Torque for SAN controller-based IPMSM drive under full-load (2 Nm) and command speed of 188.5 rad./sec.

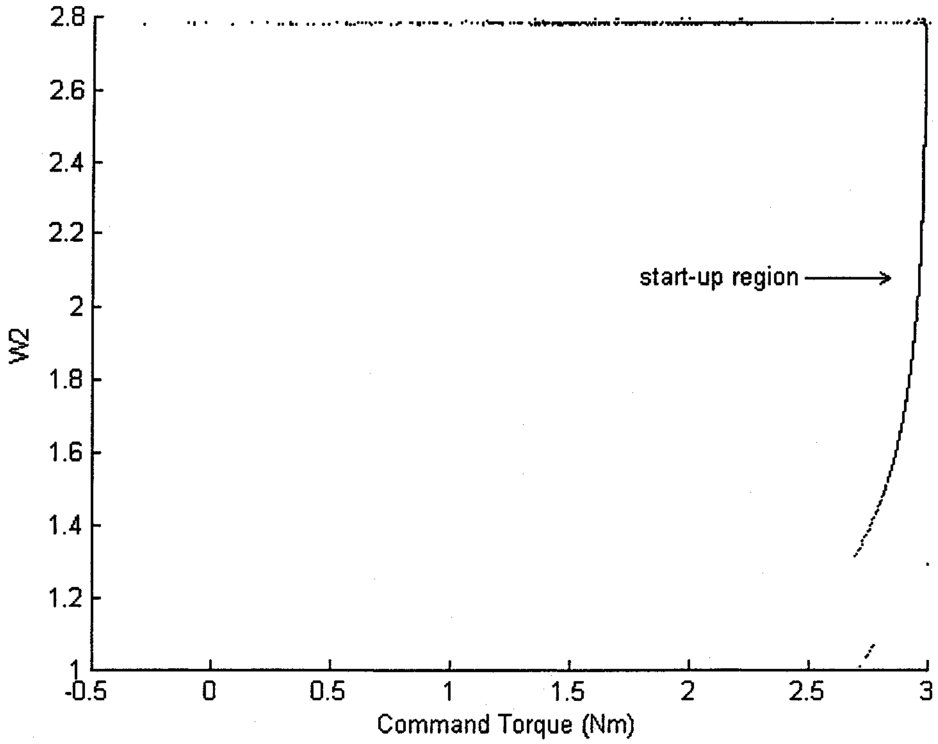


Figure C10. W2 versus Command Torque for SAN controller-based IPMSM drive under full-load (2 Nm) and command speed of 188.5 rad./sec.

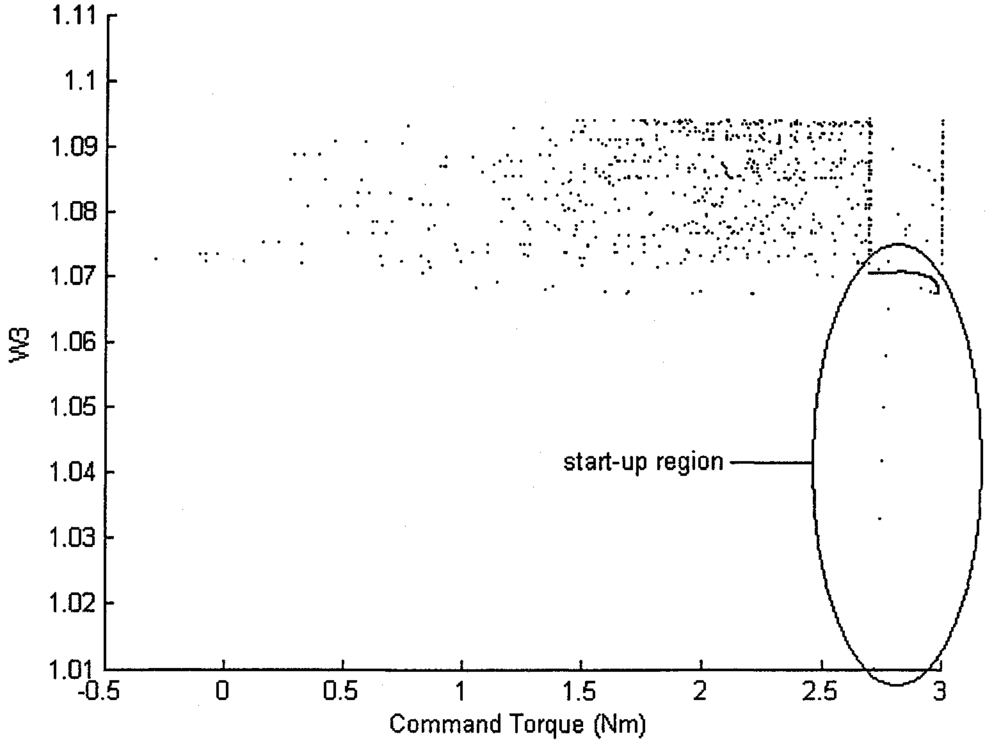


Figure C11. W3 versus Command Torque for SAN controller-based IPMSM drive under full-load (2 Nm) and command speed of 188.5 rad./sec.

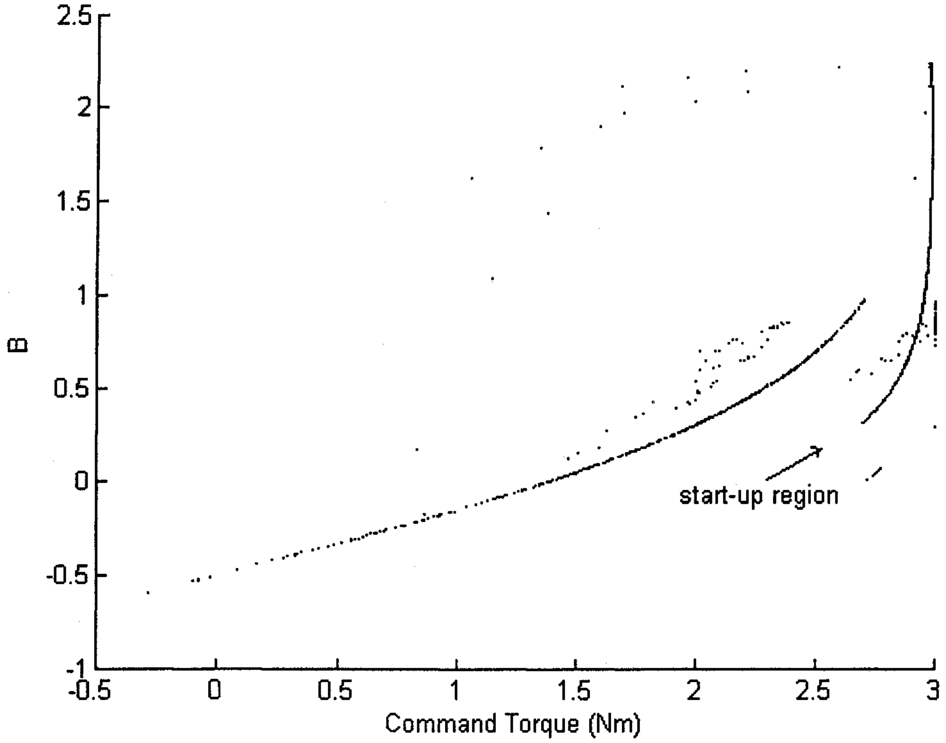


Figure C12. B versus Command Torque for SAN controller-based IPMSM drive under full-load (2 Nm) and command speed of 188.5 rad./sec.

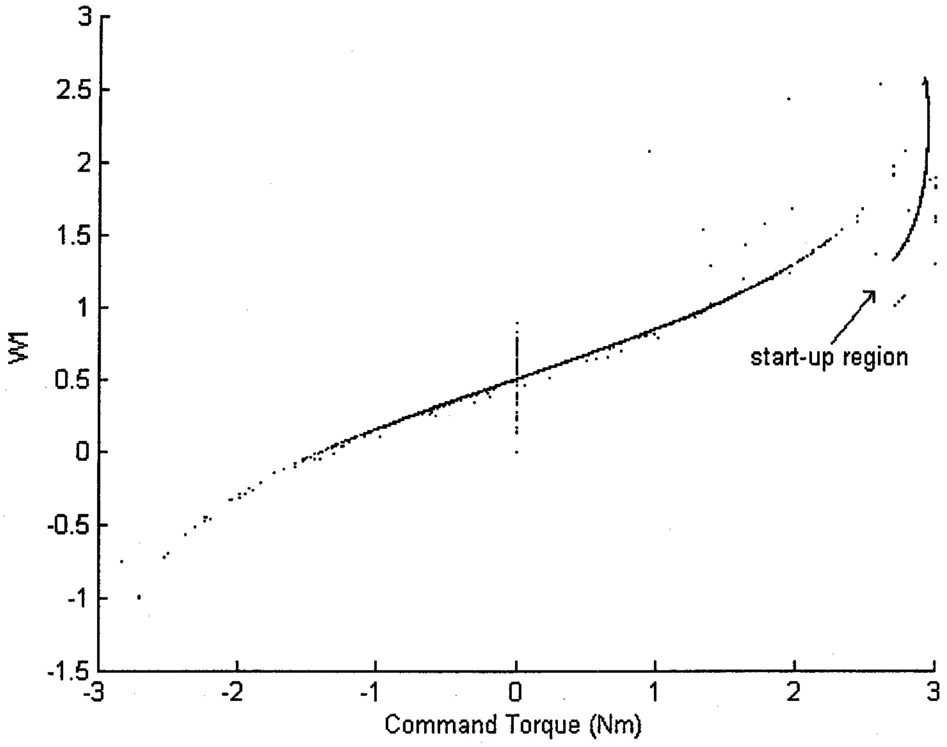


Figure C13. W1 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 90 rad./sec.

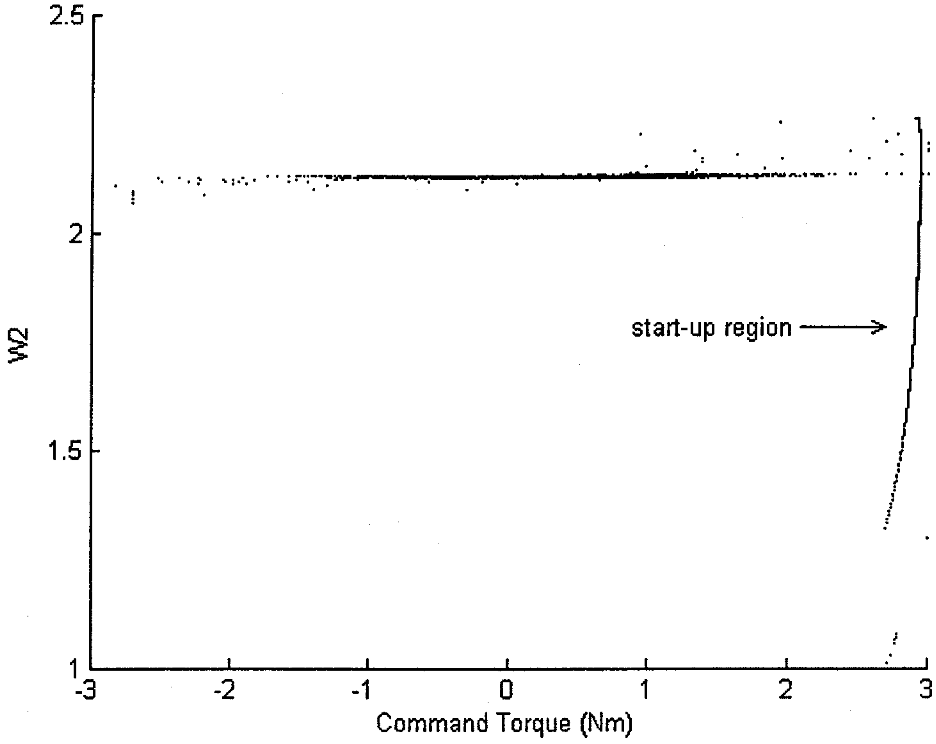


Figure C14. W2 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 90 rad./sec.

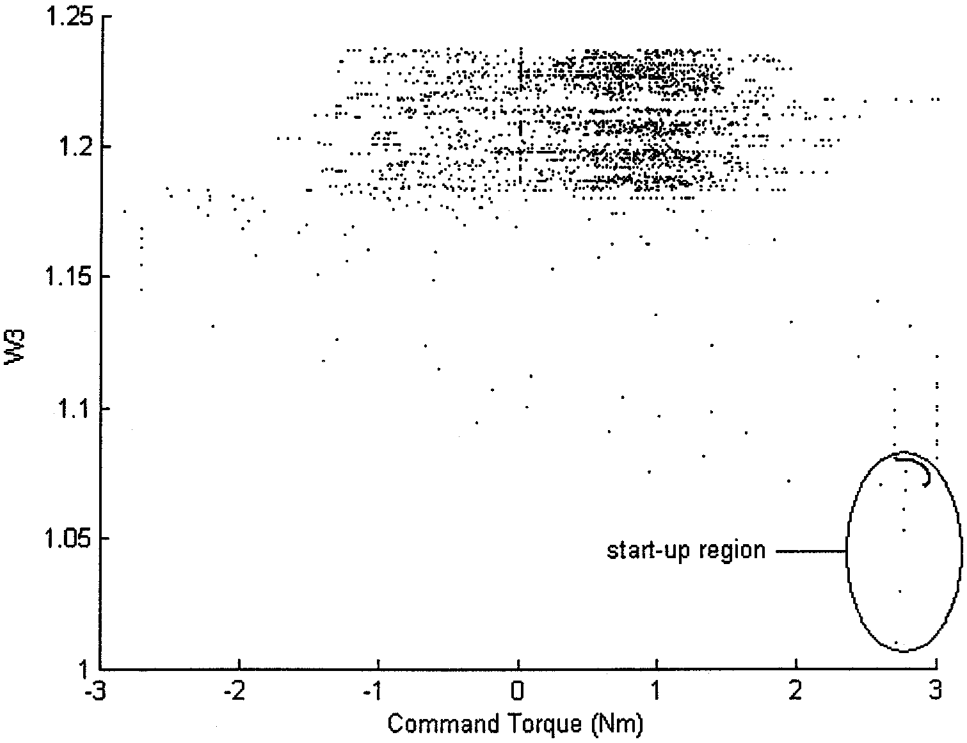


Figure C15. W3 versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 90 rad./sec.

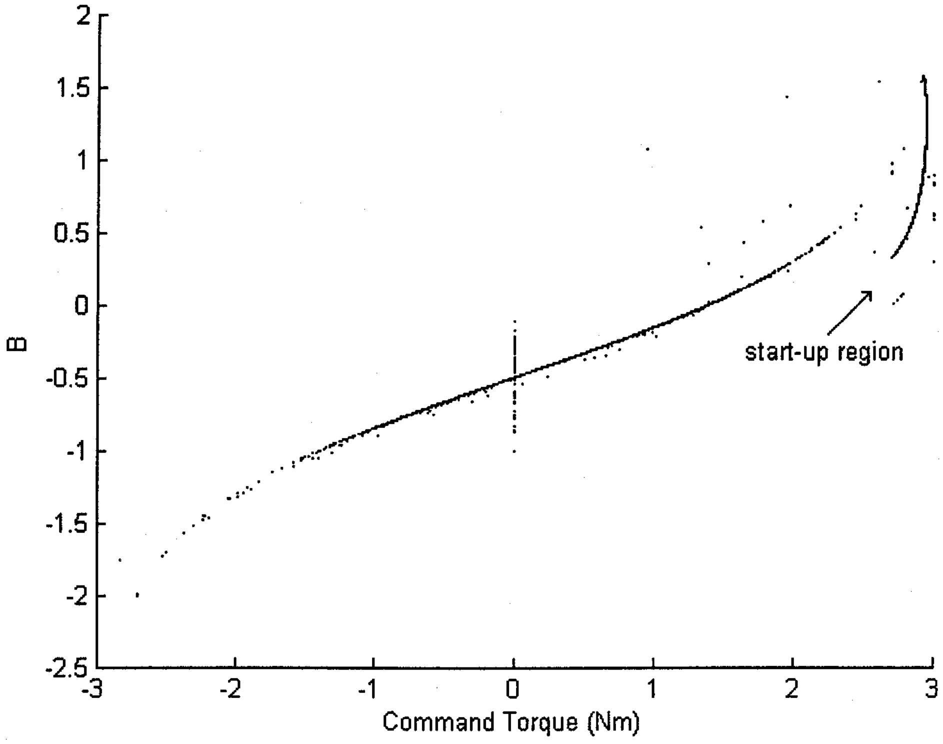


Figure C16. B versus Command Torque for SAN controller-based IPMSM drive under no load and command speed of 90 rad./sec.