# ROBUST ADAPTIVE CONTROL OF INDUCTION MACHINES USING THE $\mathcal{L}_1$ CONTROL SCHEME

BY

PAUL MICHAEL RANCURET

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Professor Philip T. Krein

# ABSTRACT

In this thesis, a robust adaptive control scheme is proposed in order to drive an induction motor with high performance, even in the presence of unknown machine electrical parameters. Magnetic saturation within an induction motor results in nonlinear, changing flux dynamics. Also, rotor resistance changes with temperature as the machine is heated due to normal use, resulting in more parametric uncertainty. The proposed control scheme will cause the motor to track a desired reference model behavior, even in the presence of these unknown parameters. An adaptive control scheme known as $\mathcal{L}_1$ adaptive control is used to ensure that this reference model behavior is realistic, so that robustness may be maintained even with very fast adaptation. This controller is limited by the flux observer used to provide information about flux magnitude and direction.

*To my wife, for believing in me, loving me, and supporting me throughout this research*

# ACKNOWLEDGMENTS

First and foremost, I would like to thank my adviser, Professor Philip T. Krein. He has given me the autonomy to research what interests me, along with the support and guidance to develop it. Also, I would like to thank Professor Naira Hovakimyan for sparking my interest in robust adaptive control theory. My gratitude goes out to Professor Daniel Liberzon as well, for deepening my understanding of adaptive control theory in general.

I am also indebted to many of my colleages here at the University of Illinois, especially those in the Power and Energy Systems Group. Thank you Kevin Colravy, Ali Bazzi, and Tutku Büyükdeğirmenci for all the time spent together in the lab. My thesis would not be possible without your indispensable help and support in implementing my research. The entire Power and Energy Systems Group has greatly helped increase my knowledge of power systems.

I also want to thank all of my family for their love and support throughout my time here at the University of Illinois. Specifically, I would like to thank my parents Vince and Carrie Rancuret, my brother Keith Rancuret, and my grandparents for all your belief in me. Not least among my family is my wife Sarah Rancuret, who has been wonderfully understanding and helpful to me. I would not be where I am today without her love and support.

Finally, I would like to show my gratitude towards all my friends, including Taylor Johnson, Rakesh Gopchandani, Alan Gostin, Brian Proulx, Seth Baker, Amanda Hartman, Bernhard Baur, Greer Williams, Jackson Marusarz, Frances Jarret, Anna Reik, Allison Stefaniak, Helen Lau, Lourdes Trunner, Melanie Meinzinger, Sanna Stegmaier, and many more. You have all helped to make my life wonderful, and given me many happy memories.

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

The induction machine is very commonly used in industry today, due to its low price, high reliability, and good performance. However, it is relatively hard to control, due to nonlinearity and coupling of flux and torque dynamics. Field-oriented control is widely used as the preferred control method for an induction machine, as it allows a decoupling of these torque and flux dynamics. It is desirable to use feedback to linearize the system in such a way that the dynamics mimic that of a simple DC motor.

Field-oriented control achieves its objective by representing all state variables in a new, two-axis reference frame. This reference frame is aligned with the rotor flux vector, so that it rotates along with the rotor magnetic field. Doing so simplifies the machine dynamics greatly, and allows for feedback linearization.

The main problem that remains with induction machine control is uncertainty of parameters. First of all, it is hard to precisely measure machine electrical parameters such as rotor resistance, rotor inductance, mutual inductance, etc. These are generally measured using tests on the machine off-line. However, even if these measurements were perfectly accurate while the machine is off-line, they change during operation of the machine. For example, magnetic saturation of the machine adds extra terms in the relation between current and flux, which could be viewed as a changing inductance. Also, as the machine is operated, it heats up. This change in temperature alone causes large variation in the rotor and stator resistances.

It is often convenient to also consider mechanical variables such as load torque and rotational inertia as unknown parameters. In many applications, the load on the machine changes very rapidly, and over a wide range.

These parameter variations greatly affect the accuracy of flux estimation techniques and control techniques for the induction motor. In this thesis, it is the control technique which is examined. Some flux observers which appear

in the literature are discussed as examples of how to improve accuracy of flux estimation techniques in lieu of parameter uncertainty. However, the primary goal of this thesis is to address the performance of the actual control laws under these uncertainties.

First, a field-oriented controller using the model reference adaptive control (MRAC) scheme is proposed and tested in Chapter 3. It is shown that this controller will cause the actual system to behave like a desired reference system. However, this type of controller assumes that perfect cancellation of the uncertain parameter estimation errors can occur. This is only the case when the form of the machine model is perfectly accurate, albeit with uncertain parameter values.

A more realistic approach is taken with the $\mathcal{L}_1$ controller proposed in Chapter 4. The approach is very similar to the MRAC controller, except that the model reference system is chosen as a more realistic model. This time, it is only assumed that cancellation of error terms may occur at lower frequencies, since there may be unmodeled fast dynamics in the system. Because of this, the controller will not expend unnecessary control effort to compensate for high frequency unmodeled dynamics. This results in a much more robust system, and allows for arbitrarily fast tuning rates without sacrificing robustness. Doing so allows the user to safely increase the performance of the controller.

This $\mathcal{L}_1$ control scheme is then implemented on an induction motor system, using an inverter to source the desired currents. It is shown that even with fast tuning of unknown parameters, this adaptive controller produces smooth, low frequency control inputs.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Nonlinearity and Uncertainty of Induction Motors

The induction machine has been studied widely, as it poses a challenging and rewarding control problem. The induction machine is desirable due to its rugged, inexpensive construction, and its performance. However, it is nonlinear due to magnetic saturation, and difficult to control (at least compared to a DC motor or synchronous machine).

In [1], Charles R. Sullivan and Seth R. Sanders develop a nonlinear $\pi$ model for an induction machine, which operates in magnetic saturation. The basic form of this magnetic model of one tooth-pair is shown in Figure 2.1. They claim this $\pi$ model of induction motor magnetics is convenient for field-oriented control designs which account for magnetic saturation. This model assumes a nonlinear invertible symmetric monotonic curve for the $B$-$H$ characteristics of the induction motor core material, as seen in Figure 2.2. First, a model is derived which describes the magnetic circuit of one tooth pair between a stator and rotor. Next, a model of a smooth stator/rotor pair is obtained by considering an infinite number of infinitesimally small teeth, integrated around the circumference of the machine. Finally, the Blondel-Park transformation is used (see [2]) to formulate a more complete model of an induction motor, using the $\pi$ model of nonlinear magnetics. This model is verified by thorough testing on an actual motor.

In this thesis, the more standard T circuit model is used for the induction motor. This model may be found in [3]. However, the model proposed by Sullivan and Sanders could be used as well, in order to explicitly represent saturation. As it turns out, the adaptive control scheme would be almost identical in either case. The only difference would be that parameter values would be different.

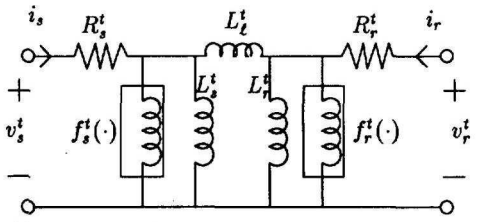


Figure 2.1: $\pi$ model of induction machine magnetics, as seen in [1]

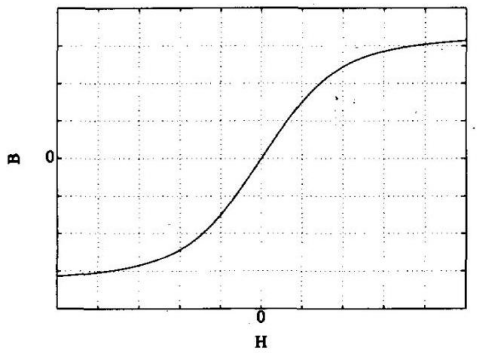


Figure 2.2: $B$-$H$ saturation characteristic as seen in [1]

Sullivan et al. have proposed control solutions for induction machines which may operate in the saturation region in [4]. These control systems use the $\pi$ magnetic model of an induction machine, which is also described in [1]. This magnetic model is used, along with the Blondel-Park transformation (see [2]), to develop machine electrical and torque dynamic equations, in the same manner as in [4]. One control scheme which is approached in [4] is field-oriented control, using the models derived from the $\pi$ magnetic circuit. An important thing to note about this model is that the inductances are represented by linear parts, $L_S$, $L_R$, and $L_l$, and the nonlinear parts are represented in parallel by a function $F_S(\vec{\lambda}_S)$. An example of this function is given in [1]. With this control, the motor is modeled in the rotating rotor flux frame. Flux vectors are used as state variables, with stator flux being viewed as a control variable. In this arrangement, torque and rotor flux control remain decoupled, even with the nonlinear magnetics model. Rotor flux magnitude is controlled by adjusting the component of stator flux aligned with rotor flux, while torque is regulated by adjusting the orthogonal component of stator flux. The dynamic equations describing the induction

motor which are used in field-oriented control, as described in [4], are

$$\dot{\rho} = \frac{R_R}{L_l}\frac{\lambda^e_{sq}}{\lambda^e_{rd}} + \dot{\theta} \tag{2.1}$$

$$\dot{\lambda}^e_{rd} = -R_R\left[\left(\frac{1}{L_R}+\frac{1}{L_l}\right)\lambda^e_{rd} - \frac{1}{L_l}\lambda^e_{sd} + f_r(\lambda^e_{rd})\right] \tag{2.2}$$

$$\dot{\vec{\lambda}}^e_S = -\dot{\rho}\mathbf{J}\vec{\lambda}^e_S + \vec{v}^e_S - R_S\left\{\left(\frac{1}{L_S}+\frac{1}{L_l}\right)\vec{\lambda}^e_S - \frac{1}{L_l}\begin{bmatrix}\lambda^e_{rd}\\0\end{bmatrix} + F_S(\vec{\lambda}^e_S)\right\} \tag{2.3}$$

$$\dot{\omega} = \frac{1}{J_{\text{inertia}}}\left(\frac{1}{L_l}\lambda^e_{sq}\lambda^e_{rd} - \tau_{\text{load}}\right) \tag{2.4}$$

$$\dot{\theta} = \omega \tag{2.5}$$

In (2.1), $\rho$ represents the angle of the rotor flux vector $\vec{\lambda}^s_r$ with respect to the stationary stator reference frame. The superscript $e$ in any variable denotes that it is in the rotor field-oriented reference frame. In the case of any flux linkage $\lambda$, the subscript $s$ or $r$ tells whether it is a stator or rotor flux linkage, the subscript $q$ means it is the vector component orthogonal to the rotor flux vector, and the subscript $d$ means it is the vector component aligned with the rotor flux vector. The function $f_r(\cdot)$ is the function which approximates the $B$-$H$ characteristics of the rotor, with the function $F_s(\cdot)$ being a scaled version of stator $B$-$H$ curve $f_s(\cdot)$. This scaling is defined by in [4] on page 143, and is given as

$$F(\vec{\lambda}) = f(\|\vec{\lambda}\|)\frac{\vec{\lambda}}{\|\vec{\lambda}\|} \tag{2.6}$$

The effects of nonlinear magnetics are further approached by Hofmann et al. in [5]. While most induction motor controllers assume linear magnetics for simplicity, Hofmann et al. present a stator-flux-based vector control which works well for machines operating in magnetic saturation, which cannot be described by linear magnetics. Since this controller accounts for magnetic saturation, higher transient torque can be achieved from the induction motor by temporarily operating within saturation. First, a $\pi$ model of the magnetic circuit is developed using the methods outlined in [1]. Using this model, a torque control is outlined which only requires knowledge of stator voltage, stator current, and stator resistance. Then, torque limits are examined, showing that this controller allows an induction motor to perform smoothly at four times rated continuous torque. The controller is then implemented

and tested experimentally.

## 2.2 Control of Induction Motors

Vector control, or field-oriented control, is widely used as the preferred method of regulating speed and flux for an induction motor. This is due to the fact that flux and torque control may be decoupled through the use of field-orientation. Here, the machine states are represented in a coordinate frame which rotates to maintain alignment with the rotor flux field.

In [6], W. Leonhard gives several methods for controlling an induction motor on pages 241-302. These methods follow the machine model developed on pages 164-176. The complete mathematical model of a general symmetrical AC machine is described on page 175 of [6]. The equations presented by Leohnard are formulated with stator and rotor currents as the primary state variables. In this thesis, however, the stator and rotor fluxes will be used as the state variables, with the stator current being an input to the system.

The goal of field-oriented control then becomes to maximize the torque produced by aligning the reference frame with the rotor flux field. Thus, the reference frame rotation will force the rotor flux to be orthogonal to the torque-producing component of the stator current. Once this field-orientation is achieved, there are many different control laws which may be used to achieve flux and torque regulation.

Kao and Liu analyze and test several different induction machine vector control schemes in [7]. In that paper, a microcontroller-based system is used to test and compare the different methods. The control methods used are all field-oriented control schemes, similar to the one proposed in this thesis. Namely, an $H_\infty$ control scheme is tested and validated. This control method aims to reduce the sensitivity of the system to disturbance signals by minimizing the $\mathcal{L}_2$ norm of the transfer function mapping these disturbances to the output. In the case of the induction motor, this disturbance is generally the load torque imposed on the machine.

In [8], a nonlinear control method is proposed in order to track a desired reference speed with unknown torque input. This field-oriented controller uses nonlinear, time-varying control gains to track the desired speed. This controller has online estimation of the unknown load torque, but assumes the

electrical parameters of the motor are exactly known.

## 2.3 Flux Observers for Induction Machines

In order to achieve the decoupling goal of field-orientation, it is necessary to know information about the magnetic flux inside the machine. In practice, it is impractical to obtain an actual real-time measurement of this flux. This would require putting magnetic sensors inside the induction machine, which negates the desirable simplicity and robustness of the motor. Therefore, observers are used in practice, in order to estimate the flux magnitude and direction in real time. Methods of using flux observers in field-oriented control are discussed in [9].

Perhaps the simplest type of flux observer is the flux simulator. This type of observer is simply a real-time simulator of the machine flux dynamics, and requires exact knowledge of the electrical parameters of the machine in order to be accurate.

Verghese and Sanders examine flux estimation for induction machines from the viewpoint of observer theory in [10]. Essential well known observer theory is first presented, then compared with existing AC motor flux estimating methods. Then, a method of observing flux is proposed which uses prediction error, so that the observer can be measured and analyzed by its error dynamics. They go on to show that this estimation scheme provides much faster parameter convergence than methods which do not use prediction error as feedback, such as a simple flux simulator. Simulations are given which compare this method to previously used methods of flux estimation. Further work on this topic is done in [11] and [12].

Adaptation is used in many cases in order to greatly decrease the flux observer's sensitivity to error in estimated electrical parameters. This is perhaps the greatest challenge to overcome when designing a flux observer.

Qinghua Zhang proposed a new method for designing adaptive observers in [13]. He shows that it guarantees global exponential stability in the absence of noise, and bounded errors if the noise signals are bounded with zero means. The paper focuses on multiple-input-multiple-output (MIMO) linear time-varying (LTV) systems. In order to adaptively estimate both the unknown parameters and the system states, the system states are separated

into portions affected by the unknown parameters, and portions affected by known signals. The latter portion is easily estimated. Adaptive laws are then derived to estimate the first portion of state, as well as the unknown parameters. Robustness analysis is then performed showing system behavior with noise, and a numerical example is given.

Further work on this principle has been done by Jadot et al. to present an adaptive regulator for an induction motor in [14]. They claim this controller guarantees robustness, while estimating the unknown electrical parameters of a motor accurately. The regulator is based on field-oriented control, with the exception that motor parameters are treated as unknowns, with adaptation laws to determine their values. Two methods of adaptation are proposed. The first method relies on slow adaptation, so that there is a time-scale separation between system dynamics and adaptation dynamics. The second method proposed, which follows the method in [13], allows for faster adaptation, but introduces extra states to the controller. A sensitivity analysis of parametric adaptation is provided, which shows that under certain operating conditions, the controller will remain robust.

An adaptive tuning controller is also developed in [15] and [16], which accounts for unknown torque load and rotor resistance. The full seven-state adaptive observer will converge to true values of torque and resistance, as well as the true rotor fluxes, as long as the system satisfies the persistency of excitation condition. This condition requires that the desired flux and speed that the motor is tracking contain enough frequency components. Thus, if the motor is tracking a constant speed and constant flux, the parameters will not converge. Also, this system assumes that the machine will never be operated near saturation. Lyapunov's method is used to prove stability of the system with this observer and control scheme. In this thesis, we do not develop an adaptive flux observer, but only adapt the control scheme to get uniform performance when flux is known, regardless of electrical parameters. Thus, the presented controller here is limited only by the performance of the flux observer. An adaptive flux observer such as the one proposed in [15] and [16] could be used in conjunction with the control scheme proposed in this thesis to achieve better performance.

Another field-oriented controller is developed with on-line tuning of motor electrical parameters in [17]. This control system uses a rotor flux simulator for flux feedback in the controller. The reactive power being used to drive the

machine is used in order to identify the leakage inductance. Also, equations are given to identify the magnetizing inductance and rotor time constant in real time using measurements of reactive power. These parameter estimates are tuned using not an integrator-based tuning law, but rather an algebraic equation. These real-time identified parameters are then used in a standard indirect field-oriented controller. This differs greatly from the control scheme proposed in this thesis, in which the parameter estimates are not directly identified. In this thesis, a reference model behavior is given, and parameters are tuned using integral tuning laws to ensure that the motor behavior tracks that of the desired reference model. The flux simulator used in this thesis is non-adaptive, so it will have error due to incorrect parameter estimates there. Further work could be done to use the identifiers presented in [17] to increase the accuracy of the flux observer used in this thesis.

Often, it is desirable to estimate speed as well as flux using an observer, in order to eliminate the need for a speed sensor. A speed-sensorless field-oriented controller is proposed in [18] which uses an adaptive flux and speed observer. The flux observer is a full-state observer, which uses rotor and stator resistance values from an adaptive identifier. The speed estimator uses integral and proportional terms to converge to the true speed value. The speed estimator is developed in more detail in [19]. These adaptive schemes are proven to converge using Lyapunov analysis, and simulations are given to verify the findings. The observers proposed in [18] could be used to improve the control scheme proposed in this thesis. Here, a non-adaptive flux simulator is used to estimate the rotor flux, which is prone to estimation error when parameters are unknown. Additionally, the adaptive speed observer proposed in [18] could be used to create a speed-sensorless version of the controller proposed in this thesis.

## 2.4 Adaptive Control of Induction Machines

This thesis focuses on the use of adaptation in the control of the machine, rather than in the accuracy of the flux observer. Several adaptive control techniques have been proposed for this application.

For instance, work has been done on a simple yet robust adaptive controller for field-oriented control of induction motors in [20] by Moreira et al. In that

paper, rotor flux is estimated by examining the third harmonic, which is a result of the flattening of the sine wave, due to magnetic saturation. This estimation is used to measure rotor flux error and develop a slip frequency correction factor using a deadbeat control model. This controller uses an approximate inverse model of the induction machine to calculate the slip correction. It is shown that when the controller is detuned (the model has incorrect values for flux linkage), the estimated parameters are adjusted to compensate for the model error. This adjustment happens as a function of the error in $d$ and $q$ components of rotor flux. This controller is then implemented and tested, showing excellent system response for a wide range of operating speeds.

A very interesting output feedback control system for an induction motor is proposed in [21]. In this scheme, output feedback from the three measurable states (direct and quadrature current, and speed) is used directly in a nonlinear control system to regulate the speed of an induction machine. This control system takes advantage of the natural stable modes of an induction machine model to create a stable tracking closed-loop scheme. No state observer is used in the control scheme proposed in [21]. In [22], that control system is further developed in order to increase the convergence rate of the system to the desired speed. Since the original control system relied on the stable damping modes of the induction motor model, the convergence rate was dependent on the rotor time constant of the machine. However, this drawback is addressed in [22] by filtering the speed error signal, so that it is possible to inject a mechanical damping term into the dynamics. Further, it is shown that the standard indirect field-oriented controller is a special case of the proposed control scheme. The concept of using a stable low-pass filter in the control law for robustness is used in the $\mathcal{L}_1$ control system proposed in [23]. However, the control schemes are quite different, since the adaptive $\mathcal{L}_1$ controller has the goal of tracking a desired reference model, and uses a flux observer instead of direct output feedback. Further work may be possible to improve the controller proposed in this thesis by using an output feedback scheme similar to the one given in [21] and [22].

Adaptive control schemes have been designed and developed for other motor applications as well. In [24], an adaptive $H_\infty$ controller is developed for permanent magnet synchronous motor drives. The PM synchronous motor is simpler to control than the induction motor, since the rotor flux angle is fixed

in relation to the rotor shaft angle. The $H_\infty$ control scheme involves minimizing the $\mathcal{L}_2$ norm of the system which determines the effect of disturbance signals on the output. In [24], the adaptive controller is made by running a real-time simulation of the motor characteristics, where the electrical parameters are at their nominal values. Adaptive laws are developed to cause the actual machine to behave like the model. The error between actual and nominal electrical parameters is treated as a disturbance signal, along with the error in load torque and inertia. The transfer function between these errors and output error is then described, and its $\mathcal{L}_2$ norm is minimized. The concept of adapting in order to achieve this nominal performance in lieu of unknown parameters is similar to the adaptive controller proposed in this thesis. However, the $\mathcal{L}_1$ adaptive controller proposed in this thesis addresses robustness by relaxing the assumption that the system model is of the correct form. Instead, the $\mathcal{L}_1$ controller only assumes the system model is accurate at lower frequencies, allowing for high frequency unmodeled dynamics. Thus, if there are unmodeled fast, stable high frequency dynamics in the actual motor, the controller will not expend extra control effort (which may cause destabilization) to correct for these unmodeled dynamics. One similarity between the $H_\infty$ and $\mathcal{L}_1$ control schemes is that they both minimize an induced transfer function norm to prove stability.

In [25], a model reference adaptive control scheme is proposed for control of an induction motor, based on direct field-oriented control. The algorithm is separated into three adaptive loops: the speed control loop, the $d$-axis flux loop, and the $q$-axis flux loop. Adaptive laws for each loop are derived and analyzed using standard Lyapunov methods. Each adaptive law is composed of both an integral and proportional adaptive term. With this scheme, a reference model for desired behavior is given for each of the three control loops. The Lyapunov methods for analyzing each control loop assume perfect cancellation of parameter estimation errors, so that the reference model may be followed exactly. This assumption may be relaxed using the $\mathcal{L}_1$ controller proposed in this thesis, resulting in a more robust system.

Hovakimyan describes a new type of adaptive controller in [26]. This controller, referred to as an $\mathcal{L}_1$ adaptive controller, is able to provide very fast adaptation of unknown parameters while maintaining robustness. It relies on a combination of Lyapunov-based analysis, along with the small gain theorem (see [27]) to guarantee stability. This controller is unique because faster

adaptation (increasing the adaptive gains in the adaptation laws) actually improves robustness, unlike most adaptive controllers. In fact, the gain and phase margins are shown to be bounded away from zero, which guarantees robustness no matter how fast the adaptation takes place.

Cao and Hovakimyan propose and analyze the novel $\mathcal{L}_1$ robust adaptive controller in [23]. It is shown that the $\mathcal{L}_1$ control strategy guarantees transient performance within explicit bounds, even with unmodeled dynamics or nonlinearities. This thesis makes use of that control strategy in order to achieve high performance without sacrificing robustness.

# CHAPTER 3

# MODEL REFERENCE ADAPTIVE DIRECT FIELD-ORIENTED CONTROLLER

In this chapter, an adaptive controller will be developed using a model reference adaptive controller (MRAC) for a current-fed induction motor model. This general control procedure is well known in the literature, and establishes a basic adaptive control system with the goal of tracking desirable dynamic behavior. In this chapter, the system is developed in detail in order to provide a framework for further robustness analysis. Shortcomings of this system will be discussed, so that they may be addressed in later chapters.

This proposed controller will use a direct field-oriented control method. First, the dynamic model used for this control will be discussed. Then, the MRAC controller will be developed, simulated, and discussed. Simulations will be given to support claims about the proposed control method.

The MRAC approach to adaptive control has been used extensively in the literature for many different applications. The approach taken in this chapter closely resembles the form of controller proposed by Goléa and Goléa in [25]. In that paper, three separate adaptive control loops are used in order to achieve field orientation, regulate the flux, and regulate the speed of the motor. This thesis also uses three MRAC controllers to accomplish the same goals, although each individual controller is developed slightly differently than the corresponding one in [25]. Wang et al. also propose an MRAC controller for an induction motor drive in [28]. However, the control system proposed by Wang is used to address parameter uncertainty for a voltage fed induction motor, in order to achieve good transient performance. In this thesis, we use a current fed induction motor model, so that the voltage to current dynamics are ignored.

The MRAC controllers developed in this thesis use an indirect[1] MRAC ap-

[1]Note the use of the words "direct" and "indirect" in different contexts. The field orientation method used in this chapter is a direct FOC scheme. The adaptive approach used to regulate each control loop is an indirect MRAC scheme.

proach. Each control loop closely follows the general case for indirect MRAC, which is outlined in [26]. With indirect MRAC, the adaptive parameters are tuned in order to estimate the values of each unknown parameter. These estimates are then used in a control law which would stabilize the system with the desired transient performance if the estimates are accurate. It is important to note that it is not actually necessary that the tuned parameters converge to the true values of the unknown parameters. They only need to converge to values which cause the system to perform similarly to the specified reference dynamics. The approach taken in [25] is slightly different, as it uses a direct MRAC scheme. This approach directly adapts the control gains, rather than developing the control laws based on parameter estimates. For better performance, Goléa uses both proportional and integral terms in the control law, and adaptively adjusts the gains in order to track the desired system performance.

The simulations presented in this chapter demonstrate the performance of this control scheme under various conditions. It is shown that the system will stabilize even with abrupt changes in the machine electrical parameters. The conditions for these tests are the same as those used to test Goléa's MRAC system under electrical parameter changes in [25]. However, this thesis also performs additional tests of the indirect MRAC controller with the presence of fast, stable, unmodeled dynamics. These tests point out robustness issues of the MRAC system which are not addressed in [25].

## 3.1 Dynamic Model

Consider the dynamics for an induction motor, as modeled by Krause in [3]. The equations used to model the motor are

$$\dot{\omega}_m = -\frac{f}{J}\omega_m + \frac{1}{J}\left\{\frac{3PL_m}{2L_r}(\lambda_{dr}i_{qs} - \lambda_{qr}i_{ds}) - \tau_{load}\right\} \tag{3.1}$$

$$\dot{\lambda}_{dr} = \frac{-R_r}{L_r}\lambda_{dr} + \omega_{sl}\lambda_{qr} + \frac{R_r L_m}{L_r}i_{ds} \tag{3.2}$$

$$\dot{\lambda}_{qr} = \frac{-R_r}{L_r}\lambda_{qr} - \omega_{sl}\lambda_{dr} + \frac{R_r L_m}{L_r}i_{ds} \tag{3.3}$$

$$\dot{\lambda}_{ds} = -R_s i_{ds} + \omega\lambda_{qs} + V_{ds} \tag{3.4}$$

$$\dot{\lambda}_{qs} = -R_s i_{qs} - \omega\lambda_{ds} + V_{qs} \tag{3.5}$$

Table 3.1: State Variable Meanings

| Variable | Meaning |
|---|---|
| $\omega_m$ | rotor angular frequency [rad/sec] |
| $\lambda_{rd}$ | rotor d-axis flux [Wb-s] |
| $\lambda_{rq}$ | rotor q-axis flux [Wb-s] |
| $i_{qs}$ | stator q-axis current [A] |
| $i_{ds}$ | stator d-axis current [A] |
| $\omega_{sl}$ | slip frequency ($\omega - P\omega_m$) [rad/sec] |
| $\omega$ | reference frame frequency [rad/sec] |

Table 3.2: Parameter Meanings and Range of Values

| Variable | Meaning | Nominal | Range of Estimate | Initial Guess |
|---|---|---|---|---|
| $P$ | number of pole pairs | 2 | $P = 2$ | 2 |
| $f$ | rotor viscosity [$N$-$m$-$s$] | 0.0003 | $1.5 \times 10^{-4} \leq f \leq 4.5 \times 10^{-4}$ | $3.2{\times}10^{-4}$ |
| $J$ | rotor inertia [$kg$-$m^2$] | 0.005 | $2.5 \times 10^{-3} \leq J \leq 7.5 \times 10^{-3}$ | $3.2{\times}10^{-3}$ |
| $R_r$ | rotor resistance [$\Omega$] | 3.3 | $1.65 \leq R_r \leq 4.95$ | 2.5 |
| $L_r$ | rotor inductance [$H$] | 0.375 | $0.188 \leq L_r \leq 0.562$ | 0.25 |
| $L_m$ | mutual inductance [$H$] | 0.34 | $0.17 \leq L_m \leq 0.51$ | 0.3 |
| $a$ | $f/J$ | 0.05 | $0.02 \leq a \leq 0.18$ | 0.1 |
| $b$ | $1/J$ | 200 | $133 \leq b \leq 400$ | 312.5 |
| $m$ | $\frac{3}{2}PL_m/L_r$ | 2.72 | $0.9 \leq m \leq 8.15$ | 3.6 |
| $\tau_{\text{load}}$ | load torque | 5 | $-10 \leq \tau_{\text{load}} \leq 10$ | 2.24 |
| $\mu$ | $bm$ | 544 | $119.7 \leq \mu \leq 3260$ | 1125 |
| $\sigma$ | $-b\tau_{\text{load}}$ | -1000 | $-4000 \leq \sigma \leq 4000$ | -700 |
| $\alpha$ | time constant $R_r/L_r$ | 8.8 | $2.94 \leq \alpha \leq 26.32$ | 10 |
| $\beta$ | $\alpha L_m$ | 2.992 | $0.5 \leq \beta \leq 13.42$ | 3 |

See Table 3.1 for information about the meaning of each state variable, and Table 3.2 for information about the meaning and range of values for each parameter. Some reasonable bounds can be given *a priori* for the parameter values.

These equations describe the equivalent circuit model of an induction motor as seen in Figure 3.1. In this figure,

$$L_{ls} = L_s - L_m,\ L'_{lr} = L_r - L_m,\ J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

$$V_s = \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix},\ V'_r = \begin{bmatrix} v_{dr} \\ v_{qr} \end{bmatrix},\ \lambda_s = \begin{bmatrix} \lambda_{ds} \\ \lambda_{qs} \end{bmatrix},\ \text{and } \lambda'_r = \begin{bmatrix} \lambda_{dr} \\ \lambda_{qr} \end{bmatrix}.$$

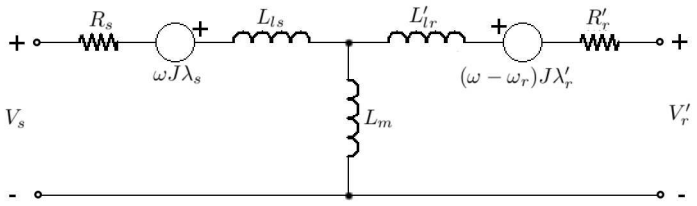


Figure 3.1: Equivalent T circuit model of induction motor

This is the most commonly used equivalent circuit to describe an induction motor. Note that all values on the rotor side of the circuit are marked with a prime.

Equations (3.4)–(3.5) give the dynamics for the stator flux with respect to stator current. In this controller, we will assume a current-fed model. This means that we use a fast actuator to force the current to its commanded value, without worrying about stator flux. In practice, this is common and achievable with the use of fast switching and a high DC bus voltage. Thus, for Equations (3.1)–(3.3), we can treat the current values $i_{ds}$ and $i_{qs}$ as inputs to the system.

In this model, we will consider all the parameters $f$, $J$, $\tau_{load}$, $L_m$, $L_r$, and $R_r$ as unknown, possibly slowly time-varying quantities. We can simplify our model by ignoring Equations (3.4)–(3.5) and grouping the unknown quantities together into new parameters. The new system representation then becomes

$$\dot{\omega}_m = -a\omega_m + \mu(\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}) + \sigma \tag{3.6}$$

$$\dot{\lambda}_{dr} = -\alpha\lambda_{dr} + \omega_{sl}\lambda_{qr} + \beta i_{ds} \tag{3.7}$$

$$\dot{\lambda}_{qr} = -\alpha\lambda_{qr} - \omega_{sl}\lambda_{dr} + \beta i_{qs} \tag{3.8}$$

The newly introduced parameters $a$, $\mu$, $\sigma$, $\alpha$, and $\beta$ are related to the old unknown parameters as shown in Table 3.2. These five parameters become our unknown values in the system.

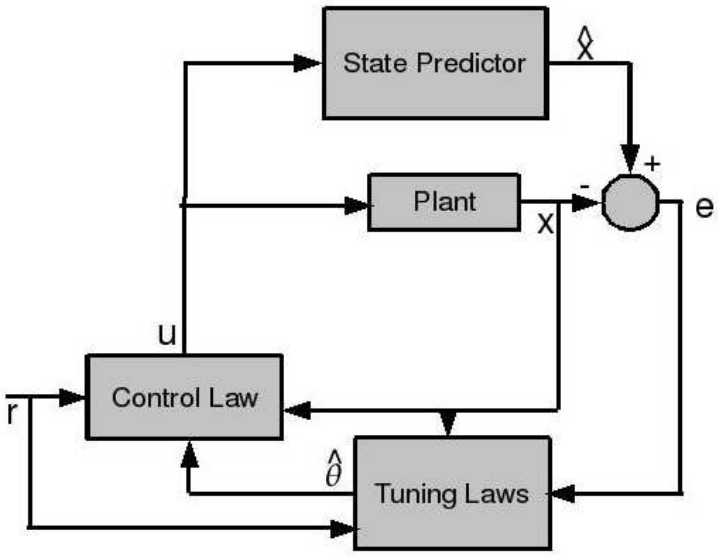


Figure 3.2: Structure of model reference adaptive controller

## 3.2 Overview of Model Reference Adaptive Control

A very common and useful method for adaptive control of dynamic systems is the model reference adaptive controller (MRAC). In [25], N. Goléa and A. Goléa develop an MRAC controller very similar to the proposed controller in this thesis.

The basic structure of an MRAC system can be seen in Figure 3.2. In this scheme, the control engineer is faced with a dynamic model for the plant with one or more unknown parameters. The engineer then specifies a dynamic model which describes the behavior he wishes the plant to have, called the reference model. The actual plant dynamics are then rewritten in such a way that the the unknown parameters now represent the difference between the plant and the reference model. The objective is to estimate these unknowns in real time, and update the control laws in such a way that the plant will behave dynamically the same as, or close to, the reference model. Usually the reference model is a simple, stable linear system with a desirable and reasonable convergence rate. In [27], H.K. Khalil develops and describes the general MRAC problem well.

To achieve this, a so called "state predictor" is used to predict the state response of the plant, essentially by running a real-time simulation of the plant with unknown parameters replaced by the current guess of their values.

This state prediction is denoted $\hat{x}$ in Figure 3.2. Often, the state predictor is referred to as a "state observer," although this thesis avoids that terminology since its role is very different from an observer in the more traditional sense. An error signal $e$ is then generated as the difference between the predicted state and the measured (or observed) state of the actual plant. This error signal, along with the plant state $x$ and desired tracking input $r$, is used by the tuning laws to generate a "guess" of the unknown parameters $\theta$ (the parameter guesses are denoted $\hat{\theta}$ in Figure 3.2). The goal of this subsystem is to create a "guess" for the unknown parameters which causes the tracking error signal to diminish. This would indicate that the state predictor is behaving similarly to the plant. Finally, the control law is developed to get the state predictor model to mimic the desired reference model. Since the state predictor model is known at all times (it depends on the parameter guesses $\hat{\theta}$, rather than the actual unknown $\theta$), this is a reasonable control objective. As the state predictor and plant dynamics become similar enough through the use of the tuning parameters, the actual plant dynamics will eventually follow the desired reference behavior as well.

## 3.3 Brief Overview of Field-Oriented Control

In this thesis, we will use adaptive control in a field-oriented control setting for an induction motor. The goal of field-oriented control is to simplify the machine dynamics and control by aligning the $d$-$q$ frame axis with the rotor flux. Thus, the rotor flux vector is only present in the direct axis as $\lambda_{dr}$, leaving the quadrature axis rotor flux $\lambda_{qr} = 0$. This results in dropping the $-\lambda_{qr} i_{ds}$ term of (3.1), which would have had a negative impact on torque production. Thus, the direct axis stator current $i_{ds}$ may be used as an input to control the rotor flux $\lambda_{dr}$ using the dynamics in Equation (3.2), without having any direct effect on torque production. Effectively, this decouples torque or speed control from flux control, allowing the controller much more flexibility. Several different examples of field-oriented controllers can be found in [3].

It is important to note that the controllers developed in this section are based on the *direct* field-oriented control technique. This means that the stator and rotor fluxes are treated as measurable, known states, and that

they are available signals to use in control. In reality, it seldom makes sense to use a magnetic flux sensor inside an induction machine. This means that in practice, one must use a flux observer to obtain these states for use in control. For simulation purposes, the controllers in this section use flux states directly from the machine model.

## 3.4 Development of MRAC for Induction Motor

For the induction motor system, recall Equations (3.6)–(3.8). Here, we will have three separate control loops to regulate the dynamics of each of these three equations. For the first loop, we treat the slip frequency $\omega_{sl}$ as an input to the dynamics in Equation (3.8), with the objective of regulating $\lambda_{qr}$ to zero. This gives us our field-orientation. In the next control loop, the direct axis stator current $i_{ds}$ is used as an input to the flux dynamics (3.7), with the objective of regulating $\lambda_{dr}$ to some desired value. This value can be used to optimize the efficiency of the machine, or to make it possible for the machine to produce extra torque. Note that as long as we have $\lambda_{qr} \longrightarrow 0$, the slip frequency $\omega_{sl}$ does not affect the direct rotor flux dynamics in steady state. Finally, our last control loop will use the quadrature axis stator current $i_{qs}$ as an input to (3.6) in order to regulate the speed or torque to some desired value. Again, since we have $\lambda_{qr} = 0$, the $-\lambda_{qr} i_{ds}$ term in (3.6) drops out, maximizing torque production and simplifying control. Note that $i_{qs}$ also enters (3.8). In that control loop, $i_{qs}$ will be treated as a known disturbance signal multiplied by an unknown parameter.

### 3.4.1 Quadrature Axis Rotor Flux Control Loop

Recall the quadrature axis rotor flux dynamics as given in Equation (3.8) on page 16. Our goal is to develop a controller which will regulate $\lambda_{qr}$ to zero by commanding the slip frequency $\omega_{sl}$ with an adaptive feedback law. Our unknown parameters are $\alpha$ and $\beta$. For this controller, we will treat $\lambda_{dr}$ and $i_{qs}$ as *known* disturbance signals, since they can be measured.[2]

[2] $\lambda_{dr}$ is generally not directly measured in practice. However, a good estimate may be obtained in real-time using a flux observer. See section 2.3 for more details.

We desire the flux $\lambda_{qr}$ to asymptotically converge to zero. Thus, a simple stable linear system is chosen as the reference model, and is given as

$$\dot{\lambda}_{qr,m} = \alpha_m \lambda_{qr,m}. \tag{3.9}$$

Here, $\alpha_m$ is the rate of exponential convergence, and must be a negative number for the model reference system to be stable. Now, we can rewrite the actual dynamics as follows:

$$\dot{\lambda}_{qr} = \alpha_m \lambda_{qr} - (\alpha_m + \alpha)\lambda_{qr} - \omega_{sl}\lambda dr + \beta i_{qs} \tag{3.10}$$

Here, we have simply added and subtracted the term $\alpha_m \lambda_{qr}$ in order to have our desired dynamics appear in the actual dynamic equation. Finally, by denoting the difference in real and desired convergence rates as $\theta_q = -(\alpha_m + \alpha)$, we can simplify the dynamics to the following form:

$$\dot{\lambda}_{qr} = \alpha_m \lambda_{qr} - \omega_{sl}\lambda_{dr} + \beta i_{qs} + \theta_q \lambda_{qr} \tag{3.11}$$

Now, the dynamics have been written in a form which assigns the unknown parameters to represent the differences between our actual dynamic system and our desired reference model. It is very important to note that this can only be done if the desired reference model shares a common form with the actual system. For instance, if the dynamic system was of second order, it would be impossible to represent the difference between that system and our first order desired system with any constant parameter. Thus, a matching assumption has been made, requiring that there exists a value $\theta_q$ which causes the rewritten system to match the original dynamic system. In the case that the dynamics in (3.8) are exactly correct, then this assumption is valid. However, in the presence of unmodeled dynamics (such as ignored higher order terms) the matching assumptions are no longer valid [29].

With the system rewritten, one can easily find the control law which would produce the desired behavior if all parameters were known. For a moment, consider the situation where we do know all parameters. If this was the case, we could choose the nominal feedback law

$$\omega_{sl,nom} = \frac{1}{\lambda_{dr}}(\beta i_{qs} + \theta_q \lambda_{qr}) \tag{3.12}$$

Substituting this feedback law into the system dynamics in (3.11) cancels all right-hand terms other than $\alpha_m \lambda_{qr}$, leaving us with our desired reference behavior, as given in (3.9).

Unfortunately, we cannot directly use this control law, since the parameter values $\beta$ and $\theta_q$ are unknown. Instead, we must try our own best guess of parameters which will result in a working control, and update these parameters in real-time until they succeed in causing the system to track the reference behaviors. We will denote our guess of $\beta$ as $\hat{\beta}_q$ and our guess of $\theta_q$ as $\hat{\theta}_q$. In order to track how well these adaptive parameters are working, we will use the state predictor

$$\dot{\hat{\lambda}}_{qr} = \alpha_m \hat{\lambda}_{qr} - \omega_{sl} \lambda_{dr} + \hat{\beta}_q i_{qs} + \hat{\theta}_q \lambda_{qr} \tag{3.13}$$

This state predictor is exactly the same as the system dynamics from Equation (3.11) except that the unknown parameters are replaced by their respective estimates, and the predicted state is denoted $\hat{\lambda}_{qr}$ in the state predictor.[3] The difference between the predicted and measured (or observed) quadrature flux state is denoted $e_q$, and is given

$$e_q = \hat{\lambda}_{qr} - \lambda_{qr} \tag{3.14}$$

The state predictor represents what the estimated dynamic behavior of the system should be based on the current estimate of unknown parameters. If we are able to update the estimated parameters in such a way that the state predictor tracks the actual system dynamics, we can then consider a new intermediate control objective of regulating the *state predictor* using a feedback control law. Thus, if the state predictor is a good model of the true plant dynamics, the actual system will be regulated with the same control input as the state predictor.

Regulating the state predictor to behave as the model reference system is a simple task compared to regulating the unknown plant. If we replace the unknown parameters in our nominal control law from Equation (3.12) with

[3]Note that the measured/observed state $\lambda_{qr}$ is used in the last term of Equation (3.13), not the predicted state $\hat{\lambda}_{qr}$.

their respective estimates, we get the implementable[4] control law

$$\omega_{sl} = \frac{1}{\lambda_{dr}}(\hat{\beta}_q i_{qs} + \hat{\theta}_q \lambda_{qr}) \tag{3.15}$$

To ensure that this control law remains bounded, steps must be taken to ensure that $\lambda_{dr}$ never approaches zero. In reality, this flux never would be zero, because there is remnant flux inside the machine. Thus, the initial condition will be bounded away from zero. Additionally, in order to produce torque, the desired flux $\lambda_{dr}$ will always be non-zero. It is therefore possible to implement a small bound away from zero for the flux $\lambda_{dr}$ in this control law, ensuring that $\omega_{sl}$ always remains bounded. Substituting Equation (3.15) into the state predictor in Equation (3.13) yields exactly the reference model dynamics given in Equation (3.9).[5]

Now that it is known how to get the state predictor model to behave exactly as the reference model, we are left with the more difficult task of making the state predictor track the behavior of the actual unknown plant dynamics. In short, our goal is to get $e_q \longrightarrow 0$, while guaranteeing that all other signals remain bounded. We denote the parameter estimation errors $\tilde{\beta}_q$ and $\tilde{\theta}_q$ as shown in

$$\tilde{\beta}_q = \hat{\beta}_q - \beta \tag{3.16}$$

$$\tilde{\theta}_q = \hat{\theta}_q - \theta_q \tag{3.17}$$

Note that it is not necessary that these parameter estimation errors converge to zero; it is only necessary that they remain bounded. In [30] it is shown that such parameter estimates do converge to stabilizing values, even if they do not converge to the correct values. To find tuning laws for the parameter estimates which will satisfy our goals, we will use Lyapunov's direct

[4]This new control law (Equation (3.15)) is implementable because it only contains estimates of parameters, instead of using the actual unknown parameter values.

[5]For this reason, it is possible in practice to run a real-time simulation of the reference model and use its state as $\hat{\lambda}_{qr}$ instead of using a state predictor.

method. Taking the derivative of the error $e_q$ in Equation (3.14) yields

$$\begin{aligned}\dot{e}_q &= \dot{\hat{\lambda}}_{qr} - \dot{\lambda}_{qr} \\ &= \alpha_m \hat{\lambda}_{qr} - \omega_{sl}\lambda_{dr} + \hat{\beta} i_{qs} + \hat{\theta}_q \lambda_{qr} \\ &\qquad - \alpha_m \lambda_{qr} + \omega_{sl}\lambda_{dr} - \beta i_{qs} - \theta_q \lambda_{qr} \\ &= \alpha_m e_q + \tilde{\beta}_q i_{qs} + \tilde{\theta}_q \lambda_{qr} \end{aligned} \tag{3.18}$$

Before we can continue with the adaptive control design, we must be sure that a solution will even exist for the dynamic system in question. A condition which guarantees the existence and uniqueness of a solution is the Lipschitz continuity condition. If, for a system

$$\dot{x}(t) = f(t, x(t)), \qquad x(0) = x_0,$$

the function $f(t, x(t))$ is said to be locally Lipschitz, then a unique solution is guaranteed to exist for the system [27]. Definition 1 describes the Lipschitz continuity condition, and may also be found in [27].

**Definition 1** (Locally Lipschitz Function)**.** *A function $f(t, x(t))$ is said to be locally* Lipschitz *if for any $\delta > 0$, there exists $L_\delta > 0$ and $B > 0$ such that*

$$|f(t, x(t)) - f(t, y(t))| \leq L_\delta \|x(t) - y(t)\|_\infty, \qquad |f(0, t)| \leq B,$$

*for all $\|x(t)\|_\infty \leq \delta$ and $\|y(t)\|_\infty \leq \delta$ uniformly in $t$.*

In the special case of a scalar system, this condition is satisfied if the function $f(t, x(t))$ always has a uniformly bounded derivative with respect to time, uniformly in $t$. For our system, the plant dynamics in Equation (3.11), state predictor model of Equation (3.13), and error dynamics in Equation (3.18) are all represented by Lipschitz functions as long as the derivatives $\dot{\tilde{\beta}}_q$, $\dot{\tilde{\theta}}_q$, $\dot{\omega}_{sl}$, $\dot{\lambda}_{dr}$, and $\dot{i}_{qs}$ remain bounded. Since all of the dynamic equations in question thus far are time-invariant, the uniformity in $t$ condition is guaranteed.

Equation (3.18), along with the dynamics for $\tilde{\beta}_q$ and $\tilde{\theta}_q$, form the error dynamics of the system. Now, we are free to choose the tuning laws $\dot{\hat{\beta}}_q$ and $\dot{\hat{\theta}}_q$ in such a way that the error dynamics become stable. With the assumption that the unknown parameters are constant or slowly varying in

time, we may take the derivative of the parameter estimation errors from Equations (3.16) and (3.17) to get

$$\dot{\tilde{\beta}}_q = \dot{\hat{\beta}}_q - \dot{\beta} = \dot{\hat{\beta}}_q \tag{3.19}$$

$$\dot{\tilde{\theta}}_q = \dot{\hat{\theta}}_q - \dot{\theta}_q = \dot{\hat{\theta}}_q \tag{3.20}$$

since the derivatives $\dot{\beta}$ and $\dot{\theta}_q$ equal zero.[6] Consider the following candidate Lyapunov function:

$$V(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \frac{1}{2}e_q^2 + \frac{1}{2\gamma}(\tilde{\beta}_q^2 + \tilde{\theta}_q^2) \tag{3.21}$$

Here, $\gamma > 0$ is an arbitrary positive constant. Taking the derivative of (3.21) with respect to time and substituting in (3.18)–(3.20) gives

$$\begin{aligned}
\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) &= e_q\dot{e}_q + \frac{1}{\gamma}(\tilde{\beta}_q\dot{\tilde{\beta}}_q + \tilde{\theta}_q\dot{\tilde{\theta}}_q) \\
&= \alpha_m e_q^2 + \tilde{\beta}_q i_{qs} e_q + \tilde{\theta}_q \lambda_{qr} e_q + \frac{1}{\gamma}(\tilde{\beta}_q\dot{\hat{\beta}}_q + \tilde{\theta}_q\dot{\hat{\theta}}_q) \\
&= \alpha_m e_q^2 + \tilde{\beta}_q(i_{qs}e_q + \frac{1}{\gamma}\dot{\hat{\beta}}_q) + \tilde{\theta}_q(\lambda_{qr}e_q + \frac{1}{\gamma}\dot{\hat{\theta}}_q)
\end{aligned} \tag{3.22}$$

Now, the tuning laws are chosen as follows:

$$\dot{\hat{\beta}}_q = -\gamma \text{Proj}(i_{qs}e_q) \tag{3.23}$$

$$\dot{\hat{\theta}}_q = -\gamma \text{Proj}(\lambda_{qr}e_q) \tag{3.24}$$

Here, the operator Proj$(\cdot)$ is used to indicate that projection-based adaptation is used. This operation ensures that the estimates stay within the reasonable bounds known *a priori* (see Table 3.2 for the bounds used in this case). Whenever the estimates $\hat{\beta}_q$ and $\hat{\theta}_q$ are within the specified bounds, the projection operation has absolutely no effect. However, if the estimates reach the boundary, the projection operation will change the tuning law temporarily so that the estimates do not stray outside the bound. In this case, the projection operation simply turns off adaptation temporarily whenever the estimate is at the boundary and being driven in a direction outside the

[6] In the event that these parameters are slowly time-varying, the effects of $\dot{\beta}$ and $\dot{\theta}_q$ are negligible compared to $\dot{\hat{\beta}}_q$ and $\dot{\hat{\theta}}_q$ under fast enough adaptation.

boundary. This may be accomplished using an integrator with saturation, since each unknown parameter is a scalar in our case. For more information on projection-based adaptation, see [31]. Several other types of tuning law modification techniques are discussed in [32].

The positive constant $\gamma$ is used here as the tuning rate. This provides some flexibility of the controller to adjust how fast the adaptation will take place. Increasing the tuning rate will cause the estimates to converge more quickly, hence increasing performance of the controller. However, increasing $\gamma$ comes with a heavy cost. It will be shown that increasing the tuning rate also results in a significant decrease in robustness.

Substituting the tuning laws (3.23) and (3.24) results in the following three cases:

(i) Both $\hat{\beta}_q$ and $\hat{\theta}_q$ are within the specified bounds and the tuning law will not immediately force either of them outside their respective bounds.

(ii) Either $\hat{\beta}_q$ or $\hat{\theta}_q$ (or both) is at the specified upper bound, and the tuning law (if no projection was used) would cause the value to increase.

(iii) Either $\hat{\beta}_q$ or $\hat{\theta}_q$ (or both) is at the specified lower bound, and the tuning law (if no projection was used) would cause the value to decrease.

In case (i), the projection operation will have no effect. Thus, the tuning laws in (3.23) and (3.24) may be substituted into (3.22) with the Proj$(\cdot)$ operation ignored. This substitution results in cancellation of both the term containing $\tilde{\beta}_q$ and the term containing $\tilde{\theta}_q$, leaving us with

$$\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \alpha_m e_q^2 \leq 0 \tag{3.25}$$

For case (ii), let us assume without loss of generality that $\hat{\beta}_q$ is at its upper bound and would be increasing without projection, while $\hat{\theta}_q$ is within its bounds. Thus, the projection operation can be ignored for $\hat{\theta}_q$, while the tuning laws for $\hat{\beta}_q$ become $\dot{\hat{\beta}}_q = 0$. Substituting these tuning laws into (3.22) causes the term containing $\tilde{\theta}_q$ to be cancelled as before, but not the term containing $\tilde{\beta}_q$. In this case, the derivative of our candidate Lyapunov function becomes

$$\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \alpha_m e_q^2 + \tilde{\beta}_q i_{qs} e_q$$

Since the estimate $\hat{\beta}_q$ is at the upper bound of its possible values, we know that

$$\hat{\beta}_q \geq \beta$$

Thus, from (3.16) we know that

$$\tilde{\beta}_q \geq 0$$

Since this case involves the situation where the tuning laws would cause $\hat{\beta}_q$ to increase without projection, we also know from Equation (3.23) that

$$\dot{\hat{\beta}}_q = -\gamma i_{qs} e_q \geq 0$$

and hence that

$$i_{qs} e_q \leq 0$$

since $\gamma > 0$. We can now combine the inequalities to conclude that

$$\tilde{\beta}_q i_{qs} e_q \leq 0$$

For case (ii), the resulting derivative of our candidate Lyapunov function now becomes

$$\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \alpha_m e_q^2 + \underbrace{\tilde{\beta}_q i_{qs} e_q}_{\leq 0} \leq \alpha_m e_q^2 \leq 0 \tag{3.26}$$

Case (iii) becomes very similar to case (ii) except the signs are reversed. Again, without loss of generality, assume that $\hat{\beta}_q$ is at its lower bound and would be decreasing without projection, while $\hat{\theta}_q$ is within its bounds. We now know that

$$\tilde{\beta}_q \leq 0$$

since $\hat{\beta}_q$ is at its lower bound. Also, it is known that

$$\dot{\hat{\beta}}_q = -\gamma i_{qs} e_q \leq 0$$

since the tuning law would cause the parameter estimation to decrease in the absence of projection. Thus, we have

$$i_{qs} e_q \geq 0 \Rightarrow \tilde{\beta}_q i_{qs} e_q \leq 0$$

Similarly to the previous case, we can now substitute the tuning laws given in (3.23) and (3.24) into (3.22), with the Proj$(\cdot)$ operation ignored in the $\hat{\theta}_q$ tuning law, and the tuning law for $\hat{\beta}_q$ forced to zero due to projection. Now, for case (iii), the derivative of the candidate Lyapunov function becomes

$$\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \alpha_m e_q^2 + \underbrace{\tilde{\beta}_q i_{qs} e_q}_{\leq 0} \leq \alpha_m e_q^2 \leq 0 \tag{3.27}$$

Examining the results from all three cases, it becomes clear from Equations (3.25)–(3.27) that even with the projection-based adaptation, we are always guaranteed to have a negative semidefinite derivative for the candidate Lyapunov function. Combining the results of all three cases gives the following final bound on $\dot{V}$:

$$\dot{V}(e_q, \tilde{\beta}_q, \tilde{\theta}_q) \leq \alpha_m e_q^2 \leq 0 \tag{3.28}$$

Now, recall Lyapunov's famous theorem, which may be found in [27], along with a proof. We may now use Lyapunov's theorem to show stability of the error dynamics, which are represented by the three states $e_q$, $\tilde{\beta}_q$, and $\tilde{\theta}_q$. The function $V(e_q, \tilde{\beta}_q, \tilde{\theta}_q)$ as given in (3.21) is positive definite and continuously differentiable, with a negative semidefinite derivative as seen in Equation (3.28). Therefore, the conditions in Lyapunov's theorem are satisfied, and the system is said to be stable. By definition of stability, we now have that the states $e_q$, $\tilde{\beta}_q$, and $\tilde{\theta}_q$ all remain bounded for all time.

It is important to note that, since the error system has three states, and the bound given in (3.28) only involves one of the states, $\dot{V}$ is *not* negative definite, but only semidefinite.[7] Therefore, we *cannot* say whether or not $e_q \longrightarrow 0$ asymptotically using Lyapunov's theorem at this point.

In order to show asymptotic convergence of $e_q$, we may use the La-Salle-Yoshizawa theorem, given below. Theorem 1 may be found in [27], along with its proof.

**Theorem 1** (La-Salle-Yoshizawa)**.** *Let $x = 0$ be an equilibrium point of the system*

$$\dot{x}(t) = f(t, x(t)), \qquad x(0) = x_0 \tag{3.29}$$

*Let $V(x(t))$ be a positive definite, continuously differentiable, and radially*

[7] This is because $\dot{V} = 0$ whenever $e_q = 0$, even when $\tilde{\beta}_q \neq 0$ or $\tilde{\theta}_q \neq 0$.

*unbounded function such that the derivative*

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x} \cdot f(t, x(t)) \leq -W(x(t)) \leq 0, \qquad \forall t \geq 0, \qquad \forall x \in \mathbb{R}^n \quad (3.30)$$

*where* $W(x(t))$ *is continuous. Then all solutions of the system given by* (3.29) *are uniformly bounded and satisfy*

$$\lim_{t \to \infty} W(x(t)) = 0 \quad (3.31)$$

*Additionally, if* $W(x(t))$ *is positive definite, then the equilibrium* $x = 0$ *of the system given by* (3.29) *is globally uniformly asymptotically stable.*

In this case, we have

$$\dot{V} \leq -W(e_q, \tilde{\beta}_q, \tilde{\theta}_q) = \alpha_m e_q^2 \leq 0$$

Invoking Theorem 1 yields

$$\lim_{t \to \infty} \alpha_m e_q^2 = 0 \qquad \Longrightarrow \qquad \lim_{t \to \infty} e_q = 0$$

We have now shown that the error $e_q$ will asymptotically converge to zero as $t \to \infty$ using the tuning laws (3.23) and (3.24), control law (3.15), and state predictor (3.13). However, since the state predictor is fed with a control law which exactly reduces its form to the model reference system, it is unnecessary to fully implement the state predictor. Instead, we may just use a model of the reference system itself. The final form of the MRAC controller for the quadrature axis flux loop is shown in Figure 3.3.

### 3.4.2 Direct Axis Rotor Flux Control Loop

The direct axis flux control loop is very similar to the quadrature axis loop, with the exception that the goal is now to track a desired non-zero flux value, rather than to regulate it to zero. Recall the direct axis flux dynamics in Equation (3.7) on page 16. Here, $w_{sl}$ and $\lambda_{qr}$ are treated as external known disturbance signals. Now, $i_{ds}$ is the input to the system, used to regulate $\lambda_{dr}$ to the desired value. Again, $\alpha$ and $\beta$ are unknown.

Again, a simple stable linear system is chosen as the model reference sys-

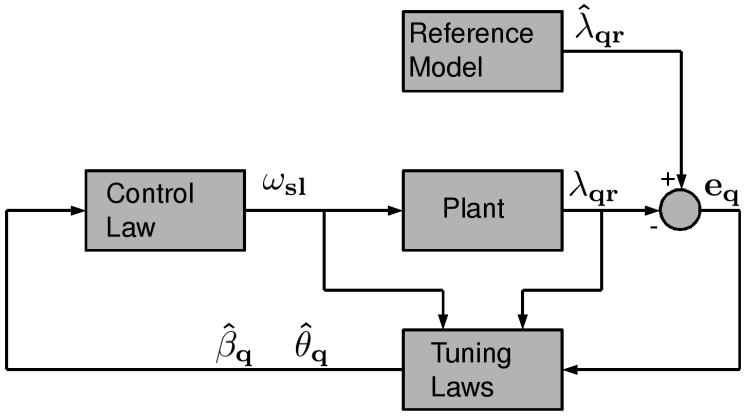


Figure 3.3: Final form of MRAC controller for q-axis flux loop

tem. We choose

$$\dot{\lambda}_{dr,m} = \alpha_m \lambda_{dr} - \alpha_m \lambda_{dr,ref} \tag{3.32}$$

We now have an input, $\lambda_{dr,ref}$, which we are trying to track with the system. Now, we may rewrite the system dynamics from Equation (3.7) as follows:

$$\begin{aligned} \dot{\lambda}_{dr} &= \alpha_m \lambda_{dr} + \omega_{sl} \lambda_{qr} + \beta i_{ds} - (\alpha_m + \alpha) \lambda_{dr} \\ &= \alpha_m \lambda_{dr} + \omega_{sl} \lambda_{qr} + \beta i_{ds} + \theta_d \lambda_{dr} \end{aligned} \tag{3.33}$$

Here, we have labeled the new unknown parameter $\theta_d = -(\alpha_m + \alpha)$. Notice that this has exactly the same meaning as the parameter $\theta_q$ from the q-axis flux control loop. However, we still have each control loop run separate estimates. This is because the tuning laws are not meant to accurately estimate the true values; the goal is only to converge to values which make each respective controller achieve its goals.

Now, we choose the state predictor

$$\dot{\hat{\lambda}}_{dr} = \alpha_m \hat{\lambda}_{dr} + \omega_{sl} \lambda_{qr} + \hat{\beta}_d i_{ds} + \hat{\theta}_d \lambda_{dr} \tag{3.34}$$

where $\hat{\beta}_d$ and $\hat{\theta}_d$ are estimates of $\beta$ and $\theta_d$, respectively. As before, a control law is chosen to make the state predictor behave exactly like the desired model reference system. We choose

$$i_{ds} = \frac{1}{\hat{\beta}_d}(-\alpha_m \lambda_{dr,ref} - \omega_{sl} \lambda_{qr} - \hat{\theta}_d \lambda_{dr}) \tag{3.35}$$

To ensure that the control law remains bounded, it is necessary to know that $\hat{\beta}_d > 0$ strictly, for all time. Since we know *a priori* that $\beta$ is strictly greater than zero, it is possible to enforce a lower bound on the estimate $\hat{\beta}_d$ in order to achieve this goal. Projection based adaptation will be used to ensure that $\hat{\beta}_d$ is never zero. Now, it is clear that the state predictor in (3.34) will behave exactly like the desired reference model (3.32), when fed with the input $i_{ds}$ from the control law in (3.35). The parameter estimates $\hat{\beta}_d$ and $\hat{\theta}_d$ must be chosen in such a way that the state predictor will also follow the behavior of the actual plant.

With the d-axis flux error and parameter estimation errors defined as

$$e_d = \hat{\lambda}_{dr} - \lambda_{dr} \tag{3.36}$$

$$\tilde{\beta}_d = \hat{\beta}_d - \beta \tag{3.37}$$

$$\tilde{\theta}_d = \hat{\theta}_d - \theta_d \tag{3.38}$$

the error dynamics of this system can be found by taking the derivative of $e_d$ and substituting (3.34) and (3.33) as follows:

$$\begin{aligned}
\dot{e}_d &= \dot{\hat{\lambda}}_{dr} - \dot{\lambda}_{dr} \\
&= \alpha_m \hat{\lambda}_{dr} + \omega_{sl}\lambda_{qr} + \hat{\beta}_d i_{ds} + \hat{\theta}_d \lambda_{dr} \\
&\qquad - \alpha_m \lambda_{dr} - \omega_{sl}\lambda_{qr} - \beta i_{ds} - \theta_d \lambda_{dr} \\
&= \alpha_m e_d + \tilde{\beta}_d i_{ds} + \tilde{\theta}_d \lambda_{dr}
\end{aligned} \tag{3.39}$$

Since all terms of (3.39) are assumed to have bounded derivative, the error dynamics are locally Lipschitz, as defined in Definition 1. Notice that we once again have the relation between the derivative of the parameter estimation error and the tuning laws as follows:

$$\dot{\tilde{\beta}}_d = \dot{\hat{\beta}}_d - \dot{\beta} = \dot{\hat{\beta}}_d \tag{3.40}$$

$$\dot{\tilde{\theta}}_d = \dot{\hat{\theta}}_d - \dot{\theta}_d = \dot{\hat{\theta}}_d \tag{3.41}$$

since $\beta$ and $\theta_d$ are assumed to be constant or very slowly time varying.

Now, consider the candidate Lyapunov function

$$V(e_d, \tilde{\beta}_d, \tilde{\theta}_d) = \frac{1}{2}e_d^2 + \frac{1}{2\gamma}(\tilde{\beta}_d^2 + \tilde{\theta}_d^2) \tag{3.42}$$

where $\gamma > 0$ is an arbitrary constant that may later be used to adjust the tuning rate. Taking the derivative of this function and substituting Equations (3.39)-(3.41) yields

$$\begin{aligned}\dot{V}(e_d, \tilde{\beta}_d, \tilde{\theta}_d) &= e_d \dot{e}_d + \frac{1}{\gamma}(\tilde{\beta}_d \dot{\hat{\beta}}_d + \tilde{\theta}_d \dot{\hat{\theta}}_d) \\ &= \alpha_m e_d^2 + \tilde{\beta}_d i_{ds} e_d + \tilde{\theta}_d \lambda_{dr} e_d + \frac{1}{\gamma}(\tilde{\beta}_d \dot{\hat{\beta}}_d + \tilde{\theta}_d \dot{\hat{\theta}}_d) \\ &= \alpha_m e_d^2 + \tilde{\beta}_d (i_{ds} e_d + \frac{1}{\gamma}\dot{\hat{\beta}}_d) + \tilde{\theta}_d (\lambda_{dr} e_d + \frac{1}{\gamma}\dot{\hat{\theta}}_d) \end{aligned} \tag{3.43}$$

The tuning laws $\dot{\hat{\beta}}_d$ and $\dot{\hat{\theta}}_d$ are chosen as follows:

$$\dot{\hat{\beta}}_d = -\gamma \text{Proj}\left(i_{ds} e_d\right) \tag{3.44}$$

$$\dot{\hat{\theta}}_d = -\gamma \text{Proj}\left(\lambda_{dr} e_d\right) \tag{3.45}$$

The Proj$(\cdot)$ operation is used again to ensure that the parameters remain within *a priori* known bounds, as described on page 24. The projection operation has no effect as long as the parameter estimates remain inside the bounds, resulting in cancellation of the unknown terms in Equation (3.43). As before, the three cases on page 25 may be applied to show that

$$\tilde{\beta}_d (i_{ds} e_d + \frac{1}{\gamma}\dot{\hat{\beta}}_d) \leq 0$$

and

$$\tilde{\theta}_d (\lambda_{dr} e_d + \frac{1}{\gamma}\dot{\hat{\theta}}_d) \leq 0$$

even when the projection operation is affecting the tuning law. The time derivative of our candidate Lyapunov function then has the upper bound as follows:

$$\dot{V}(e_d, \tilde{\beta}_d, \tilde{\theta}_d) \leq \alpha_m e_d^2 \leq 0 \tag{3.46}$$

We may now apply Theorem 1 directly to conclude that

$$\lim_{t\to\infty} \alpha_m e_d^2 = 0 \qquad \Longrightarrow \qquad \lim_{t\to\infty} e_d = 0$$

It has now been shown that the error between the state predictor output and actual system output will converge to zero. With the control input from Equation (3.35), the state predictor (3.34) assumes the exact same dynamics

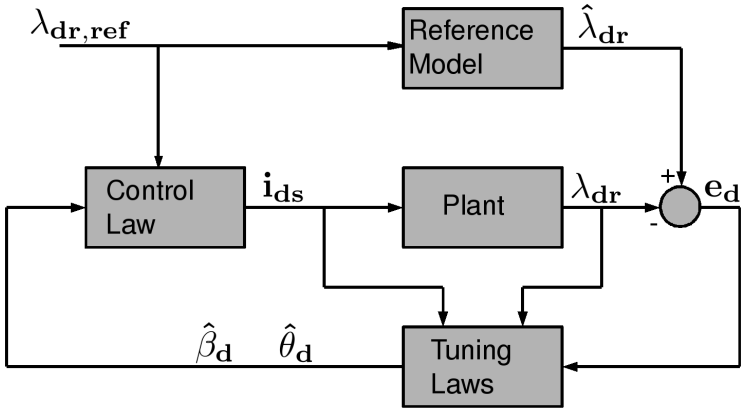


Figure 3.4: Final form of MRAC controller for d-axis flux loop

as the model reference system in (3.32). Thus, the actual system dynamics will track the desired model behavior, when also fed the control input from Equation (3.35).

As before, since the state predictor dynamics (3.34) become exactly the same as the desired reference model when the control input (3.35) is used, it is unnecessary to actually run the state predictor. Instead, the desired reference model may be used in its place, with the input $\lambda_{dr,ref}$. The final form of the d-axis flux control loop may be seen in Figure 3.4.

### 3.4.3 Speed Control Loop

The speed control loop is slightly different from the flux control loops, but the approach is the same. Recall the speed dynamics given in Equation (3.6) on page 16. This system has three unknown parameters: $a$, $\mu$, and $\sigma$. The unknown parameter $\mu$ is known to be strictly greater than zero. Also, the direct axis flux $\lambda_{dr}$ will be regulated to a non-zero value. Thus, $i_{qs}$ may be used as the input to the system. The quadrature flux $\lambda_{qr}$ will be regulated to zero, so it will not affect the system at steady state.

While $i_{qs}$ is the actual input to this system, the adaptive portion of this controller will define and consider

$$u = \lambda_{dr} i_{qs} - \lambda_{qr} i_{ds} \tag{3.47}$$

as the input to the system. Since all three signals $\lambda_{dr}$, $\lambda_{qr}$, and $i_{ds}$ are known,

it is a simple matter to calculate what $i_{qs}$ should be given the $u$ generated by the adaptive controller.

A stable linear system

$$\dot{\omega}_{mod} = a_m \omega_{mod} - a_m \omega_{ref} \tag{3.48}$$

with convergence rate $a_m$ is chosen as the reference model which we want to track. The speed command $\omega_{ref}$ is the input to this system.

Now, the system dynamics (3.6) may be rewritten as follows:

$$\begin{aligned} \dot{\omega}_m &= a_m \omega_m + \mu u + \sigma - (a + a_m)\omega_m \\ &= a_m \omega_m + \mu u + \sigma + \theta \omega_m \end{aligned} \tag{3.49}$$

Here, the new parameter $\theta = -(a+a_m)$ is introduced. Next, a state predictor is defined, again using the exact same form as these rewritten dynamics. This state predictor becomes the following:

$$\dot{\hat{\omega}}_m = a_m \hat{\omega}_m + \hat{\mu} u + \hat{\sigma} + \hat{\theta} \omega_m \tag{3.50}$$

The hat symbol over the unknown parameters again indicates that an estimate of the parameter is used, since its true value is unknown.

The control law

$$u = -\frac{1}{\hat{\mu}}(\hat{\sigma} + \hat{\theta}\omega_m + a_m \omega_{ref}) \tag{3.51}$$

will cause the dynamics of the state predictor to behave exactly the same as the dynamics of the desired reference model given in (3.48). In order for this control law to remain bounded, it is important that $\hat{\mu}$ never approaches zero. However, as Table 3.2 shows on page 15, we know *a priori* that the actual parameter $\mu$ is always greater than zero. Thus, we will again use projection-based adaptation to ensure that $\hat{\mu}$ never falls below this minimum value, and never approaches zero.

Now, the state predictor error and estimation errors are defined as

$$\begin{aligned} e_\omega &= \hat{\omega}_m - \omega_m, \\ \tilde{\theta} &= \hat{\theta} - \theta, \\ \tilde{\sigma} &= \hat{\sigma} - \sigma, \\ \text{and} \qquad \tilde{\mu} &= \hat{\mu} - \mu. \end{aligned}$$

The speed error dynamics are found by taking the derivative of $e_\omega$, and substituting the state predictor and actual dynamics from Equations (3.50) and (3.49), as follows:

$$\begin{aligned} \dot{e}_\omega &= \dot{\hat{\omega}}_m - \dot{\omega}_m \\ &= a_m\hat{\omega}_m + \hat{\mu}u + \hat{\sigma} + \hat{\theta}\omega_m \\ &\qquad - a_m\omega_m - \mu u - \sigma - \theta\omega_m \\ &= a_m e_\omega + \tilde{\mu}u + \tilde{\sigma} + \tilde{\theta}\omega_m \end{aligned} \tag{3.52}$$

The objective is now to tune the adaptive parameters $\hat{\mu}$, $\hat{\sigma}$, and $\hat{\theta}$ in such a way that their errors remain bounded, and the error $e_\omega$ converges to zero. Again, it is noted that as long as the parameter errors remain bounded, the speed error dynamics of Equation (3.52) remain bounded, and the function is Lipschitz by Definition 1. With fast adaptation and very slow changing unknown parameters, we have the following relationships:

$$\begin{aligned} \dot{\tilde{\mu}} &= \dot{\hat{\mu}} - \dot{\mu} = \dot{\hat{\mu}} \\ \dot{\tilde{\sigma}} &= \dot{\hat{\sigma}} - \dot{\sigma} = \dot{\hat{\sigma}} \\ \dot{\tilde{\theta}} &= \dot{\hat{\theta}} - \dot{\theta} = \dot{\hat{\theta}} \end{aligned}$$

Now, we may examine stability of the speed error dynamics by considering the following candidate Lyapunov function:

$$V(e_\omega, \tilde{\mu}, \tilde{\sigma}, \tilde{\theta}) = \frac{1}{2}e_\omega^2 + \frac{1}{2\gamma}(\tilde{\mu}^2 + \tilde{\sigma}^2 + \tilde{\theta}^2) \tag{3.53}$$

Again, $\gamma > 0$ is a constant, which can be adjusted to change the rate of adaptation. Taking the derivative of this function and substituting Equa-

tion (3.52) yields

$$\begin{aligned}\dot{V}(e_\omega, \tilde{\mu}, \tilde{\sigma}, \tilde{\theta}) &= e_\omega \dot{e}_\omega + \frac{1}{\gamma}(\tilde{\mu}\dot{\tilde{\mu}} + \tilde{\sigma}\dot{\tilde{\sigma}} + \tilde{\theta}\dot{\tilde{\theta}}) \\ &= a_m e_\omega^2 + \tilde{\mu} e_\omega u + \tilde{\sigma} e_\omega + \tilde{\theta} e_\omega \omega_m + \frac{1}{\gamma}(\tilde{\mu}\dot{\tilde{\mu}} + \tilde{\sigma}\dot{\tilde{\sigma}} + \tilde{\theta}\dot{\tilde{\theta}}) \\ &= a_m e_\omega^2 + \tilde{\mu}(e_\omega u + \frac{1}{\gamma}\dot{\tilde{\mu}}) + \tilde{\sigma}(e_\omega + \frac{1}{\gamma}\dot{\tilde{\sigma}}) + \tilde{\theta}(e_\omega \omega_m + \frac{1}{\gamma}\dot{\tilde{\theta}}) \quad (3.54)\end{aligned}$$

All the terms with parameter estimation errors in (3.54) may be cancelled (or forced to be less than zero) using the tuning laws

$$\dot{\tilde{\mu}} = -\gamma \text{Proj}(e_\omega u) \quad (3.55)$$

$$\dot{\tilde{\sigma}} = -\gamma \text{Proj}(e_\omega) \quad (3.56)$$

$$\dot{\tilde{\theta}} = -\gamma \text{Proj}(e_\omega \omega_m) \quad (3.57)$$

Once again, the Proj$(\cdot)$ operation is used to ensure that the parameters remain within *a priori* known bounds, as described on page 24. While the unknown parameters are within their bounds, the Proj$(\cdot)$ operation has no effect, resulting in cancellation of the unknown terms in Equation (3.54). As before, the three cases on page 25 may be applied to show that

$$\begin{aligned}\tilde{\mu}(e_\omega u + \frac{1}{\gamma}\dot{\tilde{\mu}}) &\leq 0, \\ \tilde{\sigma}(e_\omega + \frac{1}{\gamma}\dot{\tilde{\sigma}}) &\leq 0, \\ \text{and} \qquad \tilde{\theta}(e_\omega \omega_m + \frac{1}{\gamma}\dot{\tilde{\theta}}) &\leq 0\end{aligned}$$

even when the projection operation is affecting the tuning law. The time derivative of our candidate Lyapunov function then has the upper bound as follows:

$$\dot{V}(e_\omega, \tilde{\mu}, \tilde{\sigma}, \tilde{\theta}) \leq a_m e_\omega^2 \leq 0 \quad (3.58)$$

Once again, we may directly apply Theorem 1 to show that

$$\lim_{t\to\infty} a_m e_\omega^2 = 0 \qquad \Longrightarrow \qquad \lim_{t\to\infty} e_\omega = 0$$

Additionally, we know that since the derivative of the candidate Lyapunov function is non-positive, all error states remain bounded. Thus, the param-

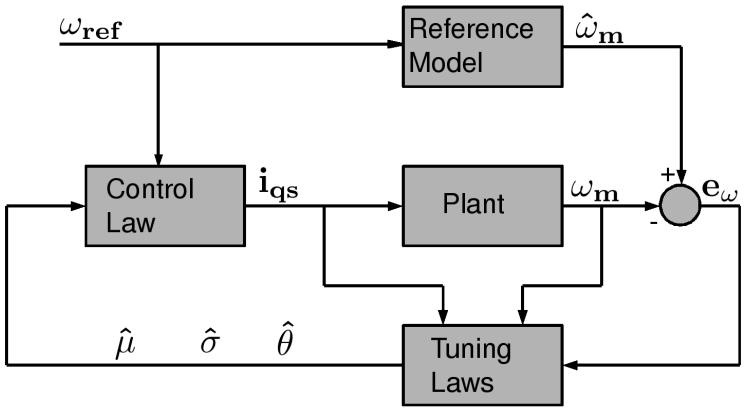


Figure 3.5: Final form of MRAC controller for speed control loop

eter estimation errors $\mu$, $\sigma$, and $\theta$ are bounded. Most importantly, since we know that $e_\omega \to 0$ as $t \to \infty$, the state predictor (3.50) will converge to the same behavior as the actual system of Equation (3.49). Since the state predictor is forced to behave exactly as the desired reference model, we know that the actual plant will converge to the desired reference behavior given in (3.48).

When fed with the control law (3.51), we know that the state predictor will assume the exact same dynamics as the desired reference model (3.48). Thus, instead of running a real-time state predictor, it is possible to simply run the reference model dynamics, with the speed command $\omega_{ref}$ directly as the input, and use the resulting output as the state predictor output $\hat{\omega}_m$. The final form of the model reference adaptive speed control loop is shown in Figure 3.5.

The overall MRAC system comprises of all three control loops acting simultaneously. This thesis has now developed the new control system based on the common design procedures as outlined in [27], for each control loop. The overall structure of this control system is very similar to the structure proposed in [25]. However, the MRAC system in this thesis is more basic. Goléa proposes the use of both proportional and integral terms when representing the error between the current and optimal control gains. There, the controller gains are tuned directly, rather than adjusting the parameter estimates which govern the control law.

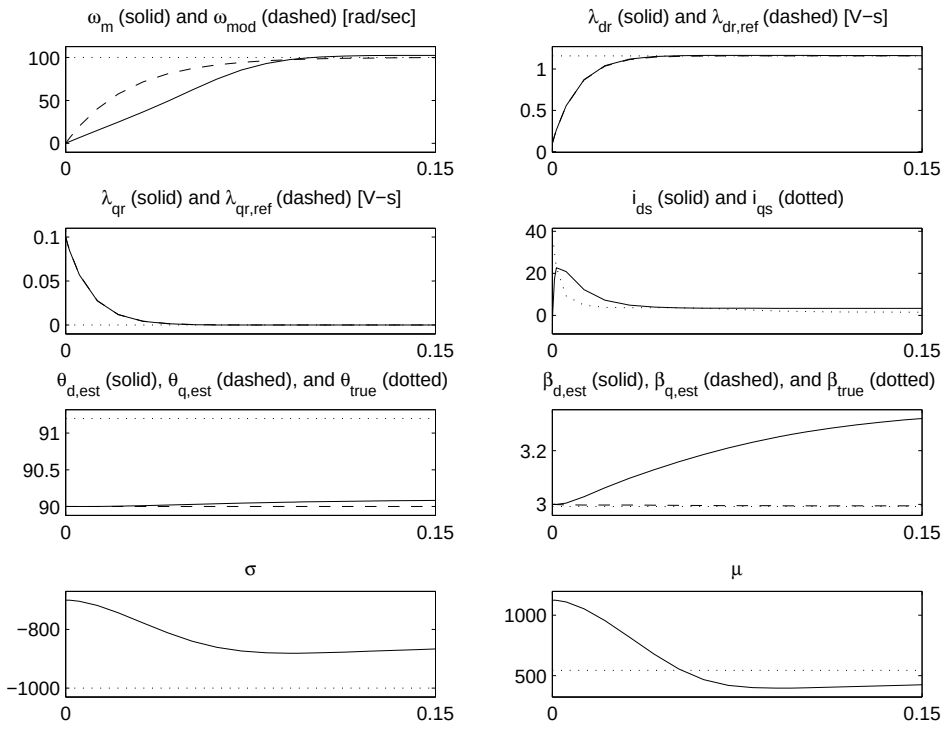


Figure 3.6: MRAC performance with $\gamma = 100$ and no parameter change, $x$ axis in seconds

## 3.5 Simulation of MRAC Controller

In this section, the model reference adaptive control system is evaluated using simulations. Table 3.2 shows the unknown parameter values used, and the *a priori* known bounds used for each parameter. In the case of the speed reference model, the convergence rate $a_m = -40$ is used, while a faster desired convergence of $\alpha_m = -100$ is used for the $q$ and $d$ axis flux control loops.

These simulation results are very similar to those in [25], as Goléa also uses model reference adaptive control in three separate control loops there. In [25], the tuning rates $\gamma$ are set to particular values for each tuning law. However, the same $\gamma$ value is used for each tuning law in this thesis. They have used the appropriate tuning rate to get the best performance and robustness from each tuning law. However, it may be argued that the necessity to adjust a controller by finding the best tuning rate for each adaptive law somewhat reduces the attractiveness of an adaptive controller. Ideally, one would want to have the fastest tuning rates possible, to ensure fast convergence. As this thesis will discuss later, increasing the tuning rate too much would result in

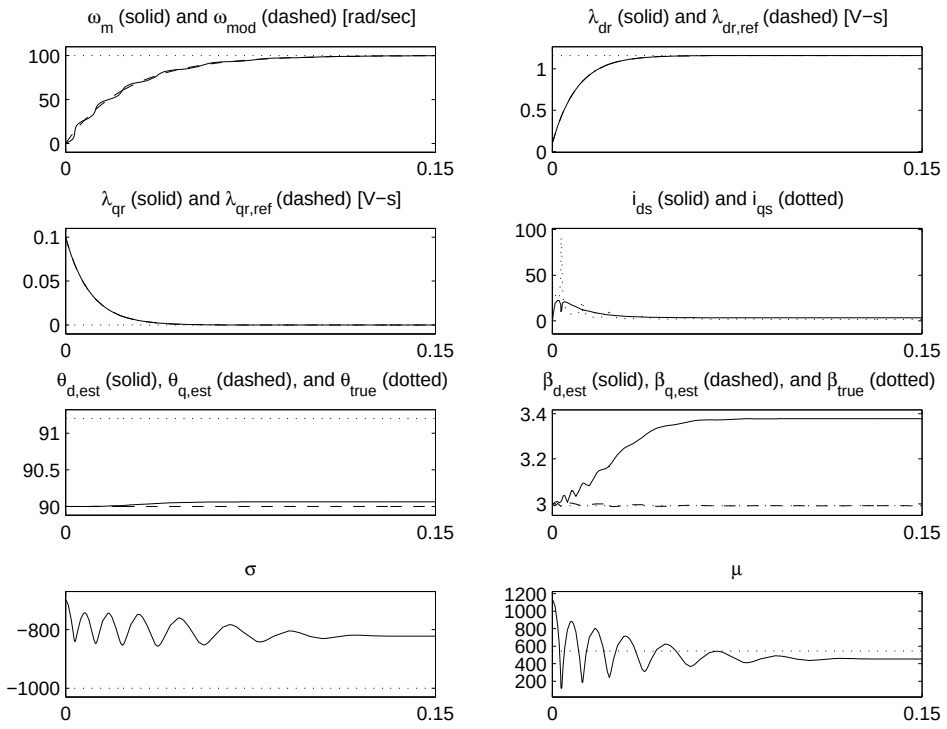


Figure 3.7: MRAC performance with $\gamma = 10,000$ and no parameter change, $x$ axis in seconds

a loss of robustness of the system for a model reference adaptive controller.

Figure 3.6 shows the simulated performance of the MRAC controller when all unknown parameters in the machine are kept at their nominal values seen in Table 3.2, with a relatively slow adaptation of $\gamma = 100$. Note that while the machine parameters are at their nominal values, the adaptive controller is still initialized with the incorrect guesses shown in the right-most column of Table 3.2. This tuning rate is fast enough to cause the flux control loops to very closely track the desired model reference behavior. However, the speed control loop takes a while before it behaves similarly to the desired reference model.

Now, if we increase the rate of adaptation to $\gamma = 10,000$, the tuned parameter estimates will have much faster convergence rates, as seen in Figure 3.7. Now, even the speed loop tracks the desired reference model fairly well. Again, the true values of each parameter remain unchanged at the nominal values given in Table 3.2.

There is a trade-off between performance and robustness when changing

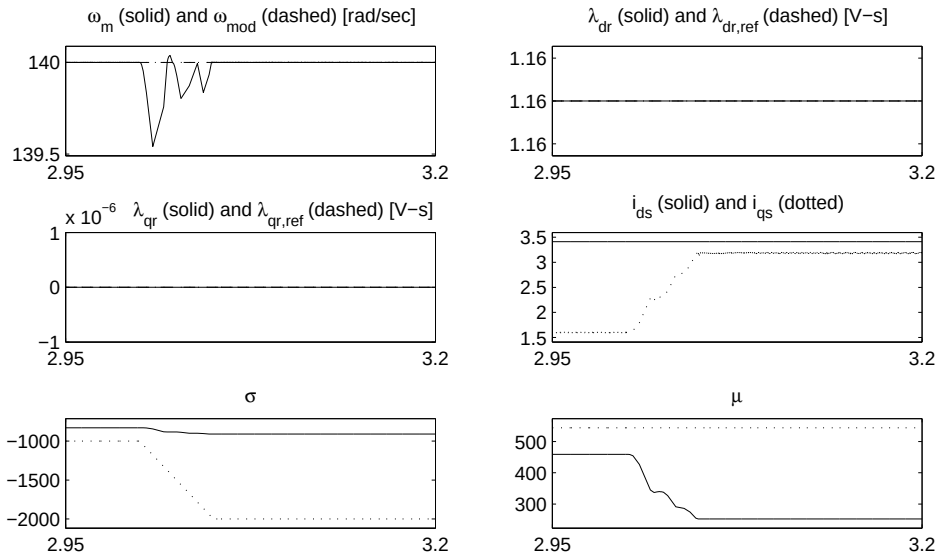


Figure 3.8: MRAC performance with torque change from 5 to 10 Nm, $x$ axis in seconds

the tuning rate with evidence in Figures 3.6 and 3.7. Higher convergence rates of the parameter estimates mean that the actual plant dynamics will perform closer to the desired model reference behavior in less time. However, this will also cause high frequency behavior in the input to the plant, which may lead to a loss of stability. Notice that in Figure 3.7, the control inputs $i_ds$ and $i_qs$ have much higher spikes.

Now, we will examine the response of this controller when the unknown parameters change. For the rest of these simulations, fast adaptation ($\gamma = 10,000$) is used. First, the controller is tested when the load torque changes from 5 to 10 Nm. From the results in Figure 3.8, it is apparent that the speed response only varies by less than 0.3% during this step change in torque. Also, the flux loops are almost completely unchanged. Still, the parameter estimation values do not converge to the true parameter values. However, as mentioned before, it is unnecessary for the parameters to converge to their true values. We only require that they remain bounded.

Next, the drive response will be tested for different inertias. Figure 3.9 shows the speed response and required quadrature axis current for different values of inertia, with slower adaptation ($\gamma = 100$). More control effort is required when the inertia is greater, but the speed tracking remains relatively consistent. This is apparent in Figure 3.9 by the fact that the quadrature

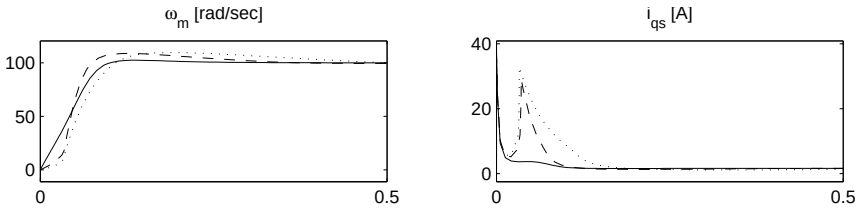


Figure 3.9: MRAC performance varying inertia: solid line, $J$; dashed line, $4J$; dotted line, $8J$; $x$ axis in seconds

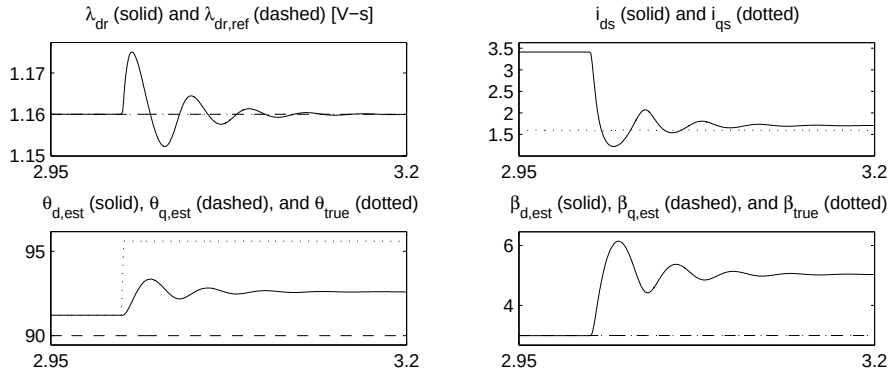


Figure 3.10: MRAC performance with $R_r$ abruptly changed by $-50\%$, $x$ axis in seconds

axis current $i_{qs}$ has much higher peak values when there is more inertia. When the inertia is eight times larger than nominal, there is some overshoot which causes a brief surge of current at over 30 A. The very fast adaptation ($\gamma = 10,000$) case is not shown because with very fast tuning, all three inertia values resulted in almost identical speed tracking.

Perhaps the most important test of the adaptive controller is under variation of electrical parameters, namely the rotor resistance, rotor inductance, and mutual inductance. Figures 3.10 and 3.11 show the direct axis flux response, control inputs, and flux parameter estimates when $R_r$ is abruptly changed by $-50\%$ and $+100\%$, respectively. Simulations of decreasing and increasing rotor inductance $L_r$ may be seen in Figures 3.12 and 3.13, respectively. The controller's performance with abrupt changes in mutual inductance $L_m$ is given in Figures 3.14 and 3.15. All of these simulations are done with fast adaptation ($\gamma = 10,000$).

Figures 3.10–3.15 give evidence that changes in the electrical characteristics of the machine have a significant impact on the system's performance.

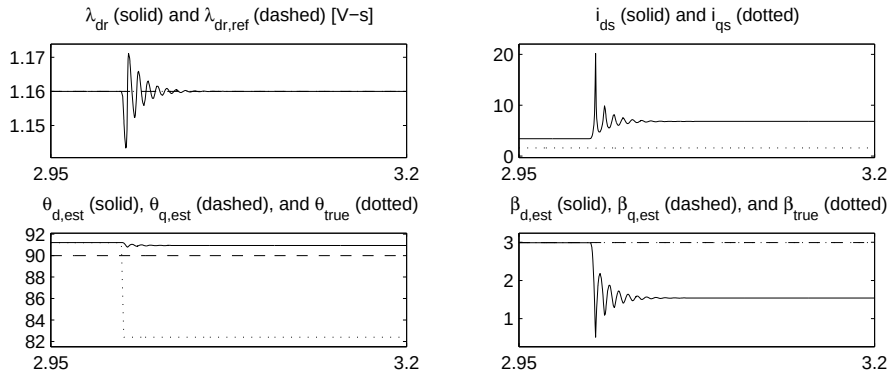


Figure 3.11: MRAC performance with $R_r$ abruptly changed by $+100\%$, $x$ axis in seconds

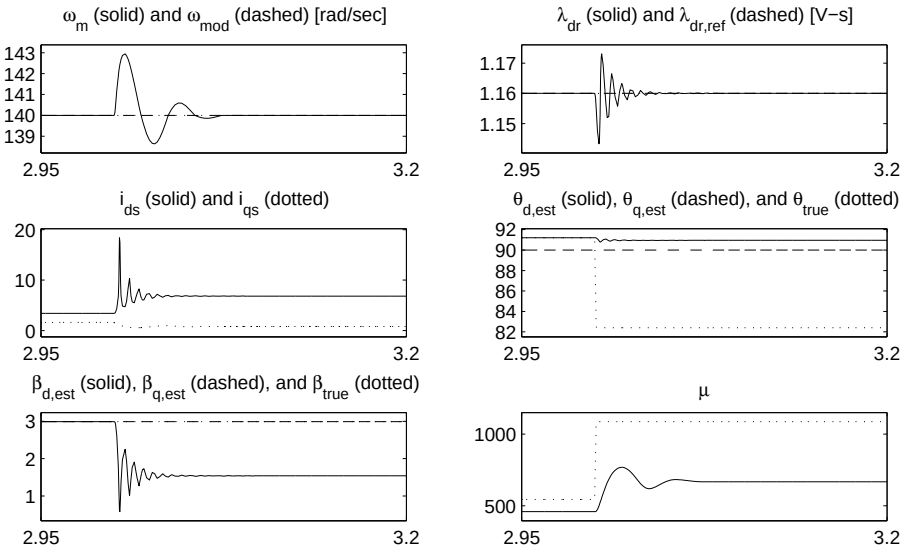


Figure 3.12: MRAC performance with $L_r$ abruptly changed by $-50\%$, $x$ axis in seconds

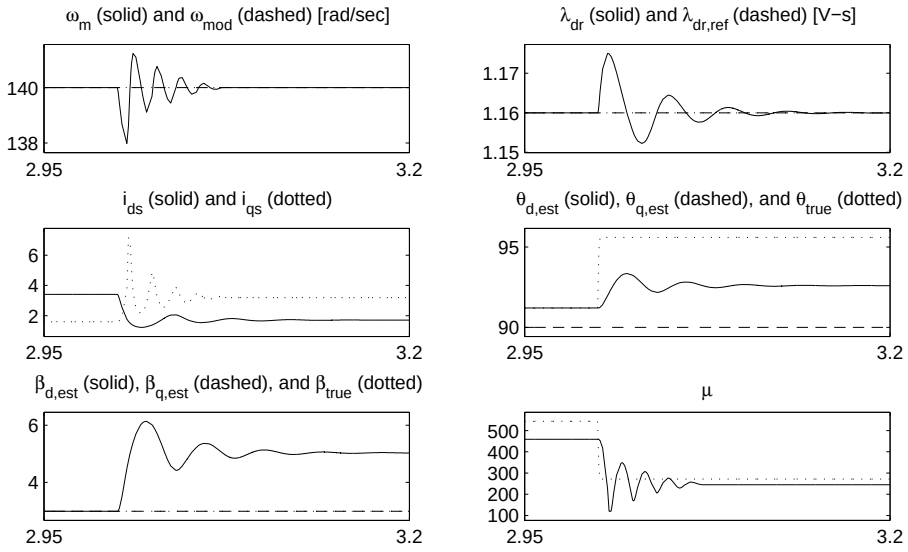


Figure 3.13: MRAC performance with $L_r$ abruptly changed by $+100\%$, $x$ axis in seconds

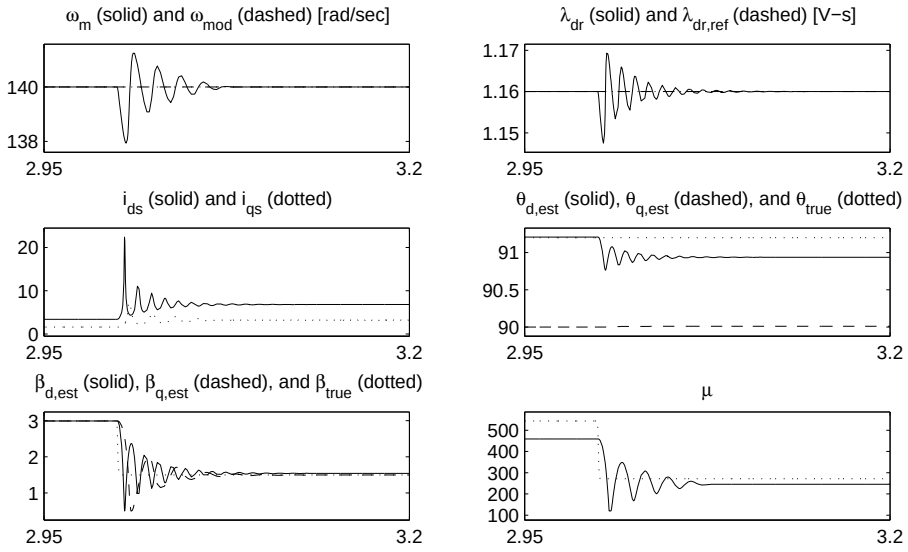


Figure 3.14: MRAC performance with $L_m$ abruptly changed by $-50\%$, $x$ axis in seconds

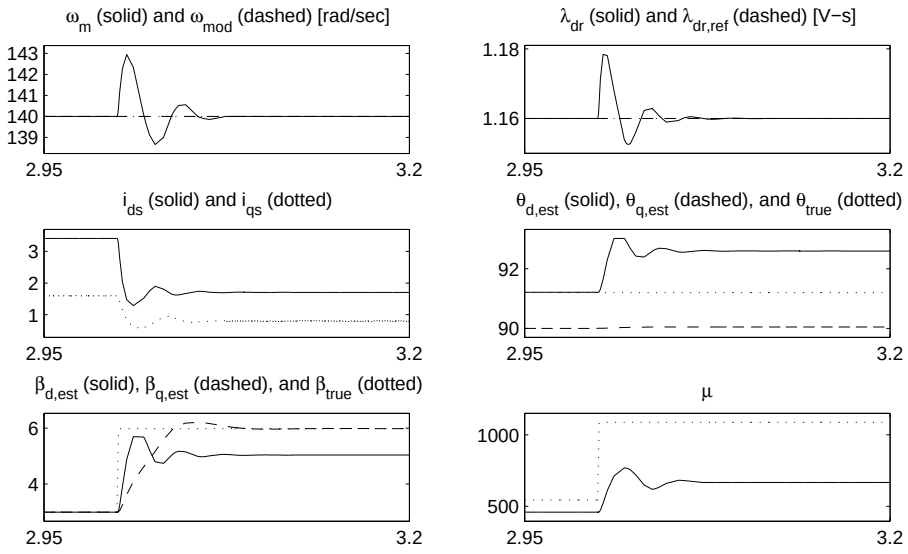


Figure 3.15: MRAC performance with $L_m$ abruptly changed by +100%, $x$ axis in seconds

Table 3.3: MRAC Error Due to Change in Electrical Parameters

| Parameter | Variation | Speed error [%] | $d$-axis flux error [%] |
|---|---|---|---|
| $R_r$ | −50% | negligible | 1.29 |
| | +100% | negligible | 1.47 |
| $L_r$ | −50% | 2.14 | 1.55 |
| | +100% | 1.43 | 1.34 |
| $L_m$ | −50% | 1.57 | 1.12 |
| | +100% | 2.21 | 1.64 |

In these figures, the electrical parameters are abruptly changed, and the system outputs are given. The errors caused by these instantaneous changes in electrical parameters are summarized in Table 3.3. However, the response due to these changes is much better than most non-adaptive controllers. Increasing the tuning rate $\gamma$ will result in even faster convergence of the error dynamics, so that the results shown could be improved even more, at the cost of robustness.

Thus far, all the simulations done have assumed that the machine is modeled correctly, with uncertainty of parameters. Now, it is important to show the characteristics of the system when any unmodeled dynamics are introduced. The next simulation conducted involves adding a single stable pole

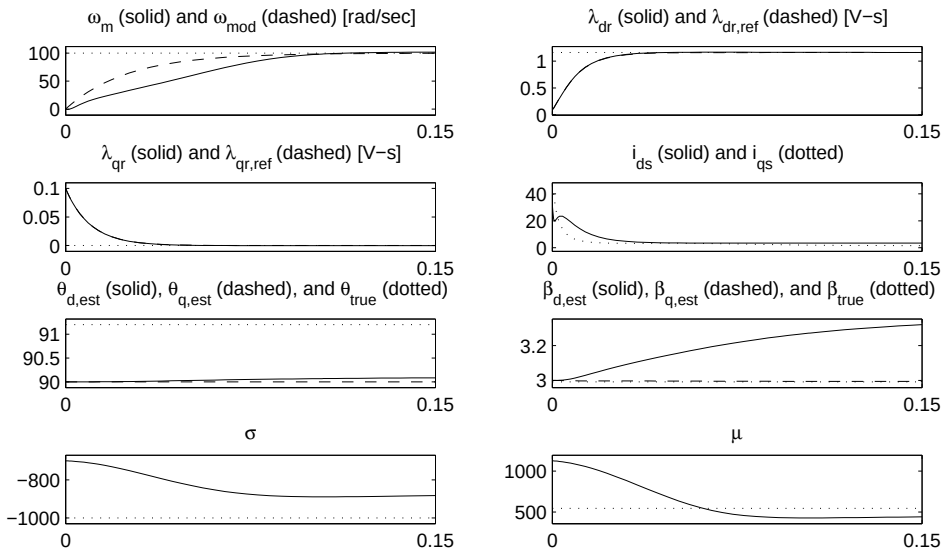


Figure 3.16: MRAC performance unmodeled fast pole and slow tuning, $x$ axis in seconds

between the controller and the machine. This pole is at $-800$ in the complex plane, so it is also much faster than the modeled dynamics of the machine (hence, its effect should disappear very quickly).

Figure 3.16 shows the response of the MRAC controller with this fast unmodeled pole present, with slow tuning of $\gamma = 100$. In this case, the unmodeled pole does not seem to have affected the performance of the controller compared to the response in Figure 3.6, where all the dynamics are modeled.

Now, the effects of turning up the tuning rate in an attempt to get better performance will be examined. In this case, the response with an unmodeled pole is shown in Figure 3.17. Now, it is apparent that this response is much different than the response without the unmodeled pole from Figure 3.7. Comparing the two figures, it is evident that with the unmodeled pole, the system is closer to instability. This can be seen by the oscillations in the adaptive parameters $\sigma$ and $\mu$, which cause oscillations in the current $i_{qs}$, which in turn results in undesirable ripple in the speed response.

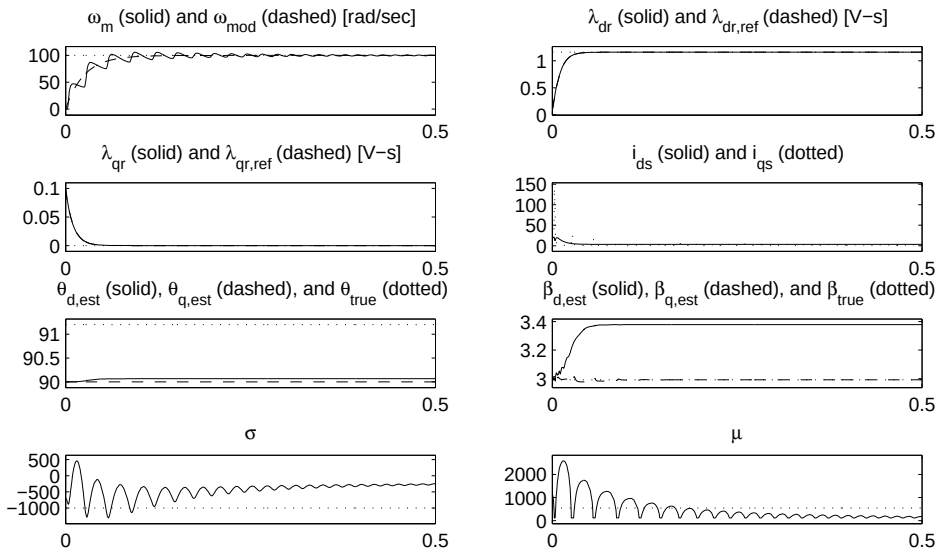


Figure 3.17: MRAC performance unmodeled fast pole and fast tuning, $x$ axis in seconds

## 3.6 Conclusions about MRAC Controller

From the above figures, the model reference adaptive controller generally works well when the system is modeled well.[8] It has been well established in the literature that the MRAC scheme may be used to control an induction motor. However, this chapter has shown that the control system is only reliable when there are no unmodeled dynamics in the system. This assumption is unrealistic in real world applications, as it is well known that high order harmonics and fast dynamics are present in most systems. The convergence rate of this controller may be increased with faster tuning rates, but with the heavy cost of less robustness.

As long as the unknown parameters are modeled and matched with estimations in the control design, the system is very robust to changes in the unknown parameter values. Table 3.3 summarizes how such changes cause instantaneous error in the speed and flux responses, under fast adaptation. This performance is much better than a conventional non-adaptive controller. Also, note that in all simulations, the controller was initialized with incorrect guesses of the unknown parameters. This shows that the system does in fact work well with parameter uncertainty.

[8]It is the form and order of the model which is important in this sense, rather than the accuracy of any parameter values.

In cases where there are unmodeled dynamics, such as the fast stable pole in the simulation case, the controller is no longer guaranteed to maintain stability. Recall page 20 when the matching assumption is discussed as necessary in order for stability to be guaranteed with the model reference adaptive controller. For the simulation in Figures 3.16 and 3.17, the extra pole at $-800$ in the complex plane affects $i_{qs}$ and $i_{ds}$ between the controller and motor. In this case, the dynamics in Equations (3.6)–(3.7) would have two extra states, as follows:

$$\dot{i}_{ds} = -800 i_{ds} + 800 i_{ds,cmd} \tag{3.59}$$

$$\dot{i}_{qs} = -800 i_{qs} + 800 i_{qs,cmd} \tag{3.60}$$

Here, $i_{ds,cmd}$ and $i_{qs,cmd}$ are the desired system inputs as manipulated by the controller. The addition of these two extra states violates the matching assumptions. On page 20, the system dynamics were re-written in a form which could be made to exactly match the desired reference model if the unknown parameter terms were cancelled out. However, if the system were modeled with the extra pole, there is no way it could be matched to the desired reference system by cancelling the unknown parameters. It is only possible to have partial cancellation of these terms at certain frequencies when the system order is mismatched between the actual dynamics and the desired reference model. The Lyapunov analysis is no longer valid for the model reference adaptive controller.

With fast tuning, as in Figure 3.17, there will be high frequency oscillations in the parameter estimates. With the control laws given in Equations (3.15), (3.35), and (3.51), these high frequency oscillations will directly enter the control input to the system. This can be seen in Figure 3.17 by examining the control commands for $i_{qs}$ and $i_{ds}$. In a real-world situation, these high frequency spikes could cause destabilization of the system.

This behavior was examined by Rohrs et al. in [33], spurring much discussion and debate about what causes this lack of robustness. In his paper, a famous counterexample is given showing loss of stability in the presence of certain unmodeled dynamics. Further analysis on these factors which destabilize adaptive controllers may be found in many papers. An example of a parameter estimate which may destabilize an adaptive control system is given by Townley in [34]. In [35], Anderson further discusses the problems

shown in the Rohrs counterexample. He contends that the reasoning for loss of stability given in [33] is illogical, and describes the issues in terms of other well-known adaptive control problems. One such issue is referred to as "bursting," in which a closed loop adaptive system will unexpectedly begin oscillating at high magnitude, and then die away again. Anderson and Dehghani further discuss the causes of bursting and other instability issues in [36].

The Rohrs counterexample is also further examined by Åström in [37]. There, some of the actual instability mechanisms for adaptive control (large command signals, high frequency command signals, and measurement noise) are studied explicitly in order to show how to avoid the problems that can arise from them. Åström further analyzes the effects of unmodeled dynamics on an adaptive control system in [38]. In order to address this issue of instability in the presence of unmodeled dynamics, Ioannou and Kokotovic propose a modified adaptive law in [39].

# CHAPTER 4

# ROBUST $\mathcal{L}_1$ ADAPTIVE DIRECT FIELD-ORIENTED CONTROL

This chapter develops a more robust adaptive controller, known as an $\mathcal{L}_1$ adaptive controller, for the purpose of an induction machine drive. The $\mathcal{L}_1$ controller has been used recently in the literature for many aerospace applications. For instance, a NASA flight test vehicle was controlled using an $\mathcal{L}_1$ adaptive scheme in [40]. However, it has not been used in the context of motor drives before this thesis. The objective is to apply the robust adaptive control theory presented in [23] to a motor drive application. This will address some of the shortcomings of a model reference adaptive controller. Namely, the $\mathcal{L}_1$ controller will allow very high tuning rates for high performance tracking, without a loss of robustness.

First, a general overview of the theory used to develop the $\mathcal{L}_1$ controller will be discussed, with heuristic justifications for how the controller can achieve its goals. Later on, an $\mathcal{L}_1$ control scheme is developed for the same machine as modeled in Chapter 3. Simulations will then be performed to verify the claims about this controller. Conclusions will then be drawn, and comparisons made between the performance of this $\mathcal{L}_1$ controller and the model reference adaptive controller developed in Chapter 3.

## 4.1 Overview of $\mathcal{L}_1$ Controller

The $\mathcal{L}_1$ adaptive controller is designed to address performance and robustness issues inherent in the design of a model reference adaptive control system. Namely, the fact that the tuning rate of the MRAC controller is chosen based on a tradeoff between performance and robustness is undesirable. The reason for this loss of robustness when the tuning rate is increased is that doing so results in high frequency oscillations in the control channel. At the heart of the proposed $\mathcal{L}_1$ controller, this is addressed with a low-pass filter in

the control law, so that high frequency oscillations which may occur in the parameter estimates do not enter the control channel [26].

The addition of a low-pass filter in the control law allows the controller to enforce desired transient performance by increasing the tuning rate arbitrarily large [23]. However, with the addition of a filter, more analysis is necessary to prove stability of the closed loop system. The filter dynamics cause a higher order desired reference model that the controller is tracking, so that the standard Lyapunov analysis (as performed in sections 3.4.1–3.4.3) is not enough to guarantee stability.

With the $\mathcal{L}_1$ controller, a new closed-loop reference system is defined, which includes the filter, and allows for better characterization of the system's transient performance. With a properly designed low-pass filter, arbitrarily fast tuning is enabled, allowing for very good transient performance without loss of robustness [23]. The very high tuning rate in this situation does not result in a high-gain control law with large control efforts, as is the case with an MRAC controller.

Compared to the MRAC controller, it will be shown that the $\mathcal{L}_1$ controller has a more realistic objective, since it relaxes the assumption that the system is modeled accurately at all frequencies. The $\mathcal{L}_1$ only assumes that the model is accurate at low frequencies, and therefore that the effects of unknown parameters may only be cancelled out at lower frequencies. This means that the overall system is much more robust to unmodeled dynamics at high frequencies, provided that these unmodeled dynamics are stable. It is very common in most real-world systems to have fast, stable dynamics which are not represented in the model used for control.

## 4.2 Development of $\mathcal{L}_1$ Controller

The $\mathcal{L}_1$ controller developed in this section is used as a direct field-oriented controller for an induction machine, with the same strategy as the MRAC system developed in section 3.4. Recall (3.6)–(3.7), which represent the dynamics of an induction machine. For each of these three dynamic equations, a separate control loop is used to get the desired behavior. Here, the parameters $a$, $\mu$, $\sigma$, $\alpha$, and $\beta$ are unknown, time varying quantities. All three control loops in this chapter are based on the $\mathcal{L}_1$ controller presented in [41].

First, a control loop is used on the $q$-axis flux dynamics of Equation (3.8). The goal of this loop is to regulate the quadrature axis rotor flux to zero, by manipulating the slip frequency $w_{sl}$. Doing so ensures that the reference frame with which the machine variables are described is in line with the rotor flux vector. This is the fundamental approach of field-oriented control, and it reduces the decoupling between torque and flux, so that they may be regulated independently. More information about field-oriented control may be found in [3], [6], or in most textbooks on modern electric machines and drives. The $\mathcal{L}_1$ adaptive controller we develop in this section adapts to compensate for the unknown parameters $\alpha$ and $\beta$.

Second, a control loop is closed around the $d$-axis flux dynamics (3.7) to achieve the desired flux level. In various operating conditions, it is desirable to use different flux levels in the machine to get the best efficiency. This control loop will manipulate the $d$-axis current going to the machine in order to get the desired direct axis rotor flux $\lambda_{dr}$. Note that when the quadrature axis flux is regulated to zero, all of the flux magnitude is present in the direct axis, so that $\|\lambda_r\| = |\lambda_{dr}|$. Again, this $\mathcal{L}_1$ adaptive controller adapts to compensate for the unknown parameters $\alpha$ and $\beta$.

Finally, a third control loop is used to regulate the machine's speed dynamics, seen in Equation (3.6). This adaptive controller manipulates the $q$-axis stator current $i_{qs}$ in order to achieve the desired speed, despite the unknown parameters $a$, $\mu$, and $\sigma$.

### 4.2.1 Quadrature Axis Flux Control Loop

Recall the quadrature axis rotor flux dynamics as given in Equation (3.8) on page 16. The goal in this section is to develop a controller which will regulate $\lambda_{qr}$ to zero by commanding the slip frequency $\omega_{sl}$ with an adaptive feedback law. Our unknown parameters are $\alpha$ and $\beta$. For this controller, we will treat $\lambda_{dr}$ and $i_{qs}$ as *known* disturbance signals, since they can be measured.[1]

The development of this controller is very similar to that of the model reference adaptive controller at first. The same linear desired reference model is used at first, seen in (3.9) on page 20. The dynamics of the actual system are then re-written to the form shown in (3.11) on page 20, where $\theta_q =$

[1] $\lambda_{dr}$ is generally not directly measured in practice. However, a good estimate may be obtained in real-time using a flux observer. See section 2.3 for more details.

$-(\alpha_m + \alpha)$.

Recall that for model reference adaptive control, a state predictor is defined with the same form as the actual plant, but with unknown parameters replaced with guesses. Then, there are two major goals to achieve in order to get the desired outcome. First, a control law (which depends on the parameter guesses) is developed which will cause this state predictor to exactly match the behavior of the desired reference model. Then, the second major goal of the MRAC controller is to develop tuning laws to adjust the parameter estimates in such a way that the state predictor converges to the same behavior as the actual plant, when the same input is fed to each. With these two goals accomplished, the actual plant will track the desired reference model behavior.

Again, we will use the same state predictor as defined for the $q$-axis flux loop in Equation (3.13). For the $\mathcal{L}_1$ controller, the second goal of causing the state predictor to track the actual plant dynamics is the same as in the MRAC case, and accomplished in the same way. The same adaptive laws from (3.23) and (3.24) will be used. The Lyapunov analysis performed on pages 23–28 remains exactly the same. Thus, we can conclude that the state predictor and plant dynamics will track one another as $t \to \infty$.

The paradigm shift in design of the $\mathcal{L}_1$ controller occurs at the first mentioned objective, which is to develop a control law which causes the state predictor to mimic the desired reference behavior. Recall the control law in Equation (3.15). When this equation is substituted into the state predictor equation, it causes complete cancellation of all terms involving the unknown parameter estimates, so that the state predictor then has the same dynamics as the desired reference model from Equation (3.9). With the $\mathcal{L}_1$ control strategy, we will relax the assumption that perfect cancellation can occur here. Instead, a stable low-pass filter will be used in the control law. This results in only partial cancellation of the unknown terms at low frequencies.

The control law that will be used for the $\mathcal{L}_1$ controller, represented in the frequency domain, is

$$\omega_{sl}(s) = C(s)\frac{1}{\lambda_{dr}(s)}\left(\hat{\beta}_q(s)i_{qs}(s) + \hat{\theta}_q(s)\lambda_{qr}(s)\right) \tag{4.1}$$

Here, $C(s)$ is the low pass filter that we have yet to design.

Substituting the control law (4.1) into the state predictor (3.13) results in

the new closed-loop system

$$\begin{aligned}\dot{\hat{\lambda}}_{qr} &= \alpha_m \hat{\lambda}_{qr}(s) - C(s)\left(\hat{\beta}_q(s) i_{qs}(s) + \hat{\theta}_q(s)\lambda_{qr}(s)\right) \\ &\quad + \hat{\beta}_q(s) i_{qs}(s) + \hat{\theta}_q(s)\lambda_{qr}(s) \\ &= \alpha_m \hat{\lambda}_{qr}(s) + (1 - C(s))\left(\hat{\beta}_q(s) i_{qs}(s) + \hat{\theta}_q(s)\lambda_{qr}(s)\right) \end{aligned} \tag{4.2}$$

Notice that if $C(s)$ were equal to one, all the right-hand terms except the first would cancel, resulting in the desired reference model dynamics. At lower frequencies, the low-pass filter $C(s)$ will be very close to one, so these cancellations will partially occur.

Now we have taken a much more realistic and relaxed assumption that our model is valid only at lower frequencies, and that the parameter estimates are good enough to cancel the unknown lower frequency dynamics of the system. The high frequency dynamics of the system are assumed to be unmodeled now, so we do not try to use control effort to correct for the high frequency behavior.[2] The low pass filter ensures that we do not have a high gain feedback system for high frequency dynamics, even if very fast tuning is used.

It must now be shown that the new reference model which the system will be tracking is stable. As described in [41], we now consider the new closed-loop reference system with control signal defined in

$$\dot{\lambda}_{qr,ref}(t) = \alpha_m \lambda_{qr,ref}(t) - \omega_{sl,ref}(t)\lambda_{dr}(t) + \beta i_{qs}(t) + \theta_q \lambda_{qr,ref}(t) \tag{4.3}$$

$$\lambda_{qr,ref}(0) = \lambda_{q0} \tag{4.4}$$

$$\omega_{sl,ref}(s) = C(s)\frac{1}{\lambda_{dr}(s)}\left(\beta i_{qs}(s) + \theta_q \lambda_{qr,ref}(s)\right) \tag{4.5}$$

Also, we define $G_q(s)$ and $r_{q0}(s)$ for the purpose of analysis in

$$G_q(s) = \frac{1 - C(s)}{s - \alpha_m} \tag{4.6}$$

$$r_{q0}(s) = \frac{\lambda_{q0}}{s - \alpha_m} \tag{4.7}$$

The closed-loop reference system may then be rewritten by describing

[2] It is this high gain control effort at high frequencies which resulted in lack of robustness for the MRAC controller.

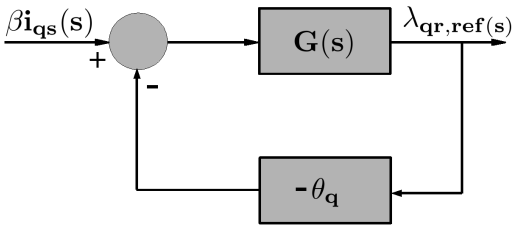


Figure 4.1: Closed-loop reference system for $q$-axis flux control

the system (4.3)–(4.4) in the frequency domain and substituting in Equations (4.5)–(4.7). The resulting system,

$$\lambda_{qr,ref}(s) = G(s)\left(\beta i_{qs}(s) + \theta_q \lambda_{qr,ref}(s)\right) + r_{q0}(s) \tag{4.8}$$

has the form of its first terms shown in Figure 4.1. Notice that the extra term $r_{q0}(s)$ is only the stable response due to non-zero initialization of the reference model. Since $\alpha_m$ is negative and $\lambda_{q0}$ is finite, it is clear from Equation (4.7) that the $L$-infinity norm $\|r_{q0}\|_{L_\infty}$ is finite.

The small gain theorem must be defined and used in order to show that the system (4.8) is stable. A proof of Theorem 2 below may be found in [27]. Also, Definition 3, which may be found in [41], gives the $\mathcal{L}_1$ norm of a stable, proper single input, single output system. This definition, along with the use of the small-gain theorem, is the backbone for proving stability of the aptly named $\mathcal{L}_1$ controller.

**Definition 2.** *A transfer function is said to be proper if the degree of its numerator is less than the degree of its denominator.*

**Definition 3.** *The $\mathcal{L}_1$ norm of a stable, proper single-input single-output system $G(s)$ is defined as*

$$\|G(s)\|_{\mathcal{L}_1} = \int_0^\infty |g(t)|dt$$

*where $g(t)$ is the impulse response of $G(s)$*

**Theorem 2** (Small-Gain Theorem)**.** *The interconnected system*

$$x(s) = G_1(s)\left(u(s) - G_2(s)x(s)\right)$$

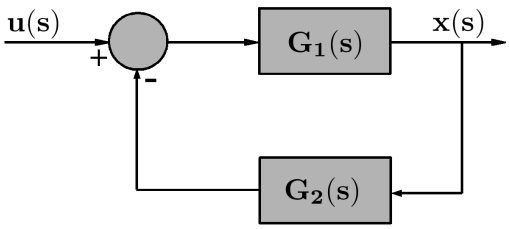


Figure 4.2: Interconnected systems

*shown in Figure 4.2, is stable if the* $\mathcal{L}_1$ *norms of each individual system satisfy*

$$\|G_1(s)\|_{\mathcal{L}_1}\|G_2(s)\|_{\mathcal{L}_1} < 1$$

The feedback terms of the closed-loop reference system in (4.8) have the same form as the interconnected system described in Theorem 2. The unknown parameter $-\theta_q$ plays the role of $G_2(s)$, and $G_q(s)$ plays the role of $G_1(s)$. By the small-gain theorem, the closed-loop reference system is stable if the condition

$$\|G_q(s)\|_{\mathcal{L}_1} L < 1 \tag{4.9}$$

is met, where

$$L = \max_{\theta \in \Theta} |\theta_q| \tag{4.10}$$

Here, $\Theta$ is the set of all possible values of $\theta_q$, based on the *a priori* known bounds.

In this controller, the bounds in Table 3.2 were used again, with $\alpha_m = -100$. Since the smallest possible value we know $\alpha$ can be is 2.94, we have $L = 97.06$. Thus, in order to satisfy the condition in Equation (4.9), we must have

$$\|G_q(s)\|_{\mathcal{L}_1} < 0.01 \tag{4.11}$$

Finding a filter $C(s)$ which satisfies the $\mathcal{L}_1$ norm condition for stability can often be difficult. In this case, however, it is possible to use a simple first order low pass Butterworth filter. Recently, a MATLAB toolbox has been developed and discussed in [42] which aids with the selection of this filter for an $\mathcal{L}_1$ controller in many situations. The form of this filter that will be used

is given as

$$C(s) = \frac{\omega_{nq}}{s + \omega_{nq}} \tag{4.12}$$

Here, $\omega_{nq}$ is the filter cutoff frequency and must be selected so that the $\mathcal{L}_1$ norm condition of Equation (4.9) is satisfied. With this simple filter, it is then easy to find analytically the impulse response $g_q(t)$ corresponding to the system $G_q(s)$. This impulse response is given as

$$g_q(t) = \frac{\frac{100}{e^{100t}} - \frac{\omega_{nq}}{e^{\omega_{nq}t}}}{\omega_{nq} - 100} \tag{4.13}$$

It has been verified by obtaining the absolute value of the impulse response $|g_q(t)|$ and integrating as described in Definition 3 that, with a cutoff frequency of $\omega_{nq} = 100$ radians per second, the $\mathcal{L}_1$ norm becomes $\|G_q(s)\|_{\mathcal{L}_1} = 0.0076$, which is small enough to satisfy the condition in (4.11).

Now, we have verified via the $\mathcal{L}_1$ norm condition that the new reference tracking system is stable. The final form of this control loop is depicted in Figure 4.3. Note that this is very similar to the MRAC scheme of Figure 3.3, except that the control law is different, and the state predictor must be used with the control input. This is unlike the MRAC controller, which was able to run the desired reference model and use its output as the state predictor.

The new control law which is used is given in (4.1). This control law will cause the state predictor in (3.13) to track the new closed loop reference system (4.8), rather than trying to exactly track the original desired reference model (3.9). Again, this controller uses the adaptive laws given in (3.23) and (3.24) in order to ensure that the state predictor tracks the actual plant dynamics.

### 4.2.2 Direct Axis Flux Control Loop

The analysis used to develop the $\mathcal{L}_1$ controller for the direct axis flux loop is very similar, but slightly more complicated than the previous analysis for the $q$-axis controller due to the fact that we now want to track a reference flux, rather than regulate to zero. The electric machine dynamics for the direct axis flux are given in (3.7). The proposed $\mathcal{L}_1$ controller will adapt to compensate for the unknown (and possibly time-varying) parameters $\alpha$ and $\beta$. Similarly to the $d$-axis flux control loop in the MRAC case, we will

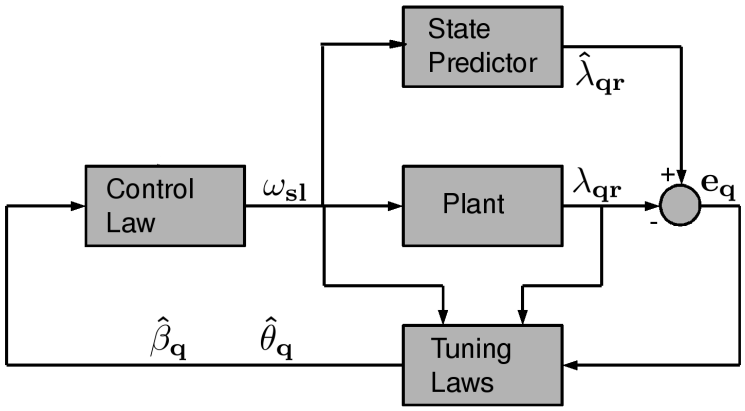


Figure 4.3: Final form of $\mathcal{L}_1$ controller for q-axis flux loop

treat the signals $w_{sl}$ and $\lambda_{qr}$ as known disturbance signals. However, if the quadrature axis control loop is performing as expected, this term should decay very quickly, and will not affect this control loop. The direct axis stator current $i_{ds}$ is the manipulated input to the induction machine, which is used to regulate the $d$-axis flux $\lambda_{dr}$ to the desired value.

Once again, the development of this controller begins by specifying a desired reference model, and re-writing the dynamics of the system to include similar terms to those in the desired model. The desired reference model will again be the stable linear system given in Equation (3.32), where $\alpha_m = -100$. The $d$-axis flux dynamics of the machine are then re-written as seen in (3.33) to include the $\alpha_m \lambda_{dr}$ term. There, the new unknown parameter $\theta_d = -(\alpha_m + \alpha)$ is introduced.

Next, the state predictor

$$\dot{\hat{\lambda}}_{dr} = \alpha_m \hat{\lambda}_{dr} + \omega_{sl}\lambda_{qr} + \hat{\beta}_d i_{ds} + \hat{\theta}_d \lambda_{dr} + \hat{\sigma}_d \tag{4.14}$$

is defined. Once again, this state predictor has almost the same form as the actual plant dynamics, except that the unknown parameters are replaced with estimates, denoted by the hat symbol above the parameter.

Notice that this time, an extra term $\hat{\sigma}_d$ is added to the state predictor. This term accounts for the nonlinear disturbance added by nonlinearity of the rotor inductance due to saturation. In essence, instead of considering the rotor time constant $\alpha$ as one slowly time-varying constant (accounted for in $\theta_d$), this state predictor considers it as a two-part unknown. First, the linear portion of the inductance contributes to the unknown $\alpha$ as before. Then, it

considers the nonlinear effects of rotor saturation as a separate disturbance signal $\sigma_d$. The reason this approach was not taken with the $q$-axis flux loop is that $\lambda_{qr}$ is regulated to zero, so the nonlinear effects of saturation do not have nearly as much effect on that loop, and it works fine without the $\sigma_d$ term. However, if $\lambda_{dr,ref}$ is high enough, the $d$-axis dynamics will be significantly affected by saturation.

To cause the state predictor to track the actual machine behavior, the same adaptive laws from (3.44) and (3.45) will be used to tune the estimates $\hat{\theta}_d$ and $\hat{\beta}_d$. The Lyapunov analysis performed on pages 30–32 remains exactly the same for those terms, except that an extra term is added to the candidate Lyapunov function to include the error $\tilde{\sigma}_d = \hat{\sigma}_d - \sigma_d$. The candidate Lyapunov function used in this case is

$$V(e_d, \tilde{\beta}_d, \tilde{\theta}_d, \tilde{\sigma}_d) = \frac{1}{2}e_d^2 + \frac{1}{2\gamma}(\tilde{\beta}_d^2 + \tilde{\theta}_d^2 + \tilde{\sigma}_d^2) \tag{4.15}$$

Here, all error term definitions are the same as seen before in Equations (3.36)–(3.38) on page 30.

Taking the derivative of this Lyapunov function and substituting the system dynamics (including the new unknown $\sigma_d$ as part of the actual dynamics) results in the following:

$$\begin{aligned}\dot{V}(e_d, \tilde{\beta}_d, \tilde{\theta}_d, \tilde{\sigma}_d) = \alpha_m e_d^2 + \tilde{\beta}_d(i_{ds}e_d + \frac{1}{\gamma}\dot{\hat{\beta}}_d) \\ + \tilde{\theta}_d(\lambda_{dr}e_d + \frac{1}{\gamma}\dot{\hat{\theta}}_d) + \tilde{\sigma}_d(e_d + \frac{1}{\gamma}\dot{\hat{\sigma}}_d)\end{aligned} \tag{4.16}$$

Now, the tuning laws in (3.45) and (3.44) are used for $\hat{\theta}_d$ and $\hat{\beta}_d$, with the newly introduced tuning law

$$\dot{\hat{\sigma}}_d = -\gamma \text{Proj}(e_d) \tag{4.17}$$

used for the estimate $\hat{\sigma}_d$.

Once again, the Proj$(\cdot)$ operation only has any effect when the estimate would go outside *a priori* known bounds on the parameter. In this case, these bounds are reasonable limits on how much saturation will affect the flux. These tuning laws result in the bound on $\dot{V}$ given below.

$$\dot{V}(e_d, \tilde{\beta}_d, \tilde{\theta}_d, \tilde{\sigma}_d) \leq \alpha_m e_d^2 \leq 0 \tag{4.18}$$

Theorem 1 may now be applied to guarantee that all error signals remain bounded, and that $e_d \to \infty$ as $t \to \infty$. Thus, we can conclude that the state predictor and plant dynamics will track one another as $t \to \infty$.

To track the desired reference behavior with the state predictor, a new control law is proposed. Once again, a stable low-pass filter will be used in the control law to prevent high frequency control efforts. These high-frequency dynamics often lie in the realm of unmodeled dynamics for the system, so it is best to avoid high gain control over these unknown dynamics for reasons of robustness.

A slightly different approach will be taken for the $d$-axis flux loop compared to the $\mathcal{L}_1$ controller for the $q$-axis loop, for two reasons. First, the control input to this system, $i_{ds}$, is directly multiplied by the unknown parameter $\beta$. Secondly, we desire to track the desired flux value $\lambda_{dr,ref}$, rather than just regulating it to zero. The procedure for this control loop is based on the controller presented in [43].

The control laws used by the $d$-axis flux $\mathcal{L}_1$ control loop are

$$i_{ds}(s) = -k_d D_d(s) r_{du}(s) \tag{4.19}$$

$$r_{du} = \hat{\beta}_d i_{ds} + \hat{\theta}_d \lambda_{dr} + \hat{\sigma}_d + \alpha_m \lambda_{dr,ref} + \omega_{sl} \lambda_{qr} \tag{4.20}$$

The transfer function $D_d(s)$ is a strictly proper stable system yet to be designed, along with a strictly positive gain $k_d$. Notice that this control law has internal feedback, since $i_{ds}$ appears appears in $r_{du}$, which affects the control law for itself in (4.19). This control law may also be re-written in the frequency domain by solving for $i_{ds}$ as seen in

$$i_{ds}(s) = -\frac{\hat{C}_d(s)}{\hat{\beta}_d} \left( \hat{\theta}_d(s) \lambda_{dr}(s) + \hat{\sigma}_d(s) + \alpha_m \lambda_{dr,ref}(s) + \omega_{sl}(s) \lambda_{qr}(s) \right) \tag{4.21}$$

where $\hat{C}_d(s)$ is given below:

$$\hat{C}_d(s) = \frac{k_d \hat{\beta}_d D_d(s)}{1 + k_d \hat{\beta}_d D_d(s)} \tag{4.22}$$

Substituting this control law into the state predictor (4.14) results in the

following dynamic behavior for the state predictor:

$$\dot{\hat{\lambda}}_{dr} = \alpha_m \hat{\lambda}_{dr} + \left(1 - \hat{C}_d(s)\right)\left(\hat{\theta}_d \lambda_{dr} + \hat{\sigma}_d + \omega_{sl}\lambda_{qr}\right) - \hat{C}_d(s)\alpha_m \lambda_{dr,ref} \quad (4.23)$$

Notice that if $\hat{C}_d(s)$ were equal to one, the middle term of Equation (4.23) would be cancelled, so that the state predictor would exactly assume the dynamics of the originally suggested reference system of Equation (3.32) on page 29. However, $\hat{C}_d(s)$ will be a low pass filter, so that this cancellation does occur partially at lower frequencies.

Now, a new closed-loop reference model will be defined to analyze the stability of the system with the above control law. Equations (4.24)–(4.26) give the resulting closed loop reference system, along with its feedback input:

$$\dot{\lambda}_{dr,mod}(t) = \alpha_m \lambda_{dr,mod}(t) + \omega_{sl}(t)\lambda_{qr}(t) + \beta i_{ds}(t) + \theta_d \lambda_{dr,mod}(t) \quad (4.24)$$

$$\lambda_{dr,mod}(0) = \lambda_{d0} \quad (4.25)$$

$$i_{ds,mod} = -\frac{C_d(s)}{\beta}\left(\theta_d \lambda_{dr,mod} + \sigma_d + \alpha_m \lambda_{dr,ref} + \omega_{sl}\lambda_{qr}\right) \quad (4.26)$$

Also, we define $H_d(s)$, $G_d(s)$, $C_d(s)$, and $r_{d0}(s)$ for the purpose of analysis as

$$H_d(s) = \frac{1}{s - \alpha_m} \quad (4.27)$$

$$G_d(s) = H_d(s)(1 - C_d(s)) \quad (4.28)$$

$$C_d(s) = \frac{k_d \beta D_d(s)}{1 + k_d \beta D_d(s)} \quad (4.29)$$

$$r_{d0}(s) = H_d(s)\lambda_{d0} \quad (4.30)$$

The closed-loop reference system may then be written by representing the open loop system (4.24)–(4.25) in the frequency domain and substituting the feedback law (4.26), along with the transfer functions from Equations (4.27)–(4.30). This results in the new reference system shown in

$$\begin{aligned}\lambda_{dr,mod}(s) = G_d(s)\left(\theta_d \lambda_{dr,mod}(s) + \sigma_d(s) + \omega_{sl}(s)\lambda_{qr}(s)\right) \\ - \alpha_m H_d(s) C_d(s) \lambda_{dr,ref}(s) + r_{d0}(s)\end{aligned} \quad (4.31)$$

Here, there are three terms for this system. The extreme right-hand term, $r_{d0}(s)$, is simply a linearly decaying effect of nonzero reference model initial-

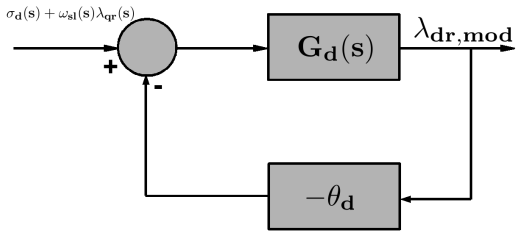


Figure 4.4: First term of $d$-axis reference system

ization. If it is nonzero, it will decay quickly since this term is stable. The middle term, $-\alpha_m H_d(s) C_d(s) \lambda_{dr,ref}(s)$, is also a stable system, as long as $C_d(s)$ is a proper stable transfer function. Since $D_d(s)$ will be chosen to be a proper stable function, it is clear that $C_d(s)$ will also be stable, by its definition in Equation (4.29). Since $-\alpha_m$ is just a positive constant, and $\lambda_{dr,ref}$ is bounded (since it is the desired flux value), this middle term will always remain bounded. The first term, $G_d(s)\left(\theta_d \lambda_{dr,mod}(s) + \sigma_d(s) + \omega_{sl}(s)\lambda_{qr}(s)\right)$, must now be shown to be stable as well in order to prove that the closed-loop reference system is stable.

The first term of (4.31) takes exactly the same form as the interconnected system described in Theorem 2. In this case, the form of the first term, $G_d(s)\left(\theta_d \lambda_{dr,mod}(s) + \sigma_d(s) + \omega_{sl}(s)\lambda_{qr}(s)\right)$, can be seen in Figure 4.4. Thus, by Theorem 2, the closed-loop reference system will be stable as long as the condition

$$\|G_d(s)\|_{\mathcal{L}_1} L_d < 1 \tag{4.32}$$

is met, with the definition of $L_d$ as

$$L_d = \max_{\theta_d \in \Theta} |\theta_d| \tag{4.33}$$

The terms $\sigma_d(s)$ and $\omega_{sl}(s)\lambda_{qr}(s)$ are all bounded. Here, the set $\Theta$ is the set of all possible values of $\theta_d$, based on the *a priori* known bounds.

To ensure that $G_d(s)$ satisfies the $\mathcal{L}_1$ norm condition, the transfer function $D_d(s)$ and the constant gain $k_d$ must be designed accordingly. In this case, the form of $G_d(s)$ is too complicated to easily compute analytically. Therefore, $D_d(s)$ was chosen to be a stable, first order, low pass Butterworth filter, and the impulse response of the resulting $G_d(s)$ was simulated for varying cutoff

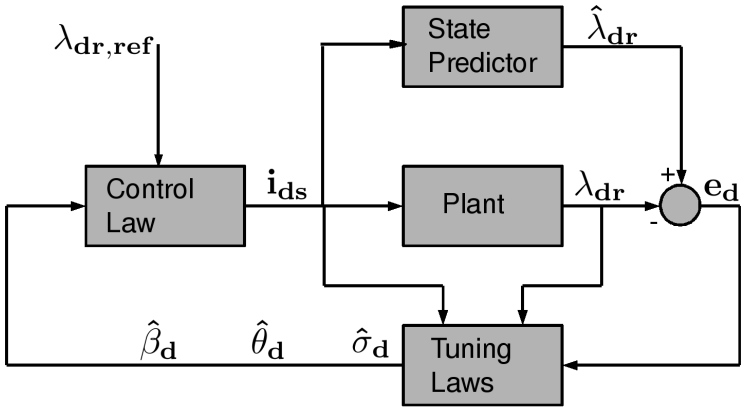


Figure 4.5: Final form of $\mathcal{L}_1$ controller for d-axis flux loop

frequencies and gains $k_d$. The final form of $D_d(s)$ is chosen as

$$D_d(s) = \frac{\omega_{nd}}{1 + \omega_{nd}} \tag{4.34}$$

It was determined via simulation that selecting the cutoff frequency $\omega_{nd} = 20$ radians per second, and feedback gain $k_d = 7$ results in an $\mathcal{L}_1$ norm of $\|G_d(s)\|_{\mathcal{L}_1} \leq 0.0072$ over all possible values of $\beta$. Again, we have $L_d = 97.06$ in this case. Multiplying the $\mathcal{L}_1$ norm of $G_d(s)$ by $L_d$ results in a product of $0.6988 < 1$, so the condition for Theorem 2 is met.

Now, we have verified via the $\mathcal{L}_1$ norm condition that the new reference tracking system is stable. The final form of this control loop is depicted in Figure 4.5. Note that this is very similar to the MRAC scheme of Figure 3.4, except that the control law is different, and the state predictor must be used with the control input, rather than just running the original desired reference model. Also note that the adaptive estimates $\hat{\beta}_d$, $\hat{\theta}_d$, and $\hat{\sigma}_d$ are used by the state predictor.

The new control law which is used is given in Equations (4.19) and (4.20). This control law will cause the state predictor (4.14) to track the new closed loop reference system (4.31), rather than trying to exactly track the original desired reference model (3.32). Again, this controller uses the adaptive laws given in Equations (3.44), (3.45), and (4.17) in order to ensure that the state predictor tracks the actual plant dynamics.

### 4.2.3 Speed Control Loop

The $\mathcal{L}_1$ adaptive controller for the speed control loop is very similar to the controller devised in the last section for the $d$-axis flux control loop. Recall again the speed dynamics for an induction machine, given in (3.6). The goal of this controller will be to track a desired speed by manipulating the torque. The unknown parameters of this system are $a$, $\mu$, and $\sigma$. Since we know that $\mu$ and $\lambda_{dr}$ are strictly greater than zero (the $d$-axis flux loop is meant to regulate the flux to a positive value), we may use $i_{qs}$ as an input to this system. Positive $\mu$ and $\lambda_{dr}$ are necessary to ensure controllability.

As done before with the MRAC speed control loop, we will consider

$$u = \lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}$$

as the input to the system for the control design. Then, the actual input $i_{qs}$ may be calculated based on this $u(t)$, along with the signals $\lambda_{dr}$, $\lambda_{qr}$, and $i_{ds}$.

The first step taken toward the design of this controller is to specify the desired nominal reference model. The stable linear system in Equation (3.48) on page 33 will be used once again for this purpose. Equation (3.49) then shows a re-written version of the system dynamics, which introduces the new parameter $\theta = -(a + a_m)$. For the $\mathcal{L}_1$ controller, the same state predictor will also be used as in the MRAC controller. This state predictor is given in (3.50) on page 33.

To cause the state predictor to track the actual system behavior, the same adaptive laws from (3.55)–(3.57) will be used. The Lyapunov analysis performed on pages 34–36 remains exactly the same. Thus, we can conclude that the state predictor and plant dynamics will track one another as $t \to \infty$.

The control law is where the speed loop $\mathcal{L}_1$ controller differs from the MRAC controller. Instead of trying to cause the state predictor to perfectly match the desired reference model behavior with complete cancellation of estimation terms, a low-pass filter will be used so that only partial cancellation of terms is assumed. Because of this, the controller will not try to expend control effort to try and track very high frequency dynamics, which may be unmodeled anyway. This is how the $\mathcal{L}_1$ controller achieves robustness. The control law presented here closely follows the scheme proposed in [43]. This

control law is expressed as

$$u(s) = -kD(s)\left(\hat{\mu}u + \hat{\theta}\omega_m + \hat{\sigma} + a_m\omega_{ref}\right) \tag{4.35}$$

where $k > 0$ is a feedback gain, and $D(s)$ is a stable proper transfer function to be designed.

As in the $d$-axis flux control law, this control law has internal feedback, since $u(s)$ also appears on the right-hand side of the equation. This control law may be rewritten and described as in

$$u(s) = -\hat{C}(s)\frac{1}{\hat{\mu}}\left(\hat{\theta}\omega_m + \hat{\sigma} + a_m\omega_{ref}\right) \tag{4.36}$$

where $\hat{C}(s)$ is defined in

$$\hat{C}(s) = \frac{kD(s)\hat{\mu}(s)}{1 + kD(s)\hat{\mu}(s)} \tag{4.37}$$

Substituting this control law into the state predictor that is used in this controller, given in (3.50), results in the following dynamic equations for the closed loop state predictor:

$$\dot{\hat{\omega}}_m = a_m\hat{\omega}_m + (1 - \hat{C}(s))\left(\hat{\theta}\omega_m + \hat{\sigma}\right) - \hat{C}(s)a_m\omega_{ref} \tag{4.38}$$

This equation illustrates the fact that at low frequencies, the $\mathcal{L}_1$ controller will achieve partial cancellation of terms to behave like the desired reference model. Notice that if $\hat{C}(s) = 1$, the state predictor would reduce to the desired reference model. Instead, it now tracks a new closed-loop reference model, which is a more realistic goal.

The new closed-loop reference model that the state predictor will track, along with its feedback input, is

$$\dot{\omega}_{m,mod}(t) = a_m\omega_{m,mod}(t) + \mu u(t) + \sigma(t) + \theta\omega_{m,mod}(t) \tag{4.39}$$

$$\omega_{m,mod}(0) = \omega_{m0} \tag{4.40}$$

$$u_{mod}(s) = -C(s)\frac{1}{\mu}\left(\theta\omega_{m,mod}(s) + \sigma(s) + a_m\omega_{ref}(s)\right) \tag{4.41}$$

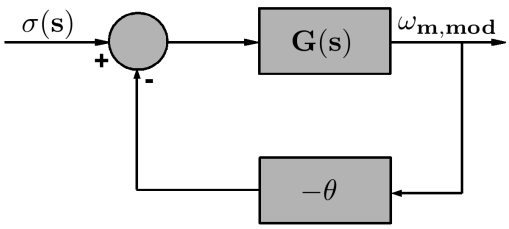


Figure 4.6: First term of speed tracking reference system

Also, we define $H(s)$, $G(s)$, $C(s)$, and $r_0(s)$ for purposes of analysis as

$$H(s) = \frac{1}{s - a_m} \tag{4.42}$$

$$G(s) = H(s)\left(1 - C(s)\right) \tag{4.43}$$

$$C(s) = \frac{kD(s)\mu}{1 + kD(s)\mu} \tag{4.44}$$

$$r_0(s) = H(s)\omega_{m0} \tag{4.45}$$

As before, we may now re-write the closed-loop reference system dynamics in the frequency domain by substituting the feedback law (4.41) and the transfer functions (4.42)–(4.45) into a frequency domain version of the reference system. The new resulting closed-loop is

$$\begin{aligned}\omega_{m,mod} = G(s)\left(\theta\omega_{m,mod}(s) + \sigma(s)\right) \\ - a_m H(s)C(s)\omega_{ref}(s) + H(s)\omega_{m0}\end{aligned} \tag{4.46}$$

Now, we must prove that this system is stable. The extreme right-hand term, $H(s)\omega_{m0}$, is simply a stable response due to any possible nonzero initialization error. This term will rapidly decay, since $H(s)$ is stable. The middle term of (4.46), $-a_m H(s)C(s)\omega_{ref}(s)$, is also bounded. This is because $D(s)$ will be chosen so that $C(s)$ is stable and proper, and $H(s)$ is stable and proper, along with the fact that the desired speed tracking input $\omega_{ref}$ is bounded. All that remains is to show that the term with $G(s)$ in it is stable as well.

In order to show that the term $G(s)\left(\theta\omega_{m,mod}(s) + \sigma(s)\right)$ in (4.46) is stable, we will again use the small-gain theorem, shown in Theorem 2. This term is shown graphically in Figure 4.6. The form of this term coincides perfectly with the interconnected systems described in Theorem 2. Thus, we may

conclude that the closed-loop reference system (4.46) is stable if the condition

$$\|G(s)\|_{\mathcal{L}_1} L_\omega < 1 \tag{4.47}$$

is met, where $L_\omega$ is defined as

$$L_\omega = \max_{\theta \in \Theta_\omega} |\theta| \tag{4.48}$$

Now, in order to ensure that this condition is met, we must choose the feedback gain $k$ and the transfer function $D(s)$. In this case, we can use the simplest choice of

$$D(s) = \frac{1}{s}$$

for the transfer function $D(s)$. In order to proceed with the design, the impulse responses of $G(s)$ were then simulated for a range of gain values for $k$, and integrated to find the $\mathcal{L}_1$ norm $\|G(s)\|_{\mathcal{L}_1}$. It was determined via simulation using MATLAB that a feedback gain of $k = 0.6$ results in the following bound:

$$\|G(s)\|_{\mathcal{L}_1} \leq 0.014$$

For this controller, the desired speed loop convergence rate was chosen to be $a_m = -40$. Thus, even if the unknown parameter is arbitrarily small, *theta* cannot be greater than 40, so $L_\omega = 40$ in our case. Multiplying the norm $\|G(s)\|_{\mathcal{L}_1}$ with $L_\omega$ results in 0.56, so the condition in Theorem 2 is met.

Now, we have once again verified via the $\mathcal{L}_1$ norm condition that the new reference tracking system is stable. The final form of this control loop is depicted in Figure 4.7. Note that this is very similar to the MRAC scheme of Figure 3.5, except that the control law is different, and the state predictor must be used with the control input, rather than just running the original desired reference model. Also note that the adaptive estimates $\hat{\mu}$, $\hat{\theta}$, and $\hat{\sigma}$ are used by the state predictor.

The new control law which is used is given in Equation (4.35). This control law will cause the state predictor (3.50) on page 33 to track the new closed loop reference system (4.46), rather than trying to exactly track the original desired reference model (3.48). Again, this controller uses the adaptive laws given in (3.55)–(3.57) in order to ensure that the state predictor tracks the

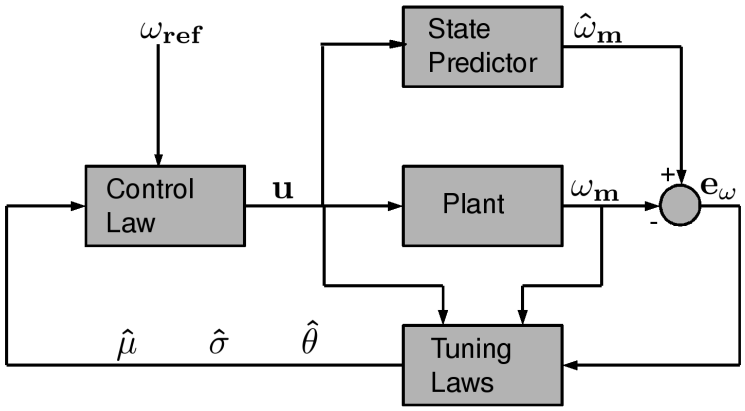


Figure 4.7: Final form of $\mathcal{L}_1$ controller for the speed loop

actual plant dynamics. Finally, the actual input $i_{qs}$ is calculated using

$$i_{qs} = \frac{1}{\lambda_{dr}} \left( u + \lambda_{qr} i_{ds} \right) \tag{4.49}$$

This equation is simply achieved by solving the $u$ Equation (3.47) for $i_{qs}$. Since the $d$-axis flux loop ensures that $\lambda_{dr}$ never approaches zero, $i_{qs}$ will remain bounded.

## 4.3 Simulation of $\mathcal{L}_1$ Controller

In this section, the $\mathcal{L}_1$ adaptive control system is evaluated using simulations. Table 3.2 on page 15 shows the unknown parameter values used, and the *a priori* known bounds used for each parameter. For comparison purposes, these are the same parameters as used in the MRAC controller. Again, in the case of the speed reference model, the convergence rate $a_m = -40$ is used, while a faster desired convergence of $\alpha_m = -100$ is used for the $q$ and $d$ axis flux control loops.

Since the $\mathcal{L}_1$ controller has a filter in the control law, very high frequency components of the tuned parameter estimates will not enter the control channel. Because of this, it is acceptable to increase the tuning rate as high as possible, without losing robustness. In fact, faster tuning is now better for robustness, since the low-pass filter will get rid of oscillations most at the highest frequencies. Thus, all the simulations in this section (except for the first one) will be done with very fast tuning, at $\gamma = 1 \times 10^5$. This will result

in an increase in both performance and robustness.

Figure 4.8 shows the simulated performance of the $\mathcal{L}_1$ controller when all unknown parameters in the machine are kept at their nominal values seen in Table 3.2, with a slow adaptation rate of $\gamma = 100$. Note that while the machine parameters are at their nominal values, the adaptive controller is still initialized with the incorrect guesses shown in the right-most column of Table 3.2. This tuning rate is fast enough to cause the flux control loops to very closely track the desired model reference behavior. However, the speed control loop takes almost 0.1 seconds before it behaves similarly to the desired reference model. This response is very similar to the response of the MRAC controller in Figure 3.6. This is because with slow tuning, the adaptive estimates have mostly low-frequency components, which will pass through the filter.

Now, if we increase the rate of adaptation to $\gamma = 1 \times 10^5$, the tuned parameter estimates will have much faster convergence rates, as seen in Figure 4.9. Now, both the speed and flux loops track their closed-loop reference system very accurately. Even from the very start of the simulation, the error between the reference system and output is less than 1%. Again, the true values of each parameter remain unchanged at the nominal values given in Table 3.2.

Recall that with the MRAC controller, the fast adaptation resulted in very large high frequency oscillations in the current commands $i_{ds}$ and $i_{qs}$. This time, however, the control effort is much more smooth, resulting in a smoother system response. However, the system still tracks the closed-loop reference system. The trade-off this time is that the closed-loop reference system is no longer the first order linear system defined in (3.48). Now, the controller is tracking a more realistic, albeit slightly more complicated system. The big advantage is that this system is tracked much more accurately than the one before, with a much more robust control effort.

Now, we will examine the response of this controller when the unknown parameters change. For the rest of these simulations, fast adaptation ($\gamma = 1 \times 10^5$) is used. First, the controller is tested when the load torque changes from 5 to 10 Nm. From the results in Figure 4.10, the speed response varies by less than 0.96% during this step change in torque. The flux control loops track their reference system as well. However, the reference system is not as ideal as the one before. This time, the quadrature flux does have a small instantaneous error when the load is changed. This error is a response by

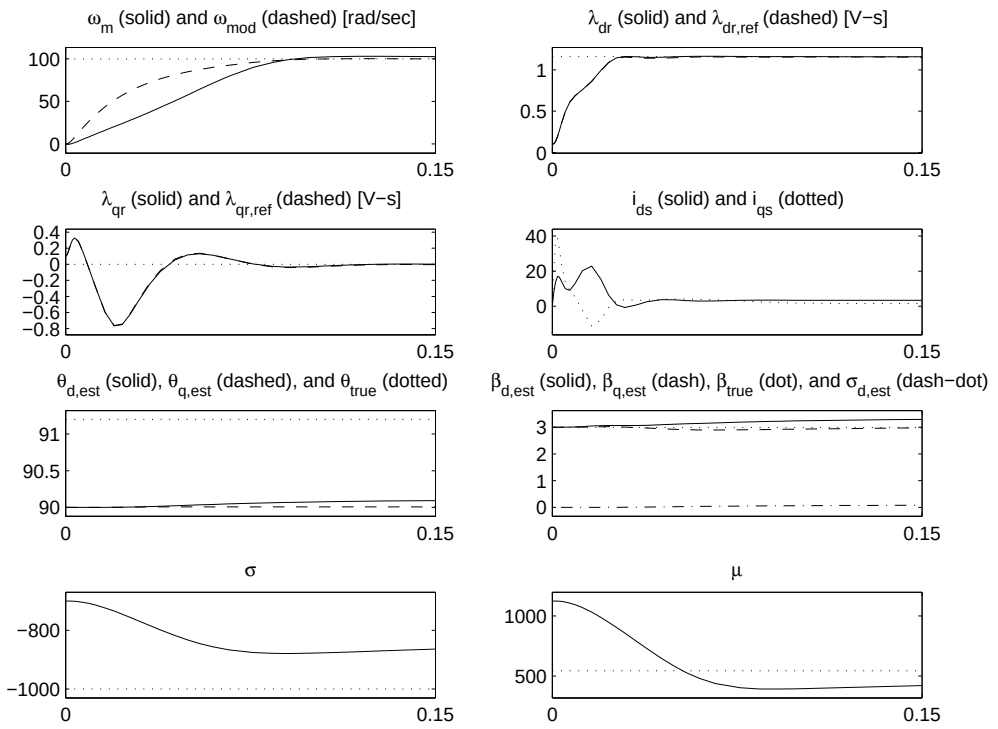


Figure 4.8: $\mathcal{L}_1$ performance with $\gamma = 100$ and no parameter change, $x$ axis in seconds

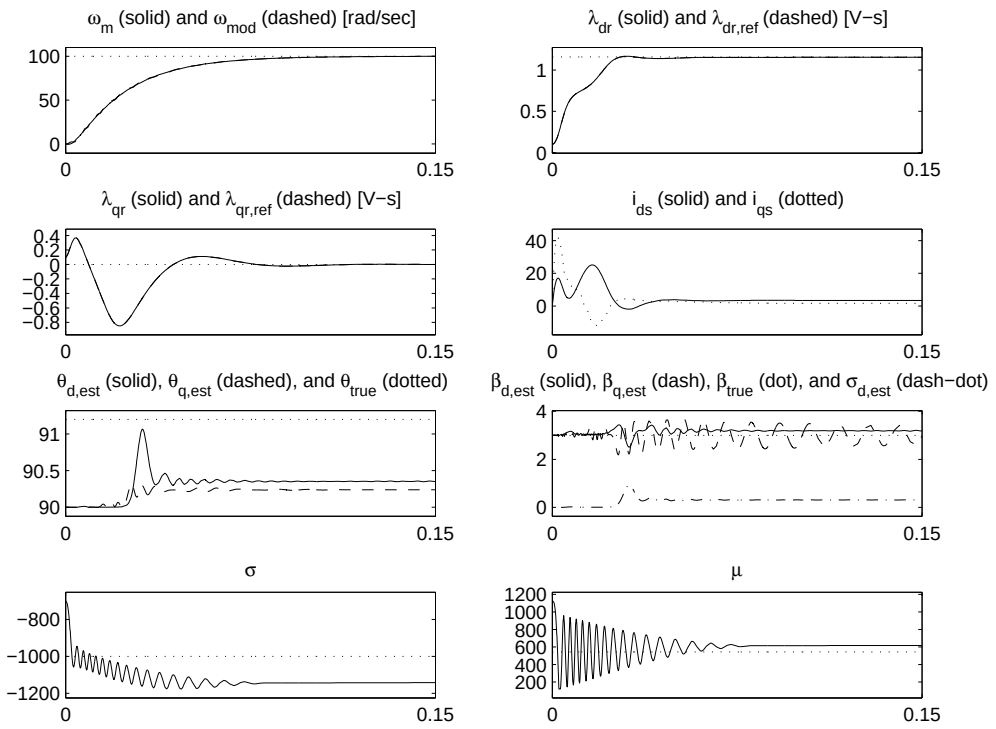


Figure 4.9: $\mathcal{L}_1$ performance with $\gamma = 1 \times 10^5$ and no parameter change, $x$ axis in seconds

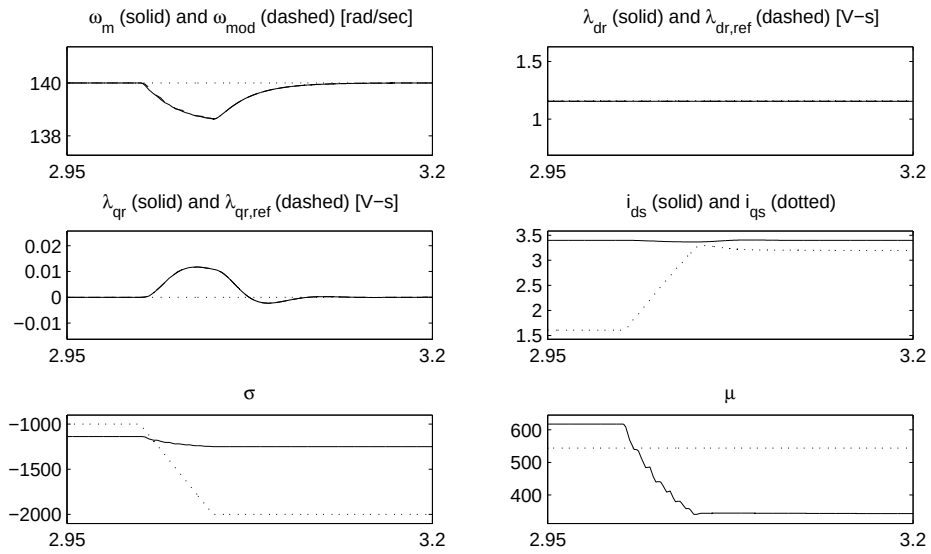


Figure 4.10: $\mathcal{L}_1$ performance with torque change from 5 to 10 Nm, $x$ axis in seconds

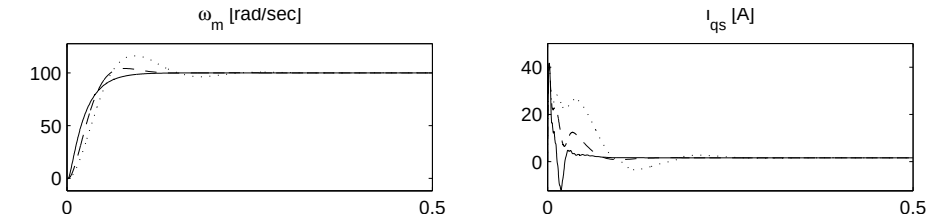


Figure 4.11: $\mathcal{L}_1$ performance varying inertia: solid line, $J$; dashed line, $4J$; dotted line, $8J$; $x$ axis in seconds

the reference system itself, which is being tracked.

Next, the drive response will be tested for different inertias. Figure 4.11 shows the speed response and required quadrature axis current for different values of inertia. Here, more control effort is required when the inertia is greater, but the speed tracking remains relatively consistent. This result is very similar to that of the MRAC controller in Figure 3.9.

The next series of tests will simulate the $\mathcal{L}_1$ controller's response to changes in machine electrical parameters, namely the rotor resistance, rotor inductance, and mutual inductance. Figures 4.12 and 4.13 show the direct axis flux response, control inputs, and flux parameter estimates when $R_r$ is abruptly changed by $-50\%$ and $+100\%$, respectively. Simulations of decreasing and increasing rotor inductance $L_r$ may be seen in Figures 4.14 and 4.15, respectively. Finally, the controller's performance with abrupt changes in mutual

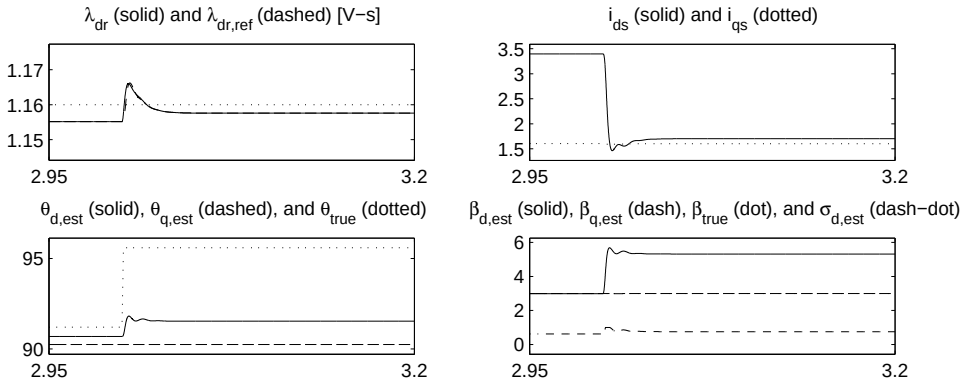


Figure 4.12: $\mathcal{L}_1$ performance with $R_r$ abruptly changed by $-50\%$, $x$ axis in seconds

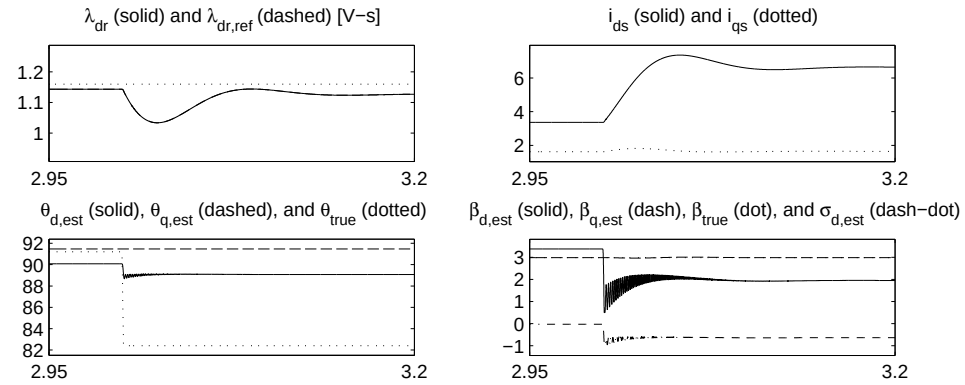


Figure 4.13: $\mathcal{L}_1$ performance with $R_r$ abruptly changed by $+100\%$, $x$ axis in seconds

inductance $L_m$ is given in Figures 4.16 and 4.17. All of these simulations are done with fast adaptation ($\gamma = 1 \times 10^5$).

From Figures 4.12–4.17, it is apparent that changes in the electrical characteristics of the machine have an impact on the system's performance. However, the response due to these changes is much better than most non-adaptive controllers. Table 4.1 summarizes the maximum instantaneous speed and flux errors resulting from the electrical parameter changes described above. These results look similar to the instantaneous errors of the MRAC controller summarized in Table 3.3. The flux error is 8.87% greater for the $\mathcal{L}_1$ controller when $R_r$ is increased. Also, the $\mathcal{L}_1$ controller has 4.92% greater error when $L_r$ is decreased. However, this is because that error is present in the slightly more complicated reference model. It is also clear

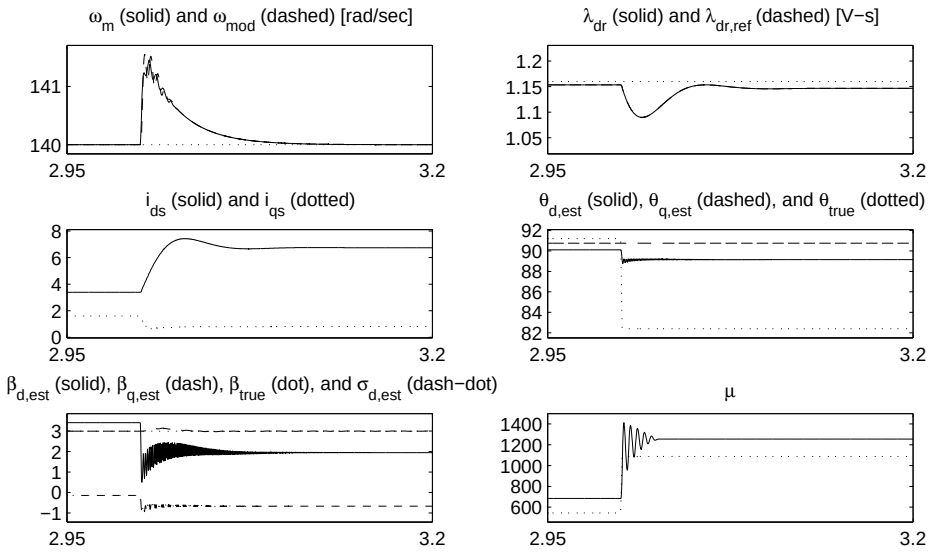


Figure 4.14: $\mathcal{L}_1$ performance with $L_r$ abruptly changed by $-50\%$, $x$ axis in seconds

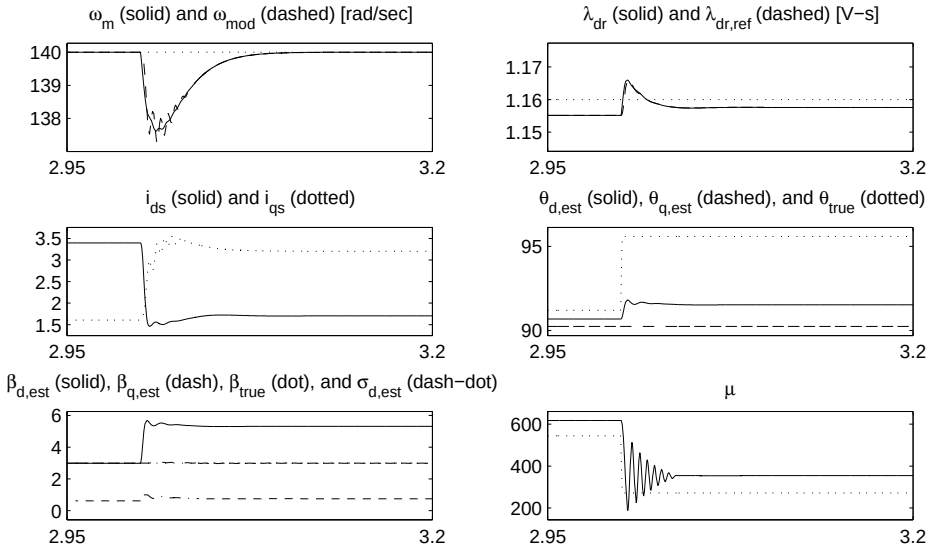


Figure 4.15: $\mathcal{L}_1$ performance with $L_r$ abruptly changed by $+100\%$, $x$ axis in seconds

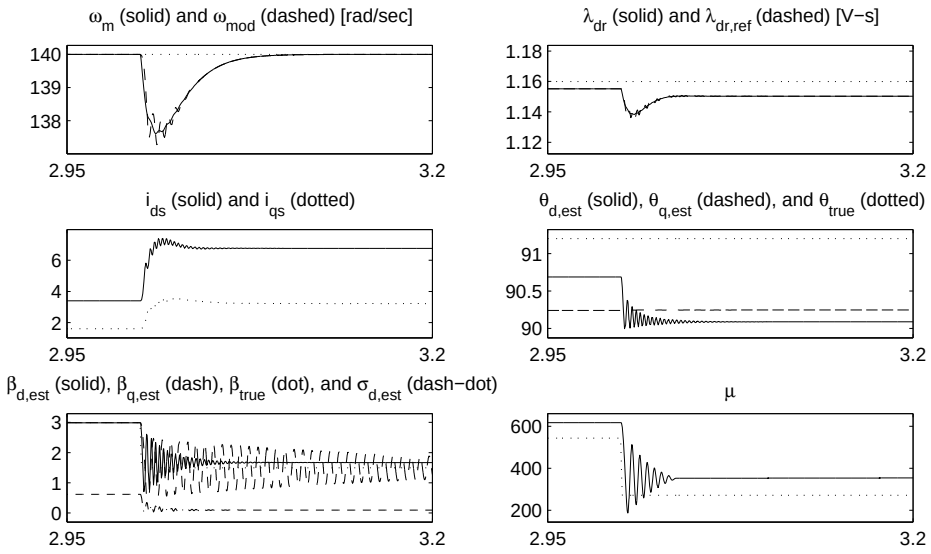


Figure 4.16: $\mathcal{L}_1$ performance with $L_m$ abruptly changed by $-50\%$, $x$ axis in seconds

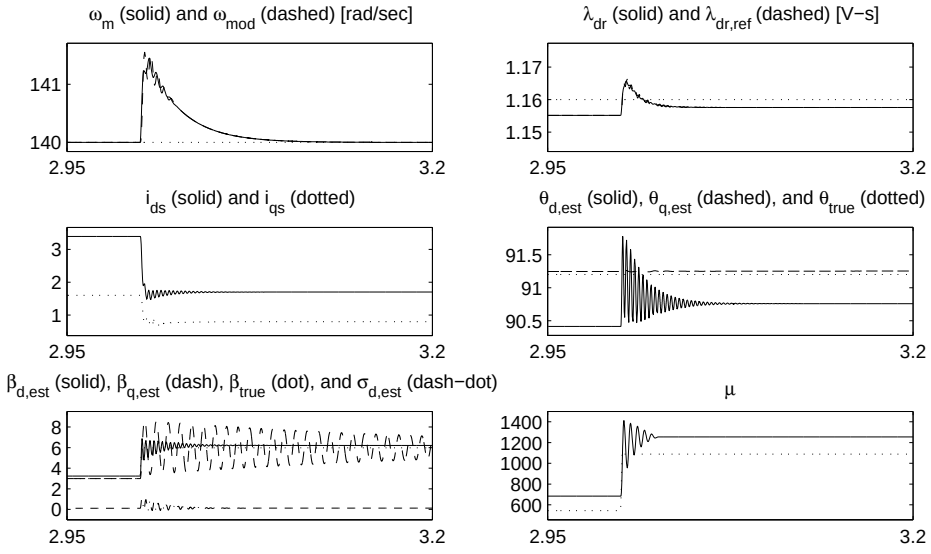


Figure 4.17: $\mathcal{L}_1$ performance with $L_m$ abruptly changed by $+100\%$, $x$ axis in seconds

Table 4.1: $\mathcal{L}_1$ Error Due to Change in Electrical Parameters

| Parameter | Variation | Speed error [%] | $d$-axis flux error [%] |
|---|---|---|---|
| $R_r$ | $-50\%$ | negligible | 0.94 |
| | $+100\%$ | negligible | 10.34 |
| $L_r$ | $-50\%$ | 1.07 | 6.47 |
| | $+100\%$ | 1.64 | 1.62 |
| $L_m$ | $-50\%$ | 1.75 | 1.81 |
| | $+100\%$ | 1.29 | .82 |

from these figures that the $\mathcal{L}_1$ controller does have some small steady-state error in the $d$-axis flux controller. This error is also due to the fact that the new closed-loop reference system has some steady state error. This may be corrected by using an integral term in the reference system, and will be discussed later.

Thus far, all these $\mathcal{L}_1$ simulations have assumed that the machine is modeled correctly, with uncertainty of parameters. Now, it is important to show the characteristics of the system when any unmodeled dynamics are introduced. The next simulation conducted involves adding a single stable pole between the controller and the machine. This pole is at $-800$ in the complex plane, so it is also much faster than the modeled dynamics of the machine (hence, its effect should damp out very quickly). This test is the same as the test with unmodeled dynamics performed on the MRAC controller. Recall that with fast tuning, the MRAC controller demonstrated undesirable robustness characteristics due to oscillations in the control channel. Here, we will also use fast tuning ($\gamma = 1 \times 10^5$).

In this case, the response with an unmodeled pole is shown in Figure 4.18. Now, it can be seen that this response is much different than the response without the unmodeled pole from Figure 3.17. Comparing the two figures, it is clear that with the unmodeled pole, the $\mathcal{L}_1$ system is much more stable. It does take longer for the flux response to settle to the desired value, but it is still very fast. However, since a smooth, low frequency current command for $i_{qs}$ and $i_{ds}$ is used, the system is much more robust to model uncertainties.

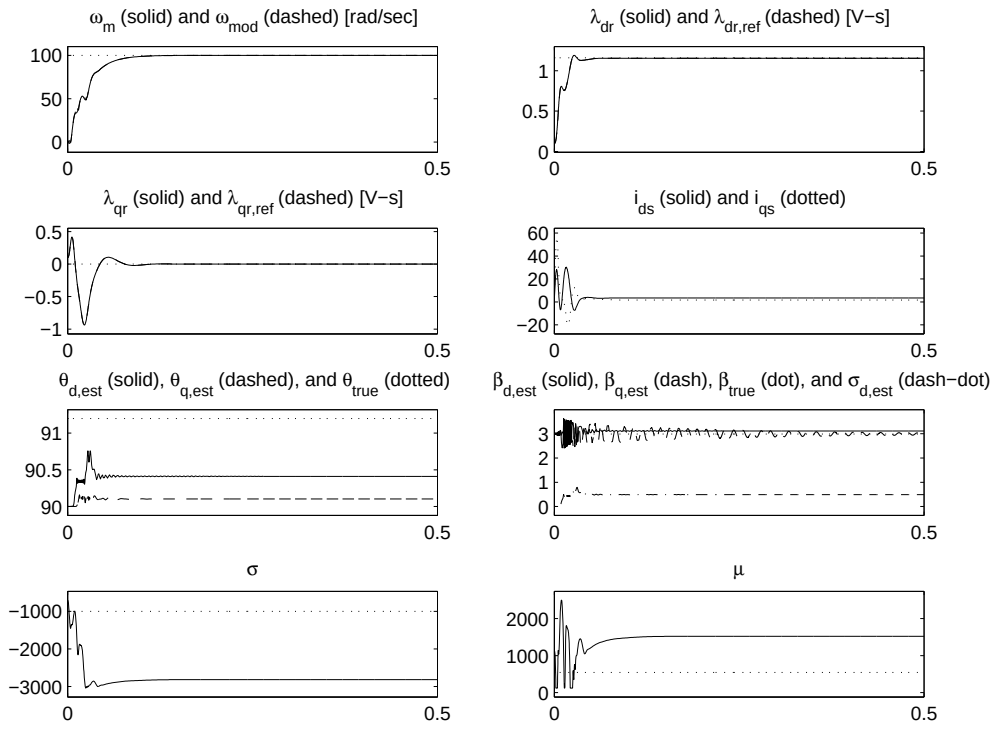


Figure 4.18: $\mathcal{L}_1$ performance unmodeled fast pole, $x$ axis in seconds

## 4.4 Analysis and Conclusions

The above figures show that the $\mathcal{L}_1$ controller works well, even when the system is not modeled perfectly. The tuning rate $\gamma$ may be increased as much as the controller is capable of calculating with an increase in both performance and robustness.

As summarized in Table 4.1, it is clear that the control system can handle changes in the unknown parameter values. Also, since the simulations were performed with incorrect initial guesses of the unknown parameters, we can conclude that the $\mathcal{L}_1$ controller works well with parameter uncertainty.

Even in case where an unmodeled pole is added to the plant dynamics, the $\mathcal{L}_1$ controller maintains stability. This is due to the fact that the matching assumptions are more relaxed. Recall page 20, where this matching assumption is discussed for the case of the MRAC controller. In that case, perfect cancellation was assumed, so that the controller could directly track the desired reference model. In the $\mathcal{L}_1$ case, a new, realistic closed-loop reference system is tracked. This time, the matching assumption only assumes that

cancellation can occur at lower frequencies. Therefore, the stability analysis is valid even with high frequency unmodeled dynamics.[3]

For the simulation in Figure 4.18 the extra pole at $-800$ in the complex plane affects $i_{qs}$ and $i_{ds}$ between the controller and motor. In this case, the dynamics in (3.6)–(3.7) would have two extra states, as shown in (3.59) and (3.60), where $i_{ds,cmd}$ and $i_{qs,cmd}$ are the desired system inputs as manipulated by the controller. Now, even though there are high magnitude, high frequency oscillations in the adaptive parameter estimates, these oscillations do not enter the control channel. Therefore, they do not cause interference with the unmodeled fast pole.

In [33], several counterexamples are given showing how a traditional MRAC system may become unstable in the presence of unmodeled dynamics. These destabilizing factors have been further studied in [36]– [34]. However, these issues are addressed with the $\mathcal{L}_1$ controller. E. Xargay directly revisits these benchmark problems in [44], and shows how the $\mathcal{L}_1$ control addresses them.

One major advantage of the $\mathcal{L}_1$ controller is the ability to quantify its performance and robustness using various metrics. In [41], performance bounds are given, which guarantee that the error between the system dynamics and closed-loop reference system will remain within a specific bound. This bound may be made arbitrarily small by increasing the tuning rate $\gamma$.

The $\mathcal{L}_1$ controller may also be analyzed with specific measures of robustness. In [45], the robustness of the $\mathcal{L}_1$ controller is studied using stability margin analysis. Namely, the time-delay margin may be used to give a quantitative bound on how much time delay can exist between the controller and plant without loss of stability. The time-delay margin for a nonlinear system is analogous to phase margins of a linear system. In [46], various adaptive controllers are studied, and their robustness is compared as well.

In summary, the $\mathcal{L}_1$ controller is introduced for the first time to a motor drive application, with the objective of achieving better performance and robustness, even in the presence of unmodeled dynamics. The control scheme used in this chapter follows the general case of an $\mathcal{L}_1$ controller as described in [23]. Simulations have been given in this chapter to demonstrate that this scheme is feasible for use in a motor drive application, and has several advantages over the MRAC control scheme which has been used before in

[3]Of course, this is only true when the unmodeled dynamics are stable, as discussed in section 4.1.

motor drive applications such as [25] and [28]. Namely, the $\mathcal{L}_1$ control scheme allows for arbitrarily fast parameter estimation without loss of robustness in the presence of unmodeled dynamics.

# CHAPTER 5

# IMPLEMENTED INDIRECT FIELD-ORIENTED $\mathcal{L}_1$ CONTROLLER

In this chapter, an implementable version of the $\mathcal{L}_1$ controller for an induction machine drive application is developed, and tested experimentally. First, a brief overview of the concepts involved with indirect field-oriented control (IFOC) will be given, along with any other necessary considerations for implementing an induction machine control system. Then, the actual control laws will be developed. Experimental tests are then performed to verify the claims about this controller, and show that it can be implemented. Finally, the actual switching methods used to actuate the current to the desired values will be discussed.

## 5.1 Basic Control Scheme Overview

This controller will use an IFOC strategy to regulate the speed and flux of the machine. The speed control loop is a PID controller. The control loop which regulates the machine flux is an $\mathcal{L}_1$ adaptive controller. It is much more important to show the performance of the $\mathcal{L}_1$ controller in the flux loop, because those dynamics will be affected the most by any variations or uncertainty of electric parameters.

Instead of assuming that $\lambda_{dr}$ and $\lambda_{qr}$ are known, we must now include the flux observer into the design of this controller. In the IFOC case, this is done by estimating only the magnitude of the rotor flux, rather than each component $\lambda_{dr}$ and $\lambda_{qr}$ individually. A separate estimator is used to estimate the angle $\rho$ at which this rotor flux is pointing. When the $d$-$q$ frame is aligned with this angle $\rho$, the flux will be in the direct component $\lambda_{dr}$. Thus, this forces the quadrature axis flux $\lambda_{qr}$ to be zero, as long as the angle $\rho$ is accurate. An estimator is designed to ensure that $\rho$ converges to the true value very quickly. As demonstrated in [47], this indirect strategy for

estimating the rotor flux magnitude and angle is stable and robust, even with some parameter uncertainty.

With the quadrature axis flux forced to zero, the system dynamics for an induction motor become more manageable, and linearization is possible. This is the basis for field-oriented control in general. More information about input-output linearization of an induction motor via field-oriented control may be found in [48].

This controller is based on the model of an induction motor given in (3.1)–(3.5). As with the direct field-oriented control scheme proposed in Chapter 4, the system is considered to be current-fed rather than voltage-fed. Thus, only (3.1)–(3.3) are used. The voltage to current dynamics given in (3.4)–(3.5) are fast, so it is possible to rapidly enforce the desired current with a large enough voltage input.

## 5.2 Development of $\mathcal{L}_1$ Adaptive IFOC Controller

In this section, each individual component of the implemented IFOC controller with $\mathcal{L}_1$ adaptive flux regulation is developed. First, the reference frame calculation which is used to convert three phase values to *d-q* reference frame quantities and back is discussed. Next, the equations used to estimate the rotor flux magnitude and direction will be given and discussed. For this control scheme, we are not assuming that the rotor flux vector can be measured, so we must discuss this estimation technique. The PID control law used to regulate motor speed will be given as well. Finally, the $\mathcal{L}_1$ adaptive controller responsible for regulating the rotor flux magnitude will be developed and discussed.

### 5.2.1 Reference Frame Calculation

To represent the machine state variables in the rotor flux frame, it is first necessary to define some important transformations. First, the three-phase stator current measurements must be transformed into two their representa-

tion in the new *d-q* rotating reference frame using

$$\begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin(\rho) & \sin\left(\rho - \frac{2\pi}{3}\right) & \sin\left(\rho + \frac{2\pi}{3}\right) \\ \cos(\rho) & \cos\left(\rho - \frac{2\pi}{3}\right) & \cos\left(\rho + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \tag{5.1}$$

Here, $\rho$ is the angle of the reference frame, $i_{as}$, $i_{bs}$, and $i_{cs}$ are the three-phase stator current measurements, and $i_{ds}$ and $i_{qs}$ are the direct and quadrature axis stator currents, respectively. This transformation, known as the Park transformation, is valid whenever the three currents are balanced, and can be found in [3]. Once the controller has determined the desired input currents, these must be transformed back from *d-q* coordinates into the original three-phase stator representation. This inverse Park transformation is represented as

$$\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} = \begin{bmatrix} \sin(\rho) & \cos(\rho) \\ \sin\left(\rho - \frac{2\pi}{3}\right) & \cos\left(\rho - \frac{2\pi}{3}\right) \\ \sin\left(\rho + \frac{2\pi}{3}\right) & \cos\left(\rho + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} \tag{5.2}$$

and may also be found in [3]. This new *d-q* reference frame will be referred to as the rotor flux frame or the $\rho$ frame, since the direct axis is at an angle $\rho$ from the stationary frame, and in line with the rotor flux vector.

### 5.2.2 Rotor Flux Magnitude and Angle Estimation

Now, we must get an estimate of the rotor flux magnitude and direction in order to establish the IFOC controller. The strategy taken by the *indirect* field-oriented controller (as opposed to the *direct* field-oriented controller) is to assume from the start that the quadrature rotor flux $\lambda_{qr} = 0$. With this assumption, the machine dynamics can be simplified. Recall the *d*-axis flux dynamics of Equation (3.2) on page 14. If $\lambda_{qr} = 0$, this equation may be simplified to the following form:

$$\dot{\lambda}_{dr} = \frac{-R_r}{L_r}\lambda_{dr} + \frac{R_r L_m}{L_r} i_{ds} \tag{5.3}$$

This equation may be used to run a real-time simulation of rotor flux magnitude in the controller, with the *d*-axis current as the input. The rotor flux

magnitude estimator used by this controller is

$$\dot{\hat{\lambda}}_{dr} = \frac{-\hat{R}_r}{\hat{L}_r}\hat{\lambda}_{dr} + \frac{\hat{R}_r\hat{L}_m}{\hat{L}_r}i_{ds} \tag{5.4}$$

where $\hat{R}_r$, $\hat{L}_r$, and $\hat{L}_m$ are estimates of the corresponding electric machine parameters. These estimates are measured off-line, and are not tuned adaptively in the controller. It is proven in [47] that the IFOC controller with this simple flux estimation will remain stable even with incorrect parameter estimates. However, if the estimates are wrong, there will be some steady state error between the estimated flux magnitude and actual flux magnitude.

Now, we must estimate the direction that the rotor flux is pointing with respect to the stationary frame. Again, with the assumption that $\lambda_{qr} = 0$, this task becomes easier. Recall the quadrature axis flux dynamics of (3.3). With $\lambda_{qr}$ equal to zero, these dynamics may be simplified to the algebraic constraint given in

$$0 = -\omega_{sl}\lambda_{dr} + \frac{R_r L_m}{L_r}i_{ds} \tag{5.5}$$

Since $\omega_{sl} = \dot{\rho} - P\omega_m$, this constraint can be further expanded and solved for $\dot{\rho}$, to give the following dynamic equation for rotor flux angle:

$$\begin{aligned}\omega_{sl} = \dot{\rho} - P\omega_m &= \frac{1}{\lambda_{dr}}\frac{R_r L_m}{L_r}i_{ds} \\ \dot{\rho} &= P\omega_m + \frac{1}{\lambda_{dr}}\frac{R_r L_m}{L_r}i_{ds}\end{aligned} \tag{5.6}$$

Equation (5.6) may be used to define the rotor flux angle estimator. The resulting estimator, with the same parameter estimates mentioned before, becomes

$$\dot{\hat{\rho}} = P\omega_m + \frac{1}{\hat{\lambda}_{dr}}\frac{\hat{R}_r\hat{L}_m}{\hat{L}_r}i_{ds} \tag{5.7}$$

This very simple estimator will track the actual flux angle very well as long as the parameter estimates are accurate. Once again, if these parameters are inaccurate, there will be some steady state error in the system.

### 5.2.3 Speed PID Control Loop

In order to keep this overall control scheme computationally simple, a PID controller is used to regulate the speed of the machine. Recalling (3.1) with $\lambda_{qr} = 0$, the speed dynamics may be influenced with the current input $i_{qs}$, independent of the direct axis current $i_{ds}$. The control law for the speed loop is defined as

$$i_{qs}(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt} \tag{5.8}$$

Here, $k_p > 0$ is the proportional gain, $k_i > 0$ is the integral gain, and $k_d \geq 0$ is the derivative gain. The error $e$ is defined as

$$e(t) = \omega_{ref}(t) - \omega_m(t) \tag{5.9}$$

For this controller, the following gains are used:

$$k_p = 0.4, \qquad k_i = 5 \times 10^{-4}, \qquad k_d = 0.3$$

### 5.2.4 $\mathcal{L}_1$ Adaptive Flux Control Loop

The $\mathcal{L}_1$ adaptive controller used to regulate the flux magnitude is very similar to the $\mathcal{L}_1$ controller developed in section 4.2.2. Now, it is slightly simplified due to the fact that we are going to assume $\lambda_{qr}$ is forced to be zero. Therefore, the term $\omega_{sl}\lambda_{qr}$ does not enter the flux dynamics.

The dynamics of the rotor flux used for this controller are

$$\dot{\lambda}_{dr}(t) = -\alpha\lambda_{dr}(t) + \beta i_{ds}(t) + \sigma_d(t) \tag{5.10}$$

where $\alpha$ and $\beta$ are unknown constant parameters, and $\sigma_d(t)$ is an unknown time-changing disturbance due to nonlinearity. In the original flux dynamics (3.2) are assumed to be linear, so the $\sigma_d$ term is not present. Now, however, we will assume that nonlinearity may be present. Thus, $\alpha$ and $\beta$ represent the linear parameters, while $\sigma_d$ is added to compensate for effects of nonlinearity.

An ideal reference model system is given as

$$\dot{\lambda}_{dr,m}(t) = \alpha_m \lambda_{dr,m}(t) - \alpha_m \lambda_{dr,ref}(t) - k_{if} \int_0^t \left(\lambda_{dr,m}(\tau) - \lambda_{dr,ref}(\tau)\right) d\tau \tag{5.11}$$

The objective of the adaptive controller is to mimic the behavior of this system. This time, the reference model system is second order, due to the inclusion of an integral term. This integral term serves to eliminate steady state error of the $\mathcal{L}_1$ controller. As mentioned before, the $\mathcal{L}_1$ system will track a closed-loop reference system similar to this desired model reference system, but not exactly the same. This new system is much more realistic, but could have some steady state error. The extra integral term in this open-loop reference model eliminates this problem. The values $\alpha_m = -60$ and $k_{if} = 80$ are used in the implementation of this controller.

The $d$-axis flux dynamics of the machine are now re-written as

$$\dot{\lambda}_{dr}(t) = \alpha_m \lambda_{dr}(t) + \beta i_{ds}(t) + \theta_d \lambda_{dr}(t) + \sigma_d(t) \tag{5.12}$$

to include the $\alpha_m \lambda_{dr}$ term. There, the new unknown parameter $\theta_d = -(\alpha_m + \alpha)$ is introduced. Next, a state predictor is defined with the exact same dynamics as the actual system, except that the unknown parameters are replaced with estimates, denoted with the hat symbol:

$$\dot{\hat{\lambda}}_{dr} = \alpha_m \hat{\lambda}_{dr} + \hat{\beta}_d i_{ds} + \hat{\theta}_d \lambda_{dr} + \hat{\sigma}_d \tag{5.13}$$

The goal of causing the state predictor to track the actual machine behavior is accomplished using the same tuning laws as the $\mathcal{L}_1$ controller described in section 4.2.2. Here, the unknown parameters $\hat{\beta}_d$ and $\hat{\theta}_d$ are tuned using the tuning laws given in (3.44) and (3.45) on page 31. The parameter $\hat{\sigma}_d$ is tuned using the law in Equation (4.17) on page 57. The Lyapunov analysis performed on pages 30–32 remains exactly the same for the first two unknown parameters, except that an extra term is added to the candidate Lyapunov function to include the error $\tilde{\sigma}_d = \hat{\sigma}_d - \sigma_d$. This candidate Lyapunov function is shown in (4.15) on page 57. Once again, all the errors are defined in (3.36)–(3.38) on page 30. The derivative of solutions along this Lyapunov function is given in (4.16) on page 57. With the tuning laws

mentioned above, this enforces the bound on the derivative of the Lyapunov function given in Equation (4.18) on page 57.

Now, Theorem 1 may be applied to guarantee that all error signals remain bounded, and that $e_d \to \infty$ as $t \to \infty$. Thus, we can conclude that the state predictor and plant dynamics will track one another as $t \to \infty$.

The control law for this $\mathcal{L}_1$ controller is almost identical to the one in section 4.2.2. The control laws used by the $\mathcal{L}_1$ flux control loop are represented as

$$i_{ds}(s) = -kD(s)\left(r_u(s) + \hat{\beta}_d i_{ds}\right) \tag{5.14}$$

$$\begin{aligned} r_u(t) = \hat{\theta}_d(t)\lambda_{dr}(t) + \hat{\sigma}_d(t) + \alpha_m \lambda_{dr,ref}(t) \\ + k_{if}\int_0^t \left(\lambda_{dr}(\tau) - \lambda_{dr,ref}(\tau)\right) d\tau \end{aligned} \tag{5.15}$$

The transfer function $D(s)$ is a strictly proper stable system yet to be designed, along with a strictly positive gain $k$.

Once again, this control law has internal feedback. This is because $i_{ds}$ appears appears in the right-hand side of (5.14). This control law may also be re-written in the frequency domain by solving for $i_{ds}$ as seen in

$$i_{ds}(s) = -\frac{\hat{C}(s)}{\hat{\beta}_d} r_u(s) \tag{5.16}$$

where $\hat{C}(s)$ is

$$\hat{C}(s) = \frac{k\hat{\beta}_d D(s)}{1 + k\hat{\beta}_d D(s)} \tag{5.17}$$

Substituting this control law into the state predictor (5.13) on page 83 results in the following dynamic behavior for the state predictor:

$$\begin{aligned} s\hat{\lambda}_{dr}(s) = \alpha_m \hat{\lambda}_{dr}(s) + \left(1 - \hat{C}(s)\right)\left(\hat{\theta}_d(s)\lambda_{dr}(s) + \hat{\sigma}_d(s)\right) \\ - \hat{C}(s)\left[\alpha_m \lambda_{dr,ref}(s) + k_{if}\frac{1}{s}\left(\lambda_{dr}(s) - \lambda_{dr,ref}(s)\right)\right] \end{aligned} \tag{5.18}$$

Notice that if $\hat{C}(s)$ were equal to one, the middle term of Equation (5.18) would be cancelled, so that the state predictor would assume the dynamics of the originally suggested reference system of Equation (5.11). However, $\hat{C}_d(s)$ will be a low pass filter, so that this cancellation does occur partially

at lower frequencies.

Since there is only partial cancellation of terms, the system is now tracking a different reference system than the suggested system of Equation (5.11). We now define a closed-loop reference system, which will actually be tracked. The resulting closed loop reference system, along with its feedback input, is

$$\dot{\lambda}_{dr,mod}(t) = \alpha_m \lambda_{dr,mod}(t) + \beta i_{ds}(t) + \theta_d \lambda_{dr,mod}(t) + \sigma_d \tag{5.19}$$

$$\lambda_{dr,mod}(0) = \lambda_{d0} \tag{5.20}$$

$$i_{ds,mod}(s) = -\frac{C(s)}{\beta} r_{u,mod}(s) \tag{5.21}$$

$$\begin{aligned} r_{u,mod}(s) = {} & \theta_d \lambda_{dr,mod}(s) + \sigma_d(s) + \alpha_m \lambda_{dr,ref}(s) \\ & + k_{if} \frac{1}{s} \left( \lambda_{dr,mod}(s) - \lambda_{dr,ref}(s) \right) \end{aligned} \tag{5.22}$$

Also, we define $H(s)$, $G(s)$, $C(s)$, and $r_0(s)$ for the purpose of analysis as

$$H(s) = \frac{s}{s^2 - \alpha_m s + k_{if} C(s)} \tag{5.23}$$

$$G(s) = H(s)(1 - C(s)) \tag{5.24}$$

$$C(s) = \frac{k\beta D(s)}{1 + k\beta D(s)} \tag{5.25}$$

$$r_0(s) = H(s)\lambda_{d0} \tag{5.26}$$

Substituting the control law (5.21) into the system (5.19), and simplifying using the transfer functions (5.23)–(5.26), yields the following new closed-loop reference system:

$$\begin{aligned} \lambda_{dr,mod}(s) = {} & G(s) \left[ \theta_d \lambda_{dr,mod}(s) + \sigma_d(s) \right] \\ & - C(s)H(s) \left( \alpha_m - \frac{1}{s} k_{if} \right) \lambda_{dr,ref}(s) + r_0(s) \end{aligned} \tag{5.27}$$

As long as $D(s)$ is chosen so that $C(s)$ is stable and proper, the transfer function $H(s)$ will be strictly proper and stable. Because of this, the $r_0(s)$ term on the far right of Equation (5.27) is stable, and will have a quickly diminishing bounded response. The system's response due to the desired flux value $\lambda_{dr,ref}$ is given in the middle term of Equation (5.27), $-C(s)H(s)\left(\alpha_m - \frac{1}{s}k_{if}\right)\lambda_{dr,ref}(s)$. Since $C(s)$ is stable and proper, and $H(s)$ is stable and strictly proper, this term is also a stable system. Thus, bounded

tracking signals $\lambda_{dr,ref}$ will not cause the closed-loop reference system to go unstable. The stability of this system was also verified via simulation using MATLAB, over the entire range of possible $\beta$ values.

Finally, in order to prove that the closed-loop reference system is stable, it is necessary to show that the first term, $G(s)\left[\theta_d \lambda_{dr,mod}(s) + \sigma_d(s)\right]$, is stable. Once again, Theorem 2 (the small-gain theorem) is used to accomplish this. Notice that this term has exactly the same form as the interconnected system described in Theorem 2. Thus, by invoking the small gain theorem, the closed-loop reference system will be stable as long as the condition

$$\|G(s)\|_{\mathcal{L}_1} L_d < 1 \tag{5.28}$$

is met, since the input term $\sigma_d(s)$ is bounded. The term $L_d$ is defined as

$$L_d = \max_{\theta_d \in \Theta} |\theta_d| \tag{5.29}$$

Here, $\Theta$ is the set of all possible values of $\theta_d$, based on the *a priori* known bounds.

In order to ensure that this condition is met, the transfer function $D(s)$, along with gains $k$ and $k_{if}$, must be chosen. The chosen form of the transfer function $D(s)$ is given as

$$D(s) = \frac{\omega_{nd}}{s + \omega_{nd}} \tag{5.30}$$

This is a stable, first order, low-pass filter, with cutoff frequency $\omega_{nd}$. The impulse response of the transfer function $G(s)$ was then simulated with various cutoff frequencies and gains, across the entire possible range of $\beta$. The magnitude of this impulse response was then integrated to get the $\mathcal{L}_1$ norm of $G(s)$. With the desired flux convergence rate $\alpha_m = -60$, the constant $L_d = 57$, since we know that $\alpha \geq 3$. It was found that with $\omega_{nd} = 5$, $k = 1$, and $k_{if} = 80$, the resulting norm of $G(s)$ satisfies $\|G(s)\|_{\mathcal{L}_1} \leq 0.0163$ for all possible values of $\beta$. This results in the product $\|G(s)\|_{\mathcal{L}_1} L \leq 0.93$, which satisfies the $\mathcal{L}_1$ norm condition given in Equation (5.28).

Now, we may invoke Theorem 2 to ensure that the closed-loop reference system is stable. The final form of this entire control loop is exactly the same as the one in Figure 4.5 on page 61, except that the control law is different.

In this controller, an integral term is added to the control law, as seen in Equations (5.14)–(5.15). The state predictor, given in Equation (5.13), will

then track the new closed loop reference system (5.27), which we have just proven is stable. The adaptive tuning laws given in Equations (3.44), (3.45), and (4.17) are used to adjust the parameter estimates in real time, in such a way that the error between the state predictor and actual system converges to zero. Thus, the actual system will track the closed-loop reference model.

### 5.2.5 Current Actuation

In previous sections, it was assumed that the controller could specify the desired currents $i_{ds}$ and $i_{qs}$, and that these currents would be instantly actuated into the induction motor. In order for this to be feasible in a real-world situation, some sort of control loop must be used to force the current to the values specified by the field-oriented controller.

This is accomplished here using hysteresis switching to regulate each phase current $i_{as}$, $i_{bs}$, and $i_{cs}$. Basically, hysteresis switching works by connecting a phase to a high positive voltage whenever the corresponding phase current is too low, and switching to a very high negative voltage whenever the current is too high. There is a certain switching band which determines how much current error there must be before a switch is made. The advantage of hysteresis switching is that it is very fast and accurate. However, it is possible for a hysteresis controller to cause extremely fast or slow switching frequencies. It is desirable to have a specific, known switching frequency for an inverter.

In order to correct this, the hysteresis switching signal (which takes values 0 or 1) is filtered using a low-pass filter. This filtered switching signal is then used as the duty ratio for a PWM channel. This PWM is then used to actually switch the inverter power electronics. With a PWM, the switching frequency will always be constant. However, the fast convergence rate and accuracy of hysteresis control is still present with this system. This technique is very similar to the one described in [49].

The desired direct axis stator current $i_{ds}$ will be specified by the flux-axis control loop, while the quadrature axis current $i_{qs}$ will be specified by the speed control loop. These desired currents are then transformed into three-phase values using (5.2). Each of these three currents is compared to the actual measured phase current, and this error signal is used to create the

hysteresis switching signal. Then, the hysteresis switching signal is filtered to obtain the PWM duty ratios.

## 5.3 Experimental Results

In this section, the entire control system developed in the previous section is implemented and tested experimentally. The tests are set up using a three-phase induction motor driven by an inverter. The switching signals for all three phases of this inverter are given using a TI TMS320F2812 digital signal processor (DSP). A 2048 slot encoder is used for speed measurement. Each phase current is sensed using a Hall-effect sensor, and sampled at a sampling frequency of 20 kHz. The 2-pole induction motor electrical parameters are estimated to be $\hat{R}_r = 2.23\ \Omega$, $\hat{L}_r = 0.2148$ H, and $\hat{L}_m = 0.2048$ H.

A hysteresis band of 0.02 A is used, and then filtered using a first-order low pass filter with a cutoff frequency of 200 Hz. This filtered signal is then used as the duty ratio for a 10 kHz PWM wave, which is actually used to switch the inverter power electronics. The time axis for each of the following figures is in seconds. In all of the following experiments, all adaptive parameters are tuned at the fast rate of $\gamma = 10,000$.

First, the speed loop is tested to ensure that the PID controller is sufficient to maintain speed. In the first test, shown in Figure 5.1, the machine is initially running at 1725 RPM, with a flux command of 300 mV-s, and a load torque of 0.1 N-m. Then, at 6.4 seconds, the load torque is increased to 0.3 N-m. As can be seen from Figure 5.1, the speed dynamics respond quickly, and the current commanded by the controller, $i_{qs}$, responds very quickly and remains stable. Next, the speed loop is tested with a change in desired speed. Figure 5.2 shows the response of the system when the speed command is changed from 1000 to 1725 RPM. One can see that this results in temporary weakening of the field. However, the flux controller does recover. Here, the load torque is constant at 0.3 N-m.

Next, the system's response to changes in flux command are tested. In Figure 5.3, $\lambda_{dr}$ is commanded to increase from 300 to 500 mV-s, while the speed is to maintain 1725 RPM. One can see that the speed control loop remains unaffected by the change in rotor flux magnitude. This time, the current $i_{ds}$ from the $\mathcal{L}_1$ flux control loop is plotted. As can be seen, this

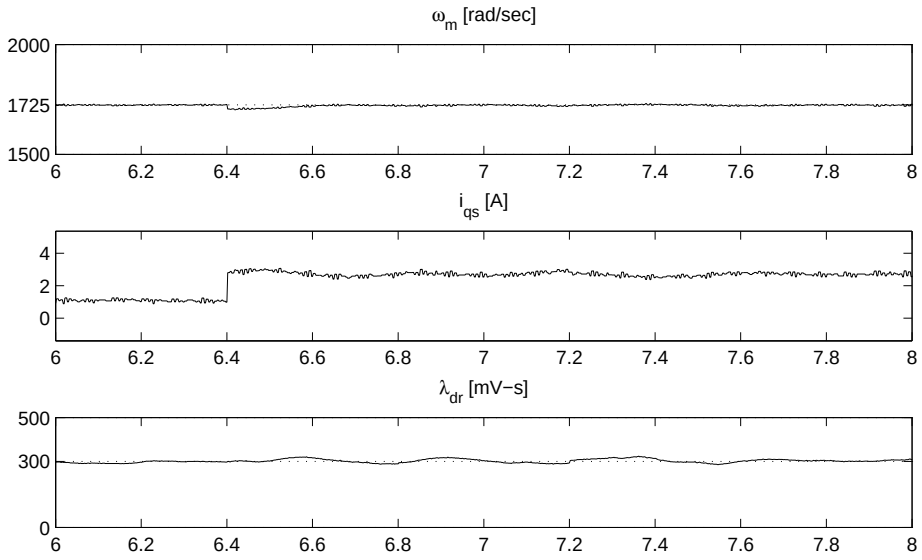


Figure 5.1: Increase in load torque, $x$ axis in seconds

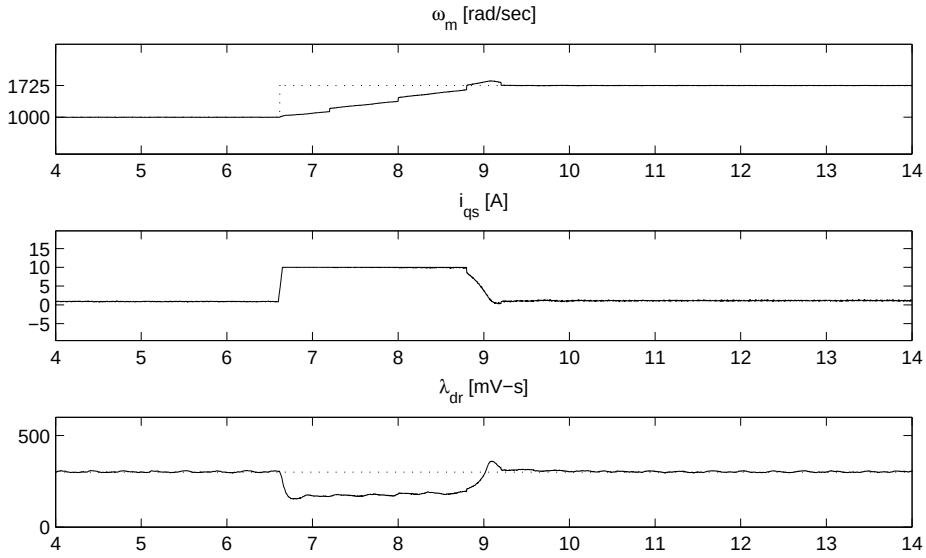


Figure 5.2: Increase in speed command, $x$ axis in seconds

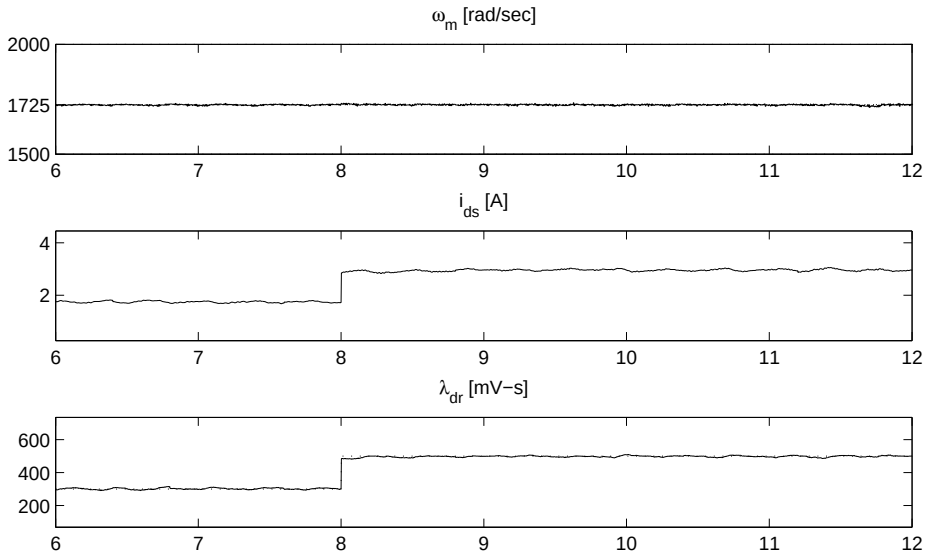


Figure 5.3: Increase in flux command, $x$ axis in seconds

current command is relatively smooth. The $\mathcal{L}_1$ controller does not try to over-actuate to compensate for high frequency dynamics such as switching dynamics. Similarly, Figure 5.4 shows the closed loop system response when the flux command is decreased from 400 to 200 mV-s at 3.6 seconds. For both of these tests, the machine is under 0.3 N-m of load torque.

The same tests are repeated again in order to measure the adaptive parameter estimates during these changes. Figure 5.5 shows the adaptive estimates when the flux command is increased from 300 to 500 mV-s at time $t = 5.2$ seconds. Then, Figure 5.6 shows the adaptive estimates when the flux command is decreased from 400 to 200 mV-s at time $t = 3$ seconds. In both situations, there are high frequency components in the parameter estimates. It is important to note that these do not enter the control channel $i_{ds}$, as seen in Figures 5.3 and 5.4.

It is also desirable to obtain the system response when field weakening is used as the speed command is increased. Figure 5.7 shows the system response when the flux command is decreased from 400 to 200 mV-s, as the speed command is simultaneously increased from 1000 to 1725 RPM. The current command $i_{ds}$ is again plotted to show the $\mathcal{L}_1$ controller's response. It is important to notice that it maintains a smooth current command, even though the adaptive parameters have very high frequency oscillations. This test is performed again to record these adaptive parameters in Figure 5.8.

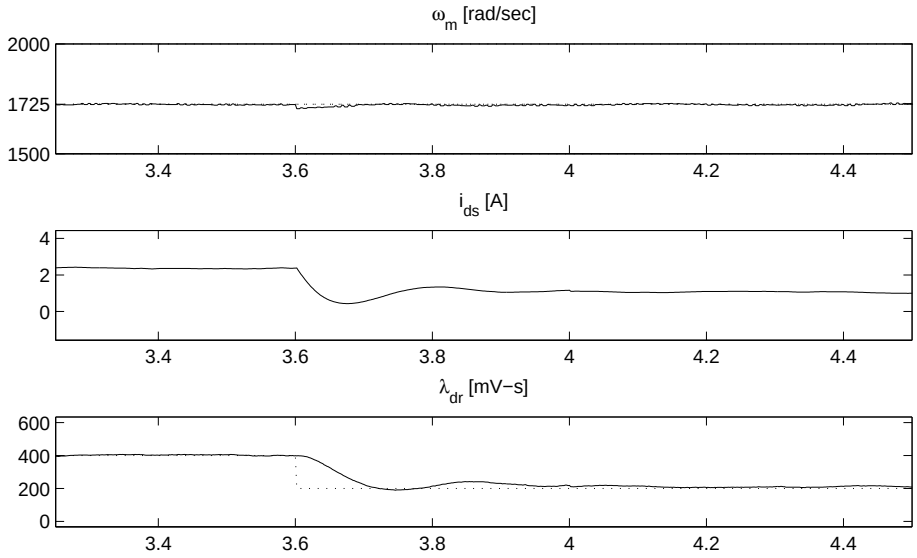


Figure 5.4: Decrease in flux command, $x$ axis in seconds

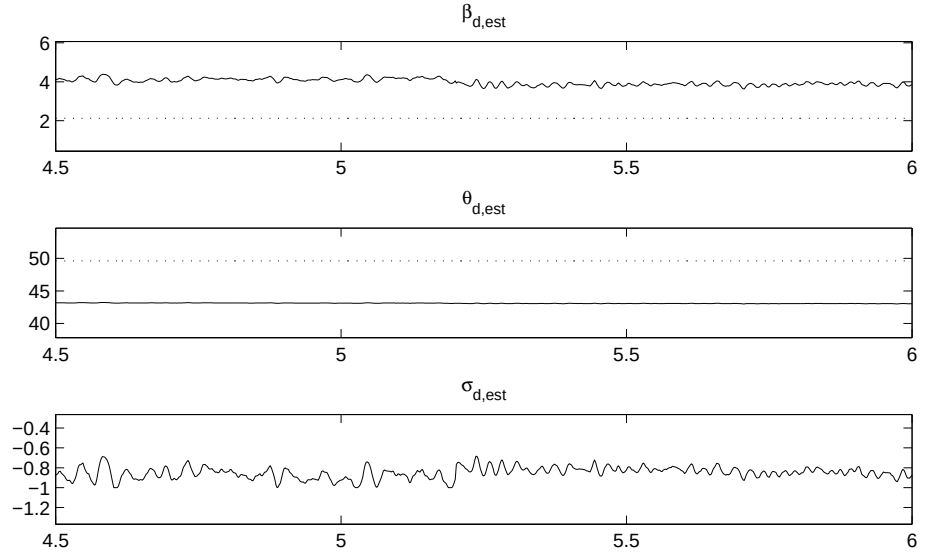


Figure 5.5: Parameter estimates during increase in flux command, $x$ axis in seconds

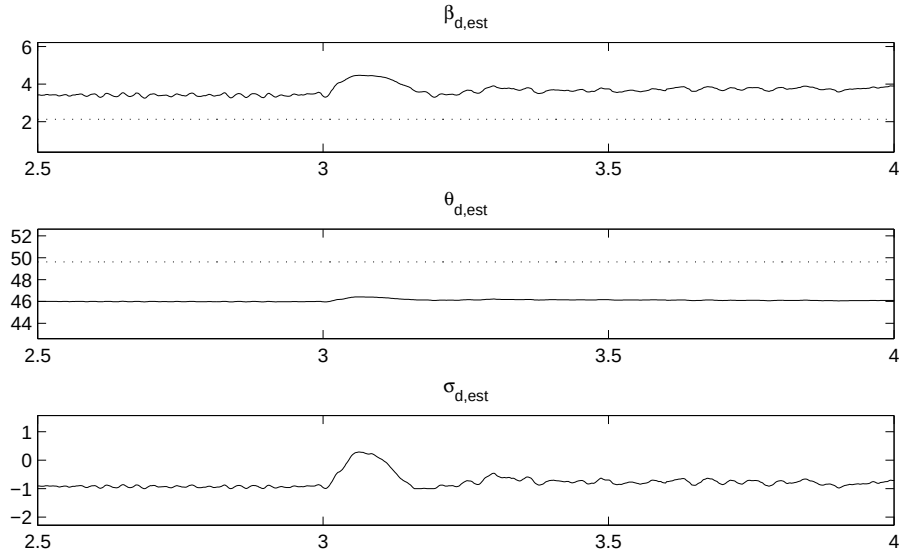


Figure 5.6: Parameter estimates during decrease in flux command, $x$ axis in seconds

Here, the speed is increased and flux decreased simultaneously at approximately time $t = 7.2$ seconds.

To ensure that the parameter estimates do cause the state predictor to track the actual system dynamics, the tests for flux changes are performed again, and this error is measured. In Figures 5.9–5.11, the error $e_d = \hat{\lambda}_{dr} - \lambda_{dr}$, where $\hat{\lambda}_{dr}$ is the state predictor output, and $\lambda_{dr}$ is the estimated machine flux. This error is given in mV-s. The flux $\lambda_{dr}$ itself is also plotted, along with the current command $i_{ds}$. Figure 5.9 gives the result when flux command is increased from 300 to 500 mV-s. Figure 5.10 gives the response with flux decreased from 400 to 200 mV-s. The response of speed increasing from 1000 to 1725 RPM while flux is decreased from 400 to 200 mV-s is shown in Figure 5.11.

One can see from Figures 5.9–5.11 that the state predictor tracks the estimated flux magnitude well. This indicates that the tuning laws are sufficient. The fast tuning rate of $\gamma = 10,000$ is used, which helps drive this error to zero very quickly. However, this fast tuning does not cause high frequency dynamics to enter the current command $i_{ds}$. This allows the $\mathcal{L}_1$ controller to maintain robustness while increasing performance.

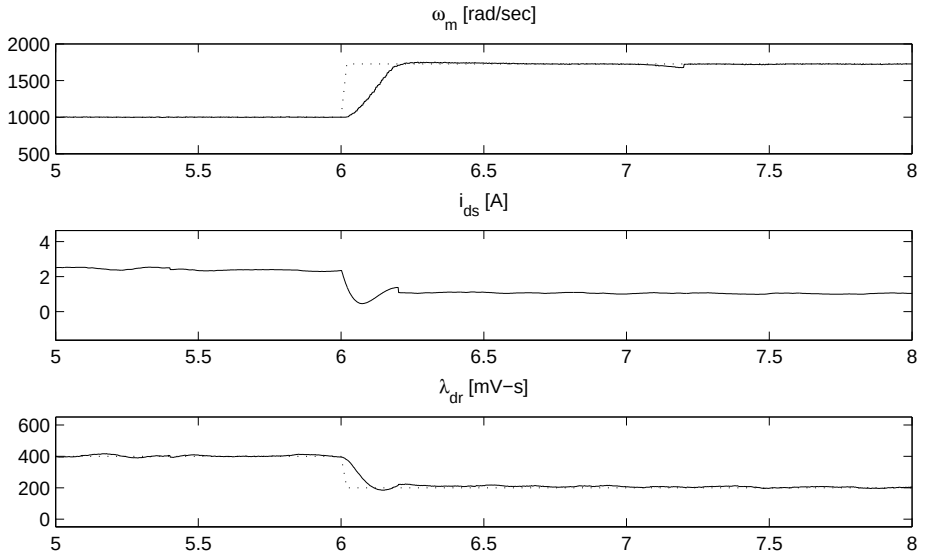


Figure 5.7: Speed increase during field weakening, $x$ axis in seconds

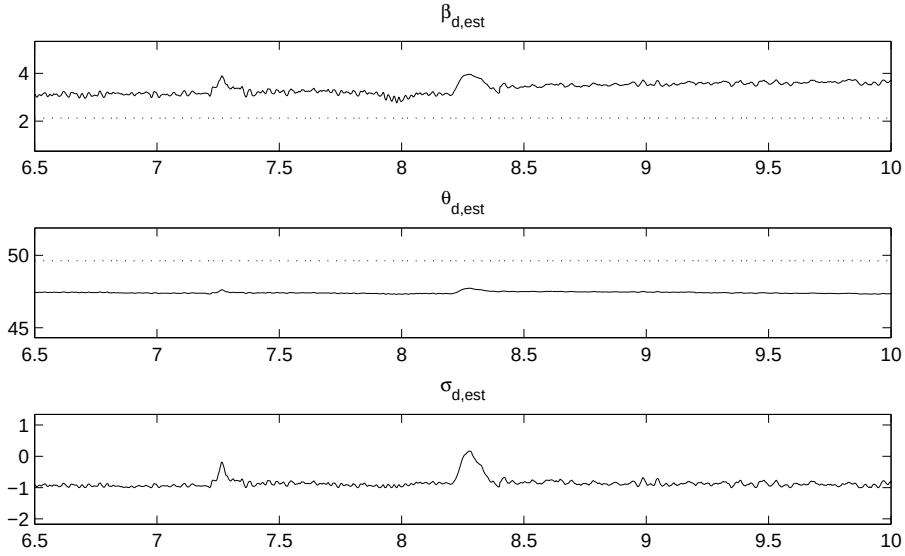


Figure 5.8: Parameter estimates during speed increase with field weakening, $x$ axis in seconds

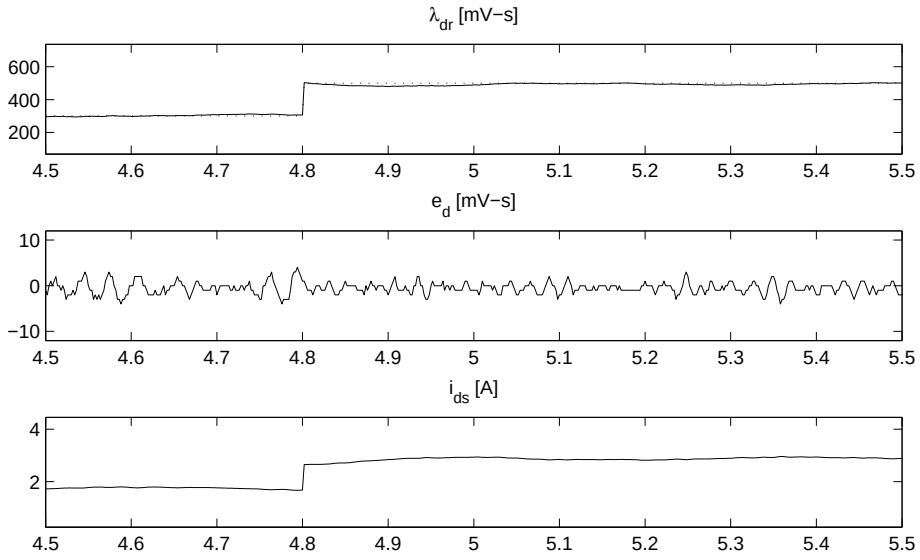


Figure 5.9: State predictor error during increase in flux command, $x$ axis in seconds

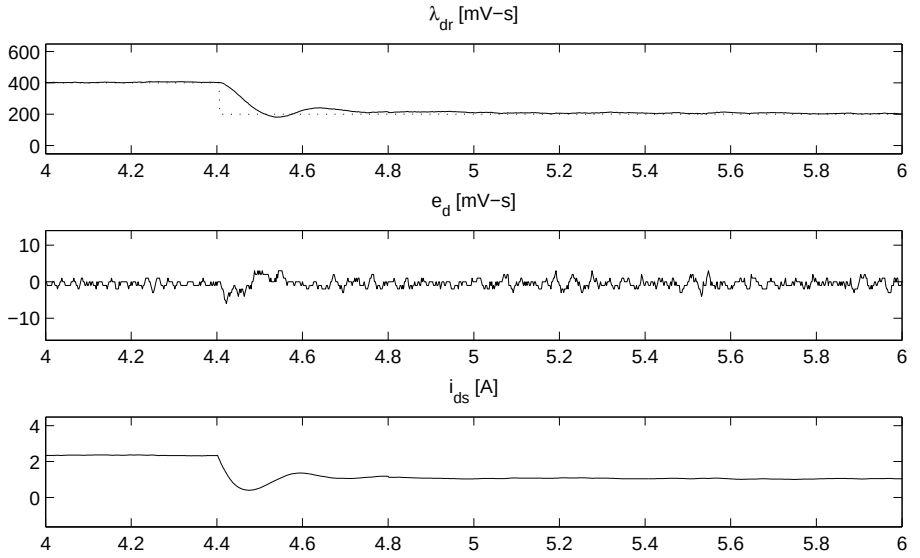


Figure 5.10: State predictor error during decrease in flux command, $x$ axis in seconds

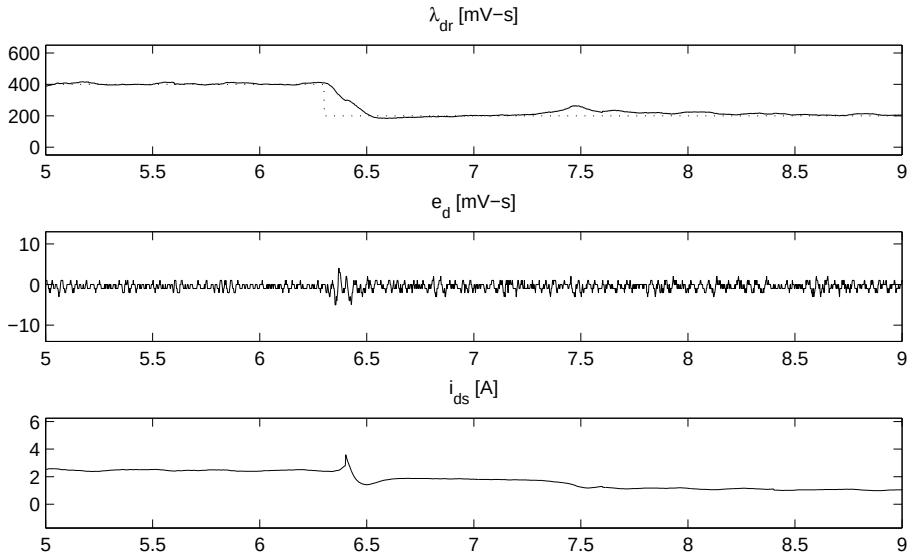


Figure 5.11: State predictor error during speed increase with field weakening, $x$ axis in seconds

# CHAPTER 6

# CONCLUSION

It has been shown that the $\mathcal{L}_1$ controller achieves very good performance and improved robustness over a model reference control system, even with very fast adaptive tuning. It was shown in Chapter 3 that the MRAC controller lacks robustness, specifically in the presence of unmodeled dynamics. These issues are then addressed with the $\mathcal{L}_1$ control scheme. While the relatively new $\mathcal{L}_1$ controller has been used in several aerospace applications, it has never been applied to a motor drive system. Chapter 5 demonstrates the feasibility of applying an $\mathcal{L}_1$ controller to an induction motor drive system.

The model of an induction motor has many uncertainties which can change during machine operation. Due to this, it was desirable to implement an adaptive controller to regulate the flux of the motor. With the proposed $\mathcal{L}_1$ control scheme, it is possible to adaptively control an induction motor so that transient performance is consistently desirable, even with uncertain parameters.

The proposed controller was proven to be stable using the small-gain theorem. This theorem ensures that the interconnected system created within this controller will be stable using conditions on its $\mathcal{L}_1$ norm. Further, this thesis tested the controller experimentally to verify its performance and stability. It is noted that this controller creates a commanded current input that is very smooth, even with high frequency oscillations in parameter estimates. These parameter estimates are only shown to remain bounded, and do not necessarily converge to their true values. They must only converge to values which cause the system to track the desired performance.

This control system causes the motor behavior to track the behavior of a reference model, which is specified as a goal for performance. Using adaptation of unknown parameters, the $\mathcal{L}_1$ controller is very effective at tracking this model behavior. This control scheme only assumes that the form of the system model is accurate at low frequencies, and that there may be fast, sta-

ble, unmodeled high frequency dynamics in the system. This more accurate and relaxed assumption results in the controller not expending unnecessary control effort to correct for these high frequency dynamics. Thus, the control inputs to the machine are smooth, and only contain lower frequency components.

Because the $\mathcal{L}_1$ controller does not create high frequency control channel inputs, this results in much better robustness. It allows the adaptive controller to increase the tuning rates arbitrarily high. Fast tuning rates for the adaptive estimates result in much better performance, since they will cause the system to converge to the desired model behavior much more quickly. With an MRAC controller, fast tuning will cause high frequency oscillations in the control channel, which may destabilize the system in the presence of unmodeled dynamics. Therefore, there is a tradeoff between performance and robustness with the MRAC controller. The $\mathcal{L}_1$ controller instead allows for arbitrarily fast tuning rates, as faster tuning is better for both performance and robustness.

Further work could be done to utilize a better flux observer, in order to increase the accuracy of this controller. The accuracy of this controller is only limited by that of the flux observer. This thesis did not propose any new advances in flux observation.

It may be possible to further develop this controller by combining the control technique with an observation technique using output feedback. Cao and Hovakimyan discuss output feedback for $\mathcal{L}_1$ control in [50] and [51]. Improving the flux observer for this controller could also be done using many of the adaptive flux observation techniques discussed in section 2.3.

# REFERENCES


[1] C. R. Sullivan and S. R. Sanders, "Models for induction machines with magnetic saturation of the main flux path," *IEEE Trans. Ind. Appl.*, vol. 31, no. 4, pp. 907–917, July/August 1995.

[2] X. Liu, G. Verghese, J. Lang, and M. Onder, "Generalizing the blondel-park transformation of electrical machines: necessary and sufficient conditions," *IEEE Trans. Circuits Syst.*, vol. 36, no. 8, pp. 1058–1067, August 1989.

[3] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*, 2nd ed. New York, NY: John Wiley & Sons, Inc., 2002.

[4] C. R. Sullivan, C. Kao, B. M. Acker, and S. R. Sanders, "Control systems for induction machines with magnetic saturation," *IEEE Trans. Ind. Electron.*, vol. 43, no. 1, pp. 142–152, February 1996.

[5] H. Hofmann, S. R. Sanders, and C. R. Sullivan, "Stator-flux-based vector control of induction machines in magnetic saturation," *IEEE Trans. Ind. Appl.*, vol. 33, no. 4, pp. 935–942, July/August 1997.

[6] W. Leonhard, *Control of Electrical Drives*, 3rd ed. New York, NY: Springer-Verlag, 2001.

[7] Y.-T. Kao and C.-H. Liu, "Analysis and design of microprocessor-based vector-controlled induction motor drives," *IEEE Trans. Ind. Electron.*, vol. 39, no. 1, pp. 46–54, February 1992.

[8] R. Ortega, C. Canudas, and S. I. Seleme, "Nonlinear control of induction motors: Torque tracking with unknown load disturbance," *IEEE Trans. Autom. Control*, vol. 38, no. 11, pp. 1675–1680, November 1993.

[9] I. Kanellakopoulos, P. T. Krein, and F. Disilvestro, "Nonlinear flux-observer-based control of induction motors," in *American Control Conference*, 1992, pp. 1700–1705.

[10] G. C. Verghese and S. R. Sanders, "Observers for flux estimation in induction machines," *IEEE Trans. Ind. Electron.*, vol. 35, no. 1, pp. 85–94, February 1988.

[11] Y. Hori, V. Cotter, and Y. Kaya, "A novel induction machine flux observer and its application to a high performance AC drive system," in *IFAC 10th Triennial World Congress*, 1987, pp. 363–368.

[12] V. Cotter, "Implementation of a sampled-data secondary flux observer for an induction motor and minimization of its sensitivity to parameter variation," M.S. thesis, University of Tokyo, 1987.

[13] Q. Zhang, "Adaptive observer for multiple-input-multiple-output (mimo) linear time-varying systems," *IEEE Trans. Autom. Control*, vol. 47, no. 3, pp. 525–529, March 2002.

[14] F. Jadot, F. Malrait, J. Moreno-Valenzuela, and R. Sepulchre, "Adaptive regulation of vector-controlled induction motors," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 3, pp. 646–657, May 2009.

[15] R. Marino, S. Peresada, and P. Tomei, "Output feedback control of current-fed induction motors with unknown rotor resistance," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 4, pp. 336–347, July 1996.

[16] R. Marino, S. Peresada, and P. Tomei, "Global adaptive output feedback control of induction motors with uncertain rotor resistance," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 967–983, May 1999.

[17] T. Noguchi, S. Kondo, and I. Takahashi, "Field-oriented control of an induction motor with robust on-line tuning of its parameters," *IEEE Trans. Ind. Appl.*, vol. 33, no. 1, pp. 35–42, January/February 1997.

[18] H. Kubota and K. Matsuse, "Speed sensorless field-oriented control of induction motor with rotor resistance adaptation," *IEEE Trans. Ind. Appl.*, vol. 30, no. 5, pp. 1219–1224, September/October 1994.

[19] H. Kubota, K. Matsuse, and T. Nakano, "DSP-based speed adaptive flux observer of induction motor," *IEEE Trans. Ind. Appl.*, vol. 29, no. 2, pp. 344–348, March/April 1993.

[20] J. Moreira, K. Hung, T. Lipo, and R. Lorenz, "A simple and robust adaptive controller for detuning correction in field oriented induction machines," *IEEE Trans. Ind. Appl.*, vol. 28, no. 6, pp. 1359–1366, Nov/Dec 1992.

[21] G. Espinosa and R. Ortega, "State observers are unnecessary for induction motor control," *Systems and Control Letters*, vol. 23, pp. 315–323, 1994.

[22] R. Ortega, P. J. Nicklasson, and G. Espinosa-Perez, "On speed control of induction motors," *Automatica*, vol. 32, no. 3, pp. 455–460, 1996.

[23] C. Cao and N. Hovakimyan, "Design and analysis of a novel $L_1$ adaptive control architecture with guaranteed transient performance," *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 586–591, March 2008.

[24] T.-S. Lee, C.-H. Lin, and F.-J. Lin, "An adaptive $h_\infty$ controller design for permanent magnet synchronous motor drives," *Control Engineering Practice*, vol. 13, no. 4, pp. 425–439, 2005.

[25] N. Goléa and A. Goléa, "High performance adaptive field oriented model reference control of current fed induction motor," *Journal of Electrical Engineering*, vol. 57, no. 2, pp. 78–86, 2006.

[26] N. Hovakimyan, "Robust adaptive control," May 2009, course lecture notes for MechSE 598 University of Illinois at Urbana Champaign.

[27] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, December 2001.

[28] Y. Wang, J. Yang, Z. Wang, Y. Hua, and J. Wang, "A novel mrac of induction motor," in *Power Electronics Specialists Conference*, June 2005, pp. 720–725.

[29] N. Hovakimyan and C. Cao, $L_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.

[30] M. Krstic, "Invariant manifolds and asymptotic properties of adaptive nonlinear stabilizers," *IEEE Trans. Autom. Control*, vol. 41, no. 6, pp. 817–829, June 1996.

[31] J.-B. Pomet and L. Praly, "Adaptive nonlinear regulation: Estimation from the Lyapunov equation," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 729–740, June 1992.

[32] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Autom. Control*, vol. 32, no. 2, pp. 134–145, February 1987.

[33] C. Rohrs, L. Valavani, M. Athans, and G. Stein, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Trans. Autom. Control*, vol. 30, no. 9, pp. 881–889, September 1985.

[34] S. Townley, "An example of a globally stabilizing adaptive controller with a generically destabilizing parameter estimate," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2238–2241, November 1999.

[35] B. D. O. Anderson, "Failures of adaptive control theory and their resolution," *Communications in Information and Systems, International Press*, vol. 5, no. 1, pp. 1–20, 2005.

[36] B. D. Anderson and A. Dehghani, "Challenges of adaptive control—past, permanent and future," *Annual Reviews in Control*, vol. 32, pp. 123–135, 2008.

[37] K. J. Åström, "Analysis of rohrs counterexamples to adaptive control," in *The 22nd IEEE Conference on Decision and Control*, 1983, pp. 982–987.

[38] K. Åström, "Interactions between excitation and unmodeled dynamics in adaptive control," in *Proceedings of 23rd Conference on Decision and Control*, 1984, pp. 1276–1281.

[39] P. A. Ioannou and P. V. Kokotovic, "Robust redesign of adaptive control," *IEEE Trans. Autom. Control*, vol. 29, no. 3, pp. 202–211, March 1984.

[40] I. Gregory, E. Xargay, C. Cao, and N. Hovakimyan, "Flight test of $L_1$ adaptive controller on the NASA AirSTAR flight test vehicle," in *AIAA Guidance, Navigation and Control Conference*, Toronto, Ontario, Aug. 2–5 2010, AIAA-2010-8015.

[41] C. Cao and N. Hovakimyan, "$L_1$ adaptive controller for systems with unknown time-varying parameters and disturbances in the presence of non-zero trajectory initialization error," *International Journal of Control*, vol. 81, no. 7, pp. 1147–1161, July 2008.

[42] A. B. Gostin, "A matlab-based toolbox for the simulation and design of $L_1$ adaptive controllers," M.S. thesis, University of Illinois at Urbana-Champaign, 2010.

[43] C. Cao and N. Hovakimyan, "$L_1$ adaptive controller for a class of systems with unknown nonlinearities: Part i." in *American Control Conference*, Seattle, WA, 2008, pp. 4093–4098.

[44] E. Xargay, N. Hovakimyan, and C. Cao, "Benchmark problems of adaptive control revisited by $L_1$ adaptive control," in *17th Mediterranean Conference on Control and Automation*, Thessalonika, Greece, June 2009, pp. 31–36.

[45] C. Cao and N. Hovakimyan, "Stability margins of $L_1$ adaptive controller," *IEEE Trans. Autom. Control*, vol. 55, no. 2, pp. 480–487, 2010.

[46] E. Kharisov, N. Hovakimyan, and K. Åström, "Comparison of several adaptive controllers according to their robustness metrics," in *AIAA Guidance, Navigation and Control Conference*, Toronto, Ontario, Aug. 2–5 2010, AIAA-2010-8047.

[47] P. D. Wit, R. Orgega, and I. Mareels, "Indirect field-oriented control of induction motors is robustly globally stable," *Automatica*, vol. 32, no. 10, pp. 1393–1402, October 1996.

[48] M. Bodson, J. Chiasson, and R. Novotnak, "High performance induction motor control via input-output linearization," *IEEE Control Systems Magazine*, vol. 14, no. 4, pp. 25–33, 1994.

[49] L. Malesani and P. Tenti, "A novel hysteresis control method for current-controlled voltage-source PWM inverters with constant modulation frequency," *IEEE Trans. Ind. Appl.*, vol. 26, no. 1, pp. 88–92, January/February 1990.

[50] C. Cao and N. Hovakimyan, "$L_1$ adaptive output feedback controller for systems of unknown dimension," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 815–821, April 2008.

[51] C. Cao and N. Hovakimyan, "$L_1$ adaptive output feedback controller for non-strictly positive real systems: Missile longitudinal autopilot design," *AIAA Journal of Guidance, Control and Dynamics*, vol. 32, no. 3, pp. 717–726, 2009.