HYBRID ELECTRIC VEHICLE POWERTRAIN: ON-LINE PARAMETER ESTIMATION OF AN INDUCTION MOTOR DRIVE AND TORQUE CONTROL OF A PM BLDC STARTER-GENERATOR

A Dissertation

Presented to

The Graduate Faculty of the University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

S.M. Nayeem Hasan

May, 2008

HYBRID ELECTRIC VEHICLE POWERTRAIN: ON-LINE PARAMETER ESTIMATION OF AN INDUCTION MOTOR DRIVE AND TORQUE CONTROL OF A PM BLDC STARTER-GENERATOR

S.M. Nayeem Hasan

Dissertation

Approved:

Advisor
Dr. Iqbal Husain

Committee Member
Dr. Tom T. Hartley

Committee Member
Dr. Joan Carletta

Committee Member
Dr. Yueh-Jaw Lin

Committee Member
Dr. Dale H. Mugler

Accepted:

Department Chair
Dr. J. Alexis De Abreu Garcia

Dean of the College
Dr. George K. Haritos

Dean of the Graduate School
Dr. George R. Newkome

Date

ABSTRACT

A hybrid electric vehicle (HEV) powertrain consists of both a mechanical power transmission path and an electric power transmission path. A supervisory vehicle controller generates the control commands for the subsystems in the powertrain based on the driver request and vehicle speed. Fuel efficiency and emissions from the internal combustion (IC) engine depend on use of the subsystems in both the power transmission paths. The major subsystems in the electric power transmission path (EPTP) are the motor drives that run either in the generating mode or in the motoring mode to process the power flow between the source and the wheels. In this research, two advanced motor drive subsystems with improved controllers have been designed and developed for an HEV powertrain. The two subsystems are the starter-generator electric drive and the propulsion motor drive. The contribution of this research will enable efficient utilization of the HEV powertrain.

An advanced electric drive controller for a high power starter-generator subsystem based on a permanent magnet brushless DC (PM BLDC) machine is presented. The PM BLDC machine is belt-coupled to a diesel engine in a series-parallel 2×2 HEV. The PM BLDC electric drive is developed for engine starting, generating and motoring. Computer simulations are performed for tuning the controller parameters, and for selecting proper

inverter rating of the starter-generator drive. The drive controller is implemented in hardware using Texas Instruments fixed point TMS320F2812 digital signal processor (DSP) and a high resolution current sensing board to achieve the best torque regulation at various load conditions. The PM BLDC starter-generator has been tested in both motoring (engine starting) and generating modes with the starter-generator mounted in the vehicle.

For the propulsion motor drive, an induction motor driven by a three-phase PWM inverter has been considered. The induction motor drive cannot deliver high static and dynamic performance without the correct parameter values in the controller. Computer simulations showed the parameter variation effects on the performance of an induction motor drive used in an electric vehicle. A novel Luenberger-sliding mode observer based induction motor controller with on-line parameter adaptation is then presented. Software-in-the-loop (SIL) and hardware-in-the-loop (HIL) simulations have been performed for a high power induction motor with electric vehicle load to verify the performance of the new Luenberger-sliding mode observer based parameter adaptation algorithm as well as to tune the control parameters. For the HIL simulation, the controller was implemented in an FPGA based control hardware, and a virtual motor model was implemented in software. The new on-line parameter adaptation algorithm has been tested experimentally on a small induction machine for a proof-of-concept demonstration. The developed algorithm provides fast convergence of parameters, rapid response characteristics of the drive, and accurate tracking of the control command for the induction motor drive. These performance features are highly desirable for the propulsion motor in HEVs and EVs.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my academic advisor, Dr. Iqbal Husain for his great efforts and a lot of enlightening ideas during the course of this research. I also wish to express my sincere appreciation to Dr. Tom T. Hartley, Dr. Joan Carletta, Dr. Yueh-Jaw Lin and Dr. Dale H. Mugler for being on my advisory committee, and for their support to make this research a success. I would specially like to thank the entire Challenge X team including administration, faculty, and students for their support towards the Challenge X program that has provided a platform for this research.

I would also like to thank my parents, my wife, and my brother and sister for their love and encouragement over the years.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

HYBRID VEHICLES AND ELECTRIC MACHINES

1.1 INTRODUCTION

Recent technological advancements in the field of power electronics and digital controls have caused revolutionary changes in electric drives for traction applications. Prior to the 1950s, variable speed and traction drives used DC machines, since AC machines were not capable of smoothly varying the speed of the system [1]. Technological developments have led to the increased use of new types of electrical machines including permanent magnet (PM) machines, induction machines and switched reluctance (SR) machines in traction applications. High efficiency and advanced manufacturing of electric machines have expanded their range of application in industrial and home appliances and also in various automotive systems. The applications of electric machines in automotives beside traction are: starter, alternator, power steering, fuel pump etc. These days, high power electric motors are used in different subsystems of electric vehicles (EV) and hybrid electric vehicles (HEV). Induction machines, permanent magnet machines and switched reluctance machines, all are potential candidates for high power starter-generator and propulsion systems of electric and hybrid electric vehicles. Improvements in control algorithms, and reduction in size and weight of electric

machines are important areas for research due to the increased use of high power electrical machines in automotives.

The recent development of digital signal processors (DSPs) in motor control applications has allowed electrical machines to deliver their highest performance in terms of torque-speed characteristics and dynamic behavior. Now complex control algorithms can be implemented, and these algorithms can be optimized considering efficiency and desired dynamic and static response. The performance and efficiency of electric machines depend not only on software, but also on hardware configuration. Numerous algorithms have been developed to replace measurements by estimations, which decrease the drive cost as well as increase robustness. The performance of a starter-generator or main propulsion type subsystem in an electric or hybrid electric vehicle depends on the efficiency, performance and robustness of the motor drives including the controller. In addition, the energy storage system of electric or hybrid electric vehicle must have sufficient capacity to supply enough power and energy to the electric motors of different subsystems so that the machines can operate at full capacity.

The most advanced control algorithms for a motor drive require a good knowledge of the machine analytical model. A motor drive cannot deliver good performance without having the correct machine parameters in the controller. Especially in HEV applications, incorrect machine parameters in the controller of the propulsion motor make a significant difference in vehicle performance. Therefore, the primary focus of this dissertation will be machine parameter estimation for efficient use of a propulsion motor. The control algorithm and motor drive selection for a high power starter/generator of an HEV will also be addressed in this research.

1.2 HYBRID ELECTRIC VEHICLES

Automobiles are an integral part of our every day life. Unfortunately, most automobiles use fossil fuels such as gasoline and diesel. Consequently, internal combustion (IC) engines release carbon monoxide, nitrogen oxides, carbon dioxide and hydrocarbons to the environment. The chemicals cause air pollution, acid rain, and buildup of greenhouse gases in the atmosphere. Electric vehicles (EV) powered by alternative energy provide the means for clean, efficient and environmentally friendly transportation. In EVs, an electric motor is the only propulsion unit, and power is supplied from a battery pack. Hybrid electric vehicles (HEV) that use both electric machines and an internal combustion (IC) engine for propulsion produce less emission as well as cause less air pollution than conventional automobiles. The IC engine used in an HEV is, of course, downsized compared to an equivalent IC engine vehicle. Electric vehicles first came to the market in the middle of 19^{th} century, even before gasoline-powered vehicles [2]. In the year 1900, 38% of the vehicles sold were electric powered. The invention of the starter motor for IC engines, improvement in engine technology, and availability of gasoline and inconvenience of battery charging challenged the existence of electric vehicles. However, during the last decade, motivated by concern over pollution and a future energy crisis, government and major automotive industries embarked on a number of initiatives to bring commercial EVs and HEVs into the market.

The architecture and component selection of powertrain of an HEV depends on vehicle architecture. The existing architectures for HEVs fall under the categories of series, parallel and series-parallel [2, 3]. In a series hybrid architecture, the IC engine acts

as a prime mover to drive an electric generator [2], but never delivers power directly to the wheels. The electric generator provides power to the propulsion motor through an energy storage link. In a parallel hybrid architecture, two energy sources provide propulsion power. A parallel hybrid drivetrain blends the power of the IC engine and the electric motor mechanically [3, 4] with both sources supplying power to the wheels in parallel. The series-parallel architecture is a mix of series and parallel hybrid drivetrain. Combining the advantages of series and parallel improves the performance and increases the fuel efficiency.

A hybrid electric vehicle has been developed at the University of Akron for a collegiate level national student competition known as Challenge X sponsored by General Motors and Department of Energy. The goal of the Challenge X competition is to reengineer a 2005 Chevrolet Equinox and convert it to a hybrid electric vehicle with the goals of increased fuel efficiency and reduced emissions without sacrificing performance or any of the customer satisfaction requirements. The HEV powertrain of the University of Akron vehicle has a series-parallel 2x2 architecture that includes an electric power transmission path (EPTP) and a mechanical power transmission path (MPTP). Figure 1.1 shows a block diagram of the hybrid electric vehicle drivetrain [5] developed for the Challenge X project at the University of Akron. The front wheels are powered by an IC engine, and the rear wheels are powered by an electric propulsion motor. The propulsion power to the wheels can be delivered by the IC engine and electric motor jointly or individually, depending on driving conditions and vehicle subsystem capabilities. The IC engine can provide power to the front wheels as well as to the rear wheels through the generator and propulsion motor. The engine also charges the energy

storage system through the generator. The vehicle can also run as an electric vehicle when the energy storage system supplies power to the wheel through the propulsion motor only. The choice of electric propulsion system depends mainly on three factors: driving profile, energy storage and vehicle constraints depending on vehicle type, weight and payload [3].

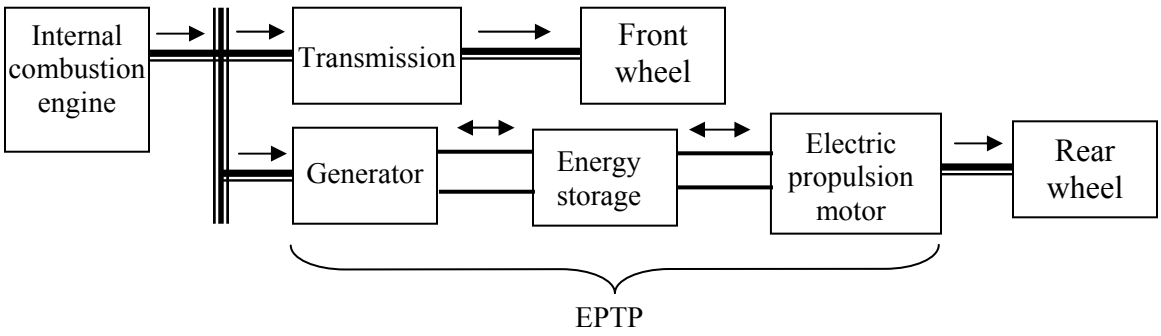


Figure 1.1 Block diagram of a series-parallel 2x2 hybrid electric vehicle drivetrain.

The power transmission paths of the hybrid electric vehicle consist of different mechanical and electrical subsystems. Optimum use of these subsystems increases fuel economy as well as reduces emissions. High performance and high efficiency of the electric power transmission path reduces the run time of the IC engine, thereby reducing emissions and increasing fuel efficiency. The Challenge X project provided a platform to design, implement and test various subsystems as well as conduct fundamental research on advanced motor drives and controllers. The research presented in this dissertation is focused on developing advanced motor drive subsystems for efficient use of the electric power transmission path.

1.3 ELECTRIC MACHINES

Electric machines can be categorized as DC and AC types. Prior to 1980s, DC motors were widely used in industries and in a number of prototype electric vehicles due to their developed technology and ease of control [2]. DC machines offer flexible torque-speed control and wide speed range operation, which is desired for an HEV propulsion motor [2]. DC machines are simple to control but they have low power to weight ratio, low efficiency, and require brush and commutator maintenance.

During the last three decades, AC machines have slowly replaced the DC machines due to the size and maintenance requirements of the latter. Recent electric and hybrid electric vehicles use AC machines both for propulsion and starter-generator applications. The types of AC machines used for these and other automotive applications are induction, permanent magnet and switched reluctance machines. These AC machines will be discussed next.

1.3.1 INDUCTION MACHINES

Induction machine technology is a mature technology with extensive research and development activities over the past 100 years. Recent development in digital signal processor and advanced vector control algorithm allow controlling an induction machine like a DC machine without the maintenance requirements [1]. Induction motors are considered as workhorses of the industry because of their low cost, robustness and reliability. Induction machines are used in electric and hybrid electric vehicle applications because they are rugged, lower-cost, operate over wide speed range, and are capable of

operating at high speed. The size of the induction machine is smaller than that of a separately excited DC machine for similar power rating. The induction machine is the most mature technology among the commutatorless motor drives. There are two types of induction machines: squirrel cage and wound rotor. In squirrel cage machines, the rotor winding consists of short-circuited copper or aluminum bars with ends welded to copper rings known as end rings. In wound rotor induction machines, the rotor windings are brought to the outside with the help of slip rings so that the rotor resistance can be varied by adding external resistance. Squirrel cage induction machines are of greater interest for industries as well as for EVs and HEVs. Instant high power and high torque capability of induction machine have made it an attractive candidate for the propulsion system of EV and HEV.

The three-phase stator windings in an induction machine are displaced by 120° (electrical) in space along the stator circumference. If three-phase voltages are applied to the stator, the stator magnetic field will cut the rotor conductors, and will induce voltages in the rotor bars. The induced voltages will cause rotor currents to flow in the rotor circuit, since the rotor is short-circuited. The rotor current will interact with the air gap field to produce torque. As a result the rotor will start rotating in the direction of the rotating field. The difference between the rotor speed and the stator flux synchronous speed is the slip speed by which the rotor is slipping from the stator magnetic field. The slip s can be represented by

$$s = \frac{\omega_e - \omega_r}{\omega_e} \tag{1.1}$$

where ω_e and ω_r are synchronous speed and rotor speed, respectively. The induced current in the rotor circuit also produces a rotating field. Its speed with respect to the rotor can be expressed as

$$\omega_{rs} = \frac{120sf}{P} = s\omega_e \tag{1.2}$$

where f and P are stator supply frequency and number of poles, respectively. Since the rotor itself is rotating at ω_r rpm, the speed of the induced rotor field in the air gap will be $\omega_r + \omega_{rs} = (\omega_e - s\omega_e) + s\omega_e = \omega_e$ rpm.

1.3.2 PERMANENT MAGNET MACHINES

Permanent magnet (PM) synchronous machines are widely used in industrial, transportation, and commercial applications. PM machines are challenging induction machines in EV and HEV applications due to their high energy density, compact size, high efficiency, and wide speed range operation. The availability of rare-earth permanent magnets in the 1960s paved the way for the development of PM machine technologies in high performance applications [6]. The principal difference between a PM machine and other types of rotating electric machines is the form of excitation. PM machines use permanent magnets in the rotor as the field exciting circuit, which produces air-gap magnetic flux. As a result, the permanent magnets provide a loss-free excitation without any external stationary electric circuit. However, the DC bus voltage control is not as easy as in an induction machine due to the permanent source of flux. The major concerns

about PM machines are broken magnet chips and demagnetization of the magnet due to heating caused by eddy currents at high speeds [2].

PM machines can be classified into two categories: permanent magnet synchronous machine (PMSM) and permanent magnet brushless DC (PM BLDC) machine. In a PMSM, the stator winding is sinusoidally distributed along the stator circumference producing a sinusoidal back-emf; in contrasts, the stator winding of a PM BLDC is concentrated which produces a trapezoidal-shaped back-emf. A three-phase balanced supply to the stator windings of a PMSM produces a sinusoidal magnetomotive force in the air gap. Permanent magnets in the rotor are shaped appropriately, and a sinusoidal rotor flux linkage can be established by controlling their magnetizing directions [2]. The electromagnetic torque is generated on the shaft by the interaction of the stator and rotor magnetic fields. There are three types of PMSM depending on the shape and position of the permanent magnets in the rotors: surface mount PM machine, inset PM machine and interior PM machine. The surface mount and inset PM machines are often collectively called surface mount PM synchronous machines. In the surface mounted PMSM, the magnets are easily epoxy-glued or wedge-fixed to the cylindrical rotor. An interior PMSM has its magnets inside the rotor, which is harder to construct but reduces eddy current effect on the magnets at high speed.

The PM BLDC machine is also called trapezoidal machine, since the back-emf is trapezoidal shaped. The PM BLDC basically operates like a DC machine but is electronically commutated through an inverter and position feedback information. In PM BLDC, current waveforms are rectangular switching polarity with alternate N and S magnet poles. Only six discrete rotor positions are necessary to synchronize phase current

with the trapezoidal-shaped back-emf for effective torque production. Rectangular shaped phase currents are supplied in synchronism with the back-emf of the respective phase. The rectangular shape of current waveforms helps to create constant electric torque. A set of three Hall sensors placed 120° apart can easily give the position information, which eliminates the need for a high-resolution encoder.

1.3.3 SWITCHED RELUCTANCE MACHINES

Switched reluctance (SR) machines are also gaining attention in HEV applications. They are inexpensive, reliable, have high fault tolerance, and weigh less than other machines of comparable power outputs. High torque-inertia ratio is an advantage for the SR machines. The SR motor is a doubly salient and singly excited reluctance machine with independent phase windings on the stator [2]. The stator winding is comprised of a set of concentrated winding coils. The rotor structure is very simple without any windings or magnets, and is made of magnetic steel laminations. Two major problems associated with SR machines are acoustic noise and significant torque ripple [2].

The SR machine is excited by a sequence of current pulses applied to each phase, and the energized phases cause the rotor to rotate in the motoring mode. The SR machine operates on the principle of varying reluctance. The reluctance is minimum (inductance is maximum) when stator and rotor poles are in the aligned position, and maximum when the poles are unaligned. A stator phase is energized when the reluctance for the respective phase is maximum. The adjacent rotor pole-pair gets attracted toward the energized stator

to minimize the reluctance of the magnetic path. When the reluctance is minimized, the next stator phase is energized. As a result, torque is developed in the direction of rotation.

1.4 RESEARCH MOTIVATION

The primary motivation for the research presented in this dissertation is to improve the overall performance and efficiency of electric and hybrid electric vehicles through the development of advanced controllers of motor drive subsystems. The major motor drive subsystems of the electric power transmission path (EPTP) of an HEV are the starter-generator and the propulsion motor. A detailed analysis and improved controller development of the starter-generator and propulsion motor subsystems used in the EPTP of an HEV have been performed in this research. Suitable component selection and sizing of subsystems used in the EPTP preceded the controller research and development activities.

This research also focuses on parameter variation effects on the performance of induction motor that is commonly used as a propulsion motor in HEVs and EVs. On-line parameter estimation for an induction motor controller is particularly important because variation in machine parameters with time and temperature degrades the performance of the motor drive subsystem.

1.5 RESEARCH OBJECTIVES

The primary objective of this research is to develop advanced controllers for motor drive subsystems (starter-generator and propulsion motor) to achieve higher efficiency in

the EPTP of an HEV. The sizing of these motor drives for the EPTP has also been addressed in the process. For the efficient use of the EPTP, the following research objectives are set forth:

- Drive system design and improved controller development of a high power starter-generator for a series-parallel 2x2 HEV.
- An investigation on the effects of parameter variation on static and dynamic performance of the induction motor drive used as a propulsion motor in an HEV or EV.
- Development of a new observer based on-line parameter estimation algorithm for induction motor drives, which is simple, easy to implement and overcomes the difficulties of existing methods.
- Design a controller that will adapt the estimated parameters in order to provide optimum performance of an induction motor drive in an HEV or EV.

1.6 DISSERTATION ORGANIZATION

The dissertation introduction addressed the research trends in the area of motor drives for HEVs. A brief description of the hybrid electric vehicle drivetrain followed by a presentation of different electric machines that are used in HEV powertrain subsystems was presented. The research motivation and objectives were then explained in detail.

Chapter II describes the architecture, components selection and sizing of the motor drive subsystems for the electric power transmission path, and highlights the issues with these subsystems that have motivated this research.

Chapter III presents control, drive structure, and modeling of advanced motor drives that have been selected for the HEV under consideration. A literature review on existing parameter estimation methods to improve the performance of propulsion motor drive system is also presented.

A PM BLDC starter-generator system with improved controller and hardware is presented in Chapter IV including simulation and experimental results. Chapter V describes the advanced controller with parameter adaptation algorithm for the propulsion induction motor drive including software-in-the-loop simulation results. Chapter VI is dedicated to hardware-in-the-loop simulation and experimentation of the induction motor drive with parameter adaptation.

Chapter VII concludes this dissertation, and presents future research topics related to the powertrain subsystems of an HEV/EV.

CHAPTER II

ELECTRIC POWER TRANSMISSION PATH

2.1 HEV POWERTRAIN

The propulsion power in a hybrid electric vehicle (HEV) comes from one or more traction electric motors and an internal combustion engine. The propulsion power is transmitted to wheels through either the mechanical power transmission path (MPTP) or the electric power transmission path (EPTP), or the combination of the two. The vehicle powertrain is designed to meet the vehicle base load as well as the peak load during acceleration and starting. The University of Akron hybrid vehicle under consideration has a series-parallel 2x2 architecture with two electric machines and one IC engine, as shown earlier in Fig 1.1. The MPTP is associated with an internal combustion (IC) engine and transmission, whereas the EPTP consists of an energy storage system, a generator, a propulsion motor and transmission. The generator machine also serves as the engine starter of the vehicle.

2.1.1 ELECTRICAL COMPONENTS

Figure 2.1 shows the electrical components in the powertrain of the series-parallel hybrid electric vehicle. One electric machine (labeled as “Generator”) is coupled with the

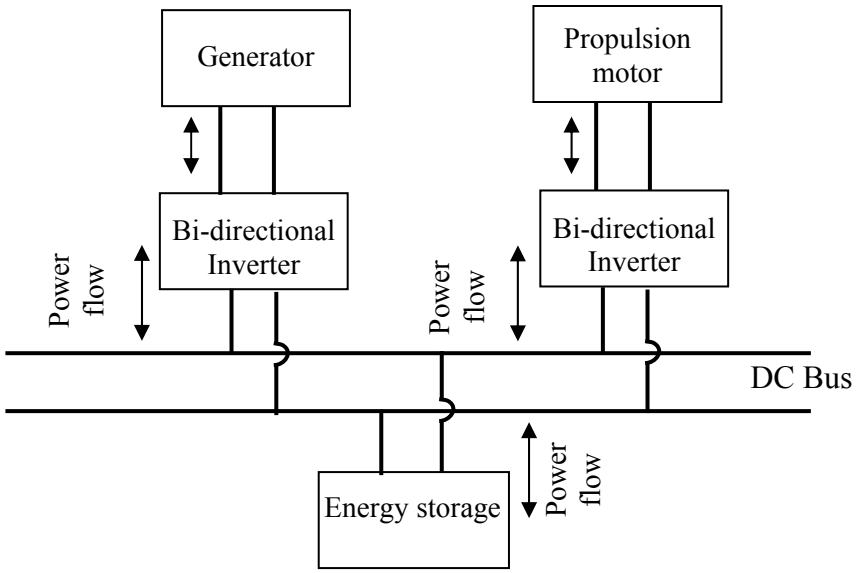


Figure 2.1 Electrical components of a series-parallel hybrid electric vehicle.

engine and can be operated as a generator as well as a motor. During generation, the power through the generator can be used to charge the energy storage using a bi-directional inverter, or to deliver energy directly to the propulsion motor through the DC bus. The generator can also be operated as a motor during engine starting and torque boosting. The energy storage system will absorb or deliver power depending on the system state of charge and driving conditions. Another bi-directional inverter conditions the flow of power for the propulsion motor, which delivers torque to the wheels. The propulsion motor can also capture regenerative energy during vehicle braking.

The fuel efficiency and run time of the IC engine in a hybrid electric vehicle depend on the efficient use of the electrical components. The components need to be selected with a suitable torque-speed operating envelope that will deliver the desired vehicle performance. The physical size of the components is also critical since they need to be properly packaged and mounted within the vehicle.

2.1.2 ELECTRIC MACHINES FOR HEV

The starter-generator and propulsion motor in the EPTP use high power electric machines. These electric machines need to have motoring and generating capability, high power density, high efficiency, and high starting torque over a wide speed range to meet performance specifications. Any one of the three machine drives, induction, PM or SR, can meet the requirements of a starter-generator and propulsion system when designed accordingly. The selection depends on the subtle features of the machines and their power electronic drives and the availability in the desired time-frame.

The plot of an electric machine torque-speed characteristic is shown in Figure 2.2. The motor delivers rated torque (T_r) up to the rated speed or base speed ω_{base} where the motor reaches its rated power condition. In the constant power region, the motor can operate at speeds higher than the rated condition but the delivered torque decreases. The natural characteristics region can be used to extend the operating region of certain motors. The power electronics based motor drive enables electric motor operation at any

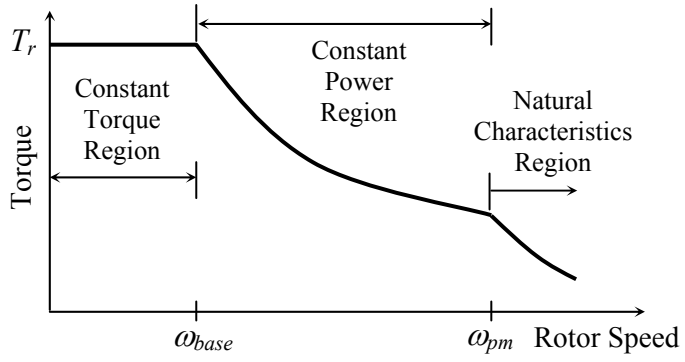


Figure 2.2. Electric motor torque-speed envelope.

point within the envelope. In HEV applications, transmission gears are used to match the higher speed of the electric motor with the lower speed of the wheels.

2.1.3 IC ENGINES

Four-stroke gasoline/patrol engines and diesel engines are both used in HEV applications. The selection of an IC engine for an HEV application is based on maximum power and torque output, brake specific fuel consumption, emissions, efficiency and driving performance [3]. The engine is sized to supply efficient power to overcome the road load comprised of aerodynamic drag, rolling resistance and roadway grade during the charge sustaining mode of operation.

The ignition in gasoline engines is initiated by a spark plug, whereas diesel engines require only compression of fuel to start combustion. Compression ignition engines with turbocharger operate more efficiently than spark ignition engines because of higher compression ratio and high combustion temperature [7]. Turbocharging and supercharging increase the power output of the compression ignition engines allowing further size and weight reduction [7]. Moreover, diesel engines use less fuel when idling.

The cranking torque and speed of the IC engine define the size of the starter motor. The starting torque of the engine depends on the compression ratio. The diesel engines have compression ratios of 14:1 to 23:1, whereas the gasoline engines used in conventional vehicles have compression ratios of 7.5:1 to 10.5:1 [8]. Because of the high compression ratio, the diesel engine requires more starting torque compared to a gasoline

engine of the same size. Diesel engines with sizes ranging from 1.6L to 2L require starting torque from 80Nm to 100Nm at speeds of 800rpm to 1200rpm [8].

2.2 HIGH POWER STARTER-GENERATOR

An electric machine serves as the starter for the IC engine, and additionally, another electric machine is needed to generate power for the auxiliary systems in a vehicle. The technology trend is to combine the IC engine starter and the generator into an integrated starter-generator system in both conventional and hybrid electric vehicles. Greater power generation capability of the starter-generator improves fuel economy and emissions of hybrid vehicles [9]. The starter-generator also performs engine cranking as well as torque boosting, which helps to downsize the IC engine of hybrid vehicles [10]. Engine cranking time, torque ripple and current drawn from source are important factors during engine starting. A typical 4-cylinder patrol engine requires 80 amperes to 200 amperes with 12V/14V dc bus system, whereas a typical 4-cylinder diesel engine requires 200 amperes to 300 amperes. Moreover, the engine cranking time is about 800ms-2s with a 12V (1.6kW to 2.2kW) dc starter motor depending on the engine size [11]. The cranking time can be reduced to less than 300ms with a high power starter-generator system. Since a higher compression ratio makes the cranking harder, diesel engines need more cranking time and generate more engine torque ripple with conventional low power starter-generators [12].

In [13], the development of a 42V, 7.7kW starter-generator using induction machine to start a 1.4L diesel engine has been presented. The cranking time was 200ms

at 200rpm with significant torque ripple. The maximum dc current drawn for cranking is about 400 amperes, which exceeds the industry standard safe value (about 200 amperes) of conductor current handling in a vehicle. In [12, 14], high power starter-generators using induction and switched reluctance (SR) machines have been demonstrated. In [14], performances of an induction machine and an SR machine with the same power rating (8kW) have been analyzed and compared. Compact winding which meets vehicle packaging requirements, and low rotor inertia are the significant benefits for SR machines [14]. SR motor requires accurate knowledge of the rotor position relative to stator. SR machine requires four times the resolutions of the induction motor encoder for proper control [14]. Other significant problems for SR machines during generation are torque ripple and acoustic noise. Larger induction machine size for the same power rating compared to an SR machine is an issue for in-vehicle packaging. Wide speed operation is necessary in the power generation mode. Though field weakening is possible, the size of induction machine increases to achieve the same power rating in the field-weakening mode compared to other machines. Moreover, the efficiency of an induction machine is lower compared to an SR machine and a permanent magnet machine.

In [15, 16], PM synchronous machines (PMSM) have been studied for an integrated starter-generator application. The significant benefits of a PMSM based starter-generator are: high efficiency, high power density, compact size, and wide speed range operation with interior permanent magnet structure. The main problem with PM synchronous machines is the difficulty of output voltage regulation to compensate for speed and load variations [17]. Since the features of a permanent magnet brushless DC machine (PM BLDC) are similar to that of a PMSM except for the trapezoidal shaped

back-emf, PM BLDC machine is also a good candidate for the starter-generator application.

The low power starter-generator used in the conventional vehicles can start the engine, but cannot meet the power generation requirement for an HEV. Moreover, engine torque ripple is undesirable especially for diesel engines when the starting time is relatively longer. A high power starter-generator is the only solution, which can start the engine faster with less torque ripple and can meet the power generation requirement.

2.2.1 RESEARCH SCOPE IN STARTER-GENERATOR TECHNOLOGIES

A sports utility vehicle sized hybrid electric vehicle requires a high power starter-generator that has high efficiency, high energy density, high starting torque, wide speed operation range, simple control, reasonable cost and compact size. Such a high power starter-generator is unavailable today either as a product or a prototype. Moreover, the design and development guidelines for such a starter-generator are non-existent in the literature. This research presents a sizing methodology for a high power starter-generator. Methods for determining optimum operating point and developing a controller with fault management are presented in this research, and will serve as a guideline for a starter-generator subsystem.

2.2.2 MACHINE SELECTION FOR STARTER-GENERATOR

PM BLDC machines offers the highest power density of all the available machines when high energy magnets are used, which makes them very attractive candidate for the

high power starter-generator subsystem of an HEV [4]. Moreover, the PM BLDC offers high starting torque with a relatively simple control method. The trapezoidal shaped back-emf of the PM BLDC machine enables the control of the PM BLDC to be like that of a DC machine but with an electronic commutator. The control of PM BLDC machines is easier than that of other machines, and requires no vector control, unlike induction machines and PM synchronous machines. Moreover, the modeling of the PM BLDC machine is easy and can be implemented in simulation for controller parameter tuning. Output voltage regulation problem can be eliminated by controlling the machine in the current control mode. A compact PM BLDC machine with high power density can be easily integrated with the engine in the engine compartment. The PM BLDC with a speed ratio of 2-2.5 between maximum speed and base speed is a good choice for a high power starter-generator.

2.2.3 STARTER-GENERATOR SIZING

The generator is typically sized to maintain series operation of the vehicle for typical urban driving conditions. Engine starting torque and torque boosting requirement must also be considered for sizing the starter-generator. In the series-parallel 2×2 HEV, the starter-generator system must be controlled as a generator over a relatively wide speed range, in comparison to that in a conventional series hybrid vehicle [18]. It also must deliver more power as a generator than the starter/generator system in a typical parallel hybrid vehicle. The traction power required for a steady state cruising velocity V for a vehicle of mass m is given by

$$P_{TR} = V\left(mg\sin\beta + \left[mgc_1 + \frac{\rho}{2} A_F c_D \right] V^2 + mgc_0 \right) \tag{2.1}$$

where

m = Vehicle mass + Driver and one passenger mass

c_0, c_1 = Rolling resistance coefficients, c_D = Aerodynamic drag coefficient

A_F = Frontal area

β = Angle of roadway slope, ρ = Air density.

The power required for cruising at a constant velocity of 40 mph with a driver and a passenger on a 1% grade ($\beta = \tan^{-1}(0.01) = 0.5729$), and 10% drivetrain losses for the University of Akron HEV under consideration is 12kW. The vehicle mass is 1995kg (mass of a crossover Chevrolet Equinox SUV), and the driver and passenger mass together has been considered to be 160kg. The vehicle parameters are: $c_0 = 0.01$, $c_1 = 0\,\text{s}^2/\text{m}^2$, $c_D = 0.45$ and $A_F = 2.5\text{m}^2$. The c_0 and c_D parameters are dimensionless. Allowing for power to recharge the energy storage, and considering engine starting as well as torque boosting requirements, a 20kW machine would be ideal for the integrated starter-generator of this HEV. A 21kW (continuous) PM BLDC machine was available, and hence selected for the project. The machine is liquid cooled with a power density of 875W/kg, which is an excellent figure of merit. None of the other machine types would be able to provide such high power density. The machine has a continuous torque rating of 37.5Nm (peak torque of 60 Nm) and a corner speed of 4,750rpm. The maximum speed of the PM BLDC is 12,500rpm. Moreover, this PM BLDC is small enough to be easily integrated in the engine compartment.

2.2.4 STARTER-GENERATOR AND ENGINE OPERATING POINT

The IC engine chosen for the Challenge X series-parallel 2X2 HEV is a Volkswagen 1.9L, 76kW diesel engine that is directly coupled to the starter-generator by a fixed gear ratio. The operating point should be set where both the starter-generator and the IC engine can operate to maximize the benefits for hybrid operation. Fuel economy, reduction in emission, and efficiency were considered in finding the operating point [19]. The 1.9L diesel engine has been characterized through a series of dynamometer testing, which showed that the efficiency peak is at 1,700rpm. The diesel engine has a maximum speed of 4,500rpm, but should not be operated above 4,250rpm. The ideal operating point for the PM BLDC starter-generator is at its corner speed (4,750rpm) in the generating mode. The final gear ratio was chosen to be 2.8:1 to give the maximum torque multiplication available within the speed constraints.

A few vehicle level experimental results are presented in this section to explain the optimum operating point of the starter-generator and the engine. Figure 2.3 shows the combined efficiency of the diesel engine and starter-generator combination as a function of load. The tests were conducted in generating mode at the maximum continuous torque rating (37.5Nm) of the generator, which reflects 105Nm on the engine side. It can be seen that the efficiency is still trending slightly upward but that the optimal point from an efficiency standpoint has nearly been reached. From this figure, it is also seen that the combined efficiency reaches its best point of 36.4% at the engine speed of 1700rpm (≈4750rpm at generator side) for an output power delivered to the DC bus of 18.65kW that is very close to the maximum continuous power rating of the generator. For the

generator, a load torque of 37.5Nm at the speed of 4750rpm is equivalent to an output power of 18.65kW ($P = T\omega$). From the tests, it can be concluded that the optimum operating point of the engine with respect to combined efficiency is at 105Nm (77.43lb-ft) and 1700rpm. Figure 2.4 shows the Brake Specific Fuel Consumption (BSFC) for the IC engine selected for this project. From this figure, it is observed that BSFC of the engine increases by only 4.8% from the minimum of 0.33025 lb_m/(hp-hr) at the selected operating point (77.43lb-ft, 1700rmp). Figure 2.5 shows brake specific NO_x production for the selected IC engine at different operating points. From this figure, it is seen that the predicted brake specific NO_x emissions will be reduced by 30% from the maximum of 0.016268 lb_m/(hp-hr) for the same operating point and BSFC. The slight increase in BSFC in the engine required to match the specifications of the starter-generator worked out quite favorably in reducing NO_x emissions. Therefore, the selected operating point

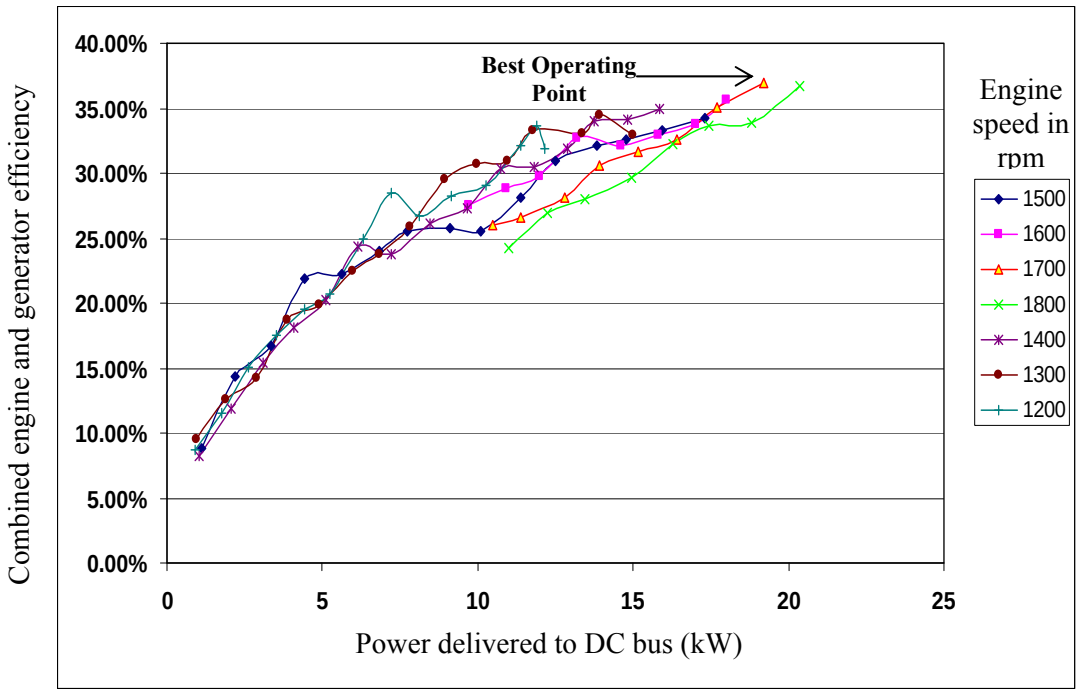


Figure 2.3. Engine and generator combined efficiency as a function of power delivered to the DC bus [20].

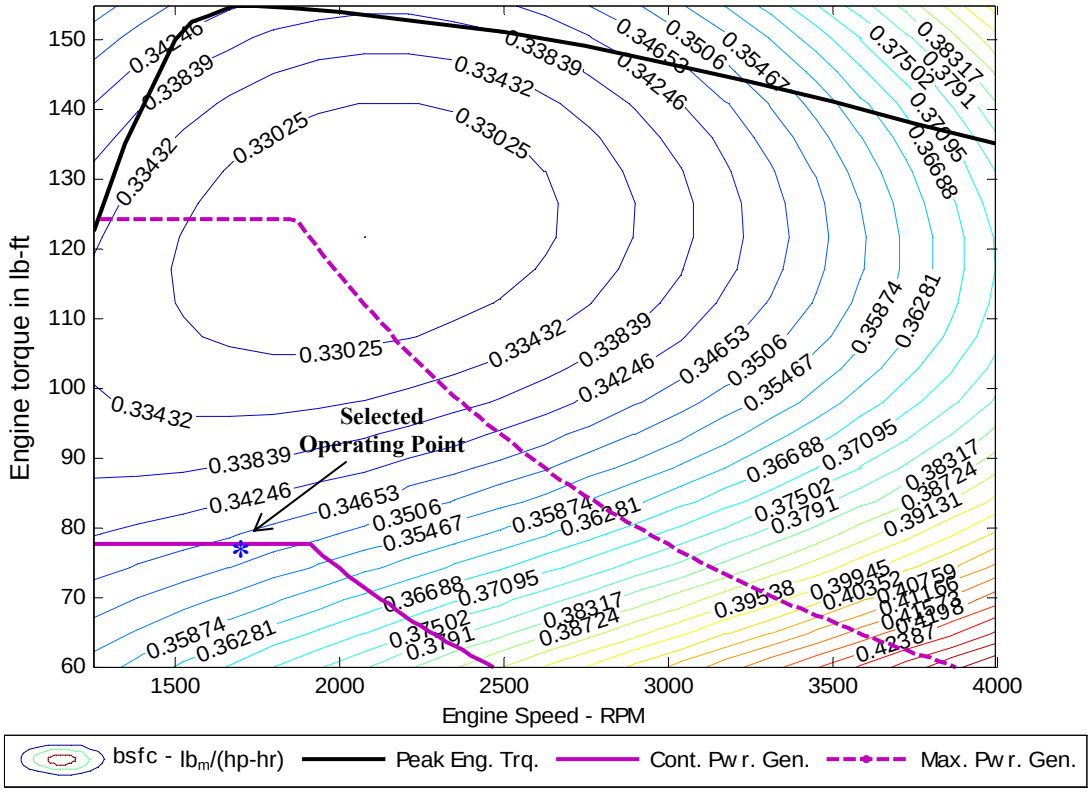


Figure 2.4. BSFC for the Volkswagen 1.9L diesel engine under consideration [20].

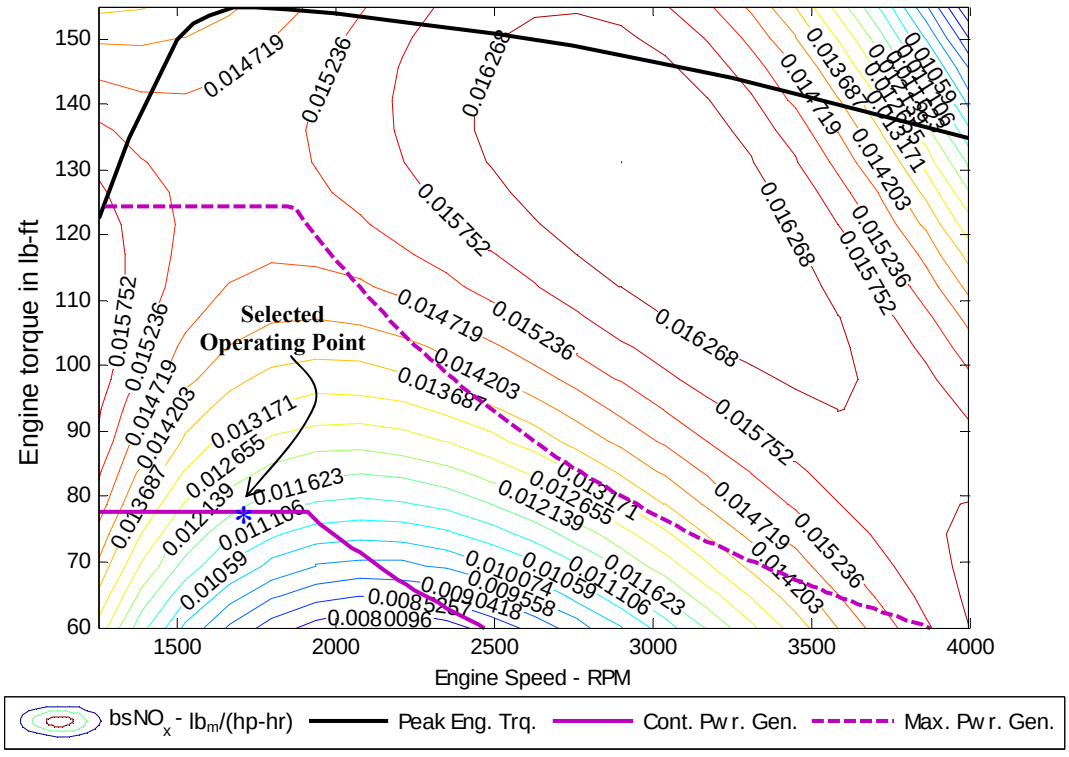


Figure 2.5 Break specific NO_x emissions for the Volkswagen 1.9L diesel engine [21].

(105Nm, 1700rpm) for the engine is a desirable point where the engine produces relatively less emissions, and exhibits reasonable efficiency while allowing the generator to run at its maximum continuous power rating (37.5Nm, 4750rpm) at its highest efficiency point.

2.3 PROPULSION MOTORS

The propulsion motor is another essential motor drive subsystem in the EPTP of an HEV or EV. Electric machines provide propulsion power in HEVs in conjunction with the IC engine. The electric machines for HEV must have the capability to deliver power as well as to reverse the direction of power flow when the vehicle is braking. The regenerative capability of electric machines extends the zero emission range of the vehicle. The important characteristics of a propulsion motor for an HEV include flexible drive control, fault tolerance, high power density, high instant torque, low acoustic noise and high efficiency [2]. The propulsion electric motor must deliver high torque and high power over a wide range of vehicle speeds to eliminate gear shifting as well as to meet the acceleration requirements [22]. DC machines, induction machines, PM machines and switched reluctance machines are the potential candidates for the HEV electric propulsion subsystem.

In early stages of HEV development activities, DC motors were used as propulsion motors because their torque-speed characteristics suit the traction requirements well and their controls are simple. A DC motor has been used as a propulsion motor in the PSA Peugeot-Citroen/Berlingo (France) HEV [23]. As technology for advanced controllers

has matured, AC machines forced DC machines out of contention for EV/HEV applications.

Induction motors are widely used as propulsion motors in HEV/EV applications, such as in HEVs Renault-Kangoo (France), Chevrolet-Silverado, DaimlerChrysler-Durango and BMW-X5 [22, 23]. In [24], an induction motor has been proposed as a propulsion motor for a series hybrid electric vehicle because of its capability of delivering high torque at low speed, high efficiency over the entire range of operation, and wide speed range. Wide speed range operation is achieved by field weakening, but a machine with a higher overall machine power rating was chosen in order to achieve rated power in the field weakening region [25]. In [26, 27], advanced induction motor drives have been presented for a hybrid vehicle propulsion system.

High power density, high efficiency, and smaller size and weight compared to induction machines have made permanent magnet synchronous machines currently a popular propulsion motor in HEV applications [3, 28]. PM machines are now being used in Nissan-Tino, Honda-Insight, Honda-Civic, Toyota-Prius, Toyota-Estima and Ford-Escape HEVs in the vehicle propulsion system [22, 23]. The rotor of a PM machine has no windings but only permanent magnets, which make the PM machine more expensive than other AC machines. A major difficulty with the PM machines at high speed is magnet heating due to eddy current, which may cause demagnetization [2]. Interior permanent magnets in the rotor of the machines allow the high-speed operation by field weakening, but this measure increases the production cost [29]. The PM brushless DC motor (PM BLDC) is also promising in HEV applications due to higher efficiency and

power density. For maximizing the energy efficiency, PM BLDC is a better choice although it does not allow a wide constant power range operation [4].

Switched reluctance (SR) machines have also been recognized as a potential candidate for HEV propulsion. SR machines have high fault tolerance, low manufacturing cost, and outstanding torque-speed characteristics for HEV propulsion [3, 30]. SR machines also have high starting torque and a high torque-inertia ratio. The SR machine has been used in the Holden/ECOmmodore (Australia) HEV for its propulsion system [23]. The major disadvantage associated with SR machines for traction applications is the high torque ripple [2, 4].

2.3.1 INDUCTION MACHINE FOR PROPULSION SYSTEM

In [23], a comparative study has been carried out among induction motors, PM machines and SR machines. The factors considered in this study are power density, efficiency, controllability, reliability, technological maturity, and most importantly cost. The study has revealed that the induction motor is the most suitable choice for HEV propulsion system even if the PM machines are hard competitors [23]. The major advantages of using induction motors for HEV/EV applications are the fastest torque response possible due to the small leakage inductances, and the ability to operate in hostile environments. Induction machines are reliable, cheap, rugged, and have a decent constant power region. In addition, the absence of no load spin losses in an induction machine drive where it is directly connected to the wheels is an advantage. The bus voltage control is easier with induction motor propulsion drives since there is no

permanent source of flux like in PM machines. Moreover, the induction motor shows a good efficiency over a wide speed range of operation. Therefore, the induction motor would be a decent choice for HEV/EV propulsion system. The advanced controller development performed in this research will focus on induction motor drives.

2.3.2 PROPULSION MOTOR SIZING

The parameters of the 2005 Chevrolet Equinox SUV based on which the University of Akron HEV is built have been used to size the propulsion motor. The tractive and drive force F_{TR} required for a vehicle can be calculated from the following equation [2, 31].

$$F_{TR} = m\frac{dv}{dt} + F_{roll} + F_{AD} + F_{grav} \tag{2.2}$$

where F_{roll}, F_{AD}, F_{grav}, m, and v are rolling resistance force, aerodynamic drag force, gravitational force, vehicle mass, and velocity, respectively. The tractive force is defined as the force required overcoming the resistive forces to move the vehicle. The instantaneous tractive power can be represented as

$$P_{TR}(t) = F_{TR}(t)v(t) \tag{2.3}$$

where $v(t)$ is the instantaneous velocity. The initial acceleration requirements are met by operating the IC engine and electric motor at their peak torque capabilities until the power limit of the two is reached [31].

The total propulsion power required for an acceleration of 0-60mph in 9sec has been calculated assuming that the vehicle accelerates with constant torque from the

propulsion system, and then switches to constant power mode once the power limit of the components are reached. The combined (engine and propulsion motor) power required has been found to be 132kW [31]. Since the IC engine delivers 76kW, the rest of the power (56kW) needs to come from the propulsion motor. Therefore, a 56kW propulsion motor would be required for this HEV to meet the initial acceleration requirement. An integrated induction motor drive system with 65 kW peak rating (45 kW continuous rating) from Ballard was available, and hence chosen for this project. The acceleration periods are of relatively short duration; electric machines can be operated at their peak rating for much longer than those periods where short bursts of power are required. Therefore, the 65kW peak rating of the Ballard induction motor drive is sufficient to meet the acceleration performance requirement. The motor drive has a maximum torque rating of 2453Nm at 253rpm after gear for a gear ratio of 10.6:1, and the powered speed range is from –680rpm to 1360rpm (after gear). It also has regenerative capability of 32.5kW.

2.3.3 RESEARCH SCOPE IN INDUCTION MACHINE CONTROLS

The propulsion motor is an essential component of the powertrain in a hybrid electric vehicle. The method of vector control allows high performance to be achieved from induction motor, but motor parameters used in the controller need to be accurate to achieve good static and dynamic performance of induction motor drive used in HEV/EV propulsion systems. Parameter variation in the motor detunes the controller as well as degrades the motor performance. In this research, parameter variation effects on induction motor performance are studied. In addition, a controller with online parameter estimation and adaptation is developed. The existing parameter estimation and adaptation

techniques, and their shortcomings will be discussed in detail in the next chapter. An advanced controller with a novel on-line parameter estimation and adaptation technique for the induction motor drive will be presented in Chapter V.

The advanced controller research work could not be implemented on the Ballard propulsion motor due to Challenge X competition constraints. Alternative induction machines available were used for the experimental part of the research.

2.4 ELECTRIC POWER TRANSMISSION PATH (EPTP) OPERATION

The operation of the electric power transmission path (EPTP) has been analyzed through simulation using the selected starter-generator and propulsion motor. A vehicle level powertrain systems analysis software built into the Matlab-Simulink platform has been used for the simulations. The software has vehicle dynamics built-in, but the subsystem models are static. The purpose of these simulations is to demonstrate the operation and usage of the EPTP subsystems at different operating modes of the HEV under consideration.

In series operation, the generator provides power to the propulsion motor as well as to the energy storage system depending on its state of charge (SOC). In this case, the control strategy has been set in such a way that the vehicle runs in electric only mode until the state of charge reaches the minimum limit. The vehicle switches to series mode when the SOC reaches the minimum limit. During simulation, a constant speed of 40mph was maintained in the series mode as well as in the electric only mode. A battery-pack has been considered as the energy storage system.

Figure 2.6 shows engine output power, generator output power, propulsion (induction) motor input power, and battery charging/discharging power profiles in series and electric only mode. The simulation shows that the propulsion power comes only from the battery in the electric only mode. In series mode, the generator provides an output power of 20.8kW, out of which 12.7kW goes to the induction motor (propulsion motor), 6.6kW goes to battery-pack for charging, and the remaining 1.5kW goes to the electric accessories. Considering a generator efficiency of 90%, generator input power or engine output power is 23.1kW. The power available at the wheels is about 10.8kW for a cruising velocity of 40mph considering an induction machine efficiency of 85% which is obtained from the load-efficiency curve of the selected induction motor drive system.

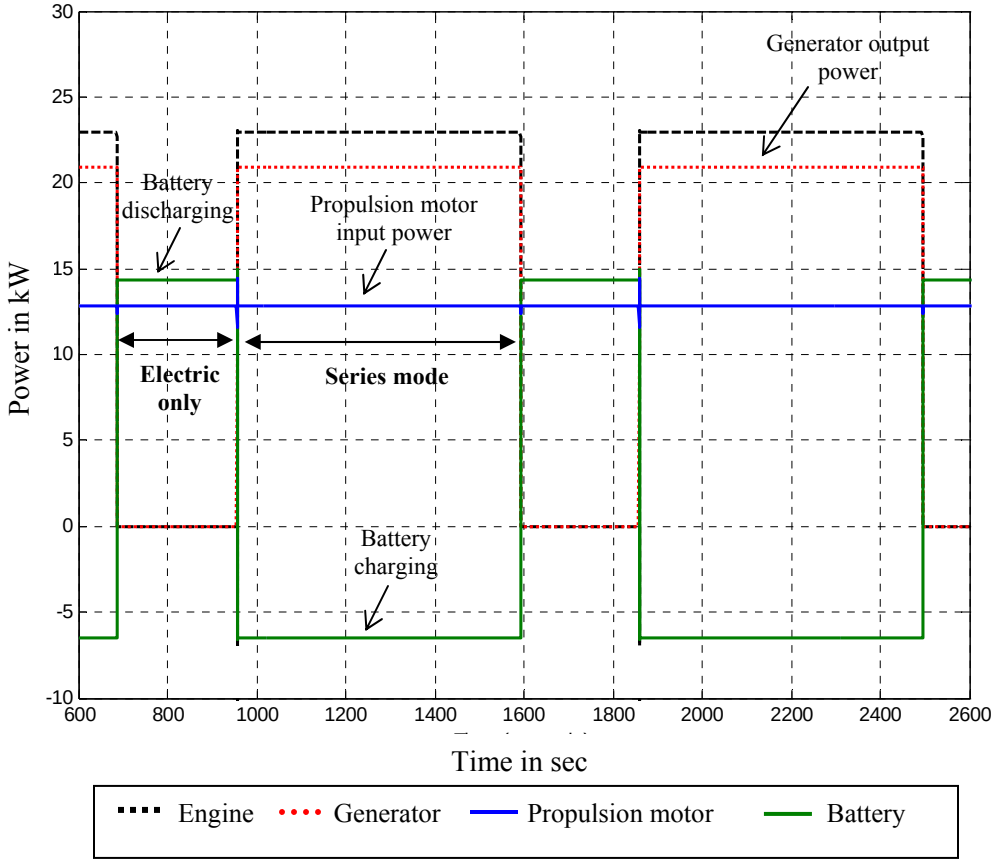


Figure 2.6. Input/output power profiles of different components in series mode.

Figure 2.7 shows engine output power, generator output power, and induction motor input power profiles in the peak parallel mode, where the generator machine delivers propulsion power in the motoring mode. A velocity profile of 0-60mph within 8secs has been applied for parallel mode simulation, which is shown in Figure 2.8. The control strategy was set in such a way that the generator starts in motoring mode when the induction motor power reaches close to the maximum continuous rating of 45kW. The induction motor peak power has been limited to 55kW. From Figure 2.7, it can be observed that at peak road-load demand, the engine supplies propulsion power of 74kW, and the generator machine supplies 19kW in motoring mode. At the same operating point, the induction motor input power is 55kW. The induction motor output power available at the wheels is 48.4kW for the efficiency of 88% obtained from the load-

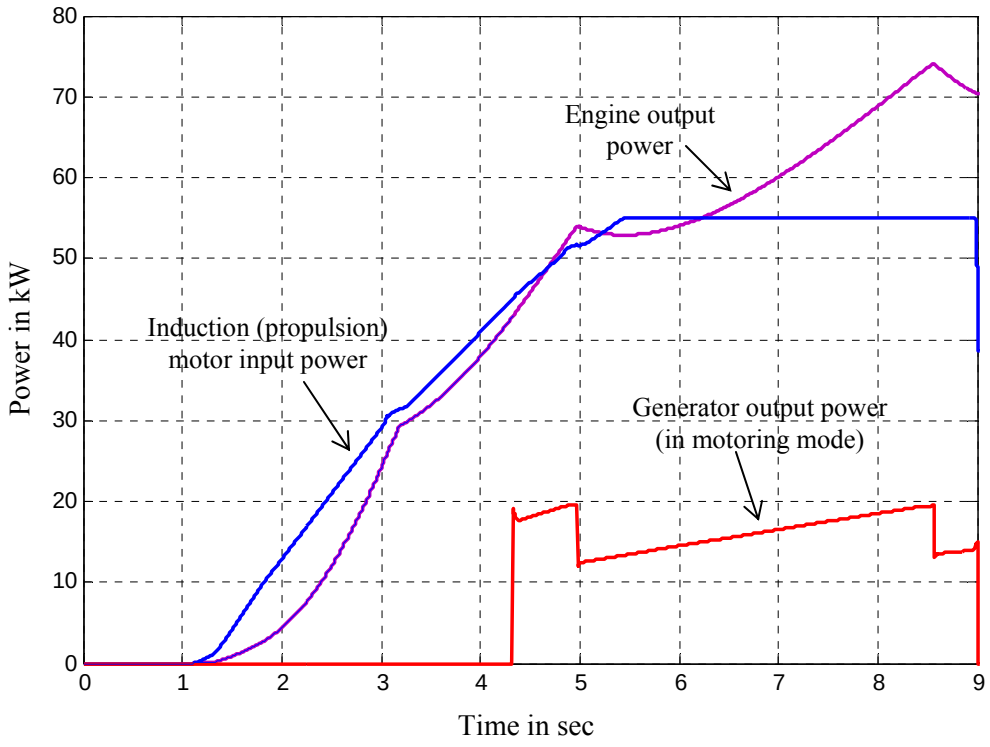


Figure 2.7. Input/output power of different components in parallel mode.

efficiency curve of the drive system. The total power available to the wheel is 141.4kW, which is sufficient to accelerate the vehicle from 0 to 60mph in 8secs.

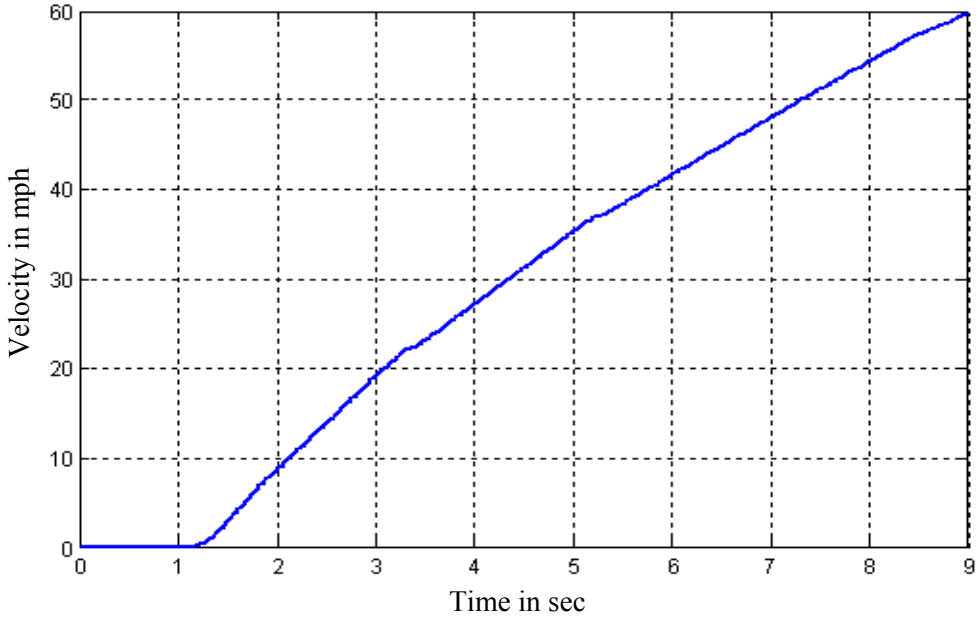


Figure 2.8 Velocity profile in parallel mode operation.

2.5 CONCLUSION

The powertrain component sizing and optimal operating point selection plays a very important role in improving the fuel efficiency and emissions of a hybrid vehicle. The next step is to ensure that the powertrain subsystems deliver the best performance. The development of advanced motor drive subsystems is, therefore, critical in improving the performance of the electric and hybrid electric vehicles. Higher efficiency of the high power starter-generator and propulsion motor drives directly translates to better fuel efficiency and increased range in HEVs and EVs.

The remainder of the dissertation presents research on the starter-generator subsystem and on an advanced controller for induction motor drives. The controller for the starter-generator has been developed to start the engine, generate electrical power for the EPTP and provide mechanical power to the front wheels. The advanced controller for the induction motor focuses on parameter estimation, and is intended to increase the robustness of the induction motor controller; these types of controllers are useful for traction and various other applications.

CHAPTER III

ADVANCED MOTOR DRIVES

3.1 PM BLDC MACHINE DRIVE

The various electrical machines that can be considered for the high power starter-generator application have been discussed in the previous chapter. The PM BLDC machine is an attractive candidate for the application primarily due to its high power and torque density. A suitable machine with ratings close to the required specifications was available, and hence was selected and purchased for the hybrid vehicle project. Modeling, drive structure, and control of the PM BLDC machine will be discussed in the following sections.

3.1.1 PM BLDC MODELING [2, 32]

The PM BLDC machine consists of three stator windings with 120° phase displacement and permanent magnets on the rotor. The stator windings are concentrated instead of sinusoidally distributed, which makes the shape of back-emf trapezoidal. Figure 3.1 shows the stator electric circuit of the PM BLDC machine. The rotor-induced current has been neglected because of high resistive magnets in the models. The stator winding resistance and self-inductance of each phase can be assumed identical, since the stator windings are identical. Further assuming that there is no

reluctance variation in the air-gap with angle, the self-inductances (L_a, L_b and L_c) and mutual inductances (L_{ab}, L_{bc} and L_{ac}) of stator windings can be represented as

$$L_a = L_b = L_c = L \text{, and } L_{ab} = L_{bc} = L_{ac} = M \text{.} \tag{3.1}$$

The voltage balance equation for three phases can be written as

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \tag{3.2}$$

where v_x, e_x and R_x are phase voltage, back-emf and stator resistance, respectively, and p is the operator d/dt. Since the three-phase currents are balanced, they satisfy the condition $i_a + i_b + i_c = 0$. Therefore,

$$Mi_b + Mi_c = -Mi_a \text{.} \tag{3.3}$$

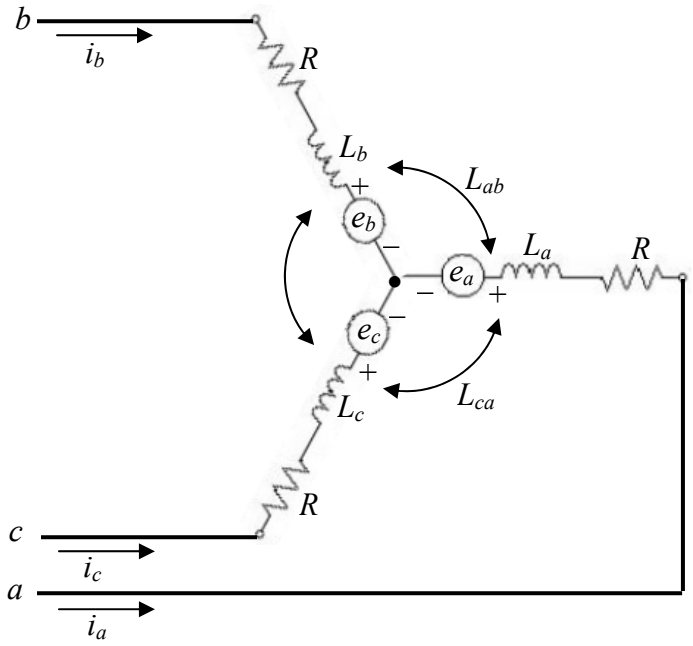


Figure 3.1. Stator electric circuit of a PM BLDC machine.

Hence

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}. \tag{3.4}$$

The rate of change of currents can be expressed in state-space form as

$$p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1/(L-M) & 0 & 0 \\ 0 & 1/(L-M) & 0 \\ 0 & 0 & 1/(L-M) \end{bmatrix} \left(\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \right). \tag{3.5}$$

Since the electrical power transferred to the rotor is equal to the mechanical power available at the shaft neglecting the machine losses, the electromagnetic torque can be represented as

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega_r} \tag{3.6}$$

where ω_r is the rotor speed. Since two phases are active at a given time, the torque equation for equal currents simplifies to

$$T_e = \frac{2 e_{\max} I}{\omega_r} \tag{3.7}$$

where e_{max} is the peak phase back-emf. The speed dynamics can be written as

$$p\omega_r = \left(T_e - T_L\right)/J \tag{3.8}$$

where T_L and J are load torque and moment of inertia, respectively.

Simulation is an essential step before hardware implementation for tuning the controller parameters as well as for analyzing performance characteristics of the machine. The motor model presented by equations 3.4 and 3.5 has been used to simulate the PM BLDC machine selected for the starter-generator of the HEV under consideration.

Equations 3.6, 3.7 and 3.8 have been used to estimate torque and speed from the simulated machine model.

3.1.2 PM BLDC DRIVE STRUCTURE AND CONTROL

The PM BLDC motor drive structure includes the machine, controller, gate driver, power electronics inverter and associated sensors. The controller processes the sensor feedback signals and controls the inverter for the desired operation. Figure 3.2 shows the PM BLDC drive structure with controller that will be used in this research. In this research, the focus will be on controlling the PM BLDC machine in generating and motoring modes. The machine will have to start with a high torque load during engine starting (motoring). To achieve fast torque response and to eliminate the terminal voltage

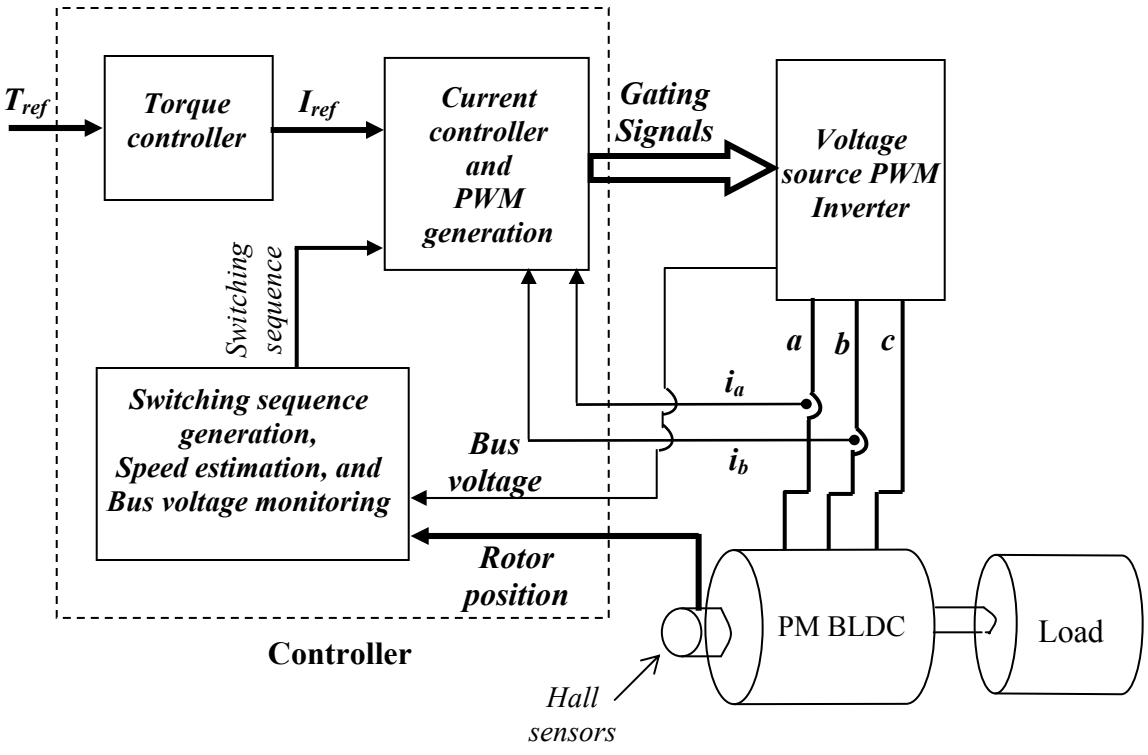


Figure 3.2. PM BLDC drive structure with controller.

regulation problem, current or torque control will be appropriate for this application. The PM BLDC machine basically operates like a DC machine. For three-phase machines, a three-leg six-switch inverter is usually used. A set of three Hall sensors are mounted on stator and placed 120^0 apart to give position information of the rotor. For a three-phase machine, only six discrete rotor positions per electrical revolution are needed to synchronize phase current with phase back-emf for effective torque production. Only two phases conduct current at a given time depending on the rotor position. Figure 3.3 shows the three-phase back-emf waveforms and ideal phase currents. Since, the back-emf waveforms are fixed with respect to rotor position, current supplied to each phase is

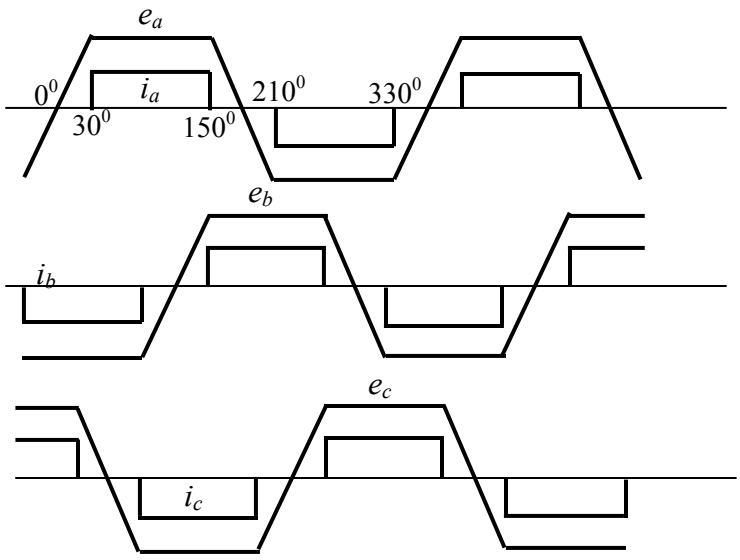


Figure 3.3. Back-emf and current waveforms of PM BLDC machine

synchronized with the back-emf peak of the respective phase. As a result, each switch of the inverter remains active for 120^0 duration in one electrical cycle, and two phases conduct at a time depending on the rotor position. There are six sequences of phase conduction in one electrical cycle with three position sensors. However, the controller

uses one reference value for all three phases. Two current sensors provide the controller with continuous information of phase current to facilitate current control, which in turn allow torque control.

The current controller and PWM generation block in Figure 3.2 uses the error between the reference current and actual current feedback to generate PWM signals for the inverter switches. A simple *PI* controller is often used in the current controller to generate the duty cycles for the PWM signals. The equation used in the torque controller block to generate reference current (I_{ref}) is

$$I_{ref} = \frac{T_{ref}}{K\phi} \tag{3.9}$$

where $K\phi$ is the machine constant, which is measured off-line. Rotor position feedback is used in the switching sequence generation block of the controller to synchronize phase back-emf with phase current, where back-emf is calculated using following equation.

Back-emf $E = K\phi \times \omega_r$. (3.10)

The rotor position and DC bus voltage feedback are also used to monitor the safe limit of speed and bus voltage, respectively.

3.2 INDUCTION MOTOR DRIVE

Different motor drives have been discussed in the previous chapter for the propulsion system of an HEV. Induction motors have excellent response characteristics and very low torque ripple for fast acceleration and smooth propulsion. The DC bus voltage management is much easier during regenerative braking compared to PM motors.

The induction machine is a popular choice for EV/HEV propulsion due to these advantages and also because of its well-developed mature technology. The method of vector control is used in propulsion system EV/HEV induction motor drive to achieve high performance. However, the controller needs to have correct motor parameter information to achieve good static and dynamic performance from the induction motor drive. Induction motor modeling, control, parameter variation effects on the motor drive, and existing parameter estimation techniques will be discussed in the next sections.

3.2.1 INDUCTION MOTOR MODELING

The electromagnetic coupling between the stator and rotor circuit depends on the rotor position; this coupling can be eliminated by referring stator and rotor equations to a common reference frame [1]. For easier algebraic manipulation and simple graphical interpretations, the three-phase or three-axis variables in an ac machine can be transformed to equivalent two-axis variables: quadrature axis (*q*) and direct axis (*d*) variables. Figure 3.4 shows the location of the rotating *d-q* axes relative to the magnetic axes of the stator and rotor. In Figure 3.4, the axes *as, bs, c*s are the stator *abc* reference frame and the axes *ar, br, cr* are the rotor *abc* reference frame. If the common reference frame is non-rotating, it is called a stationary reference frame. Alternatively, the direct (*d*) and quadrature (*q*) axes can be made to rotate with an arbitrary speed, and the reference frame is named in accordance to the chosen variable of transformation. For example, if the arbitrary reference frame is aligned with the direct and quadrature axes of the rotor and is rotating at the rotor speed then it is known as rotor reference frame. Similarly, the arbitrary reference frame can be rotating synchronously. If the *d*-axis of the arbitrary

reference frame is aligned with the rotor flux vector, and the rotor flux velocity is chosen as the speed of rotation of the reference frame then it is known as the rotor flux reference frame [1, 33]. The variables along *a*, *b* and *c* stator axes can be referred to the *d-q* axes as shown in Figure 3.4 by the following expressions [1]

$$f_{qs} = \frac{2}{3}\left[f_{as}\cos\theta + f_{bs}\cos\left(\theta - \frac{2\pi}{3}\right) + f_{cs}\cos\left(\theta + \frac{2\pi}{3}\right)\right] \tag{3.11}$$

$$f_{ds} = \frac{2}{3}\left[f_{as}\sin\theta + f_{bs}\sin\left(\theta - \frac{2\pi}{3}\right) + f_{cs}\sin\left(\theta + \frac{2\pi}{3}\right)\right] \tag{3.12}$$

where the symbol f can represent the voltage, the current or the flux linkage of the three-phase stator circuit.

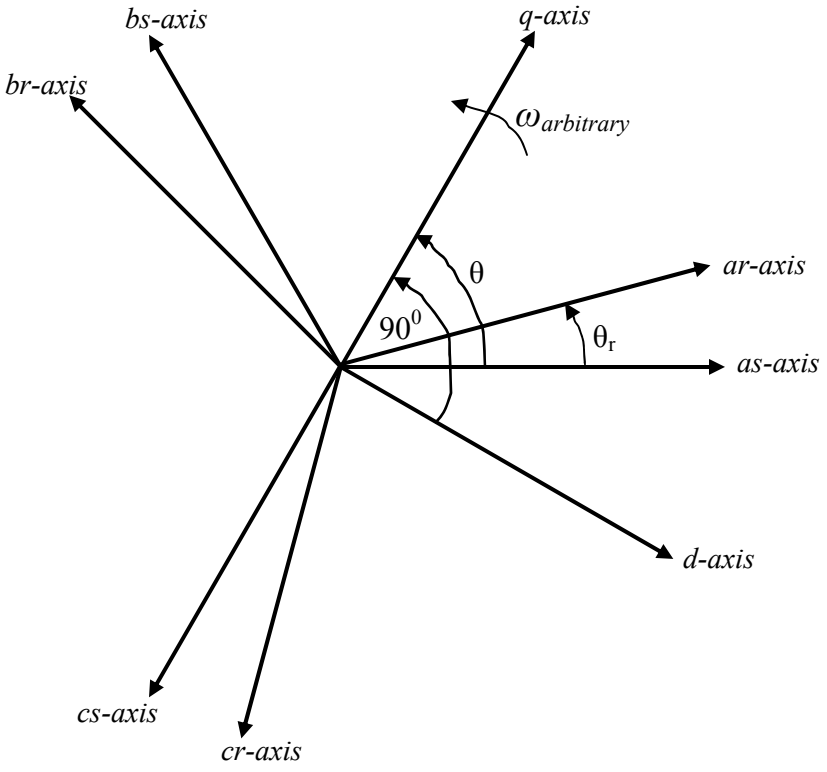


Figure 3.4. Location of rotating *d, q* axes relative to stator and rotor phase axes.

In vector controllers, both the amplitude and phase of the ac excitation are controlled, and the *d-q* axes transformation is employed to make the control simpler. The vector control can be represented in a stationary reference frame as well as in any arbitrary reference frame rotating at a speed of $\omega_{arbitrary}$. By applying the vector control method, highest dynamic and static performance of an induction motor can be achieved, and the machine can be controlled like a separately excited DC machine. Since vector control is the best choice to control induction machines for traction applications, the *d-q* model of the machine will be described in the following. The *d*-axis and *q*-axis equivalent circuit of a three-phase induction motor referred to an arbitrary reference frame rotating at a speed of ω is shown in Figure 3.5. The d-q stator voltage equations referred to the arbitrary reference frame can be represented as

$$v_{ds} = r_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega\lambda_{qs} \tag{3.13}$$

$$v_{qs} = r_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega\lambda_{ds} \tag{3.14}$$

where

$$\lambda_{ds} = L_{ls} i_{ds} + L_m \left(i_{ds} + i'_{dr}\right) \tag{3.15}$$

$$\lambda_{qs} = L_{ls} i_{qs} + L_m \left(i_{qs} + i'_{qr}\right). \tag{3.16}$$

Here λ_{ds} and λ_{qs} are direct and quadrature components of stator flux. r_s, L_m and L_{ls} are stator resistance, mutual inductance and stator self-inductance, respectively.

For the rotor:

$$v'_{dr} = r'_r i'_{dr} + \frac{d\lambda'_{dr}}{dt} - \left(\omega - \omega_r\right)\lambda'_{qr} \tag{3.17}$$

$$v'_{qr} = r'_r i'_{qr} + \frac{d\lambda'_{qr}}{dt} - (\omega - \omega_r)\lambda'_{dr} \tag{3.18}$$

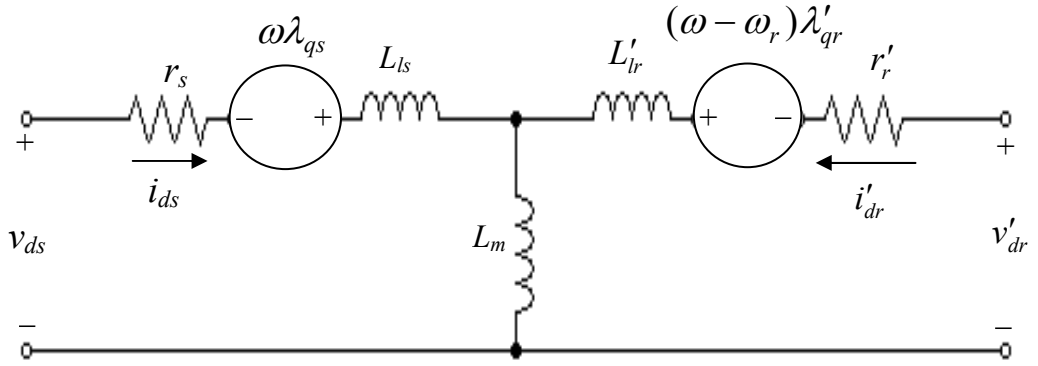
where

$$\lambda'_{dr} = L'_{lr} i'_{dr} + L_m (i_{ds} + i'_{dr}) \tag{3.19}$$

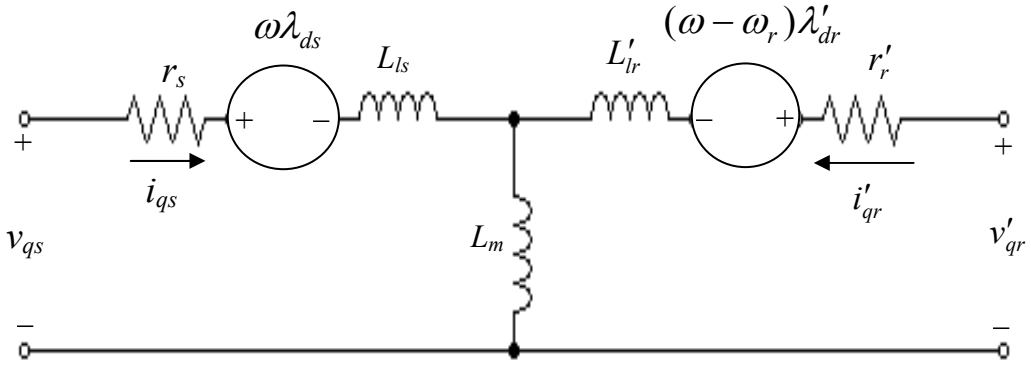
$$\lambda'_{qr} = L'_{lr} i'_{qr} + L_m (i_{qs} + i'_{qr}). \tag{3.20}$$

Here λ'_{dr} and λ'_{qr} are direct and quadrature components of rotor flux. The primed variables are rotor circuit variables referred to the stator. For example,

$$\left(\frac{N_s}{N_r}\right) v_{dr} = v'_{dr} \tag{3.21}$$



d-axis equivalent circuit



q-axis equivalent circuit

Figure 3.5. *d-q* equivalent circuit of a three-phase induction machine.

where N_s and N_r are the number of turns in stator and rotor, respectively. r_r' and L_{lr}' are rotor resistance and rotor self-inductance referred to the stator, respectively. For squirrel cage construction, the rotor voltages v_{dr}' and v_{qr}' are identically zero.

3.2.2 INDUCTION MOTOR DRIVE STRUCTURE AND CONTROL

An induction motor drive includes several components in addition to the machine itself. These include the power electronics inverter, controller and associated sensors. Two current sensors give the phase current feedback to the controller, and a speed encoder provides the rotor angular position. The controller processes the sensor feedbacks and controls the inverter for desired operation. The control algorithm can be torque control or speed control. In this research, the focus will be on speed control with inner-loop current control. The speed control method allows direct speed control as well as indirect torque control. In this case, machine actual speed is compared with the reference speed to generate the reference current, which is compared with the actual current to generate the duty cycles for PWM signals. Figure 3.6 shows an induction motor drive structure with a vector controller.

There are two general methods of vector control. One is called the direct method, for which air gap flux is measured directly. A flux sensor is necessary for the direct method of vector control. Another method is known as indirect method of vector control; this method eliminates the measurement of air gap flux but requires knowledge of the angular position of the rotor. Therefore, a speed sensor is required to estimate the rotor angle.

In this research, the indirect method of vector control is used to control the induction machine. Figure 3.7 [33] explains the indirect control principle with the help of a phasor diagram in the rotor flux reference frame. The superscript *s* is used to denote a stationary set of axes while the superscript *e* denotes electrically rotating axes with the applied voltage. The $d^s - q^s$ axes are fixed on the stator while the $d^e - q^e$ axes rotate at synchronous angular velocity ω_e. The angle θ_e is given by the sum of rotor angular position θ_r and slip angular position θ_{sl}, where $\theta_e = \omega_e t$, $\theta_r = \omega_r t$ and $\theta_{sl} = \omega_{sl} t$. The rotor flux $\hat{\lambda}_r$ consisting of air gap flux and the rotor leakage flux is aligned with the d^e axis. Therefore, for decoupling control, the stator flux component of current i_{ds} and the torque component i_{qs} are to be aligned with the d^e and q^e axes, respectively. The voltage equations for the rotor circuit in the $d^e - q^e$ reference frame are as follows [33, 34, 35].

$$\frac{d\lambda_{qr}}{dt} + R_r i_{qr} + (\omega_e - \omega_r)\lambda_{dr} = 0 \tag{3.22}$$

$$\frac{d\lambda_{dr}}{dt} + R_r i_{dr} - (\omega_e - \omega_r)\lambda_{qr} = 0 \tag{3.23}$$

where

$$\lambda_{qr} = L_r i_{qr} + L_m i_{qs} \tag{3.24}$$

$$\lambda_{dr} = L_r i_{dr} + L_m i_{ds}\,. \tag{3.25}$$

L_r and L_m are rotor inductance and mutual inductance, respectively.

From equations (3.24) and (3.25)

$$i_{qr} = \frac{1}{L_r}\lambda_{qr} - \frac{L_m}{L_r} i_{qs} \tag{3.26}$$

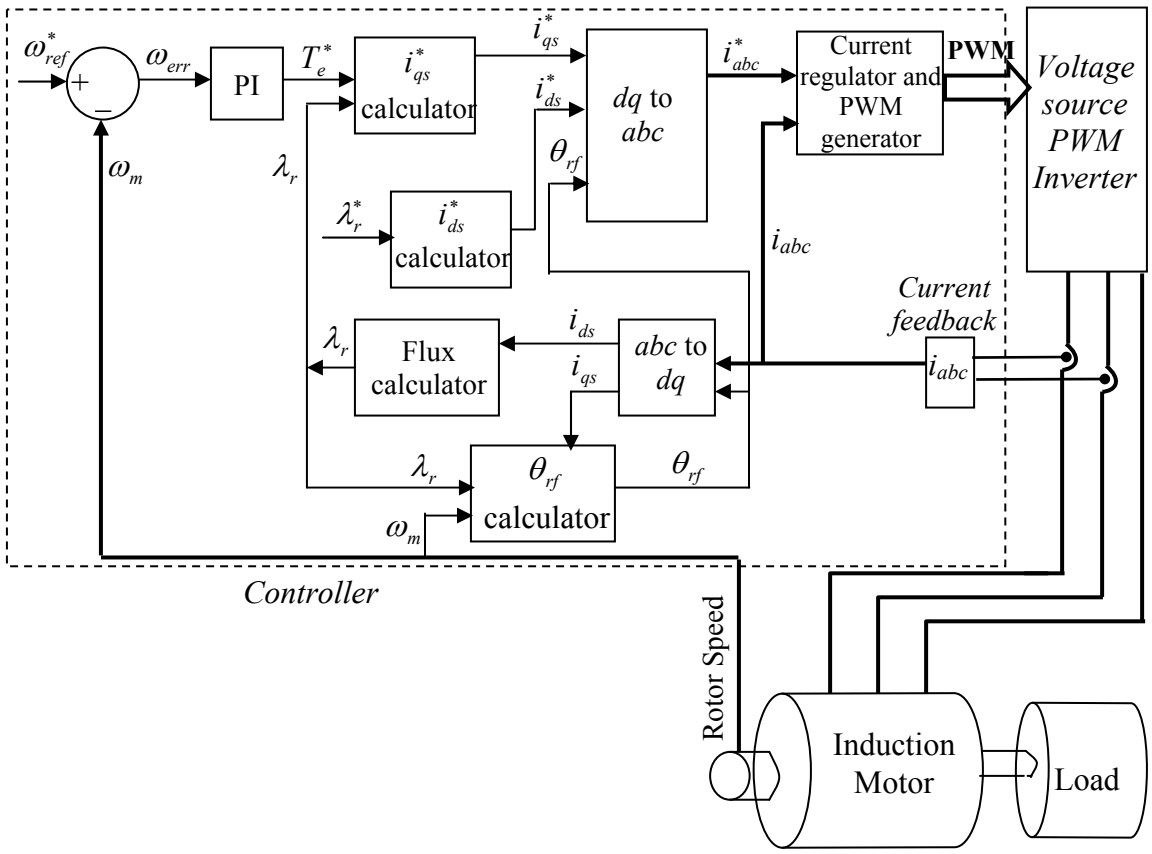


Figure 3.6. Induction motor drive structure with indirect vector controller.

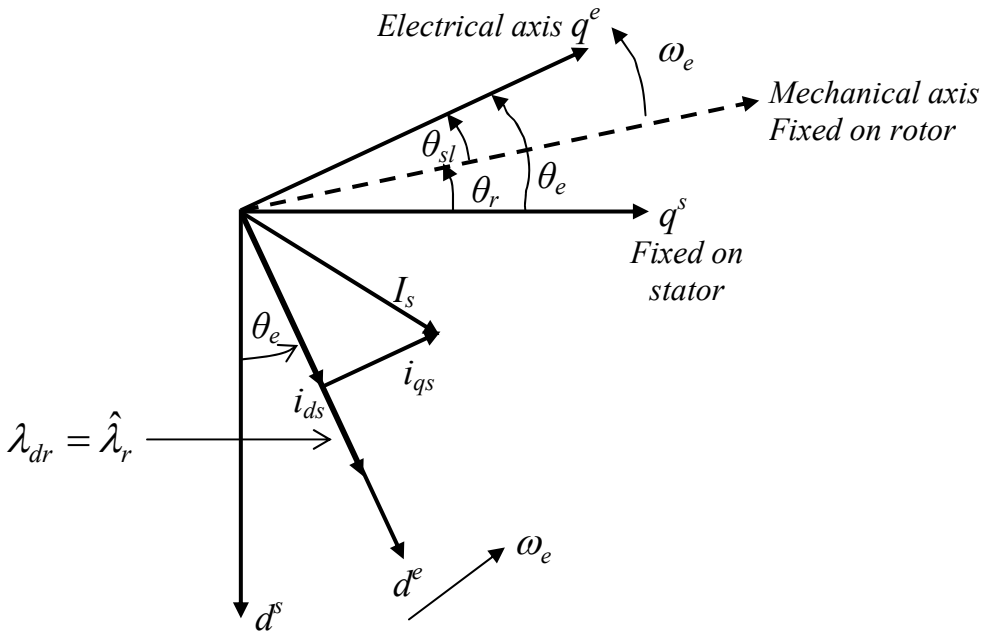


Figure 3.7. Phasor diagram for indirect vector control in rotor flux reference frame [33].

$$i_{dr} = \frac{1}{L_r}\lambda_{dr} - \frac{L_m}{L_r} i_{ds} . \tag{3.27}$$

Using equations (3.22) through (3.27) yields,

$$\frac{d\lambda_{qr}}{dt} + \frac{R_r}{L_r}\lambda_{qr} - \frac{L_m}{L_r} R_r i_{qs} + \omega_{sl}\lambda_{dr} = 0 \tag{3.28}$$

$$\frac{d\lambda_{dr}}{dt} + \frac{R_r}{L_r}\lambda_{dr} - \frac{L_m}{L_r} R_r i_{ds} - \omega_{sl}\lambda_{qr} = 0 \tag{3.29}$$

where $\omega_{sl} = \omega_e - \omega_r$. In the indirect vector control method, slip speed is used to decouple the torque and flux components of stator current as well as to obtain the instantaneous rotor flux position. For decoupling control, it is required that in the rotor flux reference frame $\lambda_{qr} = \frac{d\lambda_{qr}}{dt} = 0,\ \ \lambda_{dr} = \hat{\lambda}_r$ and in steady state $\frac{d\lambda_{dr}}{dt} = 0.$

Substituting the first two conditions, equations (3.28) and (3.29) can be simplified as

$$\omega_{sl} = \frac{L_m}{\hat{\lambda}_r}\left(\frac{R_r}{L_r}\right) i_{qs} \tag{3.30}$$

which is known as the slip relation, and

$$\frac{L_r}{R_r}\frac{d\hat{\lambda}_r}{dt} + \hat{\lambda}_r = L_m i_{ds} . \tag{3.31}$$

In steady state $\hat{\lambda}_r = L_m i_{ds}$. The instantaneous rotor flux position is given as

$$\theta_{rf} = \int (\omega_r + \omega_{sl}) dt . \tag{3.32}$$

The torque equation as a function of stator currents and stator flux is

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

where *P* is the number of poles and the stator flux components are

$$\lambda_{qs} = L_m i_{qr} + L_s i_{qs} \tag{3.34}$$

$$\lambda_{ds} = L_m i_{dr} + L_s i_{ds}\,. \tag{3.35}$$

Using equations (3.24), (3.25), (3.34) and (3.35), the stator *d-q* fluxes can be expressed as

$$\lambda_{qs} = \left(L_s - \frac{L_m^2}{L_r} \right) i_{qs} + \frac{L_m}{L_r} \lambda_{qr} \tag{3.36}$$

$$\lambda_{ds} = \left(L_s - \frac{L_m^2}{L_r} \right) i_{ds} + \frac{L_m}{L_r} \lambda_{dr}\,. \tag{3.37}$$

Eliminating the stator flux, the torque expression becomes

$$T_e = \frac{3}{2}\left(\frac{P}{2} \right) \frac{L_m}{L_r} \left(i_{qs} \lambda_{dr} - i_{ds} \lambda_{qr} \right). \tag{3.38}$$

Substituting $\lambda_{qr} = 0$ for the rotor flux reference frame and $\lambda_{dr} = \hat{\lambda}_r^{rf}$, the torque expression is

$$T_e = \frac{3}{2}\left(\frac{P}{2} \right) \frac{L_m}{L_r} i_{qs}^{rf} \hat{\lambda}_r^{rf} \tag{3.39}$$

where the superscript *'rf'* denotes the variables in the rotor flux reference frame.

The motor dynamics can be expressed in terms of the electromagnetic torque T_e and the load torque T_L as

$$\left(\frac{2}{P} \right) J \frac{d\omega_r}{dt} = T_e - T_L \tag{3.40}$$

where J is the moment of inertia.

The developed torque T_e can be controlled by varying only i_{qs}^{rf}, without affecting the flux component $\hat{\lambda}_r^{rf}$. Since the torque component and flux component are decoupled, the controller provides faster response according to the desired command.

The importance of correct machine parameters in the controller can be evaluated from reviewing the controller equations. The rotor flux is calculated in the flux calculator block of Figure 3.6 using equation 3.31. Equations 3.30 and 3.32 are used to calculate the slip speed and instantaneous rotor flux position in θ_{rf} calculator block. The reference *q*-axis stator current proportional to the torque component of stator current is calculated in i_q^* calculator block using the following equation:

$$i_{qs}^* = \frac{2}{3}.\frac{2}{P}.\frac{L_r}{L_m}.\frac{T_e^*}{\hat{\lambda}_r} \tag{3.41}$$

where a correct $\hat{\lambda}_r$ calculation depends on the correct value of rotor resistance.

The controller uses equations 3.30, 3.31, 3.32 and 3.41 to calculate the torque and flux command in terms of the *q*- and *d*-axes decoupled current commands i_{qs}^* and i_{ds}^*. These calculations depend on the machine parameters of rotor resistance R_r, rotor inductances L_r and mutual inductance L_m. The parameter that varies significantly during machine operation due to heating is the rotor resistance R_r. The rotor self inductance L_{lr} that contribute to L_r and mutual inductance L_m are unaffected by temperature. These inductances change primarily with saturation. Tables of inductances as a function of current are typically used in the controller to take saturation into account. However, the rotor resistance variation cannot be related to any variable in the machine model since it depends on temperature. Therefore, the rotor resistance is considered as the dominant parameter that can cause controller performance degradation due to the use of its incorrect value. The operation of the controller with an incorrect value of machine

parameter is known as detuned operation. In the following section, the parameter variation effects will be addressed in further detail.

3.2.3 PARAMETER VARIATION EFFECTS

In the indirect field oriented vector control method, the slip relation presented in equation 3.30 is employed to obtain the correct subdivisions of the stator current into the torque and flux components. Again, slip calculation depends on the rotor time constant L_r/R_r. Therefore, the rotor resistance used in the controller needs to be accurate. The rotor resistance can change because of motor heating, skin effect or other non-idealities. Temperature has significant effect on rotor resistance as well as stator resistance. Rotor resistance can change from ambient temperature values to hot (up to maximum operating temperature) values by about 20% to 40% [36, 37]. For high power induction motors with class F conductor insulation, the operating temperature can rise up to 115^0C [37]. The temperature dependence of conductor resistance is given as

$$R_T = R_0\left(1 + \alpha\left(T - T_0\right)\right) \tag{3.42}$$

where R_0 is the resistance of the conductor measured at the room (ambient) temperature T_0, and α is the temperature coefficient of the conductor material. For aluminum conductor $\alpha = 0.004308/\,°\text{C}$ and for copper conductor $\alpha = 0.004041/\,°\text{C}$, both measured at 20°C [38]. For an aluminum conductor, temperature rise from 20°C to 115°C leads to a change of 40.93% in its resistance value.

If the rotor open-circuit time constant L_r/R_r is not correctly known, or if it changes, the subdivision of the stator current in the *d-q* axis will not be correctly attained. The

result will be detuning of the controller due to the loss of correct field orientation. The detuning causes incorrect calculations of rotor field angle and stator current components. The effects of detuning are: (1) the flux level is not properly maintained, (2) the resulting steady-state torque is not the command value, (3) the torque response is not instantaneous, and (4) more stator current is drawn for the same load. The steady state effect of detuning is easily visualized from the rotor loop equation in the conventional steady state per-phase equivalent circuit [1] shown in Figure 3.8. From the equivalent circuit

$$j\omega_e L_m I_s + \left[\frac{R_r}{s} + j\omega_e\left(L_m + L_{lr}\right)\right] I_r = 0\,. \tag{3.43}$$

This equation can be rearranged as

$$I_r = \frac{-j\omega_s L_m I_s}{R_r + j\omega_s L_r} = \frac{-j\omega_s L_m I_s}{Z_r \angle \varphi_r} \tag{3.44}$$

where s is the slip, $\omega_s = s\omega_e$ is the slip speed and $L_r = L_m + L_{lr}$. Figures 3.9 and 3.10 show the indirect field orientation with correct value of slip frequency and with incorrect value of slip frequency, respectively [1]. In Figure 3.9, the slip calculator has the correct value of $L_r / R_{r\_initial}$. I_{qs} and I_{ds} establish I_{qds} which in turn sets the phasor $-j\omega_s L_m I_{qds}$.

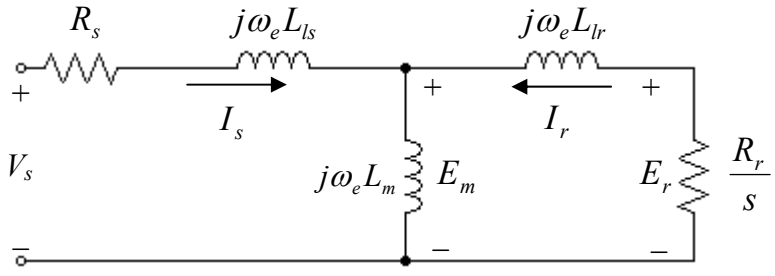


Figure 3.8. Steady state per phase equivalent circuit of an induction motor.

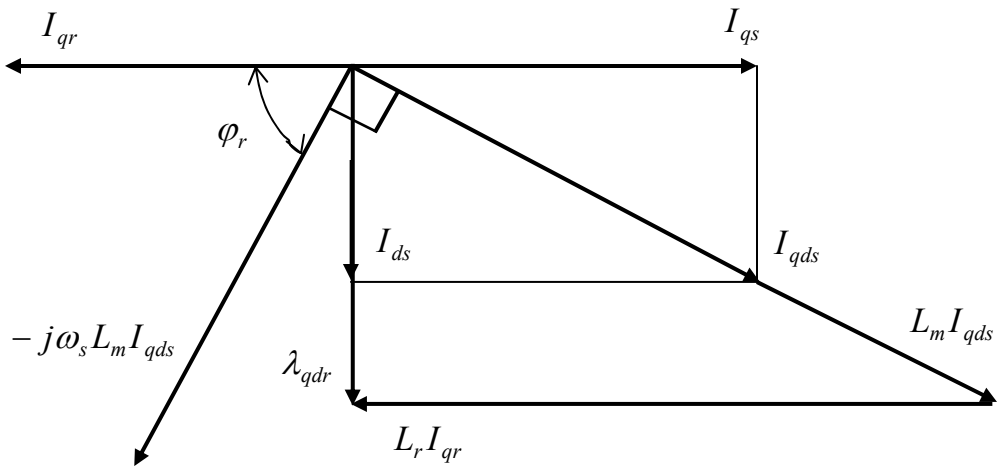


Figure 3.9. Indirect field orientation with correct slip frequency.

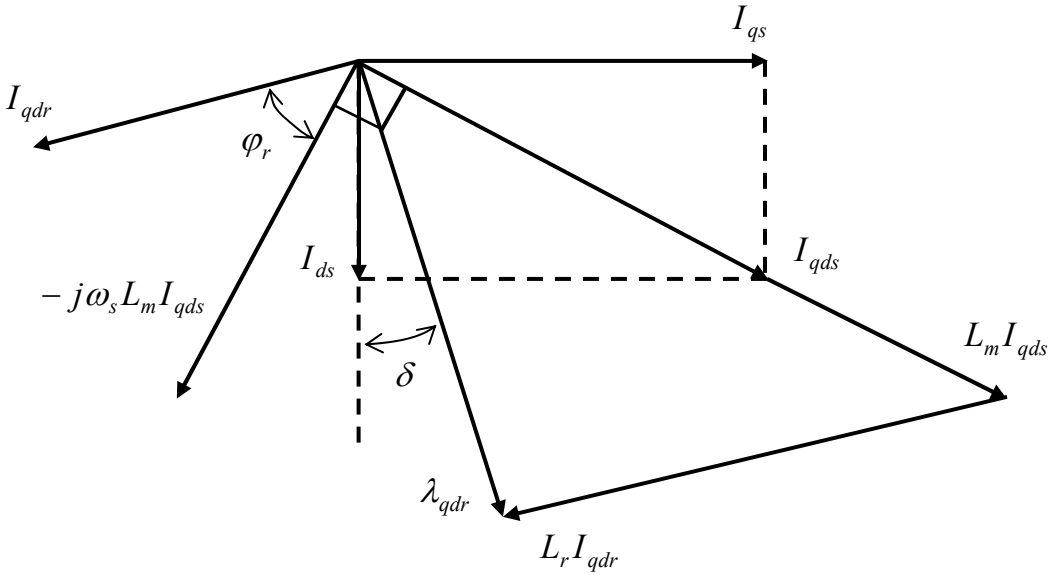


Figure 3.10. Indirect field orientation with incorrect slip frequency.

Correct slip frequency provides correct phase angle of rotor self-impedance $\varphi_r = \tan^{-1}\frac{\omega_s L_r}{R_r}$, which puts I_{qr} 180° phase shifted from I_{qs}. The rotor flux is orthogonal to the rotor current and aligned with I_{ds}. Figure 3.10 illustrates the situation when slip frequency is incorrect ($L_r/R_{r\_initial} > L_r/R_{r\_changed}$). In this case, phasor

$-j\omega_s L_m I_{qds}$ has the same position as in Figure 3.9, but is smaller in magnitude. The angle φ_r is also smaller and the rotor current does not lie on the negative q axis in opposition to I_{qs}. As a result the rotor flux λ_{dqr} is not aligned with the actual machine d axis; instead, it is shifted by an angle δ. In the controller, the measured motor currents are transformed into d and q-axes components to calculate the rotor flux using the estimated rotor flux angle θ_{rf} (Figure 3.6); this rotor flux estimator functions as an open loop observer in parallel to the motor dynamics. The controller d and q- axes currents will be different from the actual motor d and q- axes currents when the angle δ is non-zero. The magnitude of the rotor flux will also be incorrect. The q-axis current component in the machine may be larger or smaller than the controller q-axis current depending on the sign of δ. This is because the actual machine rotor flux and rotor flux angle is different from what has been estimated in the controller. As a result, steady state torque will not be the commanded torque; in addition, the torque response will not be instantaneous.

The effect of detuning varies with the operating region of the induction motor drive. A small value of the ratio $\frac{i_{qs}^{rf}}{i_{ds}^{rf}}$, which is in the slip relation, results in a low sensitivity to detuning [1]. Therefore, the detuning effects are much more significant in the field weakening region, where the d-axis component of the stator current is smaller. Drives in propulsion systems operate over a wide range of speeds, and the detuning effects in the field weakening region at higher speeds are significant. Machine saturation also plays an interesting role in a detuned controller. The overall influence of saturation is to make the effect of detuning less sensitive with respect to both flux and torque

variations. The effects of detuning suddenly become prominent when the machine enters the field weakening mode and the machine core becomes desaturated.

In speed or position controlled drives, the presence of the outer loop controller often compensates for the less responsive detuned behavior [39]. The effects of detuning is unnoticeable when the motor is partially loaded, since the outer loop controller will generate the required q-axis current reference to produce the torque even though the flux may be somewhat increased or reduced. There will be loss of torque only when the machine is fully loaded due to current limits. The transient effects of detuning are less severe than the steady state power loss increase in these drives. However, in torque controlled drives, such as those used in propulsion systems, the loss of instantaneous torque control due to detuning is a much more severe problem.

This problem can be overcome by tracking and adapting the rotor resistance within the controller in real time so that controller has the correct value of the slip frequency. Moreover, the stator or rotor resistance change is directly related to stator or rotor temperature. An estimation of the resistances will indicate the thermal condition of machine; this information could be utilized in the cooling subsystem of the machine to regulate the flow of coolant. In this research, an observer based on-line parameter estimation and adaptation will be addressed.

3.2.4 REVIEW OF EXISTING PARAMETER ESTIMATION AND ON-LINE ADAPTATION TECHNIQUES

There are several types of state variable estimation algorithms used to identify the parameters of an induction machine. Observer based estimation is one such method. The

observer runs in parallel to a dynamic system to estimate the state variables of the plant [40]. There are two forms of observers or estimators: open loop and closed-loop. In a closed-loop estimator, a correction term, involving estimation error, is used to adjust the response of the estimator [40]. A closed-loop estimator is typically referred to as an observer. In an open loop estimator, there is no correction term or state variable feedback.

Observers used for rotor and stator resistance estimation usually estimate the states (especially stator current and rotor flux) of the induction motor that are involved in resistance estimation. The most commonly used observers are Luenberger and Kalman types. The Luenberger observer (LO) is of deterministic type, and the Kalman filter (KF) is of stochastic type [40]. The Luenberger observer, which estimates stator current and rotor flux, can be expressed as [41, 42]

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{V}_s + \boldsymbol{L}\tilde{\boldsymbol{i}}_s, \quad \tilde{\boldsymbol{i}}_s = \boldsymbol{C}\hat{\boldsymbol{x}} \tag{3.45}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -\frac{1}{\sigma}\left(\frac{1}{\tau_s} + \frac{1-\sigma}{\tau_r}\right)\boldsymbol{I} & \frac{L_m}{\sigma L_s L_r}\left(\frac{\boldsymbol{I}}{\tau_r} - \omega_r \boldsymbol{J}\right) \\ \frac{L_m \boldsymbol{I}}{\tau_r} & -\frac{\boldsymbol{I}}{\tau_r} + \omega_r \boldsymbol{J} \end{bmatrix}, \; B = \begin{bmatrix} \frac{\boldsymbol{I}}{\sigma L_s} & 0 \end{bmatrix}^T, \; C = \begin{bmatrix} \boldsymbol{I} & 0 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{i}_s & \boldsymbol{\lambda}_r \end{bmatrix}^T, \; \boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; \boldsymbol{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \; \sigma = 1 - \frac{L_m^2}{L_s L_r}, \; \tau_s = \frac{L_s}{R_s}, \; \tau_r = \frac{L_r}{R_r}, \; \tilde{\boldsymbol{i}}_s = \boldsymbol{i}_s - \hat{\boldsymbol{i}}_s .$$

Here $\wedge$ denotes estimated quantities, and $\boldsymbol{L}$ is the Luenberger gain matrix, which is determined by the motor model. Therefore, it is necessary to know the accurate motor model. Since the Luenberger observer is predetermined, it cannot work properly in the event of relatively great disturbance in the measurement, great parameter differences, or

internal noise in the system. In [41, 42], error between actual stator current and estimated stator current and rotor flux have been used to estimate the stator and rotor resistances. The equations used for the stator and rotor time constant estimation are

$$\frac{d\hat{R}_S}{dt} = -\lambda_1 \left(e_{ids} \hat{i}_{ds} + e_{iqs} \hat{i}_{qs} \right) \tag{3.46}$$

$$\frac{d}{dt}\left(\frac{\hat{R}_r}{L_r} \right) = \frac{\lambda_2}{L_r \left[e_{ids} \left(\hat{\lambda}_{dr} - L_m \hat{i}_{ds} \right) + e_{iqs} \left(\hat{\lambda}_{qr} - L_m \hat{i}_{qs} \right) \right]} \tag{3.47}$$

where λ_1 and λ_2 are arbitrary positive gain, and $e_{ids} = i_{ds} - \hat{i}_{ds}$, $e_{iqs} = i_{qs} - \hat{i}_{qs}$. Equation 3.47 suffers from singularity problem (division by zero) especially when errors go to zero. In [43], equation (3.46) is used for stator resistance estimation but in rotor resistance estimation a ratio factor is used which is given by

$$\hat{R}_r = R_{sm} \hat{R}_s \tag{3.48}$$

where R_{sm} is the ratio of the nominal values of the stator and rotor resistance. Rotor resistance estimation is dependent on stator resistance, but the change in stator and rotor resistance might not be correlated. In [44, 45, 46], an extended Luenberger observer and an extended Kalman filter are used to estimate the rotor flux as well as the motor parameters; these estimators are robust in the event of large disturbance in measurement, internal noise and parameter variation. The basic Kalman filter is only applicable to linear system, and for non-linear systems an extended Kalman filter is used. The extended Kalman filter is given as [40]

$$\frac{d\hat{\boldsymbol{x}}}{dt} = \boldsymbol{A}(\hat{\boldsymbol{x}})\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{V}_s + \boldsymbol{K}(\boldsymbol{i}_s - \hat{\boldsymbol{i}}_s) \tag{3.49}$$

where

$$\boldsymbol{A}=\begin{bmatrix} -1/\tau_s^* & 0 & L_m/(L_sL_r\tau_r) & \omega_r L_m/(L_sL_r) & 0 \\ 0 & -1/\tau_s^* & -\omega_r L_m/(L_sL_r) & L_m/(L_sL_r\tau_r) & 0 \\ L_m/\tau_r & 0 & -1/\tau_r & -\omega_r & 0 \\ 0 & L_m/\tau_r & \omega_r & -1/\tau_r & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix} 1/L_s & 0 \\ 0 & 1/L_s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{C}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},\ \boldsymbol{x}=\begin{bmatrix} i_{ds} & i_{qs} & \lambda_{dr} & \lambda_{qr} & \omega_r \end{bmatrix}^T,\ \frac{1}{\tau_s^*}=\frac{R_s+R_r(L_m/L_r)^2}{L_s}.$$

Here $\boldsymbol{K}$ is the Kalman gain matrix. Extended Kalman filters and extended Luenberger observers use rotor speed as a state variable, and are typically higher order observers. Moreover, gain matrices require intensive calculations. As a result, these methods are complex, and need significant computational time, which is a burden for a processor. Another difficulty of these methods is to obtain an imposed dynamic response in the presence of errors in estimated quantities [47]. In [48, 49], a ninth-order estimation algorithm and a least-square technique have been presented which are complex and computationally intensive. Moreover, the least-square method presented in [49] needs stator voltages, stator currents and their derivatives as inputs. In [47], a parameter estimation approach has been presented based on adequate modeling of induction motor. In this paper, a full-order observer is used which is given by

$$\dot{\hat{\boldsymbol{x}}}=\boldsymbol{A}\hat{\boldsymbol{x}}+\boldsymbol{B}\boldsymbol{V}_s+\boldsymbol{K}\boldsymbol{C}(\boldsymbol{x}-\hat{\boldsymbol{x}}) \tag{3.50}$$

where

$$\boldsymbol{A}=\begin{bmatrix} \dfrac{-\alpha_s-\alpha_r}{1-k_sk_r}+jp\omega_r & \dfrac{(\alpha_r-jp\omega_r)}{1-k_sk_r} \\ -\alpha_s & 0 \end{bmatrix}=[a_{ik}],\ \boldsymbol{B}=\begin{bmatrix} \dfrac{1}{L_t(1-k_sk_r)} & \dfrac{1}{L_t} \end{bmatrix}^T=[b_i]$$

$$\alpha_s = \frac{r_s}{l_s + L_m}, \qquad k_s = \frac{L_m}{l_s + L_m}, \qquad \boldsymbol{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad k_1 = \hat{a}_{11} - \lambda_1 - \lambda_2$$

$$k_2 = \hat{a}_{21} + \frac{\lambda_1 \lambda_2}{\hat{a}_{12}}, \ \boldsymbol{x} = \begin{bmatrix} i_s & i_{ms} \end{bmatrix}^T = \begin{bmatrix} x_i \end{bmatrix}, \ i_{ms} = i_s + k_s i_r .$$

Here λ_1 and λ_2 are complex Eigen values of the error matrix, and i_{ms} is the magnetizing current corresponding to the stator flux. The equation used for parameter estimation is

$$\hat{p}_{h,n+1} = \hat{p}_{h,n} + \Delta p_{h,n}$$

where p_h is the *h*-th motor parameter, and

$$\Delta a_{ij} \cong \sum_{h=1}^{n_p} \frac{\delta \hat{a}_{ij}}{\delta p_h} \Delta p_h + \frac{\delta \hat{a}_{ij}}{\delta \omega_r} \Delta \omega_r \ , \ \Delta b_i \cong \sum_{h=1}^{n_p} \frac{\delta \hat{b}_i}{\delta p_h} \Delta p_h \ . \tag{3.51}$$

The observer used in [47] needs an accurate motor model, and parameter estimation algorithm requires matrix error as well as parameter error, which is a computational burden for real time implementation. Moreover, rotor resistance estimation depends on correct value of stator resistance, and the observer does not have the characteristics to be robust in the event of disturbance in measurement and internal noise.

In [50], an adaptive sliding mode observer is used to estimate rotor flux and stator current. Two observers are used for estimating stator current and one observer is used for rotor flux estimation whereas, observers are dependent on each other. It is tedious and time consuming to determine the sliding mode gain. Moreover, using several observers require large memory and extensive processor time. In [51], artificial neural network is used for on-line resistance estimation. Implementation of neural networks requires large memory and high-speed processor, and the weights between the neurons need to be

trained for every new environment. A genetic algorithm is presented in [52] for parameter estimation, which uses complex cubic polynomial. The algorithm can only be used in off-line estimation. In [53], rotor resistance is estimated based on motor parameters, state variables and stator voltage. The equation presented for rotor resistance estimation is

$$R_r = (\omega_s - \omega_r)L_r \sqrt{\frac{V_{sq} - \omega_s \sigma L_s i_{sd}^s}{\omega_s i_{sd}^s \left(\sigma L_s + \frac{L_m^2}{L_r} \right) - V_{sq}^s}} \quad . \tag{3.52}$$

The rotor resistance estimation depends on correct values of other motor parameters, and the estimation algorithm does not have the characteristics to be robust against disturbance and noise. In [54], short duration of pulses has been added to the constant flux reference current for rotor resistance estimation but pulse injection creates ripple in output torque. Another approach based on adequate modeling is model reference adaptive systems (MRAS) to estimate motor variables or parameters. In MRAS methods [40, 55, 56], some state variables of induction motor are estimated in a reference model and then compared with the state variables estimated by using an adaptive model. The output errors are used to drive suitable adaptation mechanism to generate desired estimated variables or parameters. MRAS gives accurate convergence with accurate model but the convergence is slower [57]. Moreover, these methods require integrators and the performances are limited by internal noise or great parameter variance.

Existing on-line parameter estimation and adaptation methods are complex, and need extensive computations and large memories. Some of them are highly sensitive to the internal noise, and cannot provide satisfactory performance in the event of large

parameter variations. Moreover, some of them have slow convergence towards the actual value of parameter.

3.3 CONCLUSION

The modeling, control, and drive structure of the PM BLDC machine and induction motor that have been selected for two subsystems of a series-parallel 2x2 hybrid electric vehicle implementation have been discussed. The significant effects in dynamic and static performance of induction motor due to parameter variation, which is being used in the HEV as the propulsion motor, have been addressed. The existing technologies to compensate the parameter variation effects on induction motor have also been studied.

There exist a lot of research opportunities in the design, control and efficient use of motor drives in the electric power transmission path of an HEV. As mentioned previously, the research presented in this dissertation is focused on the design and implementation of advanced controllers of motor drives for the HEV electric power transmission path. The design and implementation of an advanced controller for the starter-generator of the HEV system will be presented in the next chapter. Subsequently, the focus will shift to advanced controller development for the propulsion motor. Parameter variation and the estimation in real time for electric motor drives is still a critical issue for high performance drives such as that required in HEVs or EVs. A novel technique to compensate for the parameter variation in induction motor drive intended for the propulsion system of an HEV or EV has been developed through this research.

CHAPTER IV

PM BLDC STARTER-GENERATOR

Although various types of electric motor drives exist for use in HEVs, there are still opportunities for research to develop more efficient ones with better performance for the electrical subsystems. A starter-generator with start-stop features is an essential part of a hybrid electric vehicle. The hybrid electric vehicle under consideration requires a starter-generator that has high power density, high efficiency, wide speed operation range, robust controller and compact size to be accommodated inside the engine compartment while satisfying the generation and engine starting requirements.

4.1 STARTER-GENERATOR OPERATION

A starter-generator is an essential component of the electric power transmission path of an HEV that starts the engine, generates power for the DC bus and provides boost torque during peak propulsion demand. A conventional starter and alternator cannot meet the power generation requirement, and contributes to significant engine torque ripple. A high power starter-generator in an HEV can start the engine very quickly, thereby reducing tailpipe emissions and also generate sufficient power to supply the electric powertrain for series mode operation in city driving conditions. The PM BLDC exhibits high efficiency with a maximum speed to base speed ratio of 2-2.5. Moreover, control is

as simple as a DC machine control with an electronic commutator, and does not require vector control like an induction machine or a permanent magnet synchronous machine. Control can be easily switched from motoring to generating and vice versa. The controller for the PM BLDC to be developed has three modes of operation: generating, engine starting and motoring for torque boosting. The operating modes are described in the following.

4.1.1 GENERATION

The PM BLDC starter-generator is used as a generator for charging the energy storage system during cruising and stop-time in urban/highway driving depending on the state of charge of the subsystem. The generator also provides power to the propulsion motor in series mode during cruising in urban driving. The engine and the starter-generator will be cycled on and off with the operating point of both components chosen to maximize their respective efficiencies. The starter-generator is directly coupled to the engine through a gear. The maximum power and torque of the engine (Volkswagen 1.9L TDI) chosen for this HEV project are 76KW (102hp) and 239Nm, respectively, and the operating speed range is between 1,000 rpm and 4,000 rpm. However, through a series of dynamometer testing, the most efficient operating speed with high load torque has been found to be around 1700 rpm. The selected gear ratio of 2.8:1 enables the PM BLDC to operate near its full load around the corner speed of 4,750 rpm for the optimal engine speed. Depending on the state of charge, the energy storage will be charged with a certain current command.

4.1.2 ENGINE STARTING

Starting the engine with the high power PM BLDC has lot of benefits in addition to the elimination of an additional starter. The cranking time, engine ripple and current drawn from source are important factors in engine starting. The main objectives during starting are starting the engine quickly and reducing the engine torque ripple. The diesel engines with sizes up to 2L from different manufacturers do not require starting torque more than 100Nm at a speed of 800rpm to 1200rpm [8]. The Volkswagen 1.9L TDI diesel engine requires starting torque of 83Nm to 92Nm at 800rmp to 1000rpm [58]. The selected PM BLDC machine is capable of providing 37.5Nm at the starting, and the low rotor inertia helps to reach a speed of 2800rpm within the desired time. With a gear ratio of 2.8:1, the torque and speed reflected at the engine shaft are 105Nm and 1000 rpm, respectively. Therefore, the selected PM BLDC machine has sufficient starting torque and power to start the engine. The fast response time has been verified through simulation, which will be presented later.

4.1.3 MOTORING

The PM BLDC will also be used in motoring mode for supplying additional torque to the wheels during peak acceleration demand. The PM starter-generator goes to high speeds in this mode, since the engine is powering the wheels through the transmission. The vehicle speed and transmission shifter position determines the PM starter-generator speed. The controller can accommodate the speed changes since the PM starter-generator electrical time constant is much smaller than the mechanical time constant.

The controller has been designed to operate in the torque controlled mode during motoring similar to that in the generating mode. The controller is capable to accommodate an input DC bus voltage swing between 200V and 400V.

4.2 CONTROLLER OF PM BLDC DRIVE

The PM BLDC drive power electronics, sensors, and controller have been configured for two-quadrant torque-controlled operation to provide engine starting, power generation, and motoring assist. The controller has been designed with a feedback current control algorithm in the inner loop to meet the starting, generating and motoring modes of operation of the PM BLDC. The current controller also eliminates compensation requirements for speed and load variations. In the generating mode, controlled power generation starts above a minimum speed, since the back-emf voltage needs to be less than the DC bus voltage. The charging of the energy storage system is stopped when the dc bus voltage reaches the upper limit. The speed and dc bus voltage are also monitored in the controller.

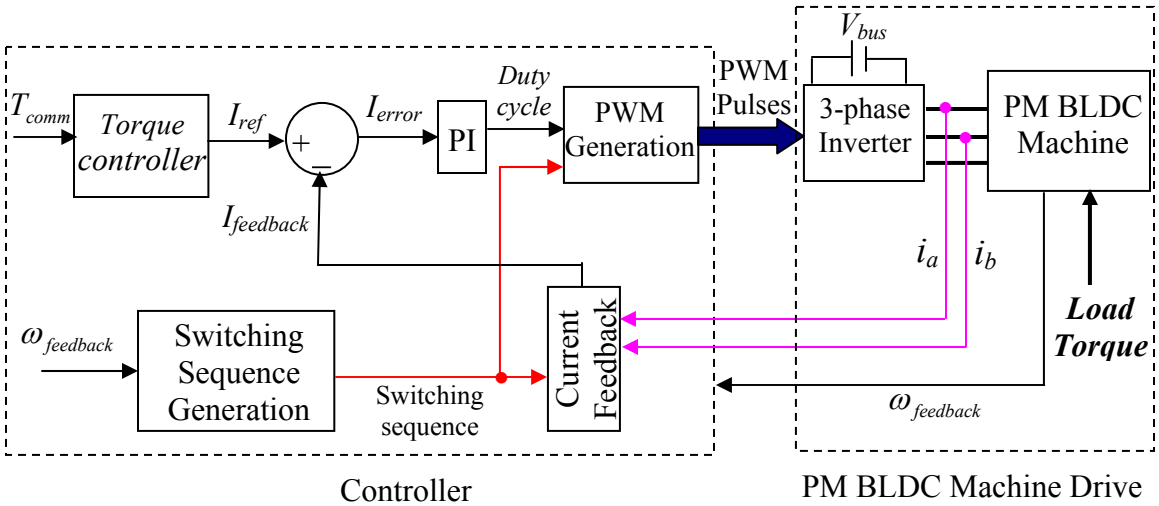


Figure 4.1. Controller block diagram for the PM BLDC drive.

Figure 4.1 shows a block diagram of the PM BLDC machine controller. The current reference I_{ref} is generated from the torque command T_{comm} that is generated by the vehicle supervisory controller. The feedback current $I_{feedback}$ is compared with I_{ref} to generate the duty cycle. The duty cycle and the switching sequences generated from speed feedback $\omega_{feedback}$ are used to generate PWM pulses for the inverter switches.

A gain scheduled and improved PI current controller with *antiwindup integrator* has been developed for current regulation. Integrator windup is a large-scale phenomenon that is caused by the finite limit of the inverter output voltage [59]. In PM BLDC operation, any two of the phases conduct at a time, depending on the position of rotor. However, the controller uses one reference value for all three phases. There are six segments of phase conduction in one electrical cycle. One new phase starts to conduct at the beginning of each segment, which requires a step in reference current from zero to the reference value I_{ref}. Without antiwindup, the integrator term can saturate to its upper limit. The antiwindup guarantees safe operation of the inverter by clamping the overshoot at the rising edge. Figure 4.2 shows the block diagram of the PI current controller. The duty cycle generated by the PI controller is

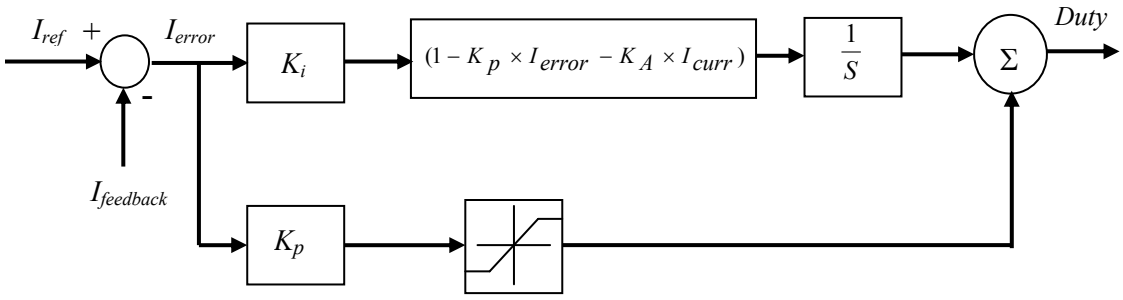


Figure 4.2. PI current controller with antiwindup integrator.

$$duty = K_p \times I_{error} + \left\{K_i \left(1 - K_p \times I_{error} - K_A \times I_{curr}\right) I_{error}\right\} \frac{1}{S} \tag{4.1}$$

where $I_{curr} = \frac{I_{feedback}}{I_{ref}}$ and K_p, K_i and K_A are proportional, integral and antiwindup constants, respectively. S is the Laplace operator. A look-up table (depending on I_{error} magnitude) for K_p is used for limiting the current overshoot.

The current regulation of a high power and high current machine depends on current feedback resolution. The digital signal processors (DSPs) used for controller

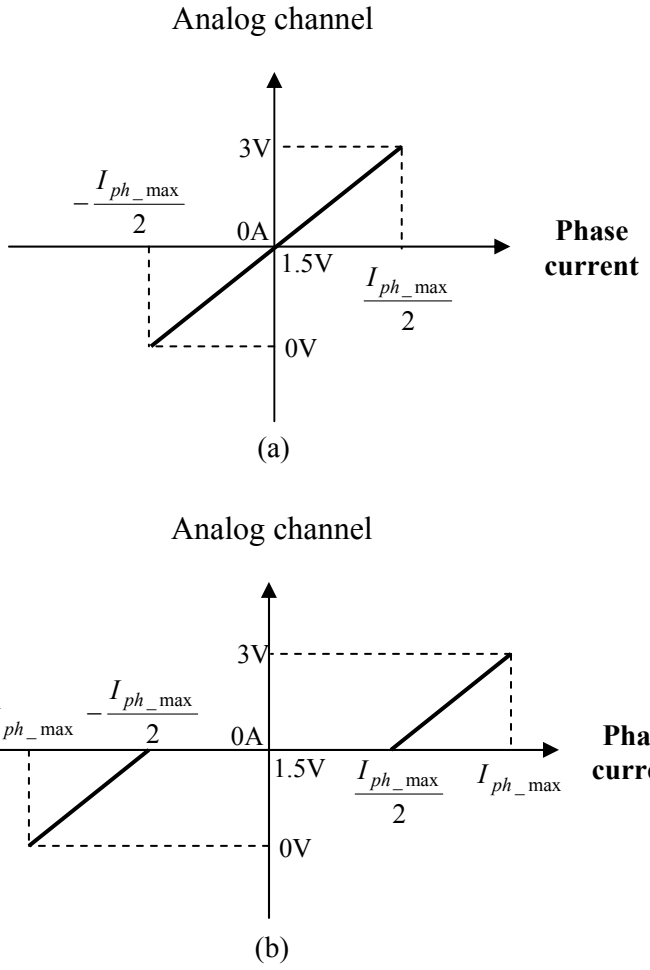


Figure 4.3. (a) Current sampling for half the maximum phase current,
(b) Current sampling for more than half the maximum phase current.

implementation typically have analog inputs with maximum input voltage of 3.3V. As a result high phase currents need to be scaled from 0 to 3V. Disturbance or noise in the measurement and hardware offsets can cause significant variations in the input of the DSP for the same current magnitude. A high-resolution current feedback is necessary to achieve the desired current regulation. In this research, an improved current sensing hardware will be used for the PM BLDC controller, which increases the resolution by a factor of 2 using the same number of current sensors. For each phase, two ranges of current are sensed by two analog input channels of the DSP, and each range is scaled from 0 to 3V. One of the A/D DSP input channels is designated for the current range $-I_{ph\_\max}/2 \le I_{ph} \le I_{ph\_\max}/2$ where $I_{ph\_\max}$ is the maximum machine phase current. The current range for the second A/D input channel is $-I_{ph\_\max} \le I_{ph} < -I_{ph\_\max}/2$ and $I_{ph\_\max} \ge I_{ph} > I_{ph\_\max}/2$. Analog electronic circuits process the signals from the two-phase current sensors before sending them to the two DSP analog inputs. Figure 4.3 shows the sampling of phase currents using two A/D channels to increase the resolution at the analog input stage. The digitized current feedback resolution inside the DSP is increased by a factor of 2 and the effect of noise is also decreased by the same factor.

4.3 MODEL BASED ANALYSIS AND SIMULATION

Model based analysis for the starter-generator has been carried out through simulation. Both the motor model and controller model have been implemented in

Matlab-Simulink. Different characteristics such as phase current, voltage, speed and torque profiles have been analyzed at different operating conditions.

The simulation model has been built using Matlab-Simulink for the three-phase 200V/400V, 4-pole PM BLDC machine with controller. Phase currents, voltages, speed and torque profiles of this machine have been analyzed that helped in controller gain tuning and inverter sizing. The PWM frequency and the nominal DC bus voltage have been selected to be 20kHz and 350V, respectively. The PM BLDC machine constant ($k\phi$) has been measured to be 0.125 volt/rad. Stator resistance and inductance are 18mΩ and 16μH, respectively. An inner loop current controller and an outer loop speed controller have been implemented in the simulation. The simulation block diagram is shown in Figure 4.4. The phase switching sequences are generated depending on rotor position. The simulation was carried out in both the motoring and generating modes.

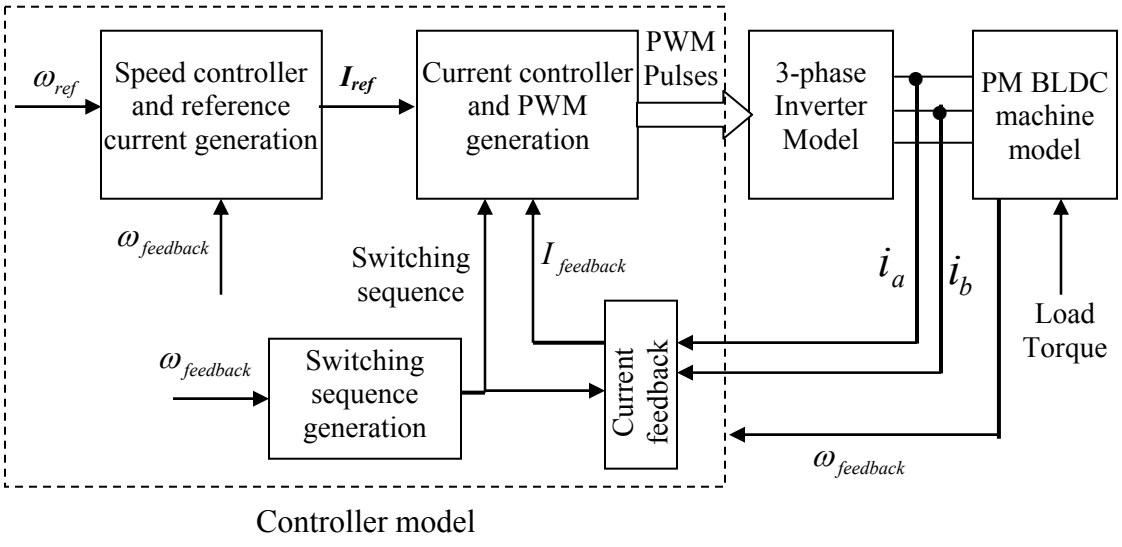


Figure 4.4. Simulation block diagram for the starter-generator.

4.3.1 SIMULATION IN MOTORING MODE

In motoring mode, a load torque of 37.5Nm has been applied, which is the continuous torque rating for the selected PM BLDC machine. A maximum speed limit was set to around the machine corner speed of 4,750rpm. Figures 4.5 and 4.6 show speed and phase current profiles in the motoring mode, respectively. From Figure 4.5, it is observed that the speed reaches 2,800rpm (1,000rpm at engine shaft for the gear ratio of 2.8:1) with 37.5Nm (105Nm at engine shaft) load torque within 100ms. The Volkswagen 1.9L diesel engine requires a starting torque of 83Nm to 92Nm at 800rmp to 1,000rpm, which will be met by this starter-generator for the gear ratio of 2.8:1. The response will get slower in the physical system due to the engine inertia that was neglected in simulation. From Figure 4.6, it is seen that phase current reaches 150A ($I_{phase} = \frac{T}{2.k\phi}$), which is the level required to deliver 37.5Nm of torque. The simulation results verify

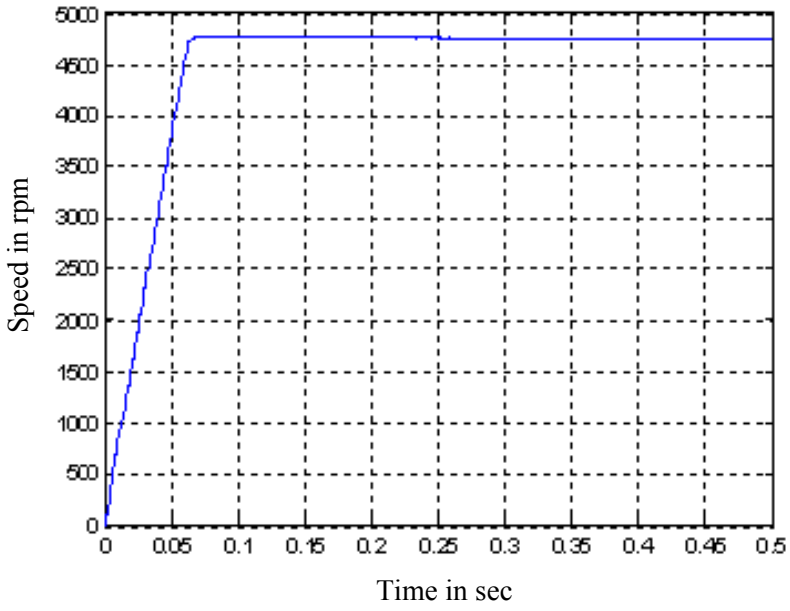


Figure 4.5. Speed profile of PM BLDC machine.

that the selected PM BLDC can provide sufficient starting torque and power to start the Volkswagen 1.9L diesel engine.

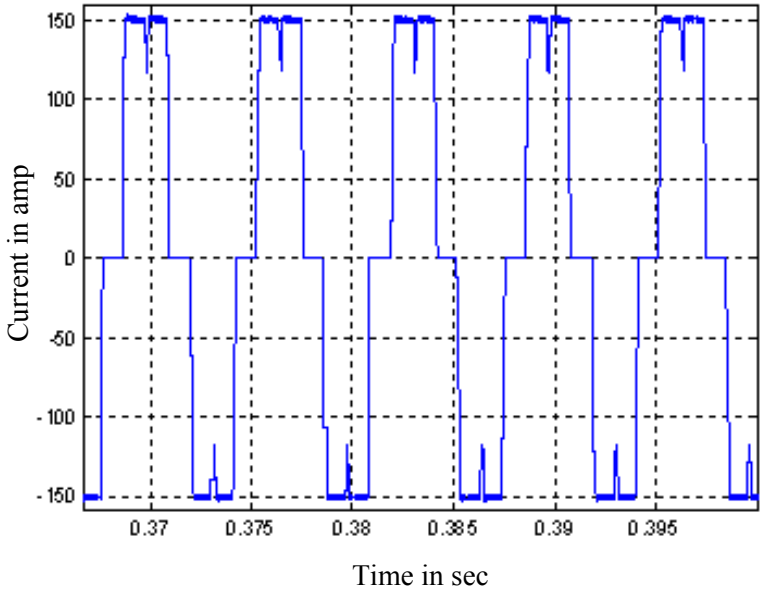


Figure 4.6. Phase current profile with 37.5Nm load.

4.3.2 SIMULATION IN GENERATING MODE

Simulation was also carried out in the generating mode, for which a constant speed of 4,750rpm (corner speed) was maintained. A torque command of –20Nm was applied (negative torque means generation), and the DC bus voltage was maintained at 350V. Figures 4.7 and 4.8 show phase current and DC charging current, respectively. From Figure 4.7, it is observed that the phase current reaches the command current of 80A ($I_{phase} = \frac{T}{2.k\phi}$) with current spikes at commutation exceeding the peak value by about 25%. The IGBTs for the inverter must be selected to handle these instantaneous current spikes to avoid inverter failure. Figure 4.8 shows that the average DC charging current is

approximately 26.5A. The input power applied is 9.948kW ($P_{in} = T\omega$), and the output power delivered by the generator is 9.275kW with an efficiency of 93.23%.

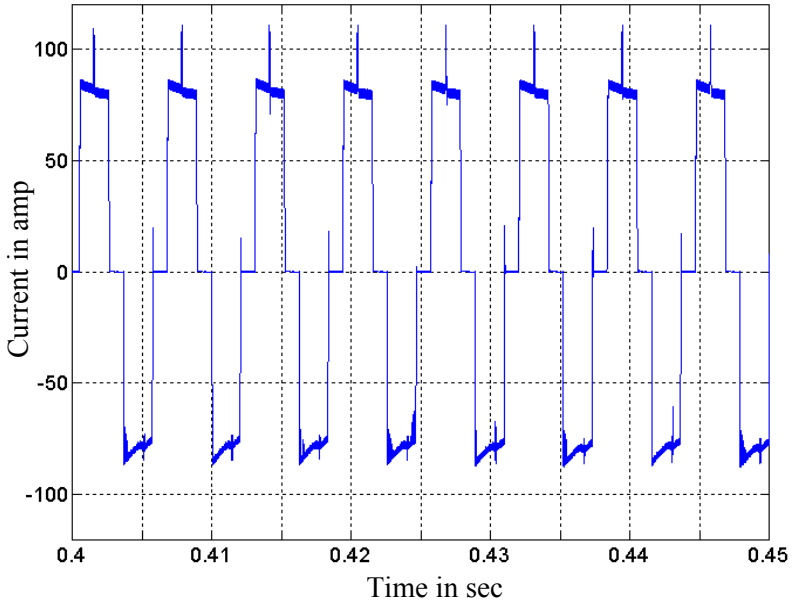


Figure 4.7. Phase current profile in generating mode.

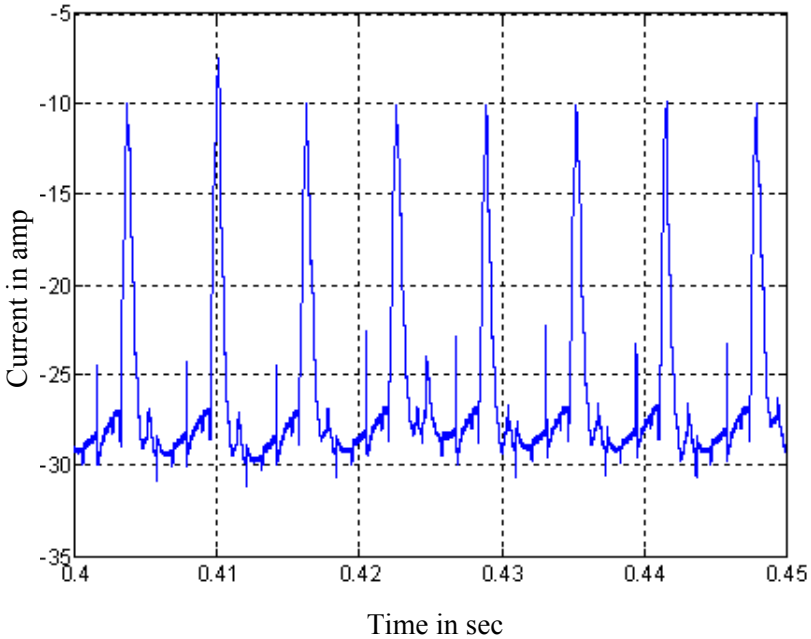


Figure 4.8. DC charging current

4.4 HARDWARE EXPERIMENTS

Hardware testing of subsystems is essential to ensure satisfactory performance of the EPTP in the integrated powertrain of the HEV. The two major subsystems of the EPTP are the high power starter-generator and the propulsion motor. This section presents the experimental setup and the results obtained used to verify the effectiveness of the high power starter-generator subsystem with an advanced controller. The starter-generator drive system has been tested after coupling the PM BLDC machine with the diesel engine inside the vehicle engine compartment.

The PM BLDC starter-generator has been tested in both motoring (engine starting) and generating modes. A battery pack of 300V has been used as the energy storage system in the DC bus for stationary tests of engine starting and generating modes. The controller and inverter were integrated and mounted under the chassis of the vehicle in its designated location following the stationary tests. For the in-vehicle testing, a 376V ultracapacitor bank was used instead of the battery pack. The diesel engine was started with the developed PM BLDC starter-generator system during in-vehicle testing. Additionally, the ultracapacitor bank was charged using the subsystem in the generating mode.

4.4.1 DRIVE HARDWARE

Figure 4.9 shows the PM BLDC starter-generator motor drive hardware subsystem. The experimental setup consists of the PM BLDC machine itself coupled to the engine, the controller implemented in a DSP, a three-phase inverter with driver, sensors, and a

hardware interfacing board. The inverter (PP400T060 from POWEREX) used for this purpose is comprised of three dual IGBT packages with a rating of 400A/600V. Three position sensors and two Hall effect current sensors are used for position, speed and phase current feedbacks.

The controller has been implemented in hardware using a TI fixed point DSP, the TMS320F2812, which has a clock frequency of 150MHz. The PWM switching frequency is set at 20kHz. The maximum PWM duty cycle is restricted to 92% of the 50us time period to give the inverter IGBTs sufficient turn-off time. The hardware interface board between the DSP and inverter processes the PWM signals, phase current feedback, position information and inverter status signals. The phase current feedbacks are processed through a high-resolution current sensing circuit. For each phase, two ranges of current are fed back to the controller through two analog input channels of the DSP with each range scaled from 0 to 3V. As a result, the current feedback resolution inside the controller is twice as high as could be achieved using a single channel; the effect of noise decreases by the same factor. The high-resolution current sensing circuit schematic for phase A is shown in Figure 4.10. Phase B circuit schematic is similar. The PWM signals generated by the DSP are applied to the hardware interface board through PWM output ports. The current, voltage and inverter status signals are fed back to the controller through ADC channels via the interfacing board.

The flow chart of the main starter-generator controller software is shown in Figure 4.11. The DSP has several built in timers that are used for various event processing. One such timer, Timer 1, has been used for controlling the duty ratio of the PWM signals for the inverter. Figure 4.12 shows the flow chart of the duty cycle generation algorithm. The

period of Timer 1 was set to be 50μsec since the PWM switching frequency was chosen to be 20kHz. Another timer, Timer 2, has been used for servicing interrupts; the corresponding interrupt service routine is shown in Figure 4.13. The period for Timer 2 was also selected 50μsec because one loop computation time was found to be 40μsec.

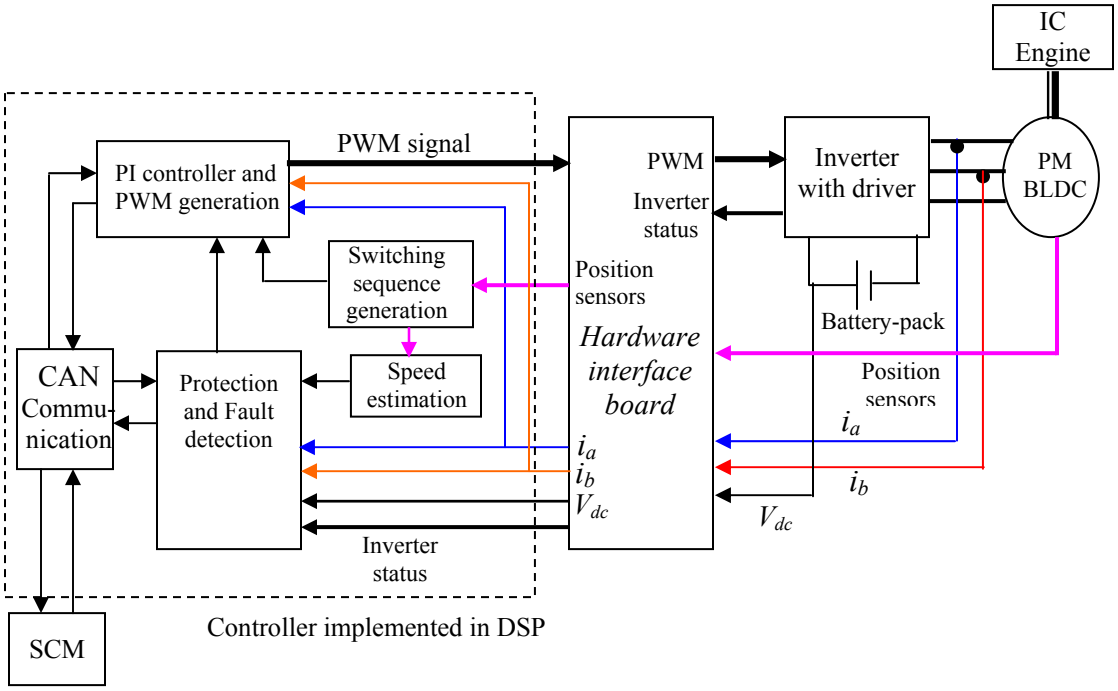


Figure 4.9. Experimental setup for the starter-generator system.

4.4.2 IN-VEHICLE COMMUNICATION AND FAULT MODES

The starter-generator controller communicates with the vehicle supervisory control module (SCM) via the CAN protocol. The CAN communication and fault mode algorithm is shown in Figure 4.14. The transmit and receive message lists for CAN are given in Tables 4.1 and 4.2, respectively.

The starter-generator controller is programmed to respond to current (torque) commands received via the CAN bus. The vehicle supervisory controller simply

transmits a certain CAN message every 10ms specifying the required maximum phase current value of the generator. The current command is a signed quantity, with a positive value corresponding to a motoring torque, and a negative value corresponding to a generating torque.

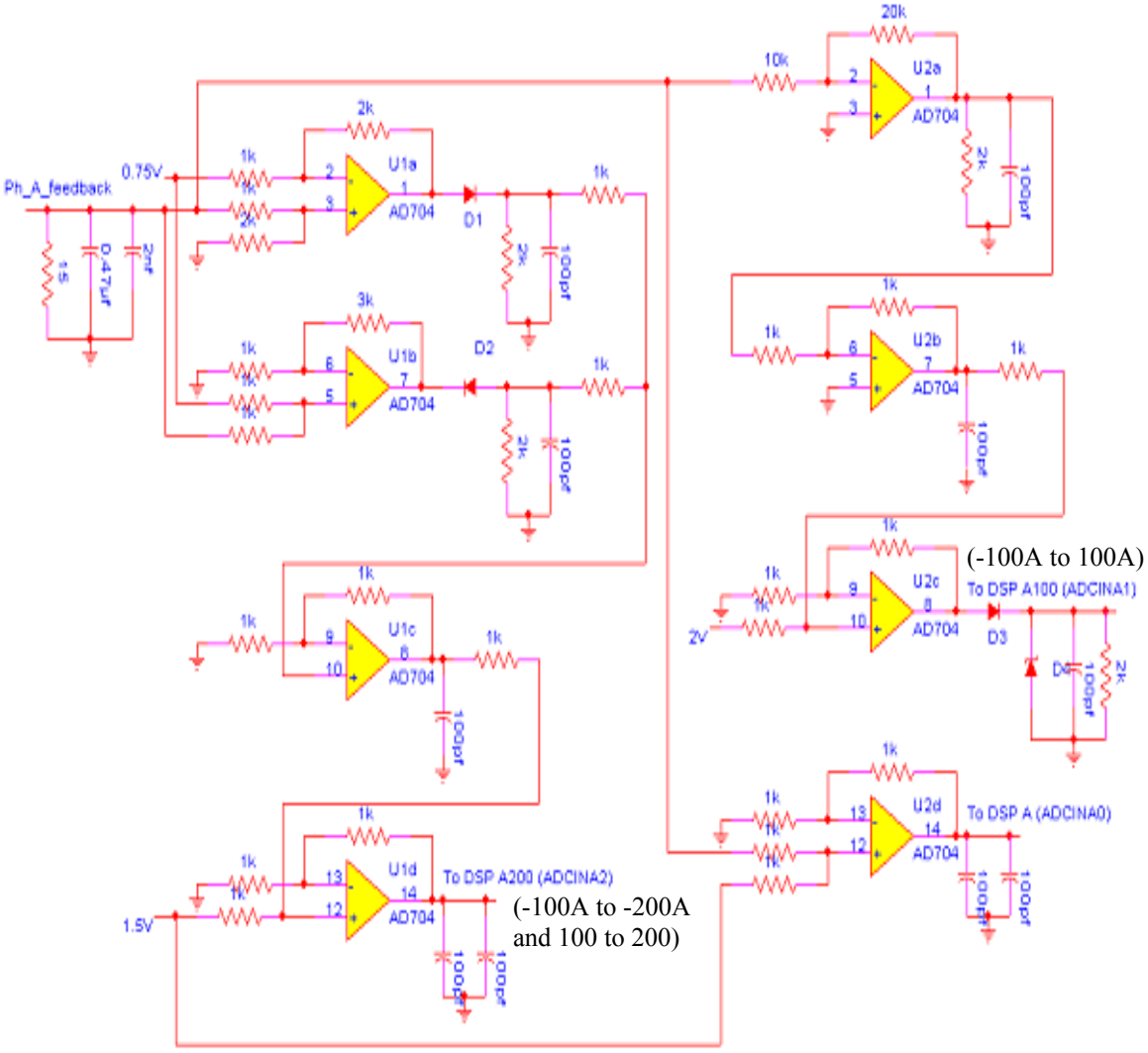


Figure 4.10 High-resolution current sensing circuit for Phase A.

The starter-generator control processor transmits two periodic CAN messages. The first is a data message transmitted every 10ms. The data message includes the computed speed, position of the PM BLDC, the measured phase current, inverter temperature and

DC bus voltage on the inverter side. The PM BLDC speed information is used by the vehicle SCM to determine the speed of the diesel engine. This speed information is essential to control the engine in series mode, and therefore requires the short 10ms

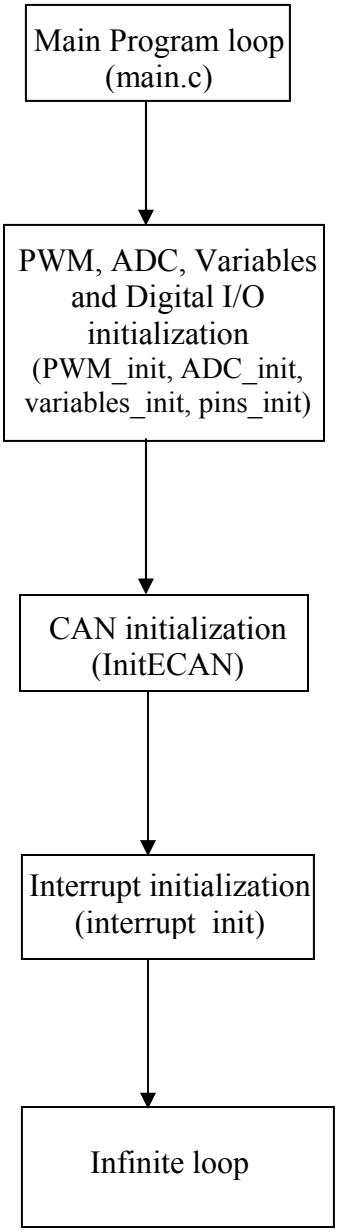


Figure 4.11. Flow chart of main Routine (main.c).

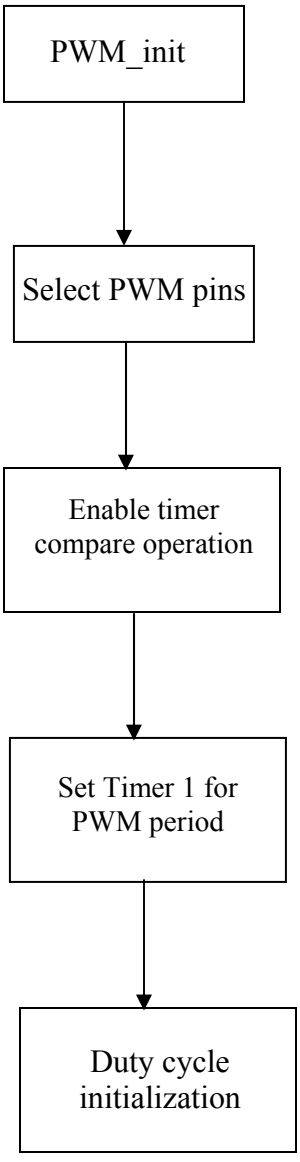


Figure 4.12. Flow chart of duty cycle generation.

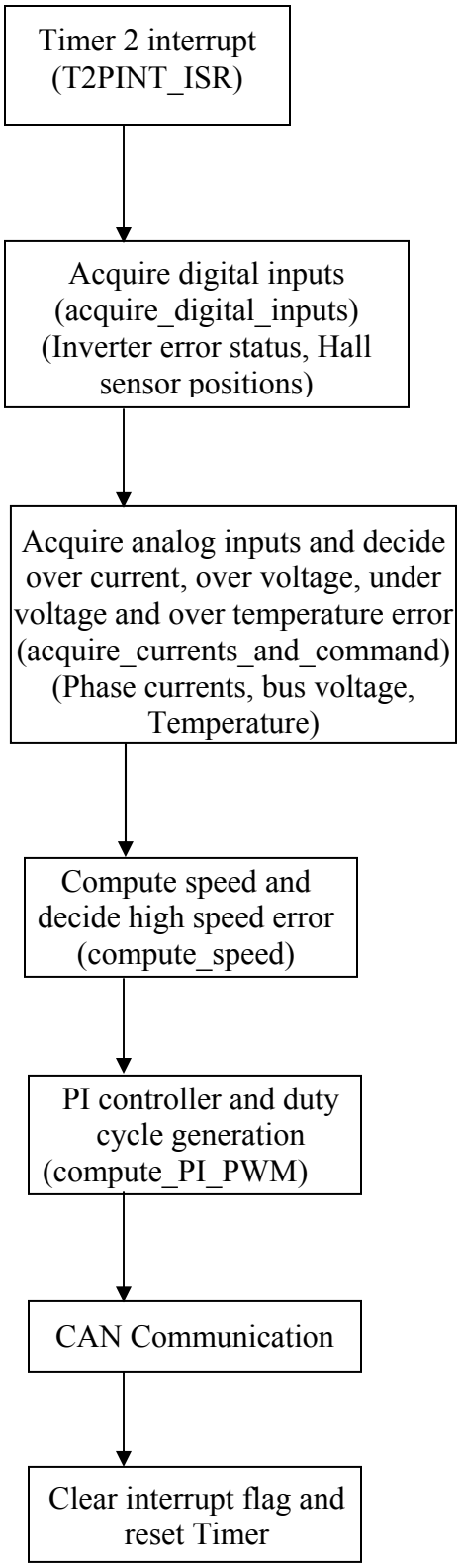


Figure 4.13. Flow chart for Timer 2 interrupts service routine.

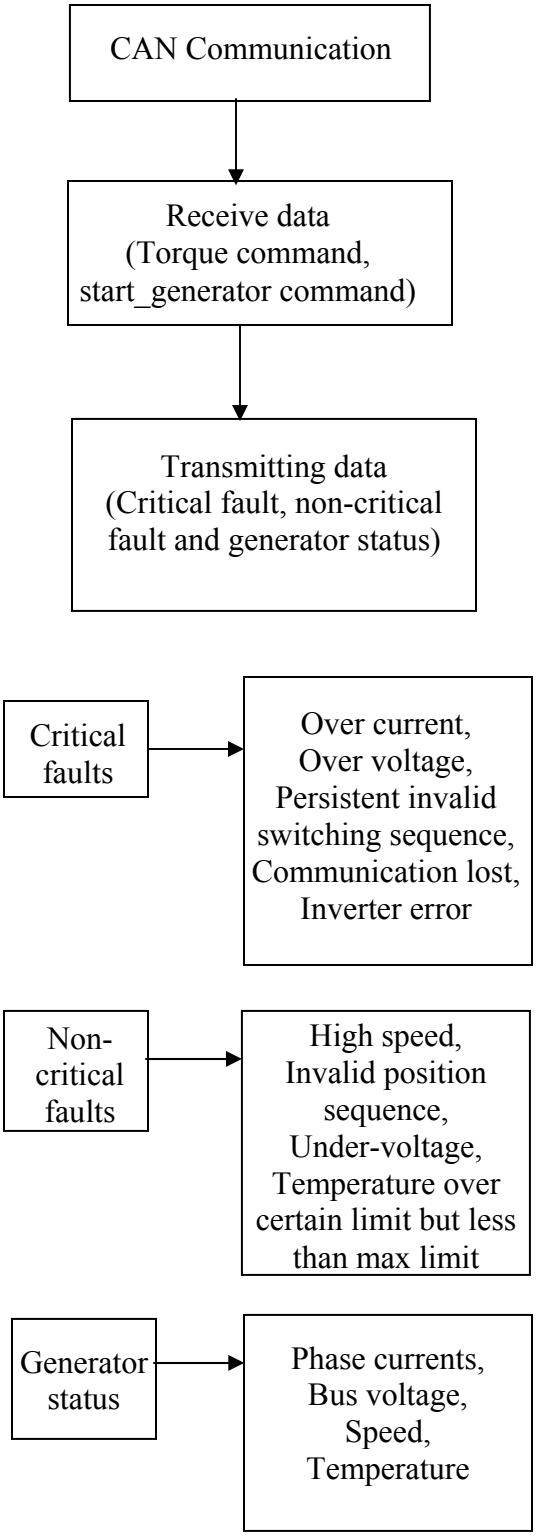


Figure 4.14. Flow chart for CAN communication; Different faults.

Table 4.1. Transmit message list

	Signal name	Data type	Length	Definition	Conversion range
Different faults CAN ID: 0x468, BYTE 1	local_critical_fault (100ms update)	US	1 bit	a fault so serious that the generator needs to be shut down	N/A
	local_noncritical_fault (100ms update)	US	1 bit	a fault serious enough that SCM may want to change strategy	N/A
	local_diagnostic_fault (100ms update)	US	1 bit	a non-serious fault that should be data logged	N/A
	four_bit_rolling_count (100ms update)	US	4 bit	a rolling counter to make sure the processor is running and up	from 0-15
Local critical faults (updates at 100ms) CAN ID: 0x468, BYTE 2 & 3 (2 bits of BYTE 3)	critical_communications_with_UALAN_lost	US	1 bit	no message from/to SCM in 50 ms	N/A
	critical_inverter_ overheat (Inverter error status)	US	1 bit	Inverter temperature above 95^0C	N/A
	critical_error_on_inverter_phaseA (Inverter error status)	US	1 bit	inverter indicates over current or short on phase A	N/A
	critical_error_on_inverter_phaseB (Inverter error status)	US	1 bit	inverter indicates over current or short on phase B	N/A
	critical_error_on_inverter_phaseC (Inverter error status)	US	1 bit	inverter indicates over current or short on phase C	N/A
	critical_instantaneous_ overcurrent	US	1 bit	Phase current > 195A for over 2ms	N/A
	critical_continuous_overcurrent	US	1 bit	Phase current > 165A for over 1sec	N/A
	critical_overvoltage	US	1 bit	DC bus Voltage ≥ 400V	N/A

Table 4.1. Transmit message list (continued)

	Signal name	Data type	Length	Definition	Conversion range
	critical_persistent_invalid_position_sequence	US	1 bit	Persistent invalid rotor position (position not advancing 60 degrees for more than ten readings)	N/A
Local non-critical faults CAN ID: 0x468, BYTE 3	noncritical_under voltage (100ms update)	US	1 bit	DC bus voltage below 200V	N/A
	noncritical_speed_high (100ms update)	US	1 bit	Speed is higher than 12500rpm	N/A
	noncritical_inverter_ overheat (100ms update)	US	1 bit	80^0C ≤ Inverter temperature < 95^0C	N/A
Local diagnostics faults CAN ID: 0x468, BYTE 3	invalid_position_ sequence (100ms update)	US	1 bit	Invalid rotor position (position not advancing 60 degrees)	N/A
	invalid_position_ sensed (100ms update)	US	1 bit	Hall sensors gives invalid position reading	N/A
Generator/inverter status CAN ID: 0x440	current_fb, BYTE 1, 2 (Current feedback; 10ms update);	S	9 bits	Machine Phase current	0-200A
	speed, BYTE 3, 4 (10ms update)	US	16 bits	Machine speed	0-16000rpm
	Vbus, BYTE 5 (10ms update)	US	8 bits	DC bus voltage	0-400V
	Temperature, BYTE 6 (10ms update)	US	8 bits	Inverter temperature	0-96^0C

US → Unsigned, S → Signed

Table 4.2. Receive message list (CAN ID: 0x340, BYTE 1, 2 & 3)

Signal name	Length	Data type	Definition	Conversion range
Connect_generator (10ms update)	1 bit	US	Connect/disconnect generator with/from Inverter	N/A
*Current_cmd (10ms update)	9 bit	S	Current command from SCM	0-200A

* Negative Current_cmd means generating mode.

update period. The phase current information is used by the SCM and by the vehicle system health monitor to ensure that the actual phase current is close to the commanded value; a discrepancy may indicate a system malfunction. The DC voltage information may also be used by the supervisory controller, and is monitored by the vehicle system health monitor. The DC voltage measurement is redundant with similar measurements provided by two other vehicle subsystems.

The second CAN message is a periodic starter-generator system status message transmitted every 100ms. This message contains fault flags that are raised by a 'Protection and Fault Detection' code. Potential faults include position-sensor errors, speed computation inconsistencies, DC bus over voltage, currents out of range and inverter overheat. The system status message also contains a four-bit rolling count that is monitored by the vehicle system health monitor as an indication that the starter-generator control processor is up and running.

The 'Protection and Fault Detection' code in the controller ensures safety and security of the inverter and generator, as well as safe operation of the vehicle. The faults have been categorized in two classes: critical faults and non-critical faults. Phase over current (continuous and instantaneous), DC bus over voltage, CAN communications lost

and inverter overheat are considered critical faults. In the event of critical faults, the controller disconnects the generator from the inverter and informs the SCM of the fault status. Speeds out of range, temperature above certain limit and DC bus under voltage are considered as non-critical faults. In the event of non-critical faults, the controller raises caution flags to alert the SCM, which may take actions necessary to avoid critical faults.

Non-critical faults are cleared automatically by the controller when the conditions that caused them revert to normal. Once a critical fault has occurred, a handshake is required before the controller will reconnect the starter-generator system. The SCM must bring its commanded current to zero, and assert a bit requesting that all faults be cleared. If the condition(s) causing the critical fault have in fact cleared, the controller will respond by resetting all fault flags and reconnecting the starter-generator system.

4.4.3 RESULTS FOR ENGINE STARTING

Starting of the 1.9L diesel engine was tested with the PM BLDC machine coupled with the engine in the vehicle. A battery pack of 300V (nominal) was used at the DC bus as the energy storage system. For engine starting, a torque command was applied for certain duration of time. Several experiments were carried out to find the minimum starting time and current drawn from the DC source without over cranking. The 1.9L diesel engine usually requires more than 80Nm at a speed of 800rpm to 900rpm. Figure 4.15 shows the speed profile of engine starting for 400ms. From this figure, it is observed that the engine cranks at a motoring speed of 2240rpm (800rpm at engine shaft), but the speed reaches more than 3000rpm during over cranking. Figure 4.16 shows the DC bus current and phase current profiles of the PM BLDC machine for an engine starting

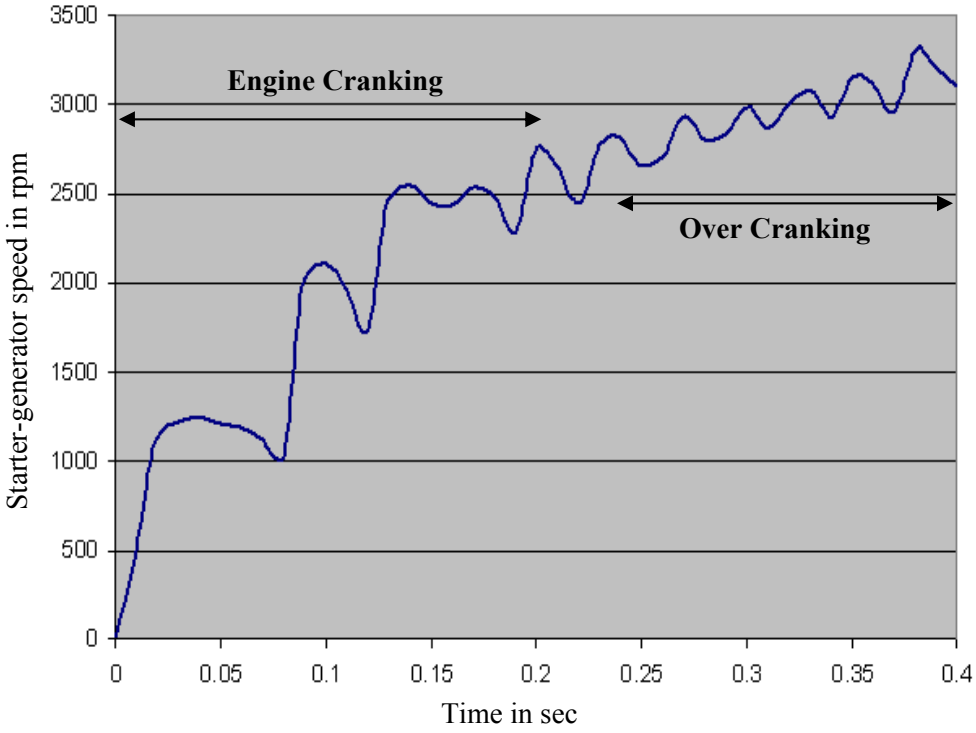


Figure 4.15. Speed profile of engine starting for 400ms.

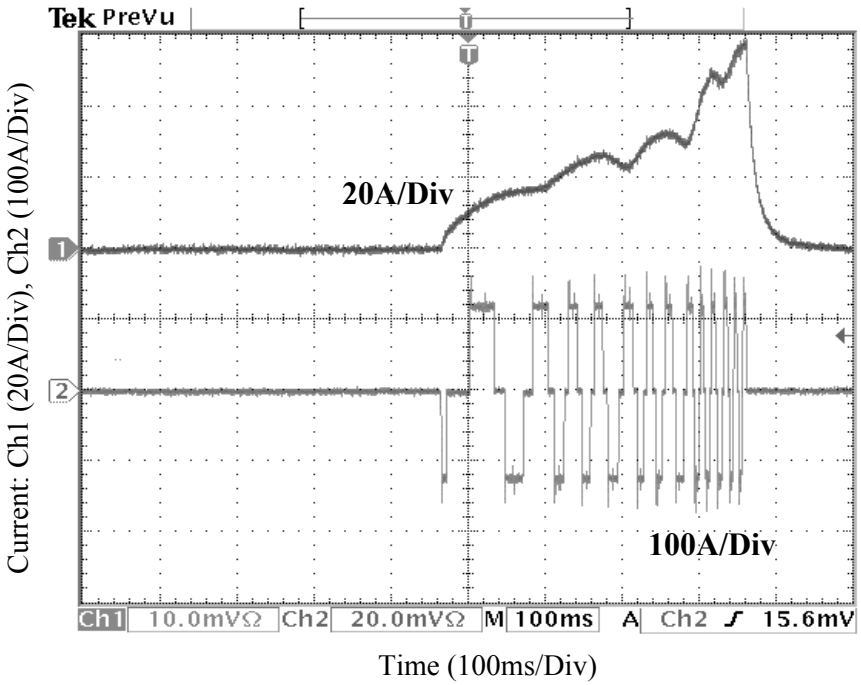


Figure 4.16. Ch1: DC bus current, and Ch2: Phase current (400ms).

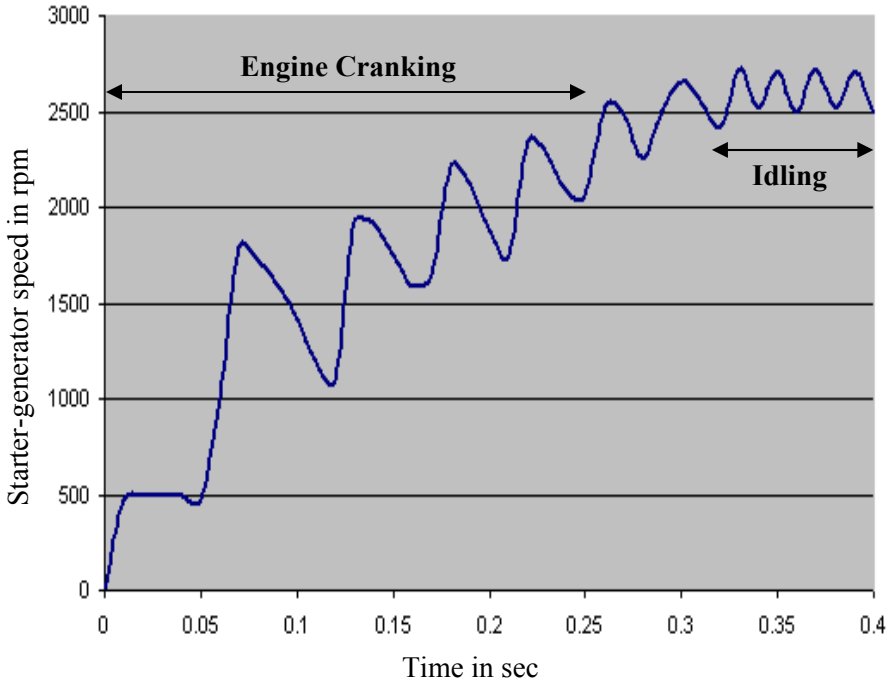


Figure 4.17. Speed profile for engine cranking within 250ms.

duration of 400ms. The DC bus current reaches 60A, and machine phase current reaches 160A during over cranking. Figure 4.17 shows the starter-generator speed profile for engine starting within 250ms. The starter-generator speed reaches 2225rpm ($\approx$ 800rpm at engine shaft) within 250ms, which is the cranking speed for the engine, and the engine idles at 900rpm to 950rpm (2520rpm to 2660rpm at generator shaft). Figure 4.18 shows the DC bus current and phase current profiles of the PM BLDC machine for the engine starting duration of 250ms. The DC bus current reaches 27A and machine phase current reaches 125A, which is much less than the safe conductor current handling limit in a vehicle. The torque available at the engine shaft corresponding to the machine phase current of 125A is 87Nm. The results show that the high power PM BLDC starter-generator starts the engine without over cranking, within 250ms, compared to 2-3secs

required by a conventional starter to start the same engine. The current drawn from the energy storage system is also less.

4.4.4 RESULTS FOR POWER GENERATION

The starter-generator has been tested in the generating mode at different loads as well as at different speeds with the diesel engine serving as the prime mover. The same

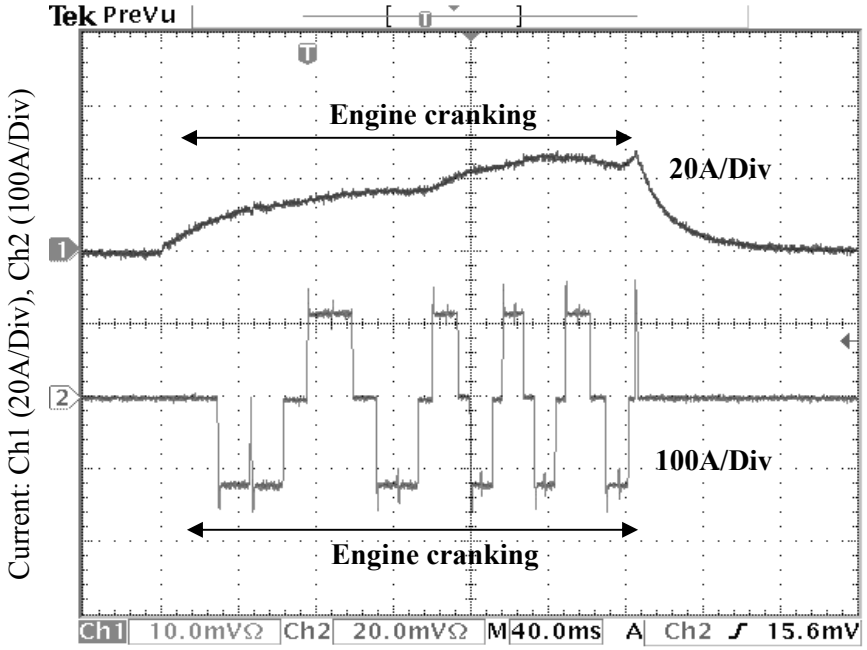


Figure 4.18. Ch1: DC bus current, and Ch2: Phase current (250ms).

battery pack of 300V (nominal) was used as the DC source, and a resistance bank was used as the load. Figure 4.19 shows the phase current and DC bus charging current at 30Nm torque command, for which the speed and charging voltage were 3200rpm and 365V, respectively. The average peak of phase current reaches 116A (corresponding torque is 29Nm) with a 3.33% error in torque/current regulation. The DC bus charging

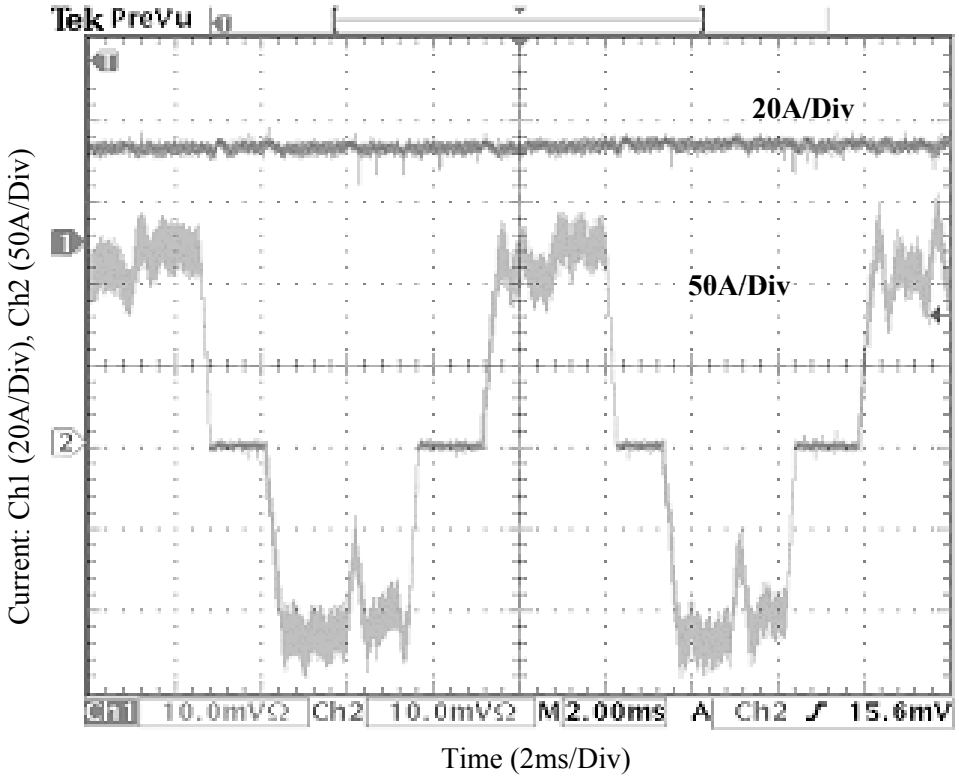


Figure 4.19. Ch1: DC bus current, and Ch2: Phase current (generation).

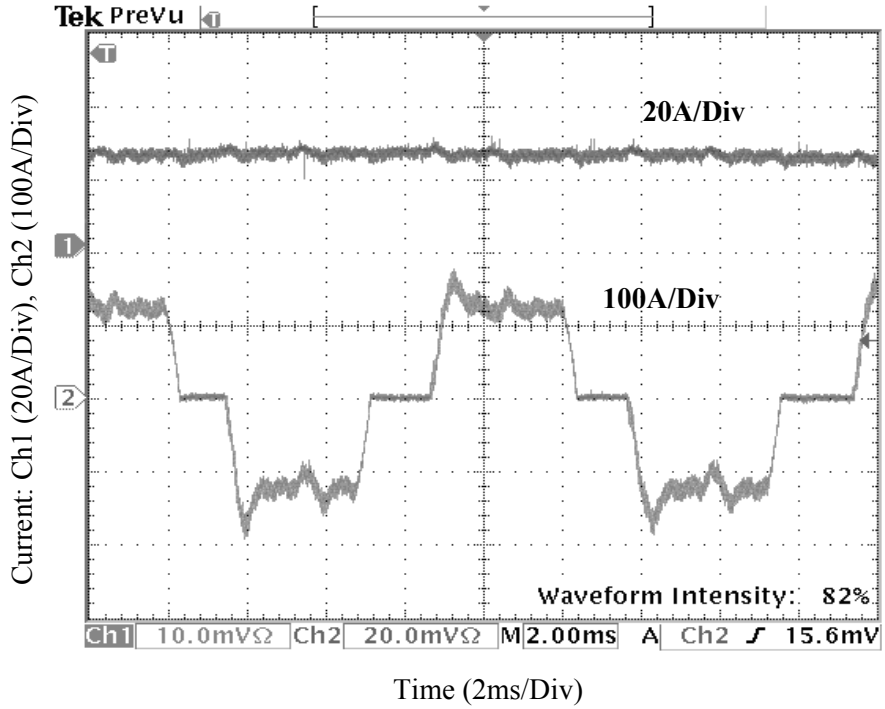


Figure 4.20. Ch1: DC bus current, and Ch2: Phase current (generation).

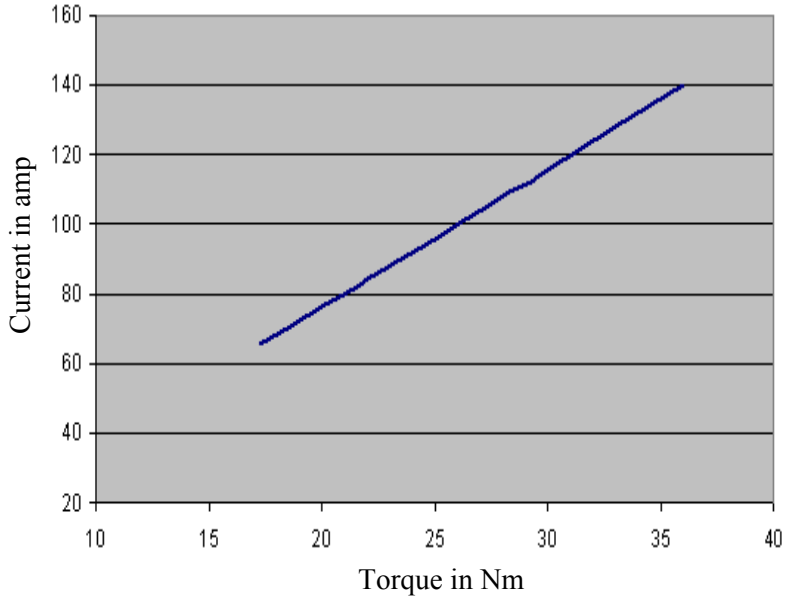


Figure 4.21. Average peak of phase current for different torque commands at 4200rpm.

current reaches 23.6A. In this experiment, the input power to the PM BLDC is 9.717kW ($P_{in} = T\omega = 29 \times \frac{2 \times \pi \times 3200}{60} = 9.717\text{kW}$), and the output power delivered to the DC bus is 8.614kW ($P_{out} = V_{dc} I_{dc} = 365 \times 23.6 = 8.614\text{kW}$), which corresponds to 88.64% efficiency for the PM starter-generator system. Figure 4.20 shows the phase current and DC bus charging current at a 36Nm torque command where the speed and charging voltage were 3000rpm and 385V, respectively. The average peak of the phase current reaches 140A (corresponding torque is 35Nm) with a tracking error of 2.77%, and the DC bus charging current reaches 26A. In this experiment, the input power to the PM BLDC is 11kW ($P_{in} = T\omega = 35 \times \frac{2 \times \pi \times 3000}{60} \approx 11\text{kW}$), and the output power delivered to the DC bus is 10.01kW ($P_{out} = V_{dc} I_{dc} = 385 \times 26 = 10.01\text{kW}$) with an efficiency of 91%.

The average peaks of phase currents maintain the linear relationship with torque commands at a constant speed of 4200rpm, as shown in Figure 4.21. Figure 4.22 shows the DC bus charging current vs. speed profiles in the generating mode for different torque commands. From this figure, it can be observed that the DC bus charging current for a certain torque command increases at higher rate with increase in speed of the generator, indicating better efficiencies near the corner speed of 4,750rpm. Table 4.3 shows the

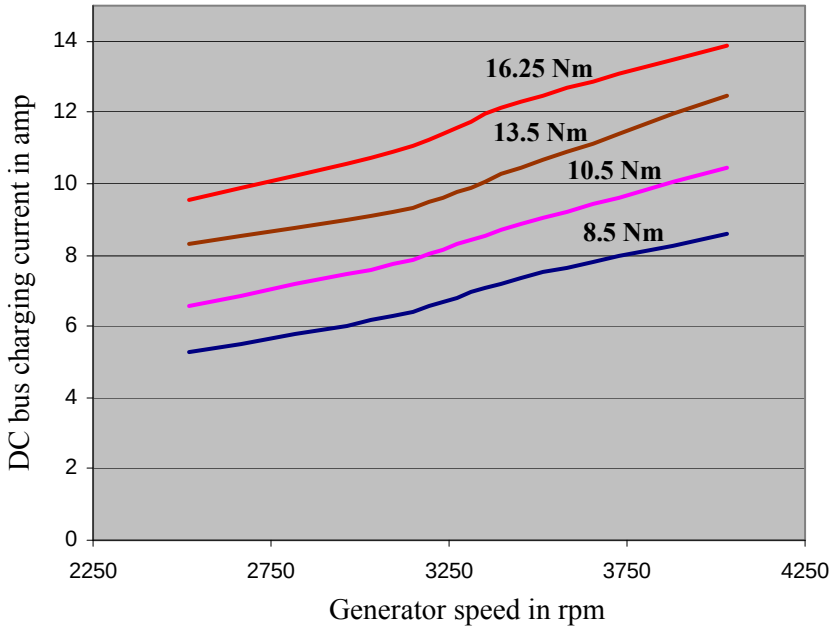


Figure 4.22. DC bus charging current vs. speed at different torque commands.

Table 4.3. Efficiency, inputs and outputs at different load conditions

P_{in} (kW)	P_{out} (kW)	$T_{command}$ (Nm)	T_{actual} (Nm)	I_{DC} (Amp)	$I_{phase, max}$ (Amp)	Charging voltage, V_{DC} (Volt)	Efficiency
11	10.01	36	35	26	140	385	91.00%
9.717	8.614	30	29	23.6	116	365	88.64%
7.52	6.4	22.5	21.5	19.4	86	330	85.10%
6.53	5.423	20	19	17	76	319	83.05%

input and output power, and efficiencies at different load conditions in generating mode. The efficiency increases as the output power increases towards the rated power.

4.5 CONCLUSION

This chapter is dedicated to the advanced controller and hardware design, simulation, and experimentation of the PM BLDC starter-generator subsystem. Simulations in both motoring and generating modes have been carried out to design the controller and inverter. From the simulation results, it was proven that the selected PM BLDC starter-generator components have enough power and torque production capability to start the 1.9L diesel engine. The simulation also helped in tuning the controller parameters as well as selecting proper inverter rating for the starter-generator drive.

The controller for the high power starter-generator has been implemented in hardware successfully using a 32-bit fixed point digital signal processor (TI DSP TMS320F2812). The experimental results demonstrate the performance of the starter-generator subsystem in different operating modes. The high power density of the starter-generator and a well-designed torque controller enabled the 1.9L diesel engine to start in 250ms, which helps to reduce the tailpipe emissions and engine torque ripple. The current drawn at starting is also minimized. The quick start-stop feature of the engine helps improve fuel economy by reducing unburnt fuel during starting and turning the engine off instead of idling in a vehicle stop situation. The starter-generator was also operated at its highest efficiency point for power generation. The experimental results in the generating mode show that the new high-resolution current sensing board enables excellent current regulation in the controller.

CHAPTER V

INDUCTION MOTOR DRIVE FOR EFFICIENT PROPULSION SYSTEM

The propulsion motor is a key component of the hybrid electric vehicle. In this research, an induction motor drive has been considered for the propulsion system. Induction motors exhibit high dynamic and static performance in traction applications as well as in HEV applications. Induction motors are cheap, reliable, robust, and have instant power and torque capability as well as wide speed operation range while maintaining a good efficiency. However, the controller needs accurate knowledge of the machine parameters to achieve high performance. Parameter mismatch between controller and the motor detunes the controller, and deteriorates the performance of the motor drive. Therefore, for efficient use of induction motor, an on-line parameter estimation and adaptation technique is necessary. This technique must be simple, reliable, and robust in the event of disturbance and internal noise. The adaptation technique must also be sufficiently simple computationally to be done in real time on a processor. From the modeling of induction motor and control equations, it has been found that rotor resistance is the dominant parameter in the controller. In addition, stator resistance information helps to know the machine temperature that is essential for thermal management. A novel observer based adaptation method for tracking and adapting the rotor and stator resistance in real time is presented in this chapter.

5.1 IMPACT OF PARAMETER VARIATION ON INDUCTION MOTOR PERFORMANCE

Variation in the rotor time constant has a significant influence on the performance of an induction motor. Rotor resistance changes because of motor heating, skin effect or other non-idealities. In this section, the impact of parameter variation on motor performance will be addressed, which have motivated this part of the research. The effects of parameter variation (especially rotor resistance) on the performance of an electric vehicle propulsion system have been studied through simulation carried out in Matlab-Simulink.

The indirect method of vector control in the rotor flux reference frame has been implemented to achieve high performance from an induction motor drive. In this case, a speed control mode has been implemented for a constant rotor flux reference. The velocity profile of the electric vehicle used in simulation is the SAE urban driving schedule B (maximum speed 20mph with 19secs of acceleration followed by 19sec of cruise time) [60]. A small sedan type vehicle of 2200 lbs. has been considered for simulation. The ratings of induction motor used in simulation are 30kW, 280V, 4 pole, and 60Hz. The switching frequency of the inverter has been chosen to be 20kHz. The machine parameters are: stator resistance 0.087Ω, rotor resistance 0.228Ω, stator/rotor inductance 0.8mH, mutual inductance 34.7mH and moment of inertia 1.662 kg-m^2.

Figures 5.1, 5.2 and 5.3 show the motor torque, stator current and speed profiles, respectively, without considering parameter variation. From Figure 5.2, it is seen that the quadrature axis component (i_q) and direct axis (i_d) component of stator current at steady

state are 12.5A and 27.5A, respectively. The simulation has also been carried out with a 50% increase in rotor resistance in the motor model where the rotor resistance in the controller remained the same as the original value. Figures 5.4, 5.5, 5.6 and 5.7 show the motor torque, stator q-axis current (i_{qs}), stator d-axis current (i_{ds}) and speed profile with 50% increase in rotor resistance in the motor model, respectively. From Figure 5.4, it is

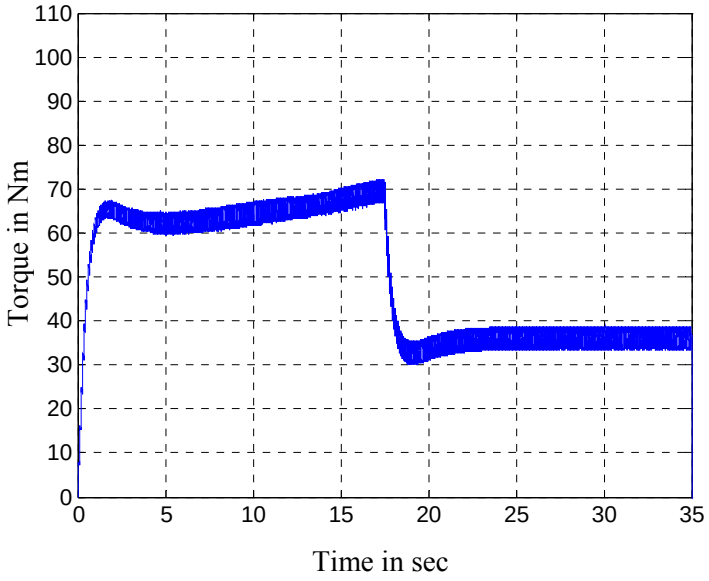


Figure 5.1. Motor Torque T_e.

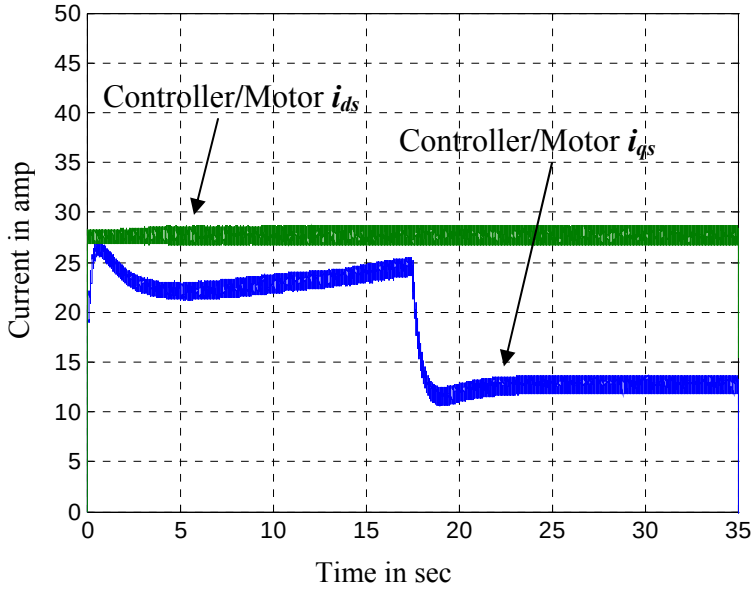


Figure 5.2. Controller/Motor $\boldsymbol{i_{qs}}$ and $\boldsymbol{i_{ds}}$ current without parameter variation.

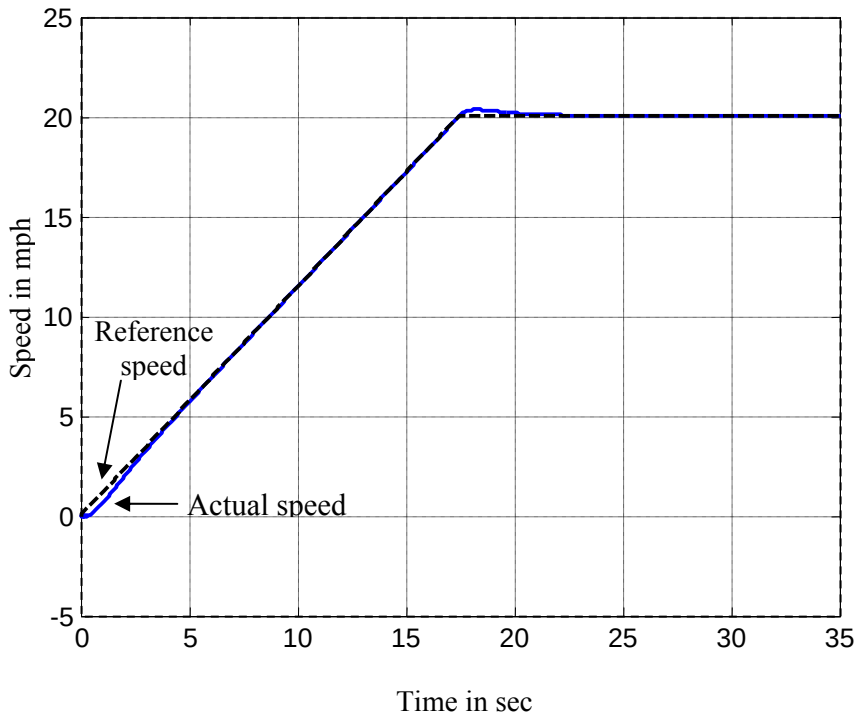


Figure 5.3. Reference and actual speed under SAE schedule B.

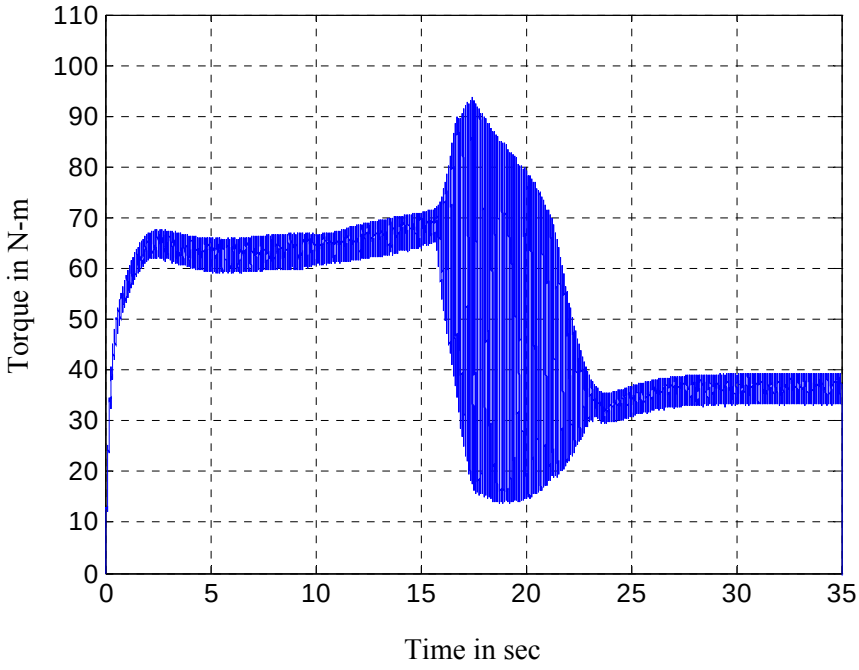


Figure 5.4. Motor torque with 50% increase of rotor resistance in motor model.

observed that the torque does not respond instantaneously; rather there is a 4sec delay. Moreover, there are oscillations during the step change in torque command. From Figures 5.5 and 5.6, it is observed that stator *d*-*q* axis currents in the controller do not match with

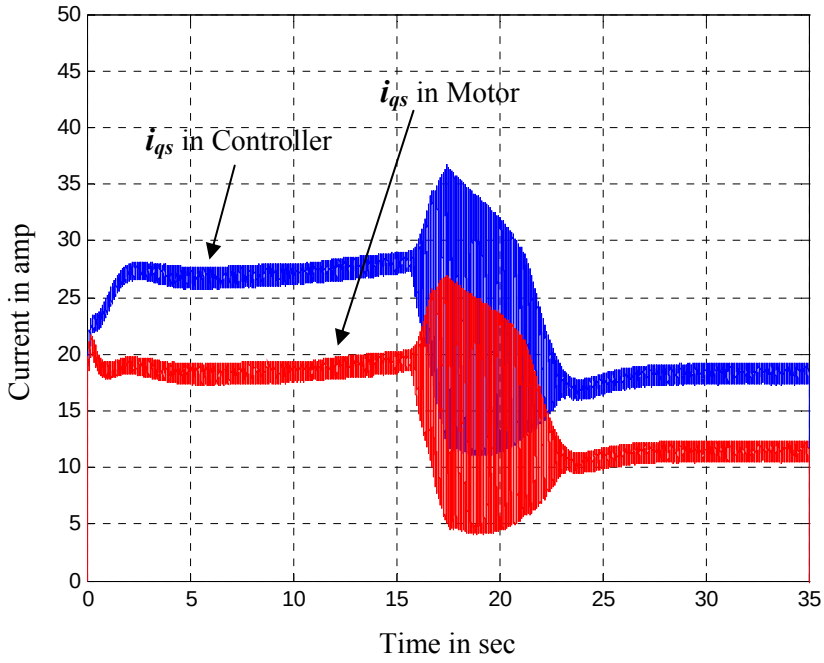


Figure 5.5. Stator *q*-axis current with 50% increase of rotor resistance in motor model.

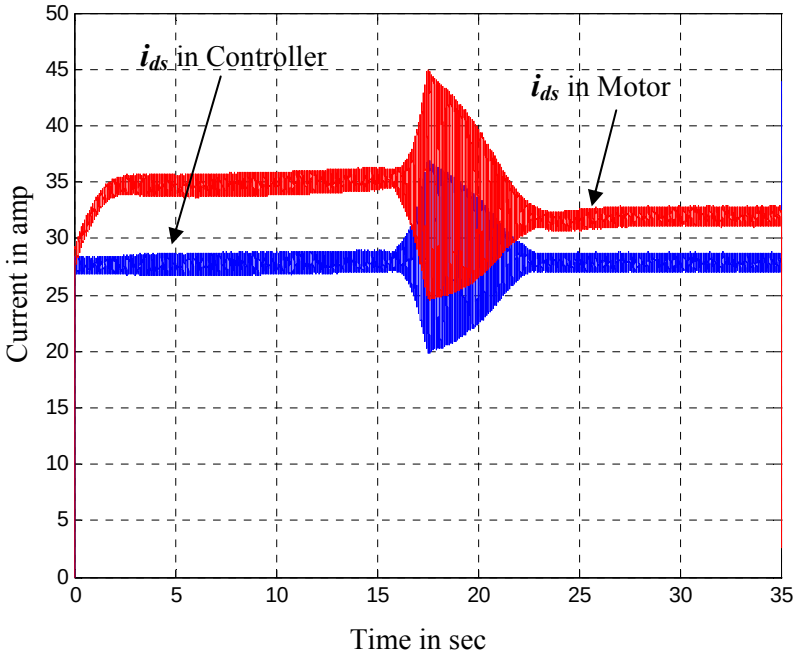


Figure 5.6. Stator *d*-axis current with 50% increase of rotor resistance in motor model.

the stator *d-q* axis currents in the motor model because of incorrect rotor resistance value in the controller. The motor i_{qs} current (11.5A) is less than the controller i_{qs} current (18.9A) but the motor i_{ds} current (31.9A) is greater than the controller i_{ds} current (27.8A) for the detuned operation. Also, the motor i_{qs} current in detuned condition is less than the motor i_{qs} current in tuned condition, but the motor i_{ds} current in detuned condition is greater than the motor i_{ds} current in tuned condition. Again, from Figures 5.2, 5.5 and 5.6, it is also observed that the net stator current $\left(i_s = \sqrt{i_q^2 + i_d^2} = \sqrt{(11.5)^2 + (31.9)^2} \approx 34A \right)$ of the motor in detuned condition is greater than the net stator current $\left(i_s = \sqrt{i_q^2 + i_d^2} = \sqrt{(12.5)^2 + (27.5)^2} \approx 30A \right)$ in tuned condition by 4A. The higher current increases the stress on the energy storage system and reduces the efficiency of the vehicle power train. It is also observed that the flux component i_{ds} does not remain constant throughout the operation; this is undesired in vector control. Figure 5.7 shows that the

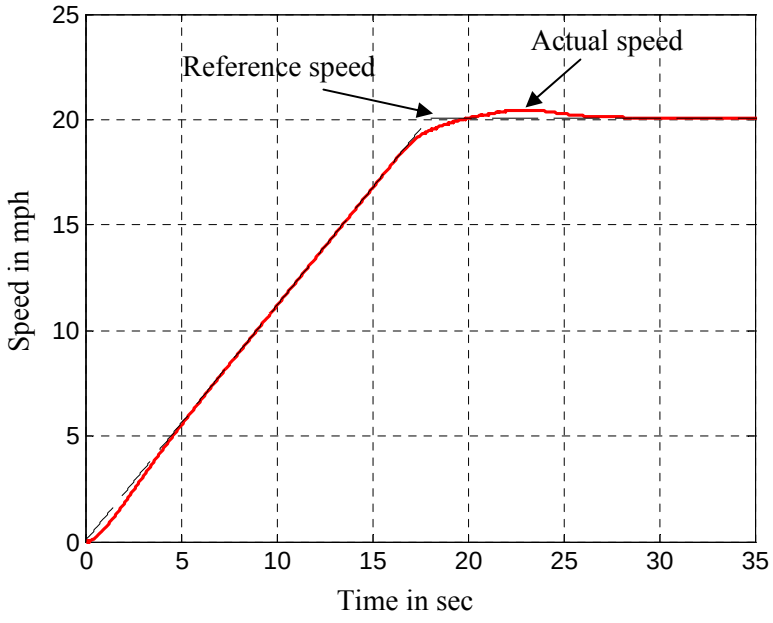


Figure 5.7. Speed profile with 50% increase of rotor resistance in motor model.

speed does not reach the cruise speed (20mph) within the acceleration time of 19sec. These drawbacks can be overcome only by on-line parameter adaptation.

5.2 CONTROLLER WITH ON-LINE PARAMETER ADAPTATION ALGORITHM

An on-line parameter adaptation technique for an induction motor drive helps to achieve high static and dynamic performance for demanding applications like hybrid electric vehicles. The existing parameter estimation and adaptation algorithms are complex, and need detailed motor modeling and extensive computation as well as memory and processor time. In this research, a Luenberger-sliding mode observer and an adaptive identification method for parameter adaptation have been developed and implemented. The observer is a simple fourth-order observer, and requires significantly less computation compared to the observers available in the literature. Since the Luenberger part of the observer is predetermined by the motor model, it is easy to set the sliding mode gain. The Luenberger part along with the sliding mode observer helps quick tracking of the parameter. The sliding mode part of the observer takes care of the relatively large disturbances in measurements, significant mismatches in the parameter, or internal noises in the system. In the event of small differences or small parameter differences, the Luenberger part of the observer is sufficient.

The state space equations of an induction motor in the stationary reference frame are [40]:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{V}_s \text{ and } \boldsymbol{i}_s = \boldsymbol{C}\boldsymbol{x} \tag{5.1}$$

where $\boldsymbol{x} = [\boldsymbol{i_s} \quad \boldsymbol{\lambda_r}]^T$, $\boldsymbol{B} = \begin{bmatrix} \dfrac{\boldsymbol{I}}{\sigma L_s} & 0 \end{bmatrix}^T$, $\boldsymbol{C} = [\boldsymbol{I} \quad \boldsymbol{0}]$, stator current $\boldsymbol{i_s} = [i_{ds} \quad i_{qs}]^T$

rotor flux $\boldsymbol{\lambda_r} = [\lambda_{dr} \quad \lambda_{qr}]^T$, $\boldsymbol{A} = \begin{bmatrix} -\dfrac{1}{\sigma}\left(\dfrac{1}{\tau_s} + \dfrac{1-\sigma}{\tau_r}\right)\boldsymbol{I} & \dfrac{L_m}{\sigma L_s L_r}\left(\dfrac{\boldsymbol{I}}{\tau_r} - \omega_r \boldsymbol{J}\right) \\ \dfrac{L_m \boldsymbol{I}}{\tau_r} & -\dfrac{\boldsymbol{I}}{\tau_r} + \omega_r \boldsymbol{J} \end{bmatrix}$

$\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\boldsymbol{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Here $\sigma = 1 - \dfrac{L_m^2}{L_s L_r}$, $\tau_s = \dfrac{L_s}{R_s}$ and $\tau_r = \dfrac{L_r}{R_r}$. i_s, V_s and λ_r are stator current, stator voltage and rotor flux, respectively. R_s, R_r, L_s, L_r and L_m are stator resistance, rotor resistance, stator inductance, rotor inductance and mutual inductance, respectively.

5.2.1 LUENBERGER-SLIDING MODE OBSERVER

The Luenberger-sliding mode observer, which simultaneously estimates the stator current and rotor flux, can be expressed by

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{V_s} + [\boldsymbol{L}\tilde{\boldsymbol{i}}_s + \boldsymbol{K}\mathrm{sgn}(\tilde{\boldsymbol{i}}_s)], \quad \hat{\boldsymbol{i}}_s = \boldsymbol{C}\hat{\boldsymbol{x}} \tag{5.2}$$

where $\tilde{\boldsymbol{i}}_s = \boldsymbol{i_s} - \hat{\boldsymbol{i}}_s$, and $\wedge$ denotes estimated quantities. The term $\boldsymbol{L}\tilde{\boldsymbol{i}}_s$ describes the Luenberger observer term and the added sign term $\boldsymbol{K}\mathrm{sgn}(\tilde{\boldsymbol{i}}_s)$ represents the sliding mode component in the proposed observer. $\boldsymbol{K}$ and $\boldsymbol{L}$ are sliding mode and Luenberger gain matrices, respectively. Figure 5.8 shows the block diagram for the proposed Luenberger-sliding mode observer. The observer scheme is comprised of parameter estimation and adaptation. The indirect vector controller has been implemented in the rotor flux

reference frame where rotor flux position (θ_{rf}) is used to decouple the stator *d-q* current components. The Luenberger-sliding mode observer and the motor model have been implemented in the stationary reference frame.

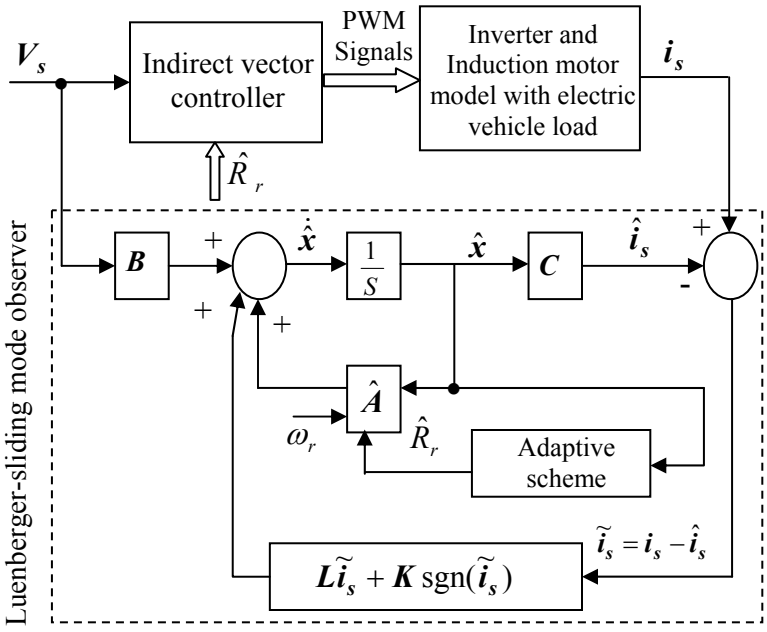


Figure 5.8. Luenberger-sliding mode observer for parameter adaptation.

The Luenberger observer term estimates the stator current and rotor flux, and the sign term guarantees the robustness of observer presented in equation 5.2. For this observer, the problem can be stated as follows: Design $\boldsymbol{L}$ and $\boldsymbol{K}$ such that $\boldsymbol{x} - \hat{\boldsymbol{x}} \to \boldsymbol{0}$ and $\boldsymbol{i_s} - \hat{\boldsymbol{i}}_s \to \boldsymbol{0}$ in the limit of time $t \to \infty$. The dimension of the sliding mode gain matrix $\boldsymbol{K}$ is 4x1; this was determined by trial and error. The observer only with Luenberger gain matrix can be represented as

$$\dot{\hat{\boldsymbol{x}}} = \left(\hat{\boldsymbol{A}} - \boldsymbol{LC}\right)\hat{\boldsymbol{x}} + \boldsymbol{BV}_s + \boldsymbol{LCx}, \quad \hat{\boldsymbol{i}}_s = \boldsymbol{C}\hat{\boldsymbol{x}}.$$

The $\boldsymbol{L}$ (Luenberger) matrix must be set in order to make the error go to zero. This can be achieved by setting up a state equation for the error as follows.

$$\dot{\tilde{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{LC})\tilde{\boldsymbol{x}} \tag{5.3}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$.

Since it is not possible to measure all the states $\boldsymbol{x}$, the observer is built to estimate the states while measuring the output $\boldsymbol{i} = \boldsymbol{Cx}$. The dimension of the $\boldsymbol{L}$ matrix depends on how many states can be measured. The gain matrix $\boldsymbol{L}$ is designed by pole placement. The observability of the system was first checked using the Matlab command **det(obsv(A,C))** before designing the observer gain matrix. A nonzero result shows that the system is observable. If the system has four states variables, four poles need to be placed in different places to design the observer gain matrix $\boldsymbol{L}$. Let the four poles be

P1= -a; P2= -b; P3= -c; P4= -d;

The pole placement can be done using the following Matlab command.

$\boldsymbol{L}$ = **PLACE(** $\boldsymbol{A'}$**,** $\boldsymbol{C'}$**,** [P1 P2 P3 P4]**)**$'$, which computes a state-feedback matrix L such that the Eigen values of $\boldsymbol{A'} - \boldsymbol{L} * \boldsymbol{C'}$ are those specified in vector [P1….P4]. Since the stator current is measured in this case, the dimension of the $\boldsymbol{L}$ matrix is 4x2, which is represented by

$$\boldsymbol{L} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \\ g_5 & g_6 \\ g_7 & g_8 \end{bmatrix}. \tag{5.4}$$

Here g_1 through g_8 are the gains. Solving the error equation 5.3, it can be observed whether the error goes to zero or not. The equation can be verified using the Matlab

command **'lsim'**. The command **'lsim'** simulates the time response of a system with arbitrary inputs and initial conditions. Using this command, it can be checked how fast the error between actual value and estimated value of a state variable goes to zero, whereas the error will be used as an initial condition. The command used for the system [$\boldsymbol{A-L*C}$] is given as follows

$$[\boldsymbol{y, x}] = \text{lsim}(\boldsymbol{A-L*C}, \boldsymbol{Z, C, 0, V_s, t, x0}) \; ; \tag{5.5}$$

Here $\boldsymbol{Z}$ = zero matrix, $\boldsymbol{V_s}$ = input, t = time, $\boldsymbol{x0}$ = initial condition (or error) of the states that are to be estimated. The above command provides a plot showing the time response of the errors.

5.2.2 PARAMETER IDENTIFICATION ALGORITHM

Estimated stator currents and rotor fluxes are used to estimate stator and rotor resistance. The output error dynamics are necessary to find the conditions for estimating rotor and stator resistances. This can be found by subtracting equation 5.2 from equation 5.1.

$$\frac{d}{dt}(\boldsymbol{e}) = \frac{d}{dt}(\boldsymbol{Cx} - \boldsymbol{C\hat{x}}) \tag{5.6}$$

$$\begin{aligned} &= \boldsymbol{C}(\boldsymbol{A} - \boldsymbol{LC})(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \boldsymbol{C}(\boldsymbol{A} - \boldsymbol{LC})\hat{\boldsymbol{x}} - \boldsymbol{C}(\hat{\boldsymbol{A}} - \boldsymbol{LC})\hat{\boldsymbol{x}} - \boldsymbol{CK}\,\text{sgn}(\boldsymbol{e}) \\ &= \boldsymbol{C}(\boldsymbol{A} - \boldsymbol{LC})\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{e} - \boldsymbol{C\Delta A\hat{x}} - \boldsymbol{CK}\,\text{sgn}(\boldsymbol{e}) \end{aligned}$$

where $\Delta\boldsymbol{A} = \hat{\boldsymbol{A}} - \boldsymbol{A}$.

Now for rotor resistance estimation

$$\hat{\boldsymbol{A}} = \begin{bmatrix} -\frac{1}{\sigma}\left(\frac{R_s}{L_s} + \frac{\hat{R}_r(1-\sigma)}{L_r}\right) & 0 & \frac{L_m \hat{R}_r}{\sigma L_s L_r^2} & \frac{L_m \omega_r}{\sigma L_s L_r} \\ 0 & -\frac{1}{\sigma}\left(\frac{R_s}{L_s} + \frac{\hat{R}_r(1-\sigma)}{L_r}\right) & -\frac{L_m \omega_r}{\sigma L_s L_r} & \frac{L_m \hat{R}_r}{\sigma L_s L_r^2} \\ \frac{L_m \hat{R}_r}{L_r} & 0 & -\frac{\hat{R}_r}{L_r} & -\omega_r \\ 0 & \frac{L_m \hat{R}_r}{L_r} & \omega_r & -\frac{\hat{R}_r}{L_r} \end{bmatrix}$$

where $\hat{R}_r$ is the estimated rotor resistance,

and

$$\Delta\boldsymbol{A} = \begin{bmatrix} -\frac{1-\sigma}{\sigma}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 \\ 0 & -\frac{1-\sigma}{\sigma}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) \\ L_m\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 & -\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 \\ 0 & L_m\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) & 0 & -\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right) \end{bmatrix}.$$

Lyapunov's stability criterion has been used to determine the error dynamics, which gives sufficient condition for asymptotic stability of a non-linear system. The following Lyapunov's function is introduced to estimate the rotor resistance.

$$V = \boldsymbol{e}^T \boldsymbol{e} + \frac{\left(\hat{R}_r - R_r\right)^2}{a} \tag{5.7}$$

where a is a positive constant, and $\boldsymbol{e} = \begin{bmatrix} e_{ids} & e_{iqs} \end{bmatrix}^T$, $e_{ids} = i_{ds} - \hat{i}_{ds}$ and $e_{iqs} = i_{qs} - \hat{i}_{qs}$. This function is zero when the error e is zero and estimated rotor resistance is equal to the actual rotor resistance. The time derivative of V is

$$\frac{dV}{dt} = \boldsymbol{e}\left[\frac{d\left(\boldsymbol{e}^T\right)}{dt}\right] + \boldsymbol{e}^T\left[\frac{d\boldsymbol{e}}{dt}\right] + \frac{2\left(\hat{R}_r - R_r\right)}{a}\frac{d\hat{R}_r}{dt} \tag{5.8}$$

$$= \boldsymbol{e}^T\left[\left\{\boldsymbol{C}\left(\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C}\right)\boldsymbol{C}^T\right\}^T + \left\{\boldsymbol{C}\left(\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C}\right)\boldsymbol{C}^T\right\}\right]\boldsymbol{e} - \boldsymbol{K}^T\boldsymbol{C}^T\boldsymbol{e}.\mathrm{sgn}\left(\boldsymbol{e}\right) - \boldsymbol{e}^T\boldsymbol{C}\boldsymbol{K}\,\mathrm{sgn}\left(\boldsymbol{e}\right)$$

$$-\left(\hat{\boldsymbol{x}}^T\Delta\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{e} + \boldsymbol{e}^T\boldsymbol{C}\Delta\boldsymbol{A}\hat{\boldsymbol{x}}\right) + \frac{2\left(\hat{R}_r - R_r\right)}{a}\frac{d\hat{R}_r}{dt}.$$

Now

$$\hat{\boldsymbol{x}}^T\Delta\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{e} = \begin{bmatrix}\hat{i}_{ds} & \hat{i}_{qs} & \hat{\lambda}_{dr} & \hat{\lambda}_{qr}\end{bmatrix}\begin{bmatrix} -\frac{1-\sigma}{\sigma}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right)e_{ids} \\ -\frac{1-\sigma}{\sigma}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right)e_{iqs} \\ -\frac{L_m}{\sigma L_s L_r}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right)e_{ids} \\ -\frac{L_m}{\sigma L_s L_r}\left(\frac{\hat{R}_r}{L_r} - \frac{R_r}{L_r}\right)e_{iqs}\end{bmatrix} \tag{5.9}$$

$$= \frac{L_m\left(\hat{R}_r - R_r\right)}{L_r\left(L_s L_r - L_m^2\right)}e_{ids}\left(\hat{\lambda}_{dr} - L_m\hat{i}_{ds}\right) + \frac{L_m\left(\hat{R}_r - R_r\right)}{L_r\left(L_s L_r - L_m^2\right)}e_{iqs}\left(\hat{\lambda}_{qr} - L_m\hat{i}_{qs}\right)$$

where $\frac{1-\sigma}{\sigma} = \frac{L_m^2}{L_s L_r - L_m^2}$.

Similarly,

$$\boldsymbol{e}^T\boldsymbol{C}\Delta\boldsymbol{A}\hat{\boldsymbol{x}} = \frac{L_m\left(\hat{R}_r - R_r\right)}{L_r\left(L_s L_r - L_m^2\right)}e_{ids}\left(\hat{\lambda}_{dr} - L_m\hat{i}_{ds}\right) + \frac{L_m\left(\hat{R}_r - R_r\right)}{L_r\left(L_s L_r - L_m^2\right)}e_{iqs}\left(\hat{\lambda}_{qr} - L_m\hat{i}_{qs}\right). \tag{5.10}$$

Using equations 5.8, 5.9 and 5.10, the derivative of Lyapunov's function can be represented as

$$\frac{dV}{dt} = \left[\boldsymbol{e}^T\left[\left\{\boldsymbol{C}(\boldsymbol{A}-\boldsymbol{LC})\boldsymbol{C}^T\right\}^T + \left\{\boldsymbol{C}(\boldsymbol{A}-\boldsymbol{LC})\boldsymbol{C}^T\right\}\right]\boldsymbol{e} - \boldsymbol{K}^T\boldsymbol{C}^T\boldsymbol{e}.\mathrm{sgn}(\boldsymbol{e}) - \boldsymbol{e}^T\boldsymbol{CK}\,\mathrm{sgn}(\boldsymbol{e})\right]$$
$$-\frac{2L_m\left(\hat{R}_r - R_r\right)}{L_r\left(L_sL_r - L_m^2\right)}\left\{e_{ids}\left(\hat{\lambda}_{dr} - L_m\hat{i}_{ds}\right) + e_{iqs}\left(\hat{\lambda}_{qr} - L_m\hat{i}_{qs}\right)\right\} + \frac{2\left(\hat{R}_r - R_r\right)}{a}\frac{d\hat{R}_r}{dt}. \tag{5.11}$$

A sufficient condition for asymptotic stability is that V has to decrease when error is not zero, the derivative of Lyapunov's function is negative definite. Since the first term in equation 5.10 is always negative, the above condition is satisfied if the sum of the last two terms is zero. Therefore, the equation for identification of the rotor resistance is obtained as

$$\frac{d\hat{R}_r}{dt} = k_r\left[e_{ids}\left(\hat{\lambda}_{dr} - L_m\hat{i}_{ds}\right) + e_{iqs}\left(\hat{\lambda}_{qr} - L_m\hat{i}_{qs}\right)\right] \tag{5.12}$$

where $k_r = \dfrac{L_m}{L_r\left(L_sL_r - L_m^2\right)}a$. Similarly, using Lyapunov's function, the equation for stator resistance identification can be derived as

$$\frac{d\hat{R}_s}{dt} = -k_s\left(e_{ids}\hat{i}_{ds} + e_{iqs}\hat{i}_{qs}\right) \tag{5.13}$$

where $k_s = \dfrac{a}{\sigma L_s}$.

The proposed approach is a novel method of combining the simple structure of a Luenberger observer and the robustness of a sliding mode observer for parameter estimation and on-line adaptation in motor drives. The method has been verified by simulation and experiment.

5.3 MODEL BASED ANALYSIS AND SIMULATION

The developed Luenberger-sliding mode observer with parameter estimation and on-line adaptation algorithms has been verified through software-in-the-loop (SIL) simulation as well as hardware-in-the-loop (HIL) simulation before experimentation. SIL and HIL have been carried out for two induction machines, one much smaller in size than the other. The smaller machine has been utilized for the proof-of-concept simulation and experiments, while the larger machine has been used for simulations only. Software-in-the-loop simulation results are presented in this chapter, and HIL simulation results followed by experimental results will be presented in the next chapter.

5.3.1 SOFTWARE-IN-THE-LOOP SIMULATION

The software-in-the-loop (SIL) simulation is only numerical solution, and there is no real hardware. The controller model and machine model have been implemented in Matlab-Simulink. The Luenberger-sliding mode observer with on-line adaptation algorithm has been verified through SIL simulation using a small induction motor as well as a high power induction motor that is used in EV/HEV propulsion system. The on-line adaptation was performed by feeding back the estimated rotor resistance to the controller model from the observer and parameter estimator simulation block while the induction machine model is running. The simulation block diagram is shown in Figure 5.9.

The simulation has been carried out for the small machine using Luenberger observer and the proposed Luenberger-sliding mode observer separately to compare the

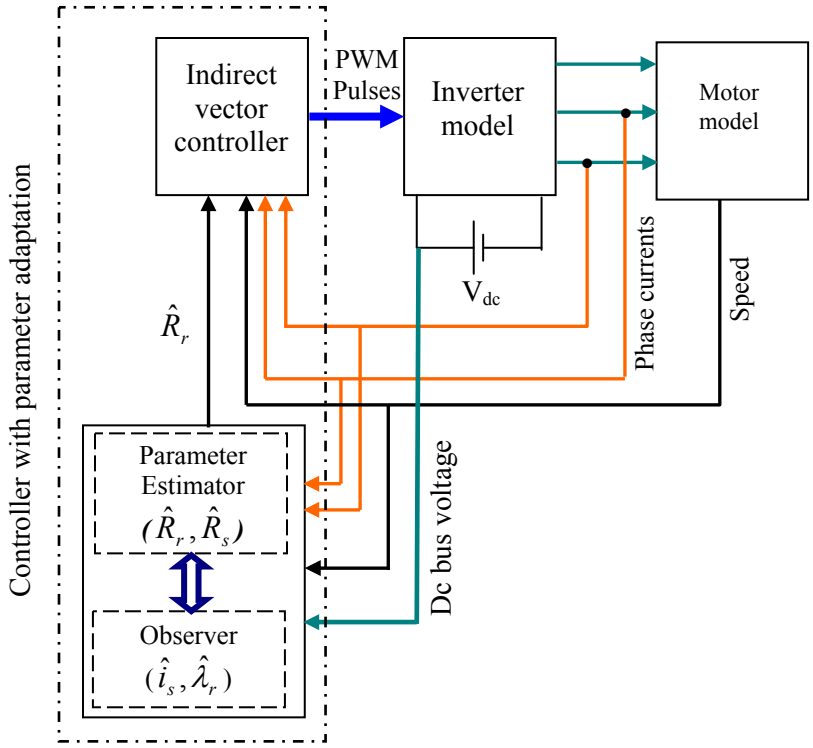


Figure 5.9. Simulation block diagram for induction motor drive with on-line parameter adaptation.

estimation performances between the two observers. The ratings of the small induction machine are: 0.5hp, 1A rms (1.414A max), 1740rpm, 60Hz and 4 poles. The machine parameters are: stator resistance (R_s) 14.6Ω, rotor resistance (R_r) 12.77Ω, stator inductance 22.2mH, rotor inductance 51.8mH, mutual inductance 296.3mH, and the moment of inertia 2.79e-4 kg-m^2. Simulation has been carried out for 30% and 50% increase of rotor resistance in motor model for an inverter switching frequency of 20kHz. Figures 5.10 and 5.11 show rotor and stator resistance estimation, respectively, using only Luenberger observer for 30% increase (R_r = 1.3*12.77Ω = 16.6Ω) in rotor resistance. From Figure 5.10, it is observed that rotor resistance converges to a value of 15.9Ω with a tracking error of 4.22%. For stator resistance estimation, the initial value

was set to 9Ω. From Figure 5.11, it is seen that stator resistance reaches a value of 14.1Ω (R_s = 14.6Ω) but does not converge to any resistance value in 5secs. Figures 5.12 and

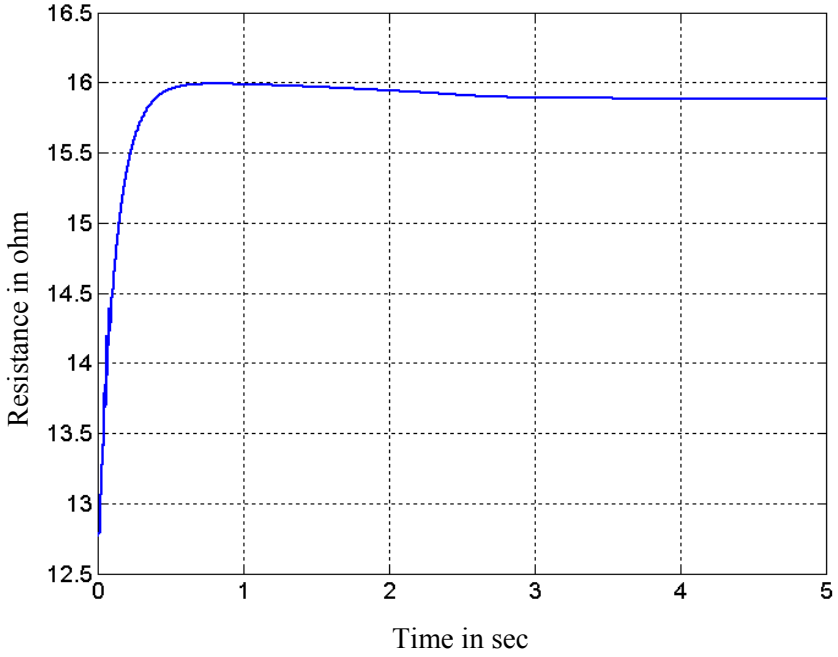


Figure 5.10. Rotor resistance estimation with 30% rotor resistance increase using only Luenberger observer.

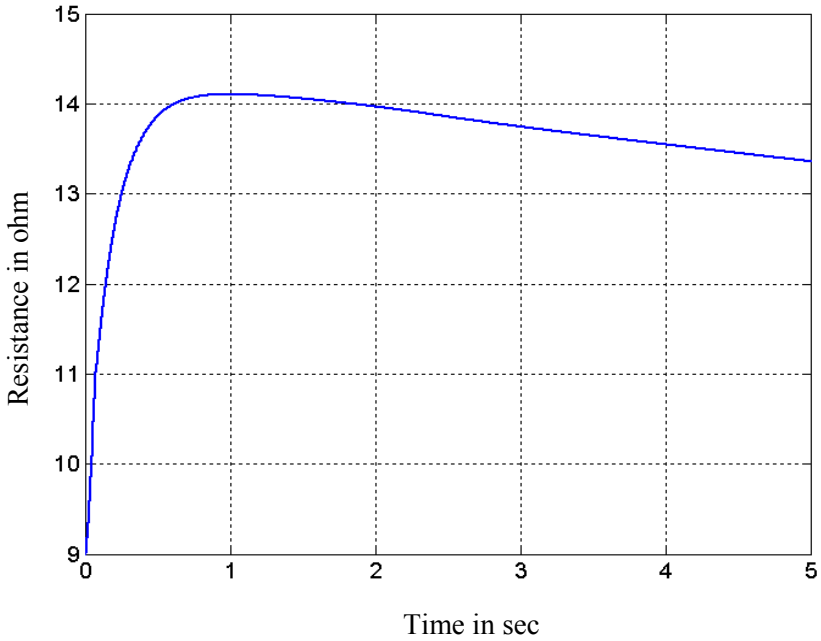


Figure 5.11. Stator resistance estimation with 30% rotor resistance increase using only Luenberger observer.

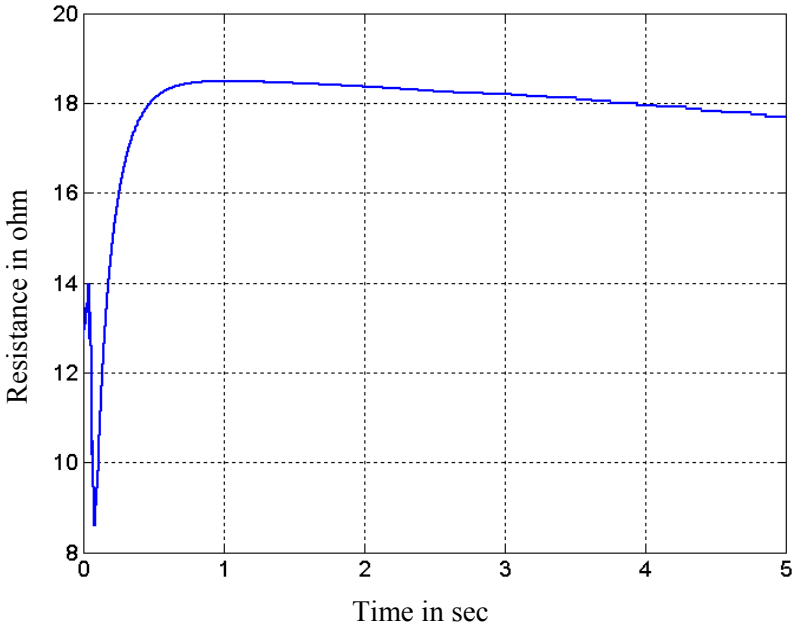


Figure 5.12. Rotor resistance estimation with 50% rotor resistance increase using only Luenberger observer.

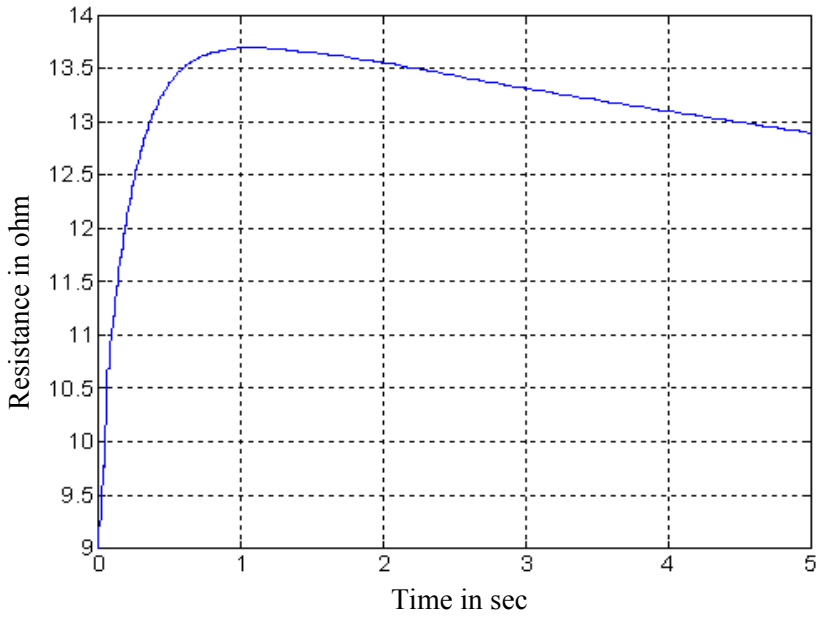


Figure 5.13 Stator resistance estimation with 50% rotor resistance increase using only Luenberger observer.

5.13 show the rotor and stator resistance estimation, respectively using only a Luenberger observer for 50% increase (R_r =1.5*12.77Ω = 19.155Ω) in rotor resistance. From Figures

5.12 and 5.13, it is observed that rotor resistance reaches to a value of 18.5Ω but does not converge in 5secs, and the stator resistance reaches a value of 13.65Ω (R_s = 14.6Ω) but does not converge at all. From the simulation, it can be observed that the Luenberger observer does not function properly in the presence of large parameter variations.

The simulation was also carried out using the Luenberger-sliding mode observer for the same small induction motor. Simulation was carried out for 30% and 50% increase of rotor resistance in the motor model. Figures 5.14 and 5.15 show rotor and stator resistance estimation, respectively, using the Luenberger-sliding mode observer for 30% (R_r = 1.3*12.77Ω = 16.6Ω) increase in rotor resistance. From Figure 5.14, it is observed that estimated rotor resistance reaches the value of 16.55Ω within 2.5secs and converges to 16.65Ω within 3.5secs with an error of 0.295%. For stator resistance (R_s = 14.6Ω) estimation, the initial value was set to the value of 9Ω. From Figure 5.15, it is observed that estimated stator resistance converges to a value of 14.5Ω within 3secs with

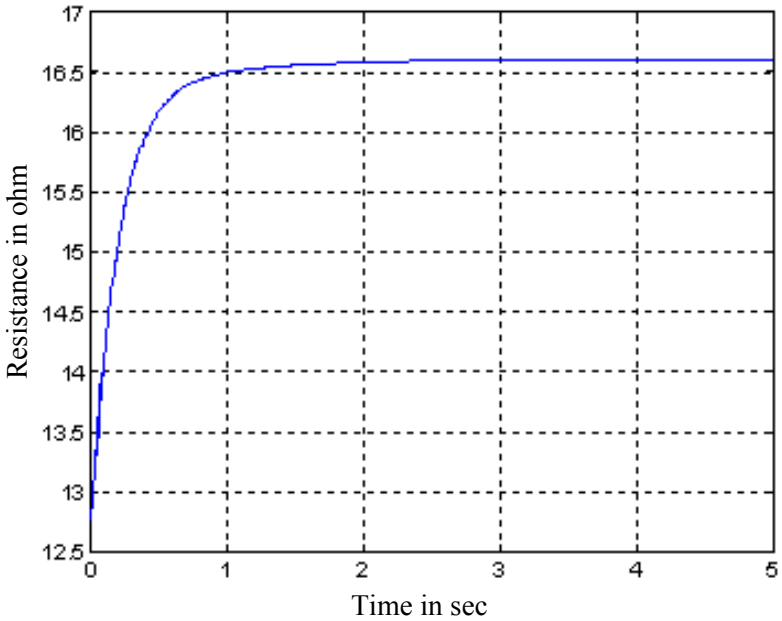


Figure 5.14. Rotor resistance estimation with 30% rotor resistance increase using the Luenberger-sliding mode observer

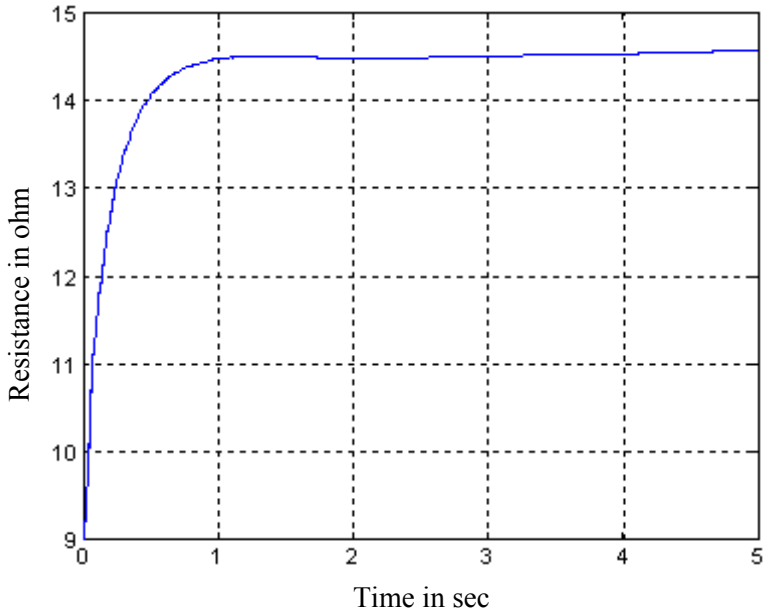


Figure 5.15. Stator resistance estimation with 30% rotor resistance increase using the Luenberger-sliding mode observer.

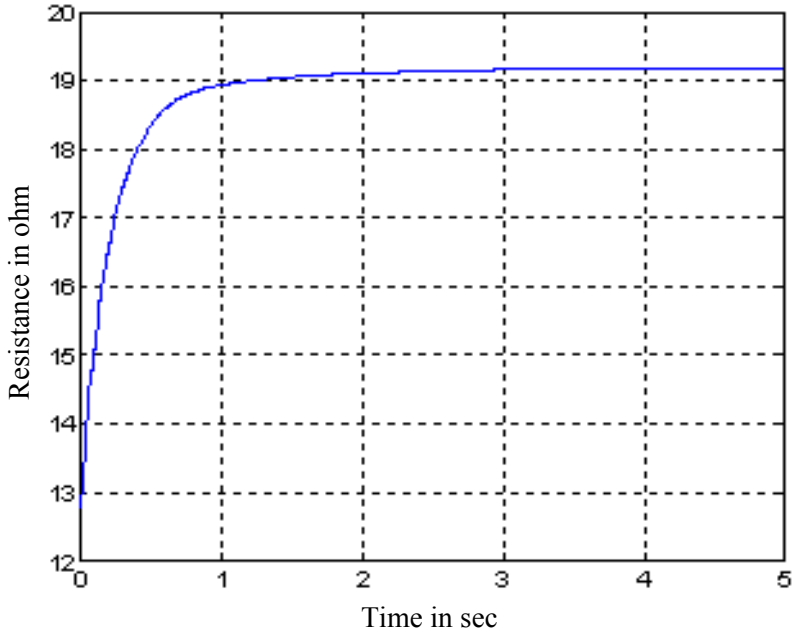


Figure 5.16. Rotor resistance estimation with 50% rotor resistance increase using the Luenberger-sliding mode observer.

an error of 0.685%. Figures 5.16 and 5.17 show rotor and stator resistance estimation, respectively using the Luenberger-sliding mode observer for 50% (R_r = 1.5*12.77Ω =

19.155Ω) increase in rotor resistance. From Figure 5.16, it is observed that estimated rotor resistance reaches the value of 19.1Ω within 2.5secs and converges to 19.2Ω within 3.5secs with a tracking error of 0.235%. From Figure 5.17, it is seen that estimated stator resistance converges to a value of 14.65Ω within 3secs with an error of 0.342%. The simulation results show that the Luenberger-sliding mode observer with estimation algorithm shows excellent accuracy despite large parameter variations and fast convergence compared to other estimation techniques mentioned in Chapter III.

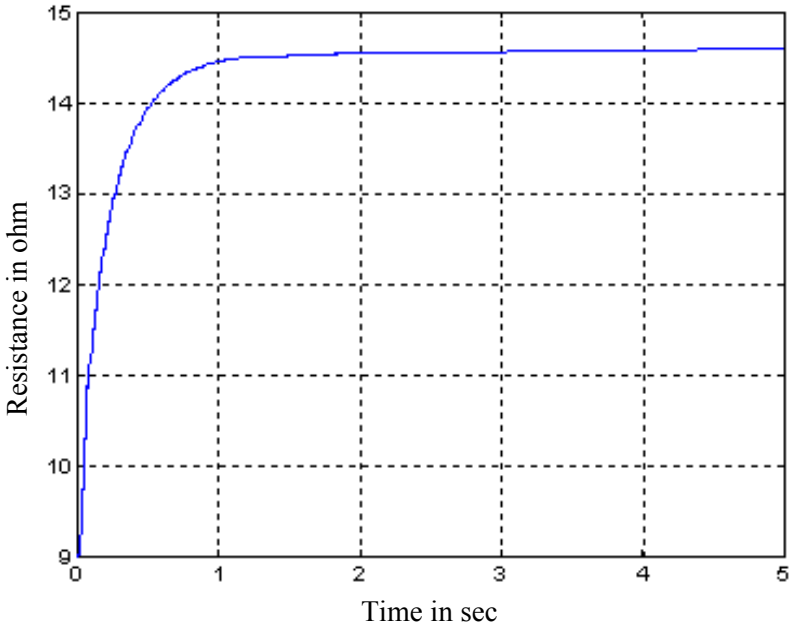


Figure 5.17. Stator resistance estimation with 50% rotor resistance increase using the Luenberger-sliding mode observer.

The simulation has also been carried out for a high power induction motor with a vehicle load to evaluate and demonstrate the performance of the controller with the new on-line parameter estimation algorithm. The Ballard induction motor used in the University of Akron HEV has not been used for this purpose because the parameters are

unavailable. Also, it is not possible to test on the Ballard motor since it is packaged, and advanced concept testing is not possible due to Challenge X competition constraints. A 30kW, 280V, 4-pole induction machine and a small sedan type electric vehicle of 1000kg have been considered for the simulation. One of the objectives of this simulation is to simulate the detailed dynamics of the induction motor.

The specifications of the electric vehicle (EV) load and the induction motor used in the simulation are given in Table 5.1. The velocity profile of the electric vehicle used in simulation is the SAE urban driving schedule B (maximum speed 20mph, acceleration time 19sec and cruise time 19sec) [60]. The load for an electric vehicle known as road load consists of gravitational force, rolling resistance of the tires, and the aerodynamic drag force [2]. The road load force F_{RL} can be represented as

$$F_{RL} = F_{gxT} + F_{roll} + F_{AD} \tag{5.14}$$

Table 5.1. Specifications of EV load and induction motor

Vehicle speed	20mph	Induction motor	30KW, 280V(L-L), 4 pole, 60Hz
Vehicle mass	1000kg	Rotor resistance, inductance	0.228Ω, 0.8mH
Rolling resistance coefficients C_0, C_1	0.009, 1.7E-6	*Stator resistance, inductance*	0.087Ω, 0.8mH
Aerodynamic drag coefficient C_D	0.2	Mutual inductance	34.7mH
Vehicle frontal area A_F	$2m^2$	Moment of inertia	$1.662kg\text{-}m^2$
Transmission gear ratio	3.2		
Wheel radius r_{wh}	0.28m		

where F_{gxT}, F_{roll} and F_{AD} are gravitational force, rolling resistance force and aerodynamic force, respectively. The tractive force F_{TR} provided by the propulsion unit of the vehicle must overcome the road load force F_{RL} to propel the vehicle at a desired velocity. The gravitational force depends on the slope of the roadway, and can be represented as [2]

$$F_{gxT} = mg\sin\beta \tag{5.15}$$

where m is the total mass of the vehicle, g is the gravitational acceleration constant and β is the grade angle with respect to horizon. The rolling resistance force is given by [2]

$$F_{roll} = \begin{cases} \operatorname{sgn}[v_{xT}]mg\cos\beta\left(C_0 + C_1 v_{xT}^2\right) & if\ v_{xT} \neq 0 \\ \left(F_{TR} - F_{gxT}\right) & if\ v_{xT} = 0\ and\ \left|F_{TR} - F_{gxT}\right| \leq C_0 mg \\ \operatorname{sgn}\left[F_{TR} - F_{gxT}\right]\left(C_0 mg\right) & if\ v_{xT} = 0\ and\ \left|F_{TR} - F_{gxT}\right| > C_0 mg \end{cases} \tag{5.16}$$

where v_{xT} is in m/sec. The aerodynamic drag force is the viscous resistance of the air working against the motion of the vehicle. The force is given by

$$F_{AD} = \operatorname{sgn}[v_{xT}]\left\{0.5\rho C_D A_F \left(v_{xT} + v_0\right)^2\right\} \tag{5.17}$$

where ρ is the air density in kg/m^3, C_D is the aerodynamic drag coefficient. A_F is the equivalent frontal area of the vehicle, and v_0 is the head-wind velocity. The electric vehicle torque T_{EV} can be obtained from the tractive force F_{TR} and wheel radius r_{wh} of the vehicle as

$$T_{EV} = F_{TR} \cdot r_{wh} \tag{5.18}$$

The torque of the induction motor together with the electric vehicle load can be written as

$$T_e = J\frac{dw_m}{dt} + Bw_m + T_{EV} \tag{5.19}$$

where w_m is the angular velocity of the rotor in rad/sec, J is the moment of inertia of the

system, and B is the viscous coefficient of the motor. In the simulation the viscous coefficient B has been considered zero.

Simulation has been carried out for 50% and 30% increased rotor resistance in motor model for an inverter switching frequency of 20kHz. The simulation results simultaneously show the performance of the new Luenberger-Sliding Mode observer for the parameter estimation using the adaptation law and state estimation of a high power induction motor. Figures 5.18 and 5.19 show rotor and stator resistance estimation, respectively, for a 50% increase (R_r = 0.228*1.5Ω = 0.342Ω) in rotor resistance. From Figure 5.18, it is observed that estimated rotor resistance reaches the value of 0.34Ω (0.586% error) within 3sec, and converges to 0.341Ω (0.3% error) within 4.2sec. For stator resistance (R_s = 0.087Ω) estimation, the initial value was set to 0.05Ω. From Figure 5.19, it is seen that estimated stator resistance converges to the value of 0.0881Ω within

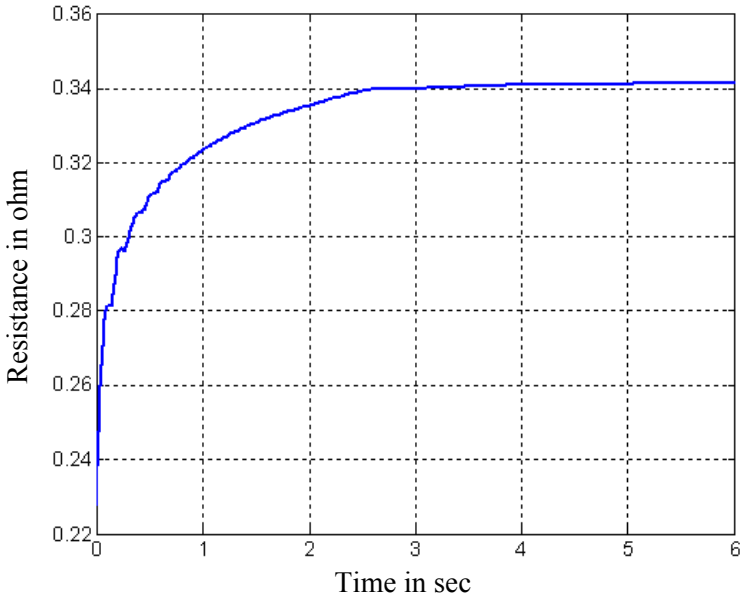


Figure 5.18. Rotor resistance estimation with 50% rotor resistance increase in high power induction motor model with electric vehicle load.

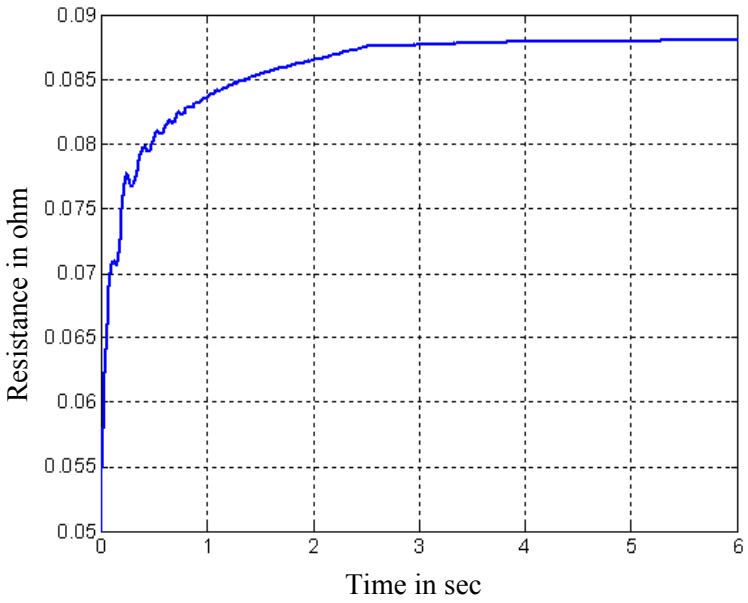


Figure 5.19. Stator resistance estimation with 50% rotor resistance increase in high power induction motor model with electric vehicle load

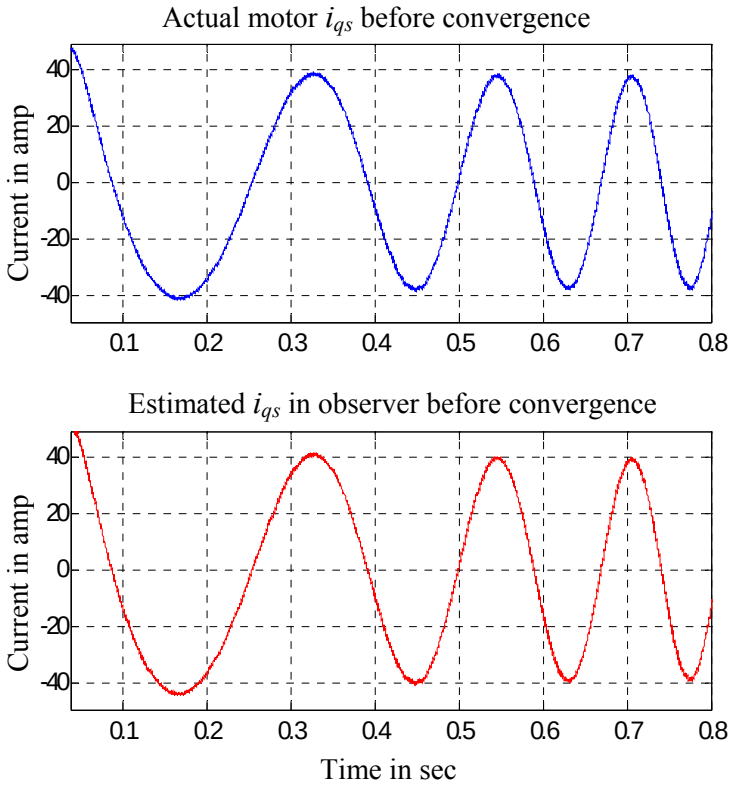


Figure 5.20. i_{qs} current profiles before rotor resistance converges.

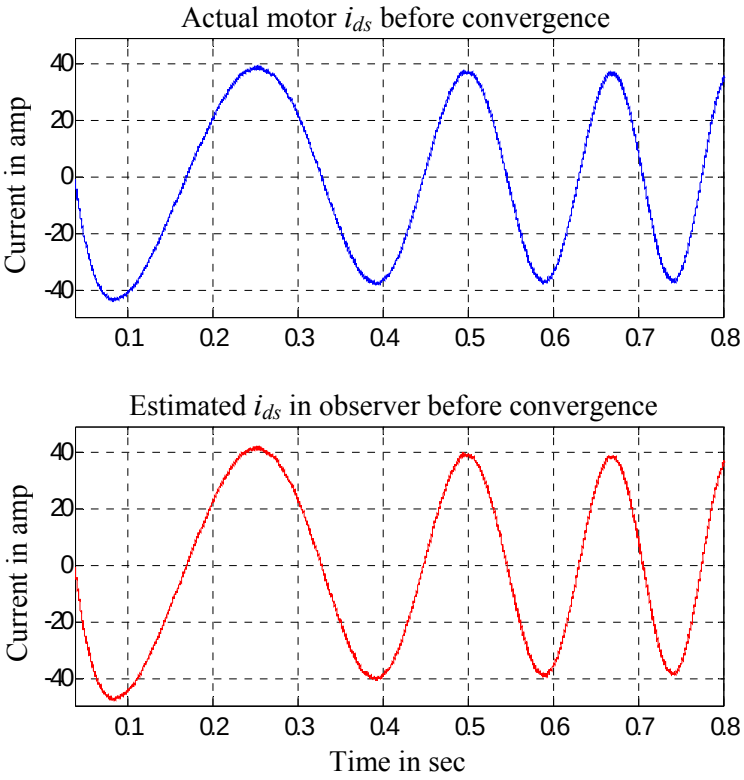


Figure 5.21. i_{ds} current profiles before rotor resistance converges.

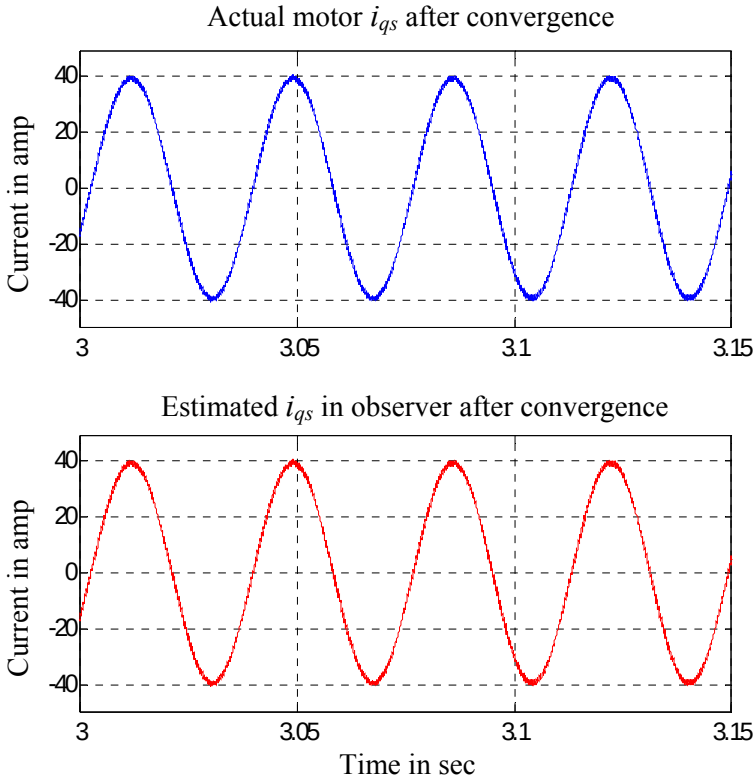


Figure 5.22. i_{qs} current profiles after rotor resistance converges.

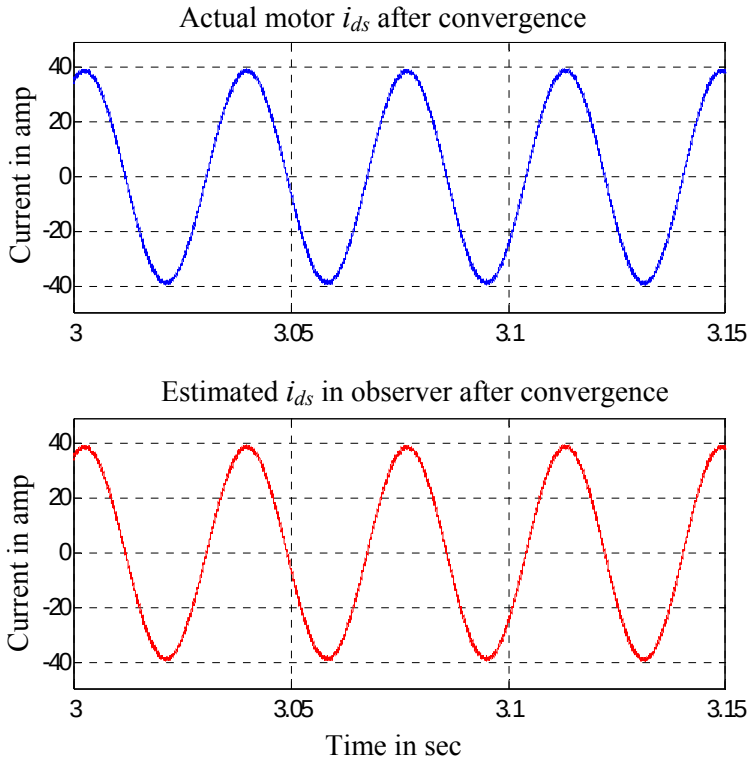


Figure 5.23. i_{ds} current profiles after rotor resistance converges.

4.5sec with a tracking error of 1.26%. The key to parameter adaptation is the correct estimation of the rotor flux λ_{qdr} in the stationary reference frame by the observer, which is driven by the current error. Figures 5.20 and 5.21 show stator *d-q* axis current profiles (i_{qs}, i_{ds}) before the convergence of the estimated rotor resistance for 50% rotor resistance increase in the motor model. From these figures, it is observed that the estimated stator *d-q* axis currents are bigger than the motor stator *d-q* currents, and decrease towards the motor stator currents with time as the estimated rotor resistance reaches close to the actual value of rotor resistance set in the motor model. Figures 5.22 and 5.23 show i_{qs} and i_{ds} profiles after the convergence of the estimated rotor resistance, respectively. The estimated stator *d-q* currents reach the same values of actual motor currents after the rotor

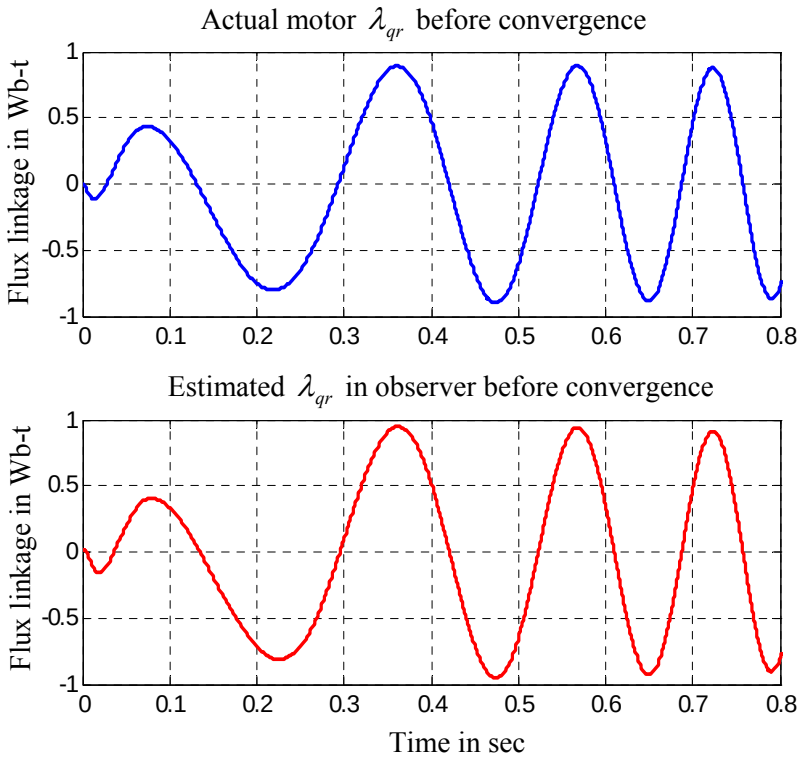


Figure 5.24. λ_{qr} flux profiles before rotor resistance converges.

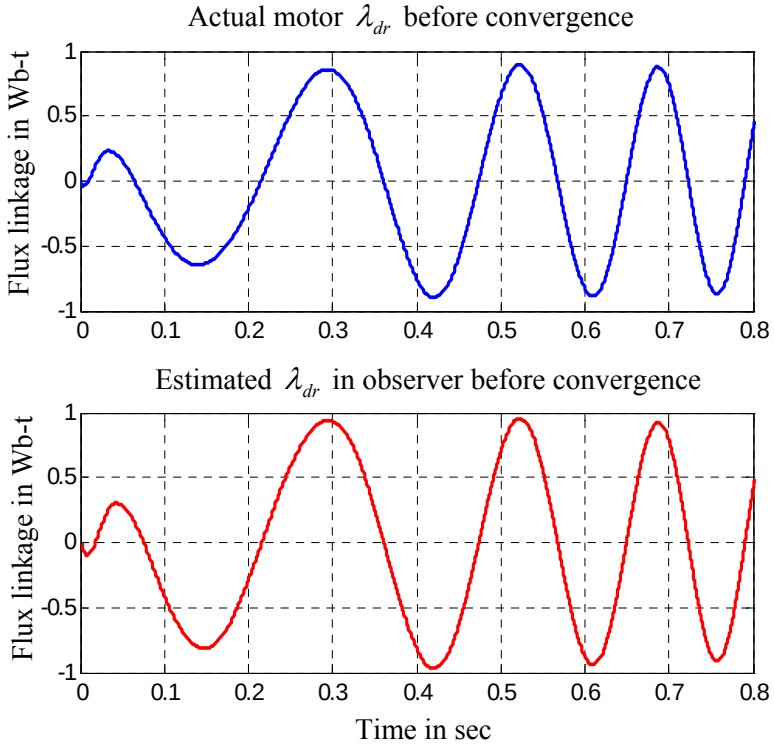


Figure 5.25. λ_{dr} flux profiles before rotor resistance converges.

resistance converges. Figures 5.24 and 5.25 show rotor *d-q* fluxes ($\lambda_{qr}, \lambda_{dr}$) before the convergence of the estimated rotor resistance. Figures 5.26 and 5.27 show λ_{qr} and λ_{dr} profiles, respectively, after the convergence of the estimated rotor resistance. From these figures it is observed that the estimated rotor fluxes reach the same values of actual rotor fluxes after the rotor resistance converges.

Figures 5.28 and 5.29 show rotor and stator resistance estimation, respectively, for 30% increase (R_r = 0.228*1.3Ω≈0.296Ω) in rotor resistance. From Figure 5.28, it is observed that estimated rotor resistance reaches the value of 0.295Ω (0.34% error) within 3sec, and converges to 0.296Ω (≈0% error) within 4.5sec. From Figure 5.29, it is also observed that estimated stator resistance converges to the value of 0.0882Ω within 4.5sec with a tracking error of 1.38%. The simulation results show that the developed

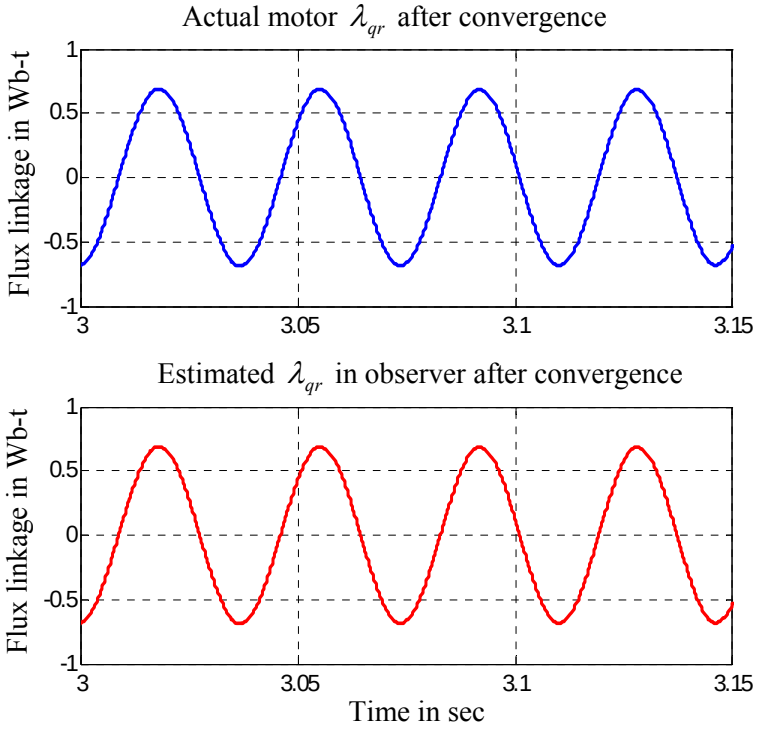


Figure 5.26. λ_{qr} flux profiles after rotor resistance converges.

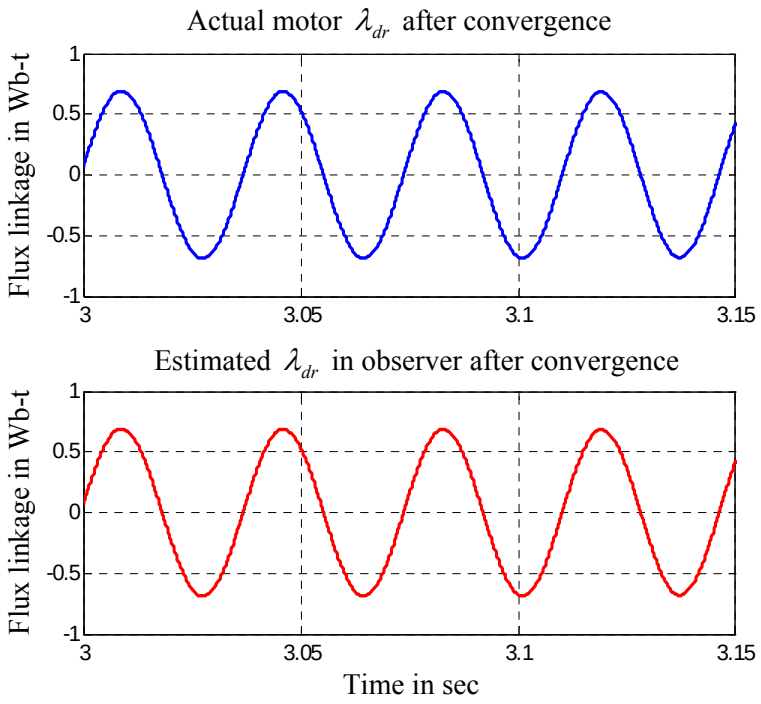


Figure 5.27. λ_{dr} flux profiles after rotor resistance converges.

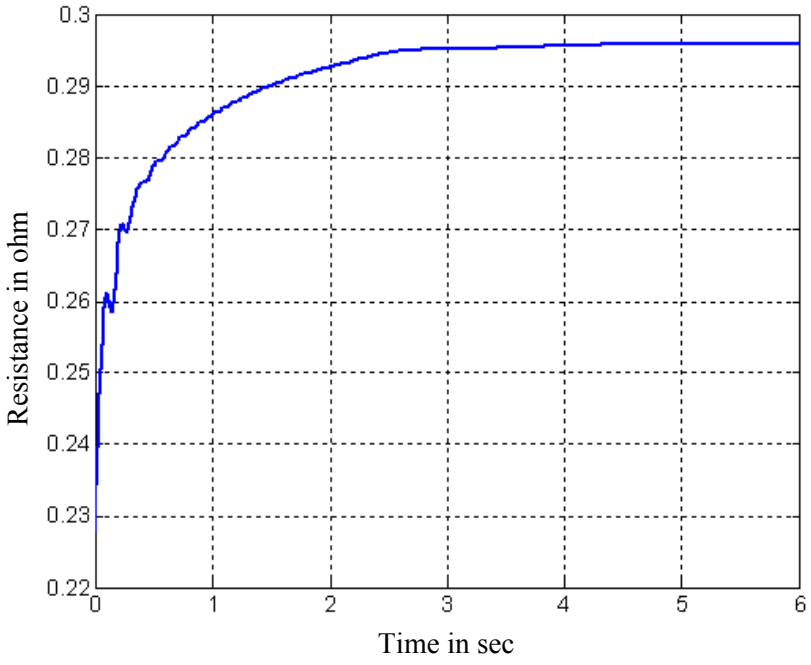


Figure 5.28. Rotor resistance estimation with 30% rotor resistance increase in high power induction motor model with electric vehicle load.

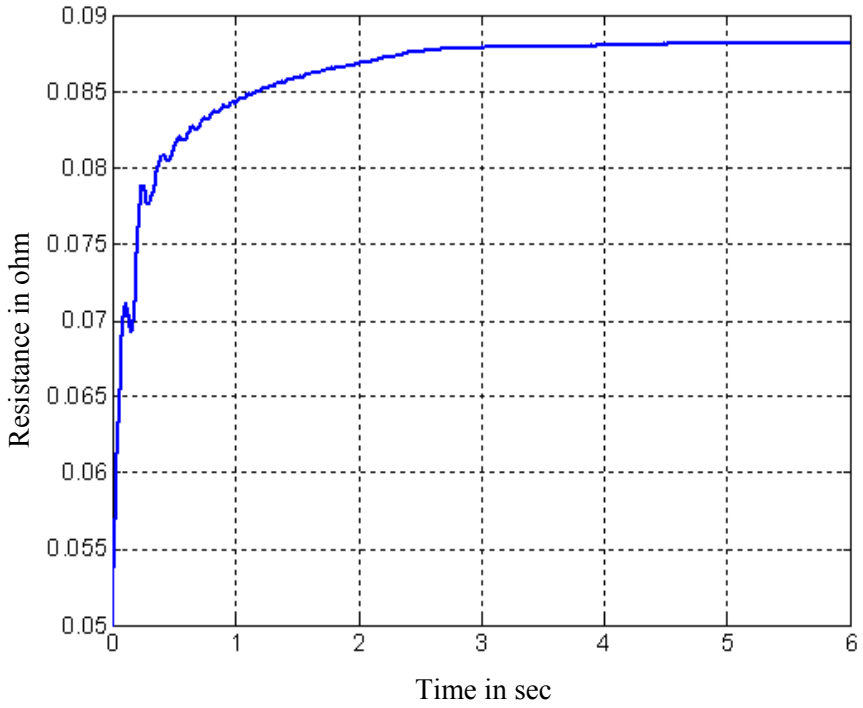


Figure 5.29. Stator resistance estimation with 30% rotor resistance increase in high power induction motor model with electric vehicle load.

Luenberger-sliding mode observer with estimation algorithm also shows excellent accuracy and fast convergence with less computation for a high power induction motor drive used for an electric vehicle load.

5.4 CONCLUSION

This chapter is dedicated to advanced controller design for an induction motor drive with on-line parameter estimation algorithm, and SIL simulation to validate the design and algorithm. At first, the simulation was carried out using Luenberger observer and the developed Luenberger-sliding mode observer for parameter estimation of a small machine. It is found that the Luenberger observer alone does not show satisfactory

results, whereas the Luenberger-sliding mode observer shows excellent accuracy and fast convergence in the presence of large parameter variation. The new observer with parameter estimation algorithm also shows excellent results for a higher power induction motor drive used for an electric vehicle. The HIL simulation and experimental results will be presented in the next chapter.

CHAPTER VI

HIL SIMULATION AND EXPERIMENTS OF INDUCTION MOTOR DRIVE

Hardware-in-the-loop simulation for the new observer with parameter estimation and adaptation algorithm has been carried out for both the small induction motor and the high power induction motor model. The simulation and experimental results will be presented after an introduction to the HIL set-up.

6.1 HARDWARE-IN-THE-LOOP (HIL) SIMULATION

Hardware-in-the-loop (HIL) simulation connects a simulated system with physical components operating in the real world. Simulation can be blended with physical hardware at different stages of the development process. In this case, a real motor controller (hardware) is used in the loop. Because of the real controller, the virtual motor (plant) model must run in real time. In HIL simulation, the dynamic behavior of the plant will be very close to a real plant, with an accurate plant model. On the other hand, SIL simulation is only numerical simulation, and there is no real controller. The simulation is that of a dynamic model of the plant and control law program. Figure 6.1 shows the time stamps in HIL and SIL simulation, where $\mathbf{t_n}$ and $\mathbf{t_{n+1}}$ are the data posting instants. SIL simulation may run slower or faster than the real time clock. Dynamic behavior of the plant and the controller could be far different from that of a real plant. Integrating a

system that has been developed with only SIL results could be dangerous or damaging to the system. Prototyping and repeated testing will be necessary.

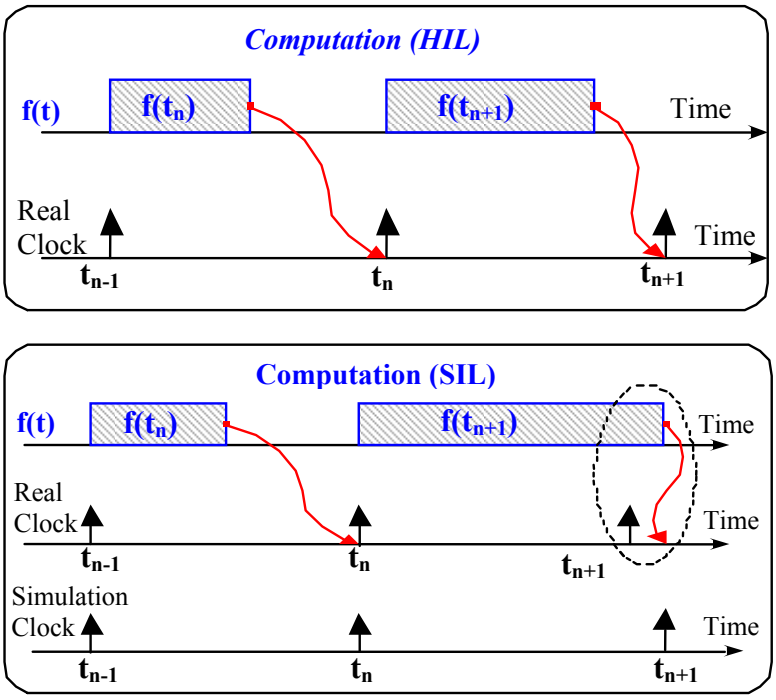


Figure 6.1. Data posting in HIL and SIL simulation.

The advantages that make the HIL simulation popular are: (1) Better controller parameter tuning with an accurate plant model, which reduces prototyping time, (2) reduction in product development time and cost, (3) expansion in the envelope of test conditions including those that are difficult and dangerous to achieve, (4) ease of testing "what-if" scenarios of the system that are difficult, unsafe, time consuming or expensive to carry out in a prototype system and (5) parallel development of mechanical, electrical and control systems. Figure 6.2 shows the HIL setup for the induction motor controller with on-line parameter adaptation. The controller has been implemented in Target-I and the motor model with inverter and load has been implemented in Target-II. The real time

libraries (RT-LAB) based on Simulink are used to simulate the models in real time kernel, and to interface hardware controller with virtual model as well as with real environment. The real time kernel is based on Linux. The real controller has been implemented using an Opal-RT FPGA based RT-LAB HIL box.

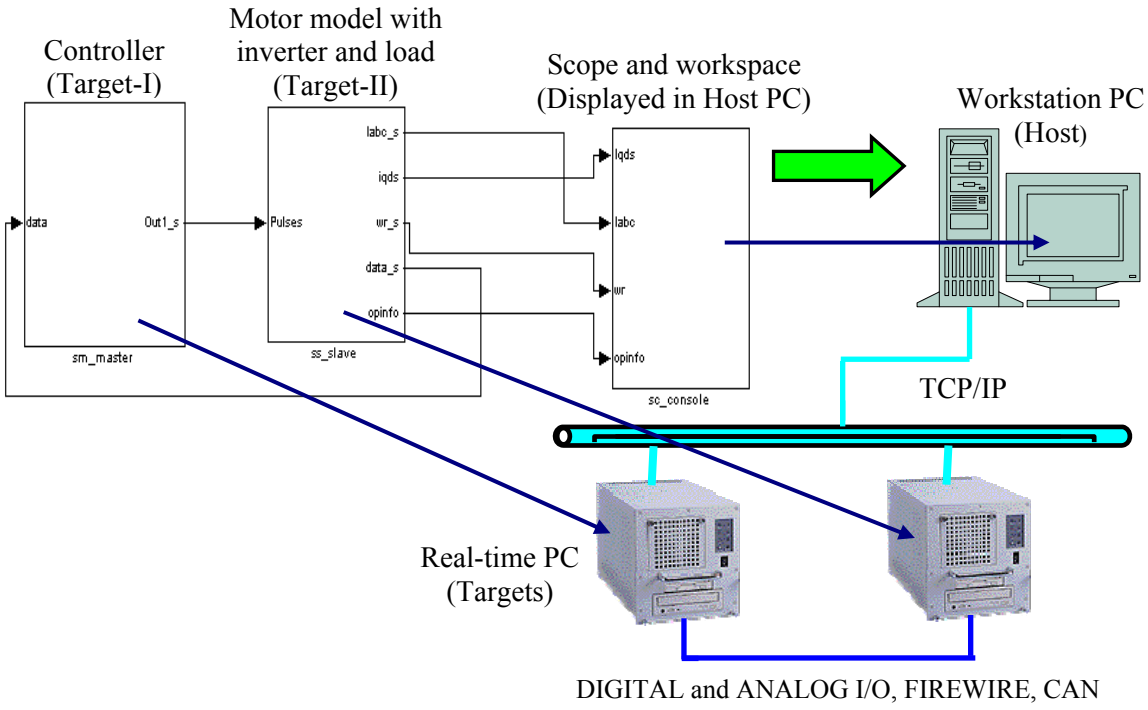


Figure 6.2. HIL simulation setup for a motor control model.

Communication between workstation PC (host) and target PCs is through *Transmission Control Protocol* (TCP)*/Internet Protocol* (IP), and communication between targets can be through FIREWIRE, CAN or FPGA based digital and analog I/O card. In the HIL setup developed for this research, communication between the targets has been established through FIREWIRE. The motor/controller model is built in workstation PC and downloaded to the target PCs through Internet protocol Ethernet.

The developed Luenberger-sliding mode observer with parameter adaptation algorithm has been verified for a small induction motor as well as for a high power induction motor with an electric vehicle load by HIL simulation. HIL simulation also helped in controller parameter tuning. The switching frequency for the inverter model was selected to be 20kHz. The HIL simulation showed that one loop computation time for the controller without the Luenberger-sliding mode observer and parameter adaptation algorithm is 25μsec, and the loop-computation time with the Luenberger-sliding mode observer and parameter adaptation algorithm is 40μsec. The total loop-computation time with 60% increase due to parameter adaptation is still within the PWM cycle time of 50μsec. The computation time of the Luenberger-sliding mode observer with parameter adaptation algorithm is only 15μsec with a processor speed of 266MHz in the HIL simulation. It has been reported that the computation time of the Extended Kalman Filter (EKF) is 130μsec with a processor speed of 60MHz [40]. The computation time of 130μsec would be no less than 30μsec with a processor speed of 266MHz, which is double the computation time required for the developed Luenberger-sliding mode observer with parameter adaptation algorithm. Thus, the proposed method can be considered to be significantly simpler computationally than an Extended Kalman filter.

The first set of HIL simulation results are for the high power induction motor with electric vehicle load. The hardware controller has been tested with 30% and 50% increased rotor resistance in the motor model. The results were captured by oscilloscope through the controller I/O port. A speed profile of 5sec ramping and 5sec cruising at a motor speed of 101rad/sec (20mph with wheel radius of 0.28m and gear ratio of 3.2) has

been applied. This 20mph is the maximum speed of SAE urban driving schedule B. Figures 6.3 and 6.4 show the command speed and the actual speed, respectively. The results show that the actual speed follows the command speed. Figures 6.5 and 6.6 show rotor and stator resistance estimations, respectively for 30% increased rotor resistance (R_r

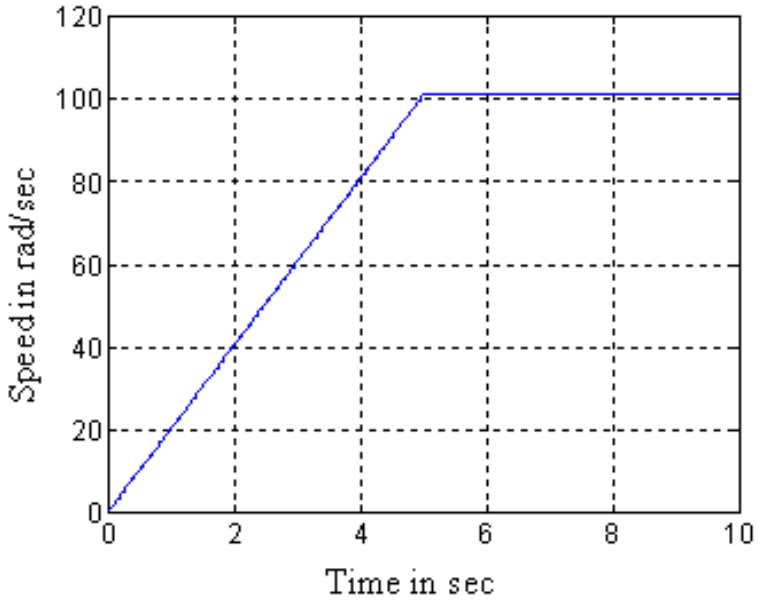
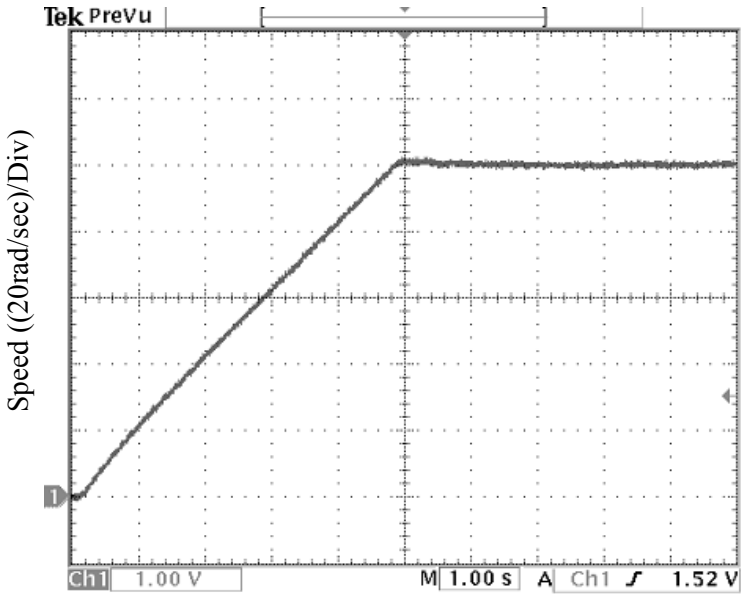


Figure 6.3. Motor command speed.



Time (1 sec/Div)

Figure 6.4. Motor actual speed.

= 0.228*1.3Ω≈0.296Ω) in the motor model. From Figure 6.5, it is observed that the estimated rotor resistance converges to the value of 0.30Ω (1.21% error) within 3.5sec. From Figure 6.6, it is observed that stator resistance reaches 0.087Ω within 3sec, and

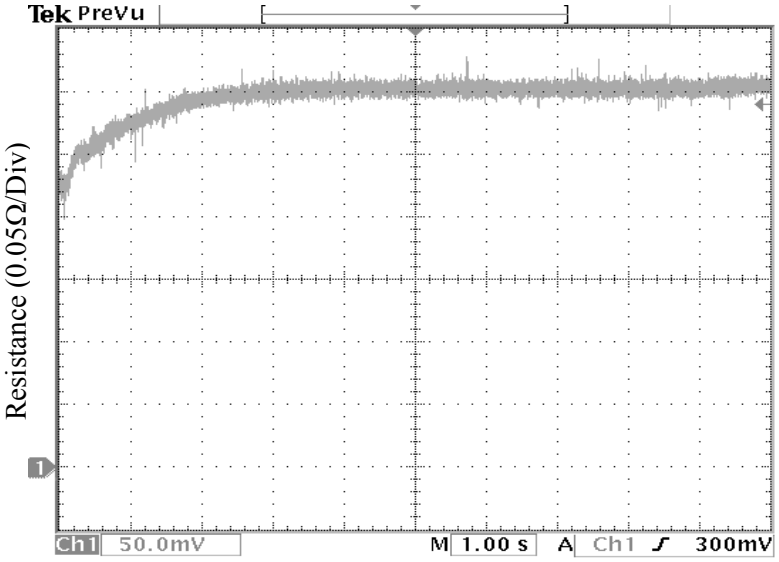


Figure 6.5. Rotor resistance estimation with 30% increased rotor resistance in motor (high power) model.

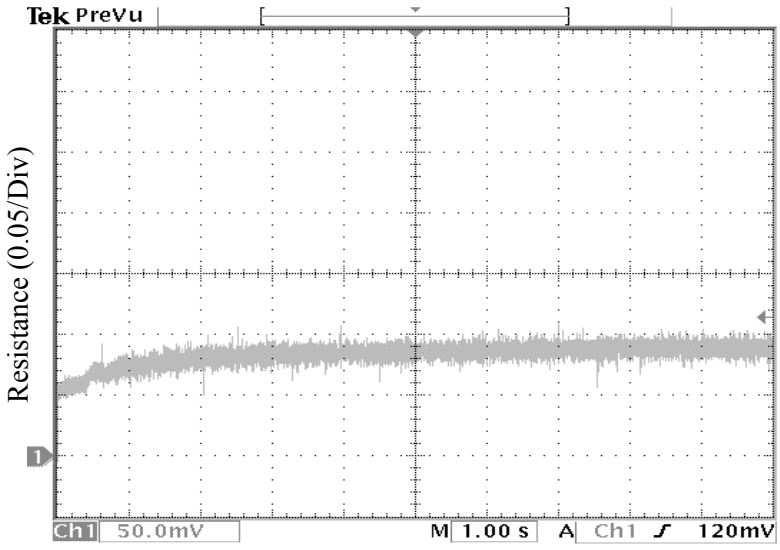


Figure 6.6. Stator resistance estimation with 30% increased rotor resistance in motor (high power) model.

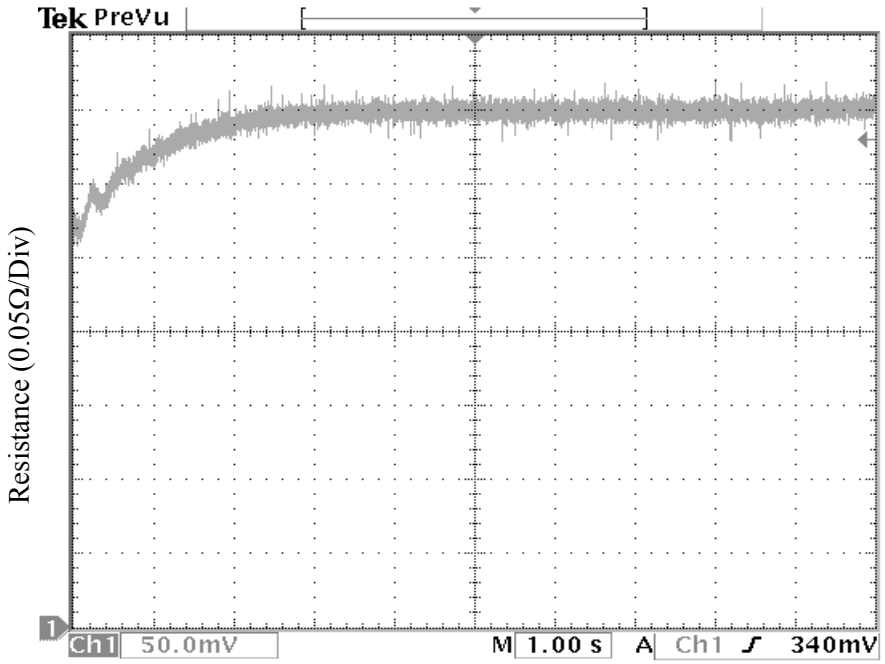


Figure 6.7. Rotor resistance estimation with 50% increased rotor resistance in motor (high power) model.

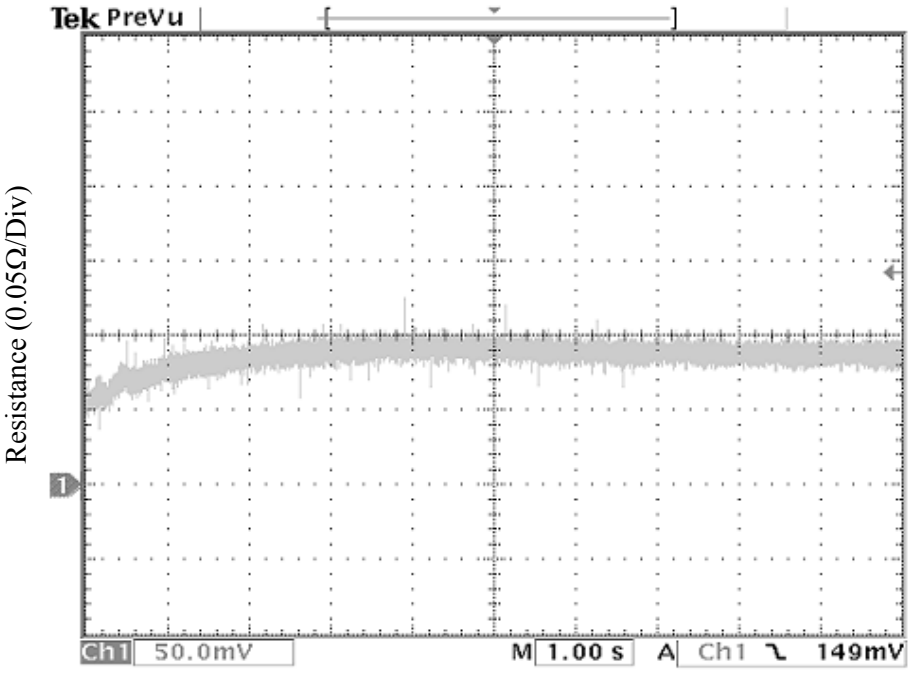


Figure 6.8. Stator resistance estimation with 50% increased rotor resistance in motor (high power) model.

converges to a value of 0.088Ω within 5sec with a tracking error of 2.3%. Figures 6.7 and 6.8 show rotor and stator resistance estimations, respectively for 50% increase in rotor resistance (R_r = 0.228*1.5Ω=0.342Ω). From Figure 6.7, it is seen that estimated rotor resistance converges to the value of 0.35Ω within 3.5secs with a tracking error of 2.3%. Figure 6.8 shows that stator resistance reaches 0.087Ω within 3secs, and converges to a value of 0.088Ω within 5secs with an error of 1.15%. The HIL simulation results for high power induction motor show that the proposed Luenberger-sliding mode observer with parameter adaptation algorithm shows excellent accuracy and fast convergence with short computation time. The fast convergence of the parameters improves the dynamic performance of the drive.

The HIL simulation has also been carried out for the small induction motor that was used in SIL simulation. A speed profile of 5sec ramping and 5sec running at a rated speed of 1740rpm (182.2rad/sec) has been applied. A constant load torque of 0.3Nm has

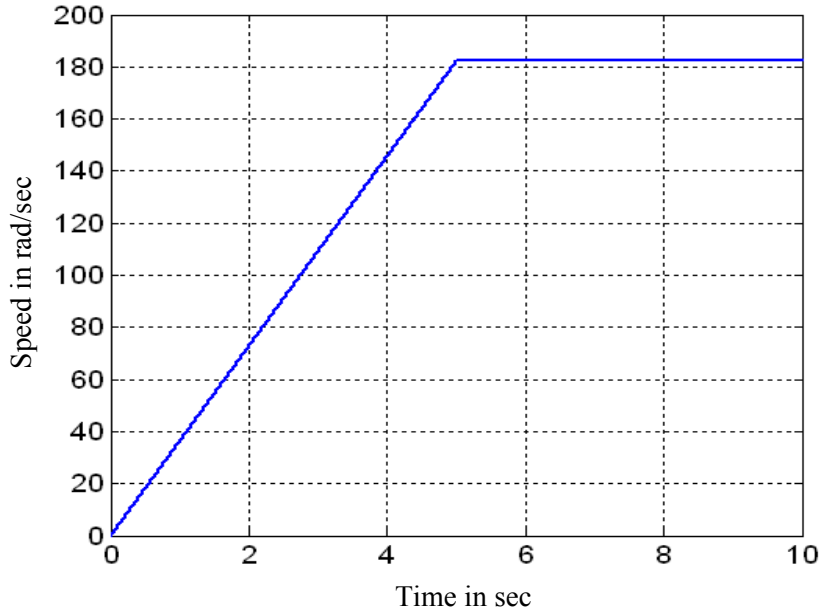
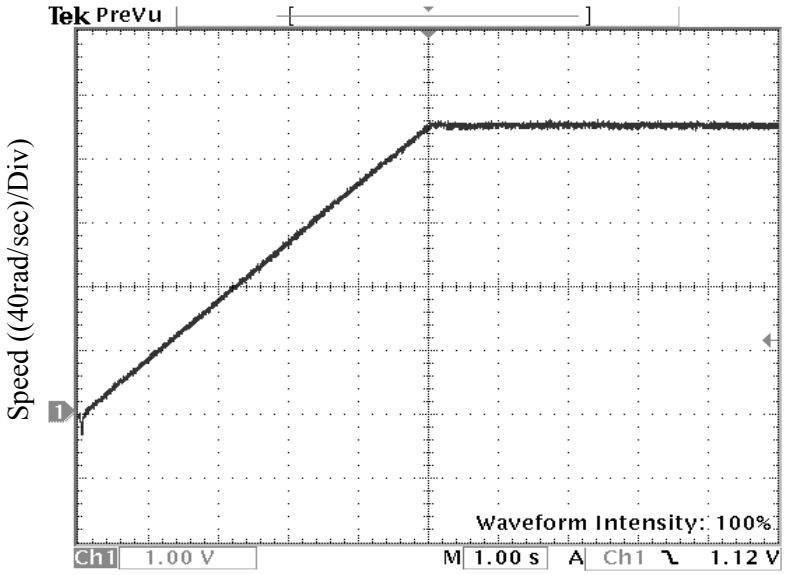
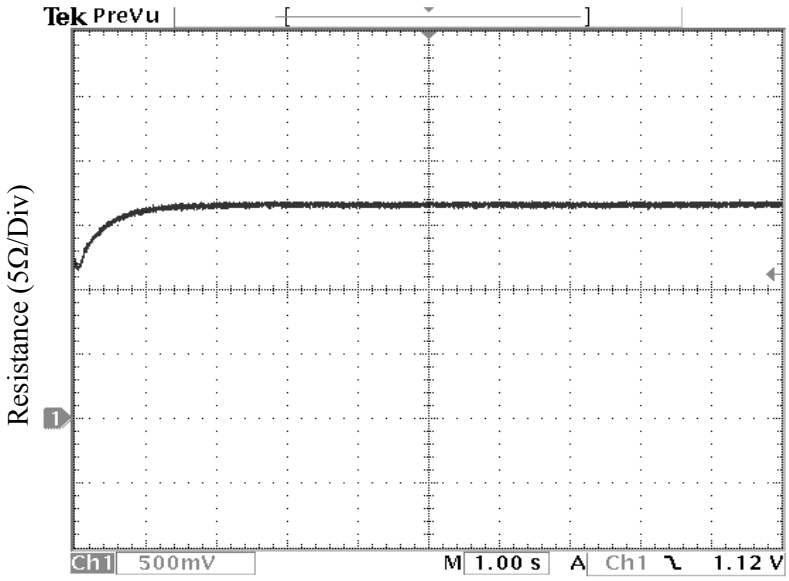


Figure 6.9. Motor command speed.



Time (1 sec/Div)

Figure 6.10. Motor actual speed.



Time (1 sec/Div)

Figure 6.11. Rotor resistance estimation with 30% increased rotor resistance in motor (small) model.

been considered in the motor model, and a switching frequency of 20kHz has been selected in the inverter model. Figures 6.9 and 6.10 show the motor command speed and actual speed profiles, respectively. The results show that the actual speed follows the

command speed. Figures 6.11 and 6.12 show rotor and stator resistance estimations, respectively for 30% increased rotor resistance (R_r = 1.3*12.77Ω = 16.6Ω) in motor model. From Figure 6.11, it is observed that estimated rotor resistance reaches 16.5Ω

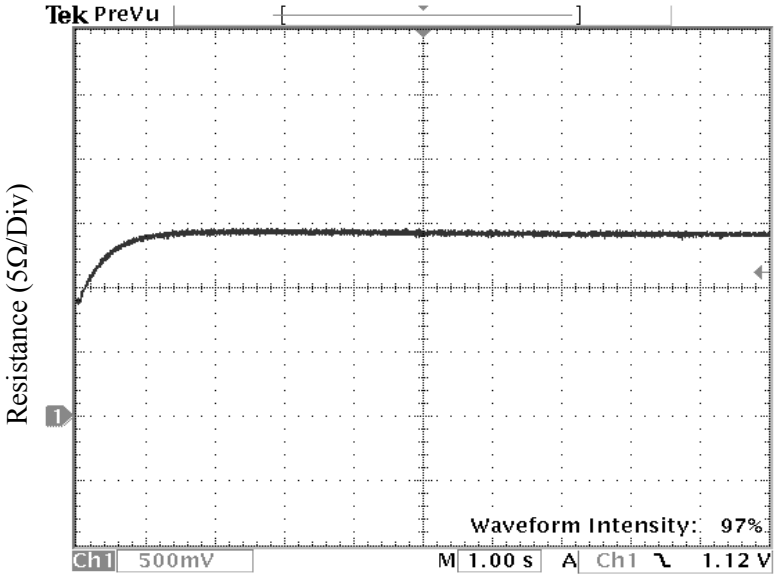


Figure 6.12. Stator resistance estimation with 30% increased rotor resistance in motor (small) model.

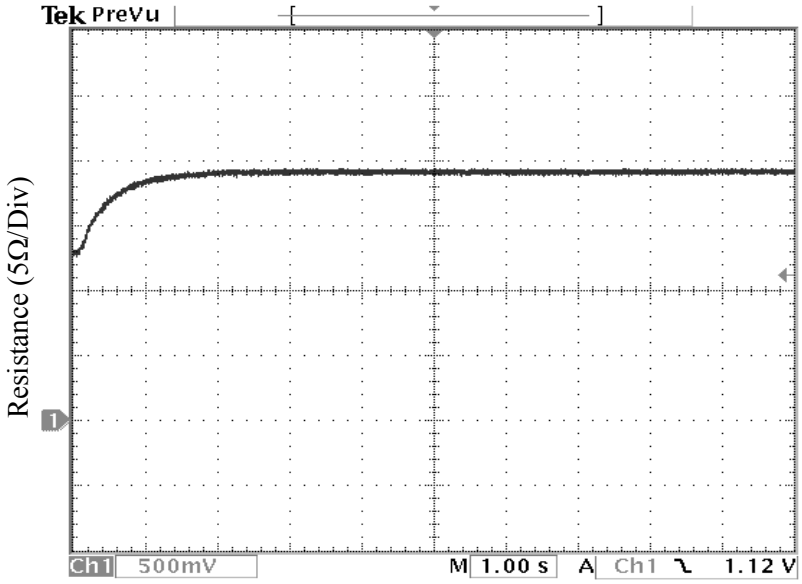


Figure 6.13. Rotor resistance estimation with 50% increased rotor resistance in motor (small) model.

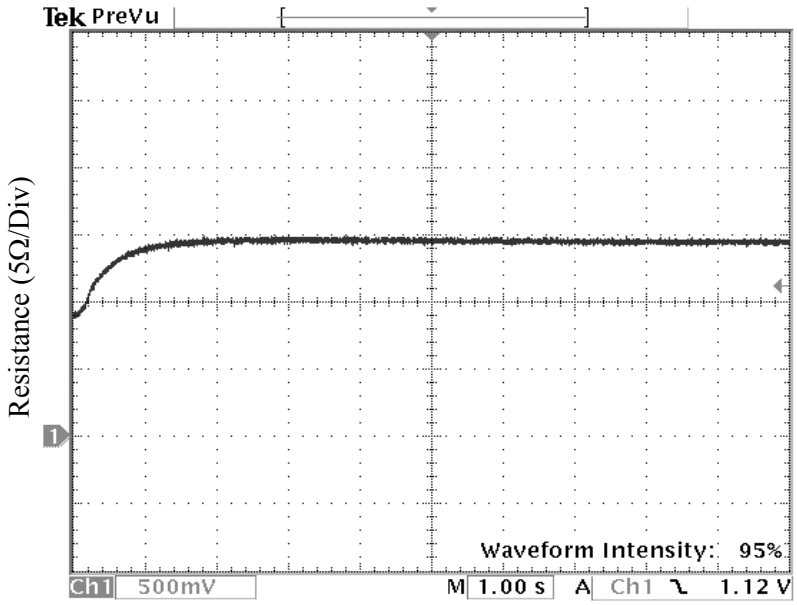


Figure 6.14. Stator resistance estimation with 50% increased rotor resistance in motor (small) model.

within 2sec and converges to a value of 16.75Ω within 3sec with a tracking error of 0.9%. Figure 6.12 shows that estimated stator resistance (R_s = 14.6Ω) converges to a value of 14.25Ω within 4.5sec with an error of 2.4%. Figures 6.13 and 6.14 show rotor and stator resistance estimations, respectively for 50% increase in rotor resistance (R_r =1.5*12.77Ω = 19.155Ω). From Figure 6.13, it is observed that estimated rotor resistance converges to a value of 19.25Ω within 3sec with an error of 0.5%. From Figure 6.14, it is seen that estimated stator resistance converges to a value of 14.4Ω within 4.7sec with a tracking error of 1.36%. The HIL simulation results show that the proposed Luenberger-sliding mode observer with parameter adaptation algorithm shows excellent accuracy for the small induction motor.

Table 6.1 summarizes the tracking error and convergence time of the estimated rotor and stator resistances. From this table, it is observed that the convergence time is

not more than 5sec, and the tracking error is less than 2.5%, which is highly satisfactory. The HIL simulation showed that the sliding mode gain matrix selected for this research has stable operation at least up to 50% variation of rotor resistance where Luenberger gain matrix was predetermined from the motor model.

Table 6.1. Tracking error and convergence time

			Rotor resistance estimation		Stator resistance estimation	
			Tracking Error	Convergence time	Tracking Error	Convergence time
High power motor	SIL Simulation	30% increase in rotor resistance	≈0%	4.5sec	1.38%	4.5sec
		50% increase in rotor resistance	0.3%	4.2sec	1.26%	4.5sec
	HIL Simulation	30% increase in rotor resistance	1.21%	3.5sec	2.3%	5sec
		50% increase in rotor resistance	2.3%	3.5sec	1.15%	5sec
Small motor	SIL Simulation	30% increase in rotor resistance	0.29%	3.5sec	0.685%	3sec
		50% increase in rotor resistance	0.235%	3.5sec	0.342%	3sec
	HIL Simulation	30% increase in rotor resistance	0.9%	3sec	2.4%	4.5sec
		50% increase in rotor resistance	0.5%	3sec	1.36%	4.7sec

6.2 EXPERIMENTAL RESULTS

Experimental verification is essential to ensure that performance of the advanced controller with parameter adaptation algorithm is satisfactory. The controller with new

on-line parameter estimation and adaptation algorithm for induction motor (propulsion motor) has been implemented and tested on the small induction machine driven by a three-phase PWM inverter supplied by the DC-link voltage. The static and dynamic performances of the proposed parameter estimation and on-line adaptation algorithm have been verified through experiments. The same small 0.5hp, 1A rms (1.414A max), 1740rpm, 60Hz and 4-pole induction machine used for SIL and HIL simulation has been used for this purpose. The machine parameters have already been presented in section 5.3.1. The values of rotor and stator resistances, which will be estimated, are 12.77Ω and 14.6Ω, respectively. The controller parameters have been tuned, and the parameter adaptation algorithm has been verified for this small motor by HIL simulation. The virtual motor model is now replaced by a real induction motor but the same FPGA based controller from HIL setup is used for the hardware experimentation.

6.2.1 EXPERIMENTAL SETUP

The experimental setup for the induction motor drive is shown in Figure 6.15. The setup consists of the FPGA based controller, host PC used as workstation, inverter, machine, sensors, and the hardware interfacing board between the controller, inverter and sensors. Two Hall Effect current sensors and an encoder are used for phase currents and speed feedback. The PWM switching frequency was set to 10kHz, and the maximum reference speed was set to 1740rpm. The hardware interface board between the controller and the inverter processes the PWM signals, phase current feedback, speed feedback and inverter DC-link voltage. Figure 6.16 shows the schematic of the hardware interfacing

board. The current feedback signals are scaled from 0 to 3V (±3A), and the PWM signals are scaled down from 12V to the inverter driver board specification of 4V.

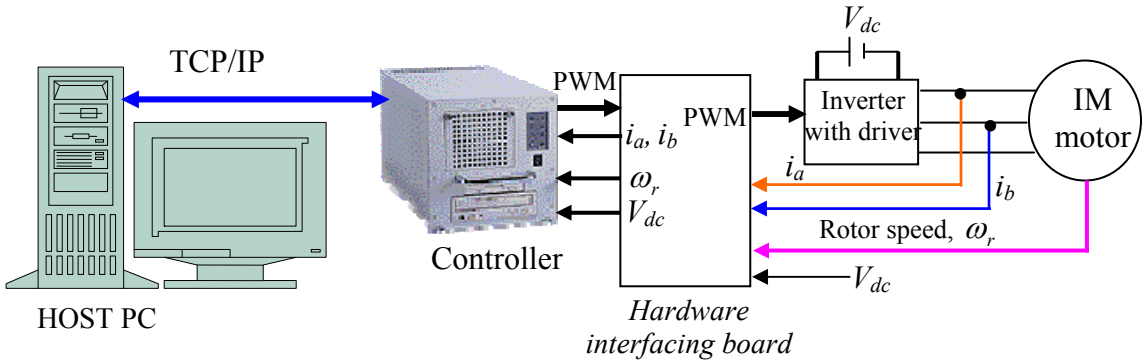


Figure 6.15. Experimental setup for the induction motor drive.

6.2.2 RESULTS FOR PARAMETER ESTIMATION AND ADAPTATION

The control algorithm is built in the host PC and downloaded to the controller through Ethernet. In the experiment, the initial value of rotor resistance was set to 50% of the actual value (0.5*12.77≈6.4Ω) in the controller and observer, and the motor was running at no load. The objective was to verify whether the parameter estimation algorithm could track the actual values of rotor and stator resistances. The on-line adaptation was performed by feeding back the estimated rotor resistance to the controller during the induction machine operation. The initial value of the stator resistance was set to 9Ω. The estimated rotor resistance, stator resistance and estimated phase currents were captured by the oscilloscope through the analog output port of the controller. The actual phase currents were also captured by the oscilloscope through a current transducer.

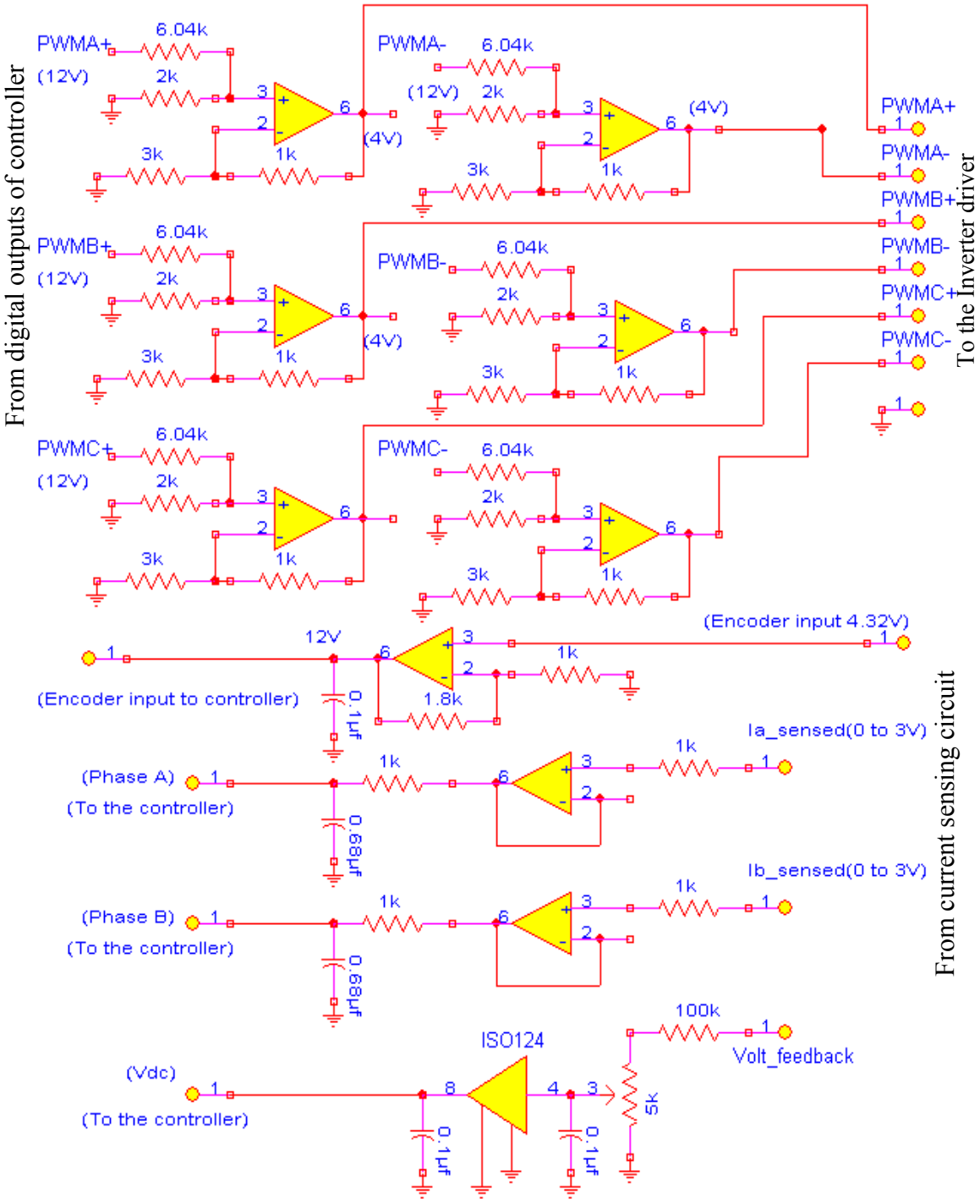
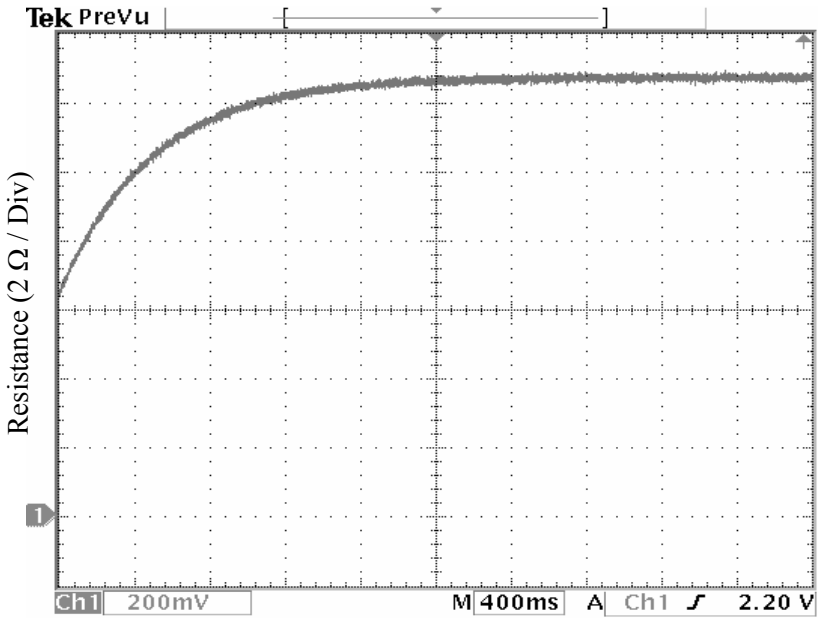
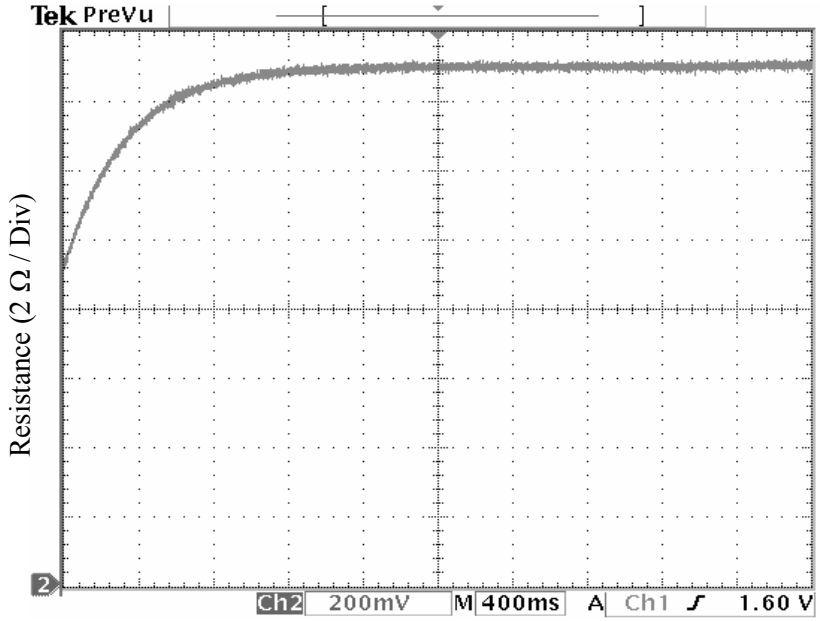


Figure 6.16. Circuit schematic of Hardware interfacing board.



Time (400ms/Div)

Figure 6.17. Rotor resistance estimation.



Time (400ms/Div)

Figure 6.18. Stator resistance estimation.

Figures 6.17 and 6.18 show the estimated rotor resistance and stator resistance, respectively. Figure 6.17 shows that the estimated rotor resistance converges to the value of 12.5Ω (actual value 12.77Ω) with a tracking error of 2.1%, within 2.5secs. From

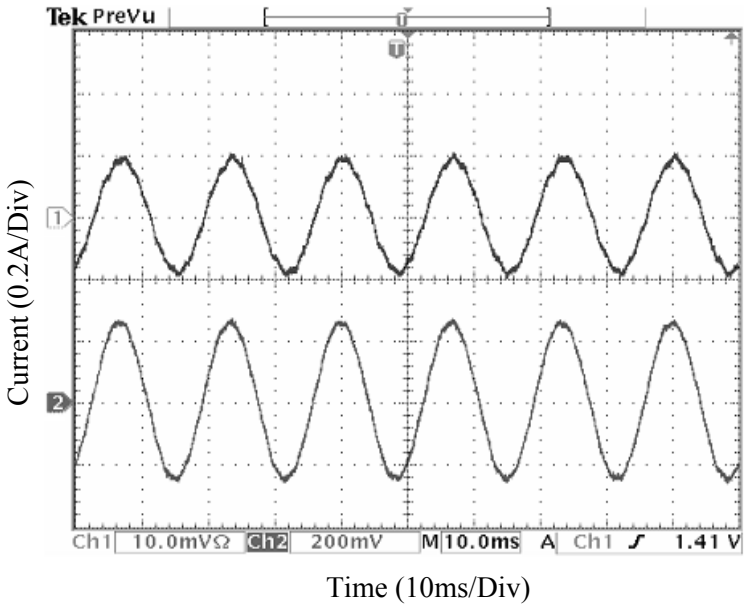


Figure 6.19. Phase current profiles before rotor resistance converges:
Ch1: Actual phase current, Ch2: Estimated phase current.

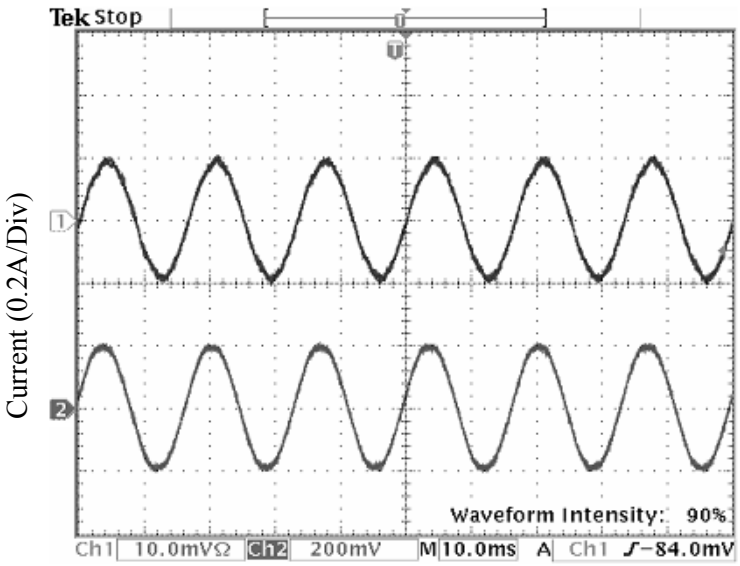


Figure 6.20. Phase current profiles after rotor resistance converges:
Ch1: Actual phase current, Ch2: Estimated phase current.

Figure 6.18, it is also observed that estimated stator resistance converges to the value of 14.95Ω (actual value 14.6Ω) with an error of 2.4%, within 2.4secs. Figures 6.19 and 6.20

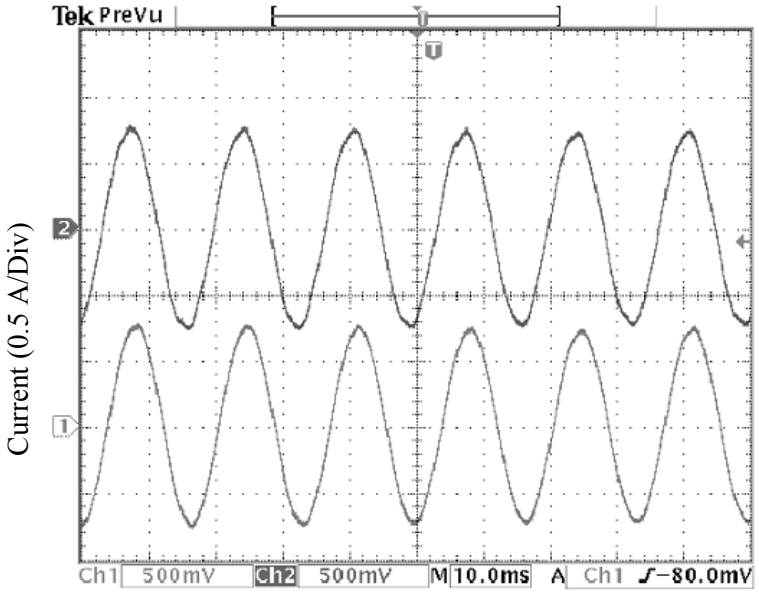


Figure 6.21. Phase current profiles for 53% load after convergence:
Ch1: Actual phase current, Ch2: Estimated phase current.

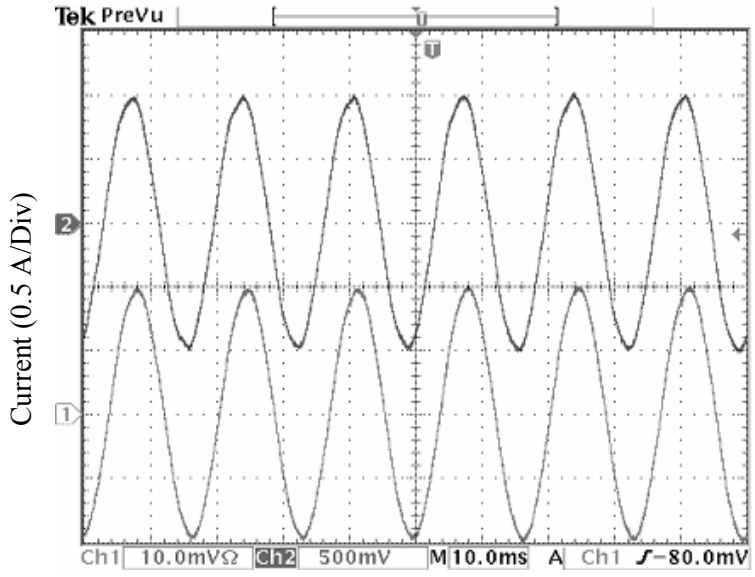


Figure 6.22. Phase current profiles for 70.7% load after convergence:
Ch1: Actual phase current, Ch2: Estimated phase current.

show the phase current profiles before and after the convergence of rotor resistance, respectively. The estimated phase current reaches the same value of actual phase current after the convergence of rotor resistance.

Several loads were applied momentarily after the convergence of rotor resistance. Figures 6.21 and 6.22 show the phase current profiles for 53% load (0.75A max) and for 70.7% load (1A max), respectively. The estimated phase current is observed to be equal to the actual phase current since the controller has the correct rotor resistance. From the experimental results, it is observed that the Luenberger-sliding mode observer with parameter estimation and adaptation algorithm shows excellent accuracy and fast convergence. One loop computation time for the controller only (without parameter adaptation algorithm) including analog/digital interface block and event detector/generator block for PWM output from real time libraries is 35μsec, whereas one loop computation time for the controller with parameter adaptation algorithm is 50μsec. The total loop-computation time with 43% increase due to parameter adaptation is still within the PWM cycle time of 100μsec.

6.3 CONCLUSION

The proposed observer with parameter adaptation algorithm for induction motor drive has been verified by HIL simulation and experimentation. The controller with on-line parameter adaptation algorithm improves the performance of the induction motor drive which is widely used as propulsion motor in HEV/EV. The HIL simulation helped in controller parameter tuning and facilitated ‘what-if’ studies. After getting satisfactory

results from HIL simulation, an experimental setup has been built where the controller with parameter adaptation algorithm has been implemented in an FPGA based control hardware. The results show that parameter estimation and adaptation algorithm provides good accuracy and fast convergence compared to other techniques.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1 SUMMARY

In recent years, the hybrid electric vehicle has received increased attention, and has become popular in the automotive market as a reduced emission vehicle. The hybrid electric vehicle is expected to play a significant role in increasing fuel economy and reducing air pollution. Two motor drive subsystems of the electric power transmission path (EPTP) in EVs and HEVs have been studied in this research. The focus was on component selection and control algorithm development of the two advanced motor drives for the efficient use of the EPTP.

The high power starter-generator drive system with engine starting, generating and motoring capabilities is a key component in the EPTP of an HEV. The component selection and advanced controller development of a high power starter-generator for a series-parallel 2x2 HEV have been addressed in this research. A simulation has been carried out for the PM BLDC based high power starter-generator to verify the engine starting, generation and motoring capabilities as well as to size the three-phase inverter that drives the starter-generator. A gain scheduled PI current controller with an *antiwindup integrator* has been designed and implemented in hardware using a TI fixed point DSP TMS320F2812 that has a clock frequency of 150MHz. A new improved high-

resolution current sensing board has also been implemented to improve the current regulation compared to commonly used techniques. The PM BLDC based starter-generator system has been tested in the generating and motoring modes with the starter-generator mounted inside the engine compartment of the vehicle and directly coupled to the engine. The high power density of the starter/generator and a well-designed torque/current controller enabled fast cranking of a diesel engine, and reduced the engine torque ripple. From simulation and experimental results, it has also been observed that the starter-generator can be operated at the desired highest efficiency point for power generation, which is useful in the vehicle's series propulsion mode.

The electric motor drive used for propulsion is another key component in the powertrain of an EV or HEV. The effects of parameter variation on static and dynamic performance of an induction motor drive used as the propulsion motor in an HEV or EV have been investigated through simulation. The difficulties and issues associated with existing on-line parameter estimation algorithms for induction motor drives have been addressed. For efficient use of the propulsion system, a novel observer with parameter estimation and adaptation algorithm for an induction motor drive has been presented. The new Luenberger-sliding mode observer is simple, easy to implement and overcomes the difficulties of existing observers. The induction motor controller with the novel parameter estimation and adaptation algorithm has been verified through SIL and HIL simulation as well as by experiment. The HIL simulation helped to tune the controller parameters, and facilitated 'what-if' studies. The controller with the proposed parameter adaptation algorithm has been implemented in FPGA based control hardware. The results show that the proposed Luenberger-sliding mode observer with parameter adaptation algorithm

provides excellent accuracy and fast convergence with less computation compared to other estimation techniques.

7.2 RESEARCH CONTRIBUTION

This research has produced the following contributions towards the development of advanced motor drive subsystems for electric power transmission path of an HEV.

- High power starter-generator
  - Study of different electric machines for the selection of a suitable machine for the high power starter-generator of an HEV.
  - An improved current mode controller and drive system for a PM BLDC based high power starter-generator of a series-parallel 2x2 HEV.
  - A starter-generator controller implemented in hardware using TI DSP TMS320F2812 with CAN communications and fault diagnostics.
  - A new high-resolution current feedback circuit to achieve higher accuracy in current/torque regulation.

- Induction motor (propulsion motor) drive
  - Study of the effects of parameter variation on the static and dynamic performance of induction motor drives.
  - A novel observer for on-line parameter estimation and adaptation and validated with SIL and HIL simulation.

- An FPGA based control hardware with the new parameter estimation and adaptation algorithm implemented.
- HIL simulation and experimental results showing parameter tracking capability with excellent accuracy even in the event of large parameter variations.

7.3 LIMITATIONS IN EXPERIMENTAL SETUP

Several issues were encountered during the testing of starter-generator and induction motor drive with the current experimental setup. Improvements in hardware packaging and mounting will deliver better performance of the developed algorithms.

- *Starter-generator:* The controller was situated a small distance away from the inverter and three-phase high power lines during stationary testing to minimize the electromagnetic interference (EMI) due to high frequency switching on the controller and the communication bus. However, there was still some noise in the current feedback signals that was affecting the precision of the ADC channels of the DSP. During on-road testing at high currents, the EMI noise traveling through the high power three-phase lines and from the inverter was affecting the control signals as well as interrupting the communication bus due to improper shielding of the inverter and the three-phase lines, with the inverter and the controller mounted inside the vehicle. Since only two current sensors were used in phases A and B of the PM BLDC, the current transition during the commutation from phase C to A was not smooth, which affected the current regulation during that commutation instant.

- *Induction motor drive:* The encoder used for the induction motor drive testing was a single pulse-train encoder, which was mounted on the induction motor. As a result, the speed control at low speed was not satisfactory. A quadrature encoder would have given better performance at low speed.

7.4 FUTURE WORK

The following research is suggested for the starter-generator subsystem of the hybrid electric vehicle powertrain.

- Since at high current higher switching frequency increases switching loss, a variable switching frequency profile depending on the magnitude of phase current can be used to reduce the switching loss of the inverter.
- An appropriate DC bus filter need to be designed to keep the bus voltage and current ripple within a limit for reducing the torque ripple and stress on the energy storage system. The resonance frequency of the system including the DC bus has to be far away from the lower limit of the switching frequency profile to avoid overshoots in the phase current at the beginning of a phase conduction period.
- It is essential to use a properly shielded case with a shielding ground to chassis for the inverter as well as shielded three-phase high power lines with shielding grounds at both ends (inverter and machine) of the cables to reduce the EMI effects on controller operation. This will establish a robust communication between starter-generator controller and SCM through CAN protocol.

- Three current sensors with high-resolution current sensing circuit can be used to make the current transition smooth during each commutation instant.
- On-road testing of the starter-generator for a long period is necessary to observe the robustness of the controller.
- The starter-generator needs to be tested in the torque boosting mode.

In order to achieve high performance from the induction motor drive, a high-resolution encoder is necessary. The advanced controller for the induction motor drive developed in this research needs to be tested for the high power induction motor mounted in an HEV to verify the performance and robustness of the controller as well as to observe the advantages in HEV application. It would also be interesting to use the observer for estimating other electrical (self inductance and magnetizing inductance) and mechanical (moment of inertia and damping coefficient) parameters off line to identify the parameters of an induction machine. The same approach of parameter estimation can also be applied for other types of machines.

REFERENCES

[1] D. W. Novotny and T. A. Lipo, *Vector Control and Dynamics of AC Drives*, Clarendon Press, Oxford, 1996.

[2] I. Husain, *Electric and Hybrid Vehicles Design Fundamentals*, CRC press LLC, 2003.

[3] C. C. Chan, "The State of Art of Electric and Hybrid Vehicles," *Proceedings of the IEEE*, Vol. 90, No. 2, Feb. 2002, pp. 247-275.

[4] S. E. Gay, H. Gao et al., "Fuel Cell Hybrid Drive Train Configurations and Motor Drive Selection," *IEEE 56th Vehicular Technology Conference Proceedings*, vol. 2, Sept. 2002, pp. 1007-1010.

[5] The University of Akron Challenge X report # 5, "Series-Parallel 2x2 Hybrid Electric Vehicle Architecture Design," May 2005.

[6] S. A. Naser, I. Boldea and L. E. Unnewehr, *Permanent Magnet, Reluctance and Self-Synchronous Motors,* CRC Press, Ann Arbor, 1993.

[7] J. Heywood, *Internal Combustion Engine Fundamentals*, McGraw-Hill, NY, 1988.

[8] A. W. Judge, *Automobile Engines in Theory, Design, Construction, Operation and Testing*, Robert Bentley Inc., Massachusetts, 1972.

[9] J. G. Kassakian, H. C. Wolf et al., "The Future of Automotive Electrical Systems," *IEEE Automotive Electronics Conference Proceedings*, Dearborn, MI, Oct. 1996, pp. 3-12.

[10] J. Wang, Z. Ping et al., "Three-Phase Modular Permanent Magnet Brushless Machine for Torque Boosting on a Downsized ICE Vehicle," *IEEE Transaction on Vehicular Technology*, vol. 54, no. 3, pp. 809-816, May 2005.

[11] J. G. Kassakian, J. M. Miller et al., "Automotive Power Electronics Up," *IEEE Spectrum Mag.*, May 2000, pp. 34-39.

[12] R. I. Davis, R. D. Lorenz, "Engine Torque Ripple Cancellation with an Integrated Starter Alternator in a Hybrid Electric Vehicle: Implementation and Control," *IEEE Transactions on Industry Applications*, vol. 39, no. 6, pp. 1765-1774, Nov./Dec.

2003.

[13] A. K. Jain, S. Mathapati et al., "Integrated Starter Generator for 42-V Powernet Using Induction Machine and Direct Torque Control Technique," *IEEE Transactions on Power Electronics*, vol. 21, no. 3, pp. 701-710, May 2006.

[14] J. M. Miller, A. R. Miller et al., "Starter-Alternator for Hybrid Electric Vehicle: Comparison of Induction and Variable Reluctance Machines and Drives," *IEEE 33rd Annual Industry Applications Society Conference*, vol. 1, Oct. 1998, pp. 513-523.

[15] E. C. Lovelace, T. M. Jahns et al., "Design and Experimental Verification of a Direct-Drive Interior PM Synchronous Machine Using a Saturable Lumped-Parameter Model," *IEEE 37th Annual Industry Applications Society Conference*, vol. 4, Oct.2002, pp. 2486-2492.

[16] E. C. Lovelace, T. M. Jahns et al., "Mechanical Design Considerations for Conventionally-Laminated, High-Speed, Interior PM Synchronous Machine Rotors," *IEEE Transactions on Industry Applications*, vol. 40, no. 3, pp. 806-812, May/June 2004.

[17] M. Naidu, N. Boules, R. Henry, "A High-Efficiency, High Power Generation System for Automobiles," *IEEE Transactions on Industry Applications*, vol. 33, no. 6, pp. 1535-1543, Nov./Dec. 1997.

[18] R. Hodkinson and J. Fentos, "Lightweight Electric/Hybrid Vehicle Design," SAE, Warrendale, PA 2001.

[19] N. M. Picot, "A Strategy to Blend Series and Parallel Modes of Operation in a Series-Parallel 2-by-2 Hybrid Diesel/Electric Vehicle," Thesis, MS Electrical Engineering, The University of Akron, December 2007.

[20] The University of Akron Challenge X Spring Technical Report 2007, "Validation of Series-Parallel Hybrid Electric Vehicle," April 2007.

[21] R. N. Pacciotti, "An Evaluation of Nitrogen Oxide Emission from a Light-Duty Hybrid-Electric Vehicle to Meet U.S.E.P.A Requirements Using a Diesel Engine," Thesis, MS Mechanical Engineering, The University of Akron, August 2007.

[22] Harry L. Husted, "A Comparative Study of the Production Applications of Hybrid Electric Powertrains," *SAE Future Transportation Technology Conference*, Document no. 2003-01-2307, June 23-25, 2003.

[23] M. Zeraoulia, M. E. H. Benbouzid et al., "Electric Motor Drive Selection Issues for HEV Propulsion Systems: A Comparative Study," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 6, pp. 1756-1764, Nov. 2006.

[24] P. Caratozzolo, M. Serra et al., "A Proposal for the Propulsion System of a Series Hybrid Electric Vehicle," *IEEE 34th Annual Power Electronics Specialists Conference*, vol. 2, June 2003, pp. 586-591.

[25] M. Osama and T. A. Lipo, "A New Inverter Control Scheme for Induction Motor Drives Requiring Wide Speed Range," *IEEE Transaction on Industry Applications*, vol. 32, no. 4, pp. 938-944, July/Aug. 1996.

[26] B. Kou, L. Li et al., "Operating Control of Efficiently Generating Induction Motor for Driving Hybrid Electric Vehicle," *IEEE Transactions on Magnetics,* vol. 41, no. 1, pp. 488-491, January 2005.

[27] K. Asano, S. Okada et al., "Vibration Suppression of Induction-Motor-Driven Hybrid Vehicle Using Wheel Torque Observer," *IEEE Transaction on Industry Applications*, vol 28, no. 2, pp. 441-447, March/April 1992.

[28] T. M. Jahns, "Motion Control with Permanent Magnet AC Machines," *IEEE Proceedings*, vol. 82, no. 8, Aug. 1994, pp. 1241-1252.

[29] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior PM Synchronous Motor Drive," *IEEE Transaction on Industry Applications*, vol. 25, no. 4, pp. 681-689, July/Aug. 1987.

[30] T. J. E. Miller, *Switched Reluctance Motors and Their Control*, Magna Physics, Hillsboro, Ohio; Clarendon Press, Oxford, U.K; Oxford University Press, New York, 1993.

[31] The University of Akron Challenge X report # 2, "The University of Akron Vehicle Architecture Selection and Analysis," November 2004.

[32] R. Pillay and R. Krishnan, "Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part II: the Brushless DC Motor Drive," *IEEE Transactions on Industry Applications*, vol. 25, no. 2, pp. 274-279, March/April 1989.

[33] B. K. Bose, *Power Electronics and AC Drives*, Prentice-Hall, New Jersey 07632, 1986.

[34] W. Slabiak and L. Lawson, "Precise Control of a Three-Phase Squirrel Cage Induction Motor Using a Practical Cycloconverter," *IEEE Trans. Ind. Gen. Applicat.*, vol. IGA-3, pp. 274-280, July/Aug. 1996.

[35] W. Shepherd and J. Stanway, "An Experimental Closed Loop Variable Speed Drive Incorporating a Thyristor Driven Induction Motor," *IEEE Trans. Ind. Gen. Applicat.*, vol. IGA-3, pp. 559-565, Nov./Dec. 1967.

[36] M. Anibal Valenzuela, Juan A. Tapia et al, "Thermal Evaluation of TEFC Induction Motors Operating on Frequency-Controlled Variable-Speed Drives," *IEEE Transactions on Industry Applications*, vol. 40, no. 2, pp. 692-698, March/April 2004.

[37] Thomas H. Bishop, "Temperature Monitoring is Key to Motor Reliability," *Maintenance Technology Magazine*, Applied Technology Publications, Barrington, IL, July 2004, pp. 34-F. (http://www.mt-online.com/articles/0704easa.cfm, visited on July 23, 2007)

[38] http://www.allaboutcircuits.com/vol_1/chpt_12/6.html, visited on July 23, 2007.

[39] W. Leonhard, *Control of Electrical Drives*, Springer-Verlag, 1997.

[40] P. Vas, *Sensorless Vector and Direct Torque Control*, Oxford University Press, New York, 1998.

[41] A. M. Trzynadlowski, *Control of Induction Motors*, Academic Press, New York, 2001.

[42] H. Kuboto, and K. Matsuse, "Speed Sensorless Field-Oriented Control of Induction Motor with Rotor Resistance Adaptation," *IEEE Transactions on Industry Applications*, vol. 30, no. 5, pp. 1219-1224, Sept-Oct. 1994.

[43] H. Kuboto, K. Matsuse et al., "DSP-Based Speed Adaptive Flux Observer of Induction Motor," *IEEE Transactions on Industry Applications*, vol. 29, no. 2, pp. 344-348, March/April 1993.

[44] T. Orlowska-Kowalska, "Application of Extended Luenberger Observer for Flux and Rotor Time-Constant Estimation in Induction Motor Drives," *IEE Proceedings on Control Theory and Applications*, vol. 136, no. 6, Nov. 1989, pp. 324-330.

[45] C. E. Moucary, G. G. Soto et al., "Robust Rotor Flux, Rotor Resistance and Speed Estimation of an Induction Machine Using the Extended Kalman Filter," *Proceedings of the IEEE International Symposium on Industrial Electronics*, Bled, Slovenia, vol. 2, July 1999, pp. 742-746.

[46] L. Loron and G. Laliberte, "Application of the Extended Kalman Filter to Parameter Estimation of Induction Motors," *Conf. Proceedings EPE'93*, Brighton, U.K., 1993, pp. 85-90.

[47] C. Attaianese, G. Tomasso, et al, "A Novel Approach to Speed and Parameters Estimation in Induction Motor Drives," *IEEE Trans. on Energy Conversion*, vol. 14, no. 4, pp. 939-945, Dec. 1999.

[48] R. Marino, S. Peresada and P. Tomei, "On-Line Stator and Rotor Resistance Estimation for Induction Motor," *IEEE Trans. on Control System Technology*, vol. 8, no. 3, pp. 570-579, May 2000.

[49] M. Cirrincione, M. Pucci et al., "A New Experimental Application of Least-Squares Techniques for the Estimation of the Induction Motor Parameters," *IEEE Trans. on Ind. Appl.*, vol. 39, no. 5, pp. 1247-1256, Sept./Oct. 2003,.

[50] J. Li, L. Xu and Z. Zhang, "An Adaptive Sliding Mode Observer for Induction Motor Sensorless Speed Control," *IEEE Transactions on Industry Applications*, Vol. 41, No. 4, pp. 1039-1046, July/Aug. 2005.

[51] B. Karanayil, M. F. Rahman et al., "Implementation of an On-Line Resistance Estimation Using Artificial Neural Networks for Vector Controlled Induction Motor Drive," *IEEE 29th Annual Industrial Electronics Society Conference (IECON)*, vol. 2, Nov. 2003, pp. 1703-1708.

[52] A. Trentin, P. Zanchetta et al., "A New Method for Induction Motors Parameter Estimation Using Genetic Algorithms and Transient Speed Measurements," *IEEE 41st Annual Industry Applications Society Conference*, vol. 5, Oct. 2006, pp. 2435-2440.

[53] A. Hren, and K. Jezernik, "A New Rotor Time Constant Estimation Method for Indirect Vector Controlled Induction Motor Drive," *Proceedings of the IEEE International Symposium on Industrial Electronics*, Portugal, vol. 3, July 1997, pp. 1029-1033.

[54] S. Wade, M. W. Dunnigan et al., "A New Method of Rotor Resistance Estimation for Vector-Controlled Induction Machines," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 2, pp. 247-257, April 1997.

[55] C. Schauder, "Adaptive Speed Identification for Vector Control of Induction Motors without Rotational Transducers," *IEEE Transaction on Industry Applications*, vol. 28, no. 5, pp. 1054-1061, Sept./Oct. 1992,.

[56] F. Z. Peng and T. Fukao, "Robust Speed Identification for Speed-Sensorless Vector

Control of Induction Motors," *IEEE Transaction on Industry Applications*, vol. 30, no. 5, pp. 1234-1240, Sept./Oct. 1994,.

[57] K. Ohnishi, Y. Ueda, and K. Miyachi, "Model Reference Adaptive System Against Rotor Resistance Variation in Induction Motor Drive," *IEEE Transaction on Industrial Electronics*, vol. IE-33, pp. 217-223, June 1986.

[58] *Power System Analysis Tool* (PSAT), Data delivered by *Oak Ridge National Laboratory*.

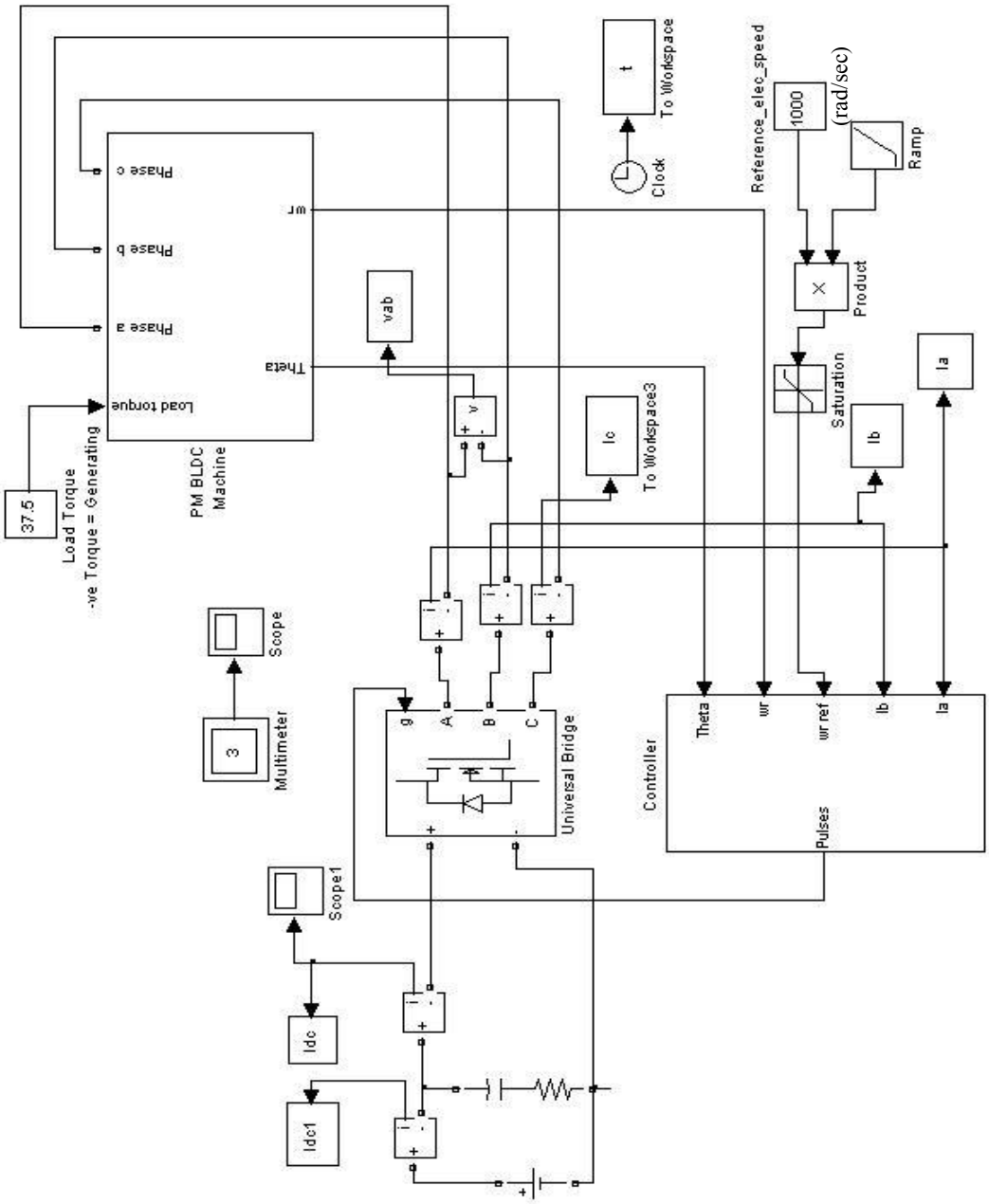
[59] S. E. Schulz and K. Rahman, "High-Performance Digital PI Current Regulator for EV Switched Reluctance Motor Drives," *IEEE Transaction on Industry Applications*, vol. 39, no. 4, pp. 1118-1126, July/Aug. 2003.

[60] I. Husain, M. S. Islam, "Design, Modeling and Simulation of an Electric Vehicle System," *SAE International Congress and Exposition*, Michigan, March 1-4, 1999.

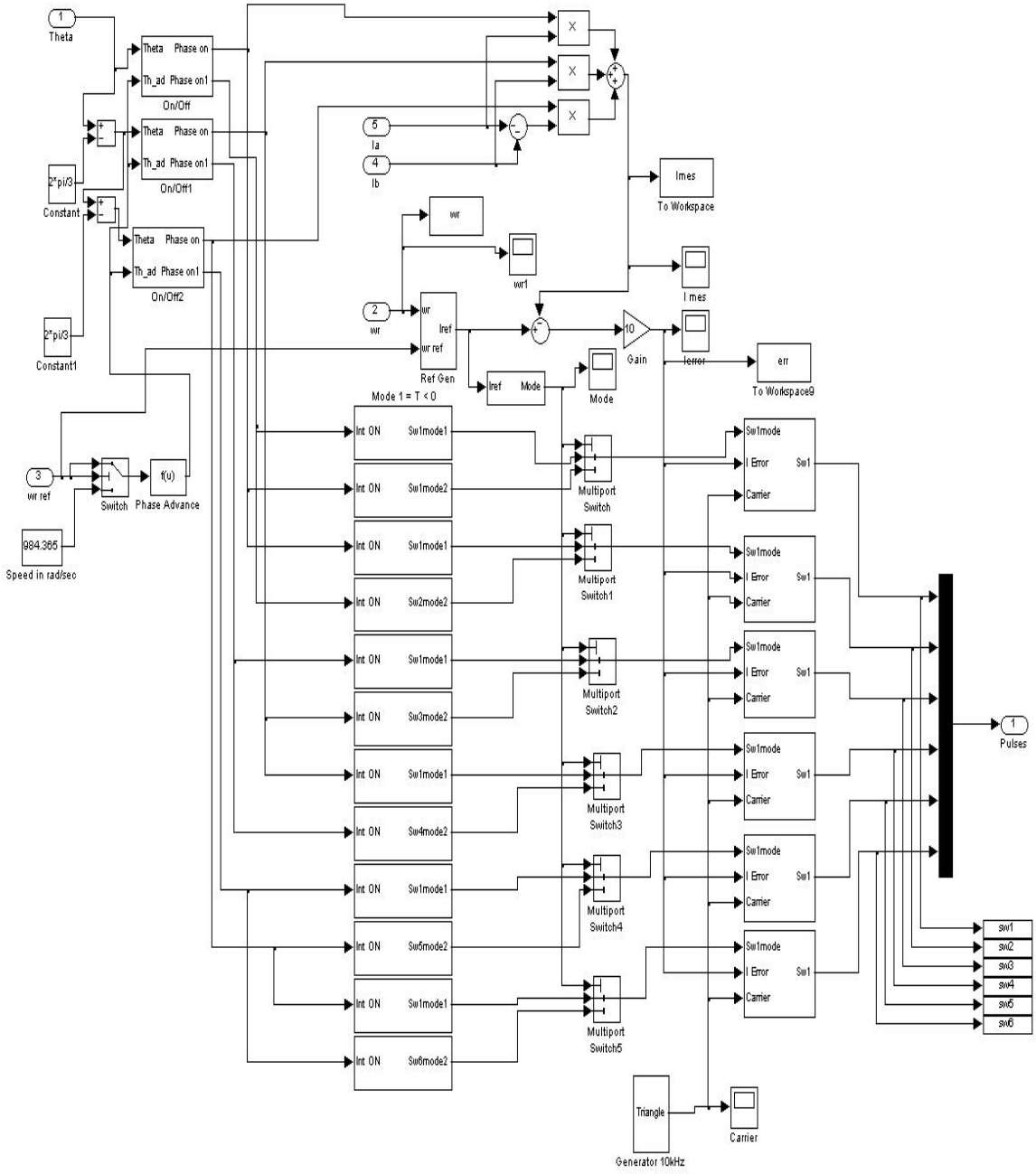
APPENDICES

APPENDIX A

SIMULINK MODEL OF PM BLDC STARTER-GENERATOR DRIVE

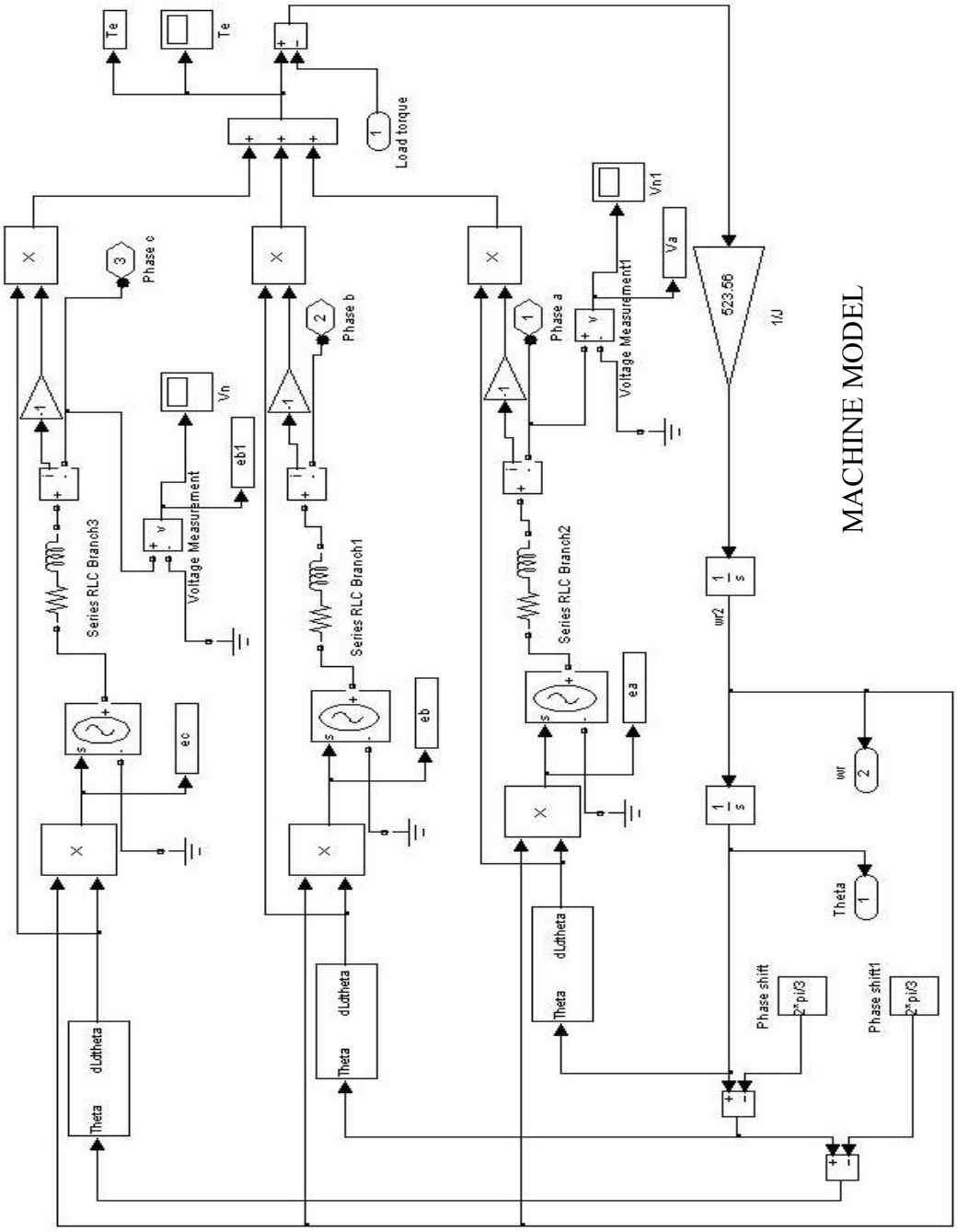


SIMULINK MODEL OF PM BLDC CONTROLLER



CONTROLLER

SIMULINK MODEL OF PM BLDC MACHINE



APPENDIX B

'C' CODE FOR TORQUE CONTROL OF STARTER-GENERATOR

```
#include "DSP281x_Device.h"
#define CMPR1        EvaRegs.CMPR1       // PWM1 duty cycle
#define CMPR2        EvaRegs.CMPR2       // PWM2 duty cycle
#define CMPR3        EvaRegs.CMPR3       // PWM3 duty cycle
#define CMPR4        EvbRegs.CMPR4       // time elapsed counting
#define T1PR  EvaRegs.T1PR          // PWM period
#define T2PR  EvaRegs.T2PR          // time counting period
#define T4PR  EvbRegs.T4PR          // time counting period
#define Kp      1                           // PI proportional gain
#define Ki      1                           // PI integration gain * sample period
#define antiwindup 0.02                     // antiwindup constant
#define KPHI  0.125610551
#define TOTORQUE 0.251221102    // 2*KPHI
#define MAX_CURRENT 150                     // Current limit for the BLDC
#define PWM_PERIOD 1875                     // 50 us 1875, 2500==15000khz
#define Tech 5                              // sampling time for the PI (5*10
                                               microseconds)
unsigned long int speed=0;                  // Actual speed of the BLDC

unsigned int generator_starting = 0;  //WRITE CODE FOR THIS related to
connect_generator

unsigned int raw_position=0, old_raw_position=0, position_cnt=0, relay_open_delay=0,
relay_flag=0;

int torque_cmd=0, torque_cmd1=0,check=0;                    // wanted current
int mode=0, mode1=0;

int position=0;                             // Actual position computed from Hall
sensors
int last_position=0;                        // the previous position, used to compute the speed

unsigned  int actual_time;      // chronometer in microseconds
unsigned  long int chronograph;  // number of periods elapsed since last position change
unsigned int curr_fdb=0, volt_temp=0;
```

```
unsigned int compt = 0;
//----------------------------------------
//
// Table to look up speed based on time
// since position last changed.

unsigned int generator_speed_lookup[102]= {
            20000, 16667, 14286, 12500, 11111,
                  10000, 9091, 8334, 7693, 7143, 6667, 6250, 5883, 5556, 5263,
                  5000, 4762, 4546, 4348, 4167, 4000, 3846, 3704, 3572, 3449,
                  3334, 3226, 3125, 3031, 2941, 2857, 2778, 2703, 2632, 2564,
                  2500, 2439, 2381, 2326, 2273, 2222, 2174, 2128, 2084, 2041,
                  2000, 1961, 1923, 1887, 1852, 1818, 1786, 1755, 1724, 1695,
                  1667, 1640, 1613, 1588, 1563, 1539, 1515, 1493, 1471, 1450,
                  1429, 1409, 1389, 1370, 1352, 1334, 1316, 1299, 1282, 1266,
                  1250, 1235, 1220, 1205, 1191, 1177, 1163, 1150, 1137, 1124,
                  1099, 1087, 1075, 1064, 1053, 1042, 1031, 1020, 1010, 1000,
                  990, 980, 970, 961, 952, 943, 934
        };

// Variables from sensors
//----------------------------------------
signed int ia=0,ib=0; // actual (low resolution version) currents of the phases
signed int ia100=0, ib100=0; // higher resolution versions
signed int ia200=0, ib200=0; // higher resolution versions
unsigned int Temperature = 0; // temperature
unsigned int Vbus = 0; // dc bus voltage

unsigned int polled_io_pins = 0, polled_io_pins1 = 0, polled_io_pins2 = 0,
polled_io_pins3 =0, polled_io_pins4 = 0;  // digital inputs from inverter

int ACQ[256];
unsigned char index_ACQ=0;

// current reference, used as a parameter in the PI function
// 0 to +4095 represents 0A to +200A, but leave this a
// signed variable because there may be some noise from the sensor.

signed int current_fb=0;
unsigned  int test_time1, test_result;

//----------------------------------------
// Variables helpful for debugging
//----------------------------------------
unsigned int divide_by_zero_count = 0;
```

```
unsigned int divide_by_small_count = 0;

//-----------------------------------------
// Variables for Fault Mode Analysis
//-----------------------------------------
// variables for tracking fault conditions.

unsigned int local_critical_fault = 0;          // serious fault and shut down the generator.
unsigned int local_noncritical_fault = 0;     // a fault serious enough that SCM may want
to change strategy.
unsigned int local_diagnostic_fault = 0;      // a non-serious fault-- should be data logged.
unsigned int rolling_count_from_SCM = 0;     // used to make sure SCM is still
                                                    communicating.
unsigned int invalid_position_count = 0;   // used to check for persistently bad position
                                                  signals.
unsigned int overcurrent_count = 0, overcurrent_instantaneous_count=0;
unsigned int check_faults = 0;    // Gets set to 1 after 3ms of connecting generator to HV
                                        bus.
// local critical faults.
unsigned int critical_communications_with_UALAN_lost = 0;  //invalid rolling count
                                          from SCM, or no message from/to SCM in 50 ms
unsigned int critical_inverter_overheat = 0;     //inverter indicates overheat (hotter
                                                  than 96 to 100 degrees C)
unsigned int critical_error_on_inverter_phaseA = 0;  // inverter indicates overcurrent or
                                                       short on phase A
unsigned int critical_error_on_inverter_phaseB = 0;  // inverter indicates overcurrent or
                                                       short on phase B
unsigned int critical_error_on_inverter_phaseC = 0;  // inverter indicates overcurrent or
                                                       short on phase C
unsigned int critical_error_in_computed_speed = 0;   // computed speed is slower than
                                          idle; possible coupling break or sensor malfunction
unsigned int critical_instantaneous_overcurrent = 0; // feedback current > 210 A
unsigned int critical_continuous_overcurrent = 0;    // feedback current of more than 150
                                                        A for two seconds or more
unsigned int noncritical_undervoltage = 0;          // dc bus voltage below 200 V
unsigned int critical_overvoltage = 0;              // dc bus voltage above 400 V
unsigned int critical_persistent_invalid_position_sequence = 0;  // position not advancing
                                        60 degrees for more than ten (filtered) readings
// local non-critical faults.
unsigned int noncritical_speed_high = 0;            // speed is higher than 12500 rpm
unsigned int noncritical_inverter_overheat = 0;      // temperature is higher than 80
                                                       degrees C
// local diagnostic faults.
unsigned int invalid_position_sequence = 0;         // position not advancing 60 degrees
```

```
unsigned int invalid_position_sensed = 0;          // hall sensors giving invalid position
reading (not 1 through 6)
unsigned int error_in_measured_current = 0;
unsigned int critical_Vbus = 0;
// Variables for generator command

unsigned int connect_generator = 0;

unsigned int clear_faults = 0;
unsigned int mem_connect_generator = 0;

unsigned int comp_time = 0;

//-----------------------------------------
// Variables for CAN messaging.  Bytes are
// numbered starting with 1.
//-----------------------------------------
unsigned int msg0_1=0;
unsigned int msg0_2=0;
unsigned int msg0_3=0;
unsigned int msg1_1=0;
unsigned int msg1_2=0;
unsigned int msg1_3=0;
unsigned int msg1_4=0;
unsigned int msg1_5=0;
unsigned int msg1_6=0;
unsigned int msg2_1=0;
unsigned int msg2_2=0;
unsigned int msg2_3=0;
unsigned int msg2_4=0;
unsigned int msg3_1=0;
unsigned int msg3_2=0;

unsigned int current_cmd=0,abstemp=0, new_duty=0, old_duty=0;
signed int mode;
signed int temp=0,sn_current_ref=0;

unsigned int receive=0;
unsigned int transmit=0,transmit1=0,transmit_cnt=0;
unsigned int rolling_cnt=0,four_bit_rolling_count=0;
unsigned int first=0,first1=0,first2=0;
unsigned long int dth=99960;
unsigned int generator_speed;

struct ECAN_REGS ECanaShadow;
```

```
int error;
int sum_error=0; // error memorized from last PI computation
int sum_current_limit=1300; //1200, Amp to digital : 2048 <=> 200 amps

unsigned int generator_speed_history[4]={0,0,0,0};
int duty=0;
unsigned int valid_position_this_cycle = 1;
unsigned int time_cmd=20001;
unsigned int Temperature_history [8] = {0,0,0,0,0,0,0,0};
unsigned int Vbus_history [8] = {0,0,0,0,0,0,0,0};
unsigned int thisTemperature = 0;
unsigned int thisVbus = 0;
unsigned int history_index = 0;

void delay (unsigned long int temps)
{
    unsigned long int i;
    for (i=0;i<temps;i++);
}

void PWM_init (void)
{
   EALLOW; // Allow writing to configuration registers.
   SysCtrlRegs.PCLKCR.all |= 0x0009;
   GpioMuxRegs.GPAMUX.all |= 0x033F;   // EVA PWM 1-6  pins

   EvaRegs.EXTCONA.bit.INDCOE = 0;     // enable/disable compare output

   // Timer 1 used for PWMs , timer 2 used for time counting
   EvaRegs.GPTCONA.all |= 0x1072;
   EvaRegs.COMCONA.all |= 0x8000;
   EvaRegs.COMCONA.bit.FCOMPOE = 0;          // full compare disabled
   EvaRegs.DBTCONA.all = 0x00E0;          // deadband timer
   EvaRegs.ACTRA.all = 0x666;

   T1PR = PWM_PERIOD;                              //         Timer 1 period

   EvaRegs.T1CON.all = 0x0846;

   CMPR1 = 0x0000; // duty cycles initialisation
   CMPR2 = 0x0000;
   CMPR3 = 0x0000;

   EDIS; // Disallow writing to configuration registers.
```

```
}

void interrupt_init()
{
        EALLOW; // Allow writing to configuration registers.
        //T2PR = 0x0177;   // Timer 2 period 0x177=1500/4 ==> interrupt every 10us
        T2PR = PWM_PERIOD; // 50us


EvaRegs.T2CON.all = 0x0846;
EvaRegs.EVAIMRB.bit.T2PINT =1;                  // enable interrupt on Timer 2 period

   InitPieCtrl();

        IER = 0x0000;
        IFR = 0x0000;

        InitPieVectTable();

        //Enable PIE group 3 interrupt 1 for T2PINT
  PieCtrlRegs.PIEIER3.all = M_INT1;

  //Enable CPU INT3 for T2PINT
  IER |= (M_INT3);

        EnableInterrupts();

        EDIS; // Disallow writing to configuration registers.
}


void ADC_init()
{
   EALLOW; // Allow writing to configuration registers.
        AdcRegs.ADCTRL3.bit.ADCBGRFDN = 0x3;            // Power up
bandgap/reference circuitry
        delay (32000);                          // Delay before powering up rest of ADC
        AdcRegs.ADCTRL3.bit.ADCPWDN = 1;        // Power up rest of ADC
        delay (32000);                          // Delay after powering up ADC

        AdcRegs.ADCTRL1.bit.SEQ_CASC |= 0;          // 0  not in cascaded mode
        AdcRegs.ADCTRL1.bit.ACQ_PS |= 0x8;
        AdcRegs.ADCTRL1.bit.CPS |= 0;
        AdcRegs.ADCTRL1.bit.CONT_RUN |= 0;      // Setup continuous run :1
        AdcRegs.ADCTRL1.bit.SEQ_OVRD |= 0;    // Enable Sequencer override feature
```

```
        AdcRegs.ADCTRL2.bit.EVA_SOC_SEQ1|=1;

        AdcRegs.ADCTRL3.bit.SMODE_SEL|=0;
        AdcRegs.ADCTRL3.bit.ADCCLKPS |= 0xA;    // ADC module clock =
                                    HSPCLK/2*ADC_CKPS   = 25MHz/(1*2) = 12.5MHz

        AdcRegs.ADCMAXCONV.bit.MAX_CONV1 |= 0x7;  // convert and store in 8
                                                    results registers
        AdcRegs.ADCCHSELSEQ1.bit.CONV00 |= 0x0;    // channel A0 ;current
                                                  measurement in phase A main
        AdcRegs.ADCCHSELSEQ1.bit.CONV01 |= 0x1;    //selects channel A1 ; current
                                                measurement in phase A +-100 to +-200
        AdcRegs.ADCCHSELSEQ1.bit.CONV02 |= 0x2;    //selects channel A2; current
                                                  measurement in phase A 0 to +-100
        AdcRegs.ADCCHSELSEQ1.bit.CONV03 |= 0x3;    // selects channel A3, current
                                                  measurement in phase B main
        AdcRegs.ADCCHSELSEQ2.bit.CONV04 |= 0x4;    // selects channel A4 ; current
                                                measurement in phase B +-100 to +-200
    AdcRegs.ADCCHSELSEQ2.bit.CONV05 |= 0x5;        // selects channel A5 ; current
                                                 measurement in phase B 0 to +-200
    AdcRegs.ADCCHSELSEQ2.bit.CONV06 |= 0x6;        // selects channel A6 ;
                                                    Temperature measurement
    AdcRegs.ADCCHSELSEQ2.bit.CONV07 |= 0x7;      // selects channel A7 ;DC bus
                                                    voltage measurement
        EDIS; // Disallow writing to configuration registers.
}

void pins_init()
{

        EALLOW; // Allow writing to configuration registers.

        //Settings for the 3 hall sensors inputs
        GpioMuxRegs.GPAMUX.bit.CAP1Q1_GPIOA8=0; //Configure as I/O
        GpioMuxRegs.GPAMUX.bit.CAP2Q2_GPIOA9=0;
        GpioMuxRegs.GPAMUX.bit.CAP3QI1_GPIOA10=0;

        // Configure pins from hall sensors as digital inputs
        GpioMuxRegs.GPADIR.bit.GPIOA8 = 0;
        GpioMuxRegs.GPADIR.bit.GPIOA9 = 0;
        GpioMuxRegs.GPADIR.bit.GPIOA10 = 0;

        // Configure pins as digital inputs coming from inverter.
        GpioMuxRegs.GPAMUX.bit.T1PWM_GPIOA6=0; //Configure as I/O
```

```
        GpioMuxRegs.GPAMUX.bit.T2PWM_GPIOA7=0;
        GpioMuxRegs.GPAMUX.bit.TDIRA_GPIOA11=0;
        GpioMuxRegs.GPAMUX.bit.TCLKINA_GPIOA12=0;

        GpioMuxRegs.GPADIR.bit.GPIOA6 = 0;   // inverter too hot
        GpioMuxRegs.GPADIR.bit.GPIOA7 = 0;   // inverter phase A error
        GpioMuxRegs.GPADIR.bit.GPIOA11 = 0;  // inverter phase B error
        GpioMuxRegs.GPADIR.bit.GPIOA12 = 0;  // inverter phase C error

        //output to check computing time
        GpioMuxRegs.GPBMUX.bit.CAP4Q1_GPIOB8=0;
        GpioMuxRegs.GPBDIR.bit.GPIOB8=1;
        //output to send the relay command

        GpioMuxRegs.GPBMUX.bit.PWM10_GPIOB3=0;
        GpioMuxRegs.GPBDIR.bit.GPIOB3=1;

        // Initialize the position signals.
        switch((GpioDataRegs.GPADAT.all & 0x0700)>>8) // Get the three sensors
values here
                {
                        case 5 : position=300; break; // H1=1 H2=0   H3=1
                        case 4 : position=240; break; // H1=1 H2=0   H3=0
                        case 6 : position=180; break; // H1=1 H2=1   H3=0
                        case 2 : position=120;  break; // H1=0         H2=1   H3=0
                        case 3 : position=60;   break; // H1=0 H2=1   H3=1
                        case 1 : position=0; break; // H1=0    H2=0   H3=1
                        default: break;
                }
   last_position = position;

        EDIS; // Disallow writing to configuration registers.
}

void shut_down_relay()
{

        duty = 0;
        // Relay command = 0 relay OFF after 100 50us period
        relay_flag=1;
}

void acquire_digital_inputs()
{
        if (check_faults == 1) {
```

```
            // IO from inverter is polled twice every 50 us.  Here, we consider an input
                // to be high only if it is high both times.
        polled_io_pins = polled_io_pins1 & polled_io_pins2 & polled_io_pins3 &
                        polled_io_pins4;

                if ((polled_io_pins & 0x18C0) != 0x00) {
                    shut_down_relay();

                   // The inverter indicates at least one critical error.
                        local_critical_fault = 1;
                        // Read digital fault information from inverter.
                critical_inverter_overheat = (polled_io_pins & 0x0040)>>6;
                 critical_error_on_inverter_phaseA = (polled_io_pins & 0x0080)>>7;
                   critical_error_on_inverter_phaseB = (polled_io_pins & 0x0800)>>11;
                critical_error_on_inverter_phaseC = (polled_io_pins & 0x1000)>>12;
            }
        }

// Note that the position given is the middle position of the 60 degrees range for each step
// For example, hall1==1 && hall2==0 && hall3==1 corresponds to the range from 210
to 270 degrees so
// the given position is 240, ie the middle.
// Read hall sensors.

        raw_position = (GpioDataRegs.GPADAT.all & 0x0700)>>8;

// Filter data from hall sensors by taking position data only if 3 consecutive data are same
        if (raw_position == old_raw_position) {
           position_cnt++;
        }
        else {
           position_cnt = 0;
        }

        if (position_cnt == 2) {

                //switch((GpioD ataRegs.GPADAT.all&0x0700)>>8) // Get the three
sensors values here
                switch(raw_position) // Get the three sensors values here
                {
                        case 5 : position=300; break; // H1=1 H2=0   H3=1
                        case 4 : position=240; break; // H1=1 H2=0   H3=0
                        case 6 : position=180; break; // H1=1 H2=1   H3=0
                        case 2 : position=120;  break; // H1=0         H2=1   H3=0
                        case 3 : position=60;   break; // H1=0 H2=1   H3=1
```

```
                        case 1 : position=0; break; // H1=0    H2=0   H3=1
                        default: invalid_position_sensed = 1;
                                local_diagnostic_fault = 1;
                                break;
                }
                position_cnt=1;
        }
        old_raw_position = raw_position;

                // Check that feedback current is in safe range.
                // Don't do this check in the first 10 ms of starting.
                // 3993 in current_fb = magnitude of current is +195A
            // 2867 in current_fb = magnitude of current is +140A

        comp_time++;

        if ((connect_generator ==1) & (mem_connect_generator == 0) &
(local_noncritical_fault==0))  {

                // Turn relay on.
                GpioDataRegs.GPBDAT.bit.GPIOB3 = 1;
                comp_time = 0;
        }

        if ((comp_time <= 200))  {
        generator_starting = 1;

        }
        else  {

          generator_starting = 0;
          comp_time = 201;
        }

        if (comp_time > 60) check_faults = 1;

   if (check_faults == 1) {
            if (generator_starting == 0) {
                    if (current_fb >= 3993) { //3993

// Current over +195A for more than (0.5 <=> 500) (0.002 <=> 2)seconds,
                                    // or a count of 40= 50us periods.
                            overcurrent_instantaneous_count++;
                                    if (overcurrent_instantaneous_count >= 40) {
```

```
                                        shut_down_relay();

                                    // Any current reading over +195A.
                                    critical_instantaneous_overcurrent = 1;
                                        local_critical_fault = 1;
                                    }
                            }
                                else{
                                    overcurrent_instantaneous_count=0;
                                }
                        }
                    }
                    curr_fdb++;

    if ((curr_fdb == 20) & (check_faults == 1)) {
            if (generator_starting == 0) {

                        if (current_fb >= 3379) {
                            // Current over +165A for more than 1 seconds,
                                // or a count of 1000 50us*20 periods.
                            overcurrent_count++;
                                if (overcurrent_count >= 1000) {

                                    shut_down_relay();

                                    critical_continuous_overcurrent = 1;
                                        local_critical_fault = 1;
                                }
                            }
                            else {
                                overcurrent_count = 0;
                            }
                        }
                        curr_fdb=0;
            }
}


/*-------------------------------------------------------------

lookup_speed

Given number of 50 µsec time steps (measured by chronograph) since last position
change from Hall sensors, this function calculates the mechanical speed of generator.
```

```
------------------------------------------------------------*/
void lookup_speed (int chronograph) {

        generator_speed_history[0]=generator_speed_history[1];
        generator_speed_history[1]=generator_speed_history[2];
        generator_speed_history[2]=generator_speed_history[3];

        // Create debugging variables to watch in watch window.
        if (chronograph == 0) divide_by_zero_count++;

        // Check fault modes.
        if (check_faults == 1) {
                if (chronograph < 8) {
                   // time passed since last position change is too small to
                        // correspond to a legitimate generator speed
                        // and keep last good speed.
                   noncritical_speed_high = 1;
                        local_noncritical_fault = 1;
                        divide_by_small_count++;
                }
                else {
                   if ((chronograph > 107) & (mode == -1)) {

// time passed since last position is too large to correspond to a legitimate (non-stalled)
//generator speed. Coupling may have broken, or hall sensors may be broken.

                                shut_down_relay();
                                critical_error_in_computed_speed = 1;
                                local_critical_fault = 1;
                      generator_speed_history[3] = 0;
                   }
           }
        }

        if (chronograph < 5) {
// time passed since last position change is too small to correspond to a legitimate
//generator speed and keep last good speed.
      generator_speed_history[3] = 20000;
        }
        else {
           if (chronograph > 107) {
              generator_speed_history[3] = 0;
           }
           else {
                   // Look up the generator's mechanical speed in the table.
```

```
            generator_speed_history[3] = generator_speed_lookup[chronograph-5];
        }
    }

        if (chronograph < 68) {
            // At fast speeds, filter using four previous values.
            speed = (generator_speed_history[0]+generator_speed_history[1]
                +generator_speed_history[2]+generator_speed_history[3])>>2;
        }
        else {
            // At slow speeds, filter using only two previous values.
            speed = (generator_speed_history[0]+generator_speed_history[1])>>1;
        }
}
/*    compute_speed
Computes the mechanical speed of the generator in the rpm. The speed is re-computed
each time the Hall sensors show a change in position.  Fault mode analysis checks that
the positions reported by the Hall sensors are in the proper sequence, and that the speed
seems reasonable in that it has not changed too much from the last reported speed.
-------------------------------------------------------------*/
void compute_speed()
{
        valid_position_this_cycle = 1;

        // If the position has changed, the speed can be computed, else it remains the
same.
        if (position != last_position) {
                        // (delta position * 20000) -> to get into seconds, delta position=60
                        // (*60/360 -> rpm )/2 ->(electrical to mechanical)
                        // (/delta time) -> speed
                // Check that generator is rotating clockwise as you look into shaft.
                // (May have to modify code in car if rotation is counterclockwise.

                if ((position != 0) & (position != (last_position + 60)))
valid_position_this_cycle = 0;
                if ((position == 0) & (last_position !=300)) valid_position_this_cycle = 0;

                if ((check_faults == 1) & (speed>=1000)) {
                        if (valid_position_this_cycle == 0) {
// If position sensor value does not make sense, flag an error, and keep old position. Also,
// keep track of how many bad values we get in a row, so we can see if problem is persistent.

                                invalid_position_sequence = 1;
                                local_diagnostic_fault = 1;
                        invalid_position_count++;
```

```
                                if (invalid_position_count >= 10) {
                                        shut_down_relay();

                                critical_persistent_invalid_position_sequence = 1;
                                        local_critical_fault = 1;
                                }
                        }
                }

                if ((valid_position_this_cycle == 1) || (speed<1000)) {
                        last_position = position;
                        invalid_position_count = 0;
                }

          //  Compute the mechanical speed, and reset the chronograph on which
          // computation is based.
                lookup_speed (chronograph);
                chronograph = 0;
        }
}


void acquire_currents_and_command()
{
        /* A/D conversion --> ia,ib */

        AdcRegs.ADCTRL2.bit.SOC_SEQ1 |= 1;     // software Start-Of-Conversion
        while(AdcRegs.ADCST.bit.SEQ1_BSY==1);    // wait 'til the end of conversion

        // transformation  => [0;4095] => [-300;300] amps
        // current acquisition precision is 600/4096 = 0.146 amp
        // resulting current will be in [-2048;2047] range
        //ia_new=((signed int)((AdcRegs.ADCRESULT0>>4)&0x0FFF)-2048);
        //ia=((signed int)((AdcRegs.ADCRESULT0>>4)&0x0FFF)-2048);

        // -2048 to 2047 represents -200A to +200A
        // low resolution version used to decide whether to
        // use ia200 or ia100.
        ia=((signed int)((AdcRegs.ADCRESULT0>>4) & 0x0FFF)-50);
        ib=((signed int)((AdcRegs.ADCRESULT3>>4) & 0x0FFF)-50);

        // -2048 to 0 represents -200A to -100A and
        // 0 to 2047 represents +100A to +200A
        // high resolution version for currents with magnitude above 200A
        ia200=((signed int)((AdcRegs.ADCRESULT1>>4) & 0x0FFF)-2048-86);
```

```
   ib200=((signed int)((AdcRegs.ADCRESULT4>>4) & 0x0FFF)-2048-125);

        // -2048 to +2047 represents -100A to +100A
   // high resolution version for currents with magnitude below 200A
        ia100=((signed int)((AdcRegs.ADCRESULT2>>4) & 0x0FFF)-2100+50);
        ib100=((signed int)((AdcRegs.ADCRESULT5>>4) & 0x0FFF)-2100+50);

        // Get dc bus voltage from sensor.
        // 0 to 4095 in this variable corresponds to 0 to 2*200V.
        thisVbus=(unsigned int)((AdcRegs.ADCRESULT7>>4) & 0x0FFF);
        thisVbus=thisVbus<<1; // Multiply by 2 to compensate for voltage divider on
PCB.
   Vbus_history [history_index] = thisVbus;
        Vbus = (Vbus_history [0]
            +Vbus_history [1]
            +Vbus_history [2]
                +Vbus_history [3]
                +Vbus_history [4]
                +Vbus_history [5]
                +Vbus_history [6]
                +Vbus_history [7])>>3;

        // Get temperature from sensor.
        // 0 to 4095 in this variable corresponds to 0 to 96 deg Celsius.
        thisTemperature=(unsigned int)((AdcRegs.ADCRESULT6>>4) & 0x0FFF);
   Temperature_history [history_index] = thisTemperature;
        Temperature = (Temperature_history [0]
            +Temperature_history [1]
            +Temperature_history [2]
                +Temperature_history [3]
                +Temperature_history [4]
                +Temperature_history [5]
                +Temperature_history [6]
                +Temperature_history [7])>>3;
   history_index++;
        if (history_index == 8) history_index = 0;

        //Check DC bus voltage & inverter temperature
        volt_temp++;
        if ((volt_temp == 100) & (check_faults == 1)) {

                // Check that dc bus voltage is in safe range.
                // 4043 from ADC = 395V; 2048 on ADC = 200V
                if (Vbus >= 4043) {
```

```
                    shut_down_relay();
                critical_overvoltage = 1;
                critical_Vbus = Vbus;
                local_critical_fault = 1;
            }

                if (Vbus < 2048) {
                noncritical_undervoltage = 1;
                local_noncritical_fault = 1;
                }

        // 3448 on ADC = 80 degrees Celsius
        if (Temperature > 3448) {
                noncritical_inverter_overheat =1;
                        local_noncritical_fault=1;
                }
                volt_temp=0;
        }
}

void get_current_ref()
{

//if ((1024<=ia<3072) || (1024<=ib<3072))


 // if (((ia>=1024) & (ia<3072)) || ((ib>=1024) & (ib<3072)))
 if (current_cmd<2000)
        {

        if (mode>0)
        {
        switch(position)
        {

                case 240 :      EvaRegs.ACTRA.all=0x0410; //PWM3 active low, PWM6
                                                        active low, others off.
                                current_fb=ib100+20;    //Phase B conducts in + way, Phase C
                                                        conducts in - way
                                break;
                case 180 :      EvaRegs.ACTRA.all=0x0401; //PWM1 active low, PWM6
                                                        active low, others off.
                                current_fb=ia100+20;    //Phase A conducts in + way, Phase
                                                        C conducts in - way
                                break;
```

```
                case 120 :    EvaRegs.ACTRA.all=0x0041;  //PWM1 active low, PWM4
                                                          active low, others off.
                              current_fb=ia100+20;    //Phase A conducts in + way, Phase
                                                        B conducts in - way
                              break;
                case 60  :    EvaRegs.ACTRA.all=0x0140;  //PWM5 active low, PWM4
                                                         active low, others off.
                              current_fb=-ib100-20;   //Phase C conducts in + way, Phase B
                                                        conducts in - way
                              break;
                case 0   :    EvaRegs.ACTRA.all=0x0104;  //PWM5 active low, PWM2
                                                          active low, others off.
                              current_fb=-ia100-20;  //Phase  C conducts in + way, Phase
                                                       A conducts in - way
                              break;
                case 300 :    EvaRegs.ACTRA.all=0x0014;  //PWM3 active low, PWM2
                                                          active low, others off.
                              current_fb=ib100+20;   //Phase B conducts in + way, Phase
                                                       A conducts in - way
                              break;
                default:        break;

        }

}

else
{
        switch(position)
        {

                case 240 :    EvaRegs.ACTRA.all=0x0140;   //PWM5 active low, PWM4
                                                          active low, others off.
                              current_fb=-(ib100-20);  //Phase B conducts in - way, Phase C
                                                        conducts in + way
                              break;
                case 180 :    EvaRegs.ACTRA.all=0x0104;   //PWM2 active low, PWM5
                                                          active low, others off.
                              current_fb=-(ia100-20);  //Phase A conducts in - way, Phase C
                                                        conducts in + way
                              break;
                case 120 :    EvaRegs.ACTRA.all=0x0014;   //PWM3 active low, PWM2
                                                          active low, others off.
                              current_fb=ib100+20;   //Phase  A conducts in - way, Phase B
                                                       conducts in + way
```

```
                        break;
                case 60  :    EvaRegs.ACTRA.all=0x0410;   //PWM6 active low, PWM3
                                                            active low, others off.
                        current_fb=ib100+20;   //Phase  C conducts in - way, Phase B
                                                    conducts in + way
                                break;
                case 0   :    EvaRegs.ACTRA.all=0x0401;   //PWM1 active low, PWM6
                                                            active low, others off.
                        current_fb=ia100+20;   //Phase  C conducts in - way, Phase A
                                                    conducts in + way
                        break;
                case 300 :    EvaRegs.ACTRA.all=0x0041;   //PWM4 active low, PWM1
                                                            active low, others off.
                        current_fb=-(ib100-20); //Phase B conducts in - way, Phase
                                                    A conducts in + way
                        break;
                default:          break;
        }
        }
//      ia = 4200;
//      ib = 4200;
        }
        else if (current_cmd>=2000)
//      else if (((ia>=3072) & (ia<=4095))||(ia<1024)||((ib>=3072) &
//(ib<=4095))||(ib<1024))
        {

    if (mode>0)
        {
        switch(position)
        {

                case 240 :    EvaRegs.ACTRA.all=0x0410;   //PWM3 active low, PWM6
                                                            active low, others off.
                        current_fb=ib200+ib100-100 //  Phase B conducts in + way,
                                                        Phase C conducts in - way
                        break;
                case 180 :   EvaRegs.ACTRA.all=0x0401;    //PWM1 active low, PWM6
                                                            active low, others off.
                        current_fb=ia200+ia100-100;  //Phase A conducts in + way,
                                                        Phase C conducts in - way
                        break;
                case 120 :  EvaRegs.ACTRA.all=0x0041;  //PWM1 active low, PWM4
                                                            active low, others off.
                        current_fb=ia200+ia100-100; //Phase A conducts in + way,
```

```
                                                    Phase B conducts in - way
                        break;
                case 60  :  EvaRegs.ACTRA.all=0x0140;  //PWM5 active low, PWM4
                                                        active low, others off.
                        current_fb=-(ib200+ib100+100); //Phase C conducts in + way,
                                                        Phase B conducts in - way
                        break;
                case 0   :  EvaRegs.ACTRA.all=0x0104; //PWM5 active low, PWM2
                                                        active low, others off.
                        current_fb=-(ia200+ia100+100);  //Phase C conducts in + way,
                                                        Phase A conducts in - way
                        break;
                case 300 :  EvaRegs.ACTRA.all=0x0014; //PWM3 active low, PWM2
                                                        active low, others off.
                        current_fb=ib200+ib100-100;  //Phase B conducts in + way,
                                                        Phase A conducts in - way
                        break;
                default:        break;

        }
}
else
{
        switch(position)
        {

                case 240 :  EvaRegs.ACTRA.all=0x0140;  //PWM5 active low, PWM4
                                                        active low, others off.
                        current_fb=-(ib200+ib100+100);  //Phase B conducts in –
                                                way, Phase C conducts in + way
                        break;
                case 180 :  EvaRegs.ACTRA.all=0x0104;  //PWM2 active low, PWM5
                                                        active low, others off.
                        current_fb=-(ia200+ia100+100);  //Phase A conducts in –
                                                way, Phase C conducts in + way
                        break;
                case 120 :  EvaRegs.ACTRA.all=0x0014;  //PWM3 active low, PWM2
                                                        active low, others off.
                        current_fb=ib200+ib100-100; //Phase   A conducts in - way,
                                                        Phase B conducts in + way
                        break;
                case 60  :  EvaRegs.ACTRA.all=0x0410;  //PWM6 active low, PWM3
                                                        active low, others off.
                        current_fb=ib200+ib100-100; //Phase   C conducts in - way,
                                                        Phase B conducts in + way
```

```
                        break;
                case 0   :      EvaRegs.ACTRA.all=0x0401;  //PWM1 active low, PWM6
                                                             active low, others off.
                        current_fb=ia200+ia100-100;  //Phase  C conducts in - way,
                                                           Phase A conducts in + way
                        break;
                case 300 :      EvaRegs.ACTRA.all=0x0041;  //PWM4 active low, PWM1
                                                             active low, others off.
                        current_fb=-(ib200+ib100+100); //Phase B conducts in –
                                                        way, Phase A conducts in + way
                          break;
                default:          break;
        }

        }
//       ia = 4200;
//       ib = 4200;
        }

}

void compute_PI_PWM()
{

        // activate the right PWM depending on the sector
        get_current_ref();

        if((current_cmd!=0) & (local_noncritical_fault == 0) & (connect_generator==1))
{

                // PI controller

                error= current_cmd - current_fb;

                // Kp and Ki must be set to match the PWM period range
                // Sample Period is assumed to be included into the Ki constant
                sum_error+=((1 * error)>>8); //ki=2, 5
                if(sum_error>sum_current_limit)sum_error=sum_current_limit;
                if (sum_error<-10) sum_error=800;
                if((sum_error<0) & (current_cmd>2000))sum_error=1000;
    if ((error>320) & (error<=800)) error = (error>>1); //1
    if ((error>800) & (error<=1600)) error = (error>>2); //2
    if ((error>1600) & (error<=6400)) error = (error>>3); //3, 2400
                //duty=Kp*(error) + Ki*(sum_error);
                if ((error<0) & (current_cmd>2000)) error=0;
```

```
                duty= error + sum_error;

                // duty cycle saturation
                if(duty > (PWM_PERIOD-125))
                                duty= PWM_PERIOD-125;
                if(duty < 0)
                                duty= 0;

                // Modify the actual duty cycles
                CMPR1=duty;
                CMPR2=duty;
                CMPR3=duty;

        }
        else{
                duty = 0 ;
                CMPR1=CMPR2=CMPR3=0;
                sum_error=0;

        }
}

void variables_init()
{
        speed=0;
        CMPR1=0x0000;
        CMPR2=0x0000;
        CMPR3=0x0000;
        EvaRegs.COMCONA.bit.FCOMPOE = 1;  // Enable the PWMs.
        chronograph=0;
        torque_cmd=0;

}

void main()
{
        PWM_init();
        ADC_init();
        variables_init();
        pins_init();
       InitECan();

        // Relay command = 0 relay Off
```

```
        GpioDataRegs.GPBDAT.bit.GPIOB3 = 0;

  interrupt_init();
        first1=0;
        first2=0;
        relay_open_delay=0;

       while(1);
}

interrupt void T2PINT_ISR(void)     // EV-A
{
        GpioDataRegs.GPBSET.bit.GPIOB8=1;  //set this output to high to display the
                                                begining of the computation(test purpose)
        chronograph++;
                polled_io_pins1 = GpioDataRegs.GPADAT.all;
                acquire_digital_inputs();       // 0-10 microsec
                polled_io_pins2 = GpioDataRegs.GPADAT.all;
                acquire_currents_and_command();    // 20-30 microsec
                polled_io_pins3 = GpioDataRegs.GPADAT.all;
                compute_speed();                                        // 0-10 microsec
                polled_io_pins4 = GpioDataRegs.GPADAT.all;
                compute_PI_PWM();                                       // 0-10 microsec
                ACQ[index_ACQ++]=CMPR1;                                 // duty graph

           receive++;
           transmit++;
           transmit1++;
           rolling_cnt++;

                /*if((abstemp==1) & (mode==1) & (time_cmd<6000))
                {
                        time_cmd++;
                        //400ms <=> 8000 of 50us period  1s <=> 20000  300ms <=> 6000
200ms <=> 4000 250ms <=> 5000
                        current_cmd= 150*20;
                }
                else{
                        current_cmd=0;
                        time_cmd=8001;//20001
                }*/

        if(relay_flag==1)
        {
                        if(relay_open_delay==20)
```

```
                {
                        GpioDataRegs.GPBDAT.bit.GPIOB3 = 0;
                        relay_flag=0;

                }
            relay_open_delay++;

    }


if (ECanaRegs.CANRMP.bit.RMP2 == 1)
  {
        // ECanaRegs.CANRMP.all |= 0x0004;
            ECanaRegs.CANRMP.bit.RMP2 = 1;

            mem_connect_generator = connect_generator;
            msg2_1 = (unsigned int)ECanaMboxes.MBOX2.MDL.byte.BYTE1;

            connect_generator = (msg2_1 & 0x80)>>7;
            clear_faults = (msg2_1 & 0x40)>>6;

            if(connect_generator == 0)  {
                        check_faults = 0;
                        shut_down_relay();
            }
            // Clear all fault flags.

            if (clear_faults == 1)  {
                        relay_open_delay=0;
                        relay_flag=0;
                        local_critical_fault = 0;
                        local_noncritical_fault = 0;
                        local_diagnostic_fault = 0;
                        four_bit_rolling_count = 0;
                        critical_communications_with_UALAN_lost = 0;
                        critical_error_in_computed_speed = 0;
                        critical_instantaneous_overcurrent = 0;
                        critical_continuous_overcurrent = 0;
                        critical_overvoltage = 0;
                        critical_persistent_invalid_position_sequence = 0;
                        noncritical_undervoltage = 0;
                        noncritical_speed_high = 0;
                        noncritical_inverter_overheat = 0;
                        invalid_position_sequence = 0;
                        invalid_position_sensed = 0;
```

```
                critical_inverter_overheat = 0;
                critical_error_on_inverter_phaseA = 0;
                critical_error_on_inverter_phaseB = 0;
                critical_error_on_inverter_phaseC = 0;

                   }

                 msg2_1 = (unsigned int)ECanaMboxes.MBOX2.MDL.byte.BYTE1;
                compt=(msg2_1) & 0x0F;
                // Should check this rolling count from the SCM, but didn't get there,

                msg2_2=(unsigned int)ECanaMboxes.MBOX2.MDL.byte.BYTE2;
                msg2_3=(unsigned int)ECanaMboxes.MBOX2.MDL.byte.BYTE3;
                temp = msg2_3 - ((msg2_2 & 0x01) << 8);

    if (temp < 0)
            { mode=-1;              // to know we are in generating mode
      abstemp = (unsigned int)(-temp);    // to get rid of the bit that indicates the sign
                                                of the current
            }
    else
            { mode=1;

            abstemp=temp;               // to know we are in motoring mode

            }
             current_cmd= abstemp*20;
             receive=0;
            // critical_communications_with_UALAN_lost = 0;   //for lab test
}
   else if((receive == 1000) && (ECanaRegs.CANRMP.bit.RMP2 == 0))
                                                                    //  1000=50ms
     {
            shut_down_relay();
            critical_communications_with_UALAN_lost = 1;
            local_critical_fault = 1;

            receive=0;
     }

   if (transmit == 2000)  {
     // every 100 ms

     if ((ECanaRegs.CANTA.bit.TA0==1) || (first1==0)) {
              // Don't expect a transmission ack the first time.
```

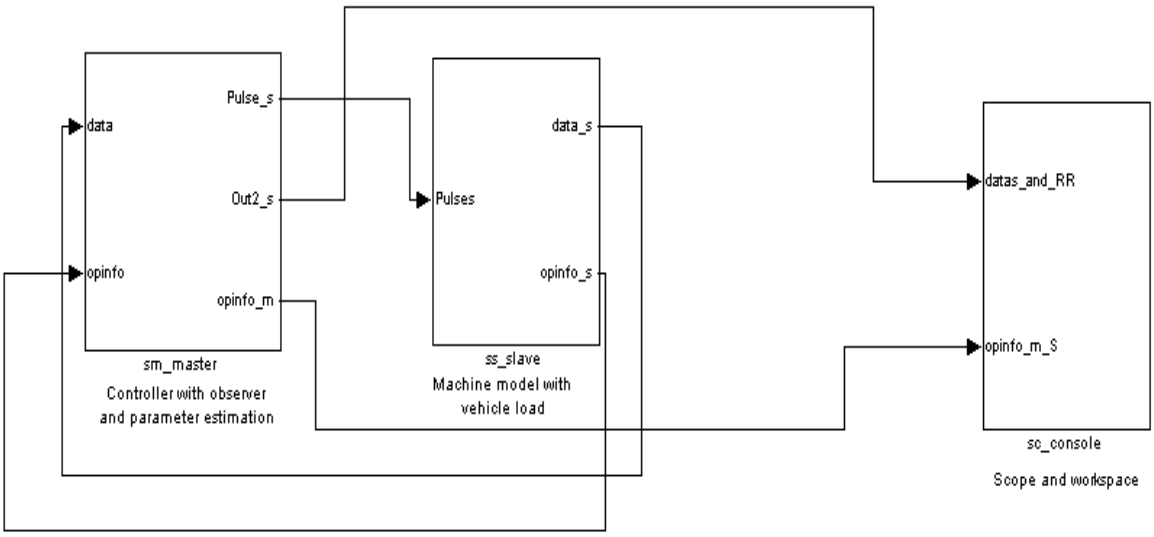
```
                // Prepare four bits of rolling count.
                four_bit_rolling_count = rolling_cnt & 0x00F0;
                four_bit_rolling_count = four_bit_rolling_count >> 4;

                msg0_1 = 0;
                        // never have a system critical fault; leave top bit of byte to zero.
                msg0_1 = msg0_1 | ((local_critical_fault<<6) & 0x0040);
                msg0_1 = msg0_1 | ((local_noncritical_fault<<5) & 0x0020);
                msg0_1 = msg0_1 | ((local_diagnostic_fault<<4) & 0x0010);
                msg0_1 = msg0_1 | (four_bit_rolling_count & 0x000F);
                msg0_2 = 0;
                msg0_2= msg0_2 | ((critical_communications_with_UALAN_lost<<7) &
0x0080);
                msg0_2= msg0_2 | ((critical_inverter_overheat<<6) & 0x0040);
                msg0_2= msg0_2 | ((critical_error_on_inverter_phaseA<<5) & 0x0020);
                msg0_2= msg0_2 | ((critical_error_on_inverter_phaseB<<4) & 0x0010);
                msg0_2= msg0_2 | ((critical_error_on_inverter_phaseC<<3) & 0x0008);
                msg0_2= msg0_2 | ((critical_error_in_computed_speed<<2) & 0x0004);
                msg0_2= msg0_2 | ((critical_instantaneous_overcurrent<<1) & 0x0002);
                msg0_2= msg0_2 | (critical_continuous_overcurrent & 0x0001);
                msg0_3 = 0;
                msg0_3= msg0_3 | ((noncritical_undervoltage<<7) & 0x0080);
                msg0_3= msg0_3 | ((critical_overvoltage<<6) & 0x0040);
                msg0_3= msg0_3 | ((critical_persistent_invalid_position_sequence<<5) &
0x0020);
                msg0_3= msg0_3 | ((noncritical_speed_high<<4) & 0x0010);
                msg0_3= msg0_3 | ((noncritical_inverter_overheat<<3) & 0x0008);
                msg0_3= msg0_3 | ((invalid_position_sequence<<2) & 0x0004);
                msg0_3= msg0_3 | ((invalid_position_sensed<<1) & 0x0002);

                                // Clear non-critical and diagnostic fault flags.
                                noncritical_speed_high = 0;
                                noncritical_inverter_overheat = 0;
                                noncritical_undervoltage= 0;
                                invalid_position_sequence = 0;
                                invalid_position_sensed = 0;
                                local_noncritical_fault = 0;
                                local_diagnostic_fault = 0;

                // Move data bytes into CAN engine, mailbox 0.
        ECanaMboxes.MBOX0.MDL.byte.BYTE1= msg0_1;
        ECanaMboxes.MBOX0.MDL.byte.BYTE2= msg0_2;
        ECanaMboxes.MBOX0.MDL.byte.BYTE3= msg0_3;
        //ECanaRegs.CANTRS.bit.TRS0 = 1;  //set flag in transmit request register
```

```
            ECanaShadow.CANTRS.all = 0;
            ECanaShadow.CANTRS.bit.TRS0 = 1; // Set TRS for mailbox under test
            ECanaRegs.CANTRS.all = ECanaShadow.CANTRS.all;

            ECanaShadow.CANTA.all = 0;
            ECanaShadow.CANTA.bit.TA0 = 1; // Clear TA0
            ECanaRegs.CANTA.all = ECanaShadow.CANTA.all;
                        }
      first1=1;
      transmit=0;
   }

   if (transmit1 == 200)  // every 10 ms
                {

                 if ((ECanaRegs.CANTA.bit.TA1==1) || (first2==0))

                   {
                        msg1_1 = 0x0000;

                                // Fill in mode of operation in top two bits of first byte.
                                switch (mode) {
                                   case 0:  break;
                                   case -1: msg1_1 = msg1_1 | 0x0080; break;
                                        case +1: msg1_1 = msg1_1 | 0x0040; break;
                                        default: break;
                                }
                                // if emergency shut down, force 11 in these two bits.
        if (local_critical_fault != 0) msg1_1 = msg1_1 | 0x00C0;
// Fill in 9-bit current feedback in bottom bit of first byte, and all of second byte.

if (mode == -1) sn_current_ref = -(current_fb>>4);  // Sending signed bit for current_ref
            else  sn_current_ref = (current_fb>>4);
                                msg1_1 = (sn_current_ref & 0x0100)>> 8;
                                msg1_2 = sn_current_ref & 0x00FF;

                // Fill in 16-bit speed in bytes 3 and 4.
                                msg1_3 = (speed & 0xFF00)>>8;  // MSB 2 bits
                                msg1_4= speed & 0x00FF;

                // Fill in 8-bit dc bus voltage in byte 5.
                                msg1_5=(Vbus>>4) & 0x00FF;   // Voltage bus info.:400V in 8bit

                // Fill in 8-bit temperature in byte 6.
        msg1_6 = (Temperature>>4) & 0x00FF; // Temperature info :95deg in 8bit
```

```
ECanaMboxes.MBOX1.MDL.byte.BYTE1= msg1_1 ;
   ECanaMboxes.MBOX1.MDL.byte.BYTE2= msg1_2;
   ECanaMboxes.MBOX1.MDL.byte.BYTE3= msg1_3;
   ECanaMboxes.MBOX1.MDL.byte.BYTE4= msg1_4;
   ECanaMboxes.MBOX1.MDH.byte.BYTE5= msg1_5;
            ECanaMboxes.MBOX1.MDH.byte.BYTE6= msg1_6;

   ECanaShadow.CANTRS.all = 0;
ECanaShadow.CANTRS.bit.TRS1 = 1; // Set TRS for mailbox under test
ECanaRegs.CANTRS.all = ECanaShadow.CANTRS.all;
ECanaShadow.CANTA.all = 0;
ECanaShadow.CANTA.bit.TA1 = 1; // Clear TA0
ECanaRegs.CANTA.all = ECanaShadow.CANTA.all;
                }

else
            {
                transmit_cnt++;
                }

          first2=1;
          transmit1=0;

          }

if (transmit_cnt==5)
  {

          shut_down_relay();


          critical_communications_with_UALAN_lost = 1;
          local_critical_fault = 1;

          transmit_cnt=0;
          }

   GpioDataRegs.GPBCLEAR.bit.GPIOB8=1; //set this output to low to display the
end of the computation(test purpose)

   EvaRegs.EVAIFRB.bit.T2PINT = 0x0001; // clear interrupt flag
   PieCtrlRegs.PIEACK.all = PIEACK_GROUP3; // acknowledge this interrupt
}
```

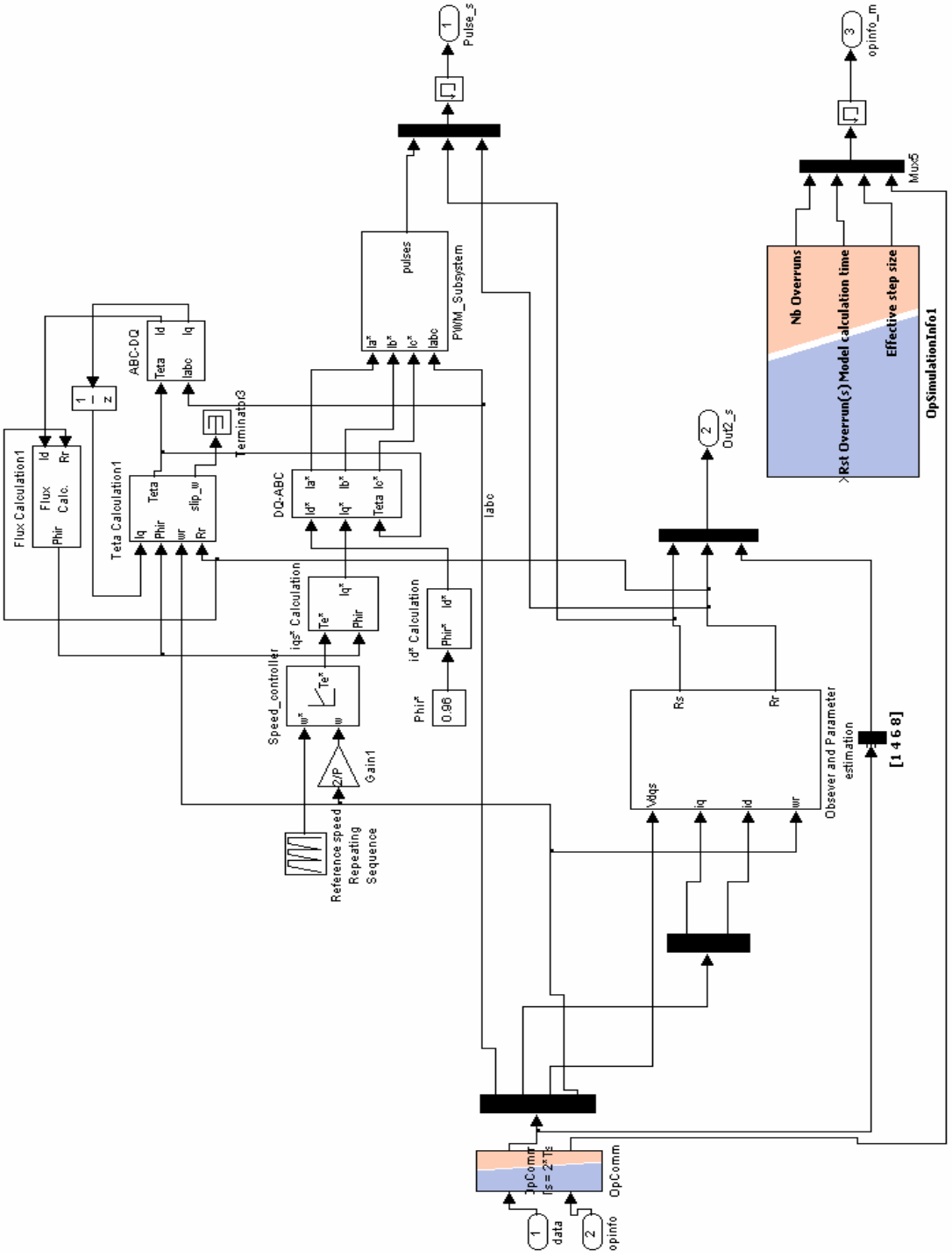
APPENDIX C

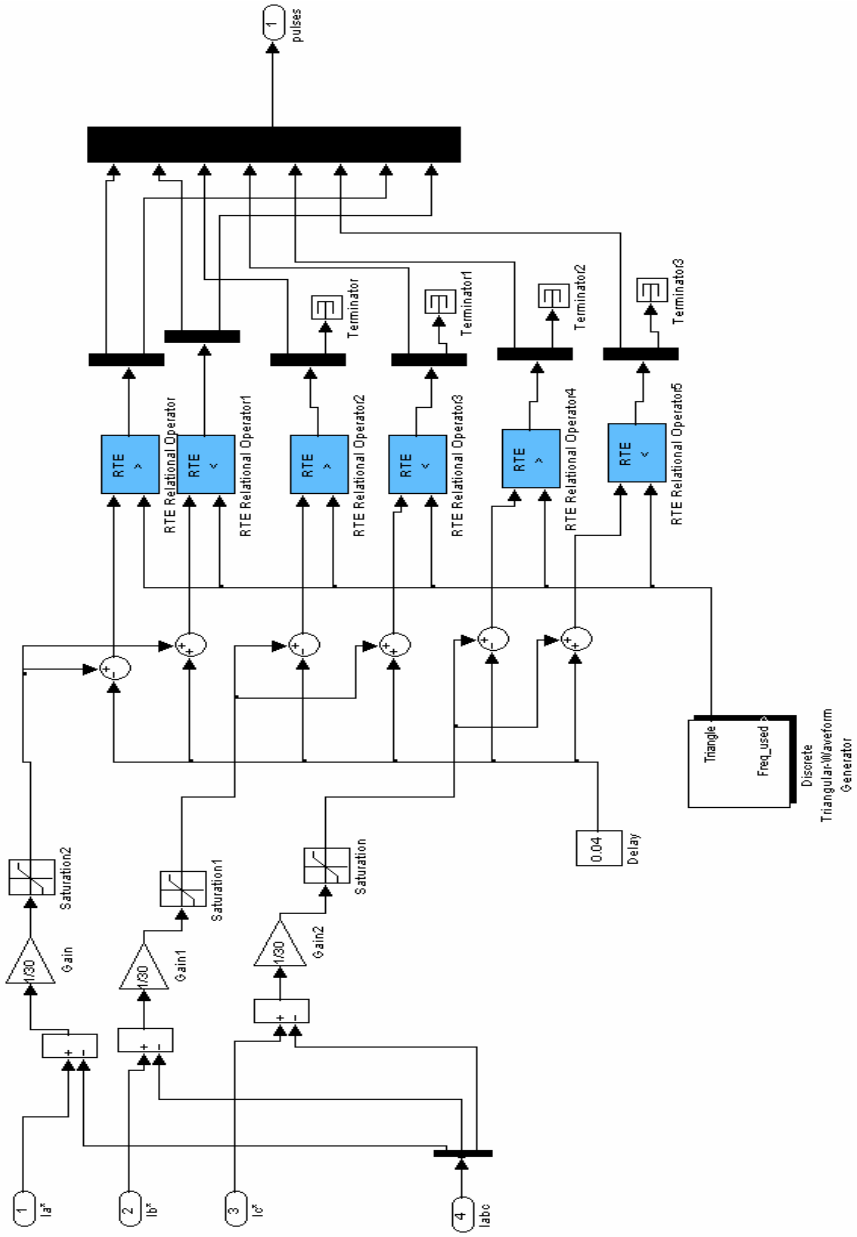
SIMULINK MODEL OF INDUCTION MOTOR DRIVE FOR HIL SIMULATION

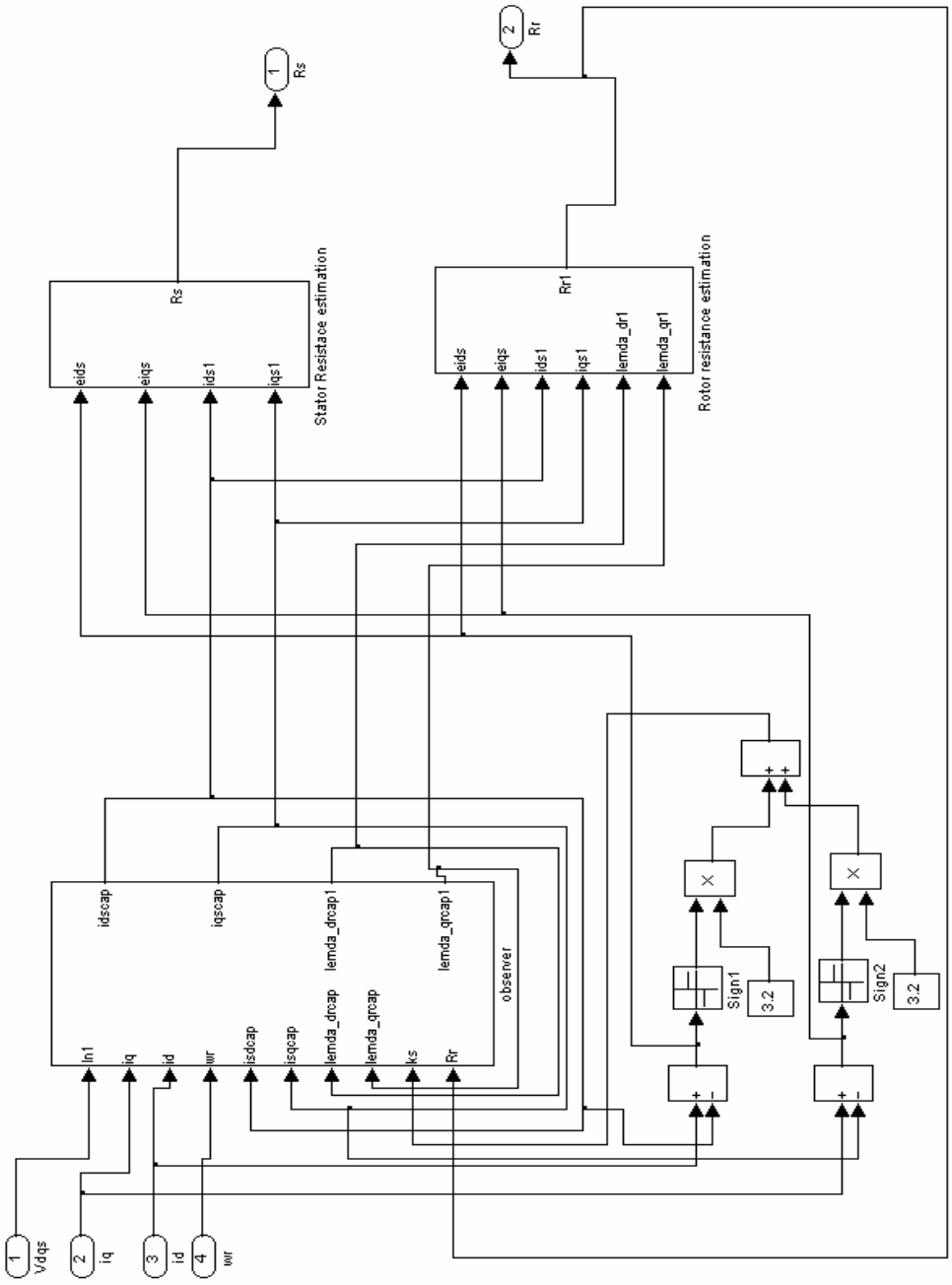


CONTROLLER AND INDUCTION MOTOR MODEL WITH ELECTRIC VEHICLE LOAD

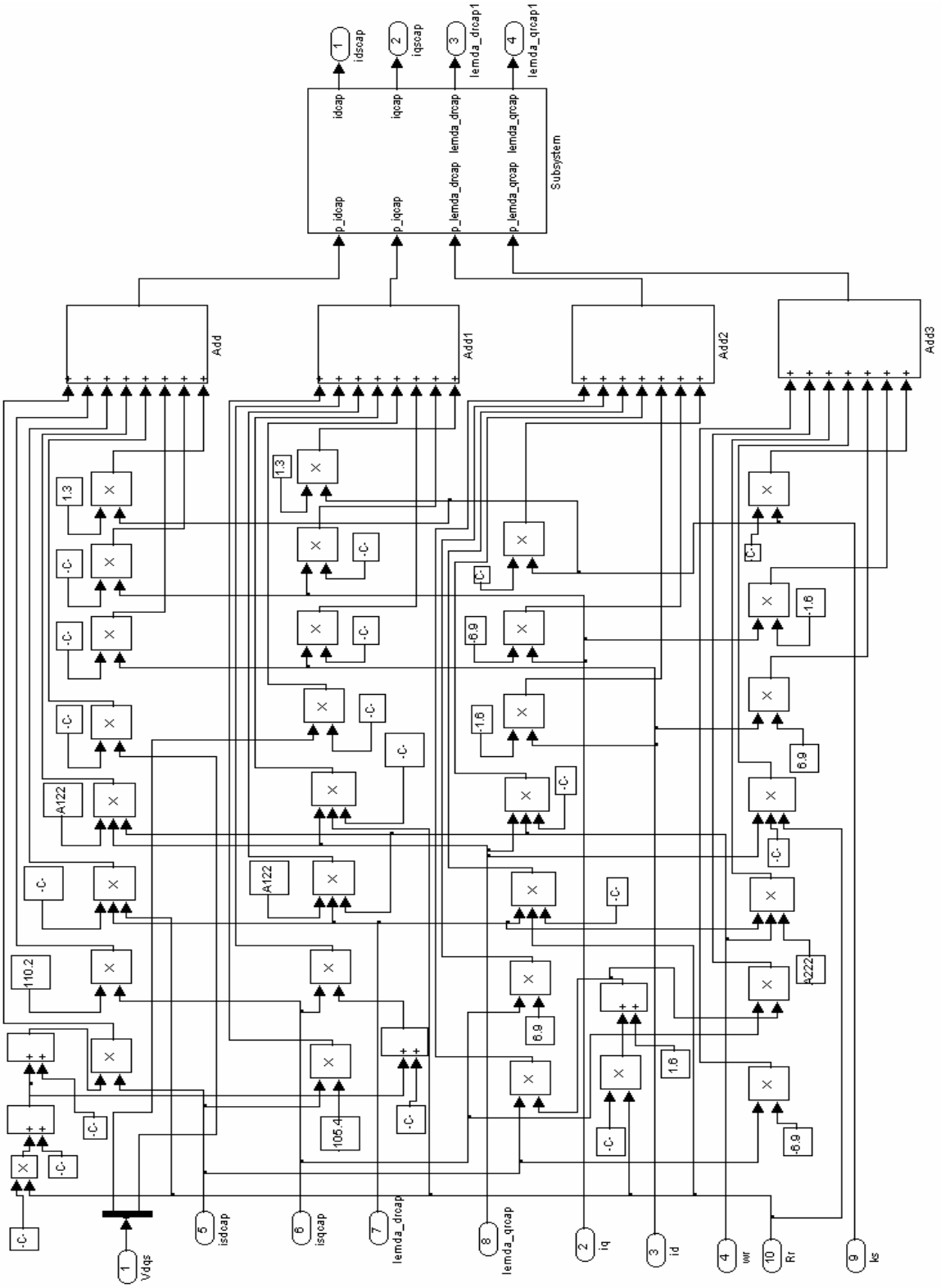
INDUCTION MOTOR CONTROLLER WITH LUENBERGER-SLIDING MODE OBSERVER BASED PARAMETER ADAPTATION ALGORITHM




1
pulses
Terminator
Terminator1
Terminator2
Terminator3
RTE >
RTE <
RTE Relational Operator
RTE Relational Operator1
RTE Relational Operator2
RTE Relational Operator3
RTE Relational Operator4
RTE Relational Operator5
Triangle
Freq_used
Discrete
Triangular-Waveform
Generator
0.04
Delay
Saturation2
Saturation1
Saturation
1/30
Gain
Gain1
Gain2
1
Ia*
2
Ib*
3
Ic*
4
Iabc


1
Rs
2
Rr
Rs
eids
eiqs
ids1
iqs1
Stator Resistace estimation
Rr1
eids
eiqs
ids1
iqs1
lemda_dr1
lemda_qr1
Rotor resistance estimation
idscap
iqscap
lemda_drcap1
lemda_qrcap1
observer
In1
iq
id
wr
isdcap
isqcap
lemda_drcap
lemda_qrcap
ks
Rr
Sign1
3.2
Sign2
3.2
1
Vdqs
2
iq
3
id
4
wr

LUENBERGER-SLIDING MODE OBSERVER



INDUCTION MOTOR MODEL WITH ELECTRIC VEHICLE LOAD AND REAL TIME LIBRARIES FOR DIGITAL & ANALOG I/O INTERFACE